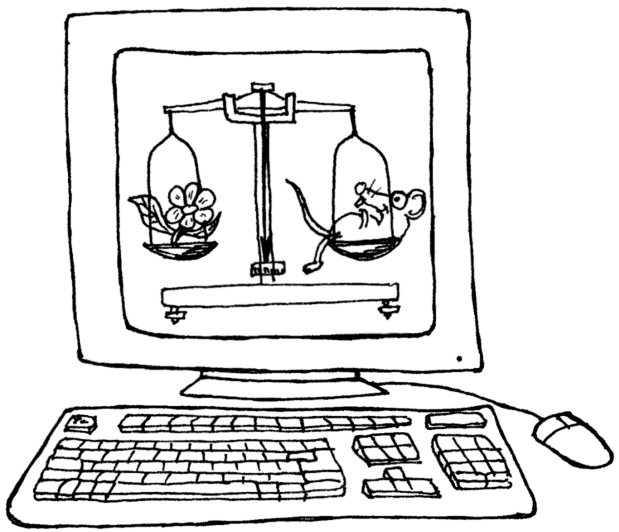
П.А. Волкова, А.Б. Шипунов

Статистическая обработка данных: это должен знать каждый



Москва ❖ 2017

Рецензенты:
канд. физ.-мат. наук, канд. филос. наук, преподаватель московского лицея № 1553 («Лицей на Донской») А.С. Горелов;
канд. геогр. наук, научный сотрудник Института проблем экологии и эволюции им. А.Н. Северцова РАН А.С. Зайцев;
канд. биол. наук, канд. философ. наук, директор Московского детско-юношеского центра экологии, краеведения и туризма Д.В. Моргун

П.А. Волкова, А.Б. Шипунов. Статистическая обработка данных: это должен знать каждый. Издание 3-е, переработанное. — М., 2017. — 89 с.

В учебном пособии обсуждаются основы статистического анализа. Применение разных методов обработки данных рассмотрено на примере свободно распространяемой программы R. Пособие предназначено для школьников и студентов, а также может использоваться педагогами, работающими в системе дополнительного образования детей.

Иллюстрации Ю.С. Быкова

Предисловие

Это пособие написано для тех, кто хочет научиться обрабатывать данные. Такая задача возникает очень часто, особенно когда нужно выяснить ранее неизвестный факт. Например: есть ли эффект от нового лекарства? Или: различаются ли рейтинги двух политиков? Или: как будет меняться курс доллара на следующей неделе? Многие люди думают, что неизвестный факт можно выяснить, если просто немного подумать над данными. К сожалению, часто это совершенно не так. Например, по опросу 262 человек, выходящих с избирательных участков, выяснилось, что 52% проголосовало за кандидата A, а 48% — за кандидата B. Значит ли это, что кандидат A победил? Подумав, многие сначала скажут «Да», а через некоторое время, «Кто его знает». Но есть очень простой (с точки зрения современных компьютерных программ) тест пропорций, который позволяет не только понять, что данных для однозначного ответа недостаточно, но и вычислить, сколько надо было опросить человек, чтобы можно было уверенно определить победителя. В описанной ситуации это примерно 2500 человек!

В общем, если бы люди знали, что можно сделать методами анализа данных, ошибок и неясностей в нашей жизни стало бы гораздо меньше. К сожалению, ситуация в этой области далека от благополучия. С другой стороны, ведь совсем не обязательно знать радиофизику для того, чтобы слушать любимую передачу по радиоприемнику. Значит, для того, чтобы анализировать данные в практических целях, не обязательно свободно владеть математической статистикой и теорией вероятностей. Эту проблему давно уже почувствовали многие английские и американские авторы — названиями типа «Статистика без слез» пест-

рят книжные полки магазинов, посвященные книгам по анализу данных.

Тут, правда, следует быть осторожным как авторам, так и читателям таких книг: многие методы анализа данных имеют, если можно так выразиться, двойное дно. Эти методы можно применять, глубоко не вникая в сущность используемой там математики, получать результаты и обсуждать эти результаты в отчетах. Однако в один далеко не прекрасный день может выясниться, что данный метод был (с позиции теории, разумеется) совершенно неприменим для Ваших данных, и поэтому полученные результаты и результатами-то назвать нельзя… Что-то похожее происходит при тестировании компьютерных программ: программа может отлично работать, выполняя все, что от нее требуется, но однажды какой-то пустяк (например, какое-то редкое слово или просто сочетание букв, набранное в окне текстового редактора) приводит к ее «зависанию» или даже к более серьезным последствиям. Дело, наверное, в том, что вероятность ошибок растет с увеличением сложности, а методы анализа данных часто очень сложны (в математическом выражении, конечно). В общем, будьте бдительны, внимательно читайте про все *ограничения* методов анализа.

Глава 1

Что такое данные и зачем их обрабатывать?

В этой главе рассказано о самых общих аспектах анализа данных. Статистики и математики, как представители любой другой профессии, выработали свой собственный язык, которым должны, хотя бы частично, овладеть те, кто желает проникнуть в их тайны.

1.1. Откуда берутся данные

«Без пруда не выловишь и рыбку из него» — говорит народная компьютерная мудрость. Действительно, если хочешь анализировать данные, надо их сначала получить.

Способов получения данных много. Можно их просто выдумать, но в таком случае результатом анализа будут сведения о том, что творится в Вашей собственной голове, а не в окружающей Вас действительности. Можно взять данные (да и выводы тоже, вот и обрабатывать ничего не надо) из книг тех авторов, которым Вы доверяете — это называется «апелляция к авторитетам», а иногда и просто «плагиат». Такой подход был широко распространен в средние века, а сейчас процветает в средней школе. Но опытный учитель знает, что если на вопрос существуют два ответа — правильный и неправильный, то большинство учеников спишет друг у друга неправильный ответ. Дело в том,

что согласно одному из законов Мерфи, «любая проблема имеет простое, изящное и неправильное решение» — неправильный ответ проще.

Чтобы не уподобляться упомянутым выше персонажам, нужно использовать данные, полученные в результате *наблюдения* или *эксперимента*.

Наблюдением будем называть такой способ получения данных, при котором воздействие наблюдателя на наблюдаемый объект сведено к минимуму. **Эксперимент** тоже включает наблюдение, но сначала на наблюдаемый объект оказывается заранее рассчитанное воздействие. Для наблюдения очень важно это «сведение воздействия к минимуму». Если этого не сделать, мы получим данные, отражающие не «исконные» свойства объекта, а его реакцию на наше воздействие.

Вот, например, встала задача исследовать, чем питается какое-то редкое животное. Оптимальная стратегия наблюдения здесь состоит в установке скрытых камер во всех местах, где это животное обитает. После этого останется только обработать снятое, чтобы определить вид пищи.

Очень часто, однако, оптимальное решение совершенно невыполнимо, и тогда пытаются обойтись, скажем, наблюдением за животным в зоопарке. Ясно, что в последнем случае на объект оказывается воздействие, и немалое. В самом деле, животное поймали, привезли в совершенно нетипичные для него условия, да и корм, скорее всего, будет непохож на тот, каким оно питалось на родине. В общем, если наблюдения в зоопарке поставлены грамотно, то выяснено будет не то, чем вообще питается данное животное, а то, чем оно питается при содержании в определенном зоопарке. К сожалению, многие (и исследователи, и те, кто потом читает их отчеты) часто не видят разницы между этими двумя вариантами наблюдений, что может привести к очень серьезным последствиям.

Вернемся к примеру из предисловия. Предположим, мы опрашиваем выходящих с избирательных участков. Часть людей, ко-

нечно, вообще окажется отвечать. Часть ответит что-нибудь, не относящееся к делу. Часть вполне может намеренно или случайно исказить свой ответ. Часть ответит правду. И все это серьезным образом зависит от наблюдателя — человека, проводящего опрос, а также от многих внешних факторов.

Даже упомянутые выше скрытые камеры приведут к определенному воздействию. Нет никакой гарантии, что наше животное или его добыча не отреагирует на них. А кто будет ставить камеры? Если это люди, то чем больше камер поставить, тем сильнее будет воздействие на окружающую среду. Сбрасывать с вертолета? Сами понимаете, к чему это может привести.

В общем, из сказанного должно быть понятно, что наблюдение «в чистом виде» более или менее неосуществимо, поскольку всегда будет внесено какое-нибудь воздействие. Поэтому для того, чтобы адекватно работать с данными наблюдений, надо всегда четко представлять, как они проводились.

Если воздействие было значительным, то надо представлять (хотя бы теоретически) какие оно могло повлечь изменения, а в отчете обязательно указать на те ограничения, которые были вызваны способом наблюдения.

Не следует без необходимости применять экстраполяцию: это значит, что если мы увидели, что А делает Б, нельзя писать «А всегда делает Б» и даже «А обычно делает Б». Можно лишь писать нечто вроде «в наших наблюдениях А делал Б, это позволяет предположить, что Б для него — обычное дело».

У эксперимента свои проблемы. Наиболее общие из них — это точный учет воздействия и наличие контроля. Например, мы исследуем действие нового лекарства. Классический эксперимент состоит в том, что выбираются две группы больных (*как* выбрать такие группы, *сколько* должно быть человек и пр. рассмотрено в последующих разделах). Всем больным сообщают, что проводится исследование нового лекарства, но его дают только больным первой группы, остальные получают так назы-

ваемое плацебо, внешне неотличимое от настоящего лекарства, но не содержащее ничего лекарственного.

Зачем это делается? Дело в том, что если больной будет знать, что ему дают «ненастоящее» лекарство, то это скажется на эффективности лечения, потому что результат зависит не только от того, что больной пьет, но и от того, что он чувствует. Иными словами, психологическое состояние больного — это дополнительный фактор воздействия, от которого в эксперименте лучше избавиться.

Очень часто не только больным, но и их врачам не сообщают, кому дают плацебо, а кому — настоящее лекарство («двойной слепой метод»). Это позволяет гарантировать, что и психологическое состояние врача не повлияет на исход лечения.

Группа, которой дают плацебо (она называется **контроль**), нужна для того, чтобы отделить эффект, который может произвести лекарство, от эффекта какого-нибудь постороннего внешнего фактора.

Известно, например, что уменьшение длины светового дня в октябре-декабре провоцирует многие нервные заболевания. Если наше исследование придется как раз на эти месяцы, и у нас не будет контроля, то увеличение частоты заболеваний мы вполне можем принять за результат применения лекарства.

1.2. Генеральная совокупность и выборка

«Статистика знает все» — писали Ильф и Петров в «Двенадцати стульях», имея в виду то, что обычно называют статистикой — сбор всевозможной информации обо всем на свете. Чем полнее собрана информация, тем, как считается, лучше. Однако лучше ли?

Возьмем простой пример. Допустим, фирма-производитель решила выяснить, какой из двух сортов производимого мороженого предпочитают покупатели. Проблем бы не было, если бы

все мороженое продавалось в одном магазине. На самом же деле продавцов несчетное множество: это оптовые рынки и гипермаркеты, средние и малые магазины, киоски, отдельные мороженщики с тележками, те, кто торгует в пригородных поездах и т.п. Можно попробовать учесть доход от продажи каждого из двух сортов. Если они сто́ят одинаково, то бо́льшая сумма дохода должна отразить больший спрос. Представим, однако, что спрос одинаков, но по каким-то причинам мороженое первого сорта тает быстрее. Тогда потерь при его транспортировке будет в среднем больше, продавцы будут покупать его несколько чаще, и получится, что доход от продажи первого сорта будет несколько выше, чем от второго. Это рассуждение, конечно, упрощает реальную ситуацию, но подумайте, сколько других неучтенных факторов стоит на пути такого способа подсчета! Анализ товарных чеков лучше, однако, многие конечные продавцы таких чеков не имеют и, поэтому в анализ не попадут. А нам-то необходимо как раз учесть спрос покупателей, а не промежуточных продавцов.

Можно поступить иначе — раздать всем конечным продавцам анкеты, в которых попросить указать, сколько какого мороженого продано; а чтобы анкеты были обязательно заполнены, вести с этими продавцами дела только при наличии заполненных анкет. Только ведь никто не будет контролировать, *как* продавцы заполняют анкеты… Вот и получит фирма большую, подробную сводную таблицу о продажах мороженого, которая ровным счетом ничего отражать не будет.

Как же поступить? Здесь на помощь приходит идея **выборочных исследований**. Всех продавцов не проконтролируешь, но ведь нескольких-то можно! Надо выбрать из общего множества несколько торговых точек (*как* выбирать, мы расскажем позже) и проконтролировать тамошние продажи силами самой фирмы или такими нанятыми людьми, которым можно доверять (исследованные объекты принято называть **выборкой**). В результате мы получим результат, который является частью общей картины. Теперь самый главный вопрос — можно ли этот результат распространить на все множество интересующих нас

объектов (**генеральную совокупность**)? Оказывается, можно, поскольку на основе теории вероятностей уже много лет назад была создана теория выборочных исследований. Её-то и называют чаще всего математической статистикой, или просто статистикой.

Пример с мороженым показывает важную вещь: выборочные исследования могут быть (и часто бывают) значительно более точными (в смысле соответствия реальности), чем сплошные.

Еще один хороший пример на эту же тему есть в результатах сплошной переписи населения России 1897 г. Если рассмотреть численность населения по возрастам, то получается, что максимальные численности («пики») имеют возраста кратные 5 и в особенности кратные 10. Понятно, как это получилось. Бóльшая часть населения в те времена была неграмотна, и свой возраст помнила только приблизительно, с точностью до пяти или до десяти лет. Чтобы все-таки узнать, каково было распределение по возрастам на самом деле, нужно не увеличивать объем данных, а наоборот, создать выборку из нескольких процентов населения и провести комплексное исследование, основанное на перекрестном анализе нескольких источников: документов, свидетельств и личных показаний. Это даст гораздо более точную картину, нежели сплошная перепись.

Естественно, сам процесс создания выборки может являться источником ошибок. Их принято называть **ошибками репрезентативности**. Однако правильная организация сбора данных позволяет их избежать.

1.3. Как получать данные

В предыдущих разделах мы неоднократно упоминали, что от правильного составления выборки серьезным образом будет зависеть качество получаемых данных. Собственно говоря, есть два основных принципа составления выборки: повторности и

рандомизация, позволяющие минимизировать влияние отклонений, вызванных посторонними причинами.

Принцип повторностей заключается в многократном исследовании одного и того же эффекта. Собственно говоря, для этого мы в предыдущих примерах опрашивали *множество* избирателей, ловили в заповеднике *много* животных, подбирали группы из *нескольких десятков* больных и контролировали *различных* продавцов мороженого. Необходимость в использовании многих повторностей возникает оттого, что все объекты (даже только что изготовленные на фабрике изделия), пусть в мелочах, но отличаются друг от друга. Эти отличия способны затуманить общую картину, если мы станем изучать объекты поодиночке. И наоборот, если мы берем несколько объектов сразу, их индивидуальные (не связанные с исследуемым эффектом) различия как бы «взаимно уничтожаются».

Не стоит думать, что создать повторности — простое дело. К сожалению, часто именно небрежное отношение к повторностям сводит на нет результаты вроде бы безупречных исследований. Главное правило: *повторности должны быть независимы друг от друга*. Это значит, например, что нельзя в качестве повторностей рассматривать данные, полученные в последовательные промежутки времени с одного и того же объекта или с одного и того же места. Предположим, что мы хотим определить размер лягушек какого-то вида. Для этого с интервалом в 15 минут (чтобы лягушки успокоились) ловим сачком по одной лягушке. Как только наберется двадцать лягушек, мы их меряем и вычисляем средний размер. Однако такое исследование не будет удовлетворять правилу независимости повторностей, потому что каждый отлов окажет влияние на последующее поведение лягушек (например, к концу лова будут попадаться самые смелые, или, наоборот, самые глупые). Еще хуже использовать в качестве повторностей последовательные наблюдения за объектом. Например, в некотором опыте выясняли скорость зрительной реакции, показывая человеку на доли секунды предмет, а затем спрашивая, что это было. Всего исследовали 10 человек, причем каждому показывали предмет пять раз. Авторы опыта посчита-

ли, что у них было 50 повторностей, однако на самом деле — только десять. Это произошло потому, что каждый следующий показ не был независим от предыдущего (человек мог, например, научиться лучше распознавать предмет).

