



Институт технологий управления

Программные средства анализа данных

01.03.05 Статистика

Профиль «Анализ данных в бизнесе и экономике»

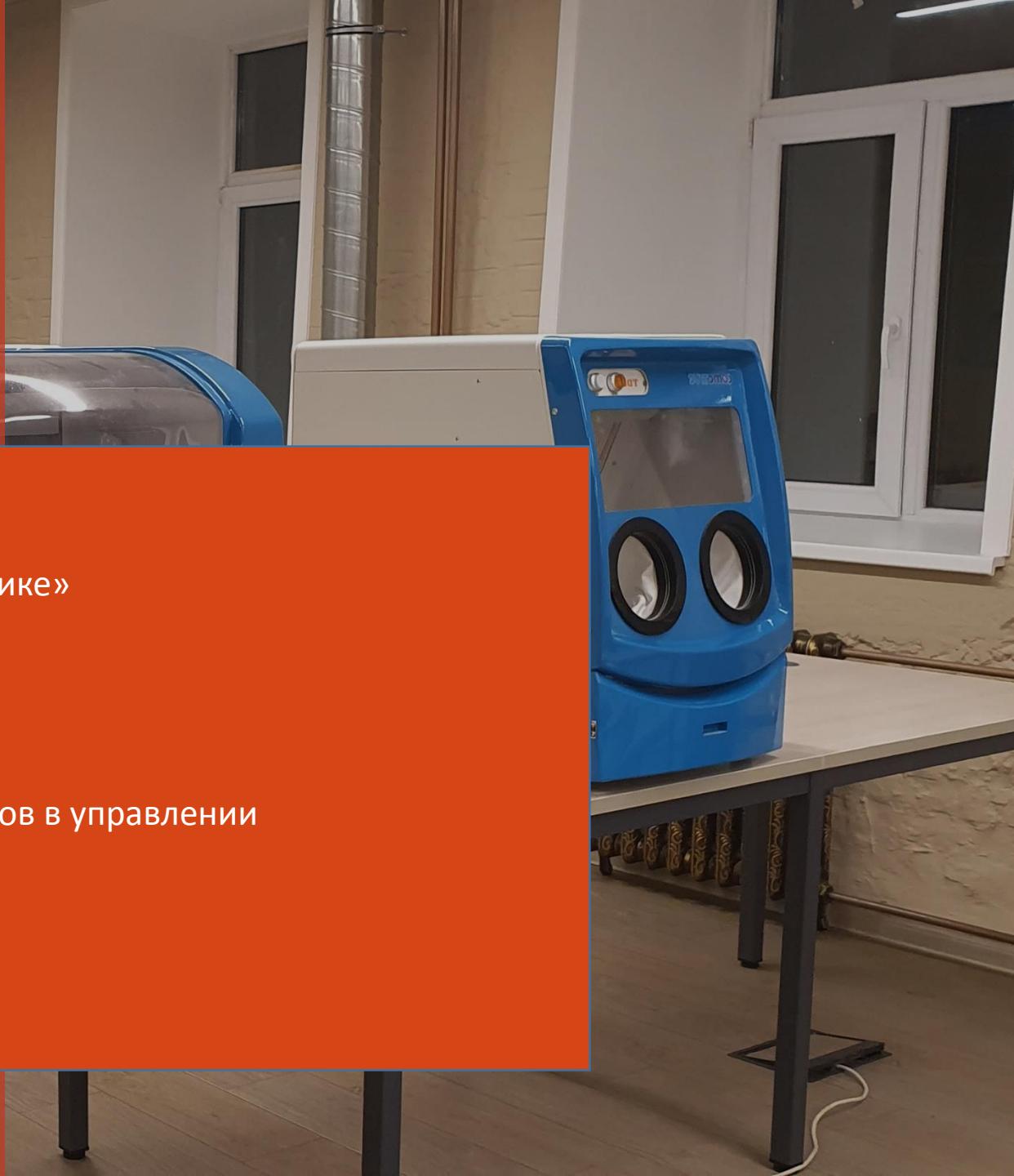
Квалификация «бакалавр»

Бурцева Татьяна Александровна

Профессор кафедры статистики и математических методов в управлении

Burceva_t@mirea.ru

Москва, 2022



Тема 1. R как программное средство статистического анализа

План лекции

1. Обзор статистических программных продуктов
2. Понятие объекта в R.
3. Виды представления данных в R.
4. Среда программирования R.
5. Циклы и функции.
6. Ввод данных в R.

1. Обзор статистических программных продуктов

***Большие данные (Big Data)** – это набор специальных методов и инструментов, которые используются для хранения и обработки огромных объемов данных в рамках конкретных задач. Такие данные могут быть как структурированными, так и неструктурированными (неорганизованными, неоднородными).

Блог <http://r-analytics.blogspot.ru/>

Учебник : <http://www.cookbook-r.com/>

Учебная среда, курс Анализ данных в R <https://stepik.org/lesson>

Обработка больших объёмов информации невозможна без специализированных цифровых продуктов для оптимизации статистического анализа, так как построение таблиц с тысячами и десятками тысяч наблюдений не целесообразно и физически не возможно.

Для решения этой задачи используются статистические пакеты программ (далее СПП) и (или) языки программирования, которые позволяют:

1. Работать с большим объёмом данных
2. Ускорять процесс описания и анализа
3. Автоматизировать статистические операции
4. Визуализировать результаты

Основные задачи обработки и анализа данных

- доступ к данным (получение данных из разных источников);
- редактирование данных (замена или удаление пропущенных значений, преобразование признаков в более удобный вид);
- аннотирование данных (чтобы помнить, что представляет собой каждый их фрагмент);
- получение общих сведений о данных (вычисление описательных статистик для того, чтобы охарактеризовать данные);
- визуализация данных (поскольку картинка на самом деле стоит тысячи слов);
- моделирование данных (нахождение зависимостей и тестирование гипотез);
- оформление результатов (подготовка таблиц и диаграмм достаточного для публикации качества).

Схема анализа данных



Зарубежные (иностранные) пакеты: STATGRAPHICS, SPSS, STATA, SAS, STATISTICA, EViews, Kxen, S-plus и т.д.
в основной своей массе не русифицированы, высокая стоимость

Отечественные пакеты: Deductor, Prognoz Platform, STADIA, ЭВРИСТА, МИЗОЗАВР, ОЛИМП: Стат-Эксперт, Статистик-Консультант, САНИ, КЛАСС-МАСТЕР и т.д.
небогатый инструментарий, отсутствие методического обеспечения, часто технической поддержки. Большинство перечисленных некоммерческих отечественных продуктов (если не все) прекратило свое существование.

Бесплатные пакеты: R, Python, Rapid Miner, BV4.1, GeoDA, Winpepi, Epi Info, X-12-ARIMA и др.

Платные: Deductor, Prognoz Platform, SPSS, STATA, SAS, STATISTICA, EVIEWS, Maple, Mathematica, MATLAB и др.

Многие бесплатные пакеты характеризуются «скучным» набором статистических методов и узко направленны.

Среди бесплатных пакетов есть и «лидеры» (R, Python, Rapid Miner) по количеству реализуемых методов, приближающихся к продуктам второй группы.

Обзор статистических пакетов программ



Универсальные пакеты (общего назначения или профессиональные): SPSS, STATA, STATISTICA, S-PLUS, SAS, Deductor, Prognoz Platform и др.

Специализированные пакеты: BioStat, EQS, ЭВРИСТА, GWR4, GeoDA, Arrow Model и др.

Табличные редакторы (процессоры) - Excel (Microsoft Office), Calc (OpenOffice), Lotus 1-2-3 (Lotus SmartSuite); Quattro Pro (WordPerfect Office); Numbers (iWork).

Математические пакеты программ (MathCad, Maple, MATLAB, Mathematica и др.) позволяют проводить аналитические исследования любой сложности, рассчитанны на исследователя с обширными математическими знаниями.

- С **открытым программным кодом** (open-source software), в этой группе лидерами являются пакеты R и Python.
- С **закрытым программным кодом** (SAS, STADIA, SPSS, STATA и др.).

- Программы для классического статистического анализа - SPSS, Statistica, Stata и т.д.

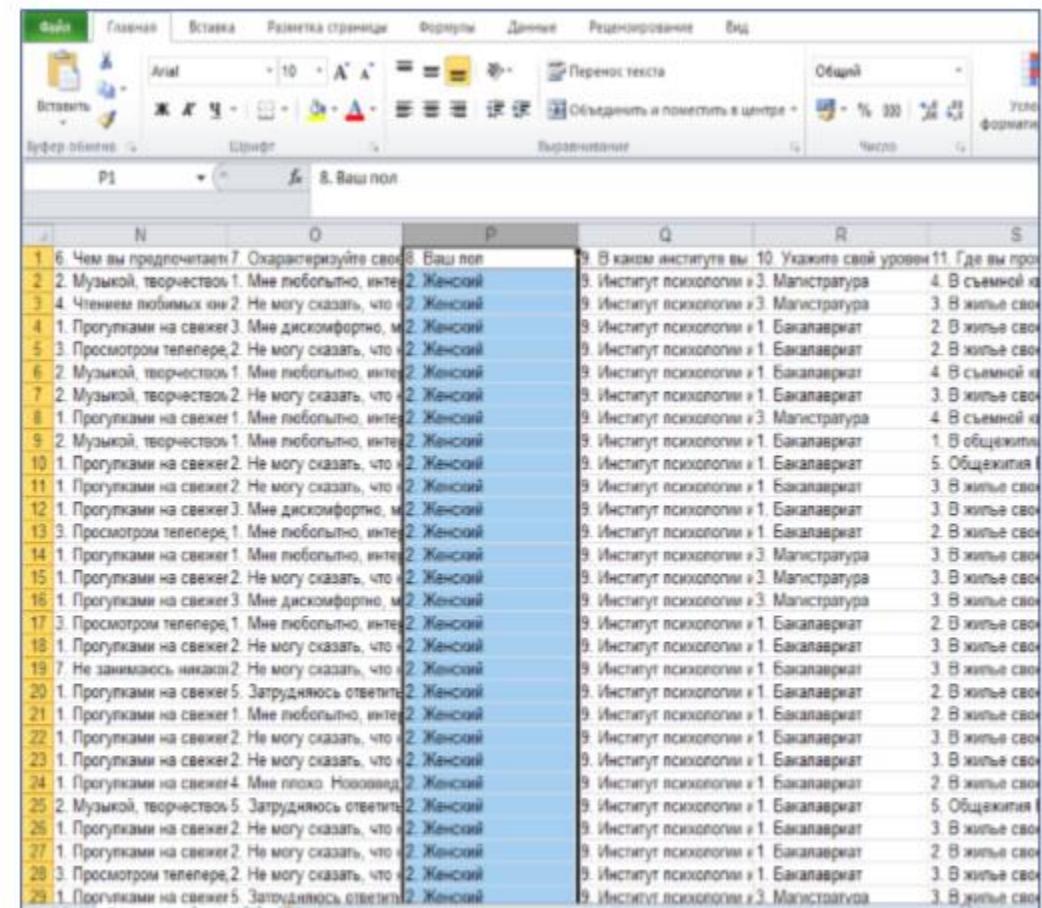
- Программы для «добычи данных» - SAS Enterprise Miner, SPSS Modeler, Rapid Miner, Statistica Data mining, R, Python, KNIME, Prognoz Platform и др.

