Сыса А. Г., Дудинская Р. А.

МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ

Учреждение образования
«Международный государственный
экологический институт имени А. Д. Сахарова»
Белорусского государственного университета

Сыса А. Г., Дудинская Р. А.

МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ

Учебно-методическое пособие

Минск
«ИВЦ Минфина»
2018

УДК 004:57:61(075.8)
ББК 32.97+28+5я7
С95

Р е ц е н з е н т ы :
кандидат экономических наук, доцент, заведующий кафедрой экономической информатики Белорусского государственного университета *Д. А. Марушко*;
кандидат физико-математических наук, доцент, заведующий кафедрой экологических информационных систем Международного государственного экологического института имени А. Д. Сахарова Белорусского государственного университета *В. А. Иванюкович*

Сыса, А. Г.

С95 Методы обработки информации : учебно-методическое пособие / А. Г. Сыса, Р. А. Дудинская. – Минск : ИВЦ Минфина, 2018. – 219 с.

ISBN 978-985-7205-41-7.

В пособие включены образовательные элементы для решения профессиональных задач с помощью прикладного программного обеспечения. Даны методичекиские указания по выполнению лабораторных работ, контрольные вопросы для усвоения учебного материала и задания для самоконтроля. Предлагается библиографический список источников для углубленного изучения вопроса.

Предназначается студентам специальности I ступени высшего образования 33 01 05 «Медицинская экология» для практического использования и оценки преимущества применения информационных технологий при обработке статистического материала.

УДК 004:57:61(076)
ББК 32.97+28+5я7

ISBN 978-985-7205-41-7

СОДЕРЖАНИЕ

ЧАСТЬ 1
ОСНОВЫ ЭПИДЕМИОЛОГИЧЕСКОГО АНАЛИЗА

Лабораторная работа № 1
Скрининговые исследования в эпидемиологии

Цель: научиться рассчитывать показатели, характеризующие диагностическую ценность теста.

Диагностическая ценность теста

Для определения чувствительности и специфичности метода с помощью четырехклеточной таблицы-макета сопоставляются результаты скрининга и полного обследования.

Таблица 1.1

Макет таблицы для определения чувствительности и специфичности скрининг метода

Наименование		Данные полного обследования		Всего
		положительные ответы	отрицательные ответы	
Скрининг	положительные ответы	a	b	a+b
	отрицательные ответы	c	d	c+d
	всего	a+c	b+d	a+b+c+d

a – истинноположительные ответы, которые совпадают как положительные при скрининговом и полном обследовании;

d – истинноотрицательные ответы, которые совпадают как отрицательные при скрининговом и полном обследовании;

b – ложноположительные;

c – ложноотрицательные.

Чувствительность метода – способность выявить большую часть истинноположительных ответов.

$$\text{Чувствительность (сензитивность)} = \frac{a}{a+c} \times 100. \qquad (1.1)$$

Специфичность метода – способность относительно редко давать ложноположительные ответы.

$$\text{Специфичность} = \frac{d}{b+d} \times 100. \quad (1.2)$$

Идеальный метод обладает высокой чувствительностью и высокой специфичностью, то есть позволяет выделить максимальное число больных и крайне редко дает ложноположительную информацию (здоровый ложно оценивается как больной).

Воспроизводимость результатов оценивается по показателям соответствия и воспроизводимости при сопоставлении данных двух обследований, проведенных в одинаковых условиях:

$$\text{показатель соответствия} = \frac{a+d}{a+b+c+d} \times 100, \quad (1.3)$$

$$\text{показатель воспроизводимости} = \frac{a}{a+b+c} \times 100. \quad (1.4)$$

Таблица 1.2

Оценочная шкала показателей воспроизводимости

Оценка	Показатель соответствия, %	Показатель воспроизводимости, %
Хорошая	90–100	75–100
Средняя	75–89	50–74
Неудовлетворительная	75	50

Задание 1

Профилактический осмотр населения, проживающего на территории Чечерского р-на Гомельской обл., включал консультацию эндокринолога, который при обнаружении патологии щитовидной железы направлял пациента на ультразвуковое исследование. После подтверждения наличия патологии пациент проходил дальнейшее обследование, включающее определение концентрации тиреоидных гормонов в крови для постановки уточненного диагноза. В табл. 1.3 приведены результаты проведения скрининга УЗИ щитовидной железы и полного обследования тиреоидной системы (на примере работы диспансера НИИ РМ МЗ РБ).

Таблица 1.3

Результаты проведения скрининга УЗИ щитовидной железы

Наименование		Полное обследование тиреоидной системы		Всего
		положительные ответы	отрицательные ответы	
Скрининг УЗИ	положительные ответы	580	30	610
	отрицательные ответы	4	86	90
	всего	584	116	700

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 2

Для выявления на медицинских осмотрах лиц с конкретным заболеванием был разработан простой и недорогой метод скрининга. Для определения чувствительности и специфичности метода он был испытан на 200 пациентах, которые для уточнения диагноза прошли одновременное и тщательное клиническое обследование (табл. 1.4).

Таблица 1.4

Результаты проведения скрининга по новой методике

Наименование		Наличие болезни согласно методу скрининга		
		да	нет	всего
Наличие болезни согласно клиническому обследованию	да	60	20	80
	нет	40	80	120
	всего	100	100	200

Рассчитайте чувствительность и специфичность, показатели соответствия и воспроизводимости. По шкале оцените полученные показатели. Сделайте выводы.

Задание 3

Профилактический осмотр населения, проживающего на территории Краснопольского р-на Могилевской обл., включал ультразвуковое исследование. Для уточнения диагноза пациенты проходили дальнейшее обследование, при котором определялись концентрации тиреоидных гормонов

в крови для постановки уточненного диагноза. В табл. 1.5 приведены результаты проведения скрининга УЗИ щитовидной железы и полного обследования тиреоидной системы.

Таблица 1.5

Результаты проведения скрининга УЗИ щитовидной железы и полного обследования тиреоидной системы

		Полное обследование тиреоидной системы		Всего
		положительные ответы	отрицательные ответы	
Скрининг УЗИ	положительные ответы	340	50	390
	отрицательные ответы	7	90	97
	всего	347	140	487

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 4

Профилактический осмотр женщин, проживающих в г. Корма Гомельской обл., включал консультацию гинеколога, который при обнаружении патологии направлял пациентку на ультразвуковое исследование. После подтверждения наличия патологии пациентка проходила дальнейшее обследование, включающее цитологическое исследование для постановки уточненного диагноза. В табл. 1.6 приведены результаты проведения скрининга органов малого таза и цитологического обследования.

Таблица 1.6

Результаты проведения скрининга органов малого таза и цитологического обследования

		Цитологическое исследование		Всего
		положительные ответы	отрицательные ответы	
Скрининг органов малого таза с применением УЗИ	положительные ответы	240	150	390
	отрицательные ответы	70	60	130
	всего	310	140	450

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 5

1379 подростков, проживающих в г. Чечерск Гомельской обл., прошли УЗИ щитовидной железы и были направлены на дальнейшее исследование, включающие определение концентрации тиреоидных гормонов в сыворотке крови. В табл. 1.7 приведены результаты проведения скрининга на наличие АИТ.

Таблица 1.7

Результаты проведения скрининга на наличие АИТ

		Результат по данным тиреоидного исследования	
		положительные ответы	отрицательные ответы
УЗИ щитовидной железы	положительные ответы	700	180
	отрицательные ответы	550	140

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 6

800 детей, посещающих ДДУ в г. Ветка Гомельской обл., прошли осмотр нервопатолога и были направлены на дальнейшее исследование, включающие специальные тесты. В табл. 1.8 приведены результаты проведения скрининга.

Таблица 1.8

Результаты проведения скрининга

		Результат по данным теста	
		положительные ответы	отрицательные ответы
Осмотр невропатолога	положительные ответы	450	80
	отрицательные ответы	220	50

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 7

1300 рабочих-строителей, постоянно работающих на открытом воздухе, были обследованы на наличие заболеваний кожи и были направлены на дальнейшее исследование, включающие и цитологическое. В табл. 1.9 приведены его результаты.

Таблица 1.9

Результаты цитологического исследования

		Результат по данным цитологического исследования	
		положительные ответы	отрицательные ответы
Осмотр онколога	положительные ответы	900	150
	отрицательные ответы	150	100

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 8

1500 детям из Буда-Кошелевского и Быховского р-ов Гомельской обл., рожденных у матерей, отселенных из зоны отчуждения, было проведено УЗИ щитовидной железы. Они были направлены на дальнейшее исследование, включающие специальные тесты. В табл. 1.10 приведены результаты проведения скрининга.

Таблица 1.10

Результаты проведения скрининга

		Результат по данным теста	
		положительные ответы	отрицательные ответы
УЗИ	положительные ответы	900	200
	отрицательные ответы	250	150

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 9

1000 рабочих кузнечного цеха завода карданных валов (г. Гродно) были обследованы на наличие патологий органа слуха и направлены дальнейшее исследование, включающие специальные тесты. В табл. 1.11 приведены результаты проведения скрининга.

Таблица 1.11

Результаты проведения скрининга

		Результат по данным теста	
		положительные ответы	отрицательные ответы
Осмотр ЛОРа	положительные ответы	600	150
	отрицательные ответы	150	100

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 10

1350 подростков из Витебской обл., проживающих вблизи ЛЭП, были опрошены на наличие у них аллергических реакций и были направлены на дальнейшее исследование, включающие специальные тесты. В табл. 1.12 приведены результаты исследования.

Таблица 1.12

Результаты исследования

		Результат по данным теста	
		положительные ответы	отрицательные ответы
Опрос	положительные ответы	800	150
	отрицательные ответы	250	150

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 11

1000 жителям г. Ветка и Ветковского р-на Гомельской обл. был проведен экспресс-анализ мочи. Они были направлены на дальнейшее исследование, включающие УЗИ-исследование и специальные тесты. В табл. 1.13 приведены результаты проведения.

Таблица 1.13

Результаты исследований

		Результат по данным УЗИ и теста	
		положительные ответы	отрицательные ответы
Экспресс-анализ	положительные ответы	500	100
	отрицательные ответы	350	50

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 12

1200 рабочих закройного цеха обувной фабрики (г. Витебск) были обследованы на наличие заболеваний кожи и направлены на цитологическое исследование. В табл. 1.14 приведены результаты проведения.

Таблица 1.14

Результаты исследований

		Результат по данным теста	
		положительные ответы	отрицательные ответы
Осмотр ЛОРа	положительные ответы	750	200
	отрицательные ответы	250	100

Рассчитайте распространенность, чувствительность и специфичность, показатели соответствия и воспроизводимости. Оцените по шкале. Сделайте выводы.

Задание 13

Для выявления на медицинских осмотрах лиц с конкретным заболеванием был разработан простой и недорогой метод скрининга. Для определения чувствительности и специфичности метода он испытывался на 800 пациентах, которые прошли одновременное и тщательное клиническое обследование для уточнения диагноза.

Таблица 1.15

Результаты исследований

		Наличие болезни согласно методу скрининга		
		да	нет	всего
Наличие болезни согласно клиническому обследованию	да	400	150	550
	нет	200	50	250
	всего	600	200	800

Рассчитайте чувствительность и специфичность, показатели соответствия и воспроизводимости. По шкале оцените полученные показатели. Сделайте выводы.

Контрольные вопросы

1. Определение, цели, направления, целесообразность применения скрининговых исследований.

2. Методика расчета чувствительности, специфичности, воспроизводимости метода с использованием таблицы 2х2.

3. Как проводится оценка прогностической значимости теста.

4. Какова методика расчета отношения правдоподобия.

5. Цель и методика расчета скорректированного значения распространенности.

Список библиографических источников

1. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

2. *Орлов А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

Лабораторная работа № 2
Расчет относительного риска

Цель: используя входную информацию, научиться рассчитывать относительный риск заболеть для лиц экспонированной группы по сравнению с лицами, относящимися к неэкспонированной группе и интерпретировать полученные данные.

Представление данных с помощью таблицы 2х2

Строки представляют собой два уровня воздействия, а столбцы – два уровня статуса болезни.

Таблица 2.1

Представление данных с помощью таблицы 2х2

		Заболевание		Всего
		есть	нет	
Воздействие	есть	a	b	a+b
	нет	c	d	c+d
	всего	a+c	b+d	a+b+c+d

Сумма всех четырех клеток представляет собой размер всей выборки.
Заболеваемость в группе, подвергшейся воздействию:

$$I_e = \frac{a}{a+b}. \qquad (2.1)$$

Заболеваемость в группе, не подвергшейся воздействию:

$$I_0 = \frac{c}{c+d}. \qquad (2.2)$$

Формула расчета относительного риска для данных в когортном исследовании, представленных в виде таблицы 2х2, выглядит следующим образом:

$$\text{ОР} = \frac{I_e}{I_0} = \frac{a/(a+b)}{c/(c+d)}. \qquad (2.3)$$

В исследованиях типа «случай–контроль», где участники выбираются на основе статуса заболевания, обычно невозможно рассчитать показатель развития заболевания в зависимости от наличия или отсутствия воздействия. Относительный риск в исследованиях «случай–контроль» может быть оценен путем расчета отношения неравенства воздействия среди случаев и среди контролей. Это отношение (ОН) выражается следующей формулой:

$$ОН = \frac{a/b}{c/d}, \quad (2.4)$$

Величина относительного риска должна характеризоваться также величиной доверительного интервала. Для расчета ДИ для ОР существует альтернативная формула расчета ДИ:

$$ДИ = ОР^{(1\pm z/\chi)}, \quad (2.5)$$

где z – значение стандартного нормального распределения, связанное с требуемым доверительным уровнем,

$\chi = \sqrt{\chi^2}$ – из теста на статистическую значимость (критерий Пирсона).

Задание 1

В исследовании «случай–контроль» для выявления связи между курением сигарет и инфарктом миокарда были получены результаты, помещенные в табл. 2.2.

Таблица 2.2

Результаты исследования

Некурящие		Курящие (число пачек в день)		
		0,5	***1***	***2***
Случаи	27	9	39	18
Контроли	2100	710	1825	605

Рассчитайте относительный риск развития инфаркта миокарда по сравнению с составленной из некурящих группой сравнения для курящих в день – 0, 5, 1, 2 пачки. Сделайте выводы.

Задание 2

Из 595 пациентов, которым было сделано переливание крови, и 712 без переливания, 75 и 16 заболели гепатитом в течение 5 лет последующего наблюдения в исследовании для оценки риска этого заболевания.

Какого типа это исследование. Рассчитайте относительный риск заболевания гепатитом в этих группах и доверительные интервалы к нему. Сделайте выводы.

Задание 3

В исследовании связи между острым лейкозом и персональной экспозицией к нефтепродуктам применялась следующая методика. В клинике было зарегистрировано последовательно 50 случаев заболевания и по производственным данным определена экспозиция больных к нефтепродуктам. Кроме того, были зарегистрированы и обследованы 100 пациентов,

обращающихся с жалобами на другие незлокачественные заболевания крови (табл. 2.3).

Таблица 2.3

Результаты исследования

	Острый лейкоз	Незлокачественные заболев
Экспонированные	18	10
Неэкспонированные	32	90

Какого типа это исследование? Есть ли связь между исследуемой экспозицией и данным заболеванием? Произведите количественные сравнения частоты заболевания в экспонированной и неэкспонированной группах.

Задание 4

У мужчин с раком гортани и представителей контрольной группы исследовалось употребление спиртных напитков (табл. 2.4).

Таблица 2.4

Результаты исследования

Употребление алкоголя (единиц/месяц)	Случаи	Контроли
0	38	87
Менее 3	41	51
3–30	111	104
Более 30	179	116

Рассчитайте риск развития гортани для каждого уровня употребления алкоголя по сравнению с непьющими. Сделайте выводы.

Задание 5

В исследовании «случай–контроль» изучалась связь между употреблением жевательной резинки фирм и возникновением кариеса зубов (табл. 2.5).

Таблица 2.5

Результаты исследования

	Экспонированные		Неэкспонированные	
	случай	контроль	случай	контроль
Фирма А	13	85	5	50
Фирма В	23	43	3	60

Вычислите риск развития кариеса зубов при употреблении жевательной резинки. Сделайте выводы.

Задание 6

В исследовании «случай–контроль» изучалась связь между условиями проживания в промышленном районе вблизи предприятий и частотой возникновения аллергических заболеваний у детей (табл. 2.6).

Таблица 2.6

Результаты исследований

	Проживающие вблизи промышленных предприятий		Проживающие вдали промышленных предприятий	
	случай	контроль	случай	контроль
До 1 года	45	20	25	15
1–4 года	30	25	35	10
5–7 лет	30	15	15	10

Вычислите риск развития аллергических заболеваний в каждой из возрастных групп. Сделайте выводы.

Задание 7

В исследовании связи между раком кожи и профессиональной экспозицией к дубильным веществам на кожзаводе было зарегистрировано в заводской поликлинике последовательно 60 случаев заболеваний мужчин раком кожи и определена экспозиция больных к дубильным веществам. Таким же образом были зарегистрированы и обследованы 90 мужчин пациентов, обращавшихся с жалобами на другие незлокачественные заболевания кожи (табл. 2.7).

Таблица 2.7

Результаты исследования

	Рак кожи	Незлокачественные заболевания кожи
Экспонированные мужчины	18	10
Неэкспонированные мужчины	42	80

Какого типа это исследование? Произведите количественное сравнение частоты заболевания в экспонированной и неэкспонированной группах Сделайте выводы.

Задание 8

В исследовании случай контроль для выявления связи между курением сигарет и раком легкого были получены результаты, помещенные в табл. 2.8.

Таблица 2.8

Результаты исследования

Некурящие		Курящие (число пачек в день)		
		0,5	***1***	***2***
Случаи	31	9	39	18
Контроли	3500	700	1800	600

Рассчитайте относительный риск рака легкого по сравнению с составленной из некурящих группой сравнения для курящих в день – 0,5, 1, 2 пачки. Сделайте выводы.

Задание 9

В исследовании связи между частотой возникновения ОРВИ и проживанием в экологически неблагоприятной обстановке было зарегистрировано в поликлинике № 14 г. Минска 60 случаев заболеваний детей, проживающих вблизи промышленных предприятий и 90 – вдали, а в поликлинике № 6 – 30 и 50 соответственно (табл. 2.9).

Таблица 2.9

Результаты исследования

	Вблизи промышленных предприятий	Вдали промышленных предприятий
Случаи п-ка № 14	18	10
Контроли п-ка № 14	42	80
Случаи п-ка № 6	10	20
Контроли п-ка № 6	20	30

Какого типа это исследование? Произведите количественное сравнение частоты заболевания в экспонированной и неэкспонированной группах. Сделайте выводы.

Задание 10

У мужчин с раком легкого и представителей контрольной группы исследовалось употребление сигарет (табл. 2.10).

Таблица 2.10

Результаты исследования

Употребление сигарет (пачек)	Случаи	Контроли
0	28	77
0,5	42	44
1	113	98
более 1	157	116

Рассчитайте риск развития рак легкого для каждого уровня употребления сигарет по сравнению с некурящими. Сделайте выводы.

Задание 11

В проведенном в Финляндии исследовании заболеваемости и смертности от инфаркта миокарда, сравнивались женатые и одинокие мужчины. Были получены следующие результаты.

Таблица 2.11

Результаты исследования заболеваемости и смертности от инфаркта миокарда мужчин 40–64 лет (стандартизованные по возрасту коэффициенты на 100 000 человеко-лет)

Семейное положение	Заболеваемость	Смертность
Женатый	1371	498
Одинокий	1228	683

Каков относительный риск:

а) инфаркта миокарда для женатых по сравнению с одинокими?

б) смерти от инфаркта для женатых по сравнению с одинокими?

Задание 12

Для исследования влияния техногенного загрязнения на риск развития патологии беременности было проведено сравнительное исследование женщин 22–25 лет, проживающих в различных экологических условиях (табл. 2.12).

Таблица 2.12

Результаты исследования

	Невынашивание беременности	Другие патологии беременности
Превышение ПДУ техногенного загрязнения	70	16
Норма ПДУ	45	18

Рассчитать относительный риск невынашивания беременности в зависимости от техногенного загрязнения.

Задание 13

В исследовании связи между уровнем шума и заболеваемостью кохлеарным невритом было зарегистрировано в заводской поликлинике последовательно 80 случаев заболеваний мужчин кохлеарным невритом (табл. 2.13).

Таблица 2.13

Результаты исследования

	Заболели кохлеарным невритом	Другие ЛОР заболевания
Рабочие кузнечных цехов	68	10
Контроль	22	78

Какого типа это исследование? Проведите количественное сравнение частоты заболевания в экспонированной и неэкспонированной группах Сделайте выводы.

Контрольные вопросы

1. Интерпретируйте полученный результат, когда ОР>1.
2. Интерпретируйте полученный результат, когда ОР<1.
3. Интерпретируйте полученный результат, когда ОР=1.
4. При каком условии определяют ОШ, а не ОР?
5. Что характеризует значение ОР?
6. Приведите примеры ситуаций, когда исследование «случай–контроль» следует предпочесть когортному исследованию.

Список библиографических источников

1. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

2. *Орлов А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

Лабораторная работа № 3
Расчет атрибутивного риска для популяции

Цель: используя входную информацию, научиться рассчитывать атрибутивный риск заболеть для лиц экспонированной группы по сравнению с лицами, относящимися к неэкспонированной группе и интерпретировать полученные данные.

Атрибутивная фракция и атрибутивный риск

Разница в уровнях риска, называемая также **атрибутивным** или **абсолютным** риском (АР), есть разница в показателях частоты возникновения болезни между подвергающимися и не подвергающимися воздействию группами. Этот показатель позволяет получить представление о масштабах проблемы здравоохранения, порождаемой воздействием данного фактора:

$$АР = IR(Э) – IR(НЭ), \quad (3.1)$$

где IR(Э) – показатель частоты возникновения случаев болезни в экспонированной группе,

IR(НЭ) – соответственно, в неэкспонированной.

Атрибутивная фракция (воздействие) (АФ) или этиологическая фракция определяется путем деления абсолютного риска на показатель частоты возникновения болезни среди населения, подвергающегося воздействию изучаемого фактора риска.

Если предположить (или считается), что воздействие изучаемого фактора является причиной болезни, атрибутивной фракцией будет процент случаев этой болезни в определенной популяции, который был бы устранен при отсутствии воздействия:

$$АФ=АР/IR(Э) \times 100, \quad (3.2)$$

где АР – атрибутивный риск, IR(Э) – показатель частоты возникновения случаев болезни в экспонированной группе, АФ – атрибутивная фракция, выраженная в процентах.

Избыточная годовая частота случаев (ИГЧС) рассчитывается путем деления атрибутивного риска на долю лиц, подвергшихся воздействию:

$$ИГЧС = АР \times (ЧС1/ЧС2), \quad (3.3)$$

где ИГЧС – избыточная годовая частота случаев,

АР – атрибутивный риск (разница в уровнях заболеваемости в экспонированной и неэкспонированной группах),

ЧС1 – частота случаев в группе, подвергающейся воздействию,

ЧС2 – частота случаев в изучаемой популяции.

ИГЧС показывает число случаев предотвратимой болезни в данной популяции на 10 населения.

Атрибутивный риск для популяции (АРП), выраженный в процентах – это доля случаев среди изучаемого населения, которая вызывается

данным воздействием и может быть устранена, если это воздействие полностью прекратится.

Атрибутивный риск для популяции вычисляется путем деления ИГЧС для популяции на частоту болезни в популяции в целом и последующим умножением на 100:

$$\text{АРП} = \frac{\text{ИГЧС}}{IR(n)} * 100, \qquad (3.4)$$

где ИГЧС – избыточная годовая частота случаев,

IR(n) – заболеваемость популяции.

Задание 1

Зависимость заболеваний верхних дыхательных путей детей Гомельской обл., проживающих в различных экологических условиях, сведена в табл. 3.1 (период исследования – 1 год).

Таблица 3.1

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
До 5 ки/км^2	300	130 000
5–15 ки/км^2	450	90 000
Более 15 ки/км^2	700	120 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции, проживающей на изучаемой территории. Сделайте выводы.

Задание 2

Зависимость заболеваний эндокринной системы от места проживания у детей, проживающих на территориях с различной степенью радионуклидного загрязнения, сведена в табл. 3.2 (период исследования – 1 год).

Таблица 3.2

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Норма	100	30 000
До 15 ки/км^2	150	40 000
Более 15 ки/км^2	400	60 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции Сделайте выводы.

Задание 3

Зависимость заболеваний сердечно-сосудистой системы (в течение года) от употребления алкоголя сведена в табл. 3.3.

Таблица 3.3

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не употребляли	80	130 000
Периодически	20	40 000
Регулярно	150	160 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 4

Зависимость заболеваний верхних дыхательных путей (в течение года) от курения сведена в табл. 3.4.

Таблица 3.4

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не курили	70	230 000
Периодически	50	120 000
Регулярно	20	60 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 5

Зависимость заболеваний ЖКТ (в течение года) от частоты мытья рук у детей сведена в табл. 3.5.

Таблица 3.5

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не мыли руки перед едой	170	230 000
Периодически	50	100 000
Регулярно	200	860 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 6

Зависимость заболеваний органов зрения (в течение года) от продолжительности пользования компьютером сведена в табл. 3.6.

Таблица 3.6

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не пользовались	20	110 000
Периодически	50	120 000
Регулярно	40	60 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 7

Зависимость заболеваний кожи (в течение года) от степени нахождения на солнце сведена в табл. 3.7.

Таблица 3.7

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не загорали	25	250 000
Периодически	30	120 000
Регулярно	25	50 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 8

Зависимость заболевания ИБС (в течение года) от курения сведена в табл. 3.8.

Таблица 3.8

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не курили	35	350 000
Бросили курить	20	100 000
Регулярно курят	25	50 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 9

Зависимость заболевания ССС (в течение года) от употребления алкоголя сведена в табл. 3.9.

Таблица 3.9

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не употребляли	40	400 000
Бросили пить	25	100 000
Регулярно употребляют	30	60 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 10

Зависимость развития рака легкого от употребления сигарет сведена в табл. 3.10.

Таблица 3.10

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не курили	40	400 000
Бросили курить	25	100 000
Регулярно курят	30	60 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 11

Зависимость развития кариеса зубов от употребления сладостей сведена в табл. 3.11.

Таблица 3.11

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не употребляли	25	200 000
Иногда	15	100 000
Регулярно употребляют	30	60 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 12

Зависимость развития анемий среди беременных женщин в зависимости от употребления железосодержащих препаратов сведена в табл. 3.12.

Таблица 3.12

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не принимали	40	100 000
Иногда	25	100 000
Регулярно принимают	40	400 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Задание 13

Зависимость развития доброкачественных новообразований кожи от степени пребывания работников на открытом воздухе сведена в табл. 3.13.

Таблица 3.13

Результаты исследования

Категории лиц	Число случаев	Число обследованного населения
Никогда не находятся и	33	100 000
Иногда	30	150 000
Регулярно находятся	20	40 000

Рассчитать атрибутивную фракцию, избыточную годовую частоту случаев, атрибутивный риск для популяции. Сделайте выводы.

Контрольные вопросы

1. Что понимают под атрибутивным риском?
2. Что характеризует атрибутивный риск?
3. Как рассчитывается атрибутивная фракция?
4. Что характеризует атрибутивная фракция?
5. Как рассчитывается избыточная годовая частота случаев?
6. Как рассчитывается атрибутивный риск для популяции?
7. Что характеризует атрибутивный риск для популяции?

Список библиографических источников

1. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

2. *Орлов А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

Лабораторная работа № 4
Стандартизация показателей в эпидемиологии

Цель: научиться проводить сравнительный анализ показателей в предлагаемых группах, используя прямой метод стандартизации показателей, а также интерпретировать полученные данные.

Метод стандартизации как способ элиминации влияния структуры совокупностей на общий итоговый показатель

Входные данные для прямого метода стандартизации:

$A1_1$, $A1_2$,…… $A1_n$, $A1_{общ}$ – величина явления по выделенным подгруппам (от 1 до n) в группе 1.

$A2_1$, $A2_2$,…… $A2_n$, $A2_{общ}$ – величина явления по выделенным подгруппам (от 1 до n) в группе 2.

$N1_1$, $N1_2$,…… $N1_n$, $N1_{общ}$ – численность среды по подгруппам (от 1 до n) в группе 1.

$N2_1$, $N2_2$,…… $N2_n$, $N2_{общ}$ – численность среды по подгруппам (от 1 до n) в группе 2.

Стандарт:

S_1, S_2,…… S_n, $S_{общ}$ – состав среды хорошо изученной популяции.

1 этап

Вычисление фактических интенсивных коэффициентов

$$I1_1 = \frac{A1_1}{N1_1}; \qquad I1_n = \frac{A1_n}{N1_n}; \qquad I1_{общ} = \frac{A1_{общ}}{N1_{общ}}.$$

$$I2_1 = \frac{A2_1}{N2_1}; \qquad I2_n = \frac{A2_n}{N2_n}; \qquad I2_{общ} = \frac{A2_{общ}}{N2_{общ1}}.$$

2 этап

Расчет ожидаемых интенсивных коэффитициентов.

Под термином «ожидаемые интенсивные коэффициенты» подразумевается значения интенсивных коэффициентов в подгруппах (от 1 до n), которые были бы в изучаемых группах, если бы структура среды в них была одинаковой и равна стандартной.

Введем следующие обозначения:

$I1_1^{ож}$ – $I1_n^{ож}$ – ожидаемые интенсивные коэффициенты в группе 1 (подгруппы от 1 до n);

$I2_1^{ож}$ – $I2_n^{ож}$ – ожидаемые интенсивные коэффициенты в группе 2 (подгруппы от 1 до n).

$$I1_1^{ож} = \frac{I1_1}{k} \times S_1 \qquad I1_n^{ож} = \frac{I1_n}{k} \times S_n$$

$$I2_1^{ожс} = \frac{I2_1}{k} \times S_1 \qquad I2_n^{ожс} = \frac{I2_n}{k} \times S_n$$

Суммируя ожидаемые интенсивные коэффициенты по подгруппам (от 1 до n) в каждой группе, получаем соответственно стандартизированные показатели: для 1 группы – C_{T1}, для 2 группы – C_{T2}.

Задание 1

Вычислите стандартизованные коэффициенты заболеваемости условной болезнью в городах А и Б в 2014 г. (табл. 4.1).

Таблица 4.1

Исходные данные

Возрастные группы	Город А (экспонированное население)		Город Б (неэкспонированное население)	
	число человеко-лет	число случаев	число человеко-лет	число случаев
0–1	3200	1	575	2
1–3	41500	1	150	4
4–9	5500	12	678	3
10–19	8300	10	8025	10
20–39	16400	6	11750	3
40–59	10600	2	3450	1
60 и старше	1850	1	375	–
Всего	***50000***	***33***	***25000***	***23***

Стандартизацию провести по прямому методу. За стандарт взять возрастное распределение области, где находятся города А и Б (табл. 4.2).

Таблица 4.2

Возрастной состав населения области

Возрастная группа	Повозрастное распределение населения в области в %
0–1	3,4
1–3	7,9
4–9	9,4
10–19	18,6
20–39	30,1
40–59	22,2
60 и старше	8,4
Всего	100,0

Задание 2

Вычислите стандартизованные коэффициенты заболеваемости условной болезнью в городах А и Б в 2014 г. (табл. 4.3).

Таблица 4.3

Исходные данные

Возрастные группы	Город А (экспонированное население)		Город Б (неэкспонированное население)	
	число человеко-лет	число случаев	число человеко-лет	число случаев
0–1	3200	1	575	2
1–3	41500	1	150	4
4–9	5500	12	678	3
10–19	8300	10	8025	10
20–39	16400	6	11750	3
40–59	10600	2	3450	1
60 и старше	1850	1	375	–
Всего	***50000***	***33***	***25000***	***23***

Стандартизацию провести по прямому методу. За стандарт взять возрастное распределение экспонированного населения.

Задание 3

Вычислите стандартизованные коэффициенты заболеваемости условной болезнью в городах А и Б в 2010 г. (табл. 4.4).

Таблица 4.4

Исходные данные

Возрастные группы	Город А (экспонированное население)		Город Б (неэкспонированное население)	
	число человеко-лет	число случаев	число человеко-лет	число случаев
0–1	3200	1	575	2
1–3	41500	1	150	4
4–9	5500	12	678	3
10–19	8300	10	8025	10
20–39	16400	6	11750	3
40–59	10600	2	3450	1
60 и старше	1850	1	375	–
Всего	***50000***	***33***	***25000***	***23***

Стандартизацию провести по прямому методу. За стандарт взять суммарное возрастное распределение городов А и Б.

Задание 4

Вычислите стандартизованные коэффициенты заболеваемости условной болезнью в городах А и Б в 2014 г. (табл. 4.5).

Таблица 4.5

Исходные данные

Возрастные группы	Город А (экспонированное население)		Город Б (неэкспонированное население)	
	число человеко-лет	число случаев	число человеко-лет	число случаев
0–1	3200	1	575	2
1–3	41500	1	150	4
4–9	5500	12	678	3
10–19	8300	10	8025	10
20–39	16400	6	11750	3
40–59	10600	2	3450	1
60 и старше	1850	1	375	–
Всего	***50000***	***33***	***25000***	***23***

Стандартизацию провести по прямому методу. За стандарт взять ее возрастное распределение города Б.

Задание 5

Вычислите стандартизованные коэффициенты заболеваемости условной болезнью в городах А и Б в 2014 г. (табл. 4.6).

Таблица 4.6

Исходные данные

Возрастные группы	Город А (экспонированное население)		Город Б (неэкспонированное население)	
	число человеко-лет	число случаев	число человеко-лет	число случаев
0–1	3200	1	575	2
1–3	41500	1	150	4
4–9	5500	12	678	3
10–19	8300	10	8025	10
20–39	16400	6	11750	3
40–59	10600	2	3450	1
60 и старше	1850	1	375	–
Всего	***50000***	***33***	***25000***	***23***

Стандартизацию провести по прямому методу. За стандарт взять возрастное распределение области, где находятся города А и Б (табл. 4.7).

Таблица 4.7

Возрастной состав населения области

Возрастная группа	Повозрастное распределение населения в области в %
0–1	5
1–3	10
4–9	10
10–19	18
20–39	12
40–59	35
60 и старше	10
Всего	100,0

Задание 6

Вычислите стандартизованные коэффициенты заболеваемости дизентерией в зависимости от типа жилищ (табл. 4.8).

Таблица 4.8

Исходные данные

Возрастные группы	Общежития		Частные дома		Многоквартирные дома	
	число человеко-лет	число случаев	число человеко-лет	число случаев	число человеко-лет	число случаев
До 2 лет	95	79	28	9	137	108
2–3 года	22	5	9	1	45	19
4–18 лет	158	14	30	14	497	57
Всего	275	98	67	34	679	184

Стандартизацию провести прямым методом. За стандарт примите средний возрастной состав, живущих во всех трех типах жилищ.

Задание 7

Вычислите стандартизованные коэффициенты заболеваемости дизентерией в зависимости от типа жилищ (табл. 4.9).

Таблица 4.9

Исходные данные

Возрастные группы	Общежития		Частные дома		Многоквартирные дома	
	число человеко-лет	число случаев	число человеко-лет	число случаев	число человеко-лет	число случаев
До 2 лет	95	79	28	9	137	108
2–3 года	22	5	9	1	45	19
4–18 лет	158	14	30	14	497	57
Всего	275	98	67	34	679	184

Стандартизацию провести прямым методом. За стандарт примите возрастной состав, живущих в многоквартирных домах

Задание 8

Вычислите стандартизованные коэффициенты заболеваемости дизентерией в зависимости от типа жилищ (табл. 4.10).

Таблица 4.10

Исходные данные

Возрастные группы	Общежития		Частные дома		Многоквартирные дома	
	число человеко-лет	число случаев	число человеко-лет	число случаев	число человеко-лет	число случаев
До 2 лет	95	79	28	9	137	108
2–3 года	22	5	9	1	45	19
4–18 лет	158	14	30	14	497	57
Всего	275	98	67	34	679	184

Стандартизацию провести прямым методом. За стандарт примите возрастной состав, живущих в общежитии.

Задание 9

Вычислите стандартизованные коэффициенты заболеваемости дизентерией в зависимости от типа жилищ (табл. 4.11).

Таблица 4.11

Исходные данные

Возрастные группы	Общежития		Частные дома		Многоквартирные дома	
	число человеко-лет	число случаев	число человеко-лет	число случаев	число человеко-лет	число случаев
До 2 лет	95	79	28	9	137	108
2–3 года	22	5	9	1	45	19
4–18 лет	158	14	30	14	497	57
Всего	275	98	67	34	679	184

Стандартизацию провести прямым методом. За стандарт примите возрастной состав, живущих в частных домах

Задание 10

Вычислите стандартизованные коэффициенты заболеваемости пневмонией в различных профессиональных группах (табл. 4.12).

Таблица 4.12

Исходные данные

Возрастные группы	Работники автостоянок		Водители грузов на дальние расстояния		Калийные шахты	
	число человеко-лет	число случаев	число человеко-лет	число случаев	число человеко-лет	число случаев
18–20	95	79	28	9	137	108
21–25	22	5	9	1	45	19
26–30	158	14	30	14	497	57
31–40	160	30	25	10	150	100
41–50	140	40	25	10	250	30
51–60	155	30	50	20	225	70
Всего	730	198	167	74	1304	384

Стандартизацию провести прямым методом. За стандарт примите возрастной состав работников автостоянок

Задание 11

Вычислите стандартизованные коэффициенты заболеваемости пневмонией в различных профессиональных группах (табл. 4.13).

Таблица 4.13

Исходные данные

Возрастные группы	Работники автостоянок		Водители грузов на дальние расстояния		Калийные шахты	
	число человеко-лет	число случаев	число человеко-лет	число случаев	число человеко-лет	число случаев
18–20	95	79	28	9	137	108
21–25	22	5	9	1	45	19
26–30	158	14	30	14	497	57
31–40	160	30	25	10	150	100
41–50	140	40	25	10	250	30
51–60	155	30	50	20	225	70
Всего	730	198	167	74	1304	384

Стандартизацию провести прямым методом. За стандарт примите возрастной состав водителей на дальние расстояния.

Задание 12

Вычислите стандартизованные коэффициенты заболеваемости пневмонией в различных профессиональных группах (табл. 4.14).

Таблица 4.14

Исходные данные

Возрастные группы	Работники автостоянок		Водители грузов на дальние расстояния		Калийные шахты	
	число человеко-лет	число случаев	число человеко-лет	число случаев	число человеко-лет	число случаев
18–20	95	79	28	9	137	108
21–25	22	5	9	1	45	19
26–30	158	14	30	14	497	57
31–40	160	30	25	10	150	100
41–50	140	40	25	10	250	30
51–60	155	30	50	20	225	70
Всего	730	198	167	74	1304	384

Стандартизацию провести прямым методом. За стандарт примите возрастной состав работников калийных шахт.

Контрольные вопросы

1. Когда применяют метод стандартизации показателей?
2. Какие методы стандартизации вы знаете?

3. Что показывают стандартизованные показатели?

4. Какую группу населения принято принимать за стандарт?

5. Какие исходные данные необходимы для стандартизации прямым способом?

6. Опишите схему стандартизации прямым способом.

7. Что применяют в качестве стандарта при прямом способе стандартизации?

8. Какой стандарт применяют для сравнения показателей в различных точках мира?

Список библиографических источников

1. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

2. *Орлов А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

ЧАСТЬ 2
ПРОГРАММИРОВАНИЕ В R ДЛЯ АНАЛИЗА ДАННЫХ

Лабораторная работа № 5
Описательная статистика и подгонка распределений

Цель: научить обучающихся определять тип распределения переменных и давать первичную характеристику данным.

Загрузка внешних данных

Прочитаем (загрузим в текущее окружение) данные о состоянии атмосферного воздуха:

- файл данных имеет расширение .txt,
- значения друг от друга отделены пробелами,
- в первой строке находятся названия столбцов.

```
air=read.table("air.txt", header = T)
str(air)
```

```
## 'data.frame':    840 obs. of  7 variables:
##  $ Area     : Factor w/ 7 levels "1","2","3","4",..: 1 1 1 1 1
1 1 1 1 1 ...
##  $ Pollutant: Factor w/ 10 levels "Acroleine","Benzole",..: 7 1
2 10 4 8 3 9 6 5 ...
##  $ PDK      : int  250 30 100 200 5000 10 30 500 300 3000 ...
##  $ Emission : num  42.3 0 0 0 1000 ...
##  $ Month    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Hazard   : num  1.3 1.3 1.3 1 0.85 1.3 1.3 1 1 0.85 ...
##  $ IZA      : num  0.1 0 0 0 0.3 0.1 0.5 0 0 0 ...
```

Загрузка встроенных данных

При базовой установке R также устанавливается ряд учебных наборов данных. Они уже присутствуют на локальном диске компьютера и не требуют никаких дополнительных команд для загрузки. Обращаться по имени данных для данных из базовой библиотеки datasets нужно следующим образом:

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

```
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..:
1 1 1 1 1 1 1 1 1 1 ...
```

Управление таблицей данных

Выведем несколько первых записей (6 – по умолчанию):

```
head(air)
```

```
##   Area Pollutant  PDK Emission Month Hazard IZA
## 1    1       NO2  250    42.34     1   1.30 0.1
## 2    1 Acroleine   30     0.00     1   1.30 0.0
## 3    1   Benzole  100     0.00     1   1.30 0.0
## 4    1    Xylole  200     0.00     1   1.00 0.0
## 5    1        CO 5000  1000.00     1   0.85 0.3
## 6    1   Phenole   10     1.04     1   1.30 0.1
```

Выведем две последние записи:

```
tail(air,2)
```

```
##      Area Pollutant  PDK Emission Month Hazard IZA
## 839 Total        HP  300       NA    12   1.00  NA
## 840 Total      Dien 3000       NA    12   0.85  NA
```

Выведем первые 5 записей по переменной «Pollutant» (загрязнитель):

```
head(air$Pollutant,5)
```

```
## [1] NO2       Acroleine Benzole   Xylole    CO
## Levels: Acroleine Benzole CHO CO Dien HP NO2 Phenole SO2 Xylole
```

Подсчитаем кратность превышения ПДК. Функция with позволяют напрямую обращаться к переменным, но для этого первым аргументом должна быть соответствующая таблица:

```
kpp=with(air,Emission/PDK)
```

Для непосредственного доступа к переменным присоединяем таблицу данных к текущему окружению:

```
attach(air)
```

Теперь обращаемся к любой переменной выведем названия загрязнителей для первых 8 строк в списке:

```
Pollutant[1:8]
```

```
## [1] NO2       Acroleine Benzole   Xylole    CO    Phenole  CHO
## [8] SO2
## Levels: Acroleine Benzole CHO CO Dien HP NO2 Phenole SO2 Xylole
```

По окончании работы с данными не следует забывать отсоединить таблицу, поскольку возможное последующее присоединение другой таблицы с совпадающими именами переменных приведет к тому, что новые переменные «перекроют» предыдущие.

```
detach(air)
```

Исключение пропущенных значений

Не всегда функции в R умеют правильно работать с пропущенными значениями, поэтому исследователи прибегают к разным формам их преобразования. Самым простым вариантов является исключение строк, содержащих такие пропуски, из базы данных (хотя он же является и самым неправильным) исключить строки, которые содержат пропущенное значение хотя бы по одной переменной:

```
air.noNA=na.omit(air)
```

Подсчитать, сколько наблюдений исключено, можно выполнив следующую команду:

```
nrow(air)-nrow(air.noNA)
```

```
## [1] 220
```

Также можно удалять пропущенные значения в отдельном столбце. Например, удалим пропущенные значения переменной «Emission» (Выбросы»):

```
Emission.noNA=with(air,Emission[!is.na(Emission)])
```

Подсчитаем, сколько пропущенных значений:

```
length(air$Emission)-length(Emission.noNA)
```

```
## [1] 220
```

Расчет описательных статистик

На основе исходной базы данных создадим выборку, содержащую только усредненные по всем пунктам наблюдения за состоянием атмосферного воздуха, значения переменных:

```
air_total = subset(air, Area=="Total")
```

Простейшие статистики

Минимальная концентрация диоксида азота:

```
min(air_total[air_total$Pollutant == "NO2", "Emission"], na.rm=T)
## [1] 21.99
```

Максимальная концентрация диоксида азота:

```
max(air_total[air_total$Pollutant == "NO2", "Emission"], na.rm=T)
## [1] 126.97
```

Диапазон изменений концентраций азота в воздухе:

```
range(air_total[air_total$Pollutant == "NO2", "Emission"], na.rm=T)
## [1]  21.99 126.97
```

Ранжирование условных данных:

```
rank(c(5,3,2,7,11))
## [1] 3 2 1 4 5
```

Сумма значений ИЗА в мае:

```
sum(air_total[air_total$Month == "5", "IZA"], na.rm=T)
## [1] 1.4
```

Кумулятивная сумма (часто используется в расчетах долей и вероятностей):

```
cumsum(c(0.1,0.2,0.3,0.4))
## [1] 0.1 0.3 0.6 1.0
```

Расчет статистик, характеризующих центр рассеяния данных

Среднее арифметическое: средняя концентрация монооксида углерода по всем месяцам

```
mean(air_total[air_total$Pollutant == "CO", "Emission"], na.rm=T)
```

```
## [1] 2672.727
```

Среднее значение ИЗА

```
mean(air_total$IZA) #есть пропущенные значения
```

```
## [1] NA
```

```
mean(air_total$IZA,na.rm=TRUE) #проп.зн. не участвуют в вычислениях
```

```
## [1] 0.2372727
```

Медиана: концентрация формальдегида по всем месяцам

```
median(air_total[air_total$Pollutant == "CHO", "Emission"],
na.rm=TRUE)
```

```
## [1] 33.26
```

Проиллюстрируем тот факт, что медиана является частным случаем квантиля, а именно: 50-й процентилью

```
quantile(air_total[air_total$Pollutant == "CHO", "Emission"],
probs=0.5, na.rm=TRUE)
```

```
##   50%
## 33.26
```

Характеристики разброса значений

Стандартное отклонение среднемесячных концентраций в воздухе фенола:

```
sd(air_total[air_total$Pollutant == "Phenole", "Emission"],
na.rm=TRUE)
```

```
## [1] 2.637679
```

Дисперсия среднемесячных концентраций в воздухе фенола:

```
var(air_total[air_total$Pollutant == "Phenole", "Emission"],
na.rm=TRUE)
```

```
## [1] 6.957349
```

```
sd(air_total[air_total$Pollutant == "Phenole", "Emission"],
na.rm=TRUE)^2 #дисперсия
```

```
## [1] 6.957349
```

Несмещенная оценка дисперсии с помощью векторного выражения (используем не встроенную функцию, а пишем формулу для расчета):

```
n = length(air_total[air_total$Pollutant == "Phenole", "Emission"])
```

Функция **length()** используется для подсчета количества наблюдений переменной.

```
x=air_total[air_total$Pollutant == "Phenole", "Emission"]
sum((x-mean(x,na.rm=TRUE))^2/(n-1),na.rm=TRUE)
```

```
## [1] 6.324863
```

Асиммерия и эксцесс

Коэффициент асимметрии (Sk, *skewness*): мера симметричности распределения переменной, если

- Sk=0: расспределение симметрично относительно математического ожидания;
- Sk>0: распределение скошено влево (левый хвост распределения «тяжелее»).;
- Sk<0: распределение скошено вправо.