Надо быть осторожным не только с данными, собранными в последовательные промежутки времени, но и просто с данными, собранными с одного и того же места. Например, если мы определяем качество телевизоров, сходящих с конвейера, не годится в качестве выборки брать несколько штук подряд — с большой вероятностью они изготовлены в более близких условиях, чем телевизоры, взятые порознь, и, стало быть, их характеристики не независимы друг от друга.

Второй важный вопрос о повторностях — сколько надо собрать данных[1]. Есть громадная литература по этому поводу, но ответа, в общем, два: «столько, сколько возможно» и «30». Выглядящее несколько юмористически «правило 30» освящено десятилетиями опытной работы. Считается (условно, конечно), что выборки, меньшие 30, следует называть малыми, а бóльшие — большúми. Бывает так, что и 30 объектов исследовать нельзя, однако огорчаться этому не стóит, поскольку многие методы анализа данных способны работать с очень малыми выборками, в том числе из пяти и даже из трех повторностей. Следует, однако, иметь в виду, что чем меньше повторностей, тем менее достоверными будут выводы.

Рандомизация — еще одно условие создания выборки, и также «с подвохом». Предположим, нам поручено случайным образом отобрать сто деревьев в лесу, чтобы впоследствии померить степень накопления тяжелых металлов в листьях. Как мы будем выбирать деревья? Если просто ходить по лесу и собирать листья с разных деревьев, с большой вероятностью они не будут собранными случайно, потому что вольно или невольно мы будем собирать листья, чем-то привлекшие наше внима-

[1] Существуют специальные методы, которые позволяют посчитать, сколько надо собрать данных для того, чтобы с определенной вероятностью высказать некоторое утверждение. Это так называемые тесты мощности.

ние (необычностью, окраской, доступностью). Этот метод, стало быть, не годится.

Возьмем метод посложнее — для этого нужна карта леса с размеченными координатами. Мы выбираем случайным образом два числа, например, 123 м к западу и 15 м к югу от точки, находящейся примерно посередине леса, затем отмеряем это расстояние на местности и выбираем дерево, которое ближе всего к нужному месту. Будет ли такое дерево выбрано случайно? Оказывается, нет. Ведь деревья растут группами, поэтому у деревьев, растущих плотно (например, у елок), шанс быть выбранными окажется значительно меньше, чем у редко растущих дубов. Принцип рандомизации состоит в том, что *все объекты генеральной совокупности имеют равные шансы попасть в выборку*. Как же быть? Надо просто перенумеровать все деревья, а затем выбрать сто номеров по жребию. Но это только звучит просто, а попробуйте так сделать! А если надо сравнить 20 различных лесов?..

В общем, требование рандомизации часто оборачивается весьма серьезными затратами на проведение исследования. Естественно поэтому, что часто рандомизацию осуществляют лишь частично. Например, в нашем случае можно случайно выбрать направление, протянуть в этом направлении бечевку через весь лес, а затем посчитать, скольких деревьев касается бечевка и выбрать каждое *n*-ное (пятое, пятнадцатое и т.п.) дерево так, чтобы всего в выборке оказалось 100 деревьев. Заметьте, что в данном случае рандомизация заключается в том, чтобы внести в исследуемую среду такой порядок, которого там заведомо нет. Конечно, у этого последнего метода есть недостатки, а какие — попробуйте сообразить сами. (*Задача 1*).

Теперь Вы знаете достаточно, чтобы ответить на еще один вопрос. В одной лаборатории изучали эффективность действия ядохимикатов на жуков-долгоносиков (их еще называют «слоники»). Для этого химикат наносили на фильтровальную бумагу, а бумагу помещали в стеклянную чашку с крышкой (чашку Петри). Жуков выбирали из банки, в которой их разводили для опытов, очень простым способом: банку с жуками открыва-

ли, и первого выползшего на край жука пересаживали в чашку с ядохимикатом. Затем засекали, сколько пройдет времени от посадки жука в чашку до его гибели. Потом брали другого жука, и так повторяли 30 раз. Потом меняли ядохимикат и начинали опыт сначала. Но однажды один умный человек заметил, что в этом эксперименте самым сильным всегда оказывался тот химикат, который был взят для исследования первым. Как Вы думаете, в чем тут дело? Какие нарушения принципов повторности и рандомизации были допущены? Как надо было поставить этот опыт? (*Задача 2*).

Для рандомизации, конечно, существует предел. Если мы хотим выяснить возрастной состав посетителей какого-то магазина, не нужно в целях рандомизации опрашивать прохожих на улице. Нужно четко представлять себе генеральную совокупность, с которой идет работа, и не выходить за ее границы. Помните пример с питанием животного? Если генеральная совокупность — это животные данного вида, содержащиеся *в зоопарках*, нет смысла добавлять к исследованию данные о питании этих животных *в домашних условиях*. Если же такие данные просто необходимо добавить (например, потому что данных из зоопарков очень мало), то тогда генеральная совокупность будет называться «множество животных данного вида, содержащихся *в неволе*».

Интересный вариант рандомизации используют, когда в эксперименте исследуют одновременно несколько взаимодействий. Например, мы хотим выяснить эффективность разных типов солевой засыпки тротуаров. Для этого логично выбрать (рандомизация!) несколько разных (по времени застройки, плотности населения, расположению и т. п.) участков города и внутри каждого участка случайным образом распределить разные типы засыпок. Потом можно, например, фиксировать (в баллах или как-нибудь еще) состояние тротуаров каждый день после нанесения засыпки, можно также повторить опыт при разной погоде. Такой подход называется «блочный дизайн». Блоками здесь являются разные участки города, а повторность обеспечивается тем, что в каждом блоке повторяются одни и те же воздействия.

С рандомизацией связано еще одно принципиальное различие между наблюдением и экспериментом. Допустим, мы изучаем эффективность действия какого-то лекарства. Вместо того, чтобы подбирать две группы больных, использовать плацебо и т.п., можно просто порыться в архивах и подобрать соответствующие примеры (30 случаев применения этого лекарства и 30 случаев, когда больных лечили по-другому), а затем проанализировать разницу между группами (например, число смертей в первый год после окончания лечения). Однако такой подход сопряжен с опасностью того, что на наши выводы окажет влияние какой-то неучтенный фактор (или несколько факторов), выяснить наличие которого из архивов невозможно. Мы просто не можем гарантировать, что соблюдали рандомизацию, анализируя архивные данные. К примеру, первая группа (случайно!) окажется состоящей почти целиком из пожилых людей, а вторая — из людей среднего возраста. Ясно, что это окажет воздействие на результаты анализа. Поэтому в общем случае эксперимент всегда предпочтительней наблюдения.

1.4. Что ищут в данных

Прочитав предыдущие разделы, читатель, наверное, уже не раз задавался вопросом: «Если так все сложно, зачем он вообще, этот анализ данных? Неужели и *так* не видно, что в один магазин ходит больше народу, одно лекарство лучше другого и т.п.?» В общем, *так* бывает видно только когда либо (1) данных и/или исследуемых факторов очень мало, либо (2) закономерность выражена очень ярко. В этих случаях, действительно, запускать всю громоздкую машину анализа данных не стоит. Однако гораздо чаще встречаются случаи, когда названные выше условия не выполняются. Давно, например, доказано, что средний человек может одновременно удержать в памяти лишь 5–9 объектов. Стало быть, анализировать в уме данные, которые насчитывают больше 10 компонентов, уже нельзя. А значит, не

обойтись без каких-нибудь, пусть и самых примитивных (типа вычисления процентов и средних величин), методов анализа данных.

Бывает и так, что закономерность, которая кажется очевидной, на самом деле не существует. Вот, например, одно из исследований насекомых-вредителей. Агрономы определяли, насколько сильно вредят кукурузе гусеницы кукурузного мотылька. Получились вполне ожидаемые результаты: разница в урожае между пораженными и непораженными растениями почти вдвое. Казалось, что и обрабатывать данные не надо — «и так все ясно». Однако нашелся вдумчивый исследователь, который заметил, что пораженные растения, различающиеся по степени поражения, не различаются по урожайности. Здесь очевидно что-то не так: если гусеницы вредят растению, то чем сильнее они вредят, тем меньше должен быть урожай. Стало быть, на какой-то стадии исследования произошла ошибка. Скорее всего, дело было так: для того, чтобы мерить урожайность, среди здоровых растений отбирали самые здоровые (во всех смыслах этого слова), ну а среди больных старались подобрать самые хилые. Вот эта ошибка репрезентативности и привела к тому, что возникли такие «хорошие» результаты. Только анализ взаимосвязи[2] поражение-урожай привел к выяснению истинного положения дел. А кукурузный мотылек, оказывается, почти и не вредит кукурузе…

Итак, статистический анализ данных необходим всегда, когда результат неочевиден и часто даже тогда, когда он кажется очевидным. Теперь разберемся, какие задачи можно решить при помощи анализа данных.

Во-первых, можно получать *общие характеристики* для больших выборок. Эти характеристики могут отражать так называемую центральную тенденцию, то есть число (или ряд чисел), вокруг которых, как пули вокруг десятки в мишени, «рассыпаны» данные. Всем известно, как считать среднее значение, но

[2] Статистики называют его регрессионный анализ.

существует еще немало полезных характеристик «на ту же тему». Другая характеристика — это разброс, который отражает, насколько сильно данные «рассыпаны» вокруг среднего.

Во-вторых, можно проводить *сравнения* между разными выборками. Например, можно выяснить, в какой из групп больных инфарктом миокарда частота смертей в первый год после лечения выше — у тех, кому делали коронарное шунтирование или у тех, к кому применяли только медикаментозные способы лечения. «На глаз» этой разницы может и не быть, а если она и есть, то где гарантия того, что эти различия не вызваны случайными причинами, не имеющими отношения к лечению? Скажем, заболел человек аппендицитом и умер после операции: к лечению инфаркта это может не иметь никакого отношения. Сравнение данных при помощи *статистических тестов* позволяет выяснить, насколько велика вероятность, что различия между группами вызваны случайными причинами. Заметьте, что полной уверенности анализ данных тоже не дает, зато позволяет количественно оценить шансы. Кстати говоря, анализ данных позволяет оценить и упомянутые выше общие характеристики для всей генеральной совокупности — вычислить так называемые доверительные интервалы[3].

Третий тип результатов, которые можно получить, анализируя данные — это сведения о взаимосвязях. Изучение взаимосвязей, наверное, самый серьезный и самый развитый раздел анализа данных. Существует множество методик выяснения и, главное, проверки «качества» связей. В дальнейшем нам понадобятся сведения о том, какие бывают взаимосвязи. Есть так называемые **соответствия**, например, когда два явления чаще встречаются вместе, нежели по отдельности (как гром и молния). Следующий тип взаимосвязей — это **корреляции**. Корреляции показывают силу и знак взаимосвязи. Наконец, есть **зависимости**, для которых можно измерить и силу, и знак, и оценить, насколько вероятно то, что они — результат случайных

[3]Диапазон значений, в котором с заданной (обычно 95%) вероятностью лежит та или иная характеристика генеральной совокупности (например, среднее значение определенного признака).

причин. Кстати говоря, последнее можно, как водится в анализе данных, сделать и для корреляций, и даже для соответствий. Еще одно свойство зависимостей состоит в том, что можно *предсказать*, как будет «вести» себя зависимая переменная в каких-нибудь до сих пор не опробованных условиях. Например, можно прогнозировать колебания спроса, устойчивость балок при землетрясении, интенсивность поступления больных и т.п.

И, наконец, анализ данных можно использовать для установления *структуры данных*. Это самый сложный тип анализа, поскольку для выяснения структуры обычно используют сразу несколько характеристик. Есть и специальное название для такой работы — «многомерная статистика». Самое главное, на что способен многомерный анализ — это создание и проверка качества классификации объектов. В умелых руках хорошая классификация очень полезна. Вот, например, мебельной фабрике потребовалось выяснить, какую мебель как лучше перевозить: в разобранном или в собранном виде. Рекомендации по перевозке зависят от многих причин (сложность сборки, хрупкость, стоимость, наличие стеклянных частей, наличие ящиков и полок и т.д.). Одновременно оценить эти факторы может лишь очень умелый человек. Однако существуют методы анализа, которые с легкостью разделят мебель на две группы, а заодно и проверят качество классификации, например, ее соответствие сложившейся практике перевозок.

Существует и другая классификация методов анализа данных. В ней все методы делятся на *предсказательные* и *описательные*. К первой группе методов относится все, что позволяет выяснить, с какой вероятностью может быть верным наш вывод. Ко второй — методы, которые «просто» сообщают информацию о нашей выборке без оценки ее применимости ко всей генеральной совокупности. В последние годы для все большего числа параметров находят способы их вероятностной оценки, и поэтому арсенал предсказательных методов все время растет.

Ответы на вопросы

[*Задача 1*] В этом случае шанс быть выбранными у елок выше, чем у дубов. Кроме того, лес может иметь какую-то структуру именно в выбранном направлении, и поэтому одной такой «диагонали» будет недостаточно для того, чтобы охарактеризовать весь лес. Чтобы улучшить данный метод, надо провести несколько «диагоналей», а расстояния между выбираемыми деревьями по возможности увеличить.

[*Задача 2*] Дело в том, что первыми вылезают самые активные особи, а чем активнее особь, тем быстрее она набирает на лапки смертельную дозу ядохимиката, и, стало быть, быстрее гибнет. Это и было нарушением принципа рандомизации. Кроме того, нарушался принцип повторности: в чашку последовательно сажали жука за жуком, что не могло не повлиять на исход опыта. Для того, чтобы поставить опыт правильно, надо было сначала подготовить (30 × число ядохимикатов) чашек, столько же листочков с бумагой, «случайным образом» распределить ядохимикаты по чашкам, а затем перемешать жуков в банке, достать соответствующее число и рассадить по чашкам.

Глава 2

Логика статистических тестов

2.1. Статистические гипотезы

Статистическая выборка должна быть **репрезентативной** (то есть адекватно характеризовать генеральную совокупность). Но как же мы можем знать, репрезентативна ли выборка, если мы не исследовали всю генеральную совокупность? Этот логический тупик называют **парадоксом выборки**. Хотя мы и обеспечиваем репрезентативность выборки соблюдением двух основных принципов ее создания (**рандомизации**, то есть случайного выбора объектов исследования, и **повторности**, то есть неоднократного исследования одного и того же эффекта), некоторая неопределенность все же остается.

Кроме того, если мы принимаем вероятностную точку зрения на происхождение наших данных (они получены путем случайного выбора объектов), то все дальнейшие суждения, основанные на этих данных, будут иметь вероятностный характер. Таким образом, мы никогда не сможем на основании нашей (репрезентативной!) выборки со стопроцентной уверенностью судить о свойствах генеральной совокупности. Мы можем лишь выдвигать гипотезы и вычислять их вероятность.

Великие философы науки (например, Карл Поппер) постулировали, что мы ничего не можем доказать, мы можем лишь что-нибудь опровергнуть (ломать — не строить). Предположим, что у нас есть 1000 фактов, подтверждающих какую-нибудь теорию.

Это не будет значить, что мы ее доказали! Вполне возможно, что 1001-ый (или 1 000 001-ый) факт опровергнет эту теорию. Поэтому при любом статистическом тесте выдвигаются две гипотезы. Одна — то, что мы хотим доказать (но не можем!) — называется **альтернативная гипотеза** (ее обозначают H_1).

Другая, противоречащая альтернативной гипотезе, — **нулевая гипотеза** (обозначается H_0). Нулевая гипотеза всегда является предположением об отсутствии какой-либо закономерности (например, зависимости одной переменной от другой или различия между двумя выборками). Стало быть, мы не можем доказать ни одну из этих гипотез. Мы можем лишь опровергнуть нулевую гипотезу и, следовательно, *принять* альтернативную. Если же мы не можем опровергнуть нулевую гипотезу, то мы вынуждены принять ее.

2.2. Статистические ошибки

Естественно, что когда мы делаем любые предположения (в нашем случае — выдвигаем статистические гипотезы), мы можем ошибаться (в нашем случае — делать статистические ошибки). Рассмотрим все четыре возможные ситуации, затрагивающие выборку и генеральную совокупность (ГС):

Для ГС / Для выборки	Верна H_0	Верна H_1
Принимаем H_0	Правильно!	Статистическая ошибка *второго рода*
Принимаем H_1	Статистическая ошибка *первого рода*	Правильно!

Если мы приняли для выборки H_0 (нулевую гипотезу) и она верна для генеральной совокупности, то мы правы, и все в порядке.