Минимальный набор статистических методов анализа, который включен во все непосредственно статистические пакеты:

- описательная статистика (базовые статистические методы, проверка нормальности распределения данных);
- дисперсионный анализ;
- непараметрическая статистика (анализ таблиц сопряженности, непараметрические сравнения, дисперсионный анализ);
- контроль качества;
- анализ выживаемости;
- кластерный анализ;
- факторный анализ;
- дискриминантный анализ;
- регрессионный анализ;
- обработка данных (сортировка, отбор, трансформация данных).

Обзор статистических пакетов программ

- Самый доступный пакет, но не специализированный
- Дружелюбный и достаточно интуитивный интерфейс (переведенный на русский язык)
Сложные математические расчеты являются дополнительными функциями и реализованы в виде встроенных формул
- Существует и возможность автоматизации расчетов и визуализации данных с помощью языка программирования VBA.
- Присутствует коммерческий пакет XLStat, позволяющий реализовать основные процедуры статистического анализа, доступные в универсальных программах, прямо в Excel.

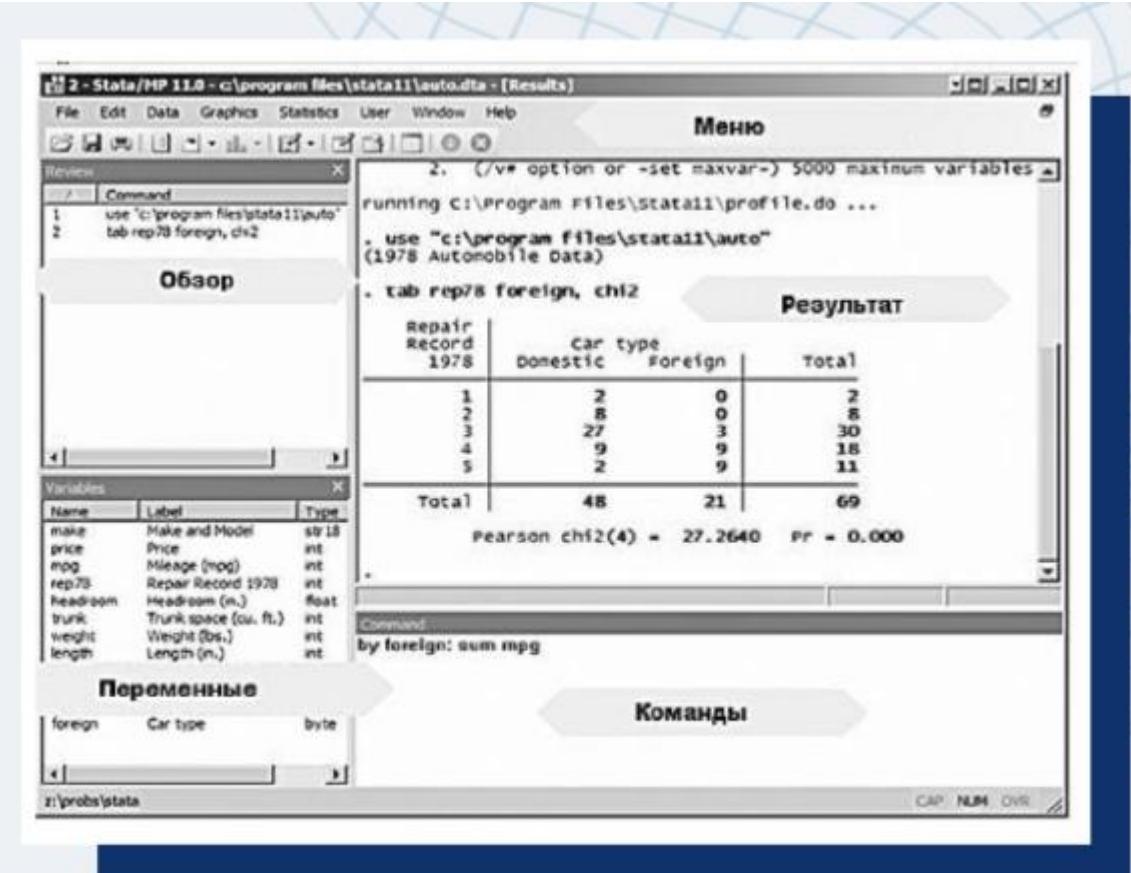


N	O	P	Q	R	S
1	6. Чем вы предпочитаете 7. Охарактеризуйте свое	8. Ваш пол	9. В каком институте вы	10. Укажите свой уровень	11. Где вы про
2	Музыкой, творчеством 1. Мне любопытно, инте	2. Женский	9. Институт психологии » 3. Магистратура	4. В съемной к	
3	Чтением любимых книг 2. Не могу сказать, что	2. Женский	9. Институт психологии » 3. Магистратура	3. В жилье сво	
4	1. Прогулками на свежем 3. Мне дискомфортно, м	2. Женский	9. Институт психологии » 1. Бакалавриат	2. В жилье сво	
5	3. Просмотром телепередач 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	2. В жилье сво	
6	6. Музыкой, творчеством 1. Мне любопытно, инте	2. Женский	9. Институт психологии » 1. Бакалавриат	4. В съемной к	
7	7. Музыкой, творчеством 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
8	1. Прогулками на свежем 1. Мне любопытно, инте	2. Женский	9. Институт психологии » 3. Магистратура	4. В съемной к	
9	9. Музыкой, творчеством 1. Мне любопытно, инте	2. Женский	9. Институт психологии » 1. Бакалавриат	1. В общежитии	
10	10. Прогулками на свежем 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	5. Общежитие	
11	11. Прогулками на свежем 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
12	12. Прогулками на свежем 3. Мне дискомфортно, м	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
13	13. Просмотром телепередач 1. Мне любопытно, инте	2. Женский	9. Институт психологии » 1. Бакалавриат	2. В жилье сво	
14	14. Прогулками на свежем 1. Мне любопытно, инте	2. Женский	9. Институт психологии » 3. Магистратура	3. В жилье сво	
15	15. Прогулками на свежем 2. Не могу сказать, что	2. Женский	9. Институт психологии » 3. Магистратура	3. В жилье сво	
16	16. Прогулками на свежем 3. Мне дискомфортно, м	2. Женский	9. Институт психологии » 3. Магистратура	3. В жилье сво	
17	17. Просмотром телепередач, 1. Мне любопытно, инте	2. Женский	9. Институт психологии » 1. Бакалавриат	2. В жилье сво	
18	18. Прогулками на свежем 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
19	19. Не занимаясь никаким 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
20	20. Прогулками на свежем 5. Затрудняюсь ответить	2. Женский	9. Институт психологии » 1. Бакалавриат	2. В жилье сво	
21	21. Прогулками на свежем 1. Мне любопытно, инте	2. Женский	9. Институт психологии » 1. Бакалавриат	2. В жилье сво	
22	22. Прогулками на свежем 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
23	23. Прогулками на свежем 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
24	24. Прогулками на свежем 4. Мне плохо Новощад	2. Женский	9. Институт психологии » 1. Бакалавриат	2. В жилье сво	
25	25. Музыкой, творчеством 5. Затрудняюсь ответить	2. Женский	9. Институт психологии » 1. Бакалавриат	5. Общежитие	
26	26. Прогулками на свежем 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
27	27. Прогулками на свежем 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	2. В жилье сво	
28	28. Просмотром телепередач, 2. Не могу сказать, что	2. Женский	9. Институт психологии » 1. Бакалавриат	3. В жилье сво	
29	29. Прогулками на свежем 5. Затрудняюсь ответить	2. Женский	9. Институт психологии » 3. Магистратура	3. В жилье сво	

Обзор статистических пакетов программ

Пакет Stata позиционируется как инструмент анализа, предназначенный для специалистов, которые занимаются научными исследованиями.

По мнению разработчиков, благодаря гибкой модульной структуре пакет применим для анализа данных из различных областей знаний: общественные науки (экономика, социология, политология и пр.), медицина (биостатистика, эпидемиология и пр.) и т.д.



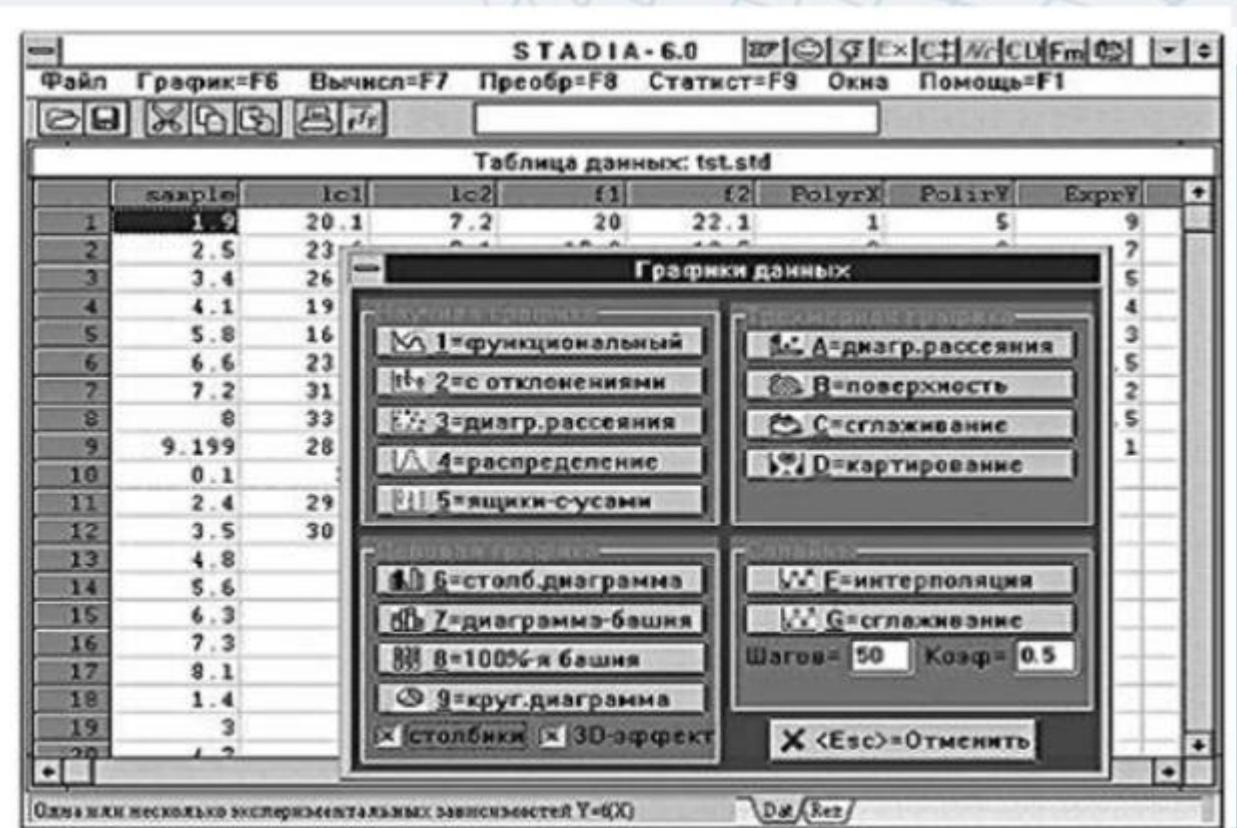
Обзор статистических пакетов программ (Stata)

Достоинства	Недостатки
Широкий набор средств статистического анализа и возможностей по управлению данными	Отсутствие возможности полноценного экспорта и импорта данных в базы данных, электронные таблицы и текстовые процессоры;
Встроенное "базовое справочное руководство" содержащее подробные примеры и ссылки на литературу по статистике	Возможность прямого открытия в программе лишь файлов с разрешением .Dta.
Возможность использования меню или программирования последовательности команд + возможность написания собственных программных модулей	
Экспорт результатов в MS office и SAS и совместимость с операционными системами Windows, Macintosh и linux	
Для работы программы требуется лишь 512 MB оперативной памяти, сама программа занимает 250 MB на жестком диске	
Создание графики полиграфического качества	

Обзор статистических пакетов программ (Stadia)

Пакет ориентирован на массового пользователя, имеющего небольшой опыт как в статистическом анализе, так и в общении с персональным компьютером, но нуждающегося в быстром и удобном средстве оформления и обработки данных

Разработан в России



Обзор статистических пакетов программ (Stadia)



Достоинства	Недостатки
Наличие системы контекстной экранной помощи, включающей объемный гипертекстственный справочник и экспертную систему по выбору метода статистического анализа	Наличие только русскоязычной версии
Обработка больших объемов данных (до 32 000 наблюдений)	Несовместимость с операционными системами, отличными от Windows
Наличие режима выдачи оглавления архива данных с комментариями + селективный поиск файлов по контексту комментариев, присвоенных архиву с данными	
Для работы программы требуется лишь 8 МВ оперативной памяти, сама программа занимает 4.1 МВ на жестком диске компьютера	
Удобный экспорт данных и результатов	

Обзор статистических пакетов программ (SPSS)

Модульная программа. Основа **SPSS Base**.