Коэффициент эксцесса (Ex, *kurtosis*): мера остроты пика распределения переменной, если

- Ex=0: острота пика нормального распределения;
- Ex>0: пик острее и хвосты «легче», чем у нормального распределения;
- Ex<0: пик более пологий, а хвосты «тяжелее», чем у нормального распределения.

```
library(e1071) #для доступа к следующим 2 функциям
```

```
skewness(x, na.rm=TRUE) #асимметрия: распределение скошено влево
```

```
## [1] 0.2113133
```

```
kurtosis(x, na.rm=TRUE) #эксцесс: пологий пик
```

```
## [1] -1.577455
```

Другие характеристики

Квантили среднемесячных концентраций в воздухе 1,3-бутадиена

```
quantile(air_total[air_total$Pollutant == "Dien", "Emission"],
probs=c(0.25,0.5,0.75), na.rm=TRUE)
```

```
##   25%   50%   75%
## 17.28 33.13 81.66
```

Интерквартильный размах среднемесячных концентраций в воздухе 1,3-бутадиена

```
IQR(air_total[air_total$Pollutant == "Dien", "Emission"],
na.rm=TRUE)
```

```
## [1] 64.38
```

Комплексные статистики

С помощью функции **summary()** можно получить ряд описательных статистик как для отдельной переменной, так и для набора данных в целом. В случае применения функции к числовым данным в качестве результата выводятся следующие статистики:

- минимальное значение,
- занчение 1-го квартиля,
- медианы,
- третьего квартиля,
- максимальное значение,
- число пропущенных значений.

```
summary(air_total$IZA)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.0000  0.0000  0.2373  0.3000  2.6000      10
```

В случае применения функции к категориальным данным в качестве результата выводятся частоты уникальных значений.

```
summary(air_total)
```

```
##     Area         Pollutant       PDK          Emission
##  1    :  0   Acroleine:12   Min.   :  10   Min.   :    0.000
##  2    :  0   Benzole  :12   1st Qu.:  30   1st Qu.:    0.000
##  3    :  0   CHO      :12   Median : 225   Median :    8.425
##  4    :  0   CO       :12   Mean   : 942   Mean   :  286.961
##  5    :  0   Dien     :12   3rd Qu.: 500   3rd Qu.:   49.367
```

```
##  6    :  0   HP       :12   Max.   :5000   Max.   :10500.000
##  Total:120   (Other)  :48                  NA's   :10
##      Month           Hazard          IZA
##  Min.   : 1.00   Min.   :0.85   Min.   :0.0000
##  1st Qu.: 3.75   1st Qu.:1.00   1st Qu.:0.0000
##  Median : 6.50   Median :1.15   Median :0.0000
##  Mean   : 6.50   Mean   :1.12   Mean   :0.2373
##  3rd Qu.: 9.25   3rd Qu.:1.30   3rd Qu.:0.3000
##  Max.   :12.00   Max.   :1.30   Max.   :2.6000
##                                 NA's   :10
```

Функция **fivenum()** вычисляет статистики для построения «ящика с усами»:

- минимальное значение,
- занчение 1-го квартиля,
- медианы,
- третьего квартиля,
- максимальное значение,

```
fivenum(air_total[air_total$Pollutant == "HP", "Emission"],
na.rm=TRUE)
```

```
## [1]  7.310 17.095 28.490 44.570 95.340
```

Другие статистики

С помощью функции **table()** можно подсчитать частоты значений для категоральной переменной, например число значений наблюдений каждого загрязнителя:

```
table(air_total$Pollutant)
```

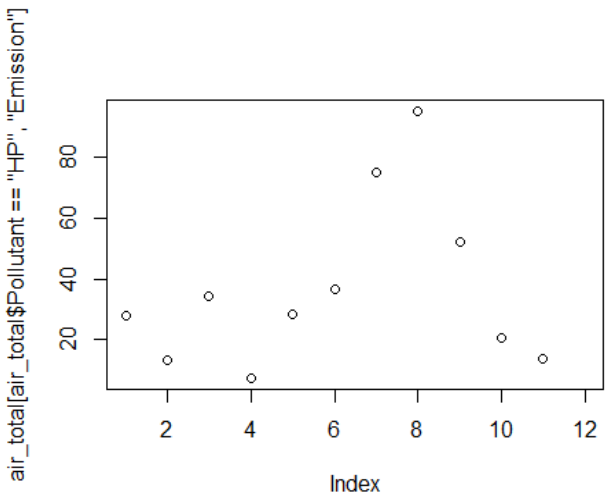
```
##
## Acroleine   Benzole      CHO       CO     Dien       HP      NO2
##        12       12        12       12       12       12       12
##   Phenole       SO2    Xylole
##        12        12        12
```

Графики
Базовая графическая система graphics
Функция plot()

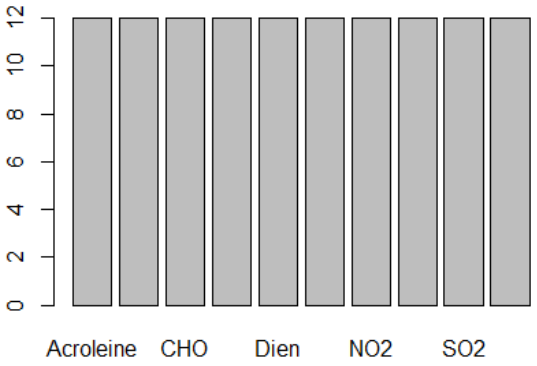
R уже в базовой версии предоставляет пользователю богатые возможности для визуализации данных.

Одна переменная

```
plot(air_total[air_total$Pollutant == "HP", "Emission"]) # для количественной переменной
```

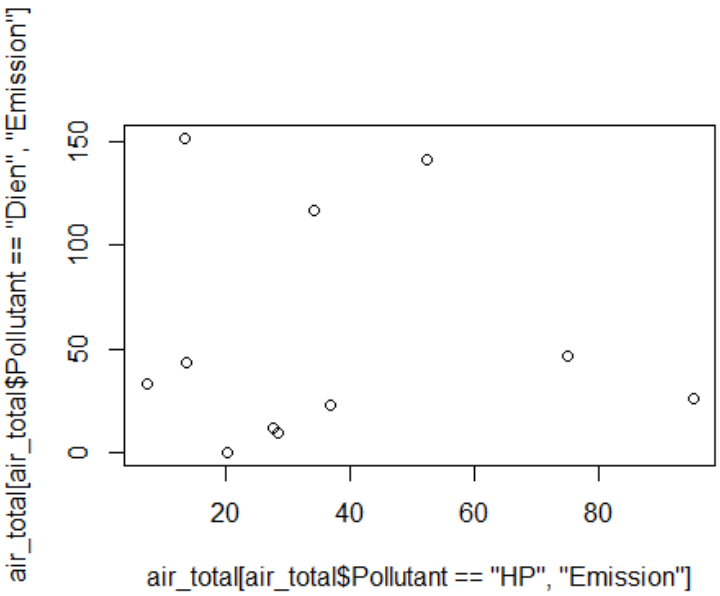



```
plot(air_total$Pollutant) #для категоральной (фактора)
```



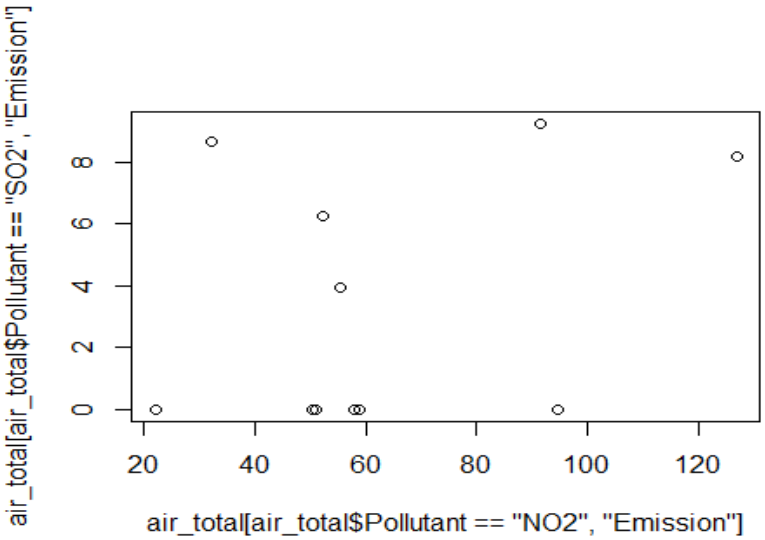
Две переменные

```
plot(air_total[air_total$Pollutant == "HP", "Emission"],
air_total[air_total$Pollutant == "Dien", "Emission"])
```



Использование интерфейса «Формула» (y~x: y зависит от x)

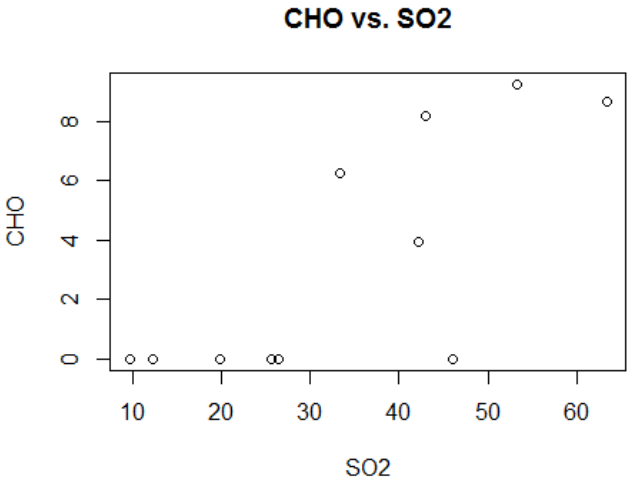
```
plot(air_total[air_total$Pollutant == "SO2", "Emission"]
~air_total[air_total$Pollutant == "NO2", "Emission"])
```



Заголок (main) и подписи осей (xlab,ylab)

При явном указании имен параметров порядок не важен:

```
plot(air_total[air_total$Pollutant == "CHO", "Emission"],
air_total[air_total$Pollutant == "SO2", "Emission"],main="CHO
vs. SO2", ylab="CHO",xlab="SO2")
```

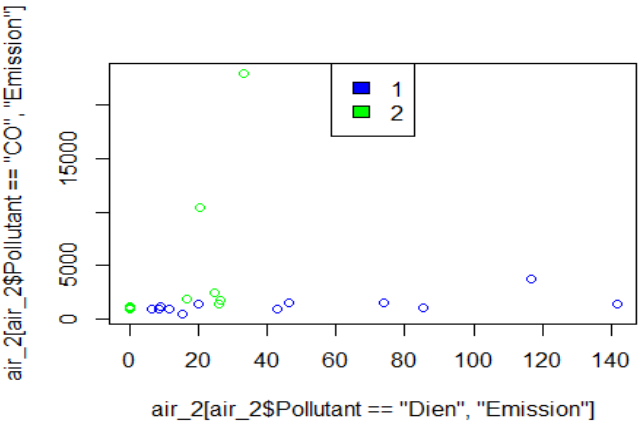


Создадим выборку, содержащую только значения монооксида углерода и 1,3-бутадиена, по двум точкам наблюдения за состоянием атмосферного воздуха:

```
air_2 = subset(air, Pollutant=="CO" & Area =="1"
| Pollutant=="Dien"  & Area =="2" | Pollutant=="CO" & Area =="2"
| Pollutant=="Dien"  & Area =="1",select=c(Area,Pollutant,Emission))
```

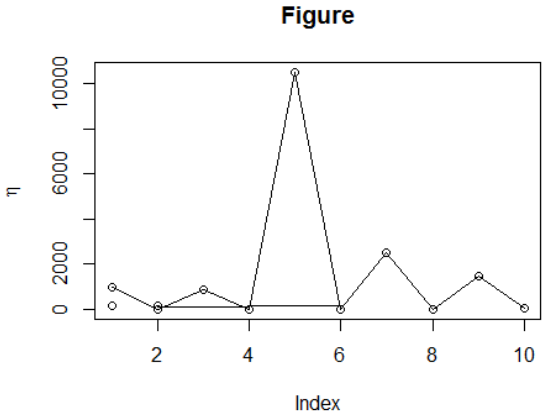
Зададим цвет точек (наблюдений) по точкам наблюдения за состоянием атмосферного воздуха и добавим легенду:

```
colors=ifelse(air_2$Area=="1","blue","green")
plot(air_2[air_2$Pollutant == "CO", "Emission"]~air_2
[air_2$Pollutant == "Dien", "Emission"],col=colors)
#легенда накладывается последовательным вызовом данной функции:
legend("top",legend=c("1","2"),fill=c("blue","green"))
```



Добавление элементов на график

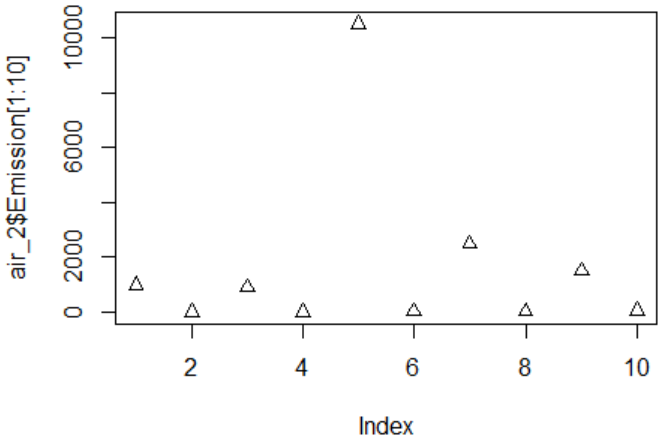
```
plot(air_2$Emission[1:10],ylab=expression(eta)) #expression: мат. символы
title("Figure") # добавить заголовок
lines(air_2$Emission[1:10]) #соединить линиями
points(c(140,150)) #добавить точки
curve(15*x+80,from=2,to=6,add = T) #добавить линию
```



Графические параметры

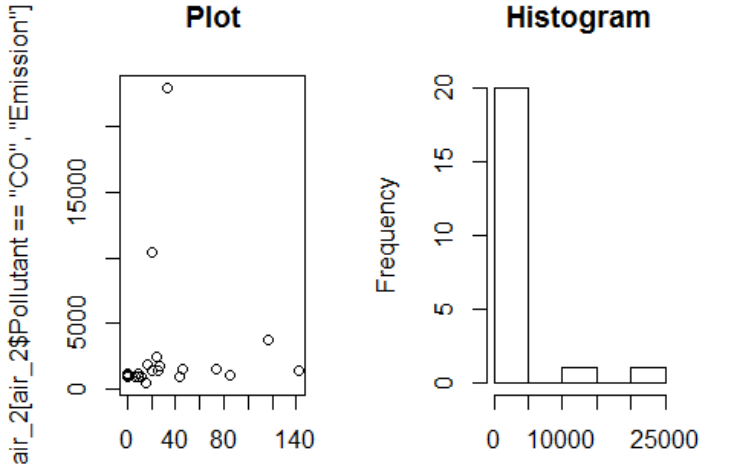
Изменим вид точки:

```
par(pch = 2)
plot(air_2$Emission[1:10])
```



Зададим количество отображаемых графиков: одна строка, два столбца

```
par(mfrow=c(1,2))
plot(air_2[air_2$Pollutant == "CO", "Emission"]~air_2
[air_2$Pollutant == "Dien", "Emission"],main="Plot")
hist(air_2[air_2$Pollutant == "CO", "Emission"],main="Histogram")
```

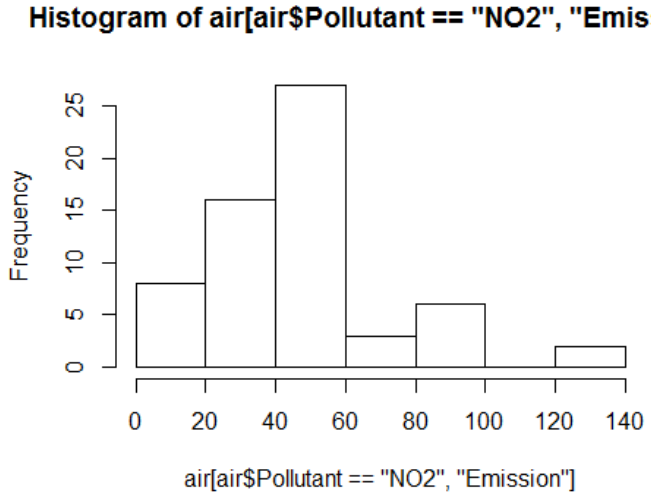



```
par(mfrow=c(1,1),pch=1) #вернуть по умолчанию
```

Гистограмма

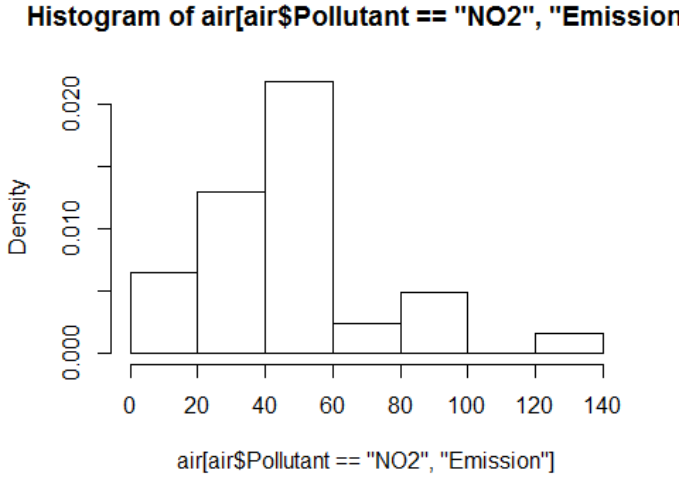
Распределение концентраций диоксида азота из выборки:

```
hist(air[air$Pollutant == "NO2", "Emission"]) #по оси y - частоты
```



По оси y отобразим относительные частоты:

```
hist(air[air$Pollutant == "NO2", "Emission"],probability = T)
#~ freq=F
```

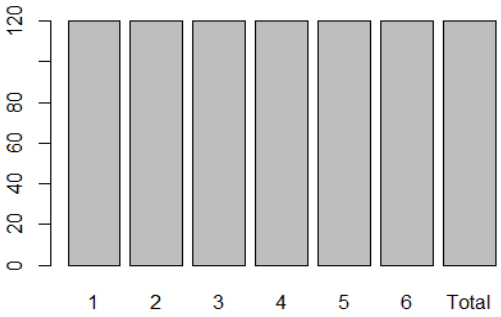


Столбиковая диаграмма

```
(t=table(air$Area)) # подсчитаем частоты значений наблюдений
по местам наблюдений
```

```
##
##     1     2     3     4     5     6 Total
##   120   120   120   120   120   120   120
```

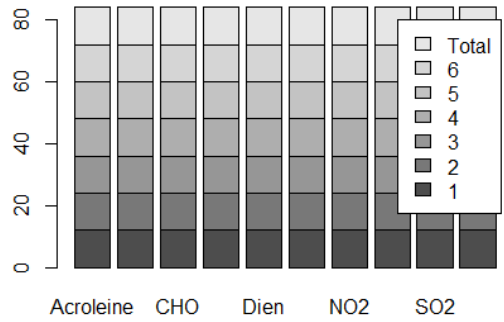
```
barplot(t) #выведем диаграмму
```




```
(t=table(air$Area,air$Pollutant))
```

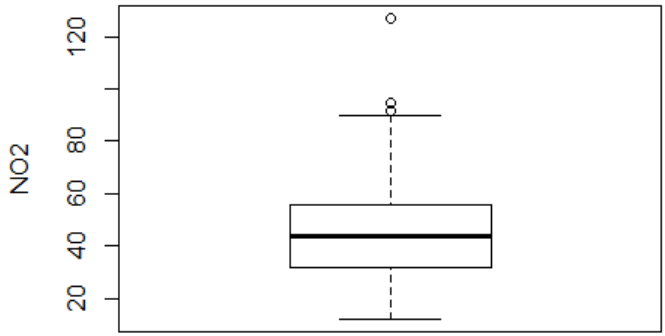
```
##
##         Acroleine Benzole CHO CO Dien HP NO2 Phenole SO2 Xylole
##   1            12      12  12 12   12 12  12      12  12     12
##   2            12      12  12 12   12 12  12      12  12     12
##   3            12      12  12 12   12 12  12      12  12     12
##   4            12      12  12 12   12 12  12      12  12     12
##   5            12      12  12 12   12 12  12      12  12     12
##   6            12      12  12 12   12 12  12      12  12     12
##   Total        12      12  12 12   12 12  12      12  12     12
```

```
barplot(t,legend.text = c("1","2", "3", "4", "5", "6", "Total"))
```



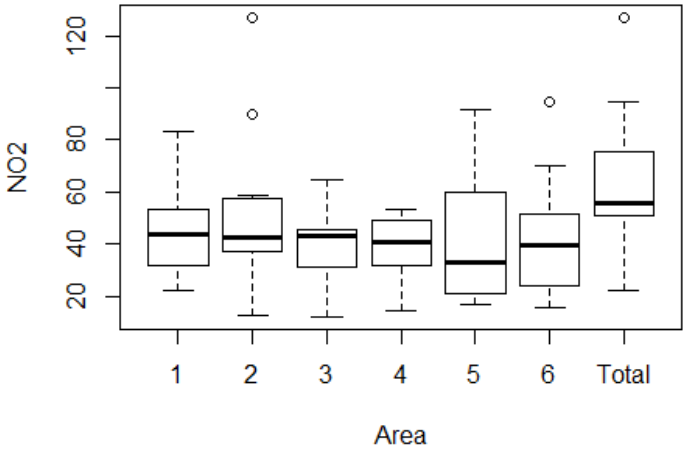
«Ящик с усами»

```
boxplot(air[air$Pollutant == "NO2", "Emission"],ylab="NO2")
```



Кружками показаны аномальные (резко выделяющиеся) значения.

```
boxplot(air[air$Pollutant == "NO2", "Emission"]~air[air$Pollu-
tant == "NO2", "Area"],ylab="NO2",xlab="Area")
```



В разрезе по месту наблюдения аномальные наблюдения зафиксированы только среди второй и шестой точек наблюдения.

Диаграмму «ящик с усами» можно построить и по нескольким переменным:

```
boxplot(air[air$Pollutant == "NO2", "Emission"],air[air$Pollutant
== "HP", "Emission"],ylab="Концентрация",names=c("NO2","HP"))
```

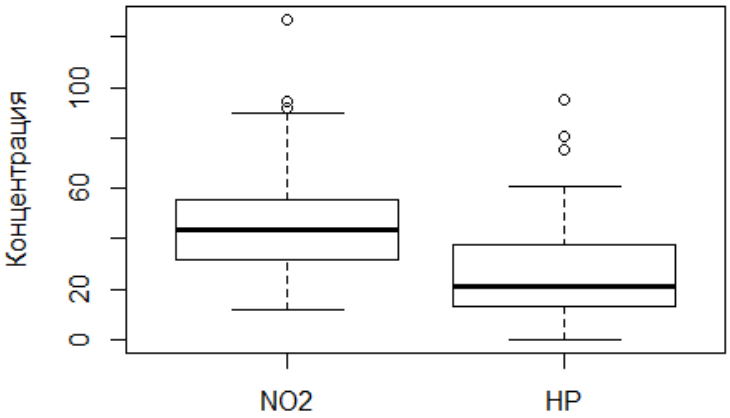


График ядерной оценки плотности

Нарисуем график распределения плотности распределения переменной:

```
d=density(air[air$Pollutant == "NO2", "Emission"], na.rm = T)
plot(d,main="Распределение для концентраций
NO2",ylab="Плотность распределения")
```

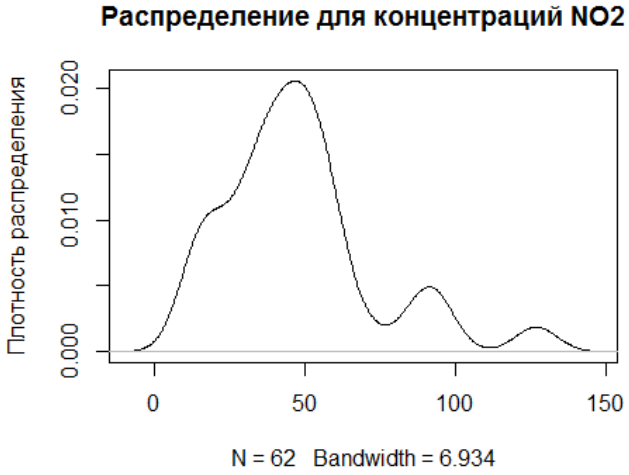


График «Квантиль-квантиль»

```
qqnorm(air[air$Pollutant == "NO2", "Emission"])
```

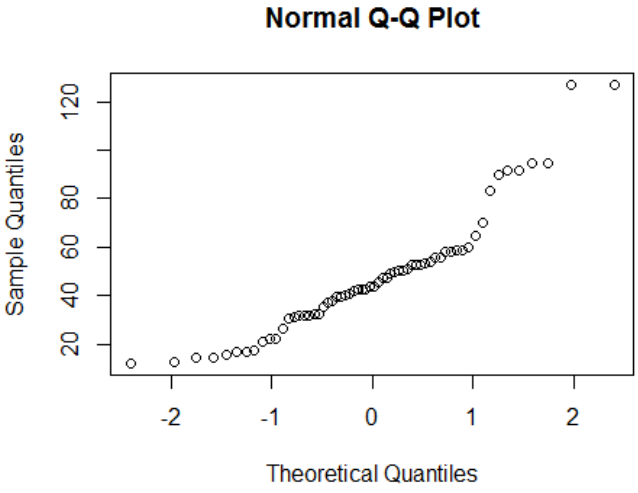
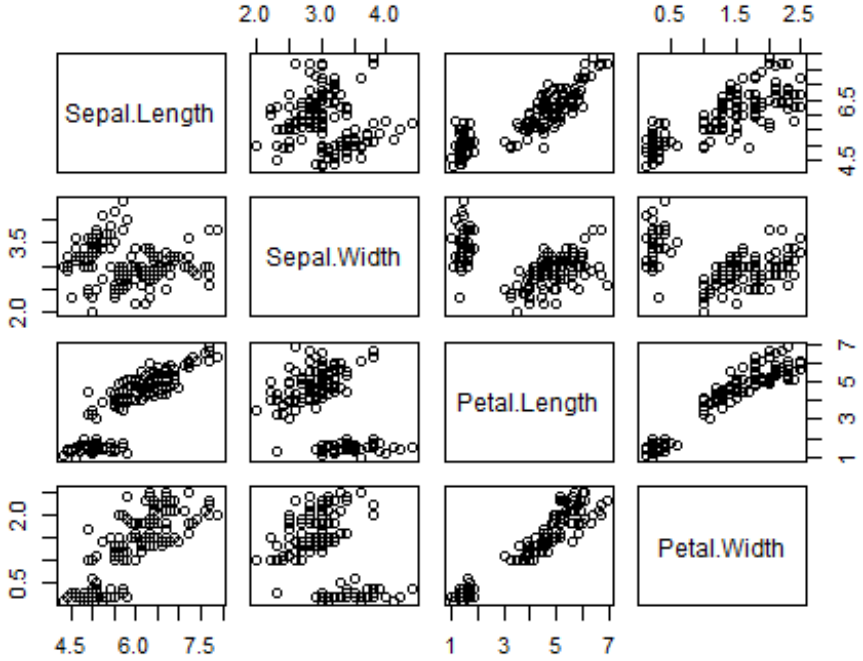


Диаграмма рассеивания

```
pairs(iris[1:4])
```



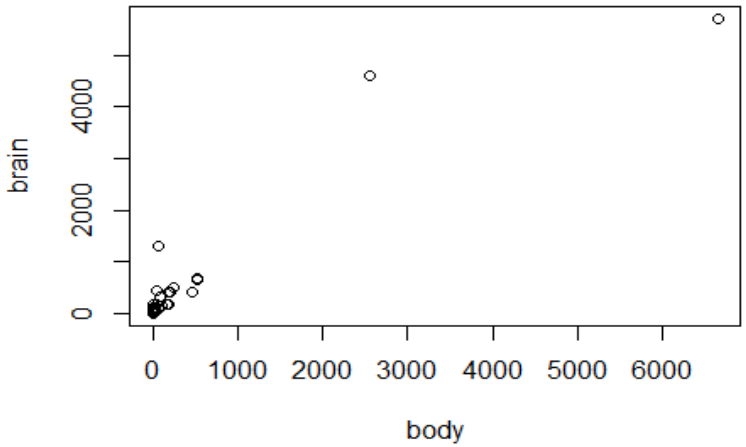
Использование разных типов шкал для графиков

Загрузим библиотеку «MASS», содержащую данные о размерах тела и мозга животных:

```
library(MASS)
```

Построим двумерный график по данным о млекопитающих (mammals):

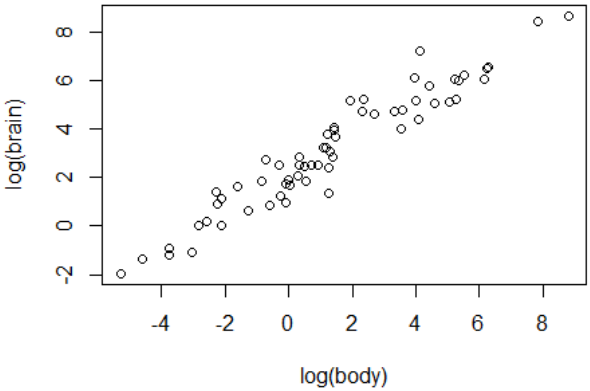
```
plot(mammals) #наблюдения
```



Как видно из данных, представленных на рисунке, большинство из них группируется в левом нижнем углу графика, что затрудняет восприятие информации.

Логарифмируем переменные:

```
attach(mammals)
plot(log(brain)~log(body))
```




```
detach(mammals)
```

Моделирование
Случайные величины
Равномерное распределение

Функции

1. runif – генерация случайных величин (СВ).
2. punif – функция распределения вероятностей (ФРВ).
3. dunif – функция плотности РВ.
4. qunif – функция вычисления квантилей (обратная ФРВ).

пример генерации СВ

```
runif(10) #10 БСВ: на интервале [0,1]
```

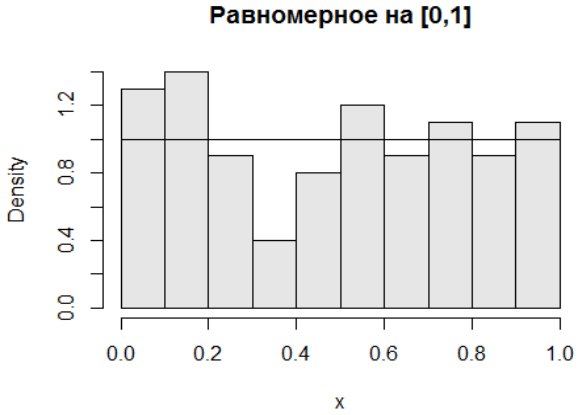
```
##  [1] 0.1587857 0.7533431 0.1569911 0.3326216 0.9798538
0.3268695 0.6695123
##  [8] 0.5613103 0.1405742 0.2987603
```

```
runif(10,1,10) #на интервале [1,10]
```

```
##  [1] 1.369596 1.431579 8.574121 5.648148 9.203231 1.423019
9.225503
##  [8] 7.658453 9.323554 7.408923
```

Построим гистограмму:

```
x=runif(100)
hist(x,probability=T,col=gray(0.9),main="Равномерное на [0,1]")
curve(dunif(x,0,1),add=T) #добавить теоретическую кривую
```




```
(x=runif(5))
```

```
## [1] 0.5263360 0.4502062 0.6260963 0.8420722 0.8204559
```

Вычислим вероятности распределения x:

```
punif(x)
```

```
## [1] 0.5263360 0.4502062 0.6260963 0.8420722 0.8204559
```

Вычислим плотность распределения:

```
dunif(x)
```

```
## [1] 1 1 1 1 1
```

Биномиальное распределение

10 бросаний монеты:

```
rbinom(n=10,size=1,p=0.5)
```

```
##  [1] 1 1 0 1 0 1 1 1 1 0
```

Имеется n=10 партий продукции по size=100 изделий с вероятностью брака p=0.05.

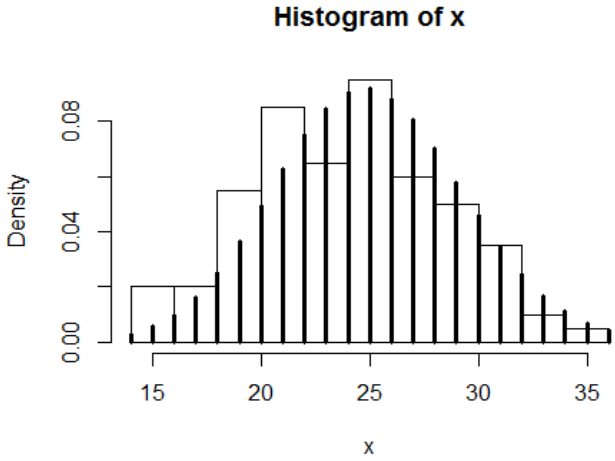
Рассчитаем количество бракованных изделий в каждой партии:

```
rbinom(10,100,0.05)
```

```
##  [1] 5 3 5 4 1 3 5 9 4 4
```

Построим гистограммую для n=100,size=50,p=0.25

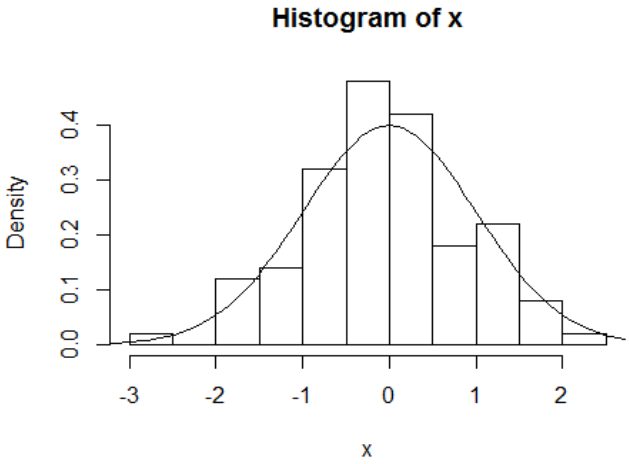
```
x=rbinom(100,100,0.25)
hist(x,probability = T)
#наложим теоретическое распределение
points(0:100,dbinom(0:100,100,0.25),type="h",lwd=3)
```



Нормальное распределение

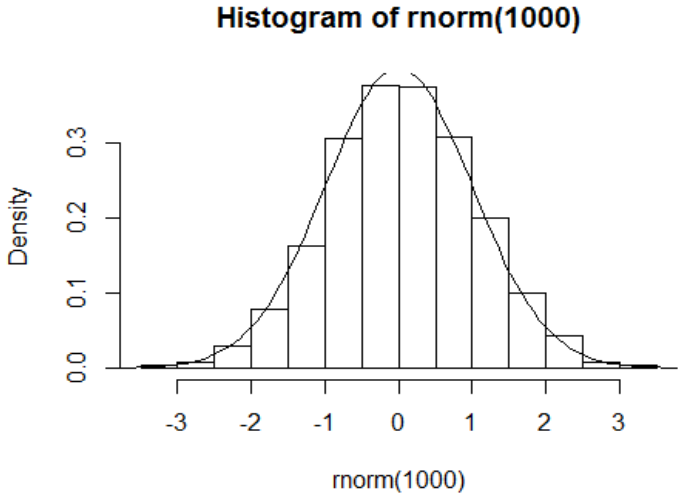
Для выборки n=100:

```
x=rnorm(100)
hist(x,freq=F)
curve(dnorm,-4,4,add=T)
```



Для большего числа наблюдений:

```
hist(rnorm(1000),freq=F)
curve(dnorm,-4,4,add=T)
```



Значения коэффициентов асимметрии и эксцеса должны быть близки к нулю:

```
x=rnorm(1000)
skewness(x)
```

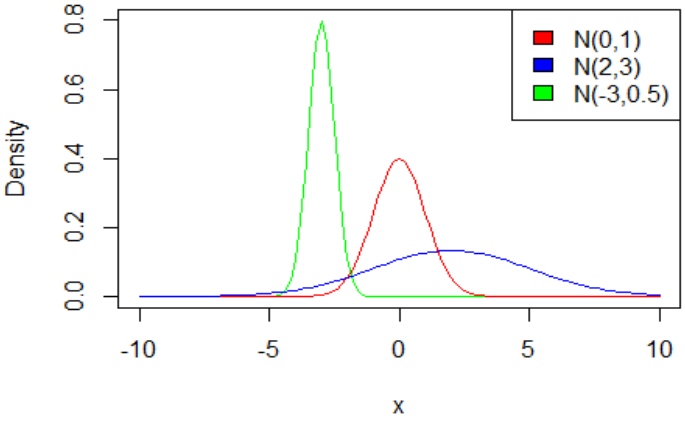
```
## [1] 0.1757516
```

```
kurtosis(x)
```

```
## [1] 0.08287609
```

Графики фаспределения в случае разных значений среднего и дисперсии:

```
x=seq(from=-10,to=10,by=0.1)
curve(dnorm(x,-3,0.5),from=-10,to=10,col="green",
      ylab="Density")
curve(dnorm(x),from=-10,to=10,col="red",add=T)
curve(dnorm(x,2,3),from=-10,to=10,col="blue",add=T)
legend("topright",c("N(0,1)","N(2,3)","N(-
3,0.5)"),fill=c("red","blue","green"))
```



Использование функций:

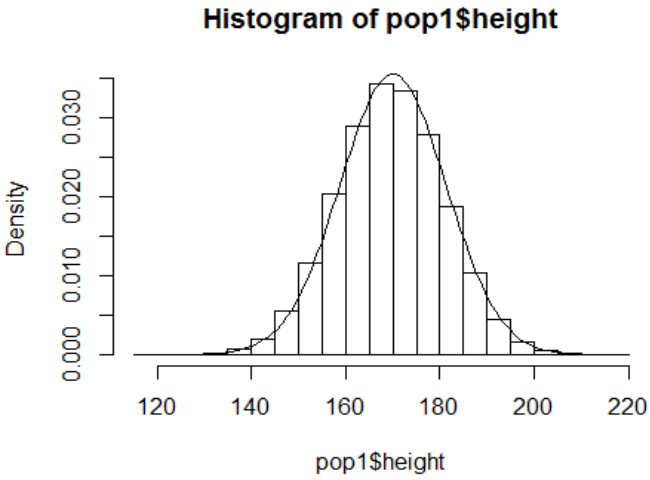
```
pnorm(2) #вероятность что СВ x< 2
## [1] 0.9772499
pnorm(2,lower.tail = F) #вероятность, что СВ x>2
## [1] 0.02275013
qnorm(0.95) #квантиль уровня 0.95: обратная функция для pnorm
## [1] 1.644854
```

Пример нормального распределения. Файл «pop1.csv» содержит данные о 100000 человек.

```
pop1=read.csv("pop1.csv")
```

Построим гистограмму по переменной «height»:

```
hist(pop1$height,freq=F) #нормальное распределение
#параметры
m=mean(pop1$height) #среднее
s=sd(pop1$height) #стандартное отклонение
x=seq(from=min(pop1$height),to=max(pop1$height),by=1)
curve(dnorm(x,m,s),add=T) #нанесем теоретическую кривую
```



Случайная подвыборка: sample

Случайный выбор без повторений: игра в лотерею

```
sample(1:100,5) #выбрать 5 чисел из 100
## [1] 70 71 54 64 97
sample(1:100,5) #снова выберем 5 чисел: другой результат
## [1] 68 86 60 64 80
```

Случайный выбор с повторениями: игра в кости

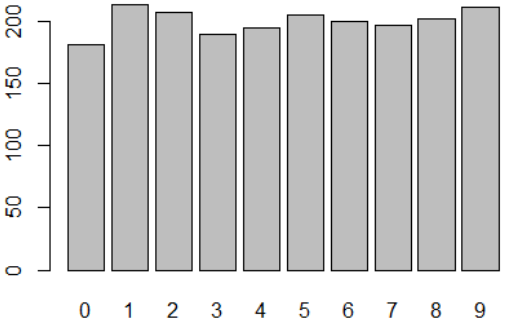
```
sample(1:6,10,replace=TRUE) #кинем кости 10 раз
##  [1] 2 2 6 1 6 6 4 3 4 1
sample(1:6,10,replace=T) #еще одна серия экспериментов
##  [1] 5 1 6 1 5 1 5 4 5 2
```

Тесты согласия: проверка распределения

Критерий Хи-квадрат Пирсона

Из файла «pi2000.txt» выпишем первые 2000 цифр после запятой для числа π:

```
pi2000=scan("pi2000.txt")
barplot(table(pi2000))
```



Проверим с помощью теста гипотезу, что цифры попадаются с одинаковой вероятностью 0.1:

```
chisq.test(table(pi2000)) #гипотеза не отклоняется
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(pi2000)
## X-squared = 4.42, df = 9, p-value = 0.8817
```

Критерий Колмогорова–Смирнова

Модельные данные

Проверим на нормальное распределение случайно сгенерированный набор чисел.

p-value > 0.05 : гипотеза о нормальном распределении (pnorm - ФРВ норм. распр.) подтверждается на уровне 0.05

```
ks.test(rnorm(1000),pnorm) # гипотеза о нормальном
распределении принимается
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  rnorm(1000)
## D = 0.039248, p-value = 0.09184
## alternative hypothesis: two-sided
```

```
ks.test(runif(1000),pnorm)
```

```
##
##  One-sample Kolmogorov-Smirnov test
```

```
##
## data:  runif(1000)
## D = 0.50044, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Реальные данные

Проверим распределение роста на нормальное на выборке из 100 000 человек (набор данных «pop1.txt»).

Извлечем только неповторяющиеся значения:

```
heights=unique(pop1$height)
```

Вычислим, сколько уникальных значений роста содержится в базе данных:

```
length(heights)
```

```
## [1] 94
```

```
m=mean(heights)
s=sd(heights)
```

Указываем выборочные параметры распределения:

```
ks.test(heights,pnorm,mean=m,sd=s)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  heights
## D = 0.060504, p-value = 0.8607
## alternative hypothesis: two-sided
```

Критерий Лиллиефорса (Lilliefors)

Критерий Лиллиефорса представляет собой модифицированный критерий Колмогорова–Смирнова.

```
library(nortest)
lillie.test(heights)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  heights
## D = 0.060504, p-value = 0.5409
```

Список библиографических источников

Основной

1. *Горяинова, Е. Р.* Прикладные методы анализа статистических данных: учеб. пособие / Е. Р. Горяинова, А. Р. Панков, Е. Н. Платонов. – М.: Изд. дом Высшей школы экономики, 2012. – 310 с.

2. *Лесковец, Ю.* Анализ больших наборов данных / Ю. Лесковец, А. Раджараман. – М.: ДМК, 2016. – 498 с.

3. *Крянев, А. В.* Метрический анализ и обработка данных / А. В. Крянев, Г. В. Лукин, Д. К. Удумян. – М.: Физматлит, 2012. – 308 с.

4. *Кулаичев, А. П.* Методы и средства комплексного анализа данных: учеб. пособие / А. П. Кулаичев. – М.: Форум, НИЦ ИНФРА-М, 2013. – 512 с.

5. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

6. 1Biostatistics with R: An Introduction to Statistics Through Biological Data (Use R!) / Babak Shahbaba, 2012.

7. Modern Epidemiology Third, Mid-cycle revision Edition / J. Kenneth Rothman, L. Timothy Lash Associate Professor, Sander Greenland, 2012.

8. Biostatistics and Epidemiology: A Primer for Health and Biomedical Professionals 4th ed. / Sylvia Wassertheil-Smoller, Jordan Smoller, 2015.

9. Introductory Time Series with R (Use R!) / Paul S. P. Cowpertwait, Andrew V. Metcalfe, 2009.

Дополнительный

10. *Новикова, Н. М.* Статистические методы в биологии и медицине: учеб.-метод. пособие / Н. М. Новикова. – Минск: МГЭУ им. А. Д. Сахарова, 2009. – 88 с.

11. *Кабаков, Р.* R в действии. Анализ и визуализация данных в программе R / Р. Кабаков. – М.: ДМК, 2016. – 588 с.

12. Биометрика – журнал для медиков и биологов, сторонников доказательной биомедицины [Электронный ресурс]. URL: http://www.-biometrica.tomsk.ru.

13. *Орлов, А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

14. *Чиркин, А. А.* Клинический анализ лабораторных данных / А. А. Чиркин. – Витебск: Медицинская литература, 2008. – 384 с.

15. *Петри, А.* Наглядная статистика в медицине / А. Петри, К. Сэбин. – М.: Изд. дом ГЭОТАР-МЕД, 2003 – 143 с.

Лабораторная работа № 6
Типы данных. Разведочный анализ данных

Цель: научиться исследовать внутреннюю структуру данных путем разведочного анализа.

Типы данных. Основные операции в R

Анализ данных (методы, которые можно применить, графики, которые можно построить) зависит от того, какие данные мы изучаем.

Типы данных:

Примеры для обсуждения:

- рост
- число студентов в классе
- страна проживания
- пол
- год создания
- цена
- удовлетворенность услугой (по шкале от 1 до 5)

Основные типы данных в R:

- numeric
- integer
- factor
- logical
- character

Для любой переменной можно проверить, к какому типу данных (классу) она относится:

```
class(100)
## [1] "numeric"
class("Hello")
## [1] "character"
class(45.7)
## [1] "numeric"
class(TRUE)
## [1] "logical"
class(1:2)
## [1] "integer"
```

Можно проверить принадлежность к конкретному типу:

```
is.integer("Hello")## [1] FALSE
is.character("Hello")
```

```
## [1] TRUE
is.integer(100)
## [1] FALSE
is.numeric(100)
## [1] TRUE
is.logical(45.7)
## [1] FALSE
is.logical(TRUE)
## [1] TRUE
```

... и преобразовать из типа в тип (но аккуратно):

```
as.character(45.7)
## [1] "45.7"
as.character(100)
## [1] "100"
as.integer("Hello")
## Warning: в результате преобразования созданы NA
## [1] NA
as.integer(100)
## [1] 100
as.numeric(TRUE)
## [1] 1
```

Создание переменной

Для создания переменных используется оператор присваивания <- или = (a <- 6). Таким образом, мы придумываем имя переменной, а затем присваиваем ей некоторое значение. Дальше мы можем работать с этим именем

```
newVariable <- TRUE
class(newVariable)
## [1] "logical"
newVariable <- as.numeric(newVariable)
newVariable
## [1] 1
newVariable + 7
## [1] 8
```

Но анализ данных подразумевает работу со многими значениями. Так и переменная может соответствовать более сложным структурам

Вектора

Вектора считаются базовыми элементами в R. Они объединяют в себе несколько однотипных значений (например, возраст для целой группы, а не возраст одного человека), что позволяет, в частности,

обрабатывать их все одновременно. Оператор ***c()*** используется для объединения нескольких элементов в один вектор

```
age <- c(23, 20, 21, 19, 20, 21, 20, 23)
age
## [1] 23 20 21 19 20 21 20 23
class(age)
## [1] "numeric"
## сколько будет лет каждому в группе через 5 лет
age + 5
## [1] 28 25 26 24 25 26 25 28
## участнику группы 21 год?
age == 21
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
```

Логические операции

```
## участнику группы 21 год? age <- c(23, 20, 21, 19, 20, 21, 20, 23)
age == 21
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
## участнику группы больше 21 года? age <- c(23, 20, 21, 19, 20, 21,
20, 23)
age > 21
## [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## участнику группы не меньше 21 года? age <- c(23, 20, 21, 19, 20,
21, 20, 23)
age >= 21
## [1]  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
## участнику группы не 20 лет? age <- c(23, 20, 21, 19, 20, 21,
20, 23)
age != 20
## [1]  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
## участнику группы от 20 до 22? age <- c(23, 20, 21, 19, 20, 21,
20, 23)
(age >= 20) & (age <= 22)
## [1] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
```

Обобщения

О том, как работают разные функции, можно узнать из справки:

```
?min
?mean
?summary
?sum
```

Пример:

```
min(age)
## [1] 19
max(age)
## [1] 23
mean(age)
## [1] 20.875
summary(age)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   20.00   20.50   20.88   21.50   23.00
```

Вектора могут быть не только числовыми, но и строковыми, логическими:

```
sex <- c("m", "m", "f", "f", "m", "f", "m", "f")
class(sex)
## [1] "character"
```

Если переменная принимает всего несколько значений, то она, скорее всего, соответствует некоторым категориям. В R категории представлены факторами (factors)

```
sex <- as.factor(sex)
class(sex)
## [1] "factor"
summary(sex)
## f m
## 4 4
## сравните с той же функцмей для age
summary(age)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   20.00   20.50   20.88   21.50   23.00
```

Таблицы данных (data frame)

Однако мы редко имеем дело только с одной переменной, чаще для одного человека/предмета/страны... у нас есть несколько показателей (не только возраст, но и, например, пол). Таким образом данные можно объединять в таблицы (составляющие их вектора должны быть одинаковой длины).