Аналогично и для H_1 (альтернативной гипотезы). Ясно, что мы не можем знать, что в действительности верно для генеральной совокупности, и сейчас просто рассматриваем все логически возможные варианты.

Если мы приняли для выборки альтернативную гипотезу, а она оказалась не верна для генеральной совокупности, то мы совершили так называемую **статистическую ошибку первого рода** (нашли несуществующую закономерность). Вероятность того, что мы совершили эту ошибку (так называемое **p-value**[1]), всегда отображается при проведении любых статистических тестов при помощи компьютерных программ. Очевидно, если вероятность этой ошибки достаточно высока, то мы должны отвергнуть альтернативную гипотезу.

Возникает естественный вопрос: какую вероятность считать достаточно высокой? Однозначного ответа на этот вопрос нет. В биологии принято соглашение считать пороговым значением 0,05 (то есть альтернативная гипотеза отвергается, если вероятность ошибки при ее принятии больше или равна 5%). В том случае, если исследователь хочет быть более уверенным в существовании найденной им закономерности (это особенно важно в медицине), он может принять пороговое значение равным 0,01 или даже 0,001.

Если мы принимаем нулевую гипотезу для выборки, в то время как для генеральной совокупности справедлива альтернативная гипотеза, то мы совершаем **статистическую ошибку второго рода** (не замечаем существующей закономерности). Значение этой ошибки характеризует так называемую **мощность статистического теста** (чем меньше вероятность статистической ошибки второго рода, то есть чем меньше вероятность не заметить существующую закономерность, тем более мощным является тест).

Решение о результатах статистических тестов принимается главным образом на основании вероятности статистической

[1]Более строгое определение: p-value — это вероятность получить тот же или больший эффект в том случае, если на самом деле верна нулевая гипотеза.

ошибки первого рода. Степень уверенности исследователя в том, что заключение, сделанное на основании выборки, будет справедливо и для генеральной совокупности, отражает **статистическая значимость**.

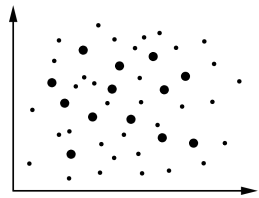
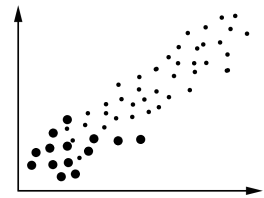
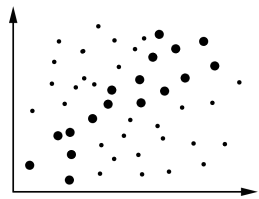
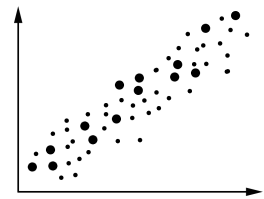
Допустим, если вероятность статистической ошибки первого рода равна 3%, то говорят, что найденная закономерность достоверна с вероятностью 97%. А если вероятность статистической ошибки первого рода равна, например, 23%, то говорят, что достоверной закономерности не найдено.

Как же возможна такая несправедливость? Почему вероятность одной ошибки гораздо важнее для нас, чем вероятность другой? В этом нет ничего необычного, такое часто встречается и в обычной жизни.

Представьте себе человека, который контролирует работу компьютерной системы слежения за межконтинентальными баллистическими ракетами. При малейшем подозрении на готовящийся ядерный удар он должен принять контрмеры. От его оценки ситуации зависит судьба всего мира. И вдруг сигнальные лампы замигали. Это может быть сигналом о начале атомной войны, в результате которой человеческая цивилизация окажется стертой с лица Земли, а может быть результатом технического сбоя. В этом случае цена ложной тревоги чрезвычайно высока.

Вот и в науке лучше не обратить внимания на реальную закономерность (раз она действительно существует, ее все равно потом кто-нибудь отыщет), чем обнаружить несуществующую (например, начать широкое внедрение лекарства, которое на самом деле не помогает от болезни).

Для полной ясности перерисуем нашу таблицу для конкретного случая — исследования связи между двумя признаками:

В ГС / В выборке	Связи нет	Связь есть
Связь не видна		
Связь видна		

(мелкие точки — объекты генеральной совокупности, ГС, крупные точки — объекты из нашей выборки)

Глава 3

Обработка данных

В этой главе рассказывается о том, как обрабатывать все данные, которые вы получили; об основных методах работы с программой R.

3.1. Как можно обрабатывать данные?

Можно обрабатывать данные вручную. Чертить графики на миллиметровке и вычислять значения статистических критериев по формулам на калькуляторе. Так делали до массового распространения компьютеров. В наше время поступать так было бы просто неразумно. Кроме того, человек не способен представлять себе многомерные (содержащие больше трех измерений) пространства, используемые при некоторых видах статистического анализа.

Можно воспользоваться программами общего назначения (например, MS Excel или LibreOffice Calc). Это не очень удобно, поскольку они, в общем-то, не предназначены для статистической обработки данных. Какие-то функции в таких программах отсутствуют, какие-то реализованы не очень удачно.

Самое разумное — воспользоваться специализированными компьютерными программами для статистической обработки данных. Они бывают двух типов. С привычным пользователям Windows интерфейсом меню и кнопок (например, STATISTICA),

и с привычной пользователям Linux командной строкой (например, R[1]). Различие между этими типами программ примерно такое же, как между автоматом для продажи напитков и рестораном.

Конечно, программы с интерфейсом меню и кнопок освобождают пользователя от необходимости осваивать новый командный язык и позволяют быстро провести многие стандартные виды статистической обработки данных. Зато программы с командной строкой предоставляют пользователю гораздо больше возможностей для обработки данных, такие программы имеют и специальный язык, что позволяет пользователю самостоятельно создавать новые алгоритмы обработки данных, отвечающие его потребностям.

В этом пособии мы будем рассказывать об обработке данных на примере программы R. Она работает не только под Windows, но и под macOS или Linux[2]. Помимо невероятной гибкости у этой программы есть еще одно важное преимущество — она не только постоянно обновляется коллективом компетентных специалистов, но и абсолютно бесплатна. Любой желающий может скачать последнюю версию программы с `http://www.r-project.org`.

Краткие указания, как тот или иной тип обработки данных проделать в R, мы будем приводить мелким шрифтом. В R все делается при помощи команд (обозначены **`жирным машинописным шрифтом`**), а команды работают с объектами (обозначены `светлым машинописным шрифтом`).

Подробный пример работы в программе R приведен в разделе 3.7. Сразу условимся, что ваши данные представлены объектом `data`. (Непонятно? Подробнее про то, как работает программа, можно прочесть в книге «R в действии» — см. список литературы

[1]Надо сказать, что и для R существует кнопочный интерфейс. Для этого нужно запустить пакет R Commander (см. ниже).

[2]В этой книге предполагается, что читатель работает под Windows; тем не менее большинство инструкций будет работать и в других операционных системах.

в конце книги). В отчете на R ссылаться нужно так, как выводит команда `citation()`.

Для того, чтобы запустить кнопочный интерфейс к R, нужно, чтобы у Вас был установлен пакет R Commander:

```
install.packages(Rcmdr)
```

Потом надо его вызвать: `library(Rcmdr)`, а если по какой-то причине окно R Commander закрыли, то повторно он вызывается командой `Commander()` — именно так, с большой буквы и с пустыми скобками. Интерфейс R Commander похож на интерфейс любой программы с интерфейсом меню и кнопок, только содержит гораздо меньше элементов (это потому, что для «серьезной» работы нужно все же пользоваться командной строкой).

3.2. Как начинать работу с данными?

Для начала надо понять, какими типами переменных (признаков) представлены наши данные. Выделяют три основных **типа переменных**:

1. **количественные** — представлены действительными числами: непрерывными (например, вес или время) или целыми (число телефонных звонков).

2. **порядковые** (шкальные) — представлены натуральными числами (например, школьная отметка), которые можно сравнивать в терминах «больше/меньше», но бессмысленно утверждать, что одно больше другого в определенное число раз.

3. **категориальные** (например, географический регион или индекс). Важно, что значения категориальных данных *не могут быть положены на числовую прямую* (почтовые индексы 119256 и 114729 нельзя сравнивать в терминах больше — меньше).

Потом надо решить, как распределены данные, как они «выглядят». Различают **нормальное распределение** данных (чем больше значение признака отличается от его среднего по выборке значения, тем реже это значение встречается в выборке) и **распределение данных, отличное от нормального** (рис. 1).

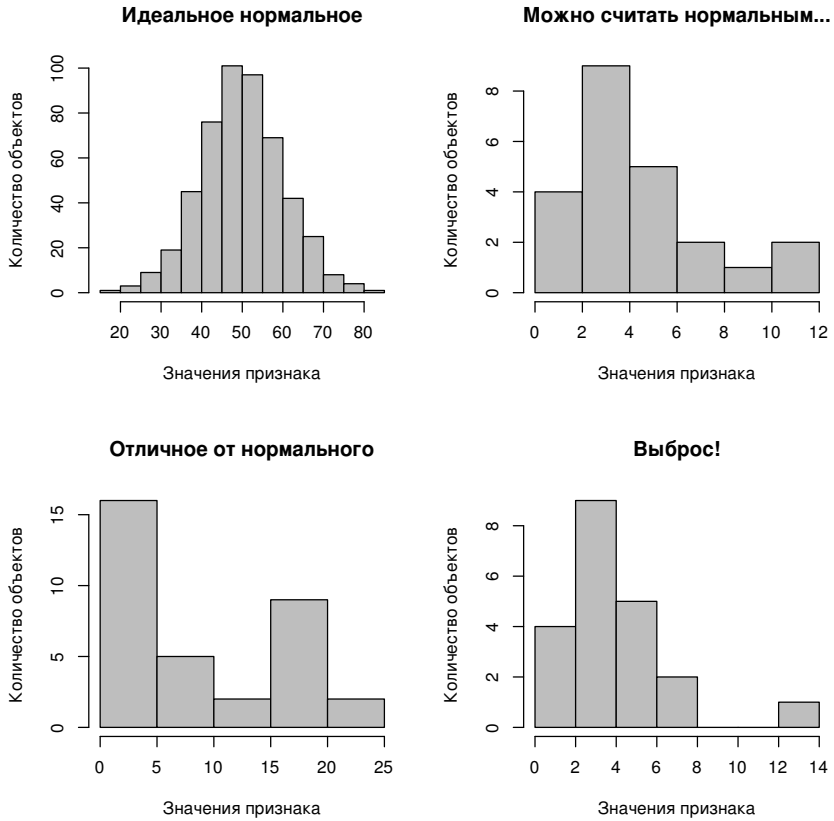


Рис. 1. Разные виды распределений данных.

Строго говоря, на сравнительно небольших выборках, с которыми мы обычно работаем, нормальное распределение данных практически не встречается. Однако данные, распределение которых не сильно отличается от нормального, при статистической обработке считают нормально распределенными. Что та-

кое «не слишком»? Четкого критерия не существует (некоторые формальные критерии, конечно, сформулированы, но они работают не слишком хорошо). В реальности каждый исследователь определяет это, опираясь на собственный опыт.

И, наконец, нужно выяснить, нет ли **пропущенных данных**, **выбросов** (значений, которые очень сильно отличаются от подавляющего большинства значений исследуемого признака), или просто опечаток.

Все эти вещи способны сильно помешать правильно проанализировать ваши данные. С опечатками все понятно. Учтите, кстати, что если вместо цифры вы случайно ввели букву или символ, то некоторые программы, воспринимают их как какое-нибудь (обычно довольно большое) число, а R воспримет весь столбец, содержащий такую ячейку, как текстовый и «откажется» его корректно обрабатывать. Есть еще очень зловредный тип опечаток, когда вместо русской буквы напечатана неотличимая от нее латинская (скажем, «е» вместо «е»), или наоборот. R считает такие буквы различными. Будьте внимательны!

Если в ваших данных есть выбросы, нужно проанализировать причину их происхождения. Выбросы, во-первых, могут быть теми же опечатками, во-вторых, они могут быть получены в результате нарушения запланированного хода сбора данных. Например, цель работы — исследовать артериальное давление у девятиклассников в спокойном состоянии. Понятно, что если какой-нибудь девятиклассник все время вертелся и подпрыгивал вместо того, чтобы сидеть спокойно, то его артериальное давление будет существенно выше, чем у остальных. В этих случаях выбросы, естественно, удаляют из данных. Что же делать, если выброс кажется «вполне нормальным» значением? Например, вы измеряли длину листьев берёзы, и все листья были 5–10 см длиной и тут вам попался лист 20 см длиной! Почему он вырос таким — тема для отдельного исследования, но из данных такое значение лучше все же исключить (только обязательно упомяните об этом в отчете), потому что оно мешает увидеть общую картину.

Наконец, мы добрались до пропущенных данных. Они тоже могут возникнуть по нескольким причинам. Допустим, вы решили измерять черешки листьев у разных видов растений. Вполне может получиться так, что у одного листа той же берёзы черешок совершенно случайно оторвется, когда вы будете измерять лист. В результате, черешок останется не измеренным. В ваших данных придется оставить пустую ячейку. Когда вы доберетесь до листьев осок, то черешка вы там не найдете вообще. Наконец, пропущенные данные могут появиться при удалении тех самых выбросов. Как же быть с получившимися пустыми ячейками?

В R пропущенные данные обозначаются как NA (от англ. not available — нет в наличии).

Есть несколько выходов (мы надеемся, что ваши данные представлены в виде таблицы, где столбцы — это исследуемые признаки, а строки — исследуемые объекты). Если пропущенных значений немного, и они принадлежат к разным признакам, можно просто удалить содержащие их строки[3]. Если у вас довольно много пропущенных значений находится в одном столбце, то можно удалить этот столбец. Если же ваши пропущенные значения в достаточном количестве распределены по всей таблице, можно попробовать заменить их на что-нибудь.

Вполне естественно в случае с черешками у листьев осок принять их длину равной 0 (то есть решить, что черешок у них как бы есть, но просто очень короткий, незаметный). Понятно, что такой подход нельзя применить к листу берёзы с оторванным черешком. Ведь его длина не равна нулю, мы просто не знаем, какая она! Здесь пропущенное значение можно заменить на среднее значение выборки, или что-нибудь вроде этого. Но такой подход нужно применять с большой осторожностью, потому что он может внести искажения в наши данные.

[3]Необязательно физически удалять строки из таблицы, можно указать это в параметрах анализа!

3.3. Выяснение общих характеристик

3.3.1. Объем выборки

Это число наблюдений (объектов) в вашей выборке. Его принято указывать в описании методики и/или результатов вашей работы.

В R нужно использовать команду **str**(). Например, **str**(data). Появится список всех переменных вашего файла данных, а начинаться он будет строчкой вроде этой: data.frame: 20 obs. of 2 variables. Это значит, что в ваших данных 20 наблюдений и 2 переменных.

3.3.2. Характеристики средней тенденции

3.3.2.1. Среднее арифметическое

Разумно применять для количественных данных с не слишком отличающимся от нормального распределением.

Используем команду **mean**(), например: **mean**(data).

Если в ваших данных имеются выбивающиеся из общей массы значения (которые сильно повлияют на среднее), можно посчитать среднее арифметическое без учета определенной части наименьших и наибольших значений. Вот, например, как посчитать среднее, «выкинув» по 10% значений «сверху» и «снизу»: **mean**(data, trim=0.2).

3.3.2.2. Медиана

Представьте себе, что все значения признака выписаны в строчку в порядке их возрастания. Медианой будет считаться то значение, которое стоит в строке посередине (а если признак имеет четное число значений, то медианой будет среднее арифметическое между двумя значениями, стоящими посередине). То

есть половина значений в выборке будет больше или равна медиане, а другая половина — меньше или равна медиане.

Четное число значений в выборке

Медиана равна (5 + 7)/2 = 6

1, 2, 3, 4, 4,	5, 7,	7, 7, 9, 15, 17
5 значений слева	два средних значения	5 значений справа

Нечетное число значений в выборке

Медиана равна 7

1, 2, 3, 4, 5,	7,	7, 9, 15, 17
5 значений слева	среднее значение	5 значений справа

Медиану разумно вычислять для порядковых данных или количественных данных, распределение которых сильно отличается от нормального.

Используем команду **median**(). Например, для первой переменной наших данных: **median**(data[,1]); а, если переменные имеют названия, то **median**(data$НАЗВАНИЕ).