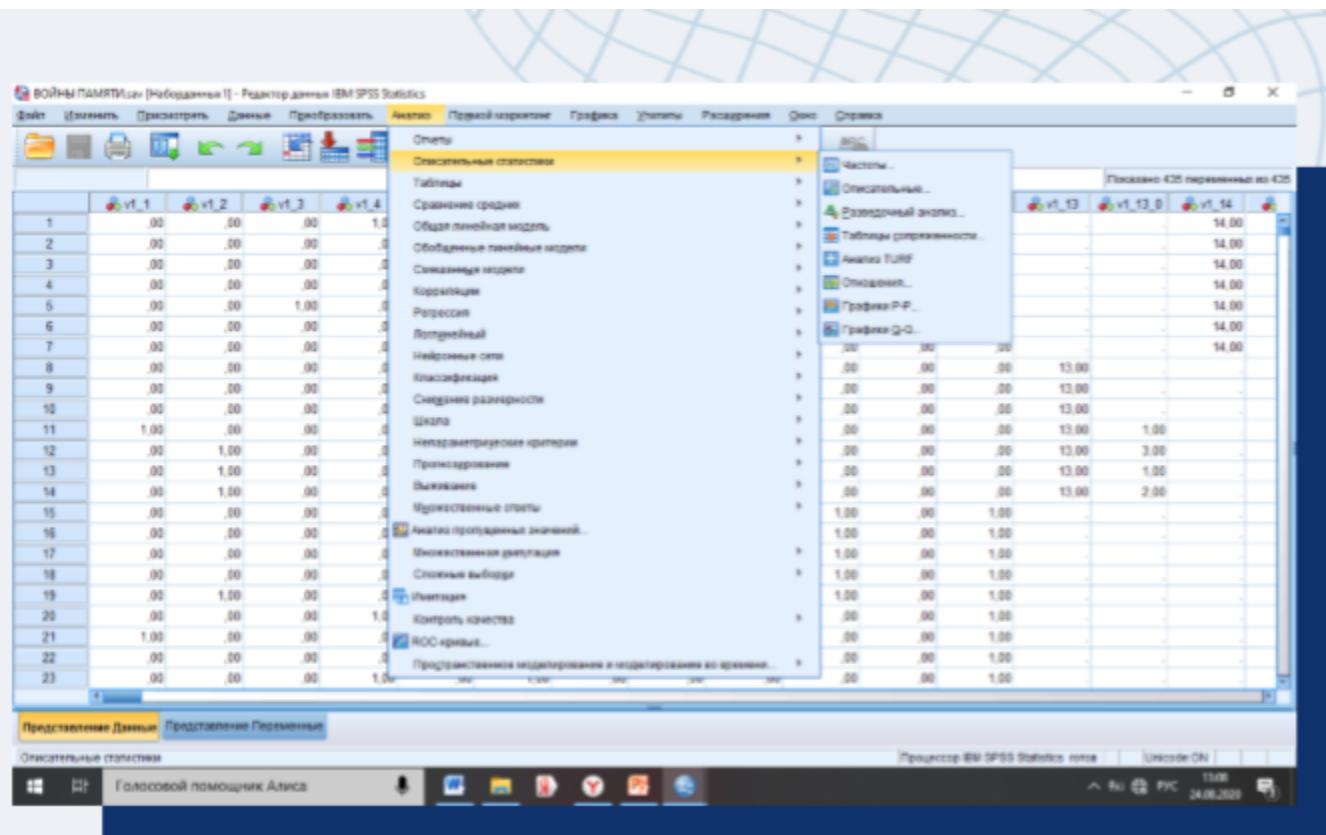
IBM SPSS Advanced Statistics предназначен для проведения анализа сложных взаимосвязей при помощи процедур. В модуль встроены мощные инструменты построения моделей.

IBM SPSS Bootstrapping ("Самогенерация") позволяет проверять устойчивость построенных моделей.

IBM SPSS Direct Marketing ("Прямой маркетинг") возможность самостоятельно выполнять основные виды анализа.

IBM SPSS Data Entry автоматизирует процесс разработки анкеты и ввода результатов опросов.

Новые модули пакета также позволяют собирать, структурировать и анализировать данные веб-пространства и работать с большими данными: SPSS Modeler, SPSS Analytic Server, SPSS Collaboration and Deployment Services, SPSS Analytic Catalyst. Есть интеграция с R и Python



Обзор статистических пакетов программ (SPSS)



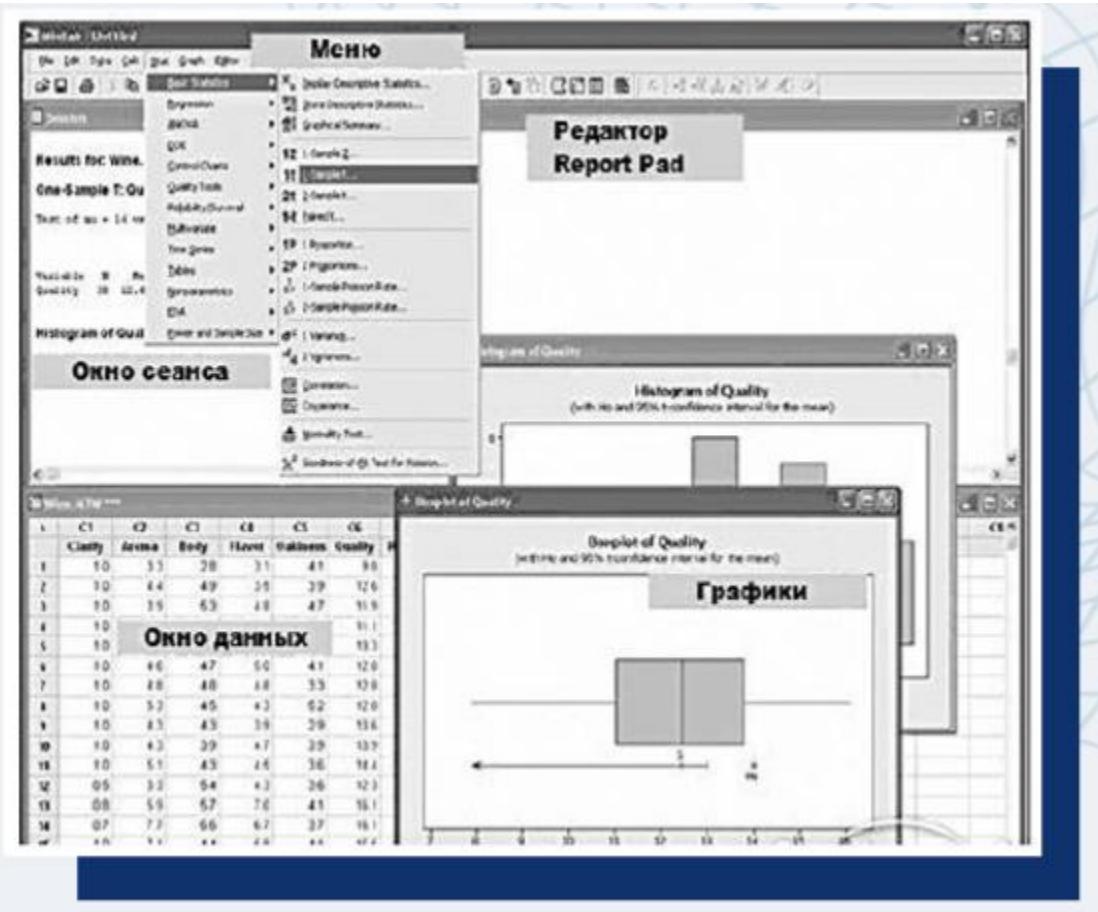
Достоинства	Недостатки
Возможность использовать как меню, так и командный синтаксис (syntax) SPSS (является своего рода языком программирования) для быстрой работы с повторяющимися операциями на обновлённом либо новом массиве данных, создания и применения пользовательских функций и т.д	Высокие требования к системе компьютера (требуется минимум 1GB оперативной памяти, 800MB памяти на жестком диске и процессор с частотой 1ghz и выше);
Совместимость с операционными системами Windows, Mac, Linux	Высокая цена по сравнению со статистическими пакетами аналогичного уровня.
Детальная контекстно-ориентированная справочная система + наличие значительного количества литературы по работе с пакетом	
Высокая скорость вычислений, простой и удобный интерфейс	
Широкий набор статистических и графических процедур (более 50 типов диаграмм) анализа данных, а также процедур создания отчетов	

Обзор статистических пакетов программ (Minitab)

Minitab позиционируется разработчиками как статистический пакет для аналитической работы на современных предприятиях.

Возможности:

- управление процессом статистической обработки данных;
- оценка мощности и объема выборки;
- планирование экспериментов;
- матричные функции;
- анализ измерительных систем, анализ надежности/выживаемости, анализ временных рядов и прогнозирование, многомерный анализ.