```
dataExample <- data.frame(age, sex)
class(dataExample)
## [1] "data.frame"
dataExample
##   age sex
## 1  23   m
```

```
## 2  20   m
## 3  21   f
## 4  19   f
## 5  20   m
## 6  21   f
## 7  20   m
## 8  23   f
head(dataExample)
##   age sex
## 1  23   m
## 2  20   m
## 3  21   f
## 4  19   f
## 5  20   m
## 6  21   f
tail(dataExample)
##   age sex
## 3  21   f
## 4  19   f
## 5  20   m
## 6  21   f
## 7  20   m
## 8  23   f
```

Более содержательный пример: характеристики 38 популярных моделей автомобилей с 1999 по 2008 гг.

```
library(ggplot2)
head(mpg)
## # A tibble: 6 <U+00D7> 11
##  manufacturer model displ  year  cyl   trans  drv  cty  hwy  fl
##     <chr> <chr> <dbl> <int> <int>    <chr> <chr> <int> <int> <chr>
## 1   audi    a4   1.8  1999     4   auto(l5)   f    18    29     p
## 2   audi    a4   1.8  1999     4 manual(m5)   f    21    29     p
## 3   audi    a4   2.0  2008     4 manual(m6)   f    20    31     p
## 4   audi    a4   2.0  2008     4   auto(av)   f    21    30     p
## 5   audi    a4   2.8  1999     6   auto(l5)   f    16    26     p
## 6   audi    a4   2.8  1999     6 manual(m5)   f    18    26     p
## # ... with 1 more variables: class <chr>
?mpg
```

Типы данных некоторых переменных (с помощью символа $ можно указать, какая именно переменная нас интересует):

```
class(mpg$displ)
## [1] "numeric"
class(mpg$year)
## [1] "integer"
class(mpg$manufacturer)
## [1] "character"
```

Характеристики данных:

```
dim(mpg)
## [1] 234  11
str(mpg)
## Classes 'tbl_df', 'tbl' and 'data.frame':  234 obs. of  11 variables:
## $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model : chr  "a4" "a4" "a4" "a4" ...
##  $ displ : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year  : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl   : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans  : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv   : chr  "f" "f" "f" "f" ...
##  $ cty   : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy   : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl    : chr  "p" "p" "p" "p" ...
##  $ class : chr  "compact" "compact" "compact" "compact" ...
summary(mpg)
##  manufacturer          model               displ            year
##  Length:234       Length:234         Min.  :1.600   Min.   :1999
##  Class :character  Class :character  1st Qu.:2.400  1st Qu.:1999
##  Mode  :character  Mode  :character  Median :3.300  Median :2004
##                                      Mean   :3.472  Mean   :2004
##                                      3rd Qu.:4.600  3rd Qu.:2008
##                                      Max.   :7.000  Max.   :2008
##       cyl          trans               drv                cty
##  Min.   :4.000   Length:234       Length:234         Min.   : 9.00
##  1st Qu.:4.000  Class :character   Class :character  1st Qu. :14.00
##  Median :6.000  Mode  :character  Mode  :character  Median :17.00
##  Mean   :5.889                                      Mean   :16.86
##  3rd Qu.:8.000                                      3rd Qu.:19.00
##  Max.   :8.000                                      Max.   :35.00
##       hwy             fl                class
##  Min.   :12.00   Length:234         Length:234
##  1st Qu.:18.00   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character
##  Mean   :23.44
##  3rd Qu.:27.00
##  Max.   :44.00
```

Задание 1

Что получится в результате выполнения следующих команд?

```
class("Пример")
class(17)
class("17")
as.character(87.9)
is.integer("1")
as.integer(4.5)
as.numeric(FALSE)
```

Задание 2

Рассмотрим данные про ирисы

```
?iris
data(iris)
```

1. О чем эти данные?
2. Сколько наблюдений и сколько переменных в данных?
3. Какого типа данные?
4. Каковы минимальные и максимальные значения переменных?

Разведочный анализ данных

Что такое **разведочный анализ данных** (Exploratory Data Analysis)? Как вы думаете?

Чеклист разведочного анализа данных

1. Сформулируйте вопрос.
2. Загрузите / подготовьте данные.
3. Проверьте, все ли в порядке с данными.
4. Выполните str().
5. Посмотрите на первые и последние строки датасета.
6. Проверьте числа (“n”s).
7. Сопоставьте как минимум с еще одним источником данных.
4. Сначала проверьте простое решение.
5. Усовершенствуйте свое решение.
6. Сделайте выводы.

1. Формулировка вопроса

- Есть ли какие-то правила и требования к вопросу?
- Является ли хорошим вопрос «Какая модель автомобиля лучше»? Что нужно, чтобы на него ответить?

2. Загрузка данных

Именно к этому пункту относится загрузка данных (в зависимости от формата файла), чистка данных: различные действия с преобразованиями классов, форматирование, исправление ошибок и опечаток и т. д.

Продолжаем работать с данными о характеристиках 38 популярных моделей автомобилей с 1999 по 2008 г.

```
library(ggplot2)
```

Загружаем данные. О чем они?

```
data(mpg)

?ggplot2::mpg
## starting httpd help server ...
##  done
```

3–7. Проверка данных

Посмотрим на данные. Сколько переменных и наблюдений мы имеем? Совпадают ли данные с описанием?

```
dim(mpg)
## [1] 234  11
head(mpg, n = 3)
## # A tibble: 3 <U+00D7> 11
## manufacturer model displ year cyl      trans drv   cty   hwy   fl
##         chr>   <chr> <dbl> <int> <int>  <chr> <chr> <int> <int> <chr>
## 1      audi    a4   1.8  1999    4  auto(l5)  f    18   29    p
## 2      audi    a4   1.8  1999    4 manual(m5) f    21   29    p
## 3      audi    a4   2.0  2008    4 manual(m6) f    20   31    p
## # ... with 1 more variables: class <chr>
str(mpg)
## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11
variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model  : chr  "a4" "a4" "a4" "a4" ...
##  $ displ  : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year   : int  1999 1999 2008 2008 1999 1999 2008 1999 1999
2008 ...
##  $ cyl    : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans  : chr  "auto(l5)" "manual(m5)" "manual(m6)"
"auto(av)" ...
##  $ drv    : chr  "f" "f" "f" "f" ...
##  $ cty    : int  18 21 20 21 16 18 18 18 16 20 ...
```

```
##  $ hwy     : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl      : chr  "p" "p" "p" "p" ...
##  $ class   : chr  "compact" "compact" "compact" "compact" ...
```

Проверим, данные о машинах какого класса у нас есть. Сколько таких классов?

```
table(mpg$class)
##
##    2seater    compact    midsize   minivan    pickup subcompact
##          5         47         41         11         33         35
##        suv
##         62
table(mpg$year)
##
## 1999 2008
##  117  117
```

Как представлены производители в датасете?

```
table(mpg$manufacturer)
##
##       audi  chevrolet      dodge       ford      honda    hyundai
##         18         19         37         25          9         14
##       jeep land rover    lincoln    mercury     nissan    pontiac
##          8          4          3          4         13          5
##     subaru     toyota volkswagen
##         14         34         27
```

Таким образом, мы выполнили п. 4–6 чеклиста.

п. 7 (Сравнение с еще одним источником данных) более актуален для реальных, а не учебных датасетов – похожи ли цифры на правильные (один и тот же порядок, одни и те же единицы измерения)

8. Простое решение

Не нужно сразу строить сложные модели и проводить особые тесты – начните с простого. Одним из удобных методов разведочного анализа являются графики. Они позволяют:

- понять основные особенности данных;
- увидеть простые закономерности;
- заложить основу для построения более сложных моделей;
- выполнить эту часть анализа достаточно быстро.

А сейчас отвлечемся от пунктов чеклиста и перейдем к инструменту, который позволит строить графики.

Пакет ggplot2

```
library(ggplot2)
```

Основные компоненты:

– data frame = источник данных

– aesthetic mappings = как данным сопоставляется цвет / размер / положение (x vs y) и т. д.

– geoms = геометрические объекты (точки, линии, формы)

– facets = графики с условиями (разделениями по классам, например)

– stats = статистические преобразования (сглаживание, разделение на промежутки и т. д.)

– scales = описание шкал (например, male = red, female = blue)

– coordinate system = система координат, в которой строится график

Любой график строится слоями:

`g <- ggplot(data, aes(variable1, variable2))` = создание объекта, соответствующего графику, и указание используемого датасета

`aes(x = variable1, y = variable2)` = описание эстетик (в данном случае – какую переменную мы берем за x, а какую – за y)

`geom_point()` = построение точечного графика на основе указанной ранее информации

Эстетики

`+ geom_point(color, size, alpha)` = определение стиля, согласно которому будет нарисованы точки

Внимание: текущее описание справедливо и для других типов графиков (geom_...)

`color = "blue"` = задание цвета точек

`aes(color = var1)` = задание цвета через категориальную (factor) переменную – каждой категории будет соответствовать свой цвет, отличающийся от остальных

`size = 4` = задание размера точек

`alpha = 0.5` = задание прозрачности точек

Что вообще можно менять на графиках?

– Цвет/заливку (color, fill).

– Размер (size).

– Прозрачность (alpha).

– Форму (shape).

P. S.: На самом деле менять можно почти все, но об этом чуть позднее.

Разведочные графики

Simple Summaries: одна переменная

– Сводная информация (Summary).

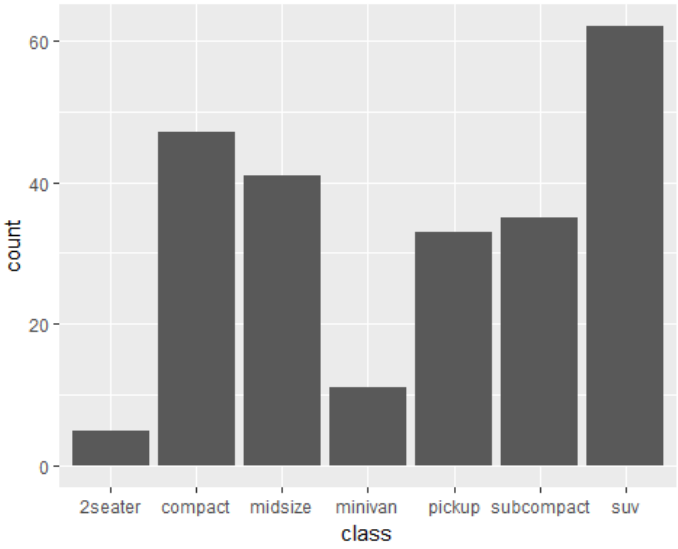
– Столбчатая диаграмма (Barplot).

– Гистограмма (Histogram).
– «Ящик с усами» (Boxplot).

```
summary(mpg)
##  manufacturer       model              displ            year     
##  Length:234        Length:234         Min.   :1.600  Min.   :1999
##  Class :character  Class :character   1st Qu.:2.400  1st Qu.:1999
##  Mode  :character  Mode  :character   Median :3.300  Median :2004
##                                       Mean   :3.472  Mean   :2004
##                                       3rd Qu.:4.600  3rd Qu.:2008
##                                       Max.   :7.000  Max.   :2008
##       cyl           trans               drv                cty       
##  Min.   :4.000  Length:234         Length:234         Min.   : 9.00
##  1st Qu.:4.000  Class :character   Class :character   1st Qu.:14.00
##  Median :6.000  Mode  :character   Mode  :character   Median :17.00
##  Mean   :5.889                                        Mean   :16.86
##  3rd Qu.:8.000                                        3rd Qu.:19.00
##  Max.   :8.000                                        Max.   :35.00
##       hwy             fl               class          
##  Min.   :12.00   Length:234         Length:234        
##  1st Qu.:18.00   Class :character   Class :character  
##  Median :24.00   Mode  :character   Mode  :character  
##  Mean   :23.44                                        
##  3rd Qu.:27.00                                        
##  Max.   :44.00
```

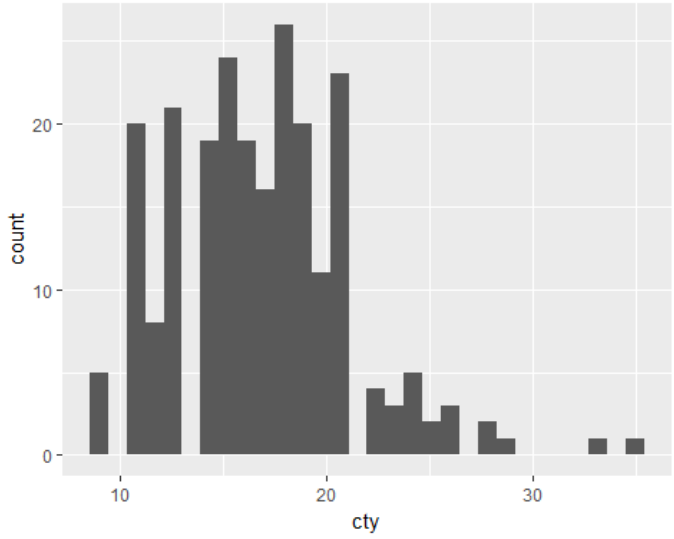
Столбчатая диаграмма:

```
ggplot() +
  geom_bar(data = mpg, aes(x = class))
```



Сколько миль (по городу) проедут автомобили на одном галлоне топлива? (Гистограмма)

```
ggplot() +
  geom_histogram(data = mpg, aes(x = cty))
```



А если нам нужно понять, насколько значения отклоняются от обобщенного показателя?

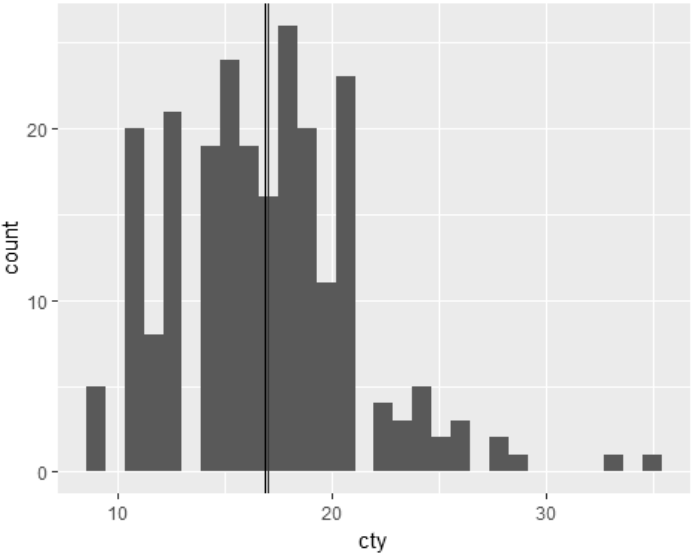
Посчитаем среднее и медиану:

- что такое медиана,
- в чем ее отличие от среднего.

```
?mean
?median

mean(mpg$cty)
## [1] 16.85897
median(mpg$cty)
## [1] 17
```

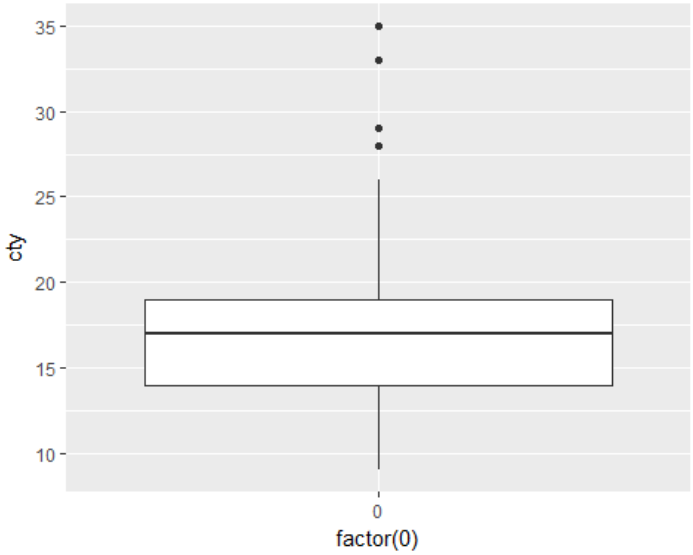
Графики в ggplot2 строятся слоями: можно сохранить результат в переменную, а потом «дорисовывать» следующие части.

```
g <- ggplot() +
  geom_histogram(data = mpg, aes(x = cty))

g + geom_vline(xintercept = mean(mpg$cty)) + geom_vline(xin-
tercept = median(mpg$cty), color="red")
```



Сколько миль (по городу) проедут автомобили на одном галлоне топлива? (Боксплот)

```
ggplot() +
  geom_boxplot(data = mpg, aes(x = factor(0), y = cty))
```



Вопросы для закрепления

1. В чем отличительные черты гистограммы и боксплота, если они визуализируют одни и те же данные? В каких случаях следует использовать боксплот, а в каких гистограмму?

2. Данные каких типов можно использовать для каких графиков?

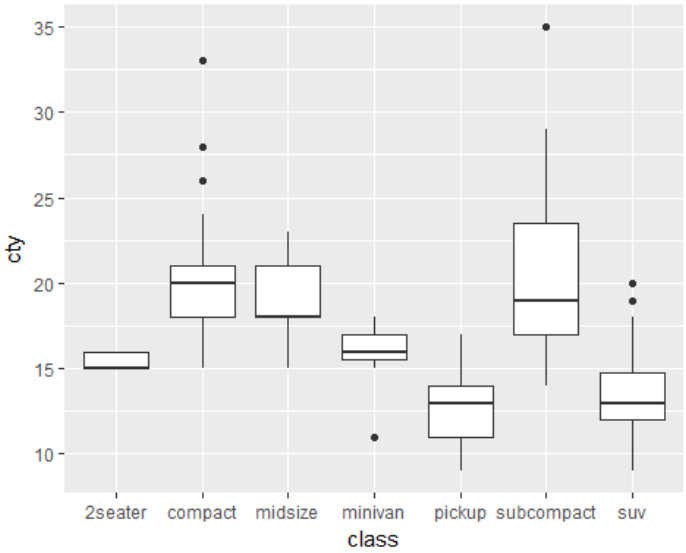
– Столбчатая диаграмма (Barplot)
– Гистограмма (Histogram)
– «Ящик с усами» (Boxplot)

Simple Summaries: две переменных

– Диаграмма рассеяния (Scatterplot)
– Комбинации графиков с одной переменной (Multiple or overlayed 1-D plot)
– Использование цвета, размера, формы для увеличения размерности

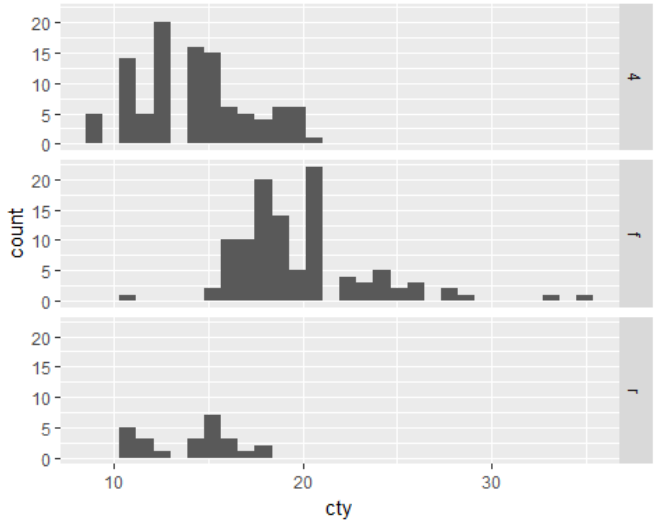
Какой класс автомобилей наиболее экономичный?

```
ggplot() +
  geom_boxplot(data = mpg, aes(x = class, y = cty))
```



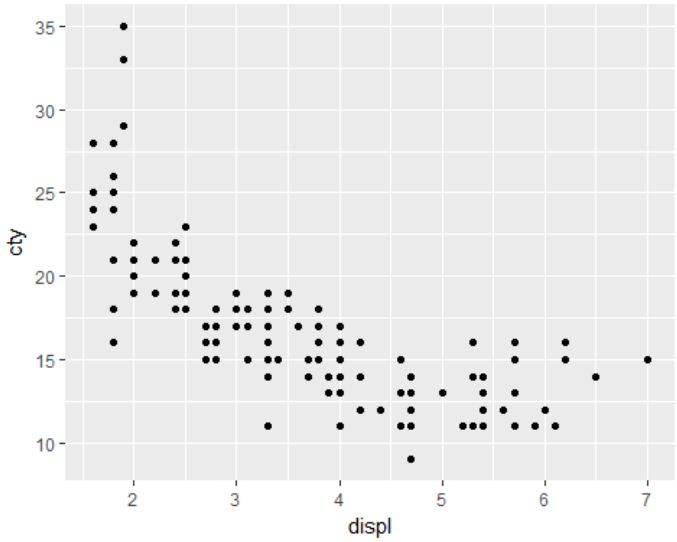
У машин с каким типом привода расход топлива больше?

```
ggplot() +
  geom_histogram(data = mpg, aes(x = cty)) +
  facet_grid(drv~.)
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

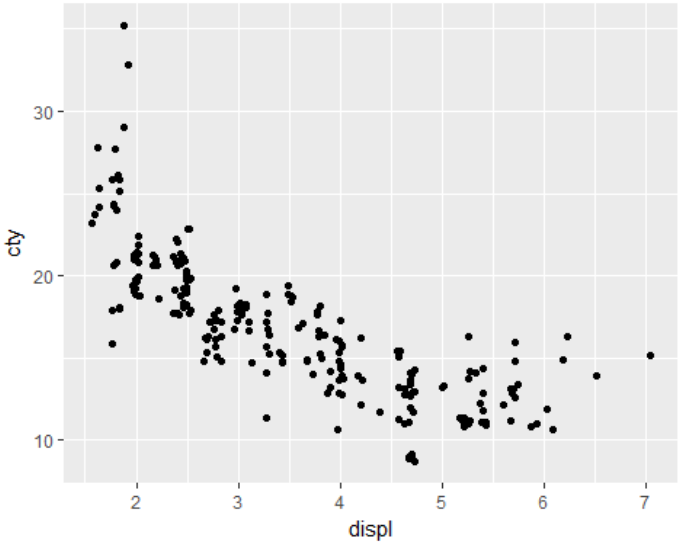


Как связана экономичность автомобиля и объем двигателя?

```
ggplot() +
  geom_point(data = mpg, aes(x = displ, y = cty))
```

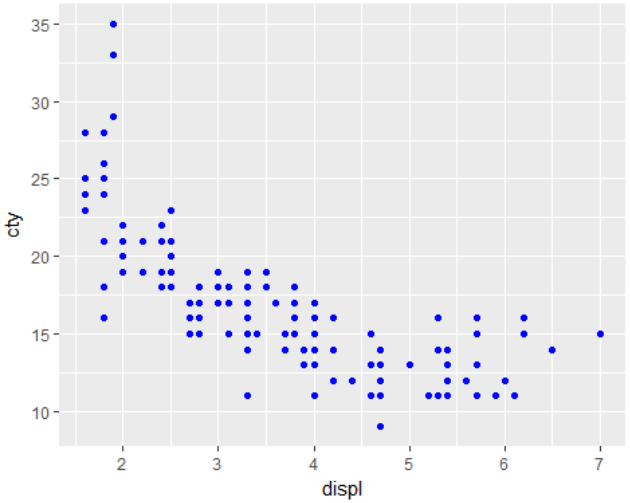



```
ggplot() +
  geom_jitter(data = mpg, aes(x = displ, y = cty))
```



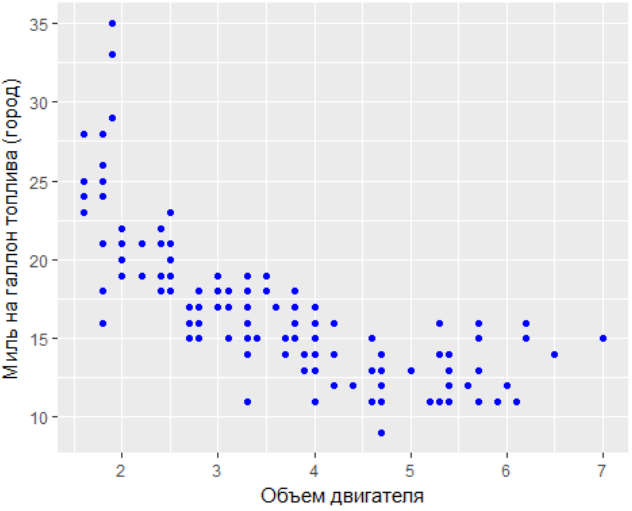
Что делать, если хочется, чтобы этот график выглядел fancy? Раскрасить его!

```
ggplot() +
  geom_point(data = mpg, aes(x = displ, y = cty), color = "blue")
```



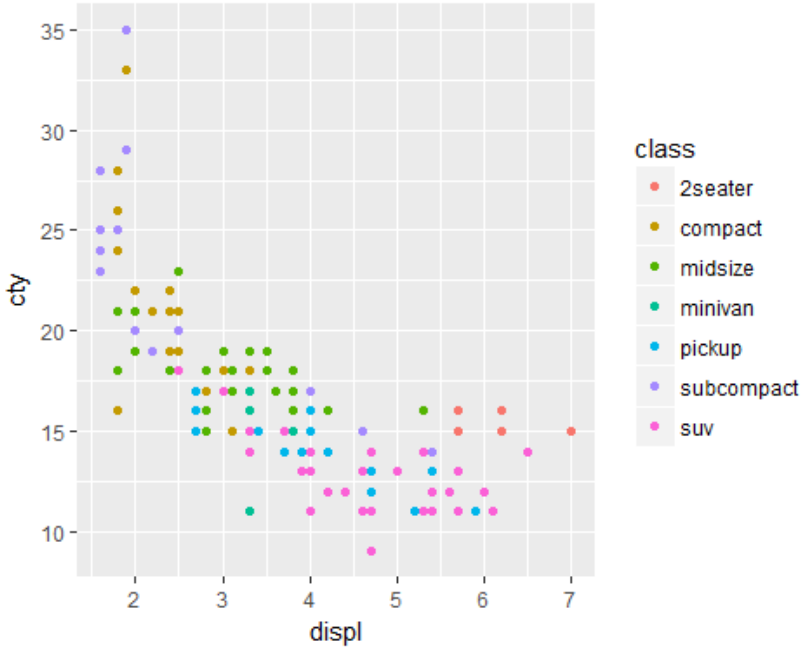
Если еще и друзьям (научному руководителю) показать захотелось, тогда надо добавить понятные подписи к осям!

```
ggplot() +
  geom_point(data = mpg, aes(x = displ, y = cty), color = "blue") +
  xlab("Объем двигателя") +
  ylab("Миль на галлон топлива (город)")
```



А если хочется посмотреть, как это двумерное распределение накладывается на разделение автомобилей на классы?

```
ggplot() +
  geom_point(data = mpg, aes(x = displ, y = cty, color = class))
```



Данные каких типов можно использовать для каких графиков?

9. Совершенствование решения

Простых графиков не всегда достаточно. Иногда, чтобы ответить на вопрос, нужно исследовать более сложные зависимости (но об этом поговорим следующий раз).

10. Выводы

Подходящие ли у нас данные?

Не нужно ли нам собрать что-то дополнительно, чтобы более полно ответить на вопрос?

Корректен ли наш вопрос?

Принципы аналитической графики:

– Показывайте сравнения.
– Показывайте структуру, механизмы, причины и следствия.
– Показывайте многомерность данных.

– Используйте разные подтверждения вашего результата (текст, цифры и т. д.).

– Описывайте и фиксируйте подтверждения/доказательства (так, чтобы зрителю было понятно).

– Главное – содержание, а не форма.

Вопрос для обсуждения: Мы рассматривали разведочный анализ данных и, в частности, разведочные графики (Exploratory Graphs). Чем, по вашему мнению, разведочные графики отличаются от финальных, публикуемых в отчетах, презенациях и т. д.?

Задание 3

Проведите разведочный анализ датасета `msleep` из пакета `ggplot2`

```
?msleep
```

Список библиографических источников

Основной

1. *Горяинова, Е. Р.* Прикладные методы анализа статистических данных: учеб. пособие / Е. Р. Горяинова, А. Р. Панков, Е. Н. Платонов. – М.: Изд. дом Высшей школы экономики, 2012. – 310 с.

2. *Лесковец, Ю.* Анализ больших наборов данных / Ю. Лесковец, А. Раджараман. – М.: ДМК, 2016. – 498 с.

3. *Крянев, А. В.* Метрический анализ и обработка данных / А. В. Крянев, Г. В. Лукин, Д. К. Удумян. – М.: Физматлит, 2012. – 308 с.

4. *Кулаичев, А. П.* Методы и средства комплексного анализа данных: учеб. пособие / А. П. Кулаичев. – М.: Форум, НИЦ ИНФРА-М, 2013. – 512 с.

5. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

6. 1Biostatistics with R: An Introduction to Statistics Through Biological Data (Use R!) / Babak Shahbaba, 2012.

7. Modern Epidemiology Third, Mid-cycle revision Edition / J. Kenneth Rothman, L. Timothy Lash Associate Professor, Sander Greenland, 2012.

8. Biostatistics and Epidemiology: A Primer for Health and Biomedical Professionals 4th ed. / Sylvia Wassertheil-Smoller, Jordan Smoller, 2015.

9. Introductory Time Series with R (Use R!) / Paul S. P. Cowpertwait, Andrew V. Metcalfe, 2009.

Дополнительный

10. *Новикова, Н. М.* Статистические методы в биологии и медицине: учеб.-метод. пособие / Н. М. Новикова. – Минск: МГЭУ им. А. Д. Сахарова, 2009. – 88 с.

11. *Кабаков, Р.* R в действии. Анализ и визуализация данных в программе R / Р. Кабаков. – М.: ДМК, 2016. – 588 с.

12. Биометрика – журнал для медиков и биологов, сторонников доказательной биомедицины [Электронный ресурс]. URL: http://www.-biometrica.tomsk.ru.

13. *Орлов, А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

14. *Чиркин, А. А.* Клинический анализ лабораторных данных / А. А. Чиркин. – Витебск: Медицинская литература, 2008. – 384 с.

15. *Петри, А.* Наглядная статистика в медицине / А. Петри, К. Сэбин. – М.: Изд. дом ГЭОТАР-МЕД, 2003 – 143 с.

Лабораторная работа № 7
Разведочный анализ данных. Агрегация данных

Цель: научиться приемам агрегации данных, которую можно осуществлять при помощи пакета dplyr.

Агрегация данных. Пакет dplyr

Сегодня мы будем учиться агрегации данных, которую можно осуществлять при помощи пакета `dplyr`. Для начала надо загрузить пакет `dplyr` и базу с данными о полетах самолетов, а также пакет `ggplot2`, с которым мы работали в прошлый раз.

Кроме того, обращайте внимание на то, что последние функции пакета dplyr можно найти в dplyr cheatsheet.

```
library(dplyr)
library(nycflights13)
library(ggplot2)
```

Чтобы познакомиться с переменными в базе, введите

```
?flights
```

```
?nycflights13::flights
```

Задание 1

Что за информация содержится в следующих переменных:
- `origin`,
- `distance`,
- `tailnum`,
- `carrier`,
- `dest`?

Посмотрим, сколько наблюдений и переменных в базе, а также выведем 3 первые строчки.

```
dim(flights)
## [1] 336776     19
head(flights, n = 3)
## # A tibble: 3 <U+00D7> 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     1     1      517            515         2      830
## 2  2013     1     1      533            529         4      850
## 3  2013     1     1      542            540         2      923
## # ...with 12 more variables: sched_arr_time <int>, arr_delay <dbl>,
```

```
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>,
dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Удалим все строки с NA (строки, в которых есть хотя бы одно пропущенное значение).

```
flData = nycflights13::flights
flData = na.omit(flData)
```

Посмотрим, сколько осталось наблюдений и переменных в базе.

```
dim(flData)
## [1] 327346     19
head(flData, n = 3)
## # A tibble: 3 <U+00D7> 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     1     1      517            515         2      830
## 2  2013     1     1      533            529         4      850
## 3  2013     1     1      542            540         2      923
## # ... with 12 more variables: sched_arr_time <int>, arr_delay
<dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>,
dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Задание 2

Вспомним, как работает ggplot.

1. Нарисуйте boxplot (ящик с усами), в котором по оси X будут аэропорты Нью-Йорка, из которых вылетают самолеты, а по оси Y – дистанция.

2. Нарисуйте гистограмму по переменной `hour`. Посмотрите, как работают параметры *coord_flip* и *coord_polar*. Вспомните, как менять цвет у графика.

3. Нарисуйте bar chart по переменной `carrier`. Закрасьте столбцы таким образом, чтобы можно было видеть, сколько перелетов для каждого перевозчика относятся к каждому из трех аэропортов, работающих в Нью-Йорке.

Переходим к исследованию более сложных зависимостей (и более сложным условиям).

Сначала напоминание про основную пунктуацию:

- < – меньше
- == – равно
- != – не равно
- | – или (выполнено хотя бы одно из условий)
- & – и (выполнены оба условия)
- ! – не
- : – интервал (a:b - все от a до b)

Переходим к основным функциям **dplyr.**

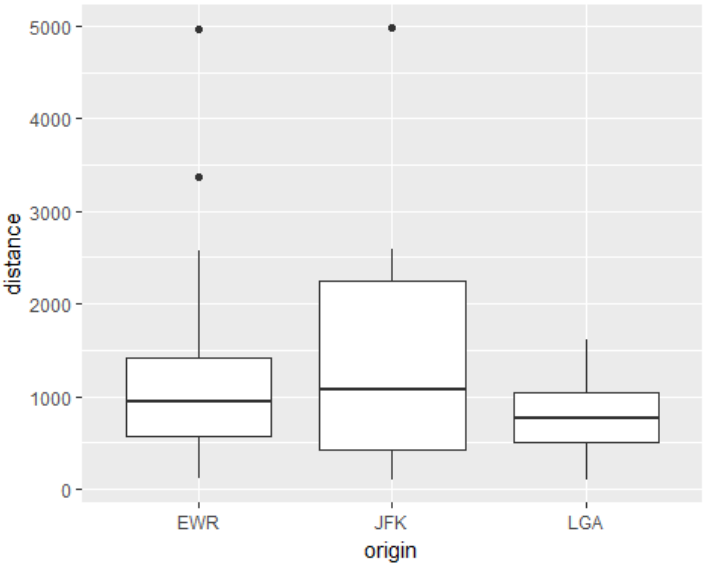
filter

Допустим, нас интересуют только полеты в летние месяцы

```
#?filter
flData = filter(flData, month == 6 | month == 7 | month == 8)
```

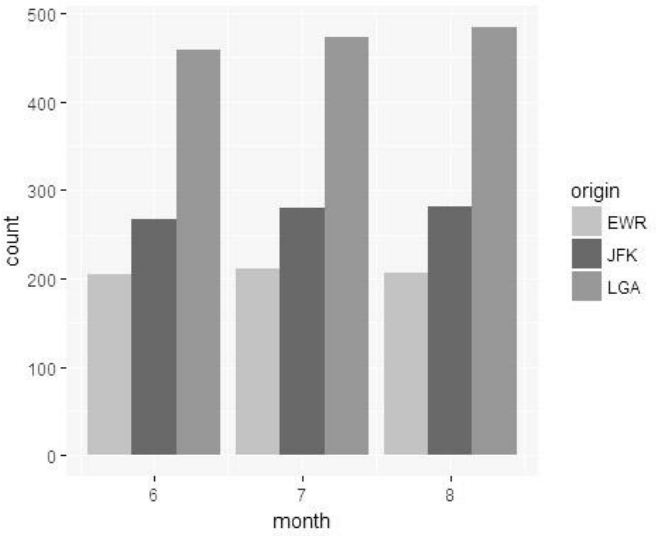
Внимание! Поскольку мы сохраняем результат фильтрации в ту же самую переменную `flights`, то дальше мы можем работать уже не с полными данными, а только с полетами в летние месяцы. Если нужно оставить возможность работы с исходными данными, сохраняйте результат в новую переменную.

```
ggplot(data = flData) + geom_boxplot(aes(x = origin, y = dis-
tance))
```



Теперь отберем только те рейсы, в которых пункт назначения Майами (MIA - Miami International Airport)

```
flData = filter(flData, dest == "MIA")
ggplot(data = flData) + geom_bar(aes(x = month, fill =
origin), position = "dodge")
```



arrange

Упорядочим рейсы по времени вылета

```
#?arrange
flData = arrange(flData, year, month, day)
head(flData)
## # A tibble: 6 <U+00D7> 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     6     1      539            540        -1      832
## 2  2013     6     1      555            605       -10      838
## 3  2013     6     1      558            600        -2      832
## 4  2013     6     1      609            615        -6      852
## 5  2013     6     1      656            700        -4      939
## 6  2013     6     1      718            725        -7     1016
## # ... with 12 more variables: sched_arr_time <int>, arr_delay
<dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>,
dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

И по задержке прилета в убывающем порядке

```
flData = arrange(flData, desc(dep_delay))
head(flData)
## # A tibble: 6 <U+00D7> 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     7    21     1555            615       580     1955
## 2  2013     6    30     1842           1125       437     2229
## 3  2013     8    28     2315           1559       436      158
## 4  2013     8     5     1917           1240       397     2226
## 5  2013     7    28      105           1925       340      401
## 6  2013     7    22     2119           1559       320      116
## # ... with 12 more variables: sched_arr_time <int>, arr_delay
<dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>,
dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

select

Отберем для дальнейшего анализа только данные о номере рейса и задержках, удалим из рассмотрения переменную `arr_time`

```
#?select
str(flData)
## Classes 'tbl_df', 'tbl' and 'data.frame': 2865 obs. of  19 variables:
##  $ year         : int  2013 2013 2013 2013 2013 2013 2013 2013 2013
2013 ...
##  $ month        : int  7 6 8 8 7 7 7 6 7 6 ...
##  $ day          : int  21 30 28 5 28 22 14 24 12 18 ...
##  $ dep_time     : int  1555 1842 2315 1917 105 2119 1420 2111 2002
1951 ...
## $ sched_dep_time: int  615 1125 1559 1240 1925 1559 905 1559 1455
1455 ...
##  $ dep_delay    : num  580 437 436 397 340 320 315 312 307 296 ...
##  $ arr_time     : int  1955 2229 158 2226 401 116 1709 28 2308
2309 ...
##  $ sched_arr_time: int  910 1440 1919 1545 2243 1922 1225 1925
1830 1830 ...
##  $ arr_delay    : num  645 469 399 401 318 354 284 303 278 279 ...
##  $ carrier      : chr  "AA" "AA" "DL" "UA" ...
##  $ flight       : int  1895 2099 1373 1186 2139 1373 874 1373 779
779 ...
##  $ tailnum      : chr  "N3EMAA" "N3BTAA" "N3763D" "N12109" ...
##  $ origin       : chr  "EWR" "LGA" "JFK" "EWR" ...
##  $ dest         : chr  "MIA" "MIA" "MIA" "MIA" ...
##  $ air_time     : num  177 202 134 134 150 138 135 143 153 164 ...
##  $ distance     : num  1085 1096 1089 1085 1096 ...
##  $ hour         : num  6 11 15 12 19 15 9 15 14 14 ...
##  $ minute       : num  15 25 59 40 25 59 5 59 55 55 ...
##  $ time_hour    : POSIXct, format: "2013-07-21 06:00:00" "2013-06-
30 11:00:00" ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:9430] 472 478
616 644 726 734 755 839 840 841 ...
##   .. ..- attr(*, "names")= chr [1:9430] "472" "478" "616" "644" ...
flData = dplyr::select(flData, tailnum, dep_delay:arr_delay)
flData = dplyr::select(flData, -arr_time)
str(flData)
## Classes 'tbl_df', 'tbl' and 'data.frame':    2865 obs. of  4 variables:
##  $ tailnum       : chr  "N3EMAA" "N3BTAA" "N3763D" "N12109" ...
##  $ dep_delay     : num  580 437 436 397 340 320 315 312 307 296 ...
##  $ sched_arr_time: int  910 1440 1919 1545 2243 1922 1225 1925
1830 1830 ...
##  $ arr_delay     : num  645 469 399 401 318 354 284 303 278 279 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:9430] 472 478
```

```
616 644 726 734 755 839 840 841 ...
##   .. ..- attr(*, "names")= chr [1:9430] "472" "478" "616" "644" ...
```

rename

Переименуем переменную `tailnum` для единообразия в `tail_num`

```
flData = rename(flData, tail_num = tailnum)
```

summarise & mutate

Задание 3

В чем отличия функций `summarise()` и `mutate()`?

```
?summarise
## starting httpd help server ...
##  done
?mutate
flData = mutate(flData, dep_arr_dif = arr_delay - dep_delay)
summarise(flData, mean_dif = mean(dep_arr_dif))
## # A tibble: 1 <U+00D7> 1
##    mean_dif
##       <dbl>
## 1 -7.719372
```

Что означает полученное число?

group_by

Если нам нужно значение не для всей выборки, а для каждого из рейсов, то мы сначала группируем по номеру рейса, а затем вычисляем нужную характеристику

```
flData = group_by(flData, tail_num)
tail_dif = dplyr::summarise(flData, median_dif = me-
dian(dep_arr_dif))
tail_dif
## # A tibble: 790 <U+00D7> 2
##    tail_num median_dif
##       <chr>      <dbl>
## 1    N12109      -11.0
## 2    N12114      -12.0
## 3    N12116      -28.0
## 4    N12125       12.0
## 5    N12216       -1.0
## 6    N12218       13.5
```

```
## 7    N12238        0.0
## 8    N13110      -20.0
## 9    N13113        1.0
## 10   N13138        0.0
## # ... with 780 more rows
```

join

Объединение нескольких датасетов в один по одной из переменных

```
?join
```

В чем разница `inner_join()` и `left_join()`?

```
flData = inner_join(x = flData, y = tail_dif, by = "tail_num")
str(flData)
## Classes 'grouped_df', 'tbl_df', 'tbl' and 'data.frame':  2865
obs. of  6 variables:
##  $ tail_num      : chr  "N3EMAA" "N3BTAA" "N3763D" "N12109" ...
##  $ dep_delay     : num  580 437 436 397 340 320 315 312 307 296 ...
##  $ sched_arr_time: int  910 1440 1919 1545 2243 1922 1225 1925 1830
1830 ...
##  $ arr_delay     : num  645 469 399 401 318 354 284 303 278 279 ...
##  $ dep_arr_dif   : num  65 32 -37 4 -22 34 -31 -9 -29 -17 ...
##  $ median_dif    : num  2 -12 -37 -11 -18 -3.5 -27 -22 -6 -14.5 ...
##  - attr(*, "vars")=List of 1
##   ..$ : symbol tail_num
flData = arrange(flData, median_dif)
flData
## Source: local data frame [2,865 x 6]
## Groups: tail_num [790]
##
##    tail_num dep_delay sched_arr_time arr_delay dep_arr_dif me-
dian_dif
##       <chr>     <dbl>          <int>     <dbl>       <dbl>     <dbl>
## 1   N3AKAA        -9         1305       -57         -48       -48
## 2   N374AA         3         1835       -43         -46       -46
## 3   N25705        -2         2332       -46         -44       -44
## 4   N393DA        26         1919       -16         -42       -42
## 5   N385DN         2         1919       -38         -40       -40
## 6   N6704Z       107         2141        68         -39       -39
## 7   N957DL        69         1558        23         -46       -39
## 8   N342AA        30         1835        -9         -39       -39
## 9   N957DL        13         1558       -19         -32       -39
## 10  N3KLAA        30         1430        -6         -36       -38
## # ... with 2,855 more rows
```

Два основных способа работы с `dplyr`

Вернемся к первоначальному датасету

```
flights = nycflights13::flights
flights = na.omit(flights)
```

1. Мы можем выполнять операции пошагово (step-by-step), сохраняя промежуточные результаты

```
a1 = group_by(flights, year, month, day)
a2 = select(a1, arr_delay, dep_delay)
## Adding missing grouping variables: `year`, `month`, `day`
a3 = summarise(a2,
  arr = mean(arr_delay, na.rm = TRUE),
  dep = mean(dep_delay, na.rm = TRUE))
a4 = filter(a3, arr > 30 | dep > 30)
```

Какие данные, как вы думаете, остались в `a4`?

2. Либо можем работать с помощью pipes (%>%)

```
delays = flights %>%
  group_by(year, month, day) %>%
  select(arr_delay, dep_delay) %>%
  summarise(
    arr = mean(arr_delay, na.rm = TRUE),
    dep = mean(dep_delay, na.rm = TRUE)
  ) %>%
  filter(arr > 30 | dep > 30)
## Adding missing grouping variables: `year`, `month`, `day`
```

Задание 4

1. Загрузить базу по колледжам

```
colleges = ISLR::College
?ISLR::College
```

2. Оставить в базе только вузы с Graduation Rate меньше 50 %

```
colleges = filter(colleges, ...)
```

3. Создать 2 новые колонки. Первая – отношение принятых (одобренных) заявлений на поступление к количеству полученных заявлений.