3.3.2.3. Мода

Наиболее часто встречающееся значение в выборке. Пожалуй, единственная характеристика (если не считать объема выборки), которая может быть вычислена для категориальных данных. Кстати говоря, данные с одной модой называются **унимодальными**, а с двумя — **бимодальными**.

Используем команду **table**(). Например, для первой переменной наших данных: **table**(data[,1]). Появляется перечень с указанием того, сколько раз встретилось каждое значение выборки. Есть еще забавная команда **stem**(), которая отражает то же самое графически, но без построения графика. (А как? Узнайте сами).

3.3.3. Показатели вариации данных относительно среднего

Возьмем две выборки со средним арифметическим и медианой, равными 5.

Первая выборка: 5, 5, 5, 5, 5, 5, 5, 5, 5, 5.

Вторая выборка: 1, 2, 3, 4, 5, 6, 7, 8, 9.

Ясно, что они существенно различаются между собой. Значит, одних характеристик среднего значения недостаточно для описания выборки. Нужно не только знать среднее значение, но и понимать, насколько сильно удалены от него отдельные значения в выборке.

3.3.3.1. Минимальное и максимальное значение

Разумно использовать в отсутствие выбросов.

Используем команды **min**() и **max**() соответственно. Например, наименьшее значение первой переменной наших данных: **min**(data[,1]).

3.3.3.2. Нижняя и верхняя ква́ртили

Если при помощи медианы мы «разбивали» нашу выборку пополам, то квартили «разделяют» ее на четыре равные части. Граница между первой и второй (в порядке возрастания) частями называется **нижней квартилью**, а между третьей и четвертой — **верхней квартилью**. Диапазон значений между нижней

и верхней квартилями называется **межквартильным размахом**.

```
20  максимум
18
17
15  верхняя квартиль
15
12
 9
 8  медиана
 6
 6
 6
 5  нижняя квартиль
 4
 2
 1  минимум
```

Разумно использовать для выборок с большим числом выбросов.

Используем команду **summary**(). Например, для первой переменной наших данных: **summary**(data[,1]). Появляется таблица, в которой последовательно указаны: минимум, нижняя квартиль, медиана, среднее арифметическое, верхняя квартиль, максимум (и число пропущенных значений, если они есть). Очень удобно!

3.3.3.3. Среднее квадратичное отклонение

Этот параметр вычисляется так. Отклонения каждого значения в выборке от среднего арифметического возводятся в квадрат (чтобы избежать отрицательных значений) и суммируются. Полученная сумма делится на число значений в выборке (чтобы можно было сравнивать выборки разного объема). Из получившегося числа извлекается квадратный корень, чтобы получить такую же размерность, как у значений выборки. В научных публикациях принято указывать значение среднего квадратичного отклонения всякий раз, когда вы упоминаете среднее арифметическое значение выборки.

Используем команду **sd**(). Например, для первой переменной наших данных: **sd**(data[,1]).

г) коэффициент вариации

Это — безразмерная величина, которая применяется для сравнения изменчивости признаков, имеющих разные единицы измерения. Она вычисляется как отношение среднего квадратичного отклонения к среднему арифметическому, умноженное на 100%.

В R специальной команды для вычисления этого коэффициента нет, но мы можем легко вычислить его по формуле. Например, для первой переменной наших данных: 100***sd**(data[,1])/**mean**(data[,1]). А можно такую команду сделать самому:

```
kov <- function(x) {100*sd(x)/mean(x)}
```

После этого можно писать: **kov**(data[,1]).

3.4. Визуальный анализ данных

Существует множество разнообразных способов графического представления данных. Ниже перечислены несколько наиболее часто употребляемых и полезных типов диаграмм (все они двухмерные).

3.4.1. Гистограмма

Гистограммы вы уже неоднократно видели, когда мы обсуждали типы распределения данных и выбросы. По оси абсцисс (горизонтальной!) указаны значения (или интервалы значений) признака, а по оси ординат (вертикальной!) указано, сколько раз в выборке встречаются такие значения признака (значения признака, попадающие в указанный интервал). Можно выводить

на график как абсолютное число значений, так и долю от общего числа значений в выборке. Число интервалов, на которые разбивается весь диапазон значений признака, можно регулировать самостоятельно. Обычно этот тип графика используется так, как мы использовали его в этом пособии (исследование типа распределения данных и поиск выбросов) или для общей характеристики большой выборки, когда мы хотим сравнить частоты встречаемости разных значений (разных интервалов значений) между собой.

Используем команду **hist**(). Например, гистограмма с десятью интервалами для первой переменной наших данных: **hist**(data[,1], breaks=10). В принципе, R и сам довольно «умен» и может вычислить оптимальное число интервалов для гистограммы. Так что можно просто написать **hist**(data[,1]).

Часто в качестве своеобразного аналога гистограммы для характеристики выборки с небольшим числом значений используется «пирог» (круг, разделенный на сектора, которые соответствуют значениям признака). Площадь секторов пропорциональна частоте соответствующего значения в выборке. Учтите, что из-за особенностей зрительного восприятия сложно понять, какую именно часть круга занимает тот или иной сектор. Например, наше впечатление будет зависеть от цвета, в который покрашен рассматриваемый сектор и его «соседи». По этой причине диаграммы такого типа не принимаются в печать рядом издательств, и мы не рекомендуем их использовать.

Несмотря на то, что в справке R так прямо и написано: «„пирог”» — очень плохой способ визуализации данных», можно все-таки построить такую диаграмму, используя команду **pie**(). Например, для первой переменной наших данных: **pie**(data[,1]). Для замены «пирогов» Кливлендом был придуман точечный график: **dotchart**(data[,1]).

3.4.2. Диаграмма рассеяния

Эта диаграмма используется для исследования связи между двумя переменными. По оси абсцисс откладывают значения од-

ной переменной, по оси ординат — второй. Отдельные объекты изображают в виде точек с координатами, соответствующими значениям этих двух переменных для конкретного объекта.

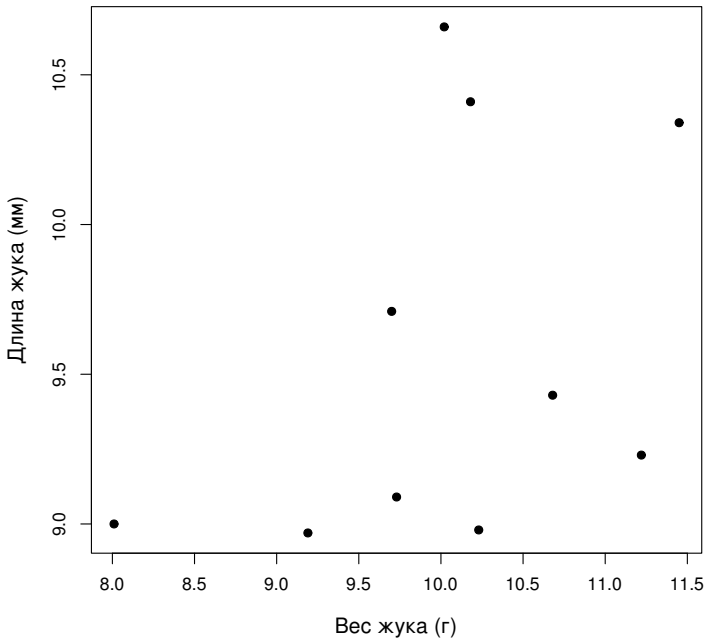


Рис. 2. Диаграмма рассеяния (scatterplot).

Используем команду **plot**(). Например, отложим значения третьей переменной наших данных по оси абсцисс, а четвертой — по оси ординат: **plot**(data[,3], data[,4]).

Команда **sunflowerplot**() позволяет отразить число наложившихся друг на друга точек.

3.4.3. Линия

Этот график строят по тому же принципу, что и диаграмму рассеяния, только точки соединяются отрезками в порядке их расположения в таблице данных.

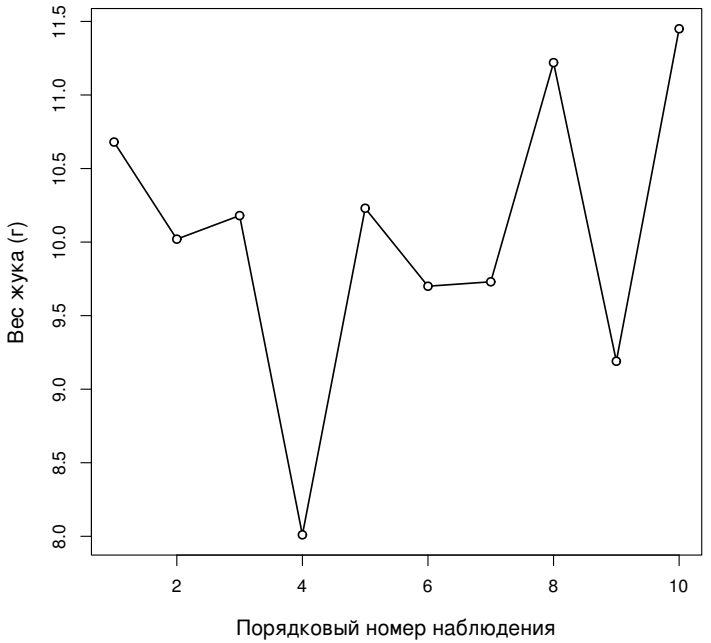


Рис. 3. График-линия (linechart).

Такие графики имеет смысл использовать для исследования изменения значений признака во времени (динамика погодных условий, поведение животных и т.п.). В приведенном выше примере на графике изображено изменение веса жука на протяжении ряда наблюдений.

Используем команду **plot**(x, y, type="o"). Например, отложим номера строк по оси абсцисс, а значения третьей переменной — по оси ординат: **plot**(**row.names**(data), data[,3], type="o").

3.4.4. Диаграмма размаха («ящик с усами»)

Очень наглядное изображение основных характеристик выборки (см. ниже рис. 4): минимум, нижняя квартиль, медиана, верхняя квартиль, межквартильный размах, максимум. Еще отобра-

жаются выбросы — значения, которые сильно «выбиваются из общего ряда»[4].

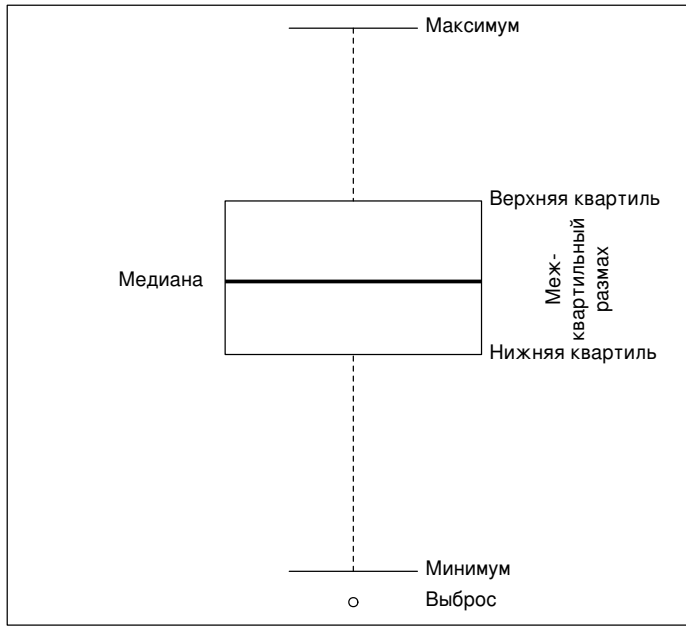


Рис. 4. Диаграмма размаха («ящик с усами», boxplot).

Особенно удобно при помощи «ящиков с усами» графически сравнивать значения признака для двух групп.

Используем команду **`boxplot`**`()`. Например, для первой переменной наших данных можно написать: **`boxplot`**`(data[,1])`; чтобы сравнить первую и вторую переменные, можно ввести:
`boxplot``(data[,1], data[,2])`.

Можно сделать и так, чтобы ширина «ящиков» была пропорциональна числу наблюдений в группе: **`boxplot`**`(data[,1], varwidth=TRUE)`.

[4]По умолчанию: больше, чем на полтора межквартильных размаха отстоят от нижней или верхней квартили. Этот параметр можно изменить. Как? Чтобы узнать это, наберите в командной строке **`help`**`(boxplot)`.

3.5. Статистические тесты

3.5.1. Общие соображения

Статистических тестов существует, наверное, так же много, как и типов диаграмм. Главное — понять логику их проведения, что мы и сделаем на примере нескольких самых распространенных и полезных, на наш взгляд, тестов. Тогда вы сможете без труда освоить любые статистические тесты, которые вам понадобятся в дальнейшем.

Все статистические тесты делятся на две большие группы: параметрические тесты и непараметрические. **Параметрические тесты** предназначены для обработки так называемых **параметрических данных**. Чтобы данные считались параметрическими, должно одновременно выполняться три условия:

1. распределение данных близко к нормальному;
2. выборка содержит не менее 30 наблюдений;
3. это количественные данные.

Если хотя бы одно из этих условий не выполняется, данные считаются **непараметрическими** и обрабатываются **непараметрическими тестами**. Несомненным достоинством непараметрических тестов является, как ни банально это звучит, их способность работать с непараметрическими (то есть с той или иной стороны «неидеальными») данными. Зато параметрические тесты имеют *бóльшую мощность* (то есть при прочих равных вероятность не заметить существующую закономерность выше при использовании непараметрических тестов).

Этому есть простое объяснение. Дело в том, что порядковые (непараметрические) данные имеют свойство «скрывать» имеющиеся различия, объединяя отдельные значения в группы. Здесь нужно заметить, что мы вообще-то можем применять

непараметрические тесты к параметрическим данным и получать корректные результаты (только вероятность статистической ошибки второго рода будет больше, чем при параметрических тестах). А вот применение параметрических тестов к непараметрическим данным будет некорректным. Здесь можно сравнить параметрические тесты с консервным ножом, а непараметрические — с топором. Конечно, консервы лучше открывать консервным ножом, но в случае отсутствия такового, топором открыть банку тоже можно, зато нарубить дрова консервным ножом вам вряд ли удастся.

Обычно стараются работать с параметрическими данными, чтобы иметь возможность применять более мощные параметрические тесты. На распределение данных мы, естественно, никак повлиять не можем[5]. Что мы можем сделать, так это постараться иметь достаточно большой объем выборки (так мы еще и увеличим ее репрезентативность), а также работать с количественными данными.

Как сделать так, чтобы ваши данные были количественными, а не порядковыми? Помните, что *категориальные данные напрямую не могут быть обработаны никаким видом статистических тестов* и должны быть преобразованы в порядковые или количественные! Можно соответствующим образом спланировать сбор данных. Например, при исследованиях размеров листьев не делить их визуально на «маленькие», «средние» и «большие», а измерить их длину и ширину при помощи линейки (на этом примере ясно, что *количественные данные содержат больше информации, чем порядковые*).

Однако иногда сбор количественных данных требует использования труднодоступного оборудования и сложных методик (например, если вы решите исследовать окраску цветков как количественную переменную, вам понадобится спектрофотометр для измерения длины волны отраженного света — количествен-

[5] Можно, однако, данные *преобразовать*, например, вычислить квадратный корень или логарифм. После преобразования распределение данных может стать близким к нормальному.

ного выражения видимого цвета). В этом случае можно выйти из положения путем *перекодирования данных* на стадии их обработки.

Например, цвет можно закодировать в значениях красного, зеленого и синего каналов компьютерной цветовой шкалы RGB. Приведем еще один пример перекодирования. Предположим, вы изучаете высоту зданий в различных городах земного шара. Можно в графе «город» написать его название (категориальные данные). Это, конечно, проще всего, но тогда вы не сможете использовать эту переменную в статистическом анализе данных. Можно закодировать города цифрами в порядке их расположения, например, с севера на юг (если вас интересует географическая изменчивость высоты зданий) — получатся порядковые данные, которые можно обработать непараметрическими методами. И, наконец, каждый город можно обозначить его географическими координатами или расстоянием от самого южного города — тогда мы получим количественные данные, которые (при наличии достаточного числа наблюдений и не слишком отличного от нормального распределения!) можно обработать параметрическими методами.

3.5.2. Различаются ли достоверно выборки?