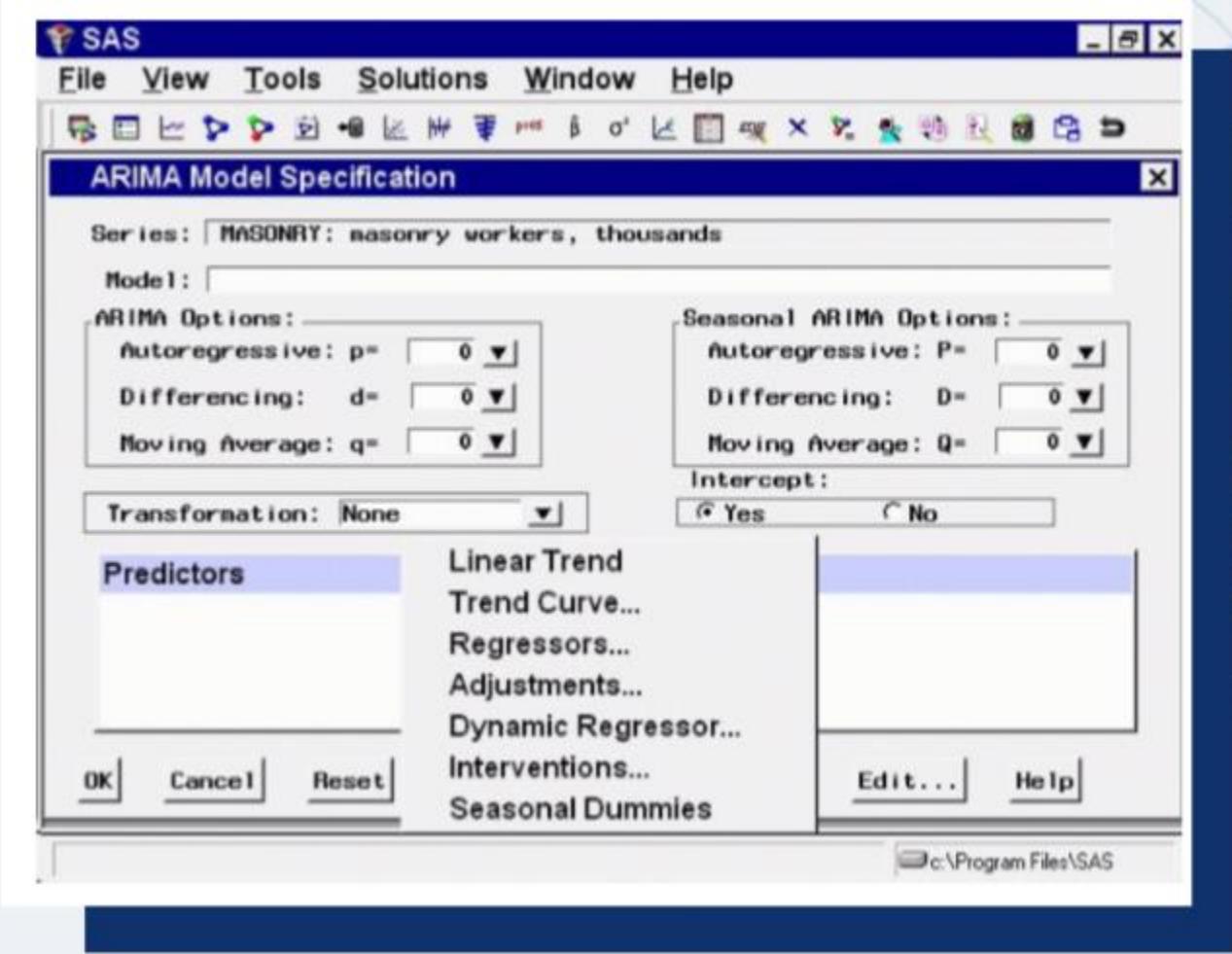


Обзор статистических пакетов программ (Minitab)

Достоинства	Недостатки
Система консультационной поддержки пользователя и интерпретация полученных результатов	Несовместимость с операционными системами, отличными от Windows
Осуществление импорта данных из текстовых и табличных процессоров, html-файлов, сохранение результатов анализа в форматах TIFF, JPEG, PNG, BMP, GIF, EMF	
Удобный инструмент для размещения нескольких графиков на одной странице, автоматическое обновление графиков при изменении исходных данных, создание и вращение трехмерных графиков	
Автоматизация заданий и создание новых функций с помощью языка макропрограммирования	
Для работы требуется лишь 512 МВ оперативной памяти, сама программа занимает 160 МВ на жестком диске компьютера	
Работа с файлами, содержащими до 4000 переменных и неограниченное число наблюдений	

Обзор статистических пакетов программ (Sas)

Пакет эффективно работает с данными различных типов:
маркетинговые базы данных, результаты клинических испытаний, медико-санитарных обследований, исследований предпочтений потребителей, исследований рынка ценных бумаг и пр. Встроенные инструменты статистического анализа могут быть применены для решения широкого круга вопросов, относящихся к различным областям деятельности



Обзор статистических пакетов программ (Sas)

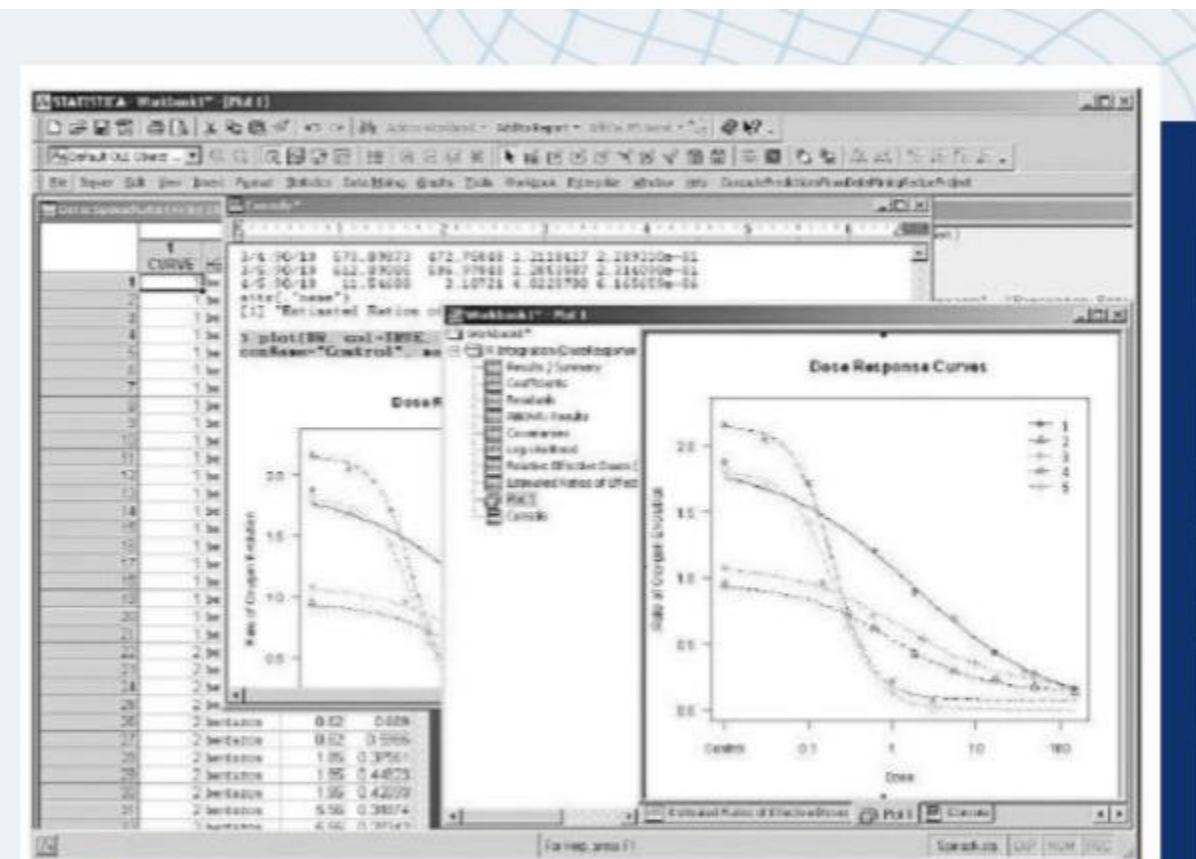


Достоинства	Недостатки
Быстрая обработка очень больших объемов данных	Сложен в освоении для неспециалистов
Возможность преобразования математических формул в программный код	Высокая стоимость
Создание пользовательских модулей	
Получение консультативной помощи в выборе методов анализа и в интерпретации его результатов, а также рекомендаций по дальнейшей работе с исходными данными	
Совместимость с операционными системами Linux и Windows	

Обзор статистических пакетов программ

Продукты серии **STATISTICA** различной комплектации Desktop, Analyst, Modeler, Data Scientist, Vizualization Designer включают широкий спектр мощных аналитических инструментов, состоящий из множества блоков. Помимо общих, в системе имеются специализированные модули, например, для проведения социологических или биомедицинских исследований, решения технических и промышленных задач, – карты контроля качества, модули анализа процессов и планирования.

Есть версия для обучения основам статистических методов – *Student Edition of STATISTICA*. Эта версия представляет собой урезанный вариант пакета и позволяет анализировать файлы данных, включающих не более 400 наблюдений.



Обзор статистических пакетов программ

Достоинства	Недостатки
Программа способна обрабатывать большие массивы данных – базы данных с числом переменных до 32 000 и практически неограниченным числом наблюдений.	Имеет достаточно высокие системные требования для установки и работы
Реализован обмен данными между STATISTICA и Windows-приложениями	Высокая стоимость
Возможность расширения системы при помощи создания программ на встроенным в STATISTICA языке программирования + запись макросов	
Построение графических объектов и анализ данных в пакете тесно интегрированы	
Возможности визуализации: несколько сотен типов графиков, возможность разработки собственного дизайна графика. Средства управления графиками позволяют работать одновременно с несколькими графиками	

Наиболее широко используемые в анализе больших данных языки программирования Python и R

- R и Python имеют простой синтаксис
- Бесплатны
- Существует множество готовых решений для визуализации
- Большинство коллег гуманитариев используют R или Python
- Доступно множество статей, книг, лекций и курсов по R и Python развитое Python-комьюнити и активно развивающееся русскоязычное R-сообщество
- Существует множество библиотек и готовых решений для парсинга сайтов (сбора любой открытой информации) и социальных сетей



Основные преимущества Python:

- язык высокого уровня, позволяющий писать простые, но функциональные и эффективные скрипты по адаптации и преобразованию данных (в том числе с онлайн-ресурсами)
- Ключевое преимущество данного языка программирования – понятный и легко воспринимаемый синтаксис, простота освоения.
- у Facebook есть открытая библиотека для Python. (<https://facebook-sdk.readthedocs.io/en/latest/>)
- программная библиотека на языке Python - Pandas. Он включает в себя огромный функционал для различных манипуляций с данными: статистическая обработка, очистка, трансформация, агрегация. Pandas может стать достойной заменой Excel
- Python обладает большими возможностями и для визуализации данных. Можно начать с освоения библиотеки Bokeh или Chartify.

Основные недостатки Python:

- сложнее для изучения чем R
- для эффективного решения задач требуется знание расширений и дополнений

Основные преимущества R:

- большое количество инструментов по адаптации и преобразованию данных и гибкая настройка среды для решения специфических задач;
- большое число модулей и пакетов, позволяющих использовать самые современные методы анализа данных и их визуализации;
- возможность самостоятельно создавать скрипты, модули, пакеты под свои задачи;
- возможность напрямую считывать машиночитаемые данные с Интернет-ресурсов и работать с ними, как с обычными данными;
- интеграция с текстовыми и издательскими системами, такими, как LaTex, Microsoft Word, позволяющая создавать аналитические отчеты непосредственно в среде R
- интеграция с Google Analytics, Яндекс.Метрикой

Основные недостатки R:

- R – узкоспециализированный язык
- для эффективного решения задач требуется знание расширений и дополнений
- часть скриптов на R написаны учеными, а не программистами, поэтому возможны сложности с чтением кода
- широкий спектром проблем по управлению открытыми данными, которые невозможно решить средствами только R.