Вторая – отношение количества поступивших студентов к количеству принятых заявлений

```
colleges = mutate(colleges, ...)
```

4. Оставить только две новые колонки, созданные на предыдущем шаге, и колонку, соответствующую типу вуза (является ли вуз частным или государственным).

```
colleges = dplyr::select(colleges, ...)
```

5. Построить графики для сравнения доли принятых заявлений между типами вузов и сравнения доли поступивших студентов между типами вузов

```
ggplot(data = colleges) + ...
```

6. Сгруппировать базу по типу вуза (частный или государственный), посчитать средние значения по оставшимся двум колонкам.

```
colleges %>% group_by(...) %>% dplyr::summarize(...)
```

7. Загрузите базу заново

```
colleges = ISLR::College
```

8. Постройте график, чтобы сравнить каких колледжей в базе больше, частных или государственных

```
ggplot(data = colleges) + ...
```

9. Создайте новую колонку, отражающую данные: приходится ли на одного преподавателя больше 13 студентов или нет

```
colleges = mutate(colleges, ...)
```

10. Постройте график, отражающий взаимосвязь между типом колледжа и тем, приходится ли в нем на одного преподавателя больше 13 студентов или нет

```
ggplot(data = colleges) + ...
```

11. Постройте график, отражающий взаимосвязь между затратами на обучение одного студента и количеством поданных заявлений в колледж

```
ggplot(data = colleges) + ...
```

12. Посчитайте среднее количество fulltime undergraduates и parttime undergraduates в частных и государственных колледжах

```
colleges %>% group_by(...) %>% dplyr::summarize(...)
```

13. Выберите колледжи, в которых суммарные затраты (personal, books, room) не превышают 6000. Каких колледжей в этой категории больше?

14. Сформулируйте свой вопрос по рассматриваемой базе. Выполните вычисления / постройте график для ответа на него.

Список библиографических источников

Основной

1. *Горяинова, Е. Р.* Прикладные методы анализа статистических данных: учеб. пособие / Е. Р. Горяинова, А. Р. Панков, Е. Н. Платонов. – М.: Изд. дом Высшей школы экономики, 2012. – 310 с.

2. *Лесковец, Ю.* Анализ больших наборов данных / Ю. Лесковец, А. Раджараман. – М.: ДМК, 2016. – 498 с.

3. *Крянев, А. В.* Метрический анализ и обработка данных / А. В. Крянев, Г. В. Лукин, Д. К. Удумян. – М.: Физматлит, 2012. – 308 с.

4. *Кулаичев, А. П.* Методы и средства комплексного анализа данных: учеб. пособие / А. П. Кулаичев. – М.: Форум, НИЦ ИНФРА-М, 2013. – 512 с.

5. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

6. 1Biostatistics with R: An Introduction to Statistics Through Biological Data (Use R!) / Babak Shahbaba, 2012.

7. Modern Epidemiology Third, Mid-cycle revision Edition / J. Kenneth Rothman, L. Timothy Lash Associate Professor, Sander Greenland, 2012.

8. Biostatistics and Epidemiology: A Primer for Health and Biomedical Professionals 4th ed. / Sylvia Wassertheil-Smoller, Jordan Smoller, 2015.

9. Introductory Time Series with R (Use R!) / Paul S. P. Cowpertwait, Andrew V. Metcalfe, 2009.

Дополнительный

10. *Новикова, Н. М.* Статистические методы в биологии и медицине: учеб.-метод. пособие / Н. М. Новикова. – Минск: МГЭУ им. А. Д. Сахарова, 2009. – 88 с.

11. *Кабаков, Р.* R в действии. Анализ и визуализация данных в программе R / Р. Кабаков. – М.: ДМК, 2016. – 588 с.

12. Биометрика – журнал для медиков и биологов, сторонников доказательной биомедицины [Электронный ресурс]. URL: http://www.-biometrica.tomsk.ru.

13. *Орлов, А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

14. *Чиркин, А. А.* Клинический анализ лабораторных данных / А. А. Чиркин. – Витебск: Медицинская литература, 2008. – 384 с.

15. *Петри, А.* Наглядная статистика в медицине / А. Петри, К. Сэбин. – М.: Изд. дом ГЭОТАР-МЕД, 2003 – 143 с.

Лабораторная работа № 8
Дата и время. Работа с данными

Цель: научиться приемам работы с временными данными при помощи пакета **`lubridate`**.

Работа с временными данными. Пакет lubridate

Для начала загрузим базу, с которой будем работать. В базе содержатся информация о сообщений очевидцев НЛО.

Файл достаточно большой, поэтому используем пакет для быстрой загрузки таблиц данных `readr`.

```
library(readr)
ufo = read_tsv("ufo_awesome.tsv", col_names=F)
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_integer(),
##   X3 = col_character(),
##   X4 = col_character(),
##   X5 = col_character(),
##   X6 = col_character()
## )
## Warning: 45 parsing failures.
##  row col  expected     actual              file
##  755  -- 6 columns 7 columns  'ufo_awesome.tsv'
## 1167  -- 6 columns 7 columns  'ufo_awesome.tsv'
## 1569  -- 6 columns 13 columns 'ufo_awesome.tsv'
## 1571  -- 6 columns 7 columns  'ufo_awesome.tsv'
## 1659  -- 6 columns 10 columns 'ufo_awesome.tsv'
## .... ... ......... .......... ..................
## See problems(...) for more details.
# naming columns
colnames(ufo) = c("Sighted", "Reported", "Location", "Shape",
"Duration", "Description")
```

Посмотрим на данные

```
head(ufo)
## # A tibble: 6 <U+00D7> 6
##    Sighted Reported             Location Shape Duration
##      <chr>    <int>                <chr> <chr>    <chr>
## 1 19951009 19951009        Iowa City, IA  <NA>     <NA>
## 2 19951010 19951011        Milwaukee, WI  <NA>   2 min.
```

```
## 3 19950101 19950103          Shelton, WA  <NA>     <NA>
## 4 19950510 19950510         Columbia, MO  <NA>   2 min.
## 5 19950611 19950614          Seattle, WA  <NA>     <NA>
## 6 19951025 19951024 Brunswick County, ND  <NA>  30 min.
## # ... with 1 more variables: Description <chr>
tail(ufo)
## # A tibble: 6 <U+00D7> 6
##    Sighted Reported              Location   Shape    Duration
##      <chr>    <int>                 <chr>   <chr>       <chr>
## 1 19960815 20000722 Kelowna (Canada),  BC    egg    2 minutes
## 2 20000731 20000801         Portland,  OR  fireball  1.5 sec.
## 3 20000623 20000624        Morrilton,  AR rectangle     2nin.
## 4 20000811 20000811        Arlington,  WA  triangle   15 Min.
## 5 19980215 20000819       Fort Lewis,   WA   <NA> 45mins to 1hr
## 6 20000708 20000709            Ceres,  CA   unknown 8 minutes
## # ... with 1 more variables: Description <chr>
```

Как вы думаете, что означают первые две колонки?
Как мы можем посчитать разницу между этими значениями?

Работа с датами

Для удобной работы с датами есть пакет `lubridate`. Загрузим его

```
library(lubridate)
##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##     date
```

В каком порядке расположена информация о дате (день, месяц, год)?

```
ufo$Sighted = ymd(ufo$Sighted)
## Warning: 124 failed to parse.
ufo$Reported = ymd(ufo$Reported)
```

Рассмотрим еще примеры

```
date1 = "18-10-2016"
date2 = "10/30/2016"
date3 = "23/3/2016 15:20"
date4 = today()
```

Сначала посмотрим, что же такое `date4`

```
date4
## [1] "2017-03-23"
class(date4)
## [1] "Date"
```

Теперь посмотрим на остальные элементы

```
class(date1)
## [1] "character"
class(date2)
## [1] "character"
class(date3)
## [1] "character"
# date2 - date1
```

Для удобной работы приведем остальные строки к единому формату даты

```
#date1 = "18-10-2016"
date1 = dmy(date1)
#date2 = "10/30/2016"
date2 = mdy(date2)
#date3 = "23/3/2016 15:20"
date3 = dmy_hm(date3)

class(date1)
## [1] "Date"
class(date2)
## [1] "Date"
class(date3)
## [1] "POSIXct" "POSIXt"
date1
## [1] "2016-10-18"
date2
## [1] "2016-10-30"
date3
## [1] "2016-03-23 15:20:00 UTC"
```

Из дат в таком формате легко извлекать отдельные части:

```
year(date1)
## [1] 2016
```

```
month(date2)
## [1] 10
month(date2, label = TRUE)
## [1] Oct
## 12 Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug <
Sep < ... < Dec
wday(date2)
## [1] 1
wday(date2, label = TRUE)
## [1] Sun
## Levels: Sun < Mon < Tues < Wed < Thurs < Fri < Sat
yday(date3)
## [1] 83
minute(date3)
## [1] 20
```

Или присваивать им новые значения

```
second(date3) = 45
date3
## [1] "2016-03-23 15:20:45 UTC"
```

Когда мы работаем со временем, нам иногда важно учитывать часовой пояс (time zone). По умолчанию используется UTC – Universal Time Coordinate (нулевой меридиан). Указывать часовой пояс можно во время преобразования строки в дату `date3 = dmy_hm(date3, tz = "Europe/Moscow")` или, уже преобразованного значения

```
date3 = force_tz(date3, tzone = "Europe/Moscow")
date3
## [1] "2016-03-23 15:20:45 MSK"
```

Подробнее о часовых поясах в R можно почитать в справке `?timezones`, `?OlsonNames`, а обозначения найти, например, в Википедии.

Кроме того, мы всегда можем посмотреть, какое время будет соответствовать исследуемому в другом часовом поясе, например, для Pacific Daylight Time (North America)

```
with_tz(date3, tzone = "US/Pacific")
## [1] "2016-03-23 05:20:45 PDT"
```

Арифметика с датами и временные интервалы

Одним из важных свойств пакета `lubridate` является возможность осуществлять арифметические операции с датами. При этом мы можем выражать временные данные в разных форматах (месяцы, годы, секунды, минуты...).

```
date1 - years(2)
## [1] "2014-10-18"
difference = date2 - date1
difference
## Time difference of 12 days
difference/ddays(1)
## [1] 12
difference/dweeks(1)
## [1] 1.714286
difference/dhours(1)
## [1] 288
```

Больше про функции lubridate: https://cran.r-project.org/web/packages/lubridate/vignettes/lubridate.html

Вернемся к НЛО. Теперь можем посчитать разницу между датой наблюдения и сообщения

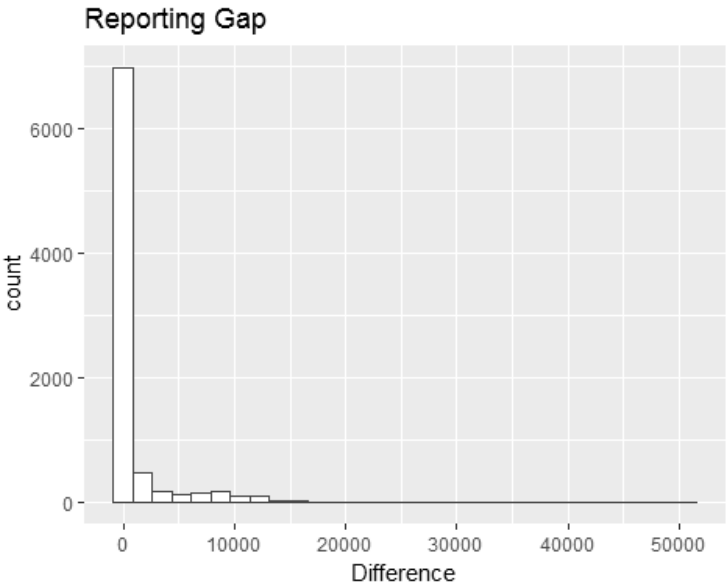
```
library(dplyr)
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
ufo = mutate(ufo, Difference = (Reported - Sighted)/ddays(1))
```

У нас есть значения, у которых разница отрицательна. Почему так может быть?

Удалим их из данных

```
ufo = filter(ufo, Difference >= 0)
library(ggplot2)
ggplot() +
```

```
  geom_histogram(data = ufo, aes(x = Difference),
                 colour = "darkgreen", fill="white") +
  ggtitle("Reporting Gap")
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

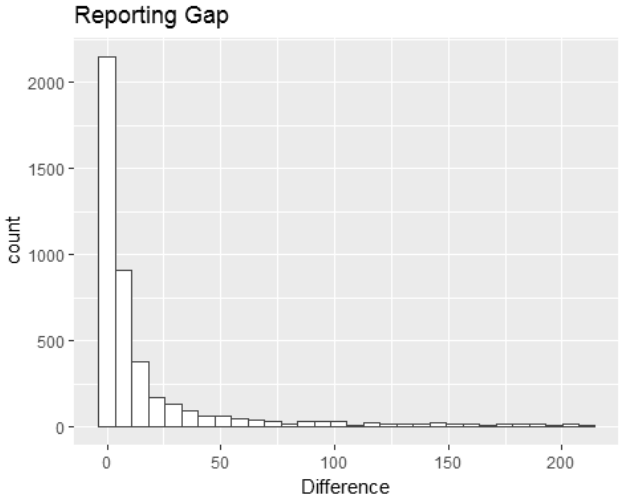


Большая часть значений близка к 0. Посмотрим на возможные значения

```
summary(ufo$Difference)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       1       4    1088     179   50750
```

Рассмотрим только значения, попавшие между 1 и 3 квартилем

```
ufo2 = filter(ufo, Difference > 0 & Difference < 213)
ggplot() +
  geom_histogram(data = ufo2, aes(x = Difference),
                 colour = "darkgreen", fill="white") +
  ggtitle("Reporting Gap")
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Задание 1

1. Постройте график, показывающий распределение наблюдений НЛО (`Sighted`) по годам (чтобы оставить от даты только год, нужно использовать функцию `year()`). Ограничьте график периодом с 1998 по 2008 г.

```
library(lubridate)
ufo3 = ufo
ufo3$Sighted = year(ufo3$Sighted)
```

2. Постройте график, показывающий распределение наблюдений НЛО (`Sighted`) по дням недели (функция `wday()`).

3. Оставьте только данные, соответствующие наблюдениям после 2004 г. Дальше работаем только с этой подвыборкой.

4. Оставьте только те данные, в которых указана форма НЛО (`Shape`). Условие на «непропущенные данные» можно задать как `!is.na(Shape)`.

5. НЛО какой формы наблюдали чаще после 2004 г.? Посчитайте и покажите на графике.

5. В какой день недели чаще наблюдали НЛО в форме сферы (sphere)?

7. НЛО какой формы имеет наибольшую среднюю разницу между временем наблюдения и сообщения (`Difference`)? Для ответа на этот вопрос используйте функции группировки (`group_by()`) и (`summarize()`) агрегации из пакета dplyr.

Список библиографических источников

Основной

1. *Горяинова, Е. Р.* Прикладные методы анализа статистических данных: учеб. пособие / Е. Р. Горяинова, А. Р. Панков, Е. Н. Платонов. – М.: Изд. дом Высшей школы экономики, 2012. – 310 с.

2. *Лесковец, Ю.* Анализ больших наборов данных / Ю. Лесковец, А. Раджараман. – М.: ДМК, 2016. – 498 с.

3. *Крянев, А. В.* Метрический анализ и обработка данных / А. В. Крянев, Г. В. Лукин, Д. К. Удумян. – М.: Физматлит, 2012. – 308 с.

4. *Кулаичев, А. П.* Методы и средства комплексного анализа данных: учеб. пособие / А. П. Кулаичев. – М.: Форум, НИЦ ИНФРА-М, 2013. – 512 с.

5. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

6. 1Biostatistics with R: An Introduction to Statistics Through Biological Data (Use R!) / Babak Shahbaba, 2012.

7. Modern Epidemiology Third, Mid-cycle revision Edition / J. Kenneth Rothman, L. Timothy Lash Associate Professor, Sander Greenland, 2012.

8. Biostatistics and Epidemiology: A Primer for Health and Biomedical Professionals 4th ed. / Sylvia Wassertheil-Smoller, Jordan Smoller, 2015.

9. Introductory Time Series with R (Use R!) / Paul S. P. Cowpertwait, Andrew V. Metcalfe, 2009.

Дополнительный

10. *Новикова, Н. М.* Статистические методы в биологии и медицине: учеб.-метод. пособие / Н. М. Новикова. – Минск: МГЭУ им. А. Д. Сахарова, 2009. – 88 с.

11. *Кабаков, Р.* R в действии. Анализ и визуализация данных в программе R / Р. Кабаков. – М.: ДМК, 2016. – 588 с.

12. Биометрика – журнал для медиков и биологов, сторонников доказательной биомедицины [Электронный ресурс]. URL: http://www.-biometrica.tomsk.ru.

13. *Орлов, А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

14. *Чиркин, А. А.* Клинический анализ лабораторных данных / А. А. Чиркин. – Витебск: Медицинская литература, 2008. – 384 с.

15. *Петри, А.* Наглядная статистика в медицине / А. Петри, К. Сэбин. – М.: Изд. дом ГЭОТАР-МЕД, 2003 – 143 с.

Лабораторная работа № 9
Регрессионные модели зависимостей между количественными переменными

Мы рассмотрим:

• Базовые идеи корреляционного анализа.

• Проблему двух статистических подходов: «Тестирование гипотез vs. построение моделей».

• Разнообразие статистических моделей.

• Основы регрессионного анализа.

Вы сможете:

• Оценить взаимосвязь между измеренными величинами.

• Объяснить что такое линейная модель.

• Формализовать запись модели в виде уравнения.

• Подобрать модель линейной регрессии.

• Проверить состоятельность модели при помощи t-критерия или F-критерия.

• Оценить предсказательную силу модели.

Знакомимся с даными

Зависит ли уровень интеллекта от размера головного мозга?

• Было исследовано 20 девушек и 20 молодых людей (праворуких, англоговорящих, не склонных к алкоголизму, наркомании и прочим смещающим воздействиям).

• У каждого индивида определяли биометрические параметры: вес, рост, размер головного мозга (количество пикселей на изображении ЯМР сканера).

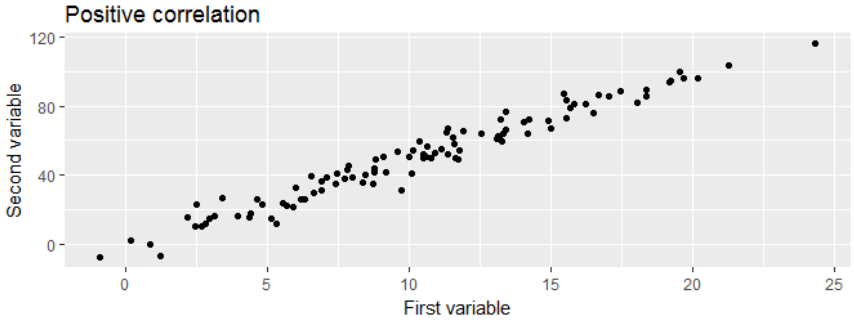
• Интеллект был протестирован с помощью IQ тестов.

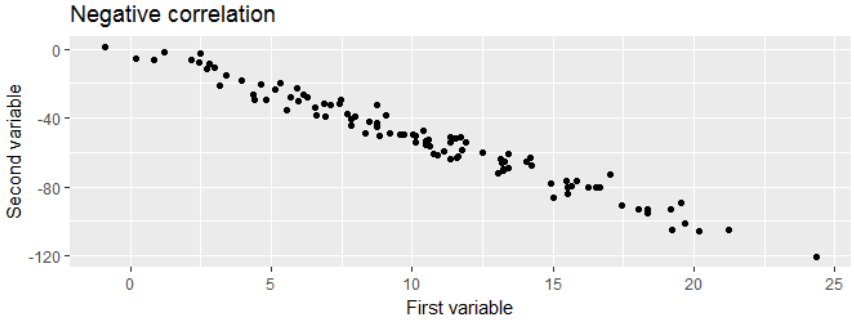
Пример взят из работы: *Willerman, L., Schultz, R., Rutledge, J. N., and Bigler, E.* In Vivo Brain Size and Intelligence // Intelligence. – 1991. – No. 15. – P. 223–228. Данные представлены в библиотеке *«The Data and Story Library»*. URL: http://lib.stat.cmu.edu/DASL/

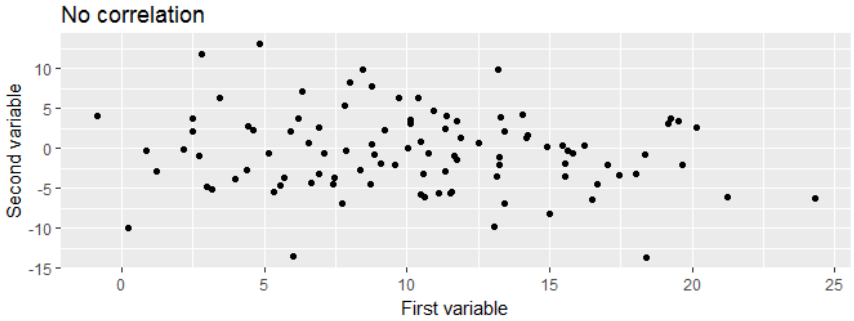
Посмотрим на датасет

```
brain <- read.csv("IQ_brain.csv", header = TRUE)
head(brain)
##   Gender FSIQ VIQ PIQ Weight Height MRINACount
## 1 Female  133 132 124    118   64.5     816932
## 2   Male  140 150 124     NA   72.5    1001121
## 3   Male  139 123 150    143   73.3    1038437
## 4   Male  133 129 128    172   68.8     965353
## 5 Female  137 132 134    147   65.0     951545
## 6 Female   99  90 110    146   69.0     928799
```

Вспомним: ***Сила и направление связи между величинами***







Основы корреляционного анализа

Коэффициенты корреляции и условия их применимости

Коэффициент	Фукция	Особенности примененения
Коэффициент Пирсона	cor(x,y,method="pearson")	Оценивает связь двух нормально распределенных величин. Выявляет только лиейную составляющую взамосвязи
Ранговые коэффициенты (коэффициент Спирмена, Кэндалла)	cor(x,y,method="spirman")cor(x,y,method="kendall")	Не зависят от формы распределения. Могут оценивать связь для любых монотонных зависимостей

Оценка достоверности коэффициентов корреляции

• Коэффициент корреляции – это статистика, значение которой описывает степень взаимосвязи двух сопряженных переменных. Следовательно, применима логика статистического критерия.

• Нулевая гипотеза $H_0: r = 0$.

• Бывают двусторонние $H_a: r \neq 0$ и односторонние критерии $H_a: r > 0$ или $H_a: r < 0$.

• Ошибка коэффициента Пирсона: $SE_r = \sqrt{\frac{1-r^2}{n-2}}$.

• Стандартизованная величина $t = \frac{r}{SE_r}$ подчиняется распределению Стьюдента с парметром $df = n - 2$.

• Для ранговых коэффициентов существует проблема «совпадающих рангов» (tied ranks), что приводит к приблизительной оценке r и приблизительной оценке уровня значимости.

• Достоверность коэффициента кореляции можно оценить пермутационным методом.

Задание

• Определите силу и направление связи между всеми парами исследованных признаков.

• Постройте точечную диаграмму, отражающую взаимосвязь между результатами IQ-теста (PIQ) и размером головного мозга (MRINACount).

• Оцените достоверность значения коэффициента корреляции Пирсона между этими двумя переменными.

Hint 1. Обратите внимание на то, что в датафрейме есть пропущенные значения. Изучите, как работают с NA функции, вычисляющие коэффициенты корреляции.

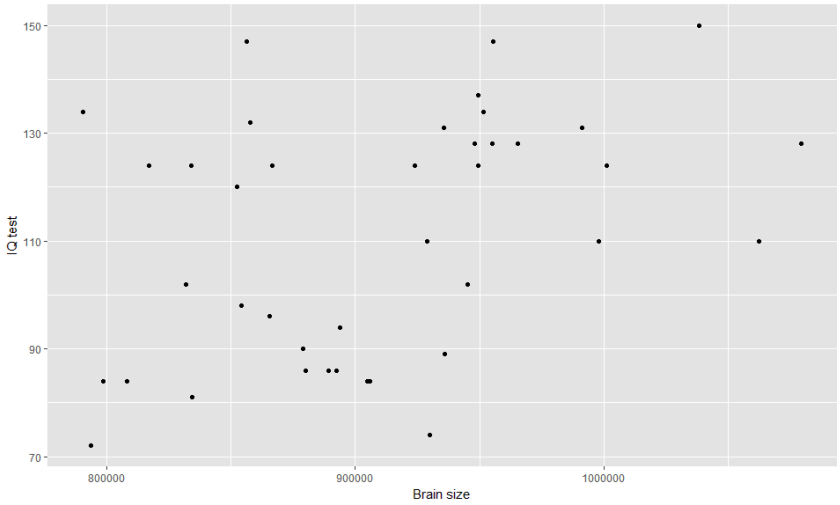
Hint 2. Для построения точечной диаграммы вам понадобится geom_point().

Решение

```
cor(brain[,2:6],
    use = "pairwise.complete.obs")
##             FSIQ      VIQ       PIQ    Weight   Height
## FSIQ    1.00000  0.94664  0.934125 -0.051483 -0.08600
## VIQ     0.94664  1.00000  0.778135 -0.076088 -0.07107
## PIQ     0.93413  0.77814  1.000000  0.002512 -0.07672
## Weight -0.05148 -0.07609  0.002512  1.000000  0.69961
## Height -0.08600 -0.07107 -0.076723  0.699614  1.00000
cor.test(brain$PIQ,
         brain$MRINACount,
         method = "pearson",
         alternative = "two.sided")
##
##  Pearson's product-moment correlation
##
## data:  brain$PIQ and brain$MRINACount
## t = 2.6, df = 38, p-value = 0.01
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.08563 0.62323
## sample estimates:
##    cor
## 0.3868
```

Решение

```
pl_brain <- ggplot(brain, aes(x = MRINACount, y = PIQ)) +
geom_point() + xlab("Brain size") + ylab("IQ test")
pl_brain
```



Частные корреляции

Частная корреляция – функция, определяющая связь между двумя переменными при условии, что влияние других переменных удалено.

Мы удаляем из X и Y ту часть зависимости, которая вызвана влиянием Z.

```
library(ppcor)
## Warning: package 'ppcor' was built under R version 3.3.3
brain_complete <- brain[complete.cases(brain),]
pcor.test(brain_complete$PIQ,         brain_complete$MRINACount,
brain_complete$Height, )
##   estimate   p.value statistic  n gp  Method
## 1   0.5373 0.0006051     3.769 38  1 pearson
```

Проведя корреляционный анализ, мы лишь ответили на вопрос: «Существет ли достоверная связь между величинами?».

Сможем ли мы, используя это знание, *предсказть* значения одной величины, исходя из знаний другой?

Простейший пример: между путем, пройденным автомобилем, и временем, проведенным в движении, несомнено есть связь. Хватает ли нам этого знания?

Для расчета величины пути в зависимости от времени необходимо построить модель: $S = Vt$, где S – зависимая величина, t – независимая переменная, V – параметр модели.

Зная параметр модели (скорость) и значение независимой переменной (время), мы можем рассчитать (*смоделировать*) величину пройденного пути.

Какие бывают модели?

Линейные и нелинейные модели

Линейные модели

$$y = b_0 + b_1 x$$

$$y = b_0 + b_1 x_1 + b_2 x_2$$

Нелинейные модели

$$y = b_0 + b_1^x$$

$$y = b_0^{b_1 x_1 + b_2 x_2}$$

Простые и многокомпонентные (множественные) модели

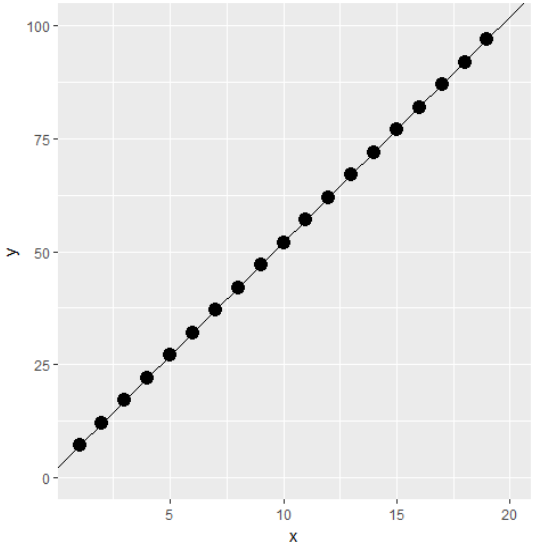
Простая модель

$$y = b_0 + b_1 x$$

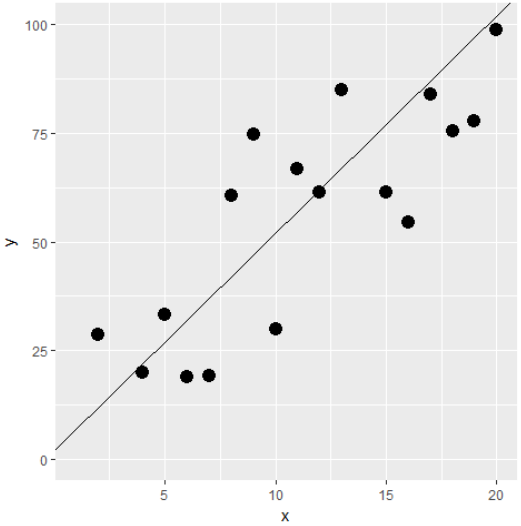
Множественная модель

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots + b_n x_n$$

Детерминистские и стохастические модели

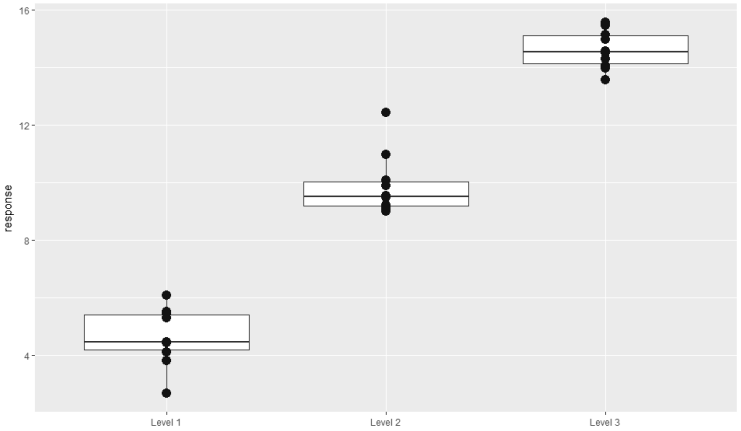


Модель $y_i = 2 + 5x_i$ Два парметра: угловой коэффициент (slope) $b_1 = 5$; свободный член (intercept) $b_0 = 2$ Чему равен y при $x = 10$?



Модель $y = 2 + 5x + \epsilon$ Появляется дополнительный член ϵ_i Он вводит в модель влияние неучтенных моделью факторов. Обычно считают, что $\epsilon \in N(0, \sigma^2)$.

Модели с дискретными предикторами

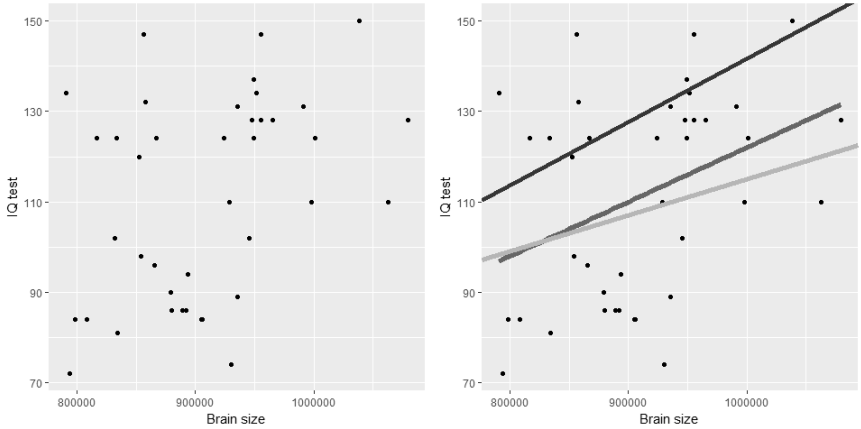


Модель для данного примера имеет такой вид $response = 4.6 + +5.3I_{Level2} + 9.9I_{Level3}$.

I_i – dummy variable.

Модель для зависимости величины IQ от размера головного мозга

Какая из линий «лучше» описывает облако точек?



"Essentially, all models are wrong, but some are useful" (Georg E. P. Box)

Найти оптимальную модель позволяет регрессионный анализ
Подбор лиии регрессии проводится с помощью двух методов:

• С помощью метода наименьших квадратов (Ordinary Least Squares) – используется для простых линейных моделей

• Через подбор функции максимального правдоподобия (Maximum Likelihood) – используется для подгонки сложных линейных и нелинейных моделей.

Подбор модели методом наменьших квадратов с помощью функци lm()

fit <- lm(formula, data)

Модель записывается в виде формулы:

Модель	Формула
Простая линейная регресся $\hat{y}_i = b_0 + b_1 x_i$	Y ~ X Y ~ 1 + X Y ~ X + 1
Простая линейная регрессия (без b_0, "no intercept") $\hat{y}_i = b_1 x_i$	Y ~ -1 + X Y ~ X - 1
Уменьшенная простая линейная регрессия $\hat{y}_i = b_0$	Y ~ 1 Y ~ 1 - X
Множественная линейная регрессия $\hat{y}_i = b_0 + b_1 x_i + b_2 x_2$	Y ~ X1 + X2

Элементы формул для записи множественных моделей:

Элемент формулы	Значение
:	Взаимодействие предикторов Y ~ X1 + X2 + X1:X2
*	Обзначает полную схему взаимодействий Y ~ X1 * X2 * X3 аналогично Y ~ X1 + X2 + X3+ X1:X2 + X1:X3 + X2:X3 + X1:X2:X3
.	Y ~ . В правой части формулы записываются все переменные из датафрейма, кроме Y

Подберем модель, наилучшим образом описывающую зависимость результатов IQ-теста от размера головного мозга:

```
brain_model <- lm(PIQ ~ MRINACount, data = brain)
brain_model
##
## Call:
## lm(formula = PIQ ~ MRINACount, data = brain)
##
## Coefficients:
## (Intercept)   MRINACount
##     1.74376      0.00012
```

Как трактовать значения параметров регрессионной модели?

• Угловой коэффициент (*slope*) показывает на сколько *единиц* изменяется предсказанное значение $\hat{y}$ при изменении на *одну единицу* значения предиктора (x).

• Свободный член (*intercept*) – величина во многих случаях не имеющая «смысла». Это поправочный коэффициент, без которого нельзя вычислить $\hat{y}$. *NB!* В некоторых линейных моделях он имеет смысл, например, значения $\hat{y}$ при $x = 0$.

• Остатки (*residuals*) – характеризуют влияние неучтенных моделью факторов.

Вопросы:

1. Чему равны угловой коэффициент и свободный член полученной модели brain_model?

2. Какое значеие IQ-теста предсказывает модель для человека с объемом мозга равным 900000?

3. Чему равно значение остатка от модели для человека с порядковым номером 10?

Ответы

```
coefficients(brain_model) [1]
## (Intercept)
##       1.744
coefficients(brain_model) [2]
## MRINACount
##  0.0001203
coefficients(brain_model) [1] + coefficients(brain_model) [2] * 900000
## (Intercept)
##         110
brain$PIQ[10] - fitted(brain_model)[10]
##    10
## 30.36
residuals(brain_model)[10]
##    10
## 30.36
```

Углубляемся в анализ модели: функция summary()

```
summary(brain_model)
##
## Call:
## lm(formula = PIQ ~ MRINACount, data = brain)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -39.6  -17.9   -1.6   17.0   42.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.7437570 42.3923825    0.04    0.967
## MRINACount   0.0001203  0.0000465    2.59    0.014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21 on 38 degrees of freedom
## Multiple R-squared:  0.15,   Adjusted R-squared:  0.127
## F-statistic: 6.69 on 1 and 38 DF,  p-value: 0.0137
```

Что означают следующие величины?

Estimate
Std. Error
t value
Pr(>|t|)

Оценки параметров регрессионной модели

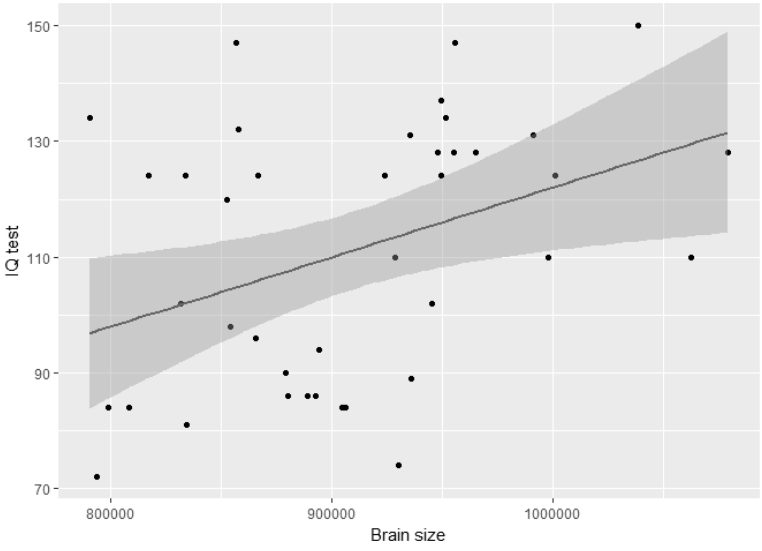
Параметр	Оценка	Стандартная ошибка
β_1	$b_1 = \frac{\sum_{i=1}^{n}[(x_i-\overline{x})(y_i-\overline{y})]}{\sum_{i=1}^{n}(x_i-\overline{x})^2}$ или проще $b_0 = r\frac{sd_y}{sd_x}$	$SE_{b_1} = \sqrt{\frac{MS_e}{\sum_{i=1}^{n}(x_i-\overline{x})^2}}$
β_0	$b_0 = \overline{y} - b_1\overline{x}$	$SE_{b_0} = \sqrt{MS_e[\frac{1}{n} + \frac{\overline{x}}{\sum_{i=1}^{n}(x_i-\overline{x})^2}]}$
ϵ_i	$e_i = y_i - \hat{y}_i$	$\approx \sqrt{MS_e}$

Для чего нужны стандартные ошибки?

- Они нужны, поскольку мы *оцениваем* параметры по *выборке*.
- Они позволяют построить доверительные интервалы для параметров.
- Их используют в статистических тестах.

Графическое представление результатов

```
pl_brain + geom_smooth(method="lm")
```



Что это за серая область?

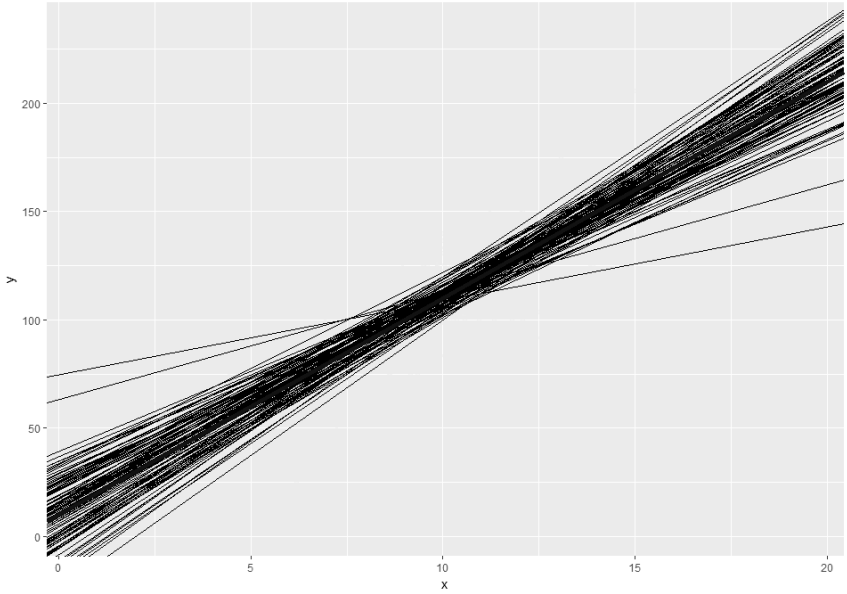
• Это *95 % доверительная зона регрессии.*

• В ней с 95 % вероятностью лежит регрессионная прямая, описывающая связь в генеральной совокупности.

• Возникает из-за неопределенности оценок коэффициентов регрессии, вследствие выборочного характера оценок.

Симулированный пример

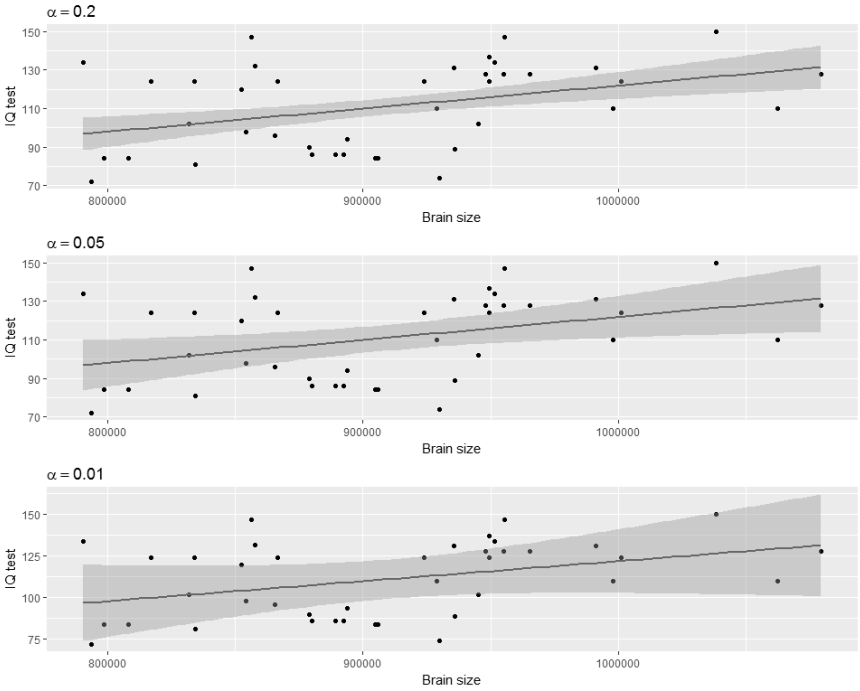
Линии регрессии, полученные для 100 выборок (по 20 объектов в каждой), взятых из одной и той же генеральной совокупности, изображенных ниже.



Доверительные интервалы для коэффициентов уравнения регрессии

```
coef(brain_model)
## (Intercept)  MRINACount
##   1.7437570   0.0001203
confint(brain_model)
##                     2.5 %     97.5 %
## (Intercept) -84.07513478 87.5626489
## MRINACount    0.00002611  0.0002144
```

Для разных α можно построить разные доверительные интервалы.



Важно! Если коэффициенты уравнения регресси – лишь приблизительные оценки параметров, то предсказать значения зависимой переменной можно только *с некоторой вероятностью*.

Какое значение IQ можно ожидать у человека с размером головного мозга 900000?

```
newdata <- data.frame(MRINACount = 900000)

predict(brain_model, newdata, interval = "confidence", level =
0.95, se = TRUE)
## $fit
##   fit   lwr   upr
## 1 110 103.2 116.7
##
## $se.fit
## [1] 3.344
##
## $df
## [1] 38
##
```

```
## $residual.scale
## [1] 20.99
```

При размере мозга 900000 среднее значение IQ будет, с вероятностью 95 %, находиться в интервале от 103 до 117 (110 ± 7).

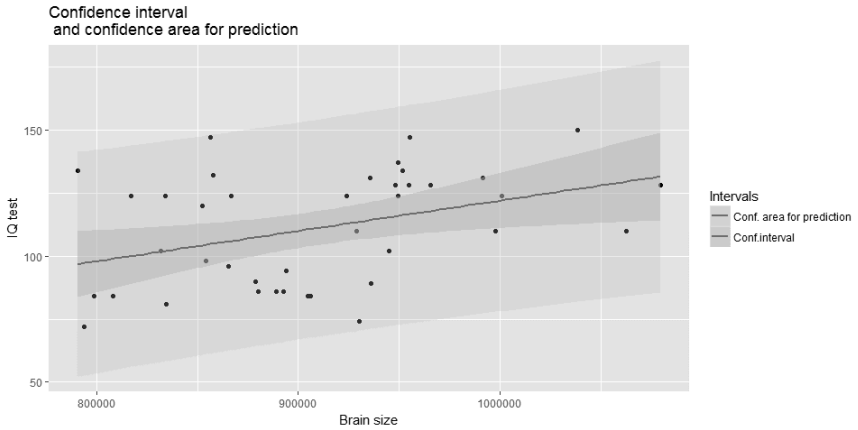
Отражаем на графике область значений, в которую попадут 95 % предсказанных величин IQ

Подготавливаем данные

```
brain_predicted <- predict(brain_model, interval="prediction")
brain_predicted <- data.frame(brain, brain_predicted)
head(brain_predicted)
##   Gender FSIQ VIQ PIQ Weight Height MRINACount   fit  lwr  upr
## 1 Female  133 132 124   118   64.5    816932  99.98 56.10 143.9
## 2   Male  140 150 124    NA   72.5   1001121 122.13 78.24 166.0
## 3   Male  139 123 150   143   73.3   1038437 126.62 81.90 171.3
## 4   Male  133 129 128   172   68.8    965353 117.83 74.48 161.2
## 5 Female  137 132 134   147   65.0    951545 116.17 72.96 159.4
## 6 Female   99  90 110   146   69.0    928799 113.44 70.37 156.5
```

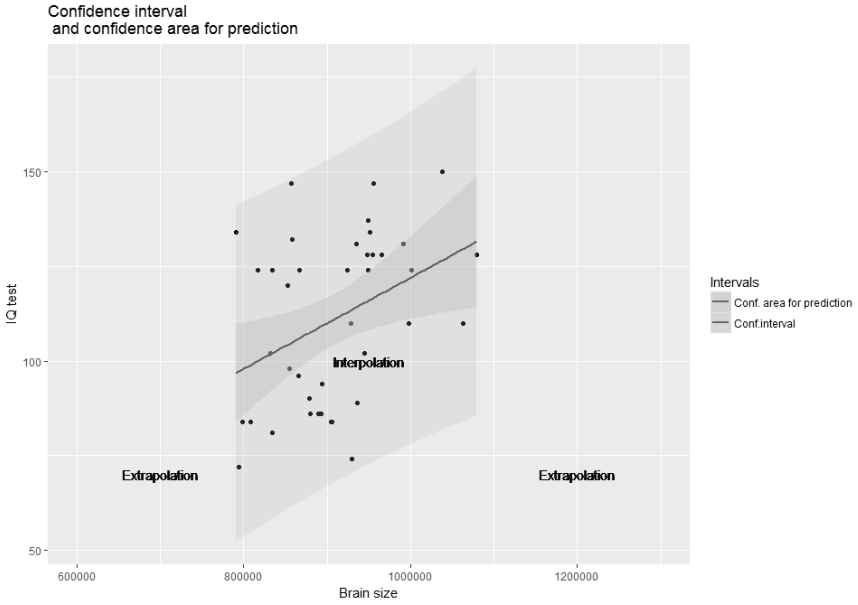
Отражаем на графике область значений, в которую попадут 95 % предсказанных величин IQ

```
pl_brain +
  geom_ribbon(data=brain_predicted, aes(y=fit, ymin=lwr, ymax=upr,
fill = "Conf. area for prediction"), alpha=0.2) +
  geom_smooth(method="lm", aes(fill="Conf.interval"), alpha=0.4) +
  scale_fill_manual("Intervals", values = c("green", "gray")) +
  ggtitle("Confidence interval \n and confidence area for predicti
on")
## Warning: Ignoring unknown aesthetics: y
```



Важно!