При ответе на этот вопрос при помощи описываемых ниже статистических тестов нужно всегда помнить, что эти тесты проверяют только различия по средним значениям, подразумевая, что разброс данных в выборках примерно одинаков. Например, уже упоминавшиеся выборки с одинаковыми параметрами средней тенденции и разными показателями разброса данных относительно нее:

```
1, 2, 3, 4, 5, 6, 7, 8, 9
```

и

```
5, 5, 5, 5, 5, 5, 5, 5, 5
```

не будут различаться по результатам описываемых ниже тестов. Конечно, существуют тесты, которые анализируют различие в разбросе данных относительно среднего, но они используются довольно редко, и здесь мы их касаться не будем.

3.5.2.1. Две выборки

Рассмотрим сначала со всех сторон наиболее часто встречающийся вариант вынесенного в заголовок этого подраздела вопроса: различаются ли достоверно две выборки. Как вы помните, для проведения статистического теста нам нужно выдвинуть две статистические гипотезы. Нулевая гипотеза: различий между (двумя) выборками нет. Альтернативная гипотеза: различия между (двумя) выборками есть.

Напоминаем, что ваши данные должны быть организованы в виде таблицы со строками-наблюдениями и столбцами-признаками. Выборки должны занимать отдельные столбцы. Например, если вы хотите узнать, различается ли достоверно рост мужчин и женщин, то в одном столбце должен быть указан рост мужчин, а в другом — рост женщин (каждая строчка — это один обследованный человек). Можно организовать свои данные и по-другому: в одном столбце указан условный номер группы, к которому относится объект (например, «0» — женщина, «1» — мужчина), а в другом столбце — значение признака для данного объекта (пример такой организации данных приведен в разделе 3.7).

Если наши данные параметрические, то нужно провести **параметрический тест Стьюдента**. Здесь есть одна тонкость. Если переменные, которые мы хотим сравнить, были получены на *разных* объектах (например, чтобы измерить рост мужчины и рост женщины, нужно как минимум два объекта — мужчина и женщина), мы будем использовать тест Стьюдента для *независимых переменных*. Если пары сравниваемых характеристик были получены на *одном* объекте (например, частоту пульса до и

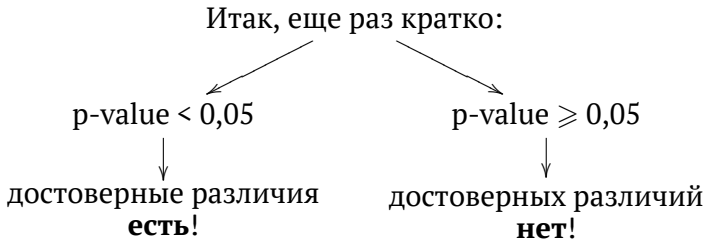
после физической нагрузки измеряли у одного и того же человека), мы будем использовать тест Стьюдента для *зависимых переменных*.

Тест для зависимых переменных более мощный. Дело здесь вот в чем. Представьте себе, что мы измеряли пульс до нагрузки у одного человека, а после нагрузки — у другого. Тогда не совсем ясно, как объяснить полученную разницу: может быть, частота пульса увеличилась после нагрузки, а может быть, этим двум людям вообще свойственна разная частота пульса. В случае же «двойного» измерения пульса каждый человек как бы является своим собственным контролем, и разница между сравниваемыми переменными (до и после нагрузки) обуславливается только тем фактором, на основе которого они выделены (наличием нагрузки).

Если же мы имеем дело с непараметрическими данными, то нам нужно провести **непараметрический тест Вилкоксона**[6]. У теста Вилкоксона тоже есть два варианта — для зависимых и для независимых переменных.

В любом случае нас интересует значение параметра p-value (или просто p). Этот параметр показывает вероятность статистической ошибки первого рода (вероятность найти несуществующую закономерность). Если эта вероятность больше или равна 0,05, мы не в состоянии отвергнуть нулевую гипотезу об отсутствии отличий между выборками и поэтому обязаны ее принять. Если же p-value меньше 0,05, мы должны отвергнуть нулевую гипотезу и тем самым принять альтернативную гипотезу о существовании различий между выборками.

[6]Или тест Манна-Уитни (это то же самое).



Используем команды **t.test**() и **wilcox.test**(). Узна́ем, есть ли достоверное различие между первой и второй переменными наших данных. Тест Стьюдента для независимых переменных:

```
t.test(data[,1], data[,2], paired=FALSE)
```

Тест Стьюдента для зависимых переменных:

```
t.test(data[,1], data[,2], paired=TRUE)
```

По умолчанию в R используется вариант теста Стьюдента, не требующий предположения о равенстве стандартных отклонений выборок.

Тест Вилкоксона:

```
wilcox.test(data[,1], data[,2])
```

В любом случае появится текст, где значение p-value будет, скорее всего, указано в третьей строчке. Если наши данные организованы подругому, и первая переменная — номер группы, а вторая — значения признака, то сравнить две группы можно так:

```
wilcox.test(data[,2] ~ data[,1])
```

Как вы помните, различия между выборками хорошо иллюстрировать при помощи диаграммы размаха. Считается, что если «ящики» (межквартильный размах) перекрываются более чем

на треть своей высоты, то выборки, скорее всего, достоверно не различаются.

А что если нам понадобится проанализировать различия между двумя выборками, значения которых представлены только нулями и единицами? Например, можно задаться вопросом: правда ли, что есть достоверная разница между частотой забывания сменной обуви мальчиками и девочками?

Можно завести две колонки — одну для мальчиков, другую для девочек — и ставить в соответствующую колонку «0», если ученик (ученица) явились в школу без сменной обуви, и «1», если он (она) принесли сменную обувь. Конечно же, мы получим непараметрические данные, которые будут анализироваться непараметрическим тестом **хи-квадрат**. Формулировка нулевой и альтернативной гипотез, а также ход рассуждений при выборе гипотезы аналогичны описанным выше.

Используем команду **chisq.test()**. Пусть данные для мальчиков и девочек будут первой и второй переменными наших данных:

```
chisq.test(data[,1], data[,2])
```

Значение p-value указано обычно в третьей строчке появившегося текста.

Представим себе, что больше половины мальчиков пришли в школу без сменной обуви. Интересно, это случайно так получилось, или они намеренно оставляли сменку дома? Иными словами, мы хотим узнать, достоверно ли отличается доля мальчиков без сменки от 50%. Здесь нам поможет **тест пропорций**.

Используем команду **prop.test()**:

```
prop.test(c(sum(data[,1])), c(length(data[,1])), 0.5)
```

Здесь **sum()** — функция, суммирующая все свои аргументы, а **length()** — функция, которая подсчитывает число значений (объек-

тов). Наконец, `c()` — функция, превращающая свои аргументы в последовательность чисел, так называемый вектор.

Этот же тест пропорций поможет нам выяснить, правда ли девочки внимательнее мальчиков (то есть доля девочек со сменкой достоверно отличается от доли таких же мальчиков).

Используем ту же функцию (обратите внимание на правильную расстановку скобок!):

```
prop.test(c(sum(data[,1]), sum(data[,2])),
        c(length(data[,1]), length(data[,2])))
```

3.5.2.2. Три выборки и больше

А что если теперь мы захотим узнать, есть ли различия между *тремя* выборками? Первое, что приходит в голову (предположим, что это параметрические данные) — это провести серию тестов Стьюдента: между первой и второй выборками, между первой и третьей и, наконец, между второй и третьей — всего три теста. К сожалению, число необходимых тестов Стьюдента будет расти чрезвычайно быстро с увеличением числа интересующих нас выборок. Например, для попарного сравнения шести выборок нам понадобится провести уже 15 тестов! А представляете, как обидно будет провести все эти 15 тестов только для того, чтобы узнать, что все выборки не различаются между собой!

Но главная проблема заключена не в сбережении труда исследователя. Дело в том, что при многократном проведении статистических тестов, основанных на вероятностных понятиях, на одной и той же выборке вероятность обнаружить достоверную закономерность *по ошибке* возрастает. Допустим, мы считаем различия достоверными при p-value < 0,05. При этом мы «позволяем себе» ошибаться (находить различия там, где их нет) в

четырех случаях из 100 (в одном случае из 25). Понятно, что если мы одновременно проведем 25 статистических тестов на одной и той же выборке, то, скорее всего, в одном случае найдем различия просто по ошибке.

Похожие рассуждения можно применить к экстремальным видам спорта. Например, вероятность того, что парашют не раскроется при прыжке, довольно мала (допустим, 0,01), и странно бы было ожидать, что парашют не раскроется как раз тогда, когда человек прыгает впервые. При этом любой десантник, имеющий опыт нескольких сотен прыжков, может рассказать несколько захватывающих историй о том, как ему пришлось использовать запасной парашют.

Поэтому для сравнения трех и более выборок используется (однофакторный) **дисперсионный анализ** (ANOVA, от английского ANalysis Of VAriance). Нулевая гипотеза: выборки не различаются между собой. Альтернативная гипотеза: *хотя бы одна* пара выборок различается между собой. Обратите внимание на формулировку альтернативной гипотезы! Результаты этого теста будут одинаковыми в случае, если различается только одна пара выборок, и в случае, если все выборки различаются между собой.

Ваши данные должны быть организованы как две переменные, в одной из которых указаны все значения всех сравниваемых выборок (например, рост брюнетов, блондинов и шатенов), а в другой — номера групп, к которым принадлежат значения первой переменной (например, будем ставить напротив значения роста брюнета «1», напротив роста блондина «2» и напротив роста шатена «3»).

Для проведения однофакторного дисперсионного анализа используем команду **`oneway.test`**`()` (у нее меньше ограничений, чем у других методов ANOVA). Если в первом столбце указан рост людей, а во втором — их цвет волос, то нужно написать:

```
oneway.test(data[,1] ~ data[,2])
```

Если данные непараметрические, то нужно использовать тест Краскела-Уоллиса:

```
kruskal.test(data[,1] ~ data[,2])
```

В появившихся результатах теста нас, конечно же, интересует p-value. Если оно больше или равно 0,05, то все выборки не различаются между собой, и говорить тут больше не о чем. Если же оно меньше 0,05, то, по крайней мере, одна пара выборок различается. А может быть, две? А может быть, все выборки различаются между собой? Узнать это мы можем при помощи специальной версии попарного теста Стьюдента (или Вилкоксона — для непараметрических данных), где применяется поправка на множественные сравнения. Мы увидим таблицу, где будут указаны p-value для всех пар выборок. Естественно, что те пары выборок, p-value для которых меньше 0,05, достоверно различаются между собой.

Для параметрических и непараметрических данных нужно использовать `pairwise.t.test()` и `pairwise.wilcox.test()` соответственно. Например, для теста Стьюдента:

```
pairwise.t.test(data[,2], data[,1])
```

3.5.3. Есть ли достоверная линейная связь между переменными?

Мерой линейной взаимосвязи между переменными является **коэффициент корреляции** (обозначается латинской буквой *r*). Значения коэффициента корреляции могут варьировать по модулю от нуля до единицы. Нулевой коэффициент корреляции говорит нам о том, что значения одной переменной совершенно не связаны со значениями другой переменной.

Коэффициент корреляции, равный по модулю единице, свидетельствует о четкой линейной связи между переменными (все

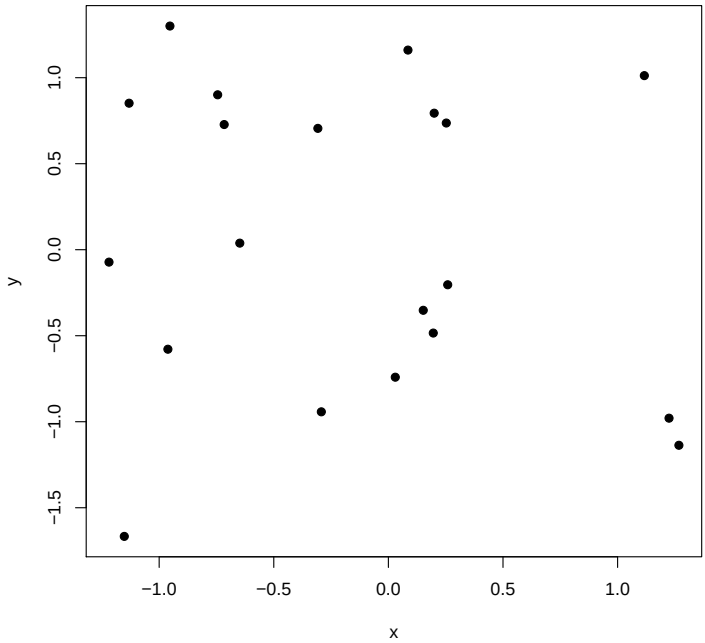


Рис. 5. Диаграмма рассеяния при коэффициенте корреляции, близком к нулю.

наблюдения ложатся на прямую $y = ax + b$, где x и y — наши переменные, a и b — числовые коэффициенты).

Положительный коэффициент корреляции свидетельствует о положительной связи (чем больше, тем больше), отрицательный — об отрицательной (чем больше, тем меньше).

Казалось бы, из определения коэффициента корреляции следует, что если, например, он увеличится в два раза (по модулю), то и степень взаимосвязи между переменными тоже возрастет вдвое. Однако это не так. На самом деле степень взаимосвязи между переменными как таковую отражает **коэффициент детерминации** (это коэффициент корреляции, возведенный в квадрат). Эта величина показывает, какая доля изменений зна-

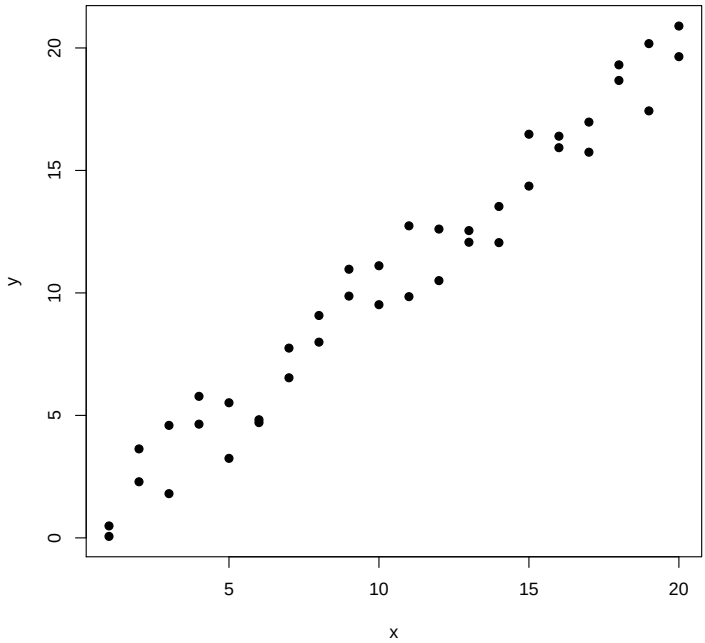


Рис. 6. Диаграмма рассеяния при коэффициенте корреляции, близком к единице.

чений одной переменной сопряжена с изменением значений другой переменной.

Значит, если коэффициент корреляции равен 0,4, то значения переменных сопряженно изменяются в 16% случаев (потому что $0{,}4^2 = 0{,}16$), а если коэффициент корреляции увеличится вдвое (0,8), то степень взаимосвязи между переменными возрастет в четыре раза ($0{,}8^2 = 0{,}64$).

Напоминаем, что обсуждаемый здесь коэффициент корреляции характеризует меру **линейной** связи между переменными. Две переменных могут быть очень четко взаимосвязаны, но если эта связь не линейная, а допустим, параболическая, то коэффициент корреляции будет близок к нулю.

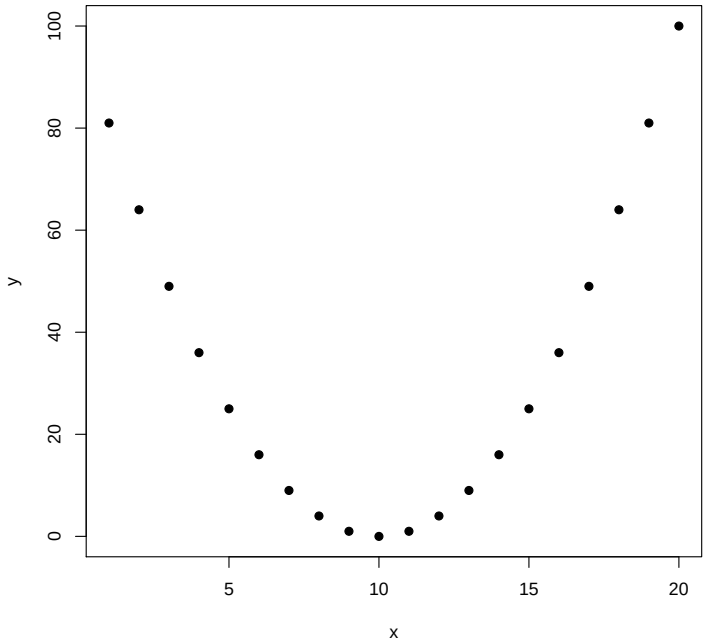


Рис. 7. Корреляция есть, но поскольку зависимость нелинейная, то коэффициент линейной корреляции близок к нулю.