R – это мощный язык для статистических вычислений и графики, который может справиться поистине с любой задачей в области обработки данных. Он работает во всех важных операционных системах и поддерживает тысячи специализированных модулей и утилит. Все это делает R замечательным средством для извлечения полезной информации из гор сырых данных

R — среда для реализации статистической обработки данных классическими методами и авторскими статистическими алгоритмами

R можно бесплатно скачать из «сетевого архива R» (Comprehensive R Archive Network, CRAN) по адресу <http://cran.r-project.org>.
(Скачать и установить последнюю версию R из архива CRAN
(<http://cran.r-project.org/bin/>))

Предварительно скомпилированные загрузочные файлы
доступны для Linux, Mac OS X и Windows.

Скачать и установить
последнюю версию дополнительных пакетов можно при помощи
функции [update.packages\(\)](#)

R – мощная программа с таким большим числом доступных аналитических и графических функций (по последним подсчётам их более 50 000), что она может в одинаковой степени навести ужас и на новичков, и на опытных пользователей.



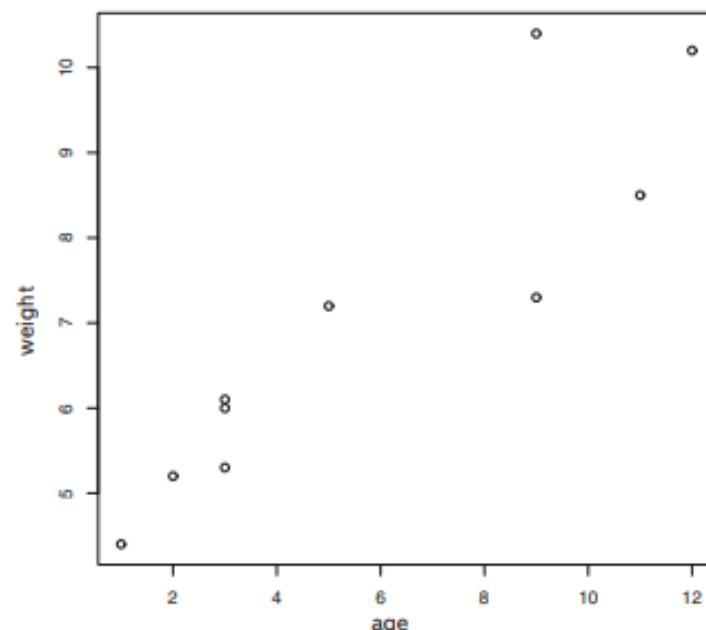
- Большинство коммерческих статистических программ стоят тысячи, если не десятки тысяч долларов. R – это **бесплатная программа!**
- R – это **мощная статистическая программа**, в которой реализованы **все способы анализа данных**;
- R имеет современные графические возможности. Если вам нужно визуализировать сложные данные, то учтите, что **в R реализованы самые разнообразные и мощные методы анализа данных из доступных**;
- Получение данных из разных источников в пригодном для использования виде может быть сложной задачей. R **может импортировать данные из самых разных источников**, включая текстовые файлы, системы управления базами данных, другие статистические программы и специализированные хранилища данных. R может также записывать данные в форматах всех этих систем;
- R представляет собой **не имеющую аналогов платформу для простого написания программ**, реализующих **новые статистические методы**;
- в R реализованы сложные статистические процедуры, ещё недоступные в других программах. На самом деле новые функции становятся доступными **для скачивания еженедельно**.
- если вы не хотите учить новый язык, существует множество графических пользовательских интерфейсов, в которых мощь R реализована в форме меню и диалогов (**R-Studio**);
- R работает на **разных операционных системах**, включая Windows, Unix и Mac OS X.

Программный код 1.1. Пример сессии в R

```
> age <- c(1,3,5,2,11,9,3,9,12,3)
> weight <- c(4.4,5.3,7.2,5.2,8.5,7.3,6.0,10.4,10.2,6.1)
> mean(weight)
[1] 7.06
> sd(weight)
[1] 2.077498
> cor(age, weight)
[1] 0.9075655
> plot(age, weight)
> q()
```

Из программного кода 1.1 видно, что средний вес этих 10 младенцев составляет 7.06 кг, а стандартное отклонение равно 2.08 кг и что существует сильная линейная взаимосвязь между возрастом и весом (коэффициент корреляции равен 0.91). Эта взаимосвязь также видна на диаграмме рассеяния на рис. 1.4. Неудивительно, что по мере взросления младенцы в среднем становятся тяжелее.

Функция	Действие
help.start()	Общая справка
help ("нечто") или ?нечто	Справка по функции нечто (кавычки необязательны)
help.search ("нечто") или ??нечто	Поиск в справке записей, содержащих нечто
example ("нечто")	Примеры использования функции нечто (кавычки необязательны)
RSiteSearch ("нечто")	Поиск записей, содержащих нечто в онлайн-руководствах и заархивированных рассылках
apropos ("нечто", mode="function")	Список всех доступных функций, в названии которых есть нечто
data ()	Список всех демонстрационных данных, содержащихся в загруженных пакетах
vignette ()	Список всех доступных руководств по загруженным пакетам
vignette ("нечто")	Список руководств по теме нечто





сайт: rstudio.com



Интегрированная среда разработки для R – языка программирования для статистической обработки данных и построения различных видов визуализации данных.

RStudio сочетает интуитивный пользовательский интерфейс с мощной консолью R. RStudio бесплатна, предназначена для пользователей Mac, Windows и Linux, позволяет использовать командную строку R в интуитивном настраиваемом интерфейсе.

Rstudio имеет более удобный интерфейс в сравнении с аналогами. Ряд особенностей, таких как цветовая подсветка, автоматическое завершение кода, удобная навигация по скрипту и другие, делают Rstudio привлекательной не только для новичков, но и для опытных программистов.

Устанавливается она после того, как уже установлен сам R.

Рабочее пространство – это текущая рабочая среда R в памяти вашего компьютера, которая включает в себя любые созданные пользователем объекты (векторы, матрицы, функции, таблицы данных или списки).

В конце каждой сессии вы можете **сохранить рабочее пространство**, и оно автоматически загрузится при следующем запуске программы.

Команды **интерактивно вводятся в ответ на приглашение к их вводу**. Можно использовать стрелки вверх и вниз для перемещения между введенными ранее командами. Это позволяет вызвать предыдущую команду, отредактировать ее и вновь выполнить ее, нажав клавишу Enter.

Текущая рабочая директория – это та директория, где находятся файлы данных и куда по умолчанию сохраняются результаты. Функция `getwd()` позволяет узнать, какая директория в данный момент является рабочей.

Вы можете назначить рабочую директорию при помощи функции `setwd()`. Если появляется необходимость импортировать файл, который находится не в рабочей директории, нужно написать полный путь к нему. Всегда заключайте в кавычки названия файлов и директорий.



Функция	Действие
<code>getwd()</code>	Вывести на экран название текущей рабочей директории
<code>setwd("моя_директория")</code>	Назначить <code>моя_директория</code> текущей рабочей директорией
<code>ls()</code>	Вывести на экран список объектов в текущем рабочем пространстве
<code>rm("список_объектов")</code>	Удалить один или несколько объектов
<code>help(options)</code>	Справка о возможных опциях
<code>options()</code>	Посмотреть или установить текущие опции
<code>history(#)</code>	Вывести на экран последние # команд (по умолчанию 25)
<code>savehistory("мой_файл")</code>	Сохранить историю команд в файл <code>мой_файл</code> (по умолчанию <code>.Rhistory</code>)
<code>loadhistory("мой_файл")</code>	Загрузить историю команд (по умолчанию <code>.Rhistory</code>)
<code>save.image("мой_файл")</code>	Сохранить рабочее пространство в файл <code>мой_файл</code> (по умолчанию <code>.Rdata</code>)
<code>save("список_объектов", file="мой_файл")</code>	Сохранить определенные объекты в файл
<code>load("мой_файл")</code>	Загрузить сохраненное рабочее пространство в текущую сессию (по умолчанию <code>.Rdata</code>)
<code>q()</code>	Выйти из программы. Появится вопрос, нужно ли сохранить рабочее пространство

Программный код 1.2. Пример действия команд, используемых для управления рабочим пространством R

```
setwd("C:/myprojects/project1")
options()
options(digits=3)
x <- runif(20)
summary(x)
hist(x)
savehistory()
save.image()
q()
```

Сначала назначена текущая рабочая директория, выведены на экран действующие значения параметров и указано, что числа должны выводиться с тремя знаками после десятичного разделителя. Затем создан вектор из 20 случайных чисел, для него вычислены основные статистики и построена гистограмма. Наконец, история команд сохранена в файл `.Rhistory`, рабочее пространство (включая объект `x`) сохранено в файл `.Rdata` и сессия завершена.

Ввод и вывод в R

1. По умолчанию запуск R начинает интерактивную сессию, где **ввод осуществляется с клавиатуры**, а результаты **выводятся на экран**.
 2. Можно также запустить команды **из ранее созданного скрипта** (файла, который содержит функции R), а вывод возможен напрямую в разные устройства.
- Ввод Функция `source("filename")` запускает скрипт. Если не прописан путь к файлу, подразумевается, что он находится в текущей рабочей директории. Например, команда `source("myscript.R")` запускает серию команд R, которые записаны в файле `myscript.R`.
3. Функция `sink("filename")` выводит все результаты в файл с названием `filename`. По умолчанию, если этот файл уже существует, новая версия записывается поверх старой. Параметр `append=TRUE` позволяет добавлять новый текст в файл, а не записывать его вместо старого текста. Параметр `split=TRUE` позволяет выводить результаты и на экран, и в текстовый файл.

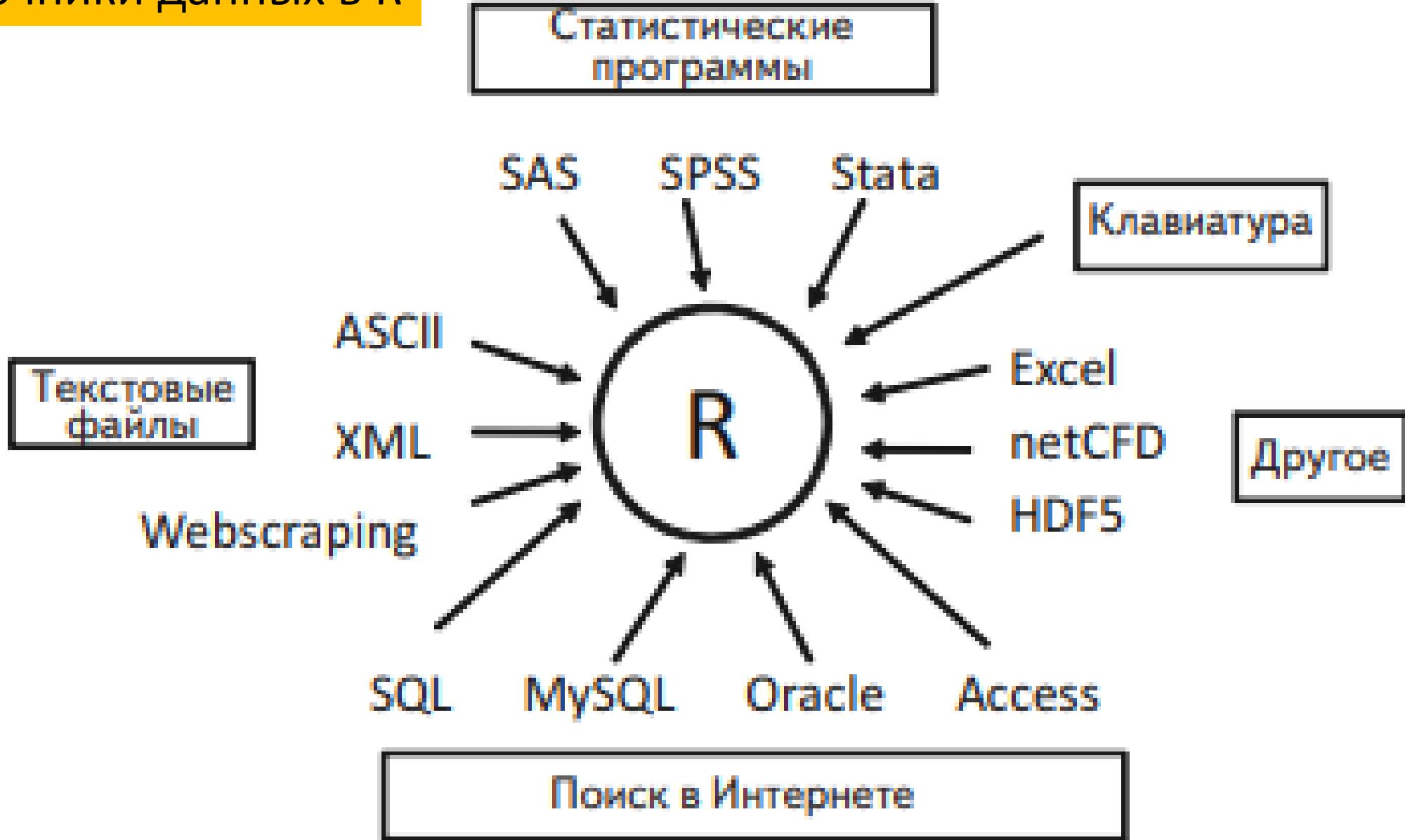
Выполнение команды `sink()` без аргументов восстановит вывод результатов только на экран.