Модель «работает» только в том диапазоне значений независимой переменной (x), для которой она построена (интерполяция). Экстраполяцию надо применять с большой осторожностью.



Итак, что означают следующие величины?

`Estimate`
Оценки праметров регрессионной модели

`Std. Error`
Стандартная ошибка для оценок

Осталось решить, что такое `t value`, `Pr(>|t|)`

Проверка состоятельности модели

Существует два равноправных способа:

1) Проверка достоверности оценок коэффициента b_1 (t-критерий).

2) Оценка соотношения описанной и остаточной дисперсии (F-критерий).

Проверка состоятельности модели с помощью t-критерия

Модель «работает» если в генеральной совокупности $\beta_1 \neq 0$

Гипотеза: $H: \beta \neq 0$; антигипотеза $H_0: \beta = 0$. Тестируем гипотезу

$$t = \frac{b_1 - 0}{SE_{b_1}}.$$

Число степеней свободы: $df = n - 2$ >- Итак, >- `t value` – Значение t-критерия >- `Pr(>|t|)` – Уровень значимости.

Состоятельна ли модель, описывающая связь IQ и размера головного мозга?

$$PIQ = 1.744 + 0.0001202 MRINACount$$

```
summary(brain_model)
##
## Call:
## lm(formula = PIQ ~ MRINACount, data = brain)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -39.6  -17.9   -1.6   17.0   42.3
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.7437570 42.3923825    0.04    0.967
## MRINACount   0.0001203  0.0000465    2.59    0.014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21 on 38 degrees of freedom
## Multiple R-squared:  0.15,   Adjusted R-squared:  0.127
## F-statistic: 6.69 on 1 and 38 DF,  p-value: 0.0137
```

Проверка состоятельности модели с помощью F-критерия

Объясненная дисперсия зависимой перменной:

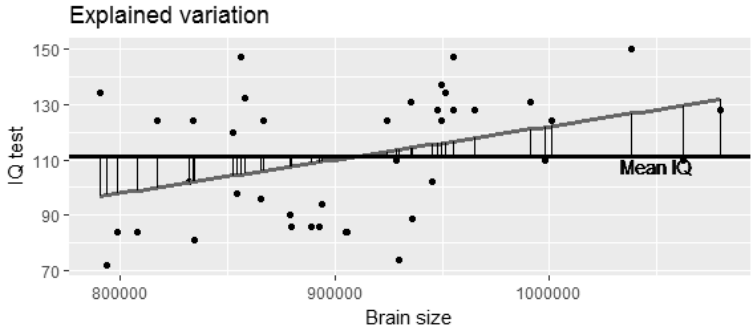
$$SS_{Regression} = \sum(\hat{y} - \overline{y})^2$$
$$df_{Regression} = 1$$
$$MS_{Regression} = \frac{SS_{Regression}}{df}$$

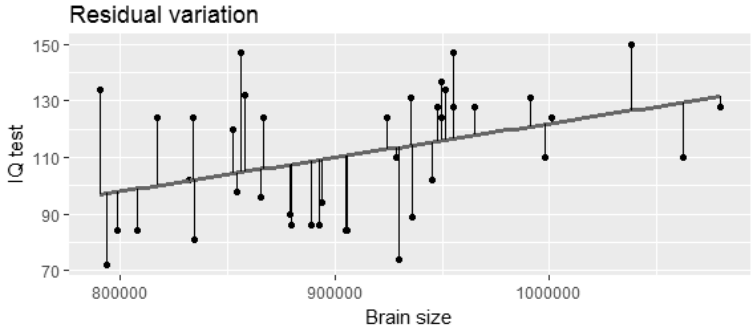
Остаточная дисперсия завсимой переменной:

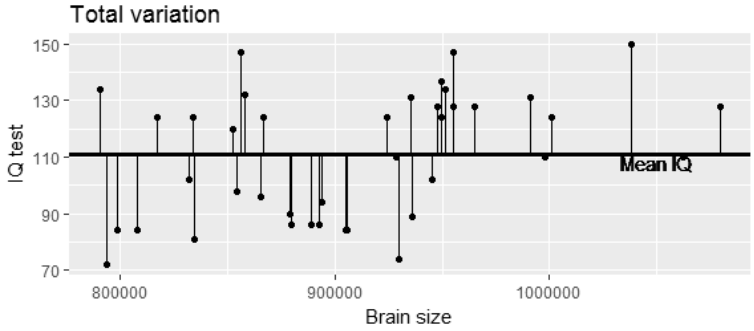
$$SS_{Residual} = \sum(\hat{y} - y_i)^2$$
$$df_{Residual} = n - 2$$
$$MS_{Residual} = \frac{SS_{Residual}}{df_{Residual}}$$

Полная дисперсия зависимой переменной:

$$SS_{Total} = \sum(\overline{y} - y_i)^2$$
$$df_{Total} = n - 1$$
$$MS_{Total} = \frac{SS_{Total}}{df_{Total}}$$





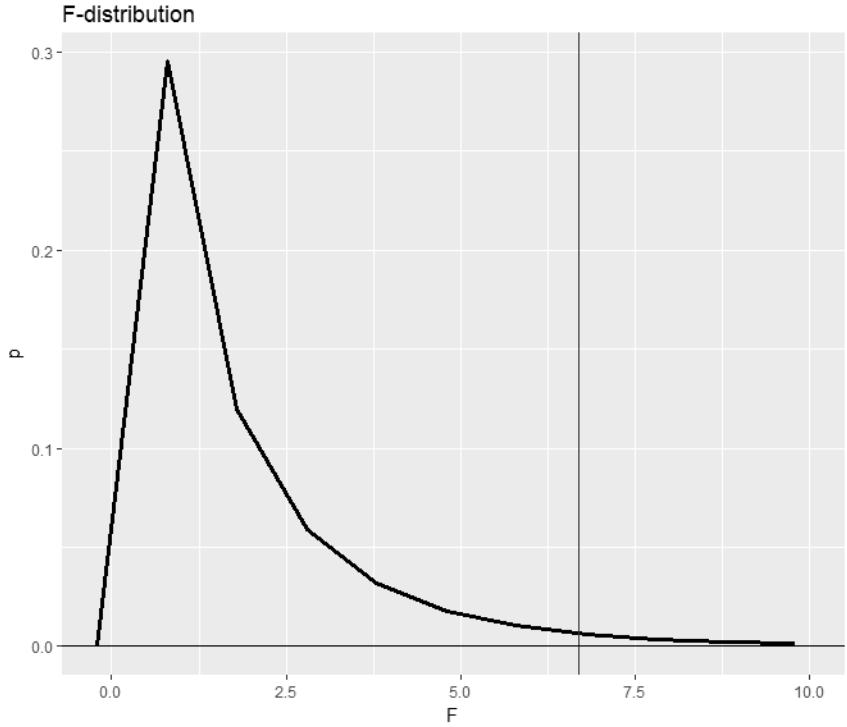


F критерий

Если зависимости нет, то $MS_{Regression} = MS_{Residual}$

$$F = \frac{MS_{Regression}}{MS_{Residual}}.$$

Логика та же, что и с t-критерием

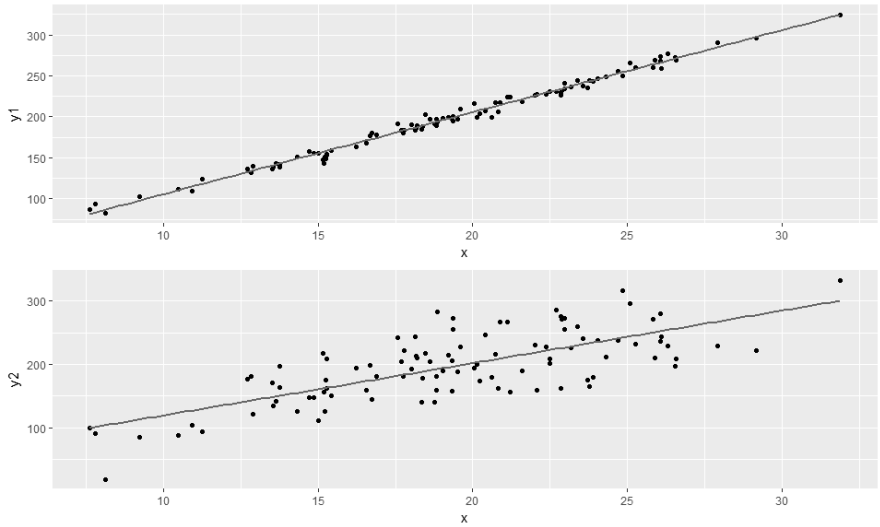


Форма F-распределения зависит от двух параметров:

$df_{Regression} = 1$ и $df_{Residual} = n - 2$.

Оценка качества подгонки модели с помощью коэффициента детерминации

В чем различие между этми двумя моделями?



Оценка качества подгонки модели с помощью коэффициента детерминации

Какую долю дисперсии зависимой переменной описывает коэффициент детерминации, объясняет модель:

$$R^2 = \frac{SS_{Regression}}{SS_{Total}}$$

$$0 < R^2 < 1$$

$$R^2 = r^2$$

Еще раз смотрим на результаты регрессионного анализа зависимости IQ от размеров мозга

```
summary(brain_model)
##
## Call:
## lm(formula = PIQ ~ MRINACount, data = brain)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -39.6  -17.9   -1.6   17.0   42.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  1.7437570 42.3923825    0.04    0.967
## MRINACount   0.0001203  0.0000465    2.59    0.014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21 on 38 degrees of freedom
## Multiple R-squared:  0.15,   Adjusted R-squared:  0.127
## F-statistic: 6.69 on 1 and 38 DF,  p-value: 0.0137
```

Adjusted R-squared – скорректированный коэффициет детерминации

Применяется если необходимо сравнить две модели с разным количеством параметров:

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-k},$$

где k – количество параметров в модели.

Вводится штраф за каждый новый параметр.

Как записываются результаты регрессионного анлиза в тексте статьи?

Мы показали, что связь между результатами теста на IQ описывается мделью вида IQ = 1.74 + 0.00012 MRINACount ($F_{1,38}$ = 6.686, p = 0.0136, R^2 = 0.149)

Summary

Модель простой линейной регрессии $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$.

Параметры модели оцениваются на основе выборки.

В оценке коэффициентов регрессии и предсказанных значений существует неопределенность: необходимо вычислять доверительный интервал.

Доверительные интервалы можно расчитать, зная стандартные ошибки. Состоятельность модели можно проверить при помощи t- или F-теста. ($H_0: \beta_1 = 0$).

Качество подгонки модели можно оценить при помощи коэффициента детерминации (R^2).

Далее рассмотрим:

- Методы диагностики линейных моделей.
- Средства R, позволяющие провести такую диагностику.

Вы сможете

Использовать диагностические средства, реализованные в среде R, для проверки соблюдения условий применимости линейных моделей.

Первичное исследование данных

Важнейший этап работы с моделями – это предварительное исследование данных (Data exploration)

Этот этап должен предшествовать построению модели.

Самый правильный путь – это построить графики рассеяния, отражающие взаимосвязи между переменными.

Исследование данных позволяет избежать неправомерного применения метода. Например, один и тот же результат можно получить для совершенно разных данных.

После того как данные исследованы, можно подобрать модель и оценить ее состоятельность.

Но! Если модель состоятельна и $\boldsymbol{H_0}$ отвергнута, то не впадайте в эйфорию!

Надо еще проверить валидна ли ваша модель!

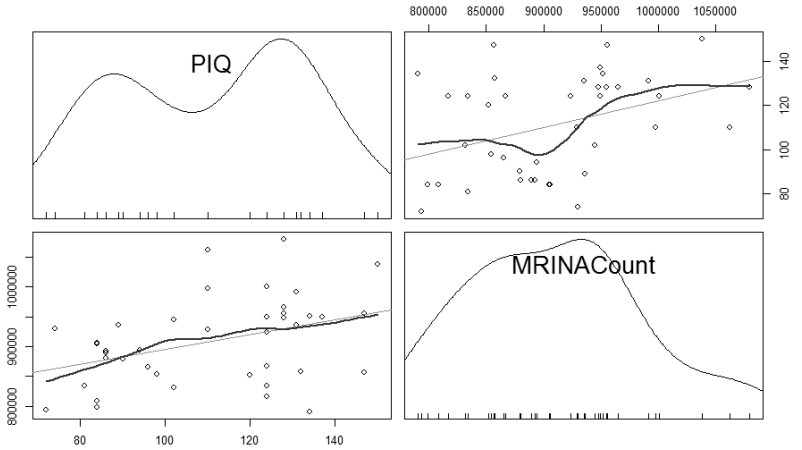
Проверка валидности модели

Певый этап анализа валидности модели – это проверка на наличие влиятельных наблюдений

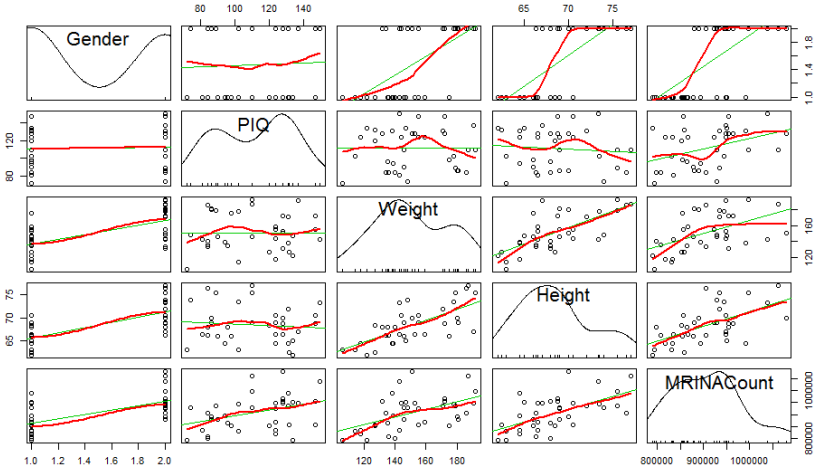
Влиятельные наблюдения – наблюдения, вносящие неправомерно большой вклад в оценку парметров (коэффициентов) модели.

Вам понадобятся следующие пакеты: ggplot2, car, lmtest, gvlma, lmodel2. Еще раз читаем данные про зависимость IQ от размера головного мозга:

```
brain <- read.csv("IQ_brain.csv", header=TRUE) brain <-
brain[, -c(2,3)] # Еще раз строим модель
brain_model <- lm(PIQ ~ MRINACount, data = brain)
library(car)
## Warning: package 'car' was built under R version 3.3.3
# Проводим data exploration
scatterplotMatrix(brain[, c("PIQ", "MRINACount")],
spread=FALSE)
```




```
r scatterplotMatrix(brain, spread=FALSE)
```



Сначала научимся извлекать из результатов необходимые сведения: Для этого служит функция `fortify()` из пакета `{ggplot2}`.

```
    require(ggplot2) brain_diag <- fortify(brain_model)
    head(brain_diag, 2)
##    PIQ MRINACount     .hat .sigma    .cooksd .fitted .resid
.stdresid ## 1 124     816932 0.06638  20.88 0.0498380   99.98
```

```
24.017    1.1840 ## 2 124    1001121 0.06687  21.27 0.0003039
122.13  1.868    0.0921
```

Уже знакомые:

`.fitted` – предсказанные значения,

`.resid` – остатки.

Новые величины, которые нам понадобятся:

`.hat` – «воздействие» данного наблюдения (*leverage*),

`.cooksd` – расстояние Кука,

`.stdresid` – стандартизованные остатки.

Типы остатков

Просто остатки:

$$e_i = y_i - \hat{y}_i$$

Стандартизованные остатки:

$$\frac{e_i}{\sqrt{MS_{Residual}}}$$

Стьюдентовские остатки:

$$\frac{e_i}{\sqrt{MS_{Residual}(1-h_i)}}$$

Последние удобнее, так как можно сказать какие остатки большие, а какие маленькие при сравнении разных моделей

Воздействие точек (Leverage)

Эта велична, показывает насколько каждое значение x_i влияет на ход линии регрессии, то есть на $\hat{y}_i$.

Точки, располагающиеся дальше от $\overline{x}$, оказывают более сильное влияние на $\hat{y}_i$.

Эта величина, в норме, варьирует в промежутке от $1/n$ до 1.

Если $h_i > 2(p/n)$, то надо внимательно посмотреть на данное значение.

Удобнее другая величина – расстояние Кука.

Расстояние Кука (Cook's distance)

Описывает, как повлияет на модель удаление данного наблюдения:

$$D_i = \frac{\sum(\hat{y}_j - \hat{y}_{j(i)})^2}{pMSE},$$

где $\hat{y}_j$ – значение предсказанное полной моделью,

$\hat{y}_{j(i)}$ – значение, предказанное моделью, построенной без учета i-го значения предиктора,

p – количество параметров в модели,

MSE – среднеквадратичная ошибка модели.

Расстояние Кука одновременно учитывает величину остатков по всей модели и влиятельность (leverage) отдельных точек.

Распределение статистики D_i близко к F-распределению. Если расстояние Кука $D_i > 1$, то данное наблюдение можно рассматривать как выброс (outlier). Для более жесткого выделения выбросов используют пороговую величину, зависящую от объема выборки:

$$D_i > 4/(N - k - 1),$$

N – объем выборки, k – число предикторов.

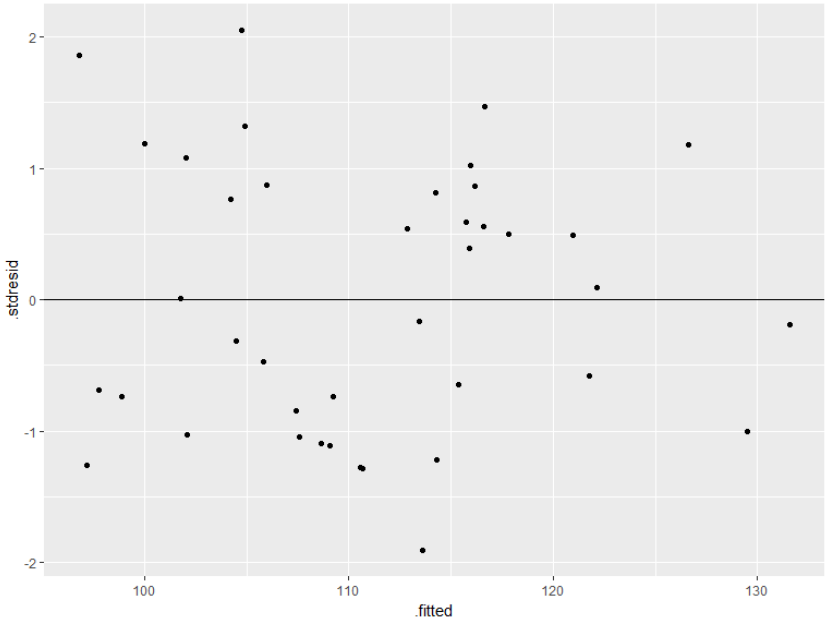
Задание

Для модели `brain_model` постройте график рссеяния стандартизированных остатков в зависимости от предсказанных значений

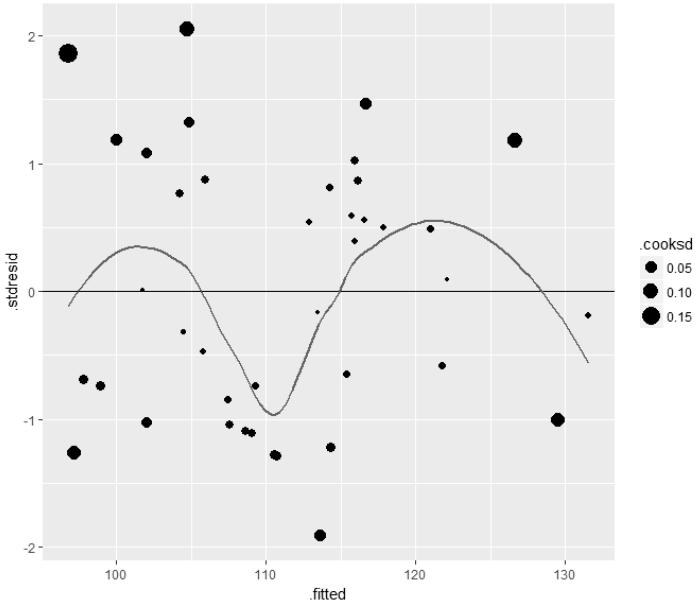
Hint: вспомните, что мы уже получили датафрейм `brain_diag`

Решение

```
ggplot(data = brain_diag, aes(x = .fitted, y = .stdresid)) +
  geom_point() + geom_hline(yintercept=0)
```



Внесем некоторые дополнения



Что мы видим?

- Большая часть стандартизованных остатков в пределах двух стандартных отклонений.
- Есть одно влиятельное наблюдение, которое нужно проверить, но сила его влияния невелика.
- Среди остатков нет тренда.
- Но есть иной паттерн!

Что делать с влиятельными наблюдениями?

Метод 1. Удаление влиятельных наблюдений

Будьте осторожны! Отскакивающие значения могут иметь важное значение. Удалять следует только очевидные ошибки в наблюдениях.

После их удаления необходимо пересчитать модель.

Метод 2. Преобразование переменных

Наиболее частые преобразования, используемые для построения линейных моделей:

Трансформация	Формула
-2	$1/Y^2$
-1	$1/Y$
-0.5	$1/\sqrt{(Y)}$
логарифмирование	$log(Y)$
0.5	$\sqrt{(Y)}$
2	Y^2
логит	$ln(\frac{Y}{1-Y})$

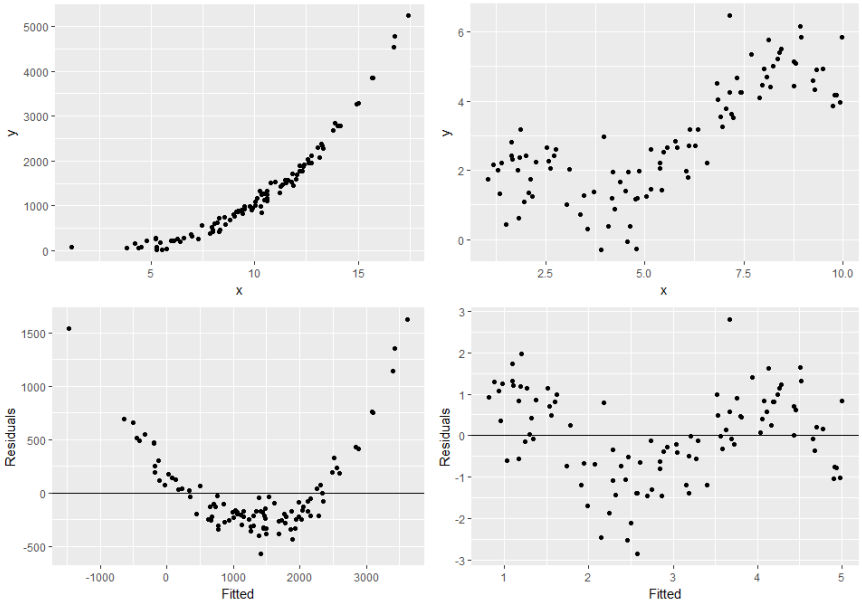
Допущения метода линейных моделей (Assumptions)

Условия применимости линейных моделей:

1. Линейность связи между зависимой перменной (Y) и предикторами (X).
2. Независимость Y друг от друга.
3. Нормальное распределение Y для каждого уровня значенй X.
4. Гомогенность дисперсии Y в пределах всех уровней значений X.
5. Фиксированные значения X.
6. Отсутствие коллинеарности предикторов (для можественной регрессии).

Допущение 1. Линейность связи

Нелинейные зависимости не всегда видны на исходных графиках в осях Y vs X. Они становятся лучше заметны на графиках рассеяния остатков (Residual plots)

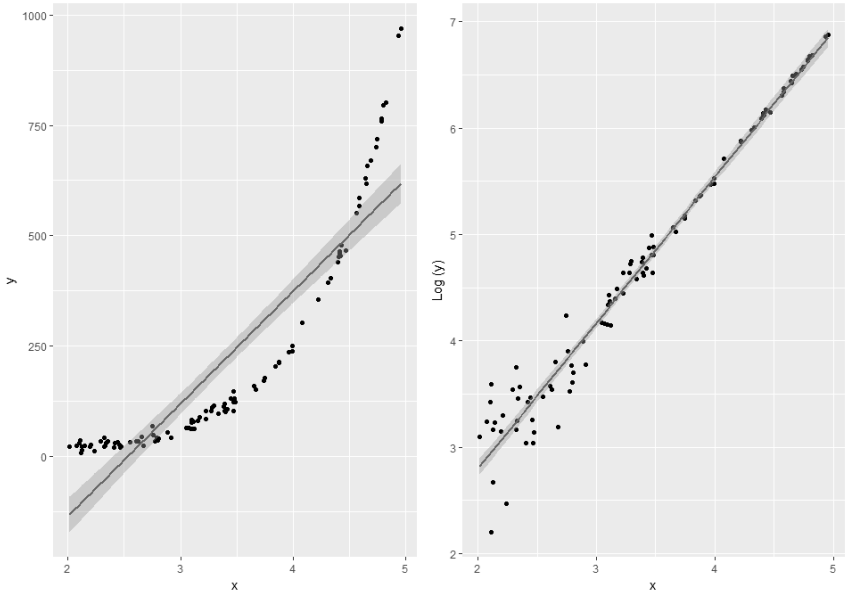


Что делать, если связь нелинейна?

• Можно применить линеаризующее преобразование.

• Можно построить нелинейную модель (об этом будем говорить отдельно).

Пример линеаризующего преобразования:



Допущение 2. Независимость *Y* друг от друга

– Каждое значение Y_i должно быть независимо от любого Y_j.

– Это должно контролироваться на этапе планирования сбора материала.

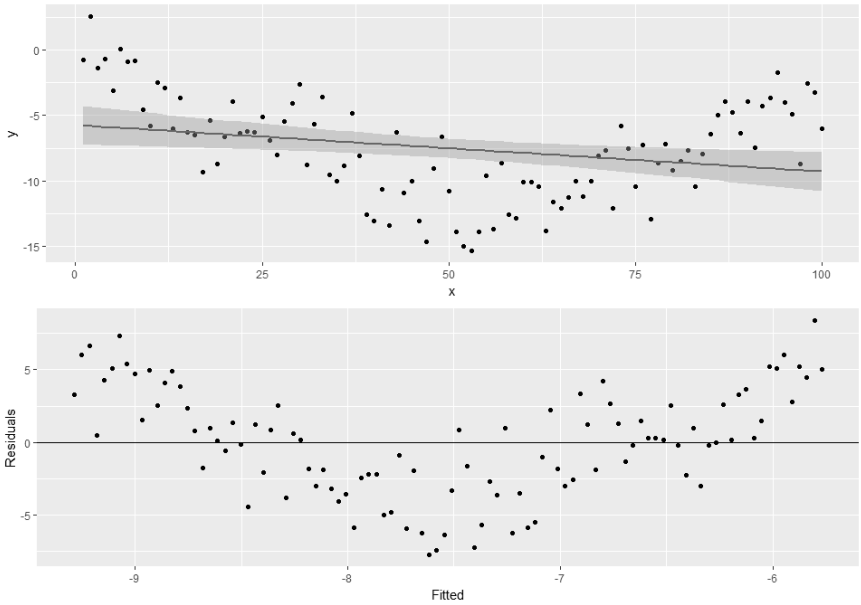
Наиболее частые источники зависимостей:

– псевдоповторности,

– временные и пространственые автокорреляции,

– взаимозависимости могут проявляться на графиках рассеяния остатков (Residual plots).

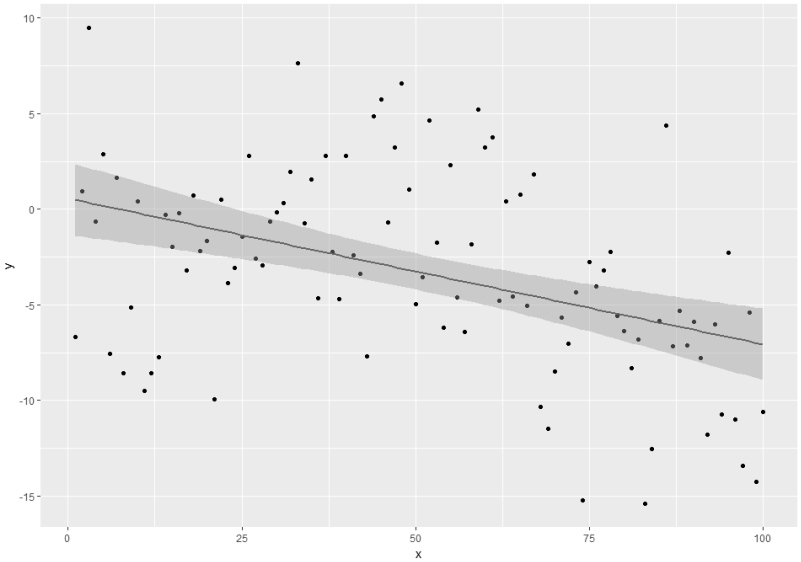
Симулированный пример: автокоррелированные данные:



Критерий Дарбина–Уотсона: Формальный тест на автокорреляцию

```
brain_model <- lm(PIQ ~ MRINACount, data = brain)
durbinWatsonTest(brain_model)
##  lag Autocorrelation D-W Statistic p-value
##    1          0.2288          1.47     0.1
##  Alternative hypothesis: rho != 0
```

Пример данных, которые имеют ярко выраженную автокорреляцию:




```
lm_autocor <- lm(y ~ x, data = dat)
durbinWatsonTest(lm_autocor)
##  lag Autocorrelation D-W Statistic p-value
##    1          0.2111         1.549    0.01
##  Alternative hypothesis: rho != 0
```

Допущение 3. Нормальное распределение *Y* для каждого уровня значений *X*

– $Y_i \sim N(\mu_{y_i}, \sigma^2)$

– В идеале, каждому X_i должно соответствовать большое количество наблюдений Y.

– На практике такое бывает только в случае моделей с дискретными предикторами.

– Соответствие нормальности распределения можно оценить по «поведению» случайной части модели.

Фиксированная и случайная часть модели

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

Фиксированная часть: $y_i = \beta_0 + \beta_1 x_i$ задает жесткую связь между x_i и y_i.

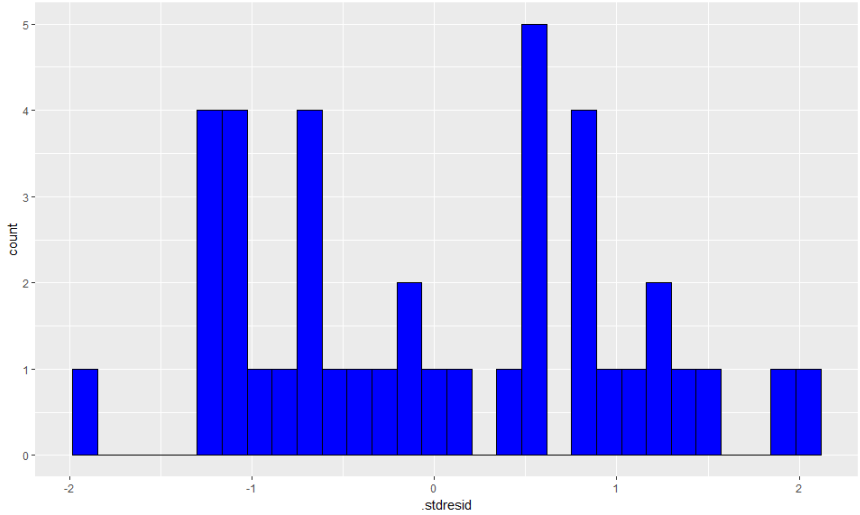
Случайная часть – ϵ_i.

Так как $y_i - \beta_0 - \beta_1 x_i = 0$, то $\epsilon_i \sim N(0, \sigma^2)$.

Если с моделью все в порядке, то условие нормального распределения остатков должно соблюдаться.

Можно построить частотное распределение остатков:

```
ggplot(brain_model, aes(x=.stdresid)) + geom_histo-
gram(bin=0.4, fill="blue", color="black")
## Warning: Ignoring unknown parameters: bin
```



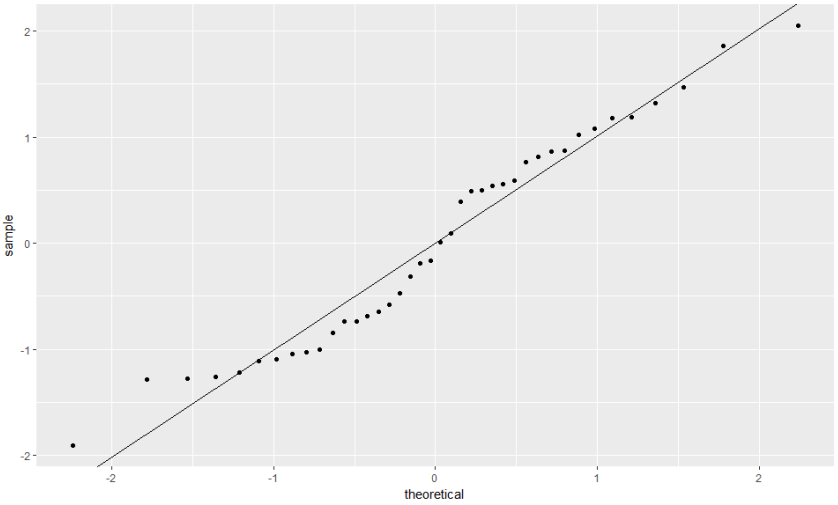
На частотной гистограмме остатков не всегда хорошо видны отклонения от нормальности.

Проверка нормальности распределения остатков с помощью нормальновероятностного графика стандартизованных остатков

Квантиль – значение, которое заданная случайная величина не превышает с фиксированной вероятностью.

Если точки – это случайные величины из $N(0,\sigma^2)$, то они должны лечь вдоль прямой $Y = X$.

```
mean_val <- mean(brain_diag$.stdresid)
sd_val <- sd(brain_diag$.stdresid)
ggplot(brain_diag, aes(sample = .stdresid)) + geom_point(stat =
="qq") + geom_abline(intercept = mean_val, slope = sd_val)
```



Мы видим, что отклонения от нормальности есть! Но! Метод устойчив к небольшим отклонениям от нормальности.

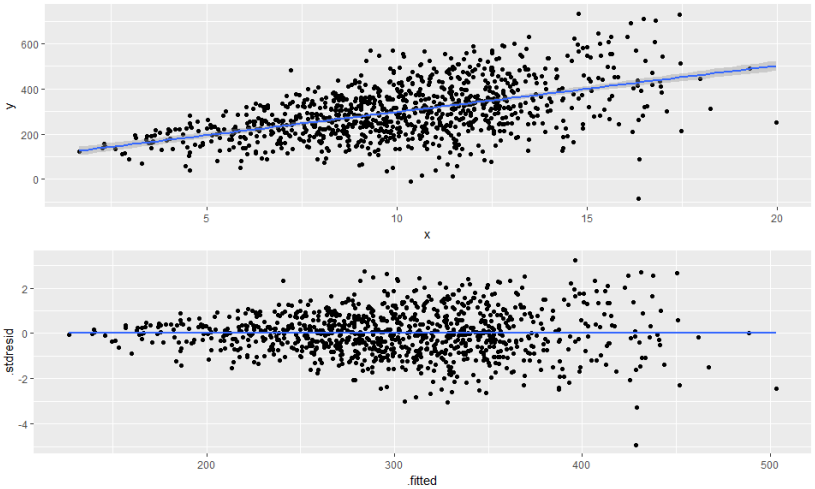
Допущение 4. Постоянство дисперсии – гомоскедастичность

Это самое важное ограничивающее условие!

Оценки коэффициентов уравнения регрессии при гетероскедастичности не смещаются, но сильно смещаются стандартные ошибки.

Многие тесты чувствительны к гетероскедастичности.

Если линия регрессии подобрана для гетероскедастичных данных, то повышается вероятность ошибки II рода.



Формальные тесты на гетероскедастичность

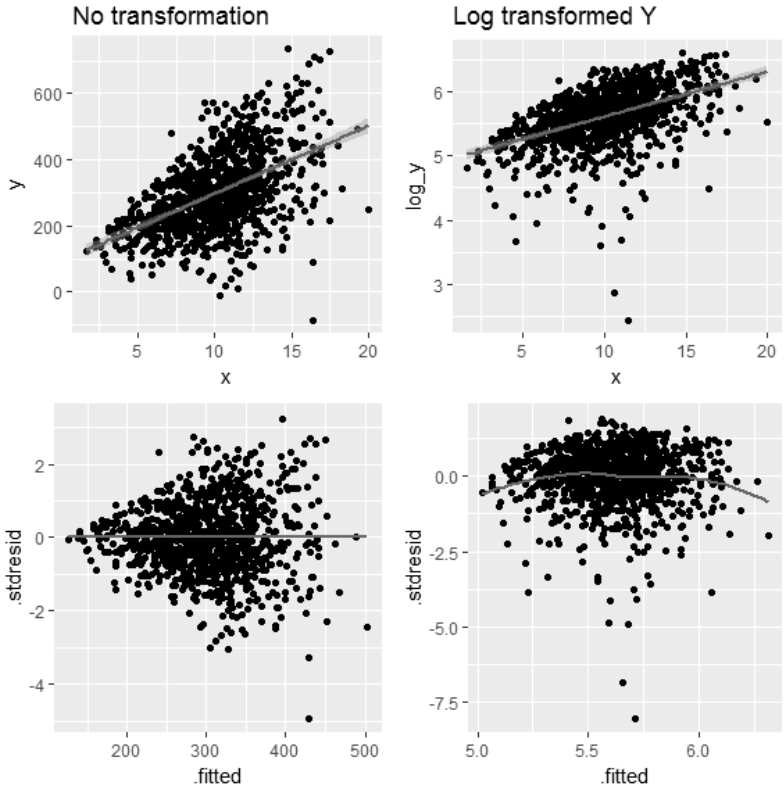
Для линейных моделей с непрерывным предиктором применяется, например, тест Бройша-Пагана (Breusch-Pagan test).

Для моделей с дискретными предикторами чаще применяют тест Кокрана (Cochran test).

```
library(lmtest)
## Warning: package 'lmtest' was built under R version 3.3.3
## Warning: package 'zoo' was built under R version 3.3.3
#Симулированные данные
bptest(y ~ x, data = dat)
##
##  studentized Breusch-Pagan test
##
## data:  y ~ x
## BP = 92, df = 1, p-value <2e-16
#Реальные данные
bptest(PIQ ~ MRINACount, data = brain)
##
##  studentized Breusch-Pagan test
##
## data:  PIQ ~ MRINACount
## BP = 2.7, df = 1, p-value = 0.1
```

Что делать если вы столкнулись с гетероскедастичностью?

Решение 1. Применить преобразование зависимой переменной (в некоторых случаях и предиктора).



Недостатки:

1. Не всегда спасает.

2. Модель описывает поведение не исходной, а преобразованной величины. *«Если вы не можете доказать А, докажите В и сделайте вид, что это было А»* (Кобаков, 2014)

Решение 2. Построить более сложную модель, которая учитывала бы гетерогенность дисперсии зависимой перменной.

Можно автоматически проверить допущения с помощью средств R

```
library(gvlma)
gvlma(brain_model)
##
## Call:
## lm(formula = PIQ ~ MRINACount, data = brain)
##
## Coefficients:
```

```
## (Intercept)   MRINACount
##     1.74376      0.00012
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = brain_model)
##
##                     Value p-value                Decision
## Global Stat        2.459   0.652 Assumptions acceptable.
## Skewness           0.121   0.728 Assumptions acceptable.
## Kurtosis           1.849   0.174 Assumptions acceptable.
## Link Function      0.100   0.751 Assumptions acceptable.
## Heteroscedasticity 0.388   0.533 Assumptions acceptable.
```

Допущение 5. Фиксированные значения предикторов

Мы рассматривали регрессионную модель I (Model I regression). Она требует, чтобы при каждом новом исследовании уровни предиктора x_i были бы теми же самыми.

В биологии это условие обычно не соблюдается, так как мы работаем с материалом, в котором выборочными значениями являются как Y, так и X.

То есть, x_i являются случайными величинами. Они не контролируются исследователем.

Если наша цель – построить модель, которая будет предсказывать значения Y в зависимсоти от X, то можно использовать регрессионную модель I, но оценка β_1 будет немного занижена. Модель будет недопредсказывать.

Если наша цель – выявление истинной взаимосвязи между Y и X (то есть важна оценка β_1) придется учитывать, что есть не только ϵ_y, но также и δ_x

В этой ситуации используют иной способ подбора линии регрессии

Подбор линии регрессии
для регрессионной модели II (Model II regression)

```
library(lmodel2)
brain_lm2 <- lmodel2(PIQ ~ MRINACount, data = brain, range.y="rel-
ative", range.x="relative", nperm=99)
brain_lm2
```

```
##
## Model II regression
##
## Call: lmodel2(formula = PIQ ~ MRINACount, data = brain, range.y
= "relative",
## range.x = "relative", nperm = 99)
##
## n = 40   r = 0.3868   r-square = 0.1496
## Parametric P-values:   2-tailed = 0.01367   1-tailed = 0.006837
## Angle between the two OLS regression lines = 0.03916 degrees
##
## Permutation tests of OLS, MA, RMA slopes: 1-tailed, tail corre-
sponding to sign
## A permutation test of r is equivalent to a permutation test of
the OLS slope
## P-perm for SMA = NA because the SMA slope cannot be tested
##
## Regression results
##   Method Intercept     Slope Angle (degrees) P-perm (1-tailed)
## 1    OLS     1.744 0.0001203         0.00689              0.01
## 2     MA     1.744 0.0001203         0.00689              0.01
## 3    SMA  -171.489 0.0003109         0.01781                NA
## 4    RMA  -499.546 0.0006719         0.03850              0.01
##
## Confidence intervals
##   Method 2.5%-Intercept 97.5%-Intercept 2.5%-Slope 97.5%-Slope
## 1    OLS         -84.08           87.56 0.00002611   0.0002144
## 2     MA         -83.81           87.30 0.00002611   0.0002144
## 3    SMA        -269.71          -98.60 0.00023068   0.0004190
## 4    RMA       -2515.70         -224.61 0.00036934   0.0028905
##
## Eigenvalues: 5224694673 429.4
##
## H statistic used for computing C.I. of MA: 0.000000008863
```

Задание

Выполните три блока кода:

Какие нарушения условий применимости линейных моделей здесь наблюдаются?

Что нужно писать в тексте статьи по поводу проверки валидности моделей?

Вариант 1. Привести электронные дополнительные материалы с необходимыми графиками.

Вариант 2. Привести в тексте работы результаты применения тестов на гомогенность дисперсии, автокоррелированность (если используются пространственые или временные предикторы) и нормальность распределения остатков.

Вариант 3. Написать в главе *«Материал и методика»* фразу вроде такой: «Визуальная проверка графиков рассеяния остатков не выявила заметных отклонений от условий равенства дисперсий и нормальности».

Следует отметить, что:

1. Перед построением модели необходимо провести исследование данных.

2. Не любая модель с достоверными результатами проверки H_0 валидна.

3. Обязательный этап работы с моделями – проверка условий применимости.

4. Если значения предикторов X случайны и принцпиальное значение для исследования имеет *значение* коэффициентов уравнения регрессии, то применяется регрессионная модель II.

Список библиографических источников

Основной

1. *Горяинова, Е. Р.* Прикладные методы анализа статистических данных: учеб. пособие / Е. Р. Горяинова, А. Р. Панков, Е. Н. Платонов. – М.: Изд. дом Высшей школы экономики, 2012. – 310 с.

2. *Лесковец, Ю.* Анализ больших наборов данных / Ю. Лесковец, А. Раджараман. – М.: ДМК, 2016. – 498 с.

3. *Крянев, А. В.* Метрический анализ и обработка данных / А. В. Крянев, Г. В. Лукин, Д. К. Удумян. – М.: Физматлит, 2012. – 308 с.

4. *Кулаичев, А. П.* Методы и средства комплексного анализа данных: учеб. пособие / А. П. Кулаичев. – М.: Форум, НИЦ ИНФРА-М, 2013. – 512 с.

5. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

6. 1Biostatistics with R: An Introduction to Statistics Through Biological Data (Use R!) / Babak Shahbaba, 2012.

7. Modern Epidemiology Third, Mid-cycle revision Edition / J. Kenneth Rothman, L. Timothy Lash Associate Professor, Sander Greenland, 2012.

8. Biostatistics and Epidemiology: A Primer for Health and Biomedical Professionals 4th ed. / Sylvia Wassertheil-Smoller, Jordan Smoller, 2015.

9. Introductory Time Series with R (Use R!) / Paul S. P. Cowpertwait, Andrew V. Metcalfe, 2009.

Дополнительный

10. *Новикова, Н. М.* Статистические методы в биологии и медицине: учеб.-метод. пособие / Н. М. Новикова. – Минск: МГЭУ им. А. Д. Сахарова, 2009. – 88 с.

11. *Кабаков, Р.* R в действии. Анализ и визуализация данных в программе R / Р. Кабаков. – М.: ДМК, 2016. – 588 с.

12. Биометрика – журнал для медиков и биологов, сторонников доказательной биомедицины [Электронный ресурс]. URL: http://www.-biometrica.tomsk.ru.

13. *Орлов, А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

14. *Чиркин, А. А.* Клинический анализ лабораторных данных / А. А. Чиркин. – Витебск: Медицинская литература, 2008. – 384 с.

15. *Петри, А.* Наглядная статистика в медицине / А. Петри, К. Сэбин. – М.: Изд. дом ГЭОТАР-МЕД, 2003 – 143 с.

Лабораторная работа № 10
Обобщенные, структурные и иные модели регрессии

1. Множественная регрессия

Мы рассмотрим:

- Технику подгонки множественных регрессионных моделей.
- Технику валидизации множественных регрессионных моделей.