Примером такой связи может служить влияние степени возбужденности человека на качество решения им математических задач. Ясно, что очень слабо возбужденный человек (засыпающий) и очень сильно возбужденный (во время футбольного матча) будет решать задачи гораздо хуже, чем умеренно возбужденный человек (на хорошо организованном уроке).

Поэтому прежде, чем оценить взаимосвязь численно (вычислить коэффициент корреляции), нужно посмотреть на ее графическое выражение (лучше всего здесь использовать диаграмму рассеяния). Существуют некоторые методы количественной оценки нелинейной связи между переменными, но здесь мы не будем их касаться.

Обращаем ваше внимание также на то, что речь идет о связи между переменными, а не о зависимости одной переменной от другой. Если мы нашли достоверную связь между переменными А и Б, то это может значить, что: (1) А зависит от Б, (2) Б зависит от А, (3) А и Б зависят друг от друга, (4) А и Б зависят от какой-то третьей переменной В, а между собой не имеют ничего общего. Например, хорошо известно, что существует четкая положительная связь между объемом продаж мороженого и количеством пожаров. Странно было бы предположить, что поедание мороженого располагает людей к небрежному обращению с огнем или что созерцание пожаров возбуждает тягу к мороженому. Все гораздо проще — оба этих параметра положительно зависят от температуры воздуха!

Итак, нулевая гипотеза: линейной связи между переменными нет. Альтернативная гипотеза: линейная связь между переменными есть.

Если данные параметрические, мы будем пользоваться **параметрическим коэффициентом Пирсона**. Если же наши данные непараметрические, то мы будем пользоваться **непараметрическим коэффициентом Спирмена**. В общем-то нам достаточно обратить внимание на все то же значение p-level (вероятность найти несуществующую закономерность).

Логика рассуждений здесь абсолютно такая же, как и в тестах на существование достоверных различий, и во всех прочих статистических тестах. Если эта вероятность (p-value) больше или равна 0,05, мы вынуждены отвергнуть альтернативную гипотезу и принять нулевую об отсутствии линейной связи между переменными. Если p-value меньше 0,05, мы должны принять альтернативную гипотезу о существовании линейной связи между переменными.

Итак, если p-value $\geqslant$ 0,05 — достоверной линейной связи между переменными нет, p-value < 0,05 — достоверная линейная связь есть! Надо сказать, что в отчетах и научных статьях наряду со значением p-value принято указывать и значение коэффициента корреляции.

В R используем команду **cor.test**(). Узнаем, есть ли достоверная связь между первой и второй переменными наших данных.

Коэффициент Пирсона:
cor.test(data[,1], data[,2], method="pearson").

Можно method="pearson" и не писать, потому что это так называемое значение по умолчанию.

А вот другие методы указывать надо, например, для коэффициента Спирмена: **cor.test**(data[,1], data[,2], method= "spearman").

Значение коэффициента корреляции указано в последней строчке вывода команды. Есть и способ вычисления корреляции сразу между несколькими парами переменных — функция **cor**().

3.6. Стандартная процедура статистического анализа

В заключение приведем рекомендуемый порядок проведения статистического анализа данных:

1. формулировка биологической задачи (надо решить, что вы хотите узнать, например, есть ли различие между выборками, есть ли связь между величинами);

2. выбор способа статистической обработки данных. Не забывайте сначала определить тип ваших данных (см. раздел «Как начать работу с данными») и провести предварительный графический анализ данных, в том числе проверить их на отсутствие выбросов и опечаток;

3. статистическая процедура (формулировка нулевой и альтернативной гипотезы, проведение расчетов, формулировка статистических выводов — какую гипотезу вы принимаете);

4. биологическая интерпретация результата.

3.7. Пример использования команд в программе R

Для того чтобы помочь начинающим пользователям, ниже мы помещаем подробный пример сессии R с краткими пояснениями. Все приведенные ниже команды будут относиться к файлу данных о десяти воображаемых жуках, состоящему из четырех столбцов (признаков): пол жука (POL: самки = 0 и самцы = 1), цвет жука: (CVET: красный=1, синий=2, зеленый=3), вес жука в граммах (VES) и длина жука в миллиметрах (ROST).

```
POL CVET  VES   ROST
0   1   10.68   9.43
1   1   10.02   10.66
0   2   10.18   10.41
1   1   8.01    9
0   3   10.23   8.98
1   3   9.7     9.71
1   2   9.73    9.09
0   3   11.22   9.23
1   1   9.19    8.97
1   2   11.45   10.34
```

Для тренировки вы можете перенести эти данные в электронную таблицу и выполнить приведенные ниже команды.

1. Начинаем

1.0. Создаем на жестком диске рабочую директорию (папку); копируем в нее файл данных в текстовом формате (делается из Excel-файла командой «Сохранить как...», значения из разных столбцов разделены знаком «;» – проверьте это, открыв полученный файл в любом текстовом редакторе, например, в «Блокноте»).

Часто Excel создает какие-то дополнительные пустые ячейки, которые выглядят как ряды из точек с запятой (;;;;;;;) при просмотре файла в текстовом редакторе. Это вызывает проблемы при чтении файла

данных в R, поэтому такие накладки нужно заранее отслеживать и исправлять.

1.1. Открываем программу R. Указываем в меню папку (директорию), где находится ваш файл данных:

```
Файл -> Изменить папку
```

1.2.1. Читаем файл данных (создаем в памяти программы объект под названием `data`, который представляет собой копию вашего файла данных). В строке ввода набираем (пусть ваш файл называется `zhuki.txt`):

```
data <- read.table("zhuki.txt", h=TRUE, sep=";")
```

... и нажимаем Enter (эту клавишу нужно нажимать каждый раз после ввода команды, при этом курсор может стоять в любом месте строки, не обязательно в ее конце).

Не ждите каких-то «похвал» от программы. Если не появилось сообщение об ошибке, то все прошло успешно.

ВНИМАНИЕ! Будьте осторожны со скобками, знаками препинания (даже типами кавычек) и регистрами (в R `data` и `DATA` — это разные объекты).

1.2.2. В R по умолчанию десятичный разделитель — точка (то есть $^1/_{10}$ записывается как 0.1, а не 0,1). Если в вашем файле разделитель — запятая, есть способ загрузить и такие данные, указав дополнительный параметр:

```
data <- read.table("zhuki2.txt", h=TRUE, sep=";",
        dec=",")
```

Параметр `h=TRUE` (или `head=T`) «сообщает» программе, что первая строка данных — это еще не сами данные, а названия столбцов. Добавив в команду **`read.table`**`()` параметр `row.names=1`, вы укажете, что в первом столбце файла данных содержатся названия объектов.

1.3. Посмотрим на ваш файл данных:

```
data
```

Если файл данных очень большой, достаточно будет посмотреть на его первые несколько строк:

```
head(data)
```

Внутри R вносить изменения в данные не очень удобно. Разумно вносить их в текстовый файл данных (открыв его, например, в Excel), а потом заново читать его в R.

1.4. Посмотрим на структуру файла данных. Сколько объектов («obs.» = observations), сколько признаков (variables), как названы признаки и в каком порядке они следуют в таблице:

```
str(data)
```

Кстати говоря, чтобы получить подсказку о любой команде (на английском языке!), введите `help(название_команды)` или просто `?название_команды`.

1.5.1. Создадим в памяти еще один объект с данными (назовем его `data.f`), куда отберем строки только для самок:

```
data.f <- data[data$POL == 0,]
```

Запятая в конце квадратных скобок и означает, что мы отбираем строки. Если бы нам понадобилось отобрать столбцы (скажем, первый и третий), то мы бы написали `data[,c(1,3)]`, поставив запятую в начале квадратных скобок. «Склеивать» столбцы (объединяя их в одну таблицу) можно так: `cbind(data[,1], data[,3])` — на примере первого и третьего столбцов. А вот как «склеивать» строки: `rbind(data[1,], data[3,])`.

1.5.2. А теперь создадим отдельный объект с данными для крупных (больше 10 мм) самцов:

```
data.m.big <- data[data$POL == 1 & data$ROST > 10,]
```

Кстати, эту команду проще не вводить заново, а получить путем редактирования предыдущей команды (обычное дело в R). *Для вызова предыдущей команды воспользуйтесь «стрелкой вверх» на клавиатуре.*

Другая очень полезная клавиша — это Tab, «табуляция». Она предлагает возможные варианты названий команд, аргументов и даже названий файлов на диске. Попробуйте начать набирать какую-нибудь команду, а затем нажать Tab (иногда нужно нажать два раза).

Использованные знаки `==` и `&` — это, соответственно, логические выражения «таких, что» и «и». Именно они служат критериями отбора. Еще есть полезный логический оператор `|` (или).

1.5.3. Отобрать крупных самцов можно и более удобным способом:

```
data.m.big <- subset(data, POL == 1 & ROST > 10,
        select=c(VES, ROST))
```

Для более впечатляющей демонстрации возможностей команды **`subset`**`()` мы отобрали не только строки (крупные самцы), но и столбцы (вес и рост, параметр `select`). Всю команду нужно набирать целиком, просто в книге она не влезла на одну строчку.

1.6. Добавим еще пару признаков к нашему файлу: удельный вес жука (отношение веса жука к его длине) — VES.R и порядковый номер жука — N:

```
data$VES.R <- data$VES/data$ROST
data$N <- c(1:10)
```

Проверьте, что новые столбцы появились, при помощи команды

```
str(data)
```

1.7. Новые признаки были добавлены только к копии вашего файла данных, который находится в памяти программы! Эта копия исчезнет, как только вы выйдете из R. Чтобы сохранить измененный файл данных под именем `zhuki_new.txt` в вашей директории на жестком диске компьютера, нужно написать:

```
write.table(data, "zhuki_new.txt", quote=FALSE)
```

Учтите, что если файл с этим названием уже существует в вашей директории, R запишет новый файл вместо старого без каких-либо предостережений и вопросов. Будьте внимательны!

2. Характеризуем выборку

2.1.1. Посмотрим основные характеристики каждого признака (это не имеет смысла для категориальных данных): минимальные («Min.» = minimum) и максимальные («Max.» = maximum) значения; среднее арифметическое («Mean»), медиана («Median»), нижняя («1st Qu.» = first quartile) и верхняя («3rd Qu.» = third quartile) квартили. Заодно там будет указано число пустых ячеек («NA», то есть «not available»). Анализируя минимальные и максимальные значения, можно выявить и явные опечатки (скажем, 100 вместо 10). Итак:

```
summary(data)
```

Кстати, строки с пропущенными значениями можно удалить из таблицы данных так: `data.o <- na.omit(data)`.

Естественно, такой подход (вычисление среднего арифметического и т.п.) не позволяет получить осмысленную информацию о категориальных данных (таких как пол и цвет жуков). Распределение значений категориальных признаков можно изучить при помощи команд 2.3.1–2.3.3 (см. ниже).

2.1.2. Конечно, команду **summary**() — как и многие прочие — можно применять как ко всему файлу данных, так и к любому отдельному признаку:

```
summary(data$VES)
mean(data$VES)
mean(data)
```

К сожалению, если есть пропущенные значения, эти команды не работают так, как нам обычно нужно. Посчитаем среднее для каждого из признаков, избавившись от пропущенных значений:
`mean(data, na.rm=TRUE)`.

2.2.1. Иногда бывает нужно вычислить сумму всех значений признака:

```
sum(data$VES)
```

2.2.2. ... или сумму всех значений одной строки (попробуем на примере второй):

```
sum(data[2, ])
```

2.2.3. ... или сумму значений всех признаков для каждой строки:

```
apply(data, 1, sum)
```

Чтобы посчитать сумму значений для каждого столбца, нужно вместо «1» написать «2».

2.3.1. Для категориальных признаков имеет смысл посмотреть, сколько раз встречается в выборке каждое значение признака (заодно узна́ем, какие значения признак принимает):

```
table(data$POL)
```

По умолчанию эта функция игнорирует пропущенные значения. Для добавления в таблицу данных о частоте пропущенных значений используйте параметр `useNA="ifany"`.

2.3.2. А теперь выразим частоту встречаемости значений признака не в числе объектов, а в процентах, приняв за 100% общее число объектов:

```
100*prop.table(table(data$POL))
```

Как видно на этом примере, команды в R можно комбинировать «по принципу матрешки», то есть результат, полученный при помощи одной команды, можно модифицировать при помощи другой команды.

2.3.3. И еще округлим значения процентов до целых чисел:

```
round(100*prop.table(table(data$POL)), 0)
```

2.4.1 Можно вычислять характеристики любого признака отдельно для самцов и для самок. Попробуем на примере среднего арифметического для веса:

```
tapply(data$VES, data$POL, mean)
```

2.4.2. Посмотрим, сколько жуков разного цвета среди самцов и самок:

```
table(data$CVET, data$POL)
```

Строки — разные цвета, столбцы — самцы и самки.

2.4.3. Добавим информацию об общем числе жуков каждого пола и каждого цвета:

```
addmargins(table(data$CVET, data$POL))
```

2.4.4. А теперь вычислим частоту встречаемости самцов и самок каждого цвета (в процентах от общего числа жуков):

```
100*prop.table(table(data$CVET, data$POL))
```

2.4.5. ... и в процентах от числа жуков соответствующего цвета

```
100*prop.table(table(data$CVET, data$POL), 1)
```

2.4.6. ... в процентах от числа жуков соответствующего пола

```
100*prop.table(table(data$CVET, data$POL), 2)
```

2.4.7. И, наконец, вычислим средние значения веса жуков отдельно для всех комбинаций цвета и пола (для красных самцов, красных самок, зеленых самцов, зеленых самок...):

```
tapply(data$VES, list(data$POL, data$CVET), mean)
```

3. Рисуем диаграммы

3.1.1. Проверим, как распределены данные, нет ли выбросов. Для этого построим гистограммы для каждого признака (вот как это выглядит на примере веса):

```
hist(data$VES)
```

Детализацию гистограммы можно изменять, варьируя число интервалов (параметр `breaks`). Например, так можно построить гистограмму, где значения признака разделены на 20 интервалов:

```
hist(data$VES, breaks=20)
```

Можно задать точную «ширину» интервалов значений признака на гистограмме (зададим «ширину» в 20 единиц, а значения признака пусть изменяются от 0 до 100):

```
hist(data$VES, breaks=c(seq(0,100,20)))
```

Вообще R достаточно «умен», чтобы самостоятельно определить оптимальную детализацию гистограммы.

3.1.2. Для категориальных признаков больше подходят столбчатые диаграммы:

```
barplot(table(data$CVET))
```

Можно графически исследовать *сопряженность между двумя категориальными признаками*:

```
counts <- table(data$CVET, data$POL) # подготавливаем данные
barplot(counts, beside=TRUE, legend=row.names(counts))
```

R «не видит» все, что стоит после знака `#`, это удобный способ добавлять к командам пояснения.

3.1.3. Более точно проверить близость распределения значений к нормальному можно при помощи двух команд:

```
qqnorm(data$VES); qqline(data$VES)
```

Обе команды «делают» один график, поэтому мы написали их в одну строчку через точку с запятой. Чем больше распределение точек на этом графике отклоняется от прямой линии, тем дальше распределение данных от нормального.

3.1.4. Не забывайте подписывать оси (параметры `xlab` и `ylab`)! А вот название диаграммы (`main`) лучше приводить не на ней самой, а в подписи к рисунку в отчете.

```
hist(data$VES, main="", xlab="Вес жука (г)",
        ylab="Количество жуков")
```

3.2.1. Построим диаграмму рассеяния, на которой объекты будут обозначены кружочками. Рост будет по оси абсцисс (горизонтальная ось!), вес по оси ординат (вертикальная ось!):

```
plot(data$ROST, data$VES)
```

3.2.2. Можно изменять размер кружочков (параметр `cex`). Сравните:

```
plot(data$ROST, data$VES, cex=0.5)
```

и

```
plot(data$ROST, data$VES, cex=2)
```

Кстати, для того, чтобы открыть новое графическое окно (новый график будет нарисован рядом со старым, а не вместо него), можно использовать команду **`x11`**`()`.