Хотя команда `sink()` управляет выводом текста, она не оказывает никакого воздействия на вывод графики. Для управления **выводом изображений используйте одну из функций**



Функция	Вывод (формат графического файла)
<code>pdf("filename.pdf")</code>	PDF
<code>win.metafile("filename.wmf")</code>	Windows metafile
<code>png("filename.png")</code>	PNG
<code>jpeg("filename.jpg")</code>	JPEG
<code>bmp("filename.bmp")</code>	BMP
<code>postscript("filename.ps")</code>	PostScript

Источники данных в R



Распространенные ошибки при программировании в R

Существует ряд распространенных ошибок, которые часто допускают и новички, и опытные программисты на языке R. Если программа выдает сообщение об ошибке, проверьте, не сделали ли вы что-то из нижеперечисленного:

- использовали неправильный регистр: `help()`, `Help()` и `HELP()` – это три разные функции (только первая будет работать);
- забыли поставить кавычки там, где они необходимы: `install.packages("gclus")` работает, а `install.packages(gclus)` выдает сообщение об ошибке;
- забыли поставить скобки при обращении к функции: к примеру, нужно набирать `help()`, а не `help`. Даже если аргументы отсутствуют, скобки все равно необходимы;
- использовали `\` в указании пути к файлу в операционной системе Windows: R воспринимает обратный слэш как знак экранирования символов. `setwd("c:\\mydata")` порождает сообщение об ошибке. Используйте вместо этого `setwd("c:/mydata")` или `setwd("c:\\\\mydata")`;
- ввели функцию из пакета, который еще не загрузили. Функция `order.clusters()` содержится в пакете `gclus`. Если вы попробуете использовать ее до того, как загрузили этот пакет, появится сообщение об ошибке.

Сообщения об ошибках в R могут быть непонятными, однако появление многих из них можно предотвратить, если внимательно следовать вышеперечисленным правилам.

Пакеты в R

Пакеты – это собрания функций R, данных и скомпилированного программного кода в определенном формате.

Директория, в которой пакеты хранятся на вашем компьютере, называется библиотекой.

Существует более 2500 созданных пользователями модулей, называемых пакетами (packages), которые вы можете скачать с <http://cran.r-project.org/web/packages>.

Функция `.libPath()` показывает, где расположена ваша библиотека, а функция `library()` выводит на экран названия всех имеющихся в библиотеке пакетов.

R поставляется уже со стандартным набором пакетов (включая `base`, `datasets`, `utils`, `grDevices`, `graphics` и `methods`). В них уже содержатся разнообразные функции и наборы данных, доступных по умолчанию. Другие пакеты нужно скачивать и устанавливать. После установки они загружаются во время сессии по мере необходимости.

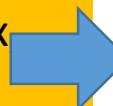
Команда `search()` выводит на экран названия загруженных и готовых к использованию пакетов.

Для установки пакета используйте команду `install.packages()`. Эта команда, введённая без аргументов, вызовет список зеркал сайта CRAN. После выбора зеркала вы увидите список всех доступных пакетов. Выберите один из них, и он будет скачан и загружен.

Функция `help(package="название пакета")` выводит короткое описание этого пакета и алфавитный указатель всех входящих в него функций и наборов данных.

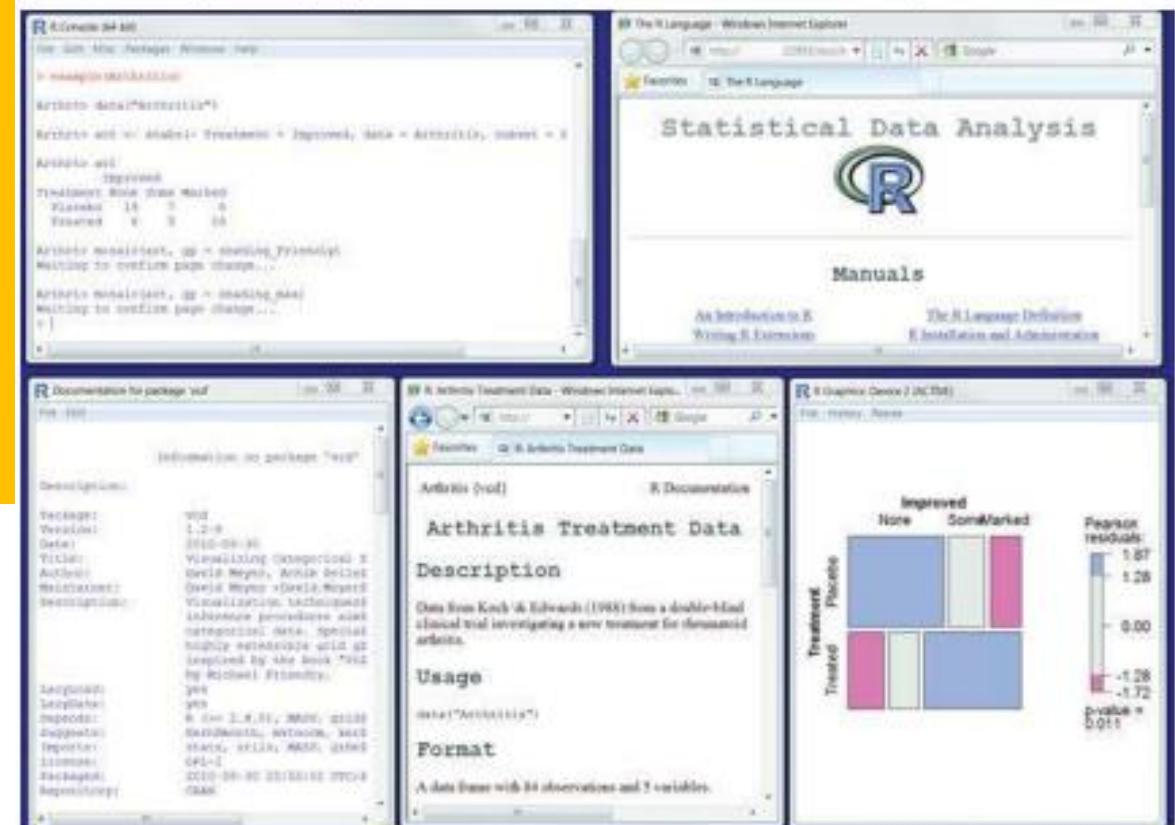
Задание:

1. Откройте общий файл справки и загляните в раздел «Введение в R».
2. Установите пакет vcd (пакет для визуализации категориальных данных).
3. Просмотрите список всех функций и наборов данных содержащихся в пакете.
4. Загрузите пакет и прочтите описание набора данных Arthritis.
5. Выведите этот набор данных на экран (набрав его название в командной строке).
6. Запустите пример, который прилагается к набору данных Arthritis. Не беспокойтесь, если вы не понимаете результаты. В общих чертах они показывают, что страдающие артритом пациенты, которые получали лекарство, выздоравливали гораздо быстрее по сравнению с теми, которые получали плацебо.
7. Выходите из программы



Программный код 1.3. Работа с новым пакетом

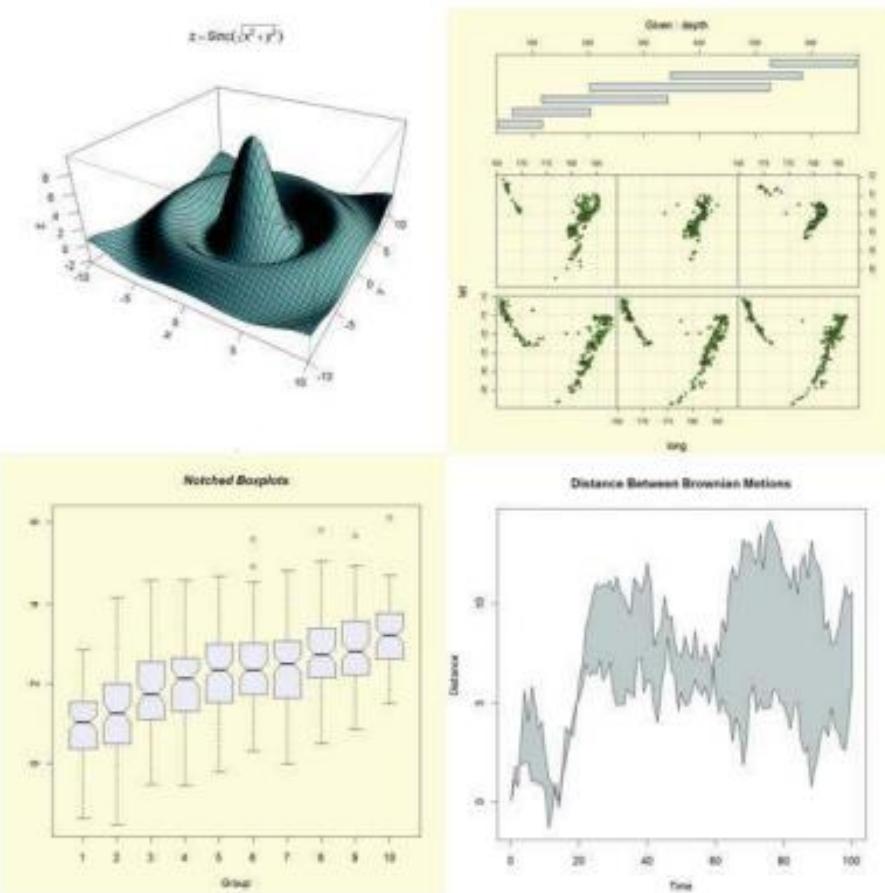
```
help.start()  
install.packages("vcd")  
help(package="vcd")  
library(vcd)  
help(Arthritis)  
Arthritis  
example(Arthritis)  
q()
```



Аналитические пакеты R для больших объемов данных

В R есть несколько пакетов, предназначенных для анализа больших объемов данных:

- Пакеты `biglm` и `speedglm` предназначены для подбора линейных или обобщенных линейных моделей для больших наборов данных с эффективным использованием памяти. В этих пакетах реализованы аналоги функций `lm()` и `glm()` для работы с большими объемами данных.
- В некоторых пакетах есть аналитические функции для работы с обширными матрицами, создаваемыми при помощи пакета `bigmemory`. Пакет `biganalytics` позволяет проводить кластерный анализ k -средних, вычислять статистики для переменных и создает программную оболочку для пакета `biglm`. Пакет `bigrtabulate` содержит аналоги функций `table()`, `split()` и `tapply()`, а пакет `bigalgebra` позволяет применять усовершенствованные функции линейной алгебры.
- В пакете `biglars` в сочетании с пакетом `ff` представлены возможности регрессии наименьшего угла (least-angle), лассо и ступенчатой регрессии для наборов данных, которые слишком велики для хранения в памяти.
- Пакет `Bigdingnag` может применяться для работы с большими числами (больше, чем 2^{1024}).