Вы сможете:

- Подобрать множественную линейную модель.
- Протестировать ее состоятельность и валидность.
- Дать трактовку результатам.

Рабочий пример: какие факторы определяют распределение функциональных групп растений?

Считается, что распределение С3 растений регулируется только температуой той местности, где они произрастают. Проверяется гипотеза о связи распределения С3 растений не только со среднегодовой температурой, но и с уровнем осадков.

Зависимая перменная:

С3 – остностельное обилие травянистых растений, демонстрирующих C_3 путь фотосинтеза.

Предикторы:

LAT – штрота.
LONG – долгота.
MAP – среднегодовое количество осадков (мм).
MAT – среднегодвая температура (градусы).
JJAMAP – доля осадков, выпадающих в летние месяцы.
DJFMAP – доля осадкв, выпадающих в зимние месяцы.

Читаем данные:

```
plant <- read.csv("paruelo.csv")
plant <- plant[,-2]
head(plant)
```

```
##     C3 MAP  MAT JJAMAP DJFMAP   LONG   LAT
## 1 0.65 199 12.4   0.12   0.45 119.55 46.40
## 2 0.65 469  7.5   0.24   0.29 114.27 47.32
## 3 0.76 536  7.2   0.24   0.20 110.78 45.78
## 4 0.75 476  8.2   0.35   0.15 101.87 43.95
## 5 0.33 484  4.8   0.40   0.14 102.82 46.90
## 6 0.03 623 12.0   0.40   0.11  99.38 38.87
```

Можно ли ответить на вопрос таким методом?

```
cor(plant)
```

```
##          C3     MAP      MAT    JJAMAP   DJFMAP    LONG     LAT
## C3  1.00000 -0.06242 -0.511388  0.02301 -0.069231 0.04153 0.66698
## MAP -0.06242 1.00000  0.355091  0.11226 -0.404512 -0.73369 -0.24651
## MAT -0.51139  0.35509  1.000000 -0.08077 0.001478 -0.21311 -0.83859
## JJAMAP 0.02301  0.11226 -0.080771 1.00000 -0.791540 -0.49156 0.07417
## DJFMAP -0.06923 -0.40451 0.001478 -0.79154 1.000000 0.77074 -0.06512
## LONG  0.04153 -0.73369 -0.213109 -0.49156 0.770744 1.00000  0.09655
## LAT   0.66698 -0.24651 -0.838590  0.07417 -0.065125 0.09655 1.00000
```

Проблема 1. Взаимосвязь между переменными может находиться под контролем других переменых (частная корреляция).

Пробема 2. Множественные сравнения.

Необходимо учесть все взаимовлияния в одном анализе.

Нам предстоит построить множественную регрессионную модель.

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3} + \ldots + \beta_p x_{i,p} + \epsilon_i,$$

где y_i – значение зависимой переменной Y при значении предикторов $X_1 = x_{i,1}$, $X_2 = x_{i,2}$ и т. д.;

β_0 – свободный член (*intercept*). Значение Y при $X_1 = X_2 = X_3 = \ldots = X_p = 0$;

β_1 – *частный* угловой коэффициент для зависимости Y от X_1. Показывает насколько единиц изменяется Y при изменении X_1 на одну единицу, при условии, что все остальные предикторы не изменяются. β_2, β_3, ..., β_p – аналогично;

ϵ_i – варьирование Y, не объясняемое данной моделью.

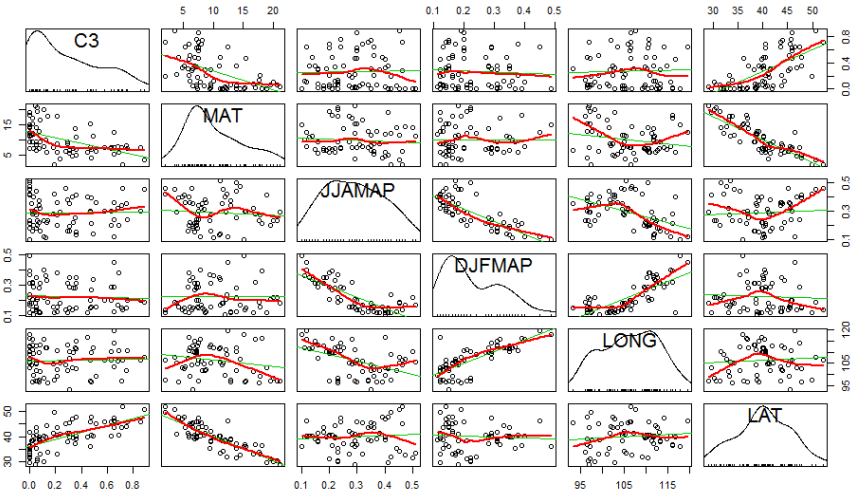
Геометрически – это плоскость в многомерном пространстве.

Проводим исследование данных.

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.3.3
```

```
scatterplotMatrix(plant[, -2], spread=FALSE)
```



Явные проблемы:

– Распределение зависимой переменной `C3` очень асимметрично.

– Есть сильные корреляции между некоторым предикторами.

– Возможна пространственная автокоррелированность.

Постоим линейную модель и сразу проверим ее на наличие автокорреляции остатков.

Задание

Напишите самостоятельно R код, необходимый для подбора уравнения множественной регрессии, и сразу проверьте модель на наличие автокорреляции остатков

Hint 1. Для того чтобы видеть названия переменных воспользуйтесь функцией `names()`

Hint 2. Подумайте, какие предикторы не следует включать в модель в соответствии с гипотезой, поставленной в исследовании.

Hint 3. Проведите тест Дарбина–Уотсона.

Решение

```
model0 <- lm(C3 ~  MAP + MAT + JJAMAP + DJFMAP, data = plant)
durbinWatsonTest (model0, max.lag = 3)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1         0.32837         1.251   0.000
##    2         0.19521         1.486   0.044
##    3         0.05101         1.728   0.350
##  Alternative hypothesis: rho[lag] != 0
```

Наличие положительных автокорреляций повашает вероятность ошибки I рода!

Возможное решение – нарушить «градиентный» характер материала.

Разделим выборку на две части.

```
plant1 <- plant[order(plant$LAT), ] # Упорядочиваем описания
в соответствии с широтой

include <- seq(1, 73, 2) # Отбираем каждое второе описание
exclude <- seq(1, 73) [!(seq(1, 73) %in% include)] # Исключаем
из списка отобранные описания

plant_modelling <- plant1[include, ]
plant_testing <- plant1[exclude, ]
```

Строим линейную модель для сокращенного набора данных:

```
model1 <- lm((C3)^(1/4) ~  MAP + MAT + JJAMAP + DJFMAP, data =
plant_modelling)

# Аналогичная запись
model1 <- lm((C3)^(1/4) ~  .-LONG -LAT, data = plant_model-
ling)
```

Проверим на автокоррелированность остатков полученную модель:

```
durbinWatsonTest(model1, max.lag = 3)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       -0.002602         1.967   0.716
##    2        0.076350         1.792   0.476
##    3       -0.137085         2.208   0.460
##  Alternative hypothesis: rho[lag] != 0
```

Смотрим на полученную модель:

```
summary(model1)
```

```
##
## Call:
## lm(formula = (C3)^(1/4) ~ . - LONG - LAT, data = plant_modelling)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6828 -0.1460  0.0434  0.1475  0.3863
##
## Coefficients:
```

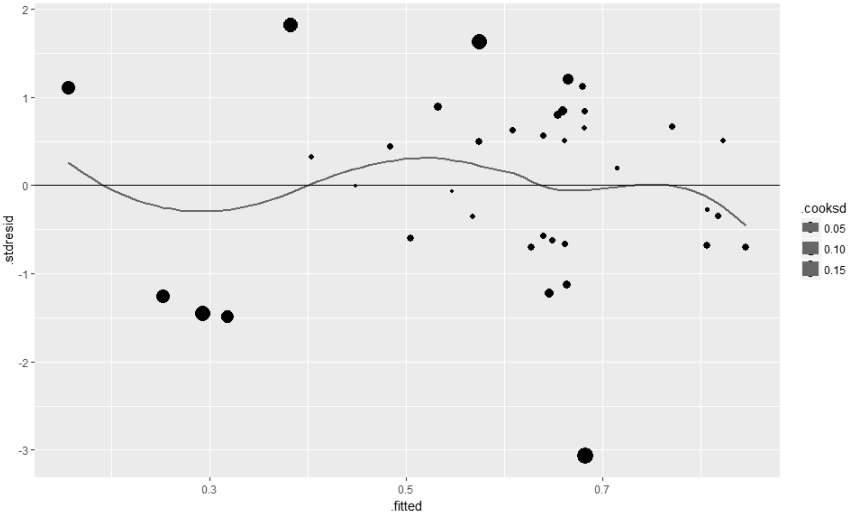
```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.708659   0.444180    3.85  0.00054 ***
## MAP          0.000216   0.000229    0.94  0.35280
## MAT         -0.029890   0.008827   -3.39  0.00189 **
## JJAMAP      -1.743669   0.663465   -2.63  0.01308 *
## DJFMAP      -1.840973   0.905072   -2.03  0.05030 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.231 on 32 degrees of freedom
## Multiple R-squared:  0.363,  Adjusted R-squared:  0.283
## F-statistic: 4.55 on 4 and 32 DF,  p-value: 0.00504
```

Проверка валидности модели:

```
library(ggplot2)
c3_diag <- fortify(model1)
```

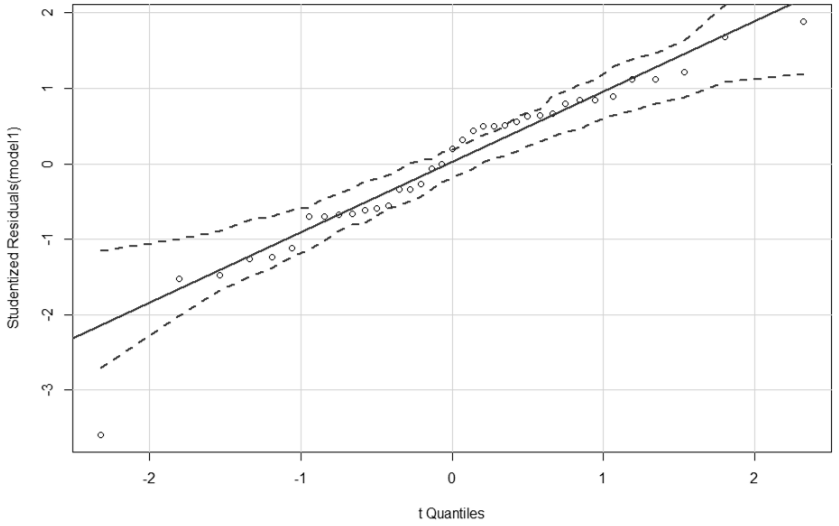
Смотрим на график распределения остатков (residual plot):

```
pl_resid <- ggplot(c3_diag, aes(x = .fitted, y = .stdresid,
size = .cooksd)) +
  geom_point() +
  geom_smooth(se=FALSE) +
  geom_hline(yintercept=0)

pl_resid
```



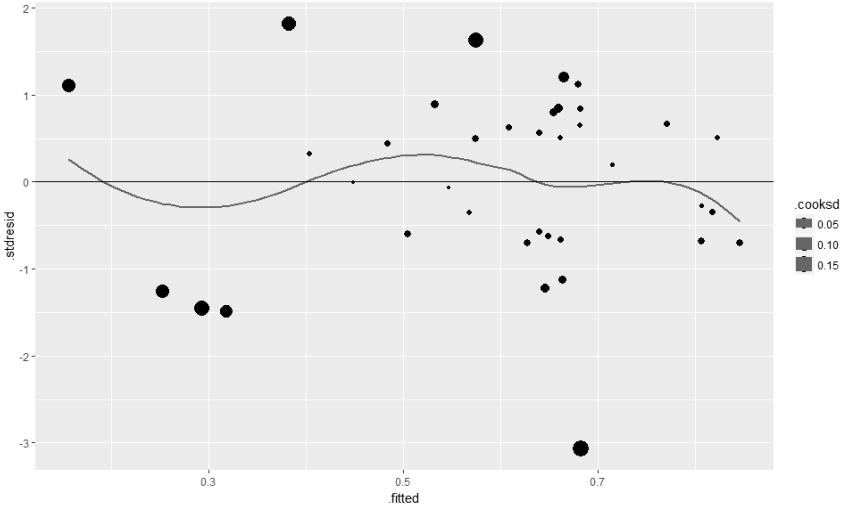
Проверяем на нормальность:

```
qqPlot(model1)
```



Проверяем на гетероскедастичность:

```
pl_resid
```



```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 3.3.3
```

```
## Warning: package 'zoo' was built under R version 3.3.3
```

```
bptest(model1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 4, df = 4, p-value = 0.4
```

Проверяем на мультиколлинеарность.

Мультиколлинеарность – наличие линейной зависимости между независимыми переменными (факторами) регрессионной модели.

При наличии мультиколлинеарности оценки параметров получаются неточными, а значит сложно будет дать интерпретацию влияния тех или иных факторов на объясняемую переменную

Признаки мультиколлинеарности: а) большие ошибки оценок параметров; б) большинство оценок параметров модели недостоверно, но F критерий всей модели свидетельствует о ее стаистической значимости.

Фактор инфляции дисперсии (Variance inflation factor)

```
vif(model1)
```

```
##    MAP    MAT JJAMAP DJFMAP
##  1.809  1.280  3.064  3.758
```

Логика вычисления VIF

1. Строим регрессионную модель:

$$x_1 = c_0 + c_2x_2 + c_2x_3 + \ldots + c_px_p.$$

2. Находим R^2 для данной модели

$$VIF = \frac{1}{1 - R^2}.$$

Что делать, если мультиколлинеарность выявлена?

Решение № 1. Удалить из модели избыточные предикторы.

1. Удалить из модели предикторы с VIF > 5.
2. Вновь провести вычисление VIF.
3. Возможно, удалить предикторы с VIF > 3.
4. Иногда полезно удалить и предикторы с VIF > 2 (это позволит сократить набор предикторов).

Решение № 2. Заменить исходные предикторы новыми перменными, полученными с помощью **метода главных компонент.**

Удалим из модели избыточный предиктор:

```
model2 <- update(model1, ~ . -DJFMAP)
vif(model2)
```

```
##    MAP    MAT JJAMAP
##  1.287  1.279  1.030
```

Смотрим на итоги:

```
summary(model2)
```

```
##
## Call:
## lm(formula = (C3)^(1/4) ~ MAP + MAT + JJAMAP, data = plant_mod-
elling)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6863 -0.1020  0.0583  0.1695  0.4101
##
## Coefficients:
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.858711   0.157635    5.45 0.0000049 ***
## MAP          0.000466   0.000202    2.31    0.0275 *
## MAT         -0.030486   0.009232   -3.30    0.0023 **
## JJAMAP      -0.643995   0.402453   -1.60    0.1191
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.242 on 33 degrees of freedom
## Multiple R-squared:  0.28,   Adjusted R-squared:  0.215
## F-statistic: 4.28 on 3 and 33 DF,  p-value: 0.0117
```

Какой из факторов MAT, JJAMAP или DJFMAP оказывает наиболее сильное влияние?

Для этого надо «уравнять» шкалы всех предикторов, то есть стандартизировать их.

Задание

Напишите R-код, который позволяет стандартизировать шкалу предиктора. Стандаризируйе, например, вектор `MAT.`

Решение

```
MAT_stand <- (plant_modelling$MAT - mean(plant_model-
ling$MAT))/sd(plant_modelling$MAT)
```

Можно использовать функцию `scale()`

```
model2_scaled <- lm((C3)^(1/4) ~ scale(MAP) + scale(MAT) +
scale(JJAMAP), data = plant_modelling)
```

Какой фактор оказывает наиболее сильное влияние на долю С3-растенний?

```
summary(model2_scaled)
```

```
##
## Call:
## lm(formula = (C3)^(1/4) ~ scale(MAP) + scale(MAT) + scale(JJAMAP),
##     data = plant_modelling)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6863 -0.1020  0.0583  0.1695  0.4101
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.5979     0.0398   15.03  2.5e-16 ***
## scale(MAP)      0.1055     0.0457    2.31   0.0275 *
## scale(MAT)     -0.1506     0.0456   -3.30   0.0023 **
## scale(JJAMAP)  -0.0655     0.0409   -1.60   0.1191
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.242 on 33 degrees of freedom
## Multiple R-squared:  0.28,   Adjusted R-squared:  0.215
## F-statistic: 4.28 on 3 and 33 DF,  p-value: 0.0117
```

Если модель хорошая, то она должна хорошо предсказывать.

Мы рассмотрим самый простой случай кросс-валидации:

```
predicted_C3_model1 <- predict(model1, newdata=plant_testing)
cor(predicted_C3_model1, plant_testing$C3)

predicted_C3_model2 <- predict(model2, newdata=plant_testing)
cor(predicted_C3_model2, plant_testing$C3)
```

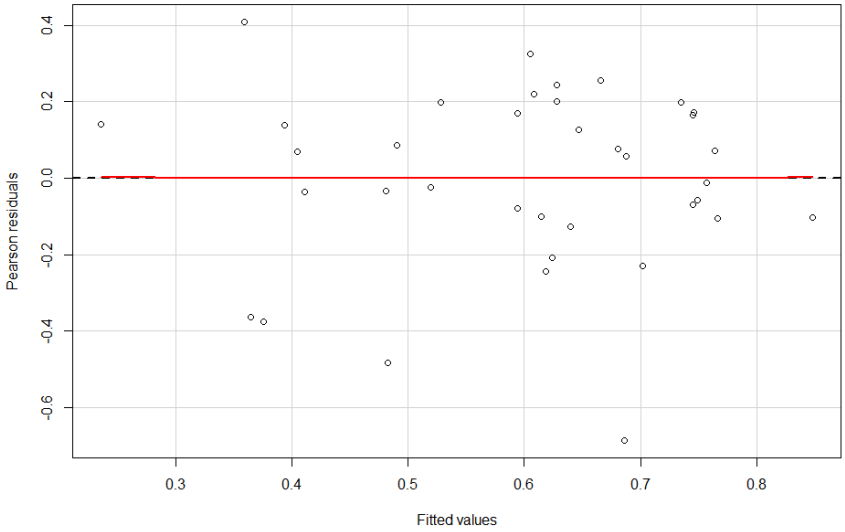
```
## [1] 0.398
## [1] 0.4067
```

Оцениваем валидность финальной модели:

```
durbinWatsonTest(model2)
bptest(model2)
vif(model2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1          -0.075         2.116   0.912
##  Alternative hypothesis: rho != 0
##
##  studentized Breusch-Pagan test
##
## data:  model2
## BP = 2.1, df = 3, p-value = 0.5
##
##    MAP    MAT JJAMAP
##  1.287  1.279  1.030
```

```
residualPlot(model2)
```



Следует отметить, что:

– при построении множественной регрессии важно, помимо проверки прочих условий применимости, проверить модель на наличие мультиколлинеарности;

– если модель построена на основе стандартизированных значений предикторов, то можно сравнивать влияние этих предикторов;

– кросс-валидация позволяет оценить степень работоспособности модели.

2. Сравнение линейных моделей

Вы сможете:

• Объяснить связь между качеством описания существующих данных и краткостью модели.

• Объяснить, что такое «переобучение» модели.

• Рассказать, каким образом происходит кросс-валидация моделей.

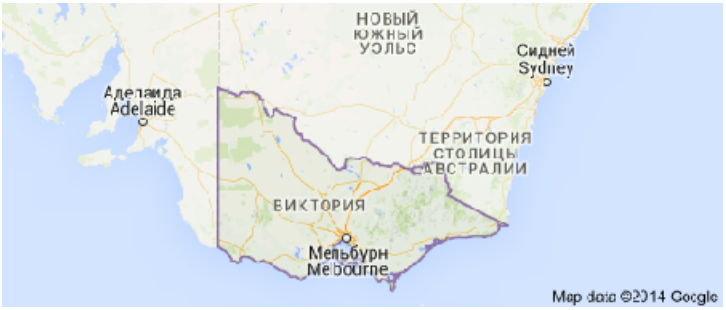
• Протестировать влияние отдельных параметров линейной регрессии при помощи сравнения вложенных моделей.

• Подобрать модель с оптимальной точностью подгонки к данным, оцененной по коэффициенту детерминации с поправкой или по C_p Маллоу.

• Оценить предсказательную силу модели при помощи k-кратной кросс-валидации.

Зачем нужно сравнивать модели?

Пример: птицы в лесах Австралии. От каких характеристик лесного участка зависит обилие птиц в лесах юго-западной Виктории, Австралия?



Штат разделен на 56 лесных участков, которые характеризуются следующими переменными:

- ABUND – обилие птиц,
- YR.ISOL – год изоляции участка,
- GRAZE – пастбищная нагрузка (1–5),
- ALT – высота над уровнем моря,
- L10DIST – логарифм расстояния до ближайшего леса,
- L10LDIST – логарифм расстояния до ближайшего большого леса,
- L10AREA – логарифм площади.

```
birds <- read.csv("loyn.csv")
```

Нужна оптимальная модель. Переменных много, хотим из них выбрать **оптимальный небольшой** набор:

1) при помощи разных критериев подберем несколько подходящих кандидатов;

2) выберем лучшую модель с небольшим числом параметров.

Принципы выбора:

1. Хорошее описание существующих данных:

- если мы включим много переменных, то лучше опишем данные;
- стандартные ошибки параметров будут большие, интерпретация сложная;
- большой R^2, маленький MSe.

2. Парсимония

- минимальный набор переменных, который может объяснить существующие данные;
- стандартные ошибки параметров будут низкие, интерпретация простая.

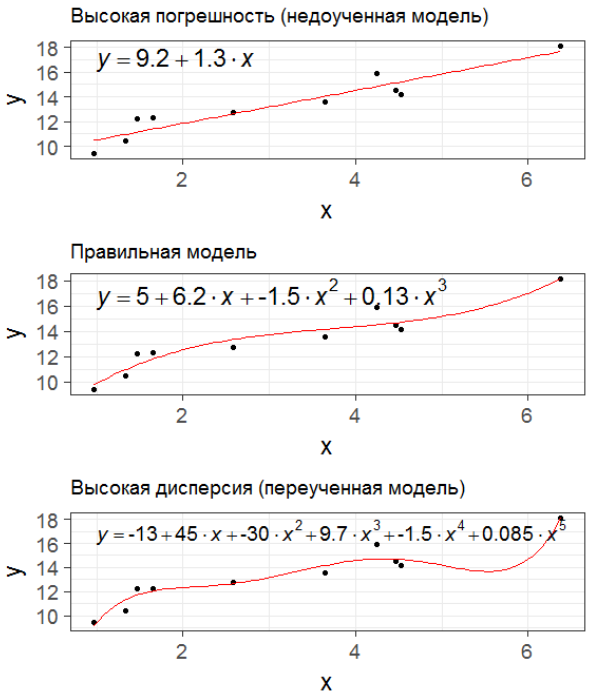
Компромисс при подборе оптимальной модели: точность или смещенная оценка.

Переобучение

Переобучение происходит, когда модель (из-за избыточного усложнения) описывает не только отношения между переменными, но и случайный шум.

При увеличении числа предикторов в модели (при ее усложнении), она точнее опишет данные, по которым подобрана, но на новых данных точность предсказаний будет низкой из-за «переобучения» (overfitting).

Легче всего проиллюстрировать на примере полиномиальной регрессии.



Критерии и методы выбора моделей зависят от задачи.

Объяснение закономерностей:

• Тестирование гипотез о влиянии факторов или удаление влияния одних переменных, для изучения других.

• Нужны точные тесты влияния предикторов: F-тесты (о нем сейчас) или likelihood-ratio тесты.

Описание закономерностей:

• Описание функциональной зависимости между зависимой переменной и предикторами.

• Нужна точность оценки параметров и парсимония: C_p Маллоу, «информационные» критерии (AIC, BIC, AICc, QAIC, и т. д.).

Предсказание:

• Предсказание значений зависимой переменной для **новых** данных.

• Нужна оценка качества модели на новых данных с использованием кросс-валидации (о ней сейчас).

Дополнительные критерии для сравнения моделей

• Диагностические признаки и качество подгонки:

– остатки, автокорреляция, кросс-корреляция, распределение ошибок, выбросы и др.

• Посторонние теоритические соображения:

– разумность, целесообразность модели, простота, ценность выводов.

Тестирование гипотез при помощи сравнения линейных моделей

Для тестирования гипотез о влиянии фактора можно сравнить модели с этим фактором и без него.

• Можно сравнивать для тестирования гипотез только вложенные модели (справедливо для F-критерия и для likelihood-ratio тестов):

Вложенные модели (nested models)

Две модели являются вложенными, если одну из них можно получить из другой, приравнивая некоторые коэффициенты более сложной модели к 0.

Какие из этих моделей вложены и в какие именно?

Полная модель (full model):

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon_i.$$

Неполные модели (reduced models):

$$y_i = \beta_0 + \beta_1 x_1 + \epsilon_i,$$
$$y_i = \beta_0 + \beta_2 x_2 + \epsilon_i.$$

Нулевая модель (null model):

$$y_i = \beta_0 + \epsilon_i.$$

• Неполные модели являются вложенными по отношению к полной модели, нулевая модель – вложенная по отношению к полной и к неполным.

• Неполные модели по отношению друг к другу – **не** вложенные.

Задание

Запишите все вложенные модели для данной полной модели

(1) $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon_i.$

• Модели:

(2) $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon_i$

(3) $y_i = \beta_0 + \beta_1 x_1 + \beta_3 x_3 + \epsilon_i$

(4) $y_i = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \epsilon_i$

(5) $y_i = \beta_0 + \beta_1 x_1 + \epsilon_i$

(6) $y_i = \beta_0 + \beta_2 x_2 + \epsilon_i$

(7) $y_i = \beta_0 + \beta_3 x_3 + \epsilon_i$

(8) $y_i = \beta_0 + \epsilon_i$

• Вложенность:

• (2)–(4) – вложены в (1);

• (5)–(7) – вложены в (1);

при этом:

(5) вложена в (1), (2), (3);

(6) вложена в (1), (2), (4);

(7) вложена в (1), (3), (4)

• нулевая модель – вложена во все

Сравнение линейных моделей при помощи F-критерия

Полная модель:

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} + \ldots + \beta_p x_{ip} + \epsilon_i$$

$$df_{reduced,full} = p,\ df_{error,full} = n - p - 1.$$

Уменьшенная модель:

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} + \epsilon_i$$

$$df_{reduced,reduced} = k,\ df_{error,reduced} = n - k - 1.$$

F-критерий для сравнения моделей

Есть ли выигрыш от включения фактора в модель?

$$F = \frac{(SS_{error,reduced} - SS_{error,full})/(df_{reduced,full} - df_{reduced,reduced})}{(SS_{error,full})/df_{error,full}}.$$

Задание

• Запишите формулу модели, которая описывает, как зависит обилие птиц в лесах Австралии (ABUND) от переменных:

• YR.ISOL – год изоляции участка;

• GRAZE – пастбищная нагрузка (1–5);

• ALT – высота над уровнем моря;

• L10DIST – логарифм расстояния до ближайшего леса;

• L10LDIST – логарифм расстояния до ближайшего большого леса;

• L10AREA – логарифм площади.

• Подберите модель, используя формулу: `frm_full <-`

• Какие переменные можно протестировать на предмет возможности исключения из модели?

Решение

L10DIST, L10LDIST, YR.ISOL не влияют

```
frm_full <- ABUND ~ L10AREA + L10DIST + YR.ISOL + L10LDIST + GRAZE
lm_full <- lm(frm_full, birds)
summary(lm_full)
```

```
#
# Call:
# lm(formula = frm_full, data = birds)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -16.368  -3.521   0.527   2.637  14.794
#
# Coefficients:
#             Estimate Std. Error t value  Pr(>|t|)
# (Intercept) -111.7820    89.7814   -1.25     0.219
# L10AREA        7.7557     1.4175    5.47 0.0000014 ***
# L10DIST       -1.2567     2.6321   -0.48     0.635
# YR.ISOL        0.0694     0.0447    1.55     0.127
# L10LDIST      -1.1065     2.0399   -0.54     0.590
# GRAZE         -1.8843     0.8881   -2.12     0.039 *
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 6.36 on 50 degrees of freedom
# Multiple R-squared:  0.681,   Adjusted R-squared:  0.649
# F-statistic: 21.3 on 5 and 50 DF,  p-value: 2.36e-11
```

Сравнение линейных моделей при помощи (частного) F-критерия: функция *anova(модель_1, модель_2)* в R

Модели обязательно должны быть вложенными!

Протестируем, нужны ли переменные L10LDIST, L10DIST, YR.ISOL.

Переменные, удаление которых **не ухудшает** модель, можно будет удалить и получить минимальную осмысленную модель.

Тестируем L10LDIST:

```
frm_ldist <- ABUND ~ L10AREA + L10DIST + YR.ISOL + GRAZE
lm_ldist <- lm(frm_ldist, birds)
anova(lm_ldist, lm_full)
```

```
# Analysis of Variance Table
#
# Model 1: ABUND ~ L10AREA + L10DIST + YR.ISOL + GRAZE
# Model 2: ABUND ~ L10AREA + L10DIST + YR.ISOL + L10LDIST + GRAZE
#   Res.Df  RSS Df Sum of Sq    F Pr(>F)
# 1     51 2036
# 2     50 2024  1      11.9 0.29   0.59
```

L10LDIST не улучшает модель – отбрасываем.

Тестируем L10DIST, при условии, что L10LDIST уже нет в модели:

```
frm_dist <- ABUND ~ L10AREA + YR.ISOL + GRAZE
lm_dist <- lm(frm_dist, birds)
anova(lm_dist, lm_ldist)
```

```
# Analysis of Variance Table
#
# Model 1: ABUND ~ L10AREA + YR.ISOL + GRAZE
# Model 2: ABUND ~ L10AREA + L10DIST + YR.ISOL + GRAZE
#   Res.Df  RSS Df Sum of Sq    F Pr(>F)
# 1     52 2071
# 2     51 2036  1      35.4 0.89   0.35
```

L10DIST не улучшает модель – отбрасываем.

Тестируем YR.ISOL, при условии, что L10DIST и L10LDIST нет в модели:

```
frm_yrisol <- ABUND ~ L10AREA + GRAZE
lm_yrisol <- lm(frm_yrisol, birds)
anova(lm_yrisol, lm_dist)
```

```
# Analysis of Variance Table
#
# Model 1: ABUND ~ L10AREA + GRAZE
# Model 2: ABUND ~ L10AREA + YR.ISOL + GRAZE
#   Res.Df  RSS Df Sum of Sq    F Pr(>F)
# 1     53 2201
# 2     52 2071  1       130 3.26  0.077 .
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

L10LDIST не улучшает модель – отбрасываем.

А вот GRAZE отбросить уже не получится:

```
frm_graze <- ABUND ~ L10AREA
lm_graze <- lm(frm_graze, birds)
anova(lm_graze, lm_dist)
```

```
# Analysis of Variance Table
#
# Model 1: ABUND ~ L10AREA
# Model 2: ABUND ~ L10AREA + YR.ISOL + GRAZE
#   Res.Df  RSS Df Sum of Sq    F  Pr(>F)
# 1     54 2867
# 2     52 2071  2       796 9.99 0.00021 ***
```

```
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

GRAZE улучшает модель – нужно оставить.

Минимальная модель:

```
frm_yrisol
```

```
# ABUND ~ L10AREA + GRAZE
```

Сравнение моделей по качеству подгонки к данным

1. Коэффициент детерминации

Обычный коэффициент детерминации оценивает долю объясненной изменчивости:

$$R^2 = \frac{SS_{regression}}{SS_{total}}.$$

$$\boldsymbol{R^2_{adjusted}}$$

Доля объясненной изменчивости с поправкой на число предикторов:

$$R^2_{adjusted} = 1 - (1 - R^2)\frac{n-1}{n-k} \leq R^2,$$

где n – число наблюдений,

k – количество параметров в модели.

У хорошей модели будет большой $\boldsymbol{R^2_{adjusted}}$

2. $\boldsymbol{C_p}$ Мэллоу (Mallow's $\boldsymbol{C_p}$)

оценивает «общую ошибку предсказания» с использованием p-параметров:

$$C_p = \frac{SS_{error,p-predictors}}{MS_{error,full}} - (n - 2p).$$

$\boldsymbol{C_p}$ Мэллоу связан с F-критерием

$$C_p = p + \big(F_p - 1\big)(m + 1 - p),$$

где m – общее число возможных параметров,

p – число параметров в уменьшенной модели.

У хорошей модели $\boldsymbol{C_p \approx p}$.

• Если нет ошибки предсказания, то $F_p \approx 1$ и $C_p \approx p$

• Если есть ошибка предсказания, то $F_p > 1$ и $C_p > p$

Найдем лучшую из всех моделей по коэффициенту детерминации

```
library(leaps)

# Warning: package 'leaps' was built under R version 3.3.3

crit_ar2 <- leaps(x = birds[, c(3, 6:10)], y = birds$ABUND,
                  names = names(birds[, c(3, 6:10)]),
                  method = "adjr2")
# crit_ar2$size # число предикторов
# crit_ar2$which # предикторы в модели
# crit_ar2$adjr2 # R^2 adj. для модели

# Номер строки лучшей модели (модели с макс. adjr2)
best_ar2 <- which.max(crit_ar2$adjr2)
# Какие переменные входят в модель?
crit_ar2$which[best_ar2, ]

#  YR.ISOL    GRAZE      ALT  L10DIST L10LDIST  L10AREA
#     TRUE     TRUE     TRUE    FALSE    FALSE     TRUE

# Записываем формулу лучшей модели по adjr2
frm_ar2 <- ABUND ~ YR.ISOL + GRAZE + ALT + L10AREA
```

В нашем случае переменных немного и мы можем перебрать все модели кандидаты. Если переменных много, можно использовать пошаговые процедуры или тестировать несколько осмысленных кандидатов.

Задание

Выберите лучшую модель по значению C_p Маллоу.

Нужно изменить параметр *method* в функции *leaps()*.

У лучшей модели будет минимальным модуль разницы между ее числом параметров и C_p.

Решение

```
crit_cp <- leaps(x = birds[, c(3, 6:10)], y = birds$ABUND, names =
names(birds[, c(3, 6:10)]), method = "Cp")
# Ищем лучшую модель
# полную модель нужно исключить
n_mod <- length(crit_cp$size)
best_cp <- which.min(abs(crit_cp$size[-n_mod] - crit_cp$Cp[-n_mod]))
# Какие переменные входят в модель?
crit_cp$which[best_cp, ]

#  YR.ISOL    GRAZE      ALT  L10DIST L10LDIST  L10AREA
#    FALSE     TRUE    FALSE    FALSE     TRUE     TRUE

frm_cp <- ABUND ~ GRAZE + L10DIST + L10AREA
```

Какую из моделей выбрать, если мы хотим предсказывать с их помощью?

Теперь у нас есть три многообещающих модели кандидата:

```
frm_yrisol

# ABUND ~ L10AREA + GRAZE

frm_ar2

# ABUND ~ YR.ISOL + GRAZE + ALT + L10AREA

frm_cp

# ABUND ~ GRAZE + L10DIST + L10AREA
```

Оценим их предсказательную силу.

Сравнение предсказательной силы линейных моделей с использованием кросс-валидации

Кросс-валидация

Если оценивать качество модели по тем же данным, по которым она была подобрана, оценки будут завышенными. Кросс-валидация решает эту проблему.

Делим данные **случайным образом** на **тренировочное и тестовое подмножества**, обычно в пропорции 60:40, 70:30 или 80:20:

Тренировочные данные:

• Используются для подбора модели (для обучения).

• Чтобы модель была хорошей, тренировочных данных **должно быть много.**

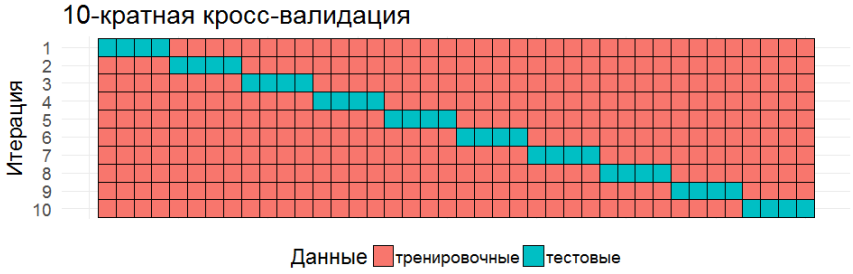
Тестовые данные:

Используются для оценки качества модели.

Чтобы надежно оценить качество модели, тестовых данных **тоже должно быть много.**

K-кратная кросс-валидация (k-fold cross-validation)

Делим данные **случайным образом** на k частей: $k-1$ часть используется для обучения, на k-й части тестируется модель. Процедура повторяется k раз.



k-кратная кросс-валидация лучше обычной, особенно, если данных не много.

RMSE – стандартная ошибка предсказания:

$$RMSE = \sqrt{\frac{\sum(\hat{y}_i - y_i)^2}{n}}.$$

Это параметр, определяющий ширину доверительных интервалов предсказаний.

Очень чувствительна к выборосам (альтернатива – MAE – средний модуль ошибок).

Можно сравнивать между моделями, только если они в одинаковых единицах (исходные данные моделей (не)преобразованы одинаково, зависимая переменная в одних и тех же единицах).

Нет жестких границ для RMSE «хорошей» модели, это относительная величина.

Бывает, что критерии противоречат друг другу, тогда решаем с учетом других соображений, например, простоты и интерпретируемости. Лучше меньше параметров.

Этапы сравнения моделей с использованием кросс-валидации:

1. Делим данные на тренировочное и тестовое подмножества.
2. Для каждой из моделей-кандидатов повторяем следующие шаги.
3. Подбираем на тренировочном подмножестве модель-кандидат.
4. Применяя тестовые данные, предсказываем ожидаемые значение y, используя модель-кандидат.
5. Рассчитываем RMSE для модели-кандидата (стандартное отклонение остатков):

$$RMSE = \sqrt{\frac{\sum(\hat{y}_i - y_i)^2}{n}}.$$

6. Сравниваем RMSE всех моделей кандидатов. Модель, у которой минимальное значение RMSE – лучшая.

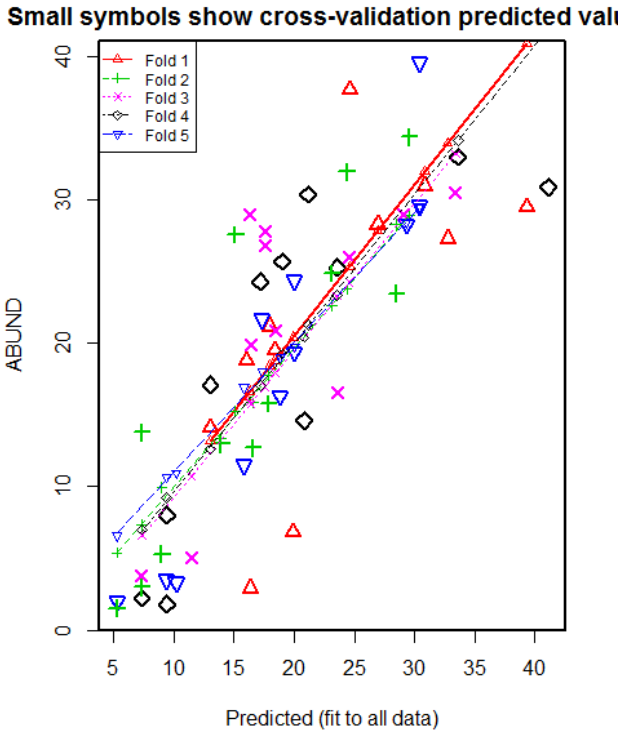
Кросс-валидация для линейных моделей

CVlm(df = *исходные_данные*, form.lm = *формула*, m = *кратность*)

```
library(DAAG)

# Loading required package: lattice

val_yrisol <- CVlm(birds, form.lm = frm_yrisol, m = 5)

# Analysis of Variance Table
#
# Response: ABUND
#           Df Sum Sq Mean Sq F value  Pr(>F)
# L10AREA    1   3471    3471    83.6 1.8e-12 ***
# GRAZE      1    666     666    16.0 0.00019 ***
# Residuals 53   2201      42
# ---
# Signif. codes:
# 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#
# fold 1
# Observations in test set: 11
#                  6    13    18     25    28    31
# Predicted   13.042 17.97  16.3 18.404 16.01  19.9
# cvpred      13.211 18.45  16.7 18.829 16.25  20.4
# ABUND       14.100 21.20   2.9 19.500 18.80   6.8
# CV residual  0.889  2.75 -13.8  0.671  2.55 -13.6
#                 45   47    50    52    55
# Predicted   26.934 24.6 30.85 32.76  39.3
# cvpred      27.841 25.3 32.02 34.02  40.9
# ABUND       28.300 37.7 31.00 27.30  29.5
# CV residual  0.459 12.4 -1.02 -6.72 -11.4
#
# Sum of squares = 720    Mean square = 65.4    n = 11
#
# fold 2
# Observations in test set: 12
#                 1     3     5    10   11    23     30
# Predicted    9.01  5.26  7.34  7.34 15.1 17.86 13.910
# cvpred       9.97  5.40  7.32  7.32 15.2 17.77 13.389
# ABUND        5.30  1.50 13.80  3.00 27.6 15.80 13.000
# CV residual -4.67 -3.90  6.48 -4.32 12.4 -1.97 -0.389
#                40    41    42    46    51
# Predicted   16.58 28.50 23.12 24.37 29.55
```

```
# cvpred      15.86 28.34 22.63 23.78 28.94
# ABUND       12.70 23.50 24.90 32.00 34.40
# CV residual -3.16 -4.84  2.27  8.22  5.46
#
# Sum of squares = 390    Mean square = 32.5    n = 12
#
# fold 3
# Observations in test set: 11
#                  7   17   22    26    29   33    34
# Predicted    7.34 16.3 11.5 18.40 16.41 17.6 17.62
# cvpred       6.62 15.9 10.7 17.98 15.78 17.0 16.97
# ABUND        3.80 29.0  5.0 20.90 19.90 27.8 26.80
# CV residual -2.82 13.1 -5.7  2.92  4.12 10.8  9.83
#                35    38       43    53
# Predicted   23.6 24.54 29.11992 33.41
# cvpred      23.3 24.26 28.99437 33.21
# ABUND       16.6 26.00 29.00000 30.50
# CV residual -6.7  1.74  0.00563 -2.71
#
# Sum of squares = 505    Mean square = 46    n = 11
#
# fold 4
# Observations in test set: 11
#                  4     8    12    14    15    19    24
# Predicted   13.04  7.34  9.41 20.82  9.41 17.19 23.57
# cvpred      12.61  7.02  9.25 20.43  9.25 17.07 23.38
# ABUND       17.10  2.20  1.80 14.60  8.00 24.30 25.30
# CV residual  4.49 -4.82 -7.45 -5.83 -1.25  7.23  1.92
#                36    39    54    56
# Predicted   21.15 19.00 33.62  41.1
# cvpred      21.32 19.29 34.19  42.3
# ABUND       30.40 25.70 33.00  30.9
# CV residual  9.08  6.41 -1.19 -11.4
#
# Sum of squares = 444    Mean square = 40.4    n = 11
#
# fold 5
# Observations in test set: 11
#                  2     9    16     20    21    27    32
# Predicted    5.26 10.19  9.41 20.044 20.04 18.87 17.36
# cvpred       6.57 10.98 10.65 19.779 19.78 19.06 18.02
# ABUND        2.00  3.30  3.50 19.400 24.40 16.30 21.70
# CV residual -4.57 -7.68 -7.15 -0.379  4.62 -2.76  3.68
#                37     44   48      49
# Predicted   15.81 29.301 30.4 30.4554
```

```
# cvpred      16.93 28.431 29.5 29.5634
# ABUND       11.50 28.300 39.6 29.6000
# CV residual -5.43 -0.131 10.1  0.0366
#
# Sum of squares = 305    Mean square = 27.7    n = 11
#
# Overall (Sum over all 11 folds)
#   ms
# 42.2
```



Задание

Посчитайте RMSE для модели val_yrisol

$$RMSE = \sqrt{\frac{\sum(\hat{y}_i - y_i)^2}{n}}$$

$\hat{y}_i$ – предсказанные во время кросс-валидации значения – *val_yrisol$cvpred*

y_i – реальные наблюдаемые значения зависимой переменной – *val_yrisol$ABUND*

Решение

```
# RMSE вручную
sqrt(mean((val_yrisol$cvpred - val_yrisol$ABUND)^2))

# [1] 6.5
```

Можно создать пользовательскую функцию для рассчета RMSE

```
rmse <- function(cv_obj, y_name){
  sqrt(mean((cv_obj$cvpred - cv_obj[, y_name])^2))
}

# теперь можно пользоваться функцией
rmse(val_yrisol, "ABUND")

# [1] 6.5
```

Задание

• Сделайте 5-кратную кросс-валидацию оставшихся двух моделей и полной модели.

```
frm_cp
frm_ar2
frm_full
```

• Посчитайте их RMSE.

Какая из моделей-кандидатов дает более качественные предсказания?

Решение

Кросс-валидация

```
val_cp <- CVlm(birds, form.lm = frm_cp, m = 5)
val_ar2 <- CVlm(birds, form.lm = frm_ar2, m = 5)
val_full <- CVlm(birds, form.lm = frm_full, m = 5)
```

Считаем RMSE

```
rmse(val_cp, "ABUND")

# [1] 6.5

rmse(val_ar2, "ABUND")

# [1] 6.84

rmse(val_full, "ABUND")

# [1] 6.93
```

Какие модели дают более качественные предсказания?

```
rmse(val_yrisol, "ABUND"); rmse(val_cp, "ABUND")
# [1] 6.5
# [1] 6.5
rmse(val_ar2, "ABUND"); rmse(val_full, "ABUND")
# [1] 6.84
# [1] 6.93
```

Судя по значениям RMSE, это модели

```
frm_yrisol
# ABUND ~ L10AREA + GRAZE
frm_cp
# ABUND ~ GRAZE + L10DIST + L10AREA
```

В данном случае можно предпочесть *frm_yrisol* как более простую.

Следует отметить, что:

• модели, которые качественно описывают существующие данные, включают много параметров, но предсказания с их помощью менее точны из-за переобучения;

• для выбора оптимальной модели используются разные критерии, в зависимости от задачи;

• сравнивая вложенные модели, можно отбраковать переменные, включение которых в модель не улучшает ее;

• оптимальный набор переменных для более качественного описания **существующих данных** можно подобрать, сравнивая модели по $R^2_{adjusted}$ и C_p Маллоу;

• оценить предсказательную силу модели на **новых данных** можно при помощи кросс-валидации, сравнив ошибки предсказаний.

3. Линейные модели с непрерывными и дискретными предикторами

Мы рассмотрим:

• Линейные модели, включающие в себя как непрерывные, так и дискретные предикторы,

• Простейший экспериментальный план, требующий анализа ковариации (ANCOVA).

Вы сможете:

• Написать R-код, необходимый для подгонки линейной модели, включающей как дискретные, так и непрерывные предикторы,

• Дать трактовку коэффициентам такой линейной модели.

Кто дольше смотрит телевизор?

• Было опрошено 2068 студентов, обучающихся на курсах по статистике одного из американских университетов.

• Каждый студент был оценен по 14 параметрам.

• Мы рассмотрим лишь часть.

Зависимая перменная:

watchtv – вемя, проводимое за просмотром телепердач (часы в неделю).

Независимые предикторы:

age – возраст (годы);

miles – расстояние от места жительства студента до университета (мили);

year – курс обучения;

gender – пол.

Пример взят с сайта *http://www.amstat.org/publications/jse/jse_data_archive.htm* файл: eyecolorgenderdata.csv

Смотрим на датасет

```
tv <- read.csv("eyecolorgenderdata.csv", header = TRUE)
head(tv)
```

```
##  gender age year eyecolor height miles brothers sisters computer-
time exercise
## 1 female  18  first   hazel    68   195      0     1         20
Yes
## 2   male  20  third   brown    70   120      3     0         24
No
## 3 female  18  first   green    67   200      0     1         35
Yes
## 4   male  23 fourth   hazel    74   140      1     1          5
Yes
## 5 female  19 second    blue    62    60      0     1          5
Yes
## 6   male  19 second   green    67     0      0     1          5
Yes
##   exercisehours musiccds playgames watchtv
## 1             3       75         6      18
## 2             0       50         0       3
## 3             3       53         8       1
## 4            25       50         0       7
```

```
## 5             4       30         2       5
## 6             8      100         0      10
```

Непрерывные (continuous) предикторы:

age
miles

Дискретные предикторы, или факторы (categorical factors):

year

```
levels(tv$year)
```

```
## [1] "first"  "fourth" "other"  "second" "third"
```

gender

```
levels(tv$gender)
```

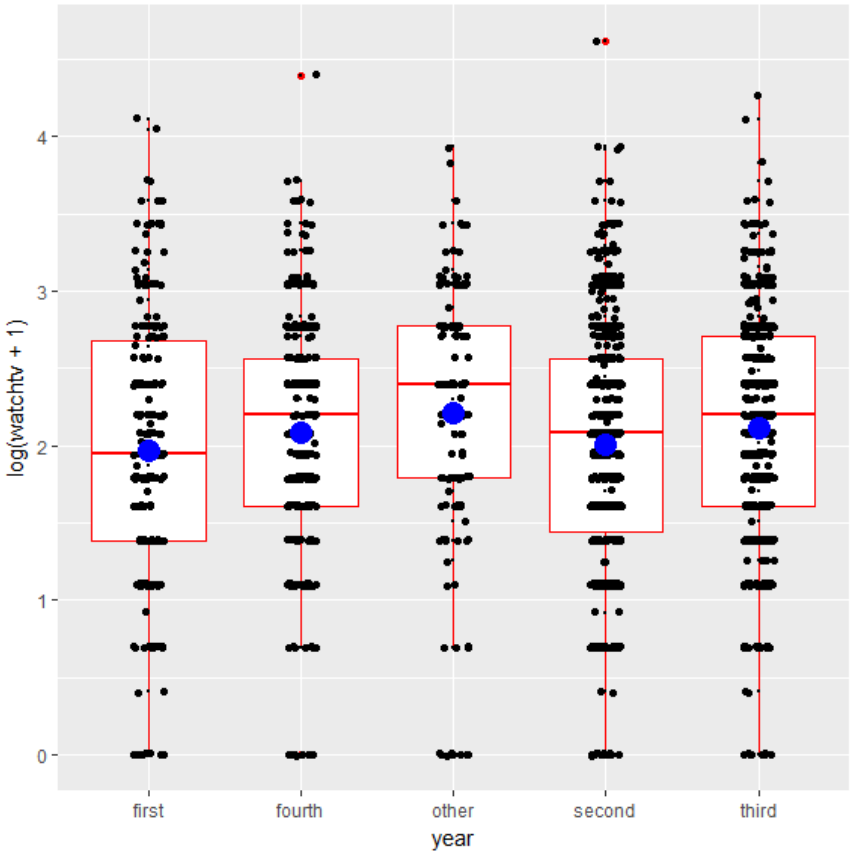
```
## [1] "female" "male"
```

Можо ли предсказать время, затраченное на просмотр телепередач, для студентов, обучающихся на разных курсах?

Это означает, что мы должны построить модель, предсказывающую время просмотра TV в зависмости от дискретного предиктора *year*.

Представляем графически связь между зависимой переменной и предиктором:

```
library(ggplot2)
ggplot(tv, aes(x = year, y = log(watchtv + 1)))  +
  geom_boxplot() + geom_point() +
  geom_jitter(position = position_jitter(width = 0.1)) +
  stat_summary(fun.y=mean, colour="blue", geom="point",
               size=5, show_guide = FALSE)
```



Строим линейую модель, связывающую зависимую переменную и предиктор

```
tv_model1 <- lm(watchtv ~ year, data = tv)
summary(tv_model1)
```

```
##
## Call:
## lm(formula = watchtv ~ year, data = tv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11.52  -5.53  -1.82   3.18  91.18
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.330      0.520   17.93   <2e-16 ***
## yearfourth     0.201      0.693    0.29    0.771
## yearother      2.192      0.869    2.52    0.012 *
## yearsecond    -0.505      0.606   -0.83    0.405
## yearthird      0.160      0.609    0.26    0.793
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.18 on 2062 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.00622,    Adjusted R-squared:  0.00429
## F-statistic: 3.22 on 4 and 2062 DF,  p-value: 0.012
```

```
anova(tv_model1)
```

```
## Analysis of Variance Table
##
## Response: watchtv
##              Df Sum Sq Mean Sq F value Pr(>F)
## year          4    863   215.7    3.22  0.012 *
## Residuals 2062 137900    66.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Суть коэффициентов в модели с дискретными предикторами

$$y_i = \beta_0 + \beta_1 I_{level_1} + \beta_2 I_{level_2} + \ldots + \beta_m I_{level_m} + \epsilon_i,$$

где β_0 – свободный член (intercept);

ϵ_i – неучтенная моделью изменчивость;

β_m – показывает, на сколько единиц отличается среднее значение *Y* для m-го уровня предиктора от уровня, выбранного за базовый.