3.2.3. Можно изменить тип значков на диаграмме. Например, обозначим по-разному самцов и самок:

```
plot(data$ROST, data$VES, type="n")
```

```
points(data$ROST, data$VES, pch=data$POL)
```

Обе команды создают одну и ту же диаграмму. Первая создает «заготовку», вторая добавляет туда значки.

Таблицу с номерами значков можно вызвать так: `example(points)`. После введения этой команды нужно нажать несколько раз Enter, пока демонстрация не окончится.

3.2.4. Можно вместо значков обозначить объекты на диаграмме кодом пола (0/1):

```
plot(data$ROST, data$VES, type="n")
text(data$ROST, data$VES, labels=data$POL)
```

3.2.5. Еще можно сделать так, чтобы разные цифры-обозначения имели и разные цвета:

```
plot(data$ROST, data$VES, type="n")
text(data$ROST, data$VES, labels=data$POL,
        col=data$POL+1)
```

При назначении цветов (параметр `col`) мы добавили единицу к обозначению пола (получилось 1 — самки, 2 — самцы), иначе самки были бы обозначены белым «цветом № 0», то есть соответствующие значки не были бы видны на диаграмме.

3.2.6. Наконец, если на нашей диаграмме есть разные типы значков (команда 3.2.3), нужно добавить условные обозначения (легенду):

```
legend(10, 9, c("самки","самцы"), pch=c(0,1),
        title="Пол")
```

Числами указано положение (координаты на графике) верхнего левого угла легенды: первая цифра (10) — абсцисса, вторая цифра (9) — ордината. Можно поступить еще проще — в явном виде указать положение легенды на диаграмме. Например, так можно разместить легенду в верхнем левом углу:

```
legend("topleft", c("самки","самцы"), pch=c(0,1)).
```

3.3. Сохраняем график при помощи меню:

```
Файл -> Сохранить как… -> PNG
```

При этом графическое окно должно быть активно — наведите на него курсор и нажмите левую кнопку мыши. Сохранить график можно и при помощи команд, например:

```
dev.copy(png, "grafik.png"); dev.off()
```

3.4.1. Рисуем линейную диаграмму:

```
plot(data$N, data$ROST, type="o")
```

3.4.2. А теперь нарисуем две линии на одном графике:

```
plot(data$N, data$ROST, type="o", ylim=c(5, 15))
lines(data$N, data$VES, lty=3)
```

Аргумент `ylim` задает длину оси ординат (от 5 до 15). Аргумент `lty` задает тип линии («3» — это пунктир).

3.5.1. Рисуем диаграмму размаха (показывает выбросы, минимум и максимум, квартильный размах, медиану):

```
boxplot(data$ROST)
```

3.5.2. ... а теперь — для самцов и самок по отдельности:

```
boxplot(data$ROST ~ data$POL, names=c("самки","самцы"))
```

4. Статистические тесты

4.1.1. Достоверность различий параметрических данных (тест Стьюдента) для двух зависимых переменных (по умолчанию в R используется вариант теста, не требующий одинакового разброса данных относительно среднего):

```
t.test(data$VES, data$ROST, paired=TRUE)
```

Если p-value < 0,05, то различие между выборками статистически значимо.

4.1.2. ... и для независимых переменных:

```
t.test(data$VES, data$ROST)
```

Нам не потребовалось писать в конце `paired=FALSE`, поскольку это параметр по умолчанию.

4.1.3. ... если нужно сравнить значения одного признака для двух групп:

```
t.test(data$VES ~ data$POL)
```

4.2. Достоверность различий непараметрических данных (тест Вилкоксона):

```
wilcox.test(data$VES, data$ROST)
```

4.3.1. Достоверность различий между тремя и более выборками параметрических данных (вариант однофакторного дисперсионного анализа, не требующий одинакового разброса данных относительно среднего) — на примере различия веса жуков трех цветов:

```
oneway.test(data$VES ~ data$CVET)
```

4.3.2. Посмотрим, какие именно пары выборок достоверно различаются (попарный тест Стьюдента с поправкой на множественные сравнения):

```
pairwise.t.test(data$VES, data$CVET)
```

4.4. А теперь проверим достоверность различий между несколькими выборками непараметрических данных (тест Краскела-Уоллиса, а затем попарный тест Вилкоксона с поправкой на множественные сравнения):

```
kruskal.test(data$VES ~ data$CVET)
pairwise.wilcox.test(data$VES, data$CVET)
```

4.5.1. Достоверность сопряженности категориальных данных (тест хи-квадрат):

```
chisq.test(data$CVET, data$POL)
```

4.5.2. Достоверность различия пропорций (тест пропорций):

```
prop.test(c(sum(data$POL)), c(length(data$POL)), 0.5)
```

Это мы проверили — правда ли, что доля самцов достоверно отличается от 50%.

4.6.1. Достоверность линейной связи между параметрическими данными (корреляционный тест Пирсона):

```
cor.test(data$VES, data$ROST, method="pearson")
```

Можно было не писать в конце `method="pearson"`, поскольку это метод по умолчанию.

4.6.2. ... и между непараметрическими (корреляционный тест Спирмена):

```
cor.test(data$VES, data$ROST, method="spearman")
```

5. Заканчиваем...

5.1. Сохраняем историю команд:

```
savehistory("zhuki.r")
```

Все введенные вами команды сохраняются в файл с расширением *.r, который можно открыть в любом текстовом редакторе и исправлять, дополнять или копировать в строку ввода программы в следующий раз. Этот файл можно даже выполнить целиком («запустить»), для этого используется команда `source("zhuki.r")`. Запускать нужно, однако, с большой осторожностью, потому что в файле могут быть команды, удаляющие или переписывающие какие-то нужные Вам файлы.

5.2. Выходим из программы через меню. На вопрос «Сохранить рабочее пространство?» отвечаем *отрицательно*.

Глава 4

Многомерный статистический анализ данных

В этой главе рассказывается о том, как использовать более сложные методы обработки данных — многомерную статистику.

4.1. Зачем нужен многомерный анализ данных?

Окружающий нас мир многомерен в том смысле, что каждый объект характеризуется множеством в разной степени взаимосвязанных параметров. Исследователь снижает размерность мира, выбирая тему своего исследования, то есть, очерчивая круг параметров, которые будут его интересовать. Однако и в этом случае чаще всего одновременно анализируют несколько, а то и несколько десятков и даже сотен признаков. Например, мы задались целью изучить зависимость артериального давления от возраста человека. Регистрировать только эти два параметра для каждого испытуемого было бы некорректно. Ясно, что на артериальное давление влияют другие (тоже взаимосвязанные) факторы (и некоторые даже сильнее, чем возраст), например, масса тела, наличие вредных привычек, физическая активность, наследственность и т.п., которые тоже придется учитывать при анализе зависимости артериального давления от возраста.

Основная проблема анализа таких многомерных матриц данных заключается в том, что человеческий мозг не способен одновременно оперировать более чем тремя измерениями пространства (поскольку пространственное воображение хорошо развито далеко не у всех людей, оптимально сократить количество измерений до двух). Для сведéния многомерных данных к двум измерениям с минимальными потерями информации была разработана специальная группа методов статистического анализа данных — многомерный анализ данных. Эти методы чрезвычайно разнообразны и основаны на довольно сложных математических расчетах. В настоящем пособии мы рассмотрим несколько самых основных и наиболее широко употребляемых методов многомерного анализа данных, не углубляясь, разумеется, в математические дебри.

4.2. Несколько практических рекомендаций

Исходные многомерные данные могут быть представлены как в виде переменных, то есть отдельных признаков объектов (более привычный нам вид), так и в виде матрицы расстояний.

Матрица расстояний представляет собой таблицу, где в первой строке и первом столбце перечислены объекты, а на пересечении строк и столбцов указаны «расстояния» между соответствующей парой объектов. Под расстояниями здесь понимается как привычное значение этого слова (примером такой матрицы могут служить таблицы расстояний между городами в туристических атласах), так и вообще любая мера различия между объектами. Например, при тестировании азбуки Морзе[1] испытуемым давали прослушать пары кодов и просили указать, являются ли они идентичными. Мерой различия («расстоянием») между парой кодов служило число испытуемых, считающих эту пару не идентичной.

[1] Система кодировки букв и цифр при помощи комбинаций из коротких и длинных сигналов. Применяется в основном в радиосвязи.

Если же данные представлены переменными, то необходимо организовать файл данных так, чтобы строки представляли собой объекты, которые вы собираетесь классифицировать, а столбцы — переменные (признаки), описывающие эти объекты, на основе которых проводится классификация. Рекомендуется назвать строки короткими условными обозначениями объектов (латинскими буквами и цифрами), поскольку именно названия строк будут обозначать объекты на полученных диаграммах классификации.

Например, если вы классифицируете мебель по ее типам, столы (tables) можно назвать `t1, t2, t3`..., кровати (beds) — `b1, b2, b3`... и т.д.

Существует определенная проблема выбора объектов, которые вы собираетесь анализировать, и признаков, на основании которых этот анализ будет построен. На первый взгляд эта проблема кажется несколько надуманной («столько труда потратили на сбор данных, теперь надо все их и проанализировать»). Однако «лишние» признаки (то есть не вносящие существенного вклада в решение поставленной задачи) способны «маскировать» реальную структуру данных. Таким «лишним» признаком, например, может быть размер обуви в приведенном выше примере про исследование артериального давления. Включение слишком выделяющихся из общего множества объектов может также затруднить интерпретацию данных (здесь мы не имеем в виду выбросы, от которых нужно немедленно избавляться). К примеру, при классификации трех близких видов растений по морфологическим признакам включение четвертого сильно отличающегося вида непременно затруднит разграничение этих трех видов.

И, наконец, помните, что практически все описываемые методы многомерного анализа данных работают с количественными и порядковыми признаками, но не категориальными! Если же категориальные признаки нужно обязательно использовать, то поможет перевод их в бинарные (0/1)[2].

[2]Функция **`daisy()`** из пакета cluster позволяет работать одновременно с разными типами данных (даже категориальными).

Например, если в колонке есть «зеленый» и «красный» цвета, то нужно сделать две колонки, и в первой поставить «1» тем образцам, которые были помечены словом «зеленый», а во второй колонке поставить «1» тем образцам, которые были помечены как «красный»; все остальные ячейки нужно заполнить нулями. Так из одного категориального признака можно получить два бинарных.

Все примеры будут основаны на данных о размерах листьев берёзы[3]. Каждая пронумерованная строка соответствует одному листу.

Всего признаков (столбцов) шесть: ширина листа в мм (SH.L), длина листа в мм (DL.L), возраст ветки, лет (VOZR.V), положение листа на ветке (POL.L: 1 — в нижней части, 2 — в середине, 3 — в верхней части ветки), длина черешка листа, мм (DL.CH) и регион произрастания (REG: 1 — Средняя Россия, 2 — Кольский полуостров, 3 — Сибирь). Вот они:

```
    SH.L DL.L VOZR.V POL.L DL.CH REG
b01   20   29      1     1    15   1
b02   26   31      1     2    12   1
b03   29   34      2     3    17   1
b04   13   20      2     1     3   2
b05   19   25      4     2     4   2
b06   15   24      7     3     7   2
b07   17   23      7     1    30   2
b08   41   60      8     2    25   3
b09   42   71     10     3    29   3
b10   47   85     10     3    16   3
```

Краткие рекомендации по применению методов многомерного анализа данных в R по-прежнему даны мелким шрифтом. Пусть наши данные будут представлены объектом `mdata`.

[3]Конечно, таких берёз не бывает, а 10 листьев вовсе недостаточно для того, чтобы обнаружить какие-либо закономерности в многомерных данных. Мы просто придумали эти данные, для примера.

4.3. Основные методы многомерного анализа данных

4.3.1. Какой метод выбрать?

Можно представить многомерные данные графически, непосредственно отражая значения переменных на диаграмме. Несомненным достоинством этого метода является отсутствие обработки исходных данных, вследствие чего мы можем «считывать» с диаграммы значения отдельных переменных. Однако такой метод хорош при небольшом числе признаков (не более пяти), в противном случае, способы отображения разных переменных начинают смешиваться в восприятии. Кроме того, на этих диаграммах можно выявить только очень четко выраженные группы или закономерности.

В случае большего числа признаков и/или сложной структуры данных (обычная ситуация!) нам необходимо будет воспользоваться собственно методами многомерного анализа данных.

Дискриминантный анализ (см. Кабаков, 2014; Шипунов и др., 2012) позволяет нам проверить сформированную теорию (о принадлежности объектов к определенным группам) и сформулировать правило распределения объектов по группам, которое можно применять для классификации объектов с неизвестной групповой принадлежностью. Последнее свойство дискриминантного анализа (так называемая «классификация с обучением»[4]) имеет большое практическое значение. Часто бывает так, что определить принадлежность объекта к группе не представляется возможным. Например, точный диагноз больному в некоторых случаях можно поставить только после вскрытия. Однако, пронаблюдав (а после смерти вскрыв) несколько сотен

[4]Конечно, дискриминантный анализ — это всего лишь один из множества существующих методов классификации с обучением (и не самый лучший). Однако этот метод широко распространен (может быть, потому, что он был придуман одним из первых), поэтому рассказывать о классификации с обучением мы будем на примере именно дискриминантного анализа.

больных, можно разработать надежную систему диагностики заболеваний по внешним признакам, которая позволит ставить правильный диагноз живым людям.

Прочие методы (**кластерный анализ**, **многомерное шкалирование** и **анализ главных компонент**) помогают выявить изначально неизвестную структуру в данных. Надо учитывать, что эти методы делают упор на визуальное представление результатов, а не на проверку их статистической значимости, оценка структуры данных «на глаз» весьма субъективна. Кроме того, разнообразие данных не всегда может быть сведено к двумерному пространству без существенных потерь информации. Поэтому полученные классификации желательно было бы как-нибудь проверять. Можно попробовать классифицировать данные несколькими методами и сравнить полученные классификации. Если они совпадают в общих чертах, то ваши результаты соответствуют реальному положению дел. Можно попробовать описать полученные группы (если они выявляются) — то есть при помощи описательных статистик или двухмерных графиков, или **деревьев классификации** найти отдельные переменные, значения которых позволяют разграничить эти группы (см. Шипунов и др., 2012). Можно, наконец, проверить достоверность классификации при помощи **многомерного дисперсионного анализа** данных (см. Кабаков, 2014).

Нужно иметь в виду, что каждый метод многомерного анализа данных имеет множество вариаций, и очень часто конечный результат зависит от многих параметров. Поэтому во всех отчетах и публикациях с применением этих методов необходимо ясно указывать как минимум:

- название компьютерной программы, в которой выполнялся анализ (разные программы могут иметь разные алгоритмы реализации одних и тех же методов);

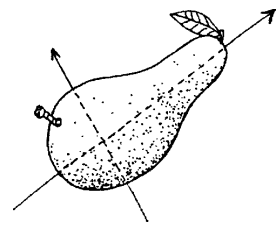
- заданные параметры (например, метод вычисления расстояния между объектами и метод кластеризации для кластерного анализа — см. ниже);

- способ определения числа групп (для тех методов, в которых оно неизвестно).

4.3.2. Анализ главных компонент

Анализ главных компонент (Principal component analysis, PCA) — это один из наиболее широко употребимых и старых методов многомерного анализа данных[5]. В основе этого метода лежит сведéние всего множества исходных признаков к нескольким новым некоррелированным переменным (собственно, главным компонентам), представляющим собой линейную комбинацию исходных переменных.

Это значит, что наши объекты можно представить как точки в n-мерном пространстве, где n — это число анализируемых признаков. Через полученное облако точек проводится прямая так, чтобы учесть наибольшую долю изменчивости признаков, то есть «пронизывая» это облако вдоль в наиболее вытянутой его части (представьте себе трехмерное облако грушевидной формы, см. рисунок) — это первая главная компонента.



[5] Существует несколько других, близких по смыслу, методов многомерного анализа данных — это анализ соответствий (Correspondence analysis) и факторный анализ (Factor analysis). Эти методы различаются в основном теоретически, а результаты дают довольно сходные, поэтому здесь мы подробно остановимся только на анализе главных компонент. В R есть специальный пакет `vegan`, в котором содержится много методов, близких к анализу главных компонент. Этот пакет создан в основном для обработки биологических данных, в частности, данных экологии и систематики.