2. Понятие объекта в R

В R объектом (object) называется все, что может быть представлено в виде **переменных**, включая константы, разные типы данных, функции и даже диаграммы. У объектов есть вид (определяет, в каком виде объект хранится в памяти) и класс (который указывает общим функциям типа `print`, как с ним обращаться).



Язык R принадлежит к семейству высокоуровневых объектно-ориентированных языков программирования. Объекты - это все, что мы создаём в ходе работы с R. Выделяют два основных типа объектов:

- Объекты, предназначенные для хранения данных («*data objects*»)**
- Функции («*function objects*»)** – это поименованные программы, предназначенные для выполнения определённых действий над другими объектами.

В R все объекты имеют два обязательных атрибута: тип данных и длина. В зависимости от значений этих и других атрибутов объекты в R делятся на вектора (vector), факторы (factor), матрицы (matrix), массивы (array), таблицы (data.frame), списки (list) и пр. Данные в R не хранятся скалярно. Основные типы данных в R — это logical — логический, numeric — числовой (`integer` и `double`), character — символьный, complex — комплексный.

Справку по имеющимся функциям в R можно получить с помощью команды `help(имя_функции)` или `?имя_функции`.

Полезные функции для работы с объектами в R

Функция	Описание
<code>length(объект)</code>	Число элементов/компонентов объекта
<code>dim(объект)</code>	Число измерений объекта
<code>str(объект)</code>	Структура объекта
<code>class(объект)</code>	Класс или тип объекта
<code>mode(объект)</code>	Способ хранения (вид) объекта
<code>names(объект)</code>	Названия частей объекта
<code>c(объект, объект, ...)</code>	Объединяет объекты в вектор
<code>cbind(объект, объект, ...)</code>	Объединяет объекты в виде столбцов
<code>rbind(объект, объект, ...)</code>	Объединяет объекты в виде строк
<code>объект</code>	Выводит на экран весь объект
<code>head(объект)</code>	Выводит на экран первую часть объекта
<code>tail(объект)</code>	Выводит на экран последнюю часть объекта
<code>ls()</code>	Выводит на экран список имеющихся объектов
<code>rm(объект, объект, ...)</code>	Удаляет один или более объектов. Команда <code>rm(list = ls())</code> удалит почти все объекты из рабочего пространства
<code>новый_объект <- edit(объект)</code>	Редактирует объект и сохраняет результат в виде нового объекта
<code>fix(объект)</code>	Редактирует объект

3. Виды представления данных в R

Типы данных в R бывают числовыми (numeric), текстовыми (character), логическими (TRUE/FALSE, правда/ложь), комплексными (мнимое число) и необработанными (байты).

R работает с самыми разными структурами данных, включая **скаляры**, **векторы**, **матрицы**, **массивы данных**, **таблицы данных** и **справочники**. Они различаются **типами данных**, **способом создания**, **сложностью устройства**, а также **способом обозначать и извлекать их отдельные элементы**.

Первый этап любого анализа данных – создание набора данных, в котором содержится информация для изучения, в подходящем формате.

В R эта задача распадается на следующие:

- выбор типа данных;
- ввод или импорт данных в выбранном формате

Набор данных – это, как правило, прямоугольный массив данных, в котором ряды соответствуют наблюдениям, а столбцы – признакам (таблица данных).

Вектор представляет собой поименованный **одномерный объект**, содержащий набор однотипных элементов (числовые, логические, либо текстовые значения - никакие сочетания не допускаются). Для создания векторов небольшой длины в R используется **функция конкатенации c()** (от "concatenate" – объединять, связывать).

В качестве аргументов этой функции через запятую перечисляют объединяемые в вектор значения, например:

`my.vector <- c(1, 2, 3, 4, 5)`, где my.vector – это переменная типа вектор, которая содержит числа 1, 2, 3, 4, 5

Для создания векторов, содержащих совокупность **последовательных чисел**, удобна функция **seq()** (от "sequence" – последовательность).

`my.vector <- seq(1,5)`, где my.vector – это переменная типа вектор, которая содержит числа 1, 2, 3, 4, 5.

Идентичный результат будет получен при помощи команды `my.vector <- 1:7`

В качестве дополнительного аргумента функции seq() можно задать шаг приращения чисел: `my.vector <- seq(from = 1, to = 5, by = 0.5)` Результат: 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

Векторы, содержащие одинаковые значения, создают при помощи функции **rep()** (от "repeat" – повторять). Например, для формирования текстового вектора Text, содержащего пять значений "test", следует выполнить команду `Text <- rep("test", 5)`, то Text содержит "test" "test" "test" "test" "test"

Матрица - это специальный вектор, который содержит два дополнительных атрибута: количество строк и количество столбцов.

Матрица в R создаётся функцией **matrix(data, nrow, ncol, ...)**. Данная функция заполняет по строкам матрицу размером $nrow \times ncol$ элементами вектора **data**. Если **data** содержит меньшее количество элементов, то данные повторяются, начиная с первого элемента.

Структура с одним типом данных внутри, но с тремя измерениями или больше, называется **массивом (array)**.

Создание многомерного массива из вектора данных **data** осуществляется вызовом функции **array(data, dim, ...)**, при этом в качестве аргумента **dim** передается вектор размеров массива в каждой размерности, например **dim = c(2,3,2)**.

Матрица (`matrix`) — это всего лишь “двумерный” вектор: вектор, у которого есть не только длина, но и ширина. Создать матрицу можно с помощью функции `matrix()` из вектора, указав при этом количество строк и столбцов. Все остальное так же как и с векторами: внутри находится данные **только одного типа**.

Пример создания матрицы:

```
A <- matrix(1:20, nrow=5,ncol=4)
```

```
A
```

Ответ в консоли:

```
[,1] [,2] [,3] [,4]
[1,] 1 6 11 16 [2,] 2 7 12 17
[3,] 3 8 13 18
[4,] 4 9 14 19
[5,] 5 10 15 20
```

Если мы знаем сколько значений в матрице и сколько мы хотим строк, то количество столбцов указывать необязательно

Поскольку матрица — это уже двумерный массив, то у него имеется два индекса. Эти два индекса разделяются запятыми.

Первый индекс — выбор строк, второй индекс — выбор колонок. Если же мы оставляем пустое поле вместо числа, то мы выбираем все строки/колонки в зависимости от того, оставили мы поле пустым до или после запятой

Пример создания трёхмерного массива:

```
array_3d <- array(1:12, c(3, 2, 2))
```

```
array_3d
```

Ответ в консоли:

```
> Aarray_3d <- array(1:12, c(3, 2, 2))
> Aarray_3d
, , 1

[,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

, , 2

[,1] [,2]
[1,]    7   10
[2,]    8   11
[3,]    9   12
```

Список (**list**) – это вектор без ограничения на одинаковые данные внутри.

В нем могут содержаться самые разные данные, в том числе и другие списки.

Функция, чтобы посмотреть, как он устроен - это **str()**.

Пример:

```
complex_list <- list(c("Wow", "this", "list", "is", "so", "big"), "16", simple_list)
str(complex_list)
```

Ответ в консоли:

```
## List of 3
## $ : chr [1:6] "Wow" "this" "list" "is" ...
## $ : chr "16"
## $ :List of 3
##   ..$ : num 42
##   ..$ : chr "Пам пам"
##   ..$ : logi TRUE
```

Чтобы добраться до самого элемента списка нужна не одна, а две квадратных скобочки **complex_list[[2]]**

Ответ в консоли: "16"

Списки довольно часто используются в R, но реже, чем в Python. Со многими объектами в R, такими как результаты статистических тестов, удобно работать именно как со списками. Но обычно после этого стоит как можно скорее превратить список в датафрейм.

Таблицы (data.frames) - в отличие от матрицы, разные столбцы могут иметь разные типы данных

Пример создания:

```
name <- c("Petr", "Eugeny", "Lena", "Misha", "Sasha")
age <- c(26, 34, 23, 27, 26)
student <- c(F, F, T, T, T)
df <- data.frame(name, age, student)
df
```

Ответ в консоли:

```
## 'data.frame': 5 obs. of 3 variables:
## $ name : chr "Petr" "Eugeny" "Lena" "Misha" ...
## $ age  : num 26 34 23 27 26
## $ student: logi FALSE FALSE TRUE TRUE TRUE
```

Команда View(df) дает таблицу:

```
> print.data.frame(df)
  name age student
1 Petr 26 FALSE
2 Eugeny 34 FALSE
3 Lena 23 TRUE
4 Misha 27 TRUE
5 Sasha 26 TRUE
```

	name	age	student
1	Petr	26	FALSE
2	Eugeny	34	FALSE
3	Lena	23	TRUE
4	Misha	27	TRUE
5	Sasha	26	TRUE

Фактор - векторный объект, используемый для спецификации дискретной классификации (группировки) компонентов других векторов одинаковой длины.

R поддерживает как упорядоченные, так и не упорядоченные факторы.

Создаются факторы с помощью функции factor():

```
> statef <- factor(state)
```



4. Среда программирования R

1. Функция объединения векторов C

Пример:

```
v1 <- c(1, 2, 3)
```

```
v2 <- c(4, 5, 6)
```

```
V <- c(v1, v2)
```

```
V
```

Ответ в консоли: 1 2 3 4 5 6

Ниже перечислены все используемые в R логические операторы:

- "Равно" ==
- "Не равно" !=
- "Меньше" <
- "Больше" >
- "Меньше либо равно" <=
- "Больше либо равно" >=
- "Логическое И" &
- "Логическое ИЛИ" |
- "Логическое НЕ" !

2. Функция выбора конкретного элемента вектора - необходимо указать имя вектора и индекс этого элемента в квадратных скобках:

Пример: создадим числовой вектор y, содержащий 5 числовых значений:

```
y <- c(5, 3, 2, 6, 1)
```

```
# проверим, чему равен третий элемент вектора y: y[3]
```

Ответ в консоли: 2

```
y[3:5]
```

Ответ в консоли: 2 6 1

Используя индексные номера, можно выполнять различные операции с выбранными элементами разных векторов:

```
y[1]*y[3]
```

Ответ в консоли: 10

```
y[c(1, 4)]
```

Ответ в консоли: 5 6

```
y[-c(1, 4)]
```

Ответ в консоли: 3 2 1

Пример: выберем из вектора y все значения >2: y[y>2]

Ответ в консоли: 5 3 6



c(...) — объединяет аргументы в вектор; ·
seq(from, to, by, length.out, ...) — генерирует последовательность из length.out
чисел от from до to с шагом by; ·
rep(x, times) — создаёт вектор из times копий элемента x.