Базовый уровень выбирается в зависимости от целей исследования. По умолчанию, функция *lm()* выбирает за базовый уровень первое по алфавиту значение (в нашей модели – year = *first*).

I_{level_m} – *dummy variable* – логическй «переключатель», множитель, принимающий значение 1 или 0, в зависимости от того, какой предиктор рассматривается.

Средние значения для каждого уровня

Среднее значение для базового уровня дискретного предиктора:

$$y_{mean_{ref}} = b_0 + \beta_1 * 0 + \beta_2 * 0 + \ldots + \beta_m * 0.$$

Среднее значение для первого, после базового, уровня:

$$y_{mean_1} = b_0 + \beta_1 * 1 + \beta_2 * 0 + \ldots + \beta_m * 0.$$

Среднее значение для второго, после базового, уровня:

$$y_{mean_2} = b_0 + \beta_1 * 0 + \beta_2 * 1 + \ldots + \beta_m * 0.$$

Среднее значение для m-го, после базового, уровня:

$$y_{mean_m} = b_0 + \beta_1 * 0 + \beta_2 * 0 + \ldots + \beta_m * 1.$$

Подставим подобранные коэффициенты в модель

```
coefficients(tv_model1)
```

```
## (Intercept)  yearfourth   yearother  yearsecond   yearthird
##      9.3300      0.2014      2.1918     -0.5050      0.1596
```

$$y_{mean} = 9.3299595 + 0.2013884 I_{fourth} + 2.1917796 I_{other} - 0.5050316 I_{second} + 0.1595771 I_{third}$$

Задание

Используя знание коэффициентов линейной модели, вычислите средние значения зависимой переменной для каждого уровня предиктора *year*.

Решение

```
#first
9.3299595 + 0.2013884*0 + 2.1917796*0 -0.5050316*0 +
0.1595771*0
```

```
## [1] 9.33
```

```
#fourth
9.3299595 + 0.2013884*1 + 2.1917796*0 -0.5050316*0 +
0.1595771*0
```

```
## [1] 9.531
```

```
#other
9.3299595 + 0.2013884*0 + 2.1917796*1 -0.5050316*0 +
0.1595771*0
```

```
## [1] 11.52
```

```
#second
9.3299595 + 0.2013884*0 + 2.1917796*0 -0.5050316*1 +
0.1595771*0
```

```
## [1] 8.825
```

```
#third
9.3299595 + 0.2013884*0 + 2.1917796*0 -0.5050316*0 +
0.1595771*1
```

```
## [1] 9.49
```

Сверим выводы с результатами, полученными иным способом:

```
means <- function(x) mean(x, na.rm=TRUE)

mean_year <- tapply(tv$watchtv, tv$year, FUN = means)
mean_year
```

```
##  first fourth  other second  third
##  9.330  9.531 11.522  8.825  9.490
```

Итак, среднее значение для каждого из уровней дискретного предиктора вычисляется следующим образом:

$$y_{mean_{ref}} = \beta_0$$

$$y_{mean_1} = \beta_0 + \beta_1$$

$$y_{mean_2} = \beta_0 + \beta_2$$

$$y_{mean_m} = \beta_0 + \beta_m$$

Что произойдет с моделью, если поменять базовый уровень?

Меняем базовый уровень:

```
tv_model2 <- lm(watchtv ~ relevel(year, ref="other"), data = tv)
coefficients(tv_model2)
```

```
##                  (Intercept)  relevel(year, ref = "other")first
##                       11.522                             -2.192
## relevel(year, ref = "other")fourth relevel(year, ref =
"other")second
##                       -1.990                             -2.697
##  relevel(year, ref = "other")third
##                            -2.032
```

Изменилась ли суть модели от изменения базового уровня?

```
summary(tv_model1)
```

```
##
## Call:
## lm(formula = watchtv ~ year, data = tv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11.52  -5.53  -1.82   3.18  91.18
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.330      0.520   17.93   <2e-16 ***
```

```
## yearfourth     0.201      0.693    0.29    0.771
## yearother      2.192      0.869    2.52    0.012 *
## yearsecond    -0.505      0.606   -0.83    0.405
## yearthird      0.160      0.609    0.26    0.793
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.18 on 2062 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.00622,    Adjusted R-squared:  0.00429
## F-statistic: 3.22 on 4 and 2062 DF,  p-value: 0.012
```

```
summary(tv_model2)
```

```
##
## Call:
## lm(formula = watchtv ~ relevel(year, ref = "other"), data = tv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11.52  -5.53  -1.82   3.18  91.18
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 11.522  0.696  16.55  < 2e-16 ***
## relevel(year, ref = "other")first -2.192  0.869  -2.52  0.01175 *
## relevel(year, ref = "other")fourth -1.990  0.833 -2.39  0.01699 *
## relevel(year, ref = "other")second -2.697  0.762 -3.54  0.00041 ***
## relevel(year, ref = "other")third  -2.032  0.765 -2.66  0.00792 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.18 on 2062 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.00622,    Adjusted R-squared:  0.00429
## F-statistic: 3.22 on 4 and 2062 DF,  p-value: 0.012
```

О чем говорят значения t-критерия?

```
summary(tv_model2)
```

```
##
## Call:
## lm(formula = watchtv ~ relevel(year, ref = "other"), data = tv)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -11.52  -5.53  -1.82   3.18  91.18
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    11.522     0.696   16.55  < 2e-16 ***
## relevel(year, ref = "other")first  -2.192  0.869   -2.52  0.01175 *
## relevel(year, ref = "other")fourth  -1.990  0.833  -2.39  0.01699 *
## relevel(year, ref = "other")second  -2.697  0.762  -3.54  0.00041 ***
## relevel(year, ref = "other")third  -2.032  0.765  -2.66  0.00792 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.18 on 2062 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.00622,    Adjusted R-squared:  0.00429
## F-statistic: 3.22 on 4 and 2062 DF,  p-value: 0.012
```

Все действия с моделями, включающими дискретные предикторы, аналогичны действиям с обычным регрессионными моделями.

Диагностика:

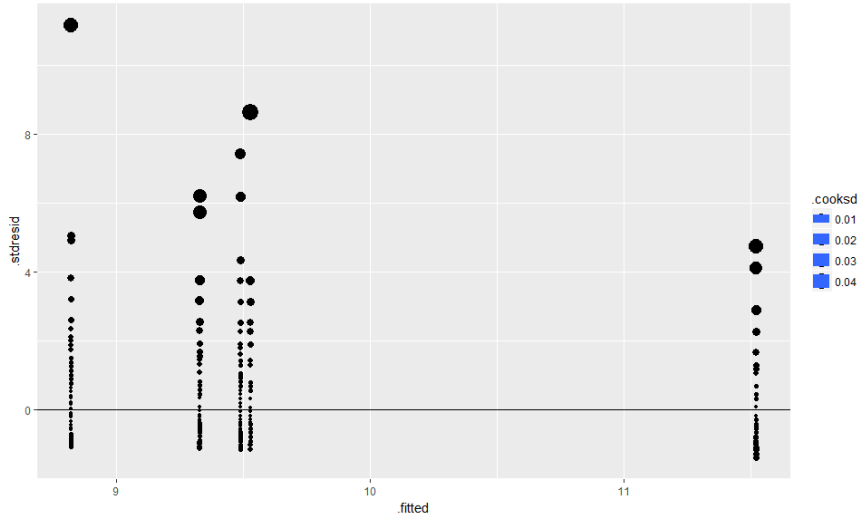
```
# library(car)
# residualPlots(tv_model1)
tv_model_diag <- fortify(tv_model1)
head(tv_model_diag)
```

```
##   watchtv  year  .hat .sigma   .cooksd .fitted .resid .stdresid
## 1   18  first 0.004049  8.178 0.000917537  9.330  8.670  1.0623
## 2   3  third 0.001495  8.179 0.000188823  9.490 -6.490  -0.7941
## 3   1  first 0.004049  8.178 0.000846969  9.330 -8.330  -1.0207
## 4   7 fourth 0.003135  8.180 0.000060450  9.531 -2.531  -0.3100
## 5   5 second 0.001441  8.179 0.000063226  8.825 -3.825  -0.4681
## 6  10 second 0.001441  8.180 0.000005967  8.825  1.175   0.1438
```

График рассеяния остатков (Residual plot):

```
ggplot(tv_model_diag, aes(x = .fitted, y = .stdresid, size =
.cooksd)) + geom_point() + geom_hline(yintercept=0) +
geom_smooth(se = FALSE)
```

```
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```




```
library(car)
```

```
## Warning: package 'car' was built under R version 3.3.3
```

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 3.3.3
```

```
## Warning: package 'zoo' was built under R version 3.3.3
```

```
bptest(tv_model1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  tv_model1
## BP = 2.9, df = 4, p-value = 0.6
```

```
durbinWatsonTest(tv_model1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1        -0.01281         2.024   0.588
##  Alternative hypothesis: rho != 0
```

Оценка состоятельности модели с помощью функции *anova()*

```
anova(tv_model1)
```

```
## Analysis of Variance Table
##
## Response: watchtv
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## year         4    863   215.7    3.22  0.012 *
## Residuals 2062 137900    66.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

В одну модель можно включить одновременно и дискретные, и неперывные предикторы!

```
tv_model3 <- lm(watchtv ~  age + miles + gender + year, data = tv)
```

Проанализируем результаты подбора модели.

Кто (мужчины или женщины) в среднем смотрит телевизор дольше и на сколько часов?

```
summary(tv_model3)
```

```
##
## Call:
## lm(formula = watchtv ~ age + miles + gender + year, data = tv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -13.96  -5.47  -1.91   3.54  85.55
##
## Coefficients:
##              Estimate Std. Error t value     Pr(>|t|)
## (Intercept)  9.237913   1.526265    6.05 0.0000000017 ***
## age         -0.046673   0.076061   -0.61      0.53953
## miles        0.000540   0.000162    3.33      0.00089 ***
## gendermale   1.807176   0.362732    4.98 0.0000006818 ***
## yearfourth   0.294231   0.740318    0.40      0.69109
## yearother    2.323852   0.955360    2.43      0.01508 *
## yearsecond  -0.428570   0.609407   -0.70      0.48198
## yearthird    0.178809   0.626024    0.29      0.77519
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.12 on 2043 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.0233, Adjusted R-squared:  0.02
## F-statistic: 6.97 on 7 and 2043 DF,  p-value: 0.000000031
```

Задание

Попробуйте оптимизировать полную модель.

Для этого вам понадбится исследовать полную и вложенную в нее редуцированную модель.

Решение

```
tv_model_reduced <- lm(watchtv ~  miles + gender +
                         year, data = tv)
anova(tv_model3, tv_model_reduced)
```

```
## Analysis of Variance Table
##
## Model 1: watchtv ~ age + miles + gender + year
## Model 2: watchtv ~ miles + gender + year
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1   2043 134610
## 2   2044 134635 -1     -24.8 0.38   0.54
```

```
summary(tv_model_reduced)
```

```
##
## Call:
## lm(formula = watchtv ~ miles + gender + year, data = tv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -13.94  -5.47  -1.93   3.52  85.68
##
## Coefficients:
##              Estimate Std. Error t value   Pr(>|t|)
## (Intercept)  8.363751   0.547654   15.27    < 2e-16 ***
## miles        0.000531   0.000162    3.29      0.001 **
## gendermale   1.788124   0.361345    4.95 0.00000081 ***
## yearfourth   0.135875   0.693776    0.20      0.845
## yearother    2.079047   0.867941    2.40      0.017 *
## yearsecond  -0.456088   0.607662   -0.75      0.453
## yearthird    0.095458   0.611016    0.16      0.876
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.12 on 2044 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.0232, Adjusted R-squared:  0.0203
## F-statistic: 8.08 on 6 and 2044 DF,  p-value: 0.0000000121
```

Классический ковариационный анализ (ANCOVA)

Часто используется в экспериментальных работах. *Пример:* как влияет на прирост массы коз интенсивность профилактики паразитарных заболеваний?

Пример взят из библиотеки данных. URL: *http://www.statlab.uni-heidelberg.de/data/ancova/goats.data.*

Из датасета удалено два измерения.

Читаем данные:

```
goat <- read.csv("Goat_treatment_1.csv")
```

Задание

1. Постройте модель, описывающую зависимость между увеличением веса коз и способом прфилактической обработки животных.
2. Оцените состоятельность этой модели.

Решение

```
goat_model <- lm(Weightgain ~ Treatment, data = goat)
summary(goat_model)
```

```
##
## Call:
## lm(formula = Weightgain ~ Treatment, data = goat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.737 -1.579  0.263  1.263  4.421
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.737      0.469   14.37   <2e-16 ***
## Treatmentstandard   -1.158      0.663   -1.75    0.089 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.04 on 36 degrees of freedom
## Multiple R-squared:  0.0781, Adjusted R-squared:  0.0525
## F-statistic: 3.05 on 1 and 36 DF,  p-value: 0.0892
```

Необходимо учитывать ковариату – начальный вес коз!

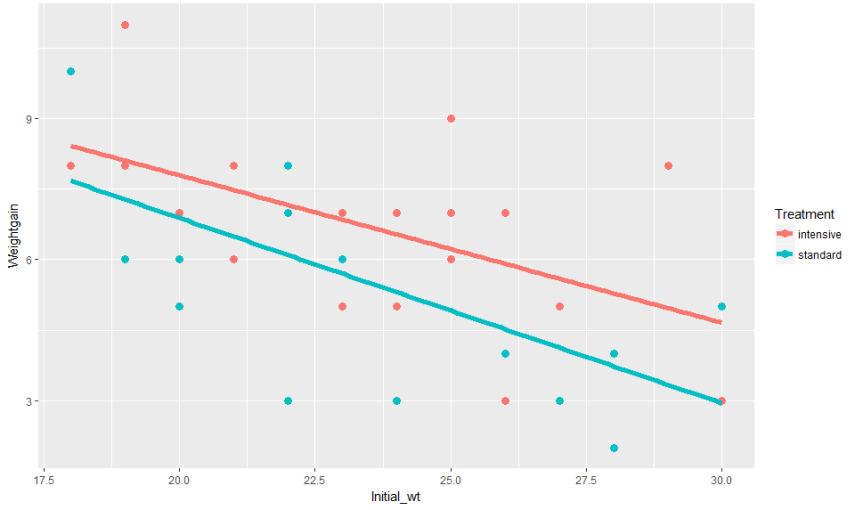
```
goat_model_cov <- lm(Weightgain ~ Treatment + Initial_wt, data =
goat)
summary(goat_model_cov)
```

```
##
## Call:
## lm(formula = Weightgain ~ Treatment + Initial_wt, data = goat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3.04  -1.23  -0.04   0.94   3.26
##
## Coefficients:
##                   Estimate Std. Error t value     Pr(>|t|)
## (Intercept)        15.0081     1.8676    8.04 0.0000000019 ***
## Treatmentstandard  -1.1765     0.5342   -2.20        0.034 *
## Initial_wt         -0.3539     0.0783   -4.52 0.0000672692 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.65 on 35 degrees of freedom
## Multiple R-squared:  0.418,  Adjusted R-squared:  0.385
## F-statistic: 12.6 on 2 and 35 DF,  p-value: 0.0000767
```

Вопрос: как измеяет прирост при использовании более интесивной профилактики?

Визуализация результатов анализа

```
ggplot(goat, aes(x = Initial_wt, y = Weightgain, color = Treatment)) +
  geom_point(size=3) + geom_smooth(method = "lm", se = FALSE, size=2)
```



Следует отметить, что:

• коэффициенты регрессии при дискретных предикторах позволяют сопоставить средние значения для каждого уровня фактора со средним уровнем некоторого базового уровня данного предиктора;

• при изменении базового уровня коэффициенты изменятся, но суть модиели останется той же;

• в одной модели можно объединить как непрерывные, так и дискретные предикторы;

• при анализе экспериментальных данных включение в анализ некоторых ковариат может помочь лучше увидеть закономерности.

4. Регрессионный анализ для бинарных данных

Мы рассмотрим:

• Регрессионный анализ для бинарных зависимых переменных.

• Некторые положения теории обобщенных линейных моделей (Generalized linear models).

Вы сможете:

• Объяснить, что такое метод максимального правдоподобия.

• Построить логистическую регрессионную модель.

• Дать трактовку параметрам логистической регрессионной модели.

• Провести анализ девианы, основанный на логистичской регрессии.

Бинарные данные – очень распространенный тип зависимых переменных:

• Вид есть – вида нет.

• Кто-то в результате эксперимента выжил или умер.

• Пойманное животное заражено паразитами или здорово.

• Команда выиграла или проиграла и т. д.

Для моделирования таких данных метод наименьших квадратов не годится.

Пример: на каком острове лучше искать ящериц?

```
liz <- read.csv("polis.csv")
head(liz)
```

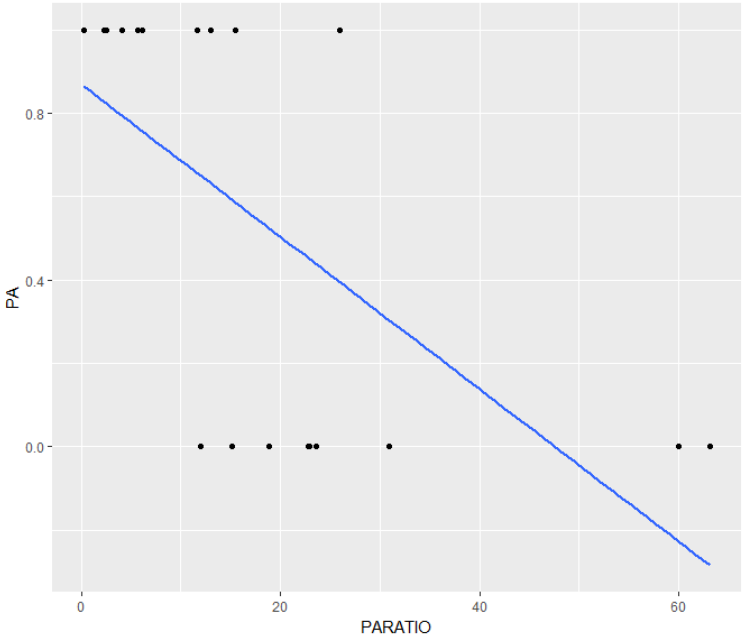
```
##     X.ISLAND PARATIO UTA PA PREDICT
## 1       Bota   15.41   P  1  0.5554
## 2     Cabeza    5.63   P  1  0.9145
## 3    Cerraja   25.92   P  1  0.1106
## 4 Coronadito   15.17   A  0  0.5684
## 5     Flecha   13.04   P  1  0.6777
## 6   Gemelose   18.85   A  0  0.3699
```

Зависит ли встречаемость ящериц от характеристик острова?
Обычную линейную регрессию можно подобрать...

```
fit <- lm(PA ~ PARATIO, data = liz)
summary(fit)
```

```
##
## Call:
## lm(formula = PA ~ PARATIO, data = liz)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.649 -0.445  0.178  0.264  0.605
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.86881    0.14088    6.17  0.00001 ***
## PARATIO     -0.01828    0.00557   -3.28   0.0044 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.413 on 17 degrees of freedom
## Multiple R-squared:  0.388,  Adjusted R-squared:  0.352
## F-statistic: 10.8 on 1 and 17 DF,  p-value: 0.00438
```

…но она категорически не годится.



Эти данные лучше описывает логистическая кривая, которая описывается формулой:

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}.$$

Зависимую величину можно представить несколькими способами:

1. Дискретный результат: 1 или 0.

2. Вероятность: $\pi = \frac{N_1}{N_{total}}$ варьирует от 0 до 1.

3. Шансы (odds): $g = \frac{\pi}{1-\pi}$ варьируют от 0 до $+\infty$.

Вероятность и шанс

Вероятность – это отношение, выражающее то, насколько возможно данное событие по отношению к другим исходам.

Шансы – это отношение того, что событие произойдет, к тому, что оно не произойдет.

• Логиты (logit): $\ln(g) = \ln(\frac{\pi}{1-\pi})$ варьируют от $-\infty$ до $+\infty$.

Логистическая модель после логит-перобразования становится линейной:

$$g(x) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x.$$

Для подбора параметров уравнения применяется метод максимального правдоподобия:

$$g(x) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x.$$

Метод максимального правдоподбия применяется для тех случаев, когда распределение остатков не может быть описано нормальным распеделением.

В результате итерационных проедур происходит подбор таких коэффициентов, при которых вероятность получения имеющегося у нас набора данных оказывается максимальной.

В основе метода максимального лежит вычисление функции правдоподобия:

$$Lik(\beta_0, \beta_1) = \Pi L_i.$$

Удобнее работать с логарифмом функции максимльного правдоподобия – $logLik$.

Функция максимального правдоподобия оценивает вероятность получения наших данных при условии страведливости данной модели.

Подберем модель с помощью функции *glm()*

```
liz_model <- glm(PA ~ PARATIO , family="binomial", data = liz)
summary(liz_model)
```

```
##
## Call:
## glm(formula = PA ~ PARATIO, family = "binomial", data = liz)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.607  -0.638   0.237   0.433   2.099
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.606      1.695    2.13    0.033 *
## PARATIO       -0.220      0.101   -2.18    0.029 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 26.287  on 18  degrees of freedom
## Residual deviance: 14.221  on 17  degrees of freedom
## AIC: 18.22
##
## Number of Fisher Scoring iterations: 6
```

В результатах появились новые термины:

z value,
Null deviance,
Residual deviance,
AIC.

Z value – величина критерия Вальда (Wald statistic) – аналог t-критерия. Используется для проверки $H_0: \beta_1 = 0$:

$$z = \frac{b_1}{SE_{b_1}}.$$

Сравнивают со стандартизованным нормальным распределением (z-распределение).

Дает надежные оценки p-value при больших выборках.

Более надежные результаты дает тест отношения правдоподобий (Likelihood ratio test):

$$LR = 2\ln\left(\frac{Lik_l}{Lik_s}\right) = 2(logLik_l - logLik_s).$$

Null deviance и Residual deviance

• *«Насыщенная» модель* – модель, подразумевающая, что каждая из n точек имеет свой собственный параметр, следовательно, надо подобрать n параметров. Вероятность существования данных для такой модели равна 1. $logLik_{satur} = ln(1) = 0$.

• *«Нулевая» модель* – модель подразумевающая, что для описания всех точек надо подобрать только 1 параметр. Эта модель включает только свободный член $g(x) = \beta_0$. У этой модели есть значение $logLik_{nul}$.

• *«Предложенная» модель* – модель, подобранная в нашем анализе $g(x) = \beta_0 + \beta_1 x$ для нее есть $logLik_{prop}$.

Девианса – это оценка отклонения логарифма максимального правдоподобия одной модели от логарифма максимального правдоподобия другой модели.

Остаточная девианса: $2(logLik_{satur} - logLik_{prop}) = -2logLik_{prop}$.

Нулевая девианса: $2(logLik_{satur} - logLik_{nul}) = -2logLik_{nul}$.

Проверим

```
(Dev_resid <- -2*as.numeric(logLik(liz_model))) #Остаточная девианса
```

```
## [1] 14.22
```

```
(Dev_nul <- -2*as.numeric(logLik(update(liz_model, ~-PARATIO)))) #Нулевая девианса
```

```
## [1] 26.29
```

По соотношению нулевой девиансы и остаточной девиансы можно понять насколько хороша модель:

$$G^2 = -2\big(logLik_{nul} - logLik_{prop}\big).$$

```
(G2 <- Dev_nul - Dev_resid)
## [1] 12.07
```

• G^2 – это девианса полной и редуцированной модели;
• G^2 – аналог $SS_{residuals}$ в обычном регрессионном анализе;
• G^2 – подчиняется χ^2 распределению (с параметом df = 1), если нулевая модель и предложенная модель не отличаются друг от друга;
• G^2 можно использовать для проверки нулевой гипотезы: $H_0: \beta_1 = 0$.

Анализ девианса

Это аналог дисперсионного анализа для сравния полной и редуцированной модели

```
anova(liz_model, test="Chi")
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: PA
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                        18       26.3
## PARATIO  1     12.1         17       14.2  0.00051 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

AIC – Информационный критерий Акаике (Akaike Information Criterion):

$$AIC = -2logLik_{prop} + 2p,$$

где $logLik_{prop}$ – логарифм функции правдоподобия для предложенной модели;

$2p$ – штраф за введение в модель p параметров.

Задание

Расчитайте вручную критерий Акаике для нашей модели

Решение

```
(2*2 - 2*as.numeric(logLik(liz_model)))
## [1] 18.22
#или

AIC(liz_model)
## [1] 18.22
```

Как трактовать коэффициенты подобранной модели?

$$g(x) = \ln(\frac{\pi(x)}{1-\pi(x)}) = \beta_0 + \beta_1 x$$

```
coef(liz_model)
## (Intercept)     PARATIO
##      3.6061     -0.2196
```

• β_0 – не имеет особого смысла, просто поправочный коэффициент;

• β_1 – *на сколько* единиц изменяется *логарифм* величины шансов (odds) при изменении на одну единицу значения предиктора. Трактовать такую величину неудобно и трудно. Хотя по знаку β_1 можно сазу понять, возрастает ли вероятность или снижается.

Немного алгебры

Посмотрим, как изменится $g(x) = \ln(\frac{\pi(x)}{1-\pi(x)})$ при изменении предиктора на 1

$$g(x+1) - g(x) = ln(odds_{x+1}) - ln(odds_x) = ln\left(\frac{odds_{x+1}}{odds_x}\right).$$

Задание

Завершите алгебраическое преобразование.

Решение

$$ln\left(\frac{odds_{x+1}}{odds_x}\right) = \beta_0 + \beta_1(x+1) - \beta_0 - \beta_1 x = \beta_1,$$

$$ln\left(\frac{odds_{x+1}}{odds_x}\right) = \beta_1,$$

$$\frac{odds_{x+1}}{odds_x} = e^{\beta_1}$$

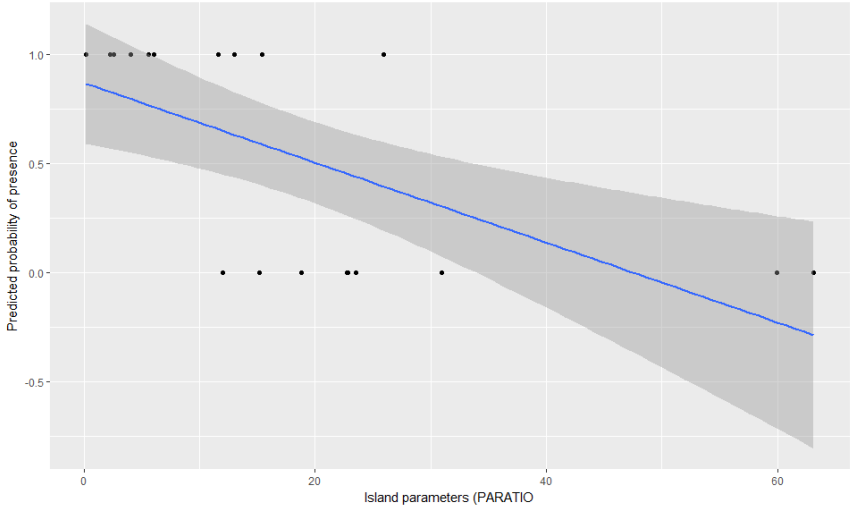
Полученная величина имеет определенный смысл и называтся отношением шансов (odds ratio)

```
exp(coef(liz_model)[2])
## PARATIO
##  0.8029
```

Отношение шансов (odds ratio) показывает, *во сколько* раз изменяется отношение шансов встретить ящерицу при увеличении отношения периметра острова к его площади на одну единицу.

Отношение шансов изменяется в 0,8029 раза. То есть, чем меньше остров, тем меньше шансов встретить ящерицу

Подобранные коэффициенты позволяют построить логистическую кривую.



Эта кривая описывается следующей формулой:

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}.$$

Кривая имеет доверительный интервал (серая область).

Задание

Найдите границы 95 % доверительного интервала для параметров построенной модели и для отношения шансов.

Решение

Доверительные интервалы для коэффициентов:

```
confint(liz_model) # для логитов
##                2.5 %   97.5 %
## (Intercept)  1.0059  8.04214
## PARATIO     -0.4849 -0.06649
exp(confint(liz_model)) # для отношения шансов
##                2.5 %    97.5 %
## (Intercept) 2.7345 3109.2748
## PARATIO     0.6157    0.9357
```

Множественная логистическая регрессия

От чего зависит уровень смертности пациентов, выписанных из реанимации?

Данные, полученные на основе изучения 200 историй болезни пациентов, одного из американских госпиталей.

STA: Статус (0 = Выжил, 1 = умер)

AGE: Возраст

SEX: Пол

RACE: Раса

SER: Тип мероприятий в реанимации (Medical, 1 = Surgical)

CAN: Присутствует ли онкология? (No,Yes)

CRN: Присутсвует ли почечная недостаточность (No, Yes)

INF: Наличие инфекции при госпитализации (No, Yes)

CPR: Была ли сердечно-легочная реанимация (No, Yes)

SYS: Давление (in mm Hg)

HRA: ЧСС (beats/min)

PRE: Были ли случаи реанимации ранее (0 = No, 1 = Yes)

TYP: Тип госпитализации (Elective, Emergency)

FRA: Есть ли случаи переломов костей (No, Yes)

PO2: Концентрация кислорода в крови (1 = >60, 2 = 60)

PH: PH from initial blood gases (1 = 7.25, 2 <7.25)

PCO: Концентрация углекислого газа в крови (1 = 45, 2 = >45)

BIC: Bicarbonate from initial blood gases (1 = 18, 2 = <18)

CRE: Концентрация кретинина в крови (1 = 2.0, 2 = >2.0)

LOC: Был ли пациент в сознании при госпитализации (1 = no coma or stupor, 2= deep stupor, 3 = coma)

```
     # Смотрим на данные
     r surviv <- read.table("ICU.csv", header=TRUE, sep=";")
head(surviv)
     ##   STA AGE    SEX   RAC      SER CAN CRN INF CPR SYS HRA
PRE       TYP FRA PO2 PH PCO BIC ## 1   0  27 Female White  Medi-
cal  No  No Yes  No 142  88  No Emergency  No   1  1   1   1 ## 2
0  59   Male White  Medical  No  No  No  No 112  80 Yes Emergency
No   1  1   1   1 ## 3   0  77   Male White Surgical  No  No  No
No 100  70  No  Elective  No   1  1   1   1 ## 4   0  54   Male
White  Medical  No  No Yes  No 142 103  No Emergency Yes   1  1
1   1 ## 5   0  87 Female White Surgical  No  No Yes  No 110 154
Yes Emergency  No   1  1   1   1 ## 6   0  69   Male White  Medi-
cal  No  No Yes  No 110 132  No Emergency  No   2  1   1   2 ##
CRE LOC ## 1   1   1 ## 2   1   1 ## 3   1   1 ## 4   1   1 ## 5
1   1 ## 6   1   1
```

Строим модель

```
surv_model <- glm(STA ~ . , family = "binomial", data = surviv)
summary(surv_model)
##
## Call:
## glm(formula = STA ~ ., family = "binomial", data = surviv)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.805  -0.603  -0.235  -0.118   2.870
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.35382    2.88759   -2.89  0.00382 **
## AGE           0.05110    0.01717    2.98  0.00292 **
## SEXMale       0.55306    0.50158    1.10  0.27019
## RACOther      1.21408    1.53381    0.79  0.42862
## RACWhite      0.80676    1.16724    0.69  0.48946
## SERSurgical  -0.63886    0.59225   -1.08  0.28073
## CANYes        2.75033    0.97762    2.81  0.00490 **
## CRNYes       -0.11002    0.77638   -0.14  0.88731
## INFYes       -0.07443    0.53292   -0.14  0.88892
## CPRYes        0.93338    0.99234    0.94  0.34692
## SYS          -0.01089    0.00798   -1.37  0.17202
## HRA          -0.00254    0.00951   -0.27  0.78941
## PREYes        0.97765    0.63549    1.54  0.12395
## TYPEmergency  2.67401    0.99546    2.69  0.00723 **
```

```
## FRAYes        1.25685    1.00964    1.24  0.21318
## PO2           0.27541    0.85580    0.32  0.74759
## PH            2.37703    1.23187    1.93  0.05365 .
## PCO          -3.14692    1.37961   -2.28  0.02255 *
## BIC          -0.76518    0.91314   -0.84  0.40205
## CRE           0.14024    1.07119    0.13  0.89584
## LOC           2.72308    0.76014    3.58  0.00034 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 200.16  on 199  degrees of freedom
## Residual deviance: 128.85  on 179  degrees of freedom
## AIC: 170.9
##
## Number of Fisher Scoring iterations: 6
```

Проведем анализ девианса

```
anova(surv_model, test="Chi")
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: STA
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                   199        200
## AGE   1     7.85       198        192   0.00507 **
## SEX   1     0.00       197        192   0.97572
## RAC   2     1.26       195        191   0.53364
## SER   1     8.32       194        183   0.00392 **
## CAN   1     0.72       193        182   0.39726
## CRN   1     2.62       192        179   0.10559
## INF   1     2.03       191        177   0.15391
## CPR   1     3.92       190        173   0.04777 *
## SYS   1     5.65       189        168   0.01741 *
## HRA   1     0.79       188        167   0.37537
## PRE   1     0.91       187        166   0.33955
## TYP   1    12.79       186        153   0.00035 ***
## FRA   1     0.48       185        153   0.48914
```

```
## PO2   1     0.03       184        153   0.87152
## PH    1     0.02       183        153   0.90183
## PCO   1     1.31       182        152   0.25312
## BIC   1     0.07       181        151   0.78999
## CRE   1     0.23       180        151   0.63282
## LOC   1    22.32       179        129 0.0000023 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Строим сокращенную модель

```
surv_model_reduced <- glm(STA ~ AGE + TYP + PH + PCO + LOC,
family = "binomial", data = surviv)

anova(surv_model, surv_model_reduced, test = "Chi")
## Analysis of Deviance Table
##
## Model 1: STA ~ AGE + SEX + RAC + SER + CAN + CRN + INF +
CPR + SYS + HRA +
##     PRE + TYP + FRA + PO2 + PH + PCO + BIC + CRE + LOC
## Model 2: STA ~ AGE + TYP + PH + PCO + LOC
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1       179        129
## 2       194        145 -15    -16.6     0.35
```

Посмотримм на AIC полной и сокращенной моделей

```
AIC(surv_model, surv_model_reduced)
##                     df   AIC
## surv_model          21 170.9
## surv_model_reduced   6 157.4
```

Предпочтение отдается той модели, у которой AIC меньше.