Затем через это облако проводится вторая, перпендикулярная первой, прямая, так чтобы учесть наибольшую оставшуюся долю изменчивости признаков — как вы уже догадались, вторая главная компонента. Эти две компоненты образуют плоскость, на которую и проецируются все точки[6].

Перед тем, как начать обработку данных, необходимо их сначала стандартизовать: например, из значений каждого признака нужно вычесть его среднее значение и разделить на стандартное отклонение — команда **`scale`**`()`. Эта операция нужна для того, чтобы размерность данных (сантиметры или километры?) не влияла на результаты анализа.

Сам анализ главных компонент реализуется так:

```
mdata.pca <- princomp(scale(mdata[,1:5]))
```

Теперь можно узнать несколько важных вещей, касающихся полученной классификации:

```
loadings(mdata.pca)
```

Во-первых, можно посмотреть, какую долю изменчивости признаков описывает каждая из выделенных главных компонент (строка Proportion Var нижней таблички) и все главные компоненты вместе (самое правое значение строки Cumulative Var). Ясно, что если две первых компоненты вместе описывают очень мало изменчивости (скажем, меньше 50%), то на такую классификацию не сто́ит и смотреть, слишком уж она малоинформативна.

Во-вторых, можно попробовать охарактеризовать полученные компоненты, понять, за что каждая из них «отвечает». В верхней из двух табличек, порождаемых командой **`loadings`**`(mdata.pca)`, указаны коэффициенты корреляции между всеми исходными признаками и каждой главной компонентой. Чем ближе коэффициент по модулю к единице, тем существеннее вклад данного признака в определенную

[6]На самом деле, обычно главных компонент выделяется больше двух (на одну меньше, чем было исходных признаков), но основную информацию об изменчивости признаков, как правило, несут первые две компоненты.

главную компоненту. Эти же сведения можно представить графически:

```
biplot(mdata.pca)
```

Запустите эту команду самостоятельно. Красные стрелки обозначают признаки, оси — первые две главные компоненты. Чем больше проекция стрелки на ось, тем больший вклад этот признак вносит в данную компоненту. Из этой же диаграммы можно делать выводы и о сопряженности признаков: чем ближе стрелки друг к другу, тем сильнее сопряжены признаки.

Классификация объектов на плоскости двух первых главных компонент приведена на рисунке 8.

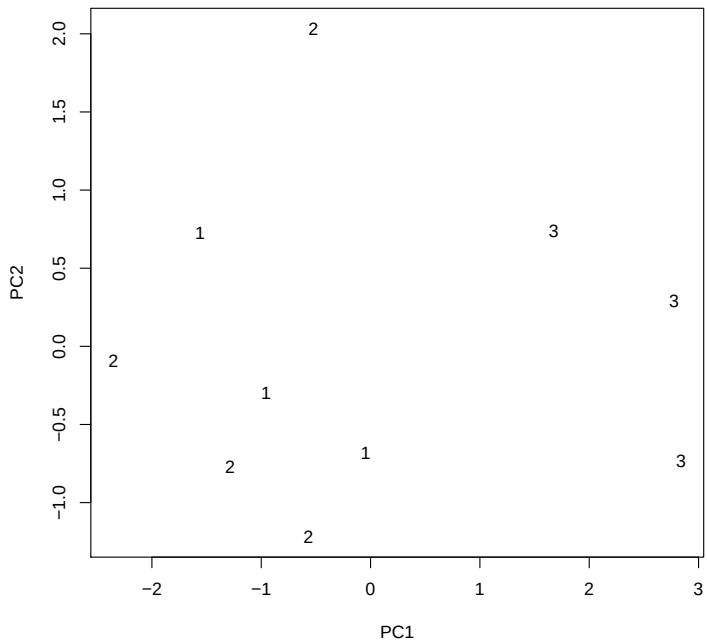


Рис. 8. Классификация листьев березы на плоскости двух первых главных компонент (цифрами обозначены регионы произрастания).

Чтобы увидеть эту классификацию, нужно ввести подряд такие команды:

```
mdata.p <- mdata.pca$scores[,1:2]
plot(mdata.p, type= "n", xlab= "PC1", ylab= "PC2")
text(mdata.p, labels=mdata[,6])
```

Каждый объект обозначен номером группы из шестой колонки. Можно обозначать объекты разных типов разными цветами и/или символами (см. команды 3.2.3—3.2.6 на стр. 64).

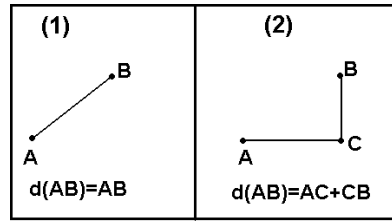
4.3.3. Иерархический кластерный анализ

Кластерный анализ основан на выделении групп сходных между собой объектов (так называемых **кластеров**). На сегодняшний день разработано множество методов кластерного анализа, из которых в биологии обычно используются методы, основанные на последовательном объединении (или, напротив, на последовательном разделении) объектов в иерархические[7] кластеры. Примером такой классификации может служить система живой природы: сходные виды объединяются в рода, сходные рода — в семейства... Нужно учитывать, что иерархический кластерный анализ подразумевает наличие в данных структуры. Он как бы «навязывает» данным структуру, хотя ее там может не быть.

Как измерять сходство (расстояние) между объектами? Методов вычисления расстояний существует очень много (не забывайте, что дело происходит в многомерном пространстве). Для количественных переменных шире всего используются:

1. евклидово расстояние (euclidian), и
2. манхеттеновское расстояние или расстояние городских кварталов (manhattan)

[7] Это значит, что несколько мелких (состоящих из небольшого числа объектов) кластеров объединяются в кластер среднего размера, несколько таких средних кластеров объединяются в кластер покрупнее и т.д.



Для бинарных признаков (данных типа «да — нет») при вычисления расстояния используют свои методы, например, просто подсчитывают число параметров, которые совпадают у объектов: это, например, коэффициент совместной встречаемости (binary).

Часто бывает полезно учитывать только положительные совпадения. Например, если какой-нибудь вид растения обнаружен на обоих островах, то это увеличивает их флористическое сходство, а если какой-то вид на обоих островах отсутствует, то это еще ни о чем не говорит: может быть, на одном острове его просто не заметили.

Только положительные совпадения учитывает, например, коэффициент сходства Жаккара, который можно рассчитать при помощи функции **`vegdist`**`(..., method="jaccard")` из пакета vegan.

Вычисляем матрицу манхеттеновских расстояний между всеми парами объектов:

```
mdata.dist <- dist(scale(mdata[,1:5]), method="manhattan")
```

Итак, пара наиболее близких объектов образовала первый кластер. **На каком основании можно присоединить к кластеру еще один объект?** Известно, по крайней мере, 12 методов присоединения к кластеру нового объекта, из которых наиболее распространены четыре:

Метод одиночной связи (single) Новый объект должен иметь наибольшее сходство (по сравнению с прочими «кандидатами на присоединение») с одним из членов кластера. Недостатком такого метода являются большие продолговатые кластеры («гребенка»). Зато это единственный ме-

тод, который нечувствителен к изменению порядка объектов в исходном файле данных и к наличию в данных выбросов.

Метод полной связи (complete) Сходство между новым объектом и всеми членами кластера должно превышать некоторое пороговое значение (вычисляемое программой). Этот метод дает компактные кластеры и хорошо работает с группами разного размера.

Метод средней связи (average) Этот метод является своеобразным компромиссом между двумя предыдущими методами, потому что расстояние между новым объектом и кластером определяется как среднее арифметическое расстояний между этим объектом и всеми членами кластера. Кластеры обычно получаются довольно продолговатыми. Хорошо работает с группами разного размера, эффективно выделяет структуру, «скрытую» случайной изменчивостью признаков.

Метод Уорда (ward.D2) Объект для присоединения выбирают так, чтобы приращение суммы квадратов отклонений от средних значений признаков внутри кластера было минимальным. Позволяет получить компактные хорошо выраженные кластеры. Хорошо работает с группами сходных размеров, эффективно выделяет структуру, «скрытую» случайной изменчивостью признаков.

«Наращиваем» кластеры методом Уорда:

```
mdata.h <- hclust(mdata.dist, method="ward.D2")
```

Выводим дендрограмму (графическое отображение полученной классификации) на экран:

```
plot(mdata.h)
```

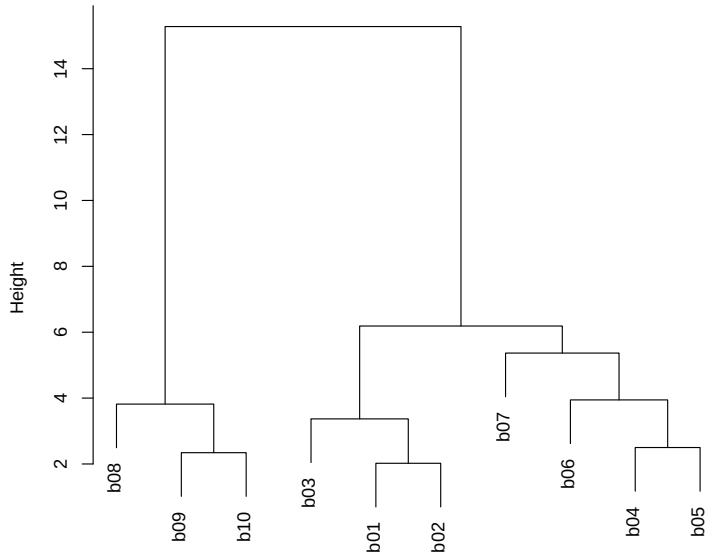


Рис. 9. Дендрограмма классификации листьев берёз.

Какой же метод выбрать? Единственного ответа на этот вопрос не существует. Известно, что разные методы порождают разные классификации для одних и тех же данных. Более того, результаты кластерного анализа изменяются, если некоторые объекты исключаются из рассмотрения или, наоборот, добавляются, или даже просто меняются местами. Единственный выход из сложившейся ситуации — это попробовать несколько методов кластеризации и посмотреть, насколько стабильна полученная классификация и как она соотносится с вашими представлениями о структуре данных. По совокупности признаков наилучшей комбинацией нам представляется *метод Уорда с расстояниями городских кварталов*.

Как определить число кластеров? Как правило, нас интересуют не все уровни классификации данных, а разделение объектов на несколько крупных групп, которое можно легко интерпретировать. Так на каком уровне нужно «обрезать» кластерное дерево (так называемую **дендрограмму**), чтобы получить оп-

тимальное число групп? Как правило, решение о числе кластеров принимается исследователем на основании личного опыта и визуального анализа дендрограммы.

Например, ясно, что на дендрограмме (рис. 9) можно выделить две главные группы: одна состоит из «сибирских» листьев 8–10, а другая — из всех остальных. В этой второй группе можно в свою очередь выделить две подгруппы: «листьев средней полосы» (1–3) и «листьев Кольского полуострова» (4–7). Существует и несколько формальных способов определения числа кластеров, но они малоэффективны.

4.3.4. Многомерное шкалирование

Многомерное шкалирование[8] «работает» с **матрицами расстояний** (которые можно создать при помощи команды **dist**(), см. стр. 80), а не со значениями признаков. Результаты многомерного шкалирования, точно так же как и результаты кластерного анализа, зависят от выбранного метода вычисления расстояний между объектами.

Подключаем дополнительный пакет команд:

```
library(MASS)
```

Он установлен по умолчанию, поэтому скачивать и устанавливать его не требуется.

Проводим многомерное шкалирование:

```
mdata.i <- isoMDS(mdata.dist)
```

Вместо **isoMDS**() можно использовать функцию **cmdscale**().

Визуальное представление классификации (рис. 10):

[8] Так называемый «метод анализа главных координат» (Principal coordinate analysis, PCO) можно считать синонимом многомерного шкалирования.

```
eqscplot(mdata.i$points, type="n")
text(mdata.i$points, row.names(mdata))
```

Здесь вместо точек мы напечатали названия наших берёз (row names).

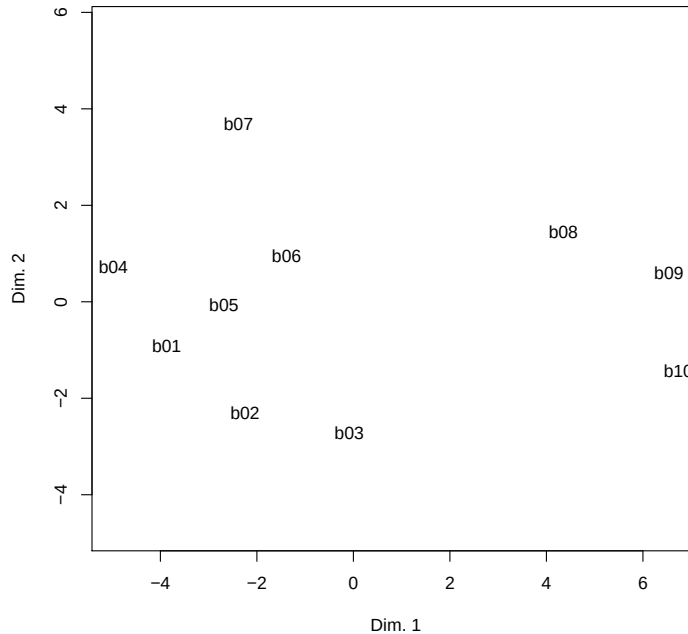


Рис. 10. Классификация листьев берёз методом многомерного шкалирования.

Литература для дополнительного чтения

Бослаф С. Статистика для всех. М.: ДМК Пресс, 2015.

Бунимович Е.А., Булычев В.А. Основы статистики и вероятность. М.: Дрофа, 2008.

Кабаков Р.И. R в действии. Анализ и визуализация данных в программе R. М.: ДМК Пресс, 2014.

Кимбл Г. Как правильно пользоваться статистикой. М.: Финансы и статистика, 1982.

Шипунов А.Б. и др. Наглядная статистика. Используем R! М.: ДМК Пресс, 2012.[9]

[9] С 2014 года находится в общественном достоянии: http://ashipunov.info/shipunov/school/books/rbook.pdf.

Оглавление

Приложение. Выбираем правильный метод статистического анализа

<table>
<tr><th colspan="2">Какие данные?
Что ищем?</th><th>Параметрические</th><th colspan="2">Непараметрические</th></tr>
<tr><th colspan="2"></th><th></th><th>количественные или порядковые</th><th>категориальные</th></tr>
<tr><td rowspan="2">Различия</td><td>две выборки</td><td>тест Стьюдента (стр. 43)</td><td>тест Вилкоксона (стр. 44)</td><td rowspan="2">тест пропорций (стр. 46)</td></tr>
<tr><td>больше двух выборок</td><td>дисперсионный анализ (стр. 48) и попарный тест Стьюдента с поправкой на множественные сравнения (стр. 49)</td><td>тест Краскела-Уоллиса (стр. 49) и попарный тест Вилкоксона с поправкой на множественные сравнения (стр. 49)</td></tr>
<tr><td colspan="2">Связи между двумя признаками</td><td>тест Пирсона: корреляция (стр. 53)</td><td>тест Спирмена: корреляция (стр. 53)</td><td>тест хи-квадрат: сопряженность (стр. 46)</td></tr>
<tr><td colspan="2">Структуру: много признаков</td><td colspan="3">многомерный анализ (стр. 70)</td></tr>
</table>

Московская гимназия на Юго-Западе (№1543)
119526, Москва, ул. 26 Бакинских комиссаров, д. 3., к. 5.
Тел. (495) 434–2658.
Fax. (495) 434–2644.
E-mail: bioclass@yandex.ru.
В 8 и 9 биологические классы нашей гимназии можно поступить, см. подробнее на `http://bioclass.ru`.

Сведения об авторах:

Волкова Полина Андреевна
Учитель биологии в профильных классах гимназии, кандидат биологических наук. Автор методических пособий для обучения школьников. Победитель конкурса лучших учителей Российской Федерации. Московская гимназия на Юго-Западе № 1543.

Шипунов Алексей Борисович
Преподаватель биологии в ВУЗе, кандидат биологических наук. Автор методических пособий для обучения школьников и студентов. Университет Северной Дакоты (Майнот, США).