```
summary(surv_model_reduced)
##
## Call:
## glm(formula = STA ~ AGE + TYP + PH + PCO + LOC, family = "bino-
mial",
##     data = surviv)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.086  -0.657  -0.331  -0.194   2.461
##
```

```
## Coefficients:
##              Estimate Std. Error z value   Pr(>|z|)
## (Intercept)   -7.5883     1.5010   -5.06 0.00000043 ***
## AGE            0.0362     0.0124    2.93     0.0034 **
## TYPEmergency   2.1275     0.7722    2.76     0.0059 **
## PH             1.6480     0.8403    1.96     0.0499 *
## PCO           -1.9096     0.9702   -1.97     0.0490 *
## LOC            2.1545     0.5478    3.93 0.00008379 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 200.16  on 199  degrees of freedom
## Residual deviance: 145.43  on 194  degrees of freedom
## AIC: 157.4
##
## Number of Fisher Scoring iterations: 6
```

Задание

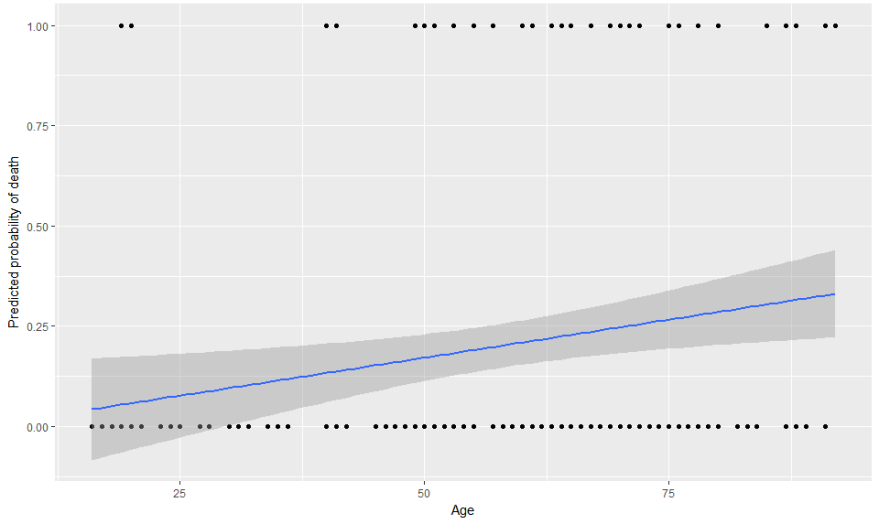
Во сколько раз изменяется отношение шансов при условии, что пациента госпитализировали в экстренном порядке?

Вам понадобится переменная *TYP*

Решение

```
exp(coef(surv_model_reduced)[3])
## TYPEmergency
##        8.393
```

Как изменяется вероятность смерти после реанимации в зависимости от возраста?



Следует отметить, что:

• для моделирования бинарных данных применяется логистическая регрессия;

• оценка коэффициентов логистической регресси производится методом максимального правдоподобия;

• для проверки нулевой гипотезы о наличии зависимости применяют тест отношений правдоподобия.

Список библиографических источников

Основной

1. *Горяинова, Е. Р.* Прикладные методы анализа статистических данных: учеб. пособие / Е. Р. Горяинова, А. Р. Панков, Е. Н. Платонов. – М.: Изд. дом Высшей школы экономики, 2012. – 310 с.

2. *Лесковец, Ю.* Анализ больших наборов данных / Ю. Лесковец, А. Раджараман. – М.: ДМК, 2016. – 498 с.

3. *Крянев, А. В.* Метрический анализ и обработка данных / А. В. Крянев, Г. В. Лукин, Д. К. Удумян. – М.: Физматлит, 2012. – 308 с.

4. *Кулаичев, А. П.* Методы и средства комплексного анализа данных: учеб. пособие / А. П. Кулаичев. – М.: Форум, НИЦ ИНФРА-М, 2013. – 512 с.

5. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

6. 1Biostatistics with R: An Introduction to Statistics Through Biological Data (Use R!) / Babak Shahbaba, 2012.

7. Modern Epidemiology Third, Mid-cycle revision Edition / J. Kenneth Rothman, L. Timothy Lash Associate Professor, Sander Greenland, 2012.

8. Biostatistics and Epidemiology: A Primer for Health and Biomedical Professionals 4th ed. / Sylvia Wassertheil-Smoller, Jordan Smoller, 2015.

9. Introductory Time Series with R (Use R!) / Paul S. P. Cowpertwait, Andrew V. Metcalfe, 2009.

Дополнительный

10. *Новикова, Н. М.* Статистические методы в биологии и медицине: учеб.-метод. пособие / Н. М. Новикова. – Минск: МГЭУ им. А. Д. Сахарова, 2009. – 88 с.

11. *Кабаков, Р.* R в действии. Анализ и визуализация данных в программе R / Р. Кабаков. – М.: ДМК, 2016. – 588 с.

12. Биометрика – журнал для медиков и биологов, сторонников доказательной биомедицины [Электронный ресурс]. URL: http://www.-biometrica.tomsk.ru.

13. *Орлов, А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

14. *Чиркин, А. А.* Клинический анализ лабораторных данных / А. А. Чиркин. – Витебск: Медицинская литература, 2008. – 384 с.

15. *Петри, А.* Наглядная статистика в медицине / А. Петри, К. Сэбин. – М.: Изд. дом ГЭОТАР-МЕД, 2003 – 143 с.

Лабораторная работа № 11
Многомерные данные: дискриминантный анализ (деревья решений)

Цель: научиться приемам агрегации данных, которую можно осуществлять при помощи алгоритма деревьев решений.

Деревья решений

Метод деревьев решений (*decision trees*) является одним из наиболее популярных для решения задач классификации и прогнозирования. Иногда его (*Data Mining*) также называют деревьями решающих *правил*, деревьями классификации и регрессии.

Как видно из последнего названия, при помощи данного метода решаются задачи классификации и прогнозирования.

Если зависимая, целевая *переменная* принимает дискретные значения, при помощи метода дерева решений решается задача классификации.

Если же зависимая *переменная* принимает непрерывные значения, то *дерево* решений устанавливает зависимость этой переменной от независимых переменных – решает задачу численного прогнозирования.

Впервые деревья решений были предложены Ховилендом и Хантом (Hoveland, Hunt) в конце 50-х гг. XX в. Самая ранняя и известная работа Ханта и др., в которой излагается суть деревьев решений («Эксперименты в индукции» («Experiments in *Induction*»)) была опубликована в 1966 г.

В наиболее простом виде *дерево* решений – это способ представления *правил* в иерархической, последовательной структуре. Основа такой структуры – ответы «Да» или «Нет» на ряд вопросов.

На рис. 6.1 приведен пример дерева решений, задача которого – ответить на вопрос: «Играть ли в гольф?» Чтобы решить задачу, принять *решение, играть* ли в гольф, следует отнести текущую ситуацию к одному из известных классов (в данном случае – «играть» или «не играть»). Для этого требуется ответить на ряд вопросов, которые находятся в узлах этого дерева, начиная с его корня.

Первый узел нашего дерева «Солнечно?» является ***узлом проверки***, то есть условием. При положительном ответе на вопрос осуществляется переход к левой части дерева, называемой левой ***ветвью***, при отрицательном – к правой части дерева. Таким образом, ***внутренний узел*** дерева является ***узлом проверки*** определенного условия. Далее идет следующий вопрос и т. д., пока не будет достигнут ***конечный узел*** дерева, являющийся *узлом решения*. Для нашего дерева существует два типа *конечного узла*: «играть» и «не играть» в гольф.

В результате прохождения от корня дерева (иногда называемого корневой вершиной) до его вершины решается задача классификации, то есть выбирается один из классов – «играть» и «не играть» в гольф.

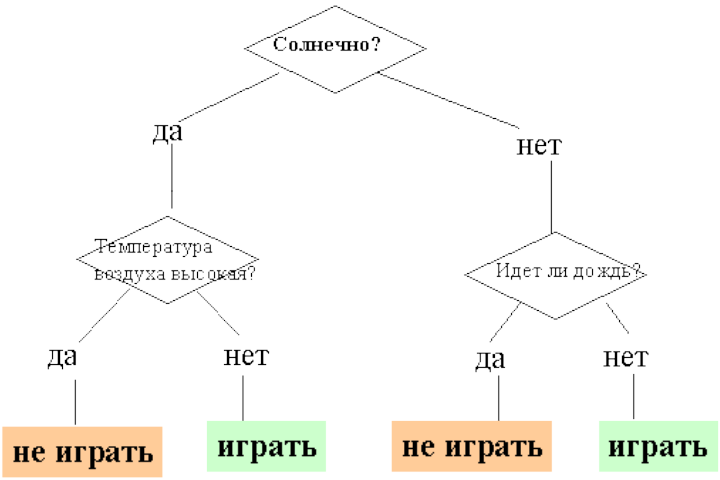


Рис. 6.1. Дерево решений «Играть ли в гольф?»

Цель построения дерева решения в нашем случае – это *определение* значения категориальной зависимой переменной.

Итак, для нашей задачи основными элементами дерева решений являются:

Корень дерева: «Солнечно?»

Внутренний узел дерева или *узел проверки*: «Температура воздуха высокая?», «Идет ли дождь?»

Лист, *конечный узел* дерева, *узел решения* или *вершина*: «Играть», «Не играть»

Ветвь дерева (случаи ответа): «Да», «Нет».

В рассмотренном примере решается задача *бинарной классификации*, то есть создается дихотомическая классификационная модель. Пример демонстрирует работу так называемых бинарных деревьев.

В узлах бинарных деревьев *ветвление* может вестись только в двух направлениях: существует возможность только двух ответов на поставленный вопрос («да» и «нет»).

Бинарные деревья являются самым простым, частным случаем деревьев решений. В остальных случаях ответов и, соответственно, *ветвей* дерева, выходящих из его *внутреннего узла*, может быть больше двух.

Каждая *ветвь* дерева, идущая от *внутреннего узла*, отмечена *предикатом расщепления*. Последний может относиться лишь к одному

атрибуту расщепления данного узла. Характерная особенность *предикатов расщепления*: каждая *запись* использует уникальный *путь* от корня дерева только к одному узлу-решению. Объединенная *информация* об *атрибутах расщепления* и *предикатах расщепления* в узле называется ***критерием расщепления*** (splitting *criterion*).

Таким образом, для данной задачи (как и для любой другой) может быть построено множество деревьев решений различного качества, с различной прогнозирующей точностью.

Качество построенного дерева решения весьма зависит от правильного выбора *критерия расщепления*. Над разработкой и усовершенствованием критериев работают многие исследователи.

Метод деревьев решений часто называют «наивным» подходом. Но благодаря целому ряду преимуществ, данный метод является одним из наиболее популярных для решения задач классификации.

Преимущества деревьев решений

Интуитивность деревьев решений. Классификационная модель, представленная в виде дерева решений, является интуитивной и упрощает понимание решаемой задачи. Результат работы алгоритмов конструирования деревьев решений, в отличие, например, от нейронных сетей, представляющих собой «черные ящики», легко интерпретируется пользователем. Это свойство деревьев решений не только важно при отнесении к определенному классу нового объекта, но и полезно при интерпретации модели классификации в целом. *Дерево* решений позволяет понять и объяснить, почему конкретный *объект* относится к тому или иному классу.

Деревья решений дают возможность извлекать *правила* из *базы данных* на **естественном языке**. Пример *правила*: Если Возраст > 35 и Доход > 200, то выдать *кредит*.

Деревья решений позволяют создавать классификационные модели в тех областях, где аналитику достаточно сложно формализовать знания.

Алгоритм конструирования дерева решений **не требует от пользователя выбора входных атрибутов** (независимых переменных). На *вход алгоритма* можно подавать все существующие атрибуты, *алгоритм* сам выберет наиболее значимые среди них, и только они будут использованы для построения дерева. В сравнении, например, с нейронными сетями, это значительно облегчает пользователю работу, поскольку в нейронных сетях выбор количества входных атрибутов существенно влияет на время обучения.

Точность моделей, созданных при помощи деревьев решений, сопоставима с другими методами построения классификационных моделей (*статистические методы*, нейронные сети).

Разработан ряд **масштабируемых алгоритмов**, которые могут быть использованы для построения деревьев решения на сверхбольших базах данных; *масштабируемость* здесь означает, что с ростом числа примеров или записей *базы данных* время, затрачиваемое на обучение, то есть построение деревьев решений, растет линейно. Примеры таких алгоритмов: SLIQ, SPRINT.

Быстрый процесс обучения. На построение классификационных моделей при помощи алгоритмов конструирования деревьев решений требуется значительно меньше времени, чем, например, на обучение нейронных сетей.

Большинство алгоритмов конструирования деревьев решений имеют возможность специальной обработки **пропущенных значений.**

Многие классические *статистические методы*, при помощи которых решаются задачи классификации, могут работать только с числовыми данными, в то время как деревья решений работают и с числовыми, и с **категориальными** типами данных.

Многие *статистические методы* являются параметрическими, и *пользователь* должен заранее владеть определенной информацией, например, знать вид модели, иметь гипотезу о виде зависимости между переменными, предполагать, какой вид распределения имеют данные. Деревья решений, в отличие от таких методов, строят непараметрические модели. Таким образом, деревья решений способны решать такие задачи *Data Mining*, в которых отсутствует априорная *информация* о виде зависимости между исследуемыми данными.

Процесс конструирования дерева решений

Напомним, что рассматриваемая нами задача классификации относится к стратегии *обучения с учителем*, иногда называемого индуктивным обучением. В этих случаях все объекты тренировочного набора данных заранее отнесены к одному из предопределенных классов.

Алгоритмы конструирования деревьев решений состоят из этапов «построение» или «*создание*» дерева (*tree* building) и «*сокращение*» дерева (*tree pruning*). В ходе *создания* дерева решаются вопросы выбора *критерия расщепления* и остановки обучения (если это предусмотрено алгоритмом). В ходе этапа *сокращения* дерева решается вопрос отсечения некоторых его *ветвей*.

Рассмотрим, как подобные задачи можно решать с помощью R. Будем использовать пакет `partykit`.

```
library(partykit)
## Loading required package: grid
```

Построение классификационных деревьев

Работаем сегодня с данными Carseats из пакета.

```
library(ISLR)
?ISLR::Carseats
## starting httpd help server ...
##  done
```

К какому типу переменных относится Sales?

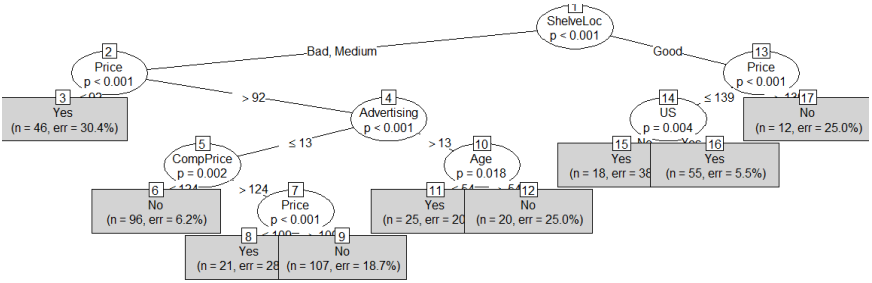
Поскольку классификационные деревья предсказывают принадлежность к той или иной категории, то изменим немного наши данные.

```
carseats                    <-                    ISLR::Carseats
High=ifelse(carseats$Sales>8,"Yes","No")
library(dplyr)
##
## Attaching package: 'dplyr'
##  The  following  objects  are  masked  from  'package:stats':
##
##     filter, lag
##  The  following  objects  are  masked  from  'package:base':
##
##     intersect, setdiff, setequal, union
Carseats         <-         data.frame(carseats,         High)
Carseats <- dplyr::select(Carseats, -Sales)
```

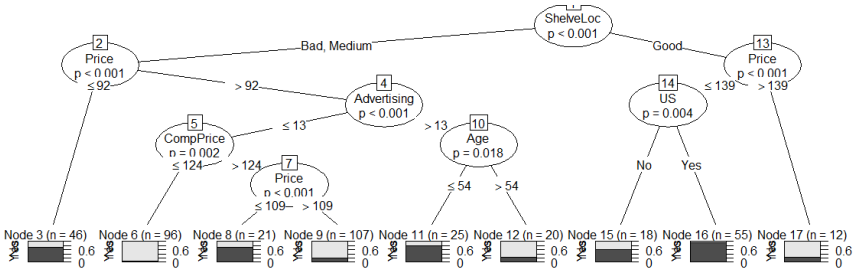
Построим дерево.

```
tree.carseats <- ctree(High~., data = Carseats)
tree.carseats
##
## Model formula:
## High ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education + Urban + US
##
## Fitted party:
## [1] root
## |   [2] ShelveLoc in Bad, Medium
## |   |   [3] Price <= 92: Yes (n = 46, err = 30.4%)
## |   |   [4] Price > 92
## |   |   |   [5] Advertising <= 13
## |   |   |   |   [6] CompPrice <= 124: No (n = 96, err = 6.2%)
## |   |   |   |   [7] CompPrice > 124
## |   |   |   |   |   [8] Price <= 109: Yes (n = 21, err = 28.6%)
```

```
## |   |   |   |   |   [9] Price > 109: No (n = 107, err = 18.7%)
## |   |   |   [10] Advertising > 13
## |   |   |   |   [11] Age <= 54: Yes (n = 25, err = 20.0%)
## |   |   |   |   [12] Age > 54: No (n = 20, err = 25.0%)
## |   [13] ShelveLoc in Good
## |   |   [14] Price <= 139
## |   |   |   [15] US in No: Yes (n = 18, err = 38.9%)
## |   |   |   [16] US in Yes: Yes (n = 55, err = 5.5%)
## |   |   [17] Price > 139: No (n = 12, err = 25.0%)
##
## Number of inner nodes:    8
## Number of terminal nodes: 9
plot(tree.carseats, type = "simple")
```




```
plot(tree.carseats)
```



– Можете ли вы «прочитать» это дерево?

– Какие «правила» получились?

– Все ли переменные из данных использованы?

Но насколько хорошо такие правила описывают закономерности?

Разделим данные на две части: на одной будем строить модель, на другой – проверять.

– Почему бы не использовать полный датасет на всех этапах?

```
set.seed(2)
train=sample(1:nrow(Carseats), 200)
Carseats.test <- Carseats[-train,]
Carseats.train <- Carseats[train,]
tree.carseats=ctree(High~.,Carseats.train)
```

Итак, согласно этой модели, предскажем значения (высокие продажи / низкие продажи) и посмотрим, совпали ли наши предсказания с фактическими значениями.

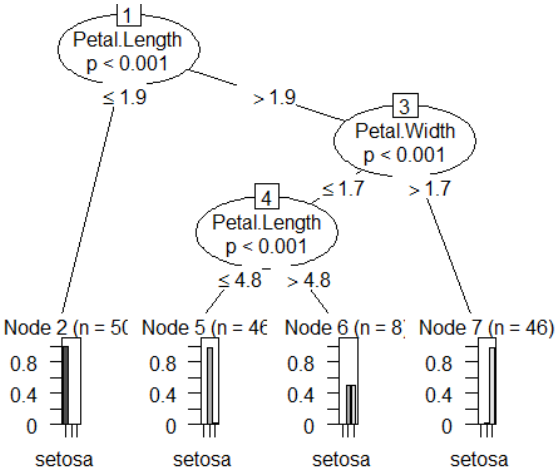
```
tree.pred <- predict(tree.carseats, Carseats.test)
tree.pred
##  1  2  5  9  10  12  13  15  16  18  21  25  28  30  33  34  36  39
##  No Yes  No  No No Yes No Yes  No Yes  No  No  No  No Yes Yes No No
## 40  43  45  48  52  53  54  58  62  63  65  66  68  72  75  76  77  80
##  No Yes  No  No  No  No  No  No  No  No  No  No Yes Yes  No  No Yes  No
##  81  82  83  86  89  95  97  99 100 101 109 110 111 112 115 116 117 119
## Yes Yes Yes  No  No Yes Yes Yes  No  No  No  No  No  No  No  No  No  No
## 120 122 127 129 132 135 136 137 138 139 140 143 144 146 148 149 150 151
##  No  No Yes  No  No  No Yes  No  No  No  No  No  No  No Yes  No  No Yes
## 152 156 158 159 161 162 164 166 167 168 169 171 172 174 175 178 181 182
## Yes Yes  No Yes  No  No  No  No Yes  No  No  No Yes  No  No  No  No  No
## 183 185 186 188 190 191 193 194 195 196 197 198 199 200 201 205 206 207
##  No  No  No  No Yes  No  No Yes Yes  No  No  No  No  No  No  No  No  No
## 211 214 217 218 219 221 222 224 227 230 231 233 234 235 236 239 240 244
##  No  No  No  No  No Yes  No  No Yes Yes  No Yes Yes Yes  No Yes  No  No
## 245 249 250 251 255 256 257 260 261 263 264 267 268 272 274 276 277 278
##  No  No  No  No Yes Yes  No  No  No Yes  No Yes  No  No Yes  No Yes  No
## 279 282 287 288 290 291 292 294 295 297 298 300 304 307 309 310 311 312
##  No Yes  No Yes Yes Yes  No Yes Yes Yes  No Yes Yes  No Yes Yes Yes  No
## 313 314 316 320 321 324 326 330 331 332 333 335 336 338 342 343 344 348
##  No Yes  No Yes  No Yes  No Yes  No Yes Yes Yes Yes  No  No  No  No Yes
## 349 350 351 353 355 360 361 364 367 368 371 372 376 377 383 385 386 389
## Yes Yes Yes Yes  No  No Yes Yes  No Yes  No  No  No Yes Yes Yes  No Yes
## 394 395
##  No  No
## Levels: No Yes
table(tree.pred, Carseats.test$High) ##confusion table
##
## tree.pred No Yes
##       No  97  30
##       Yes 19  54
(97+54)/200
## [1] 0.755
```

Посмотрим на данные про сорта ирисов.

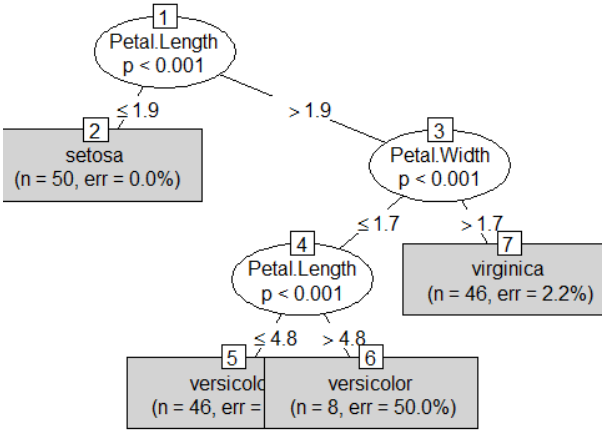
```
data(iris)
head(iris)
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
?iris
## starting httpd help server ...
##  done
```

Предскажем сорт по остальным характеристикам.

```
iris.tree <- ctree(Species ~ ., data = iris)
plot(iris.tree)
```




```
plot(iris.tree, type = "simple")
```



Качество предсказания – confusion table.

```
predictedSpecies <- predict(iris.tree, iris)
table(predictedSpecies, iris$Species)
##
## predictedSpecies setosa versicolor virginica
##       setosa         50          0         0
##       versicolor      0         49         5
##       virginica       0          1        45
```

– Какая доля ирисов virginica предсказана правильно?

– Какая доля из ирисов, отнесенных к vercicolor, действительно относится к этому сорту?

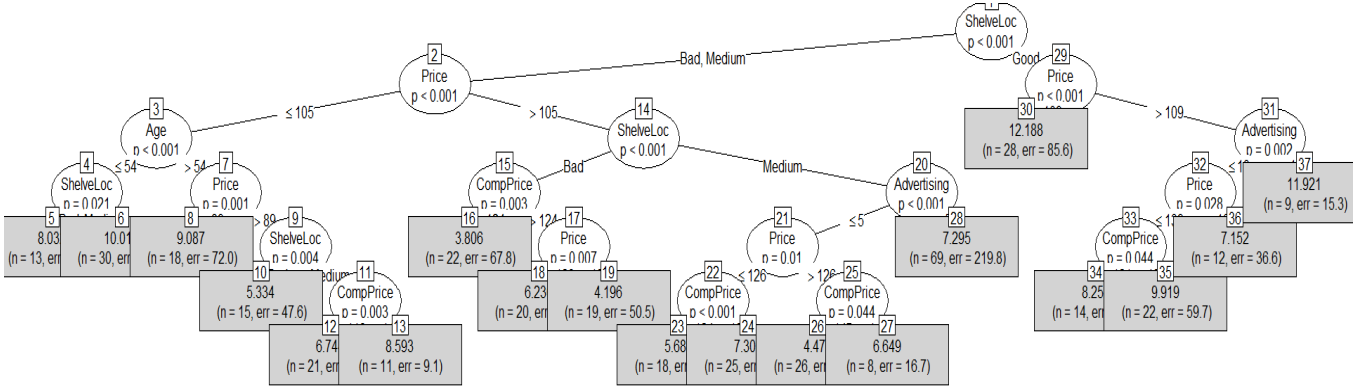
– Сколько случаев предсказаны правильно?

– Какова точность предсказания?

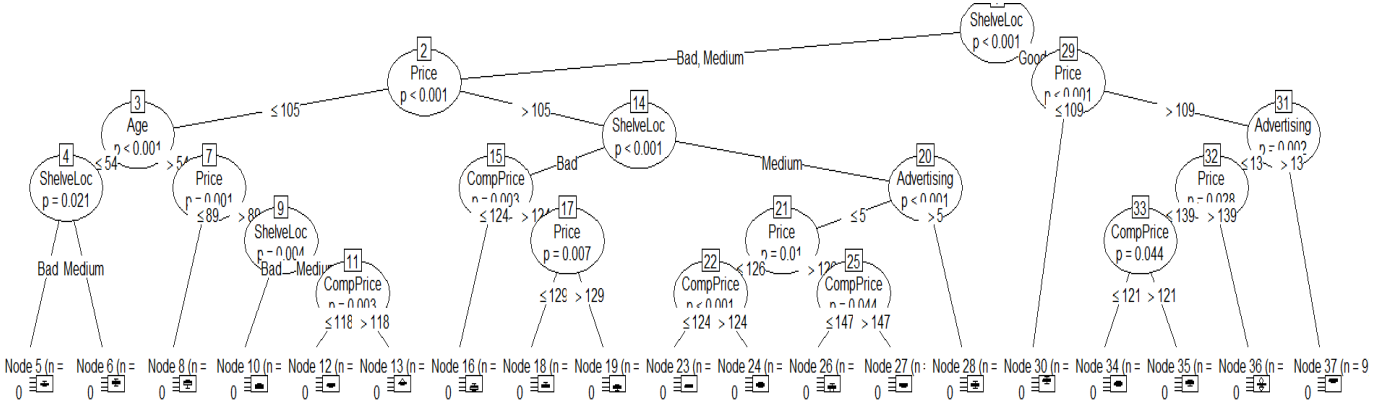
Построение регрессионных деревьев

Поменяем немного задачу: не будем делить на высокие и низкие продажи, а просто попробуем предсказать уровень продаж Sales

```
carseats <- ISLR::Carseats
tree.regr <- ctree(Sales~., data = carseats)
plot(tree.regr, type = "simple")
```


1
ShelveLoc
p < 0.001
Bad, Medium
Good
2
Price
p < 0.001
29
Price
p < 0.001
≤ 105
> 105
> 109
3
Age
p < 0.001
14
ShelveLoc
p < 0.001
30
12.188
(n = 28, err = 85.6)
31
Advertising
p = 0.002
4
ShelveLoc
p = 0.021
≤ 54
> 54
7
Price
p = 0.001
15
CompPrice
p = 0.003
Bad
Medium
20
Advertising
p < 0.001
32
Price
p = 0.028
37
11.921
(n = 9, err = 15.3)
5
8.03
(n = 13, err
6
10.01
(n = 30, err
8
9.087
(n = 18, err = 72.0)
> 89
9
ShelveLoc
p = 0.004
16
3.806
(n = 22, err = 67.8)
> 124
17
Price
p = 0.007
21
Price
p = 0.01
≤ 5
28
7.295
(n = 69, err = 219.8)
33
CompPrice
p = 0.044
36
7.152
(n = 12, err = 36.6)
10
5.334
(n = 15, err = 47.6)
11
CompPrice
p = 0.003
18
6.23
(n = 20, err
19
4.196
(n = 19, err = 50.5)
22
CompPrice
p < 0.001
≤ 126
> 126
25
CompPrice
p = 0.044
34
8.25
(n = 14, err
35
9.919
(n = 22, err = 59.7)
12
6.74
(n = 21, err
13
8.593
(n = 11, err = 9.1)
23
5.68
(n = 18, err
24
7.30
(n = 25, err
26
4.47
(n = 26, err
27
6.649
(n = 8, err = 16.7)

```
tree.regr
##
## Model formula:
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education + Urban + US
##
## Fitted party:
## [1] root
## |  [2] ShelveLoc in Bad, Medium
## |  |  [3] Price <= 105
## |  |  |  [4] Age <= 54
## |  |  |  |  [5] ShelveLoc in Bad: 8.032 (n = 13, err = 27.5)
## |  |  |  |  [6] ShelveLoc in Medium: 10.011 (n = 30, err = 95.7)
## |  |  |  [7] Age > 54
## |  |  |  |  [8] Price <= 89: 9.087 (n = 18, err = 72.0)
## |  |  |  |  [9] Price > 89
## |  |  |  |  |  [10] ShelveLoc in Bad: 5.334 (n = 15, err = 47.6)
## |  |  |  |  |  [11] ShelveLoc in Medium
## |  |  |  |  |  |  [12] CompPrice <= 118: 6.744 (n = 21, err = 34.5)
## |  |  |  |  |  |  [13] CompPrice > 118: 8.593 (n = 11, err = 9.1)
## |  |  [14] Price > 105
## |  |  |  [15] ShelveLoc in Bad
## |  |  |  |  [16] CompPrice <= 124: 3.806 (n = 22, err = 67.8)
## |  |  |  |  [17] CompPrice > 124
## |  |  |  |  |  [18] Price <= 129: 6.230 (n = 20, err = 53.4)
## |  |  |  |  |  [19] Price > 129: 4.196 (n = 19, err = 50.5)
## |  |  |  [20] ShelveLoc in Medium
## |  |  |  |  [21] Advertising <= 5
## |  |  |  |  |  [22] Price <= 126
## |  |  |  |  |  |  [23] CompPrice <= 124: 5.684 (n = 18, err = 16.7)
## |  |  |  |  |  |  [24] CompPrice > 124: 7.305 (n = 25, err = 51.3)
## |  |  |  |  |  [25] Price > 126
## |  |  |  |  |  |  [26] CompPrice <= 147: 4.475 (n = 26, err = 87.9)
## |  |  |  |  |  |  [27] CompPrice > 147: 6.649 (n = 8, err = 16.7)
## |  |  |  |  [28] Advertising > 5: 7.295 (n = 69, err = 219.8)
## |  [29] ShelveLoc in Good
## |  |  [30] Price <= 109: 12.188 (n = 28, err = 85.6)
## |  |  [31] Price > 109
## |  |  |  [32] Advertising <= 13
## |  |  |  |  [33] Price <= 139
## |  |  |  |  |  [34] CompPrice <= 121: 8.256 (n = 14, err = 25.0)
## |  |  |  |  |  [35] CompPrice > 121: 9.919 (n = 22, err = 59.7)
## |  |  |  |  [36] Price > 139: 7.152 (n = 12, err = 36.6)
## |  |  |  [37] Advertising > 13: 11.921 (n = 9, err = 15.3)
##
## Number of inner nodes:    18
## Number of terminal nodes: 19
plot(tree.regr)
```


ShelveLoc
p < 0.001
Bad, Medium
Good
2
Price
p < 0.001
29
Price
p < 0.001
≤ 105
> 105
≤ 109
> 109
3
Age
p < 0.001
14
ShelveLoc
p < 0.001
31
Advertising
p = 0.002
≤ 54
> 54
Bad
Medium
≤ 13
> 13
4
ShelveLoc
p = 0.021
7
Price
p = 0.001
15
CompPrice
p = 0.003
17
Price
p = 0.007
20
Advertising
p < 0.001
21
Price
p = 0.01
32
Price
p = 0.028
33
CompPrice
p = 0.044
≤ 89
> 89
≤ 124
> 124
≤ 5
> 5
≤ 126
≤ 139
> 139
≤ 121
> 121
Bad
Medium
9
ShelveLoc
p = 0.004
11
CompPrice
p = 0.003
≤ 129
> 129
22
CompPrice
p < 0.001
25
CompPrice
p = 0.044
≤ 124
> 124
≤ 147
> 147
≤ 118
> 118
Node 5 (n = 0
Node 6 (n = 0
Node 8 (n = 0
Node 10 (n = 0
Node 12 (n = 0
Node 13 (n = 0
Node 16 (n = 0
Node 18 (n = 0
Node 19 (n = 0
Node 23 (n = 0
Node 24 (n = 0
Node 26 (n = 0
Node 27 (n = 0
Node 28 (n = 0
Node 30 (n = 0
Node 34 (n = 0
Node 35 (n = 0
Node 36 (n = 0
Node 37 (n = 9
0

– Какие «правила» для предсказания продаж мы получили в этом случае?

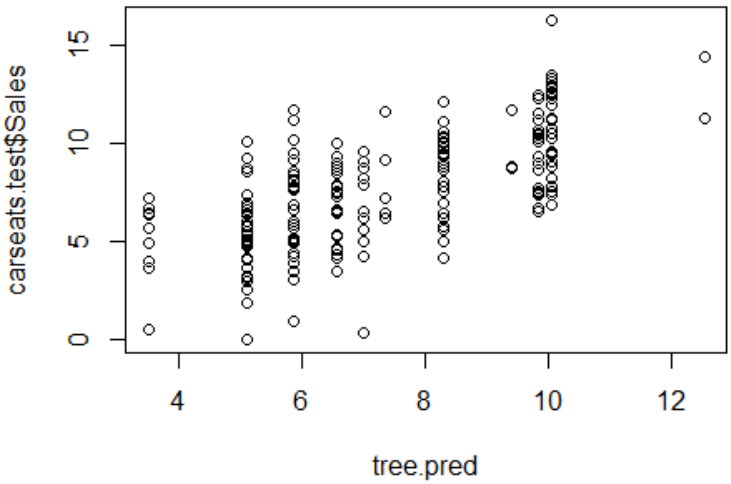
Как оценить качество регрессионной модели? Разделим выборку снова на две части.

```
set.seed(2)
train=sample(1:nrow(carseats), 200)
carseats.test <- carseats[-train,]
carseats.train <- carseats[train,]
tree.regr <- ctree(Sales~., carseats.train)
```

– Что будет, если мы попробуем построить матрицу несоответствий (confusion matrix)?

```
tree.pred <- predict(tree.regr, carseats.test)
tree.pred
##        1        2        5        9       10       12       13
##  5.883103 12.539091 5.113214 6.585500 6.585500 10.055000  3.508182
##       15       16       18       21       25       28       30
## 10.055000 7.017273 10.055000 5.113214 5.883103  6.585500  5.883103
##       33       34       36       39       40       43       45
##  7.347500 10.055000 8.296364 6.585500 5.113214  9.833750  6.585500
##       48       52       53       54       58       62       63
##  5.883103  5.883103  7.017273 8.296364 5.883103 6.585500  5.113214
##       65       66       68       72       75       76       77
##  8.296364  3.508182 8.296364 7.017273 7.017273  5.883103  9.833750
##       80       81       82       83       86       89       95
##  5.883103  5.883103 10.055000 7.347500 6.585500 6.585500  5.883103
##       97       99      100      101      109      110      111
## 10.055000 10.055000 5.883103 8.296364 5.883103  9.833750  6.585500
##      112      115      116      117      119      120      122
##  5.113214  6.585500 5.113214 5.113214 6.585500  5.113214  5.883103
##      127      129      132      135      136      137      138
## 10.055000  5.113214 9.833750 5.113214 8.296364  5.883103  6.585500
##      139      140      143      144      146      148      149
##  8.296364  9.833750 6.585500 3.508182 9.397143 10.055000  6.585500
##      150      151      152      156      158      159      161
##  9.833750 10.055000 10.055000 9.833750 9.833750 10.055000 6.585500
##      162      164      166      167      168      169      171
##  5.113214  5.883103 7.017273 3.508182 9.833750  6.585500  8.296364
##      172      174      175      178      181      182      183
##  9.833750  5.113214 5.113214 9.833750 5.113214  9.833750  5.113214
##      185      186      188      190      191      193      194
##  6.585500  8.296364  .883103 8.296364 8.296364  9.833750 10.055000
```

```
##       195       196       197       198       199       200       201
##  3.508182  5.883103 5.113214 5.113214 3.508182  3.508182  7.017273
##       205       206       207       211       214       217       218
##  9.397143  3.508182 7.017273 5.883103 7.017273  5.113214  6.585500
##       219       221       222       224       227       230       231
##  8.296364  8.296364 6.585500 6.585500 10.055000 9.833750  5.883103
##       233       234       235       236       239       240       244
## 10.055000  8.296364 10.055000 5.113214 10.055000 5.883103 8.296364
##       245       249       250       251       255       256       257
##  6.585500  6.585500 5.883103 7.347500 10.055000 5.883103  7.017273
##       260       261       263       264       267       268       272
##  5.883103  5.883103 3.508182 6.585500 10.055000 5.883103  6.585500
##       274       276       277       278       279       282       287
##  9.833750  3.508182 5.113214 8.296364 7.347500 10.055000  8.296364
##       288       290       291       292       294       295       297
##  5.883103  5.113214 8.296364 5.883103 12.539091 10.055000 10.055000
##       298       300       304       307       309       310       311
##  5.883103  8.296364 8.296364 5.113214 5.113214  5.883103  7.017273
##       312       313       314       316       320       321       324
##  7.017273  5.113214 9.833750 7.347500 5.113214  5.113214  8.296364
##       326       330       331       332       333       335       336
##  9.397143 12.539091 5.883103 5.113214 8.296364  5.883103  8.296364
##       338       342       343       344       348       349       350
##  6.585500  9.833750 8.296364 5.883103 10.055000 10.055000 8.296364
##       351       353       355       360       361       364       367
##  9.833750 10.055000 5.113214 5.113214 10.055000 10.055000 5.113214
##       368       371       372       376       377       383       385
## 12.539091  5.883103 7.017273 6.585500 10.055000 8.296364 10.055000
##       386       389       394       395
##  5.113214  5.883103  8.296364  5.113214
plot(tree.pred, carseats.test$Sales)
```



Посчитаем, насколько наше предсказание отличается от реального значения.

```
sum((tree.pred - carseats.test$Sales)^2) ##RSS
## [1] 930.8802
sum((mean(carseats.test$Sales) - carseats.test$Sales)^2) #TSS
## [1] 1605.021
sum((tree.pred - carseats.test$Sales)^2)/sum((mean(car-
seats.test$Sales) - carseats.test$Sales)^2)
## [1] 0.5799801
```

Загрузим данные.

```
load("autographs.rda")
```

В феврале 2016 г. были собраны данные с рынка виртуальных товаров steam community market в разделе Dota 2. Далее были выделены предметы, которые являются виртуальными автографами. Виртуальные автографы – это цифровые подписи известных игроков, которые один раз были созданы игроком, а затем уже продаются копии, то есть автографы одного игрока не уникальны. Ежегодно организуются турниры серии The International (TI), которые являются аналогом чемпионата мира. Также есть серия турниров Major, турниры которой вторые по престижу после TI и организуются в разных городах. На тот момент последним турниром серии Major был Shanghai Major. Были собраны данные об успехах игро-

ков на этих турнирах. Кроме этого, были включены в игровую статистику: количество профессиональных игр, винрейт, KDA ratio. Также была взята информация, связанная с публичностью игроков: количество подписчиков в twitter, а также количество интервью на медиа разных категорий. Про самих игроков также есть информация о роли в команде (от 5 до 1, роль в команде определяется персонажами, на которых игрок играет, но для нас важно, что чем ниже число – тем выше влияние игрока на исход матча), национальности и количестве призовых мест у команды, членом которой игрок является. С рынка была собрана информация о количестве копий автографа и его цене.

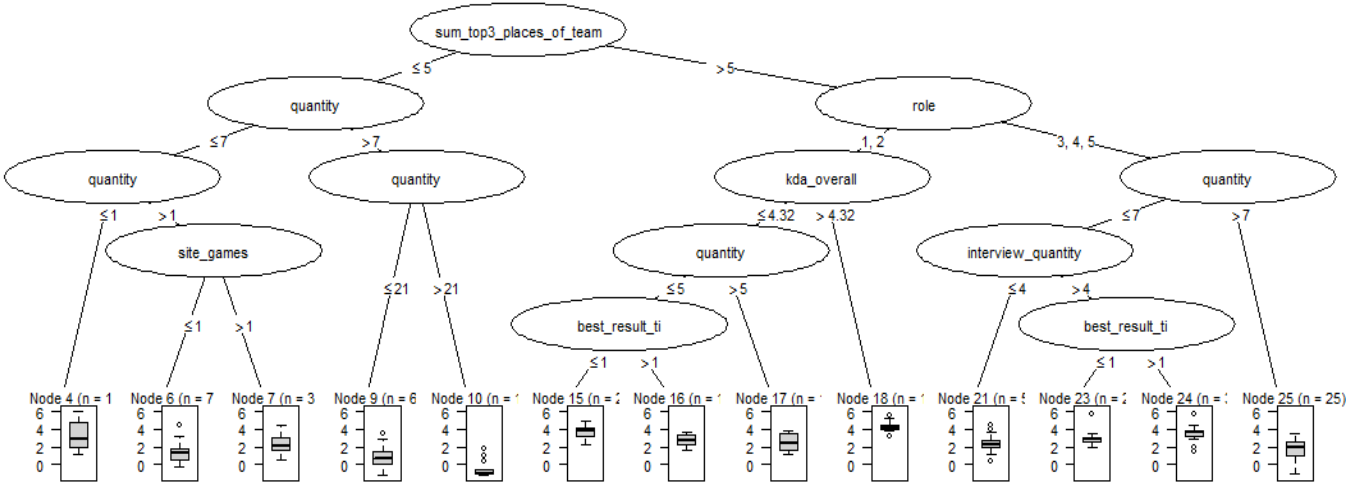
Описание переменных

- Quantity – количество копий автографа на рынке.
- Price – цена за автограф.
- Role – роль в команде.
- Country – страна происхождения / национальность.
- Last_participation_year – последний год участия на турнире The International.
- Best_result_ti – лучший результат на турнире The International за все годы.
- Place_num_recent – место на The International 2015. 0000 означает место на квалификациях, 00 – означает место на wild card.
- Is_2015 – принимал ли участие на The International 2015 или нет.
- Freq – сколько раз принимал участие на турнире The International.
- Sm_participant – является ли участником Shanghai Major.
- Games_overall – всего профессиональных игр.
- Winrate_overall – процент побед за все игры.
- Kda_overall – KDA-статистика за все игры. Kda рассчитывается по формуле: Kills+Assists / Deaths, что означает в скольких убийствах поучаствовал игрок в пересчете на каждую свою смерть.
- Follows – количество подписчиков в твитере.
- Interview_quantity – количество интервью в целом.
  - Community – интервью, взятые представителями фан сообщества (с форумов или реддита).
  - Dota – интервью, взятые для сайтов о доте.
  - Site_games – интервью для сайтов об играх в целом.
  - Not_games – интервью для сайтов, не связанных с играми (зачастую национальные СМИ).
  - Teams – интервью для сайтов профессиональных команд.
  - Tournament – интервью, взятые по ходу турнира, специальной командой, его освещающей.

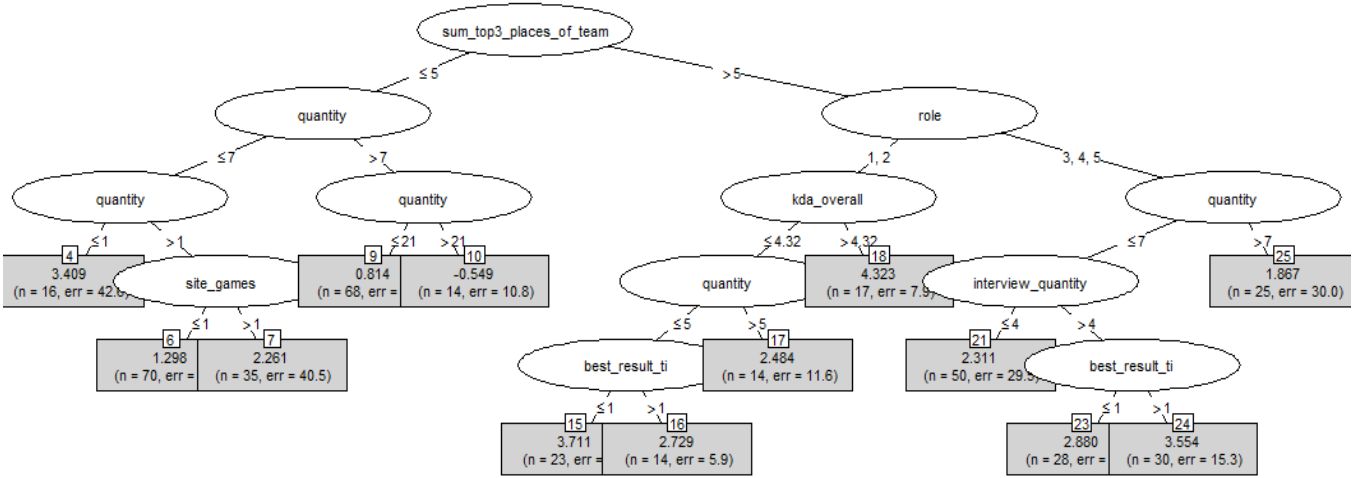
– Sum_top3_places_of_team – сумма мест от третьего и выше, занятых командой.

Построим модель для предсказания цены автографа.

```
tr1.1<-ctree(log(price)~.,data=autographs_model)
plot(tr1.1, gp = gpar(fontsize = 8),     # font size changed
to 6
      inner_panel=node_inner,
      ip_args=list(
        abbreviate = F,
        id = FALSE,
        pval=F)
 )
```

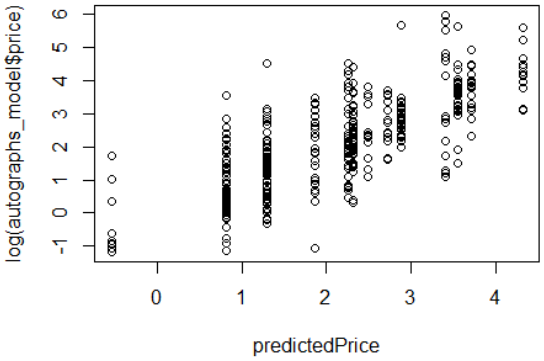

sum_top3_places_of_team
≤ 5
> 5
quantity
role
≤ 7
> 7
1, 2
3, 4, 5
quantity
quantity
kda_overall
quantity
≤ 1
> 1
≤ 4.32
> 4.32
≤ 7
> 7
site_games
quantity
interview_quantity
≤ 1
> 1
≤ 21
> 21
≤ 5
> 5
≤ 4
> 4
best_result_ti
best_result_ti
≤ 1
> 1
≤ 1
> 1
Node 4 (n = 1
Node 6 (n = 7
Node 7 (n = 3
Node 9 (n = 6
Node 10 (n =
Node 15 (n =
Node 16 (n =
Node 17 (n =
Node 18 (n =
Node 21 (n =
Node 23 (n =
Node 24 (n =
Node 25 (n = 25)
6
4
2
0

```
plot(tr1.1, gp = gpar(fontsize = 8),     # font size changed to 6
      inner_panel=node_inner,
      ip_args=list(
        abbreviate = F,
        id = FALSE,
        pval=F),
     type = "simple"
 )
```


sum_top3_places_of_team
≤ 5
> 5
quantity
role
≤ 7
> 7
1, 2
3, 4, 5
quantity
quantity
kda_overall
quantity
≤ 1
> 1
≤ 21
> 21
≤ 4.32
> 4.32
≤ 7
> 7
4
3.409
(n = 16, err = 42.0)
site_games
9
0.814
(n = 68, err =
10
-0.549
(n = 14, err = 10.8)
quantity
18
4.323
(n = 17, err = 7.9)
interview_quantity
25
1.867
(n = 25, err = 30.0)
≤ 1
> 1
≤ 5
> 5
≤ 4
> 4
6
1.298
(n = 70, err =
7
2.261
(n = 35, err = 40.5)
best_result_ti
17
2.484
(n = 14, err = 11.6)
21
2.311
(n = 50, err = 29.5)
best_result_ti
≤ 1
> 1
≤ 1
> 1
15
3.711
(n = 23, err =
16
2.729
(n = 14, err = 5.9)
23
2.880
(n = 28, err =
24
3.554
(n = 30, err = 15.3)

– Зачем мы считаем логарифм от цены?
– Какие переменные влияют на цену?

```
predictedPrice <- predict(tr1.1)
plot(predictedPrice, log(autographs_model$price))
```



Как посчитать качество в этом случае?

```
rss = sum((log(autographs_model$price)-predictedPrice)^2)
tss = sum((log(autographs_model$price) - mean(log(auto-
graphs_model$price)))^2)
1 - rss/tss
## [1] 0.6073333
```

Задание 1

1. Загрузите данные о стоимости квартир в пригородах Бостона (Boston). Более подробная информация в справке `?Boston`

```
library(MASS)
##
## Attaching package: 'MASS'
##  The  following  object  is  masked  from  'package:dplyr':
##
##     select
data(Boston)
```

2. Разделите цену на высокую и низкую (граница 20)

```
High=ifelse(Boston$medv<=20,"No","Yes")
Boston <- data.frame(Boston, High)
Boston <- dplyr::select(Boston, -medv)
```

3. Постройте модели для предсказания цены на квартиры:

- высокая/низкая,
- конкретное значение.

Задание 2

1. Загрузите данные о пассажирах Титаника

```
titanic <- read.csv("Titanic.csv")
```

2. Постройте модель для предсказания выживших по полу, возрасту и классу билета.

3. Оцените качество модели.

Список библиографических источников

Основной

1. *Горяинова, Е. Р.* Прикладные методы анализа статистических данных: учеб. пособие / Е. Р. Горяинова, А. Р. Панков, Е. Н. Платонов. – М.: Изд. дом Высшей школы экономики, 2012. – 310 с.

2. *Лесковец, Ю.* Анализ больших наборов данных / Ю. Лесковец, А. Раджараман. – М.: ДМК, 2016. – 498 с.

3. *Крянев, А. В.* Метрический анализ и обработка данных / А. В. Крянев, Г. В. Лукин, Д. К. Удумян. – М.: Физматлит, 2012. – 308 с.

4. *Кулаичев, А. П.* Методы и средства комплексного анализа данных: учеб. пособие / А. П. Кулаичев. – М.: Форум, НИЦ ИНФРА-М, 2013. – 512 с.

5. *Банержи Ашис* Медицинская статистика понятным языком / А. Банержи. – М.: Практическая медицина, 2014.

6. 1Biostatistics with R: An Introduction to Statistics Through Biological Data (Use R!) / Babak Shahbaba, 2012.

7. Modern Epidemiology Third, Mid-cycle revision Edition / J. Kenneth Rothman, L. Timothy Lash Associate Professor, Sander Greenland, 2012.

8. Biostatistics and Epidemiology: A Primer for Health and Biomedical Professionals 4th ed. / Sylvia Wassertheil-Smoller, Jordan Smoller, 2015.

9. Introductory Time Series with R (Use R!) / Paul S. P. Cowpertwait, Andrew V. Metcalfe, 2009.

Дополнительный

10. *Новикова, Н. М.* Статистические методы в биологии и медицине: учеб.-метод. пособие / Н. М. Новикова. – Минск: МГЭУ им. А. Д. Сахарова, 2009. – 88 с.

11. *Кабаков, Р.* R в действии. Анализ и визуализация данных в программе R / Р. Кабаков. – М.: ДМК, 2016. – 588 с.

12. Биометрика – журнал для медиков и биологов, сторонников доказательной биомедицины [Электронный ресурс]. URL: http://www.-biometrica.tomsk.ru.

13. *Орлов, А. И.* Прикладная статистика: учебник / А. И. Орлов. – М.: Изд-во «Экзамен», 2004. – 656 с.

14. *Чиркин, А. А.* Клинический анализ лабораторных данных / А. А. Чиркин. – Витебск: Медицинская литература, 2008. – 384 с.

15. *Петри, А.* Наглядная статистика в медицине / А. Петри, К. Сэбин. – М.: Изд. дом ГЭОТАР-МЕД, 2003 – 143 с.

Учебное издание

Сыса Алексей Григорьевич,
Дудинская Регина Александровна

МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ

Учебно-методическое пособие

Редактор *Л. М. Кореневская*
Компьютерная верстка *Д. В. Головач*
Техническое редактирование *А. В. Красуцкая*

Подписано в печать 20.06.2018. Формат 60×90 1/16.
Бумага офсетная. Печать офсетная.
Усл. печ. л. 13,69. Уч.-изд. л. 5,74.
Тираж 100 экз. Заказ № 202.

Республиканское унитарное предприятие «Информационно-вычислительный центр Министерства финансов Республики Беларусь».
Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий
№ 1/161 от 27.01.2014, № 2/41 от 29.01.2014.
Ул. Кальварийская, 17, 220004, г. Минск.