3. Операция коррекции значения в векторе

Например, так можно исправить второе значение созданного нами ранее вектора у с 3 на 0.3:

```
y[2] <- 0.3
```

```
y
```

Ответ в консоли: 5.0 0.3 2.0 6.0 1.0

4. Функция упорядочения значений вектора по возрастанию или убыванию **sort()** в сочетании с аргументом decreasing = FALSE или decreasing = TRUE соответственно ("decreasing" значит «убывающий»):

Например:

```
sort(y, decreasing = FALSE)
```

Ответ в консоли: 1, 2, 3, 5, 6

```
sort(y, decreasing = TRUE)
```

Ответ в консоли: 6, 5, 3, 2, 1

5. Функция создания вектора длины length и инициализации его значениями по умолчанию: logical(length), numeric(length) и т. п.; vector(mode = "logical", length).

Пример: **v=logical(3)**

Ответ в консоли: FALSE FALSE FALSE

При работе со статистическими данными полезны бывают следующие функции:
`length(...)`, `sort(...)`, `max(...)`, `min(...)`, `range(...)`, `sum(...)`, `mean(...)` — возвращает среднее арифметическое элементов вектора, `prod(...)` — возвращает произведение элементов вектора, `rev(...)` — переставляет элементы вектора в обратном порядке, `rank(...)` — присваивает элементу вектора его позицию в ряду всех элементов, упорядоченных по возрастанию, `cumsum(...)` — возвращает вектор накопленных сумм и т. п.
Отметим также функцию `which(...)`, возвращающую индексы элементов `TRUE` логического вектора

Для выполнения **матричных операций** в R существуют следующие функции и операторы:

`%*%` — матричное умножение;

`t(...)` — транспонирование матрицы;

`diag(...)` — выделение главной диагонали матрицы или создание единичной матрицы;

`colSums(...), rowSums(...), colMeans(...), rowMeans(...)` — подсчет сумм или средних арифметических элементов по столбцам и строкам соответственно.

Функция `solve(a, b, ...)` решает матричное уравнение вида $a \%*% x = b$, при чем b может быть как вектором, так и матрицей. Вызов функции `solve(a)` позволяет найти матрицу, обратную к a .

Набор операций для работы с массивами аналогичен работе с матрицами.

Основные функции и операции с таблицами

Используя оператор `$` и присваивание можно создавать новые колонки датафрейма:

```
df$lovesR <- TRUE #правило recycling - узнали?  
df
```

name	age	student	lovesR
<chr>	<dbl>	<lg>	<lg>
Petr	26	FALSE	TRUE
Eugeny	34	FALSE	TRUE
Lena	23	TRUE	TRUE
Misha	27	TRUE	TRUE
Sasha	26	TRUE	TRUE

Узнаем, любят ли R те, кто моложе среднего возраста в группе:

```
> df[df$age < mean(df$age), 3]  
[1] FALSE TRUE TRUE TRUE
```



4. Циклы и функции

В рамках проекта R написано большое число пакетов со специализированными функциями для прикладного статистического анализа. Однако часто бывает полезно создать собственные пользовательские функции.

В языке R имеются все основные логические операторы ($>$, $=$, $<=$, $==$, $!=$) и операторы управления:

`if (cond) expr`, `if (cond) cons.expr else alt.expr` — выполняет команды `expr`, `cons.expr` или `alt.expr` в зависимости от логического значения условия `cond`; если `cond` имеет длину, большую единицы, то проверяется только первый его элемент; ·

`ifelse(test, yes, no)` — выполняет `yes` или `no` в зависимости от логического значения условия `test`; в отличии от оператора `if` может работать с переменными любой длины; ·

`for(var in seq) expr` или `while(cond) expr` — выполняют `expr`, пока `var` находится в рамках последовательности `seq` или выполнено условие `cond` соответственно; ·

`repeat expr` — запускает бесконечный цикл выполнения `expr`, выйти из которого можно с помощью оператора `break`; ·

`switch(expr, ...)` — выполняет команду из списка `...` с номером, являющимся результатом вычисления `expr`

Функция в R создается следующим образом: `function(arglist) expr`. Формальным аргументам из списка `arglist` может присваиваться значение по умолчанию в виде `arg = value`. Телом функции `expr` является команда или блок команд. Тело может содержать в конце команду `return(value)` для явного указания возвращаемого значения. В случае, если эта команда отсутствует, функция возвращает результат выполнения последней команды из тела. Приведем пример функции, реализующей алгоритм Евклида для поиска наибольшего общего делителя двух чисел.

```
1 > Euclid <- function(x, y)
2 + {while ((x != 0) && (y != 0)) if (x>y) x <- x%%y else y<-y%%x;
3 + return (x+y)}
4 > Euclid(25367, 6375)
5 [1] 1
```

Отметим, что тип данных для аргументов не указывается, поэтому в качестве аргумента может также выступать любая другая функция.



Основные типы данных в R

— это:

`logical` — логический,

`numeric` — числовой (`integer` и
`double`),

`character` — символьный,

`complex` — комплексный.

- Большие объекты данных обычно читают как значение из внешних файлов, а не вводиться во время сеанса R с клавиатуры.
- По умолчанию числовые элементы (кроме меток строки) считаются как числовые переменные

Функция `read.table()`

Price	Floor	Area	Rooms	Age	Cent.heat
52.00	111.0	830	5	6.2	no
54.75	128.0	710	5	7.5	no
57.50	101.0	1000	5	4.2	no
57.50	131.0	690	6	8.8	no
59.75	93.0	900	5	1.9	yes

Затем фрейм данных может быть считан как:

```
> HousePrice <-read.table ("houses.data", header=TRUE)
```

где опция `header=TRUE` указывает, что первая строка - строка заголовков, что следует из формы файла, и что отсутствуют явные метки строки.

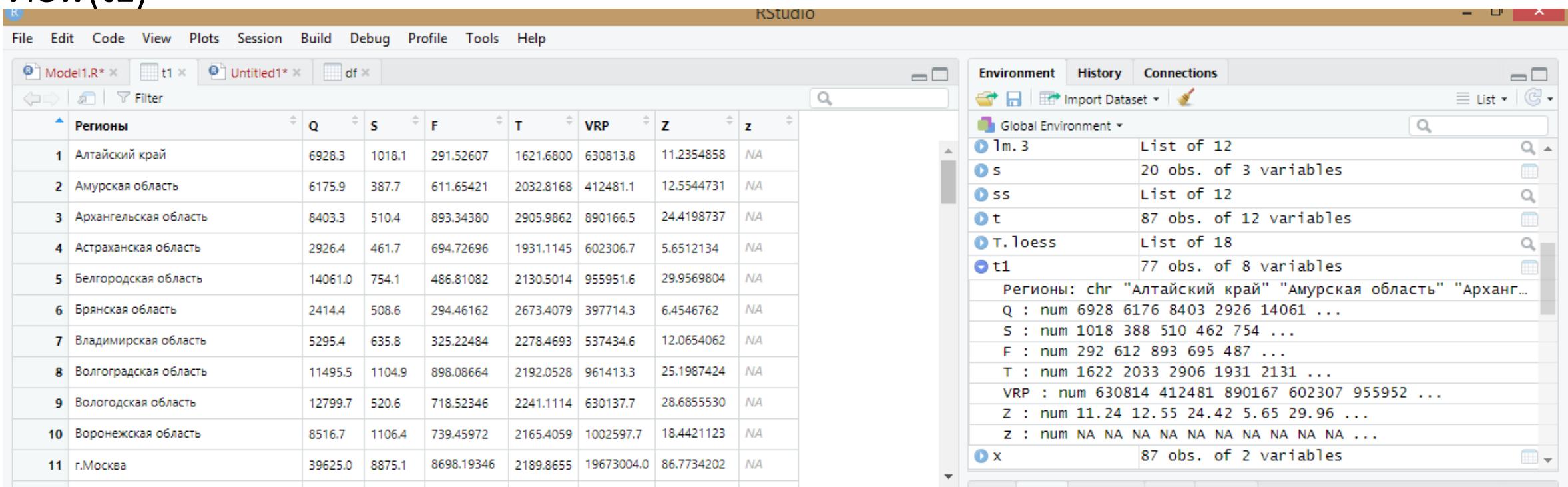
#Ввод данных из MSExcel

```
library(readxl)
```

```
types = c("text", rep("numeric", 7))
```

```
t1 <- as.data.frame(read_excel("C:/Users//компьютер/Documents/ALLData.xlsx", 1,  
                                col_types = types))
```

```
View(t1)
```



The screenshot shows the RStudio interface with the following details:

- File menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Global Environment pane:**
 - Global Environment dropdown.
 - Variables listed:
 - l1m.3: List of 12
 - s: 20 obs. of 3 variables
 - ss: List of 12
 - t: 87 obs. of 12 variables
 - t.loess: List of 18
 - t1: 77 obs. of 8 variables
 - Details for t1:
 - Регионы: chr "Алтайский край" "Амурская область" "Арханг...
 - Q : num 6928 6176 8403 2926 14061 ...
 - S : num 1018 388 510 462 754 ...
 - F : num 292 612 893 695 487 ...
 - T : num 1622 2033 2906 1931 2131 ...
 - VRP : num 630814 412481 890167 602307 955952 ...
 - Z : num 11.24 12.55 24.42 5.65 29.96 ...
 - z : num NA NA NA NA NA NA NA NA NA ...
 - x: 87 obs. of 2 variables

В пакете R присутствует набор функций для работы с некоторыми типовыми дискретными и непрерывными одномерными распределениями вероятностей

Таблица 1: Типовые распределения в R

Имя	Плотность (вероятность)	Имя в R	Параметры				
Биномиальное	$C_n^k p^k (1-p)^{n-k}$ $0 \leq p \leq 1, k = 0, 1, \dots, n$	binom	<code>size = n,</code> <code>prob = p</code>	Гамма	$\frac{x^{a-1}}{s^a \Gamma(a)}, x \geq 0,$ $s > 0, a \geq 0$	gamma	<code>shape = a,</code> <code>scale = s</code>
Пуассоновское	$\frac{\lambda^k}{k!} e^{-\lambda}$ $\lambda > 0, k = 0, 1, \dots$	pois	<code>lambda = lambda</code>	Коши	$s (\pi(s^2 + (x - l)))^{-1}$	cauchy	<code>location = l,</code> <code>scale = s</code>
Геометрическое	$p(1-p)^k$ $0 < p \leq 1, k = 0, 1, \dots$	geom	<code>prob = prob</code>	Стьюдента t	$\frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})} \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}$ $n > 0$	t	<code>df = n</code>
Отрицательно- биномиальное	$\frac{\Gamma(k+n)}{\Gamma(n)k!} p^n (1-p)^k$ $0 < p \leq 1, n = 1, 2, \dots,$ $k = 0, 1, \dots$	nbinom	<code>size = size,</code> <code>prob = prob</code>	F	$\frac{\Gamma(\frac{1}{2}(n_1+n_2))}{\Gamma(\frac{1}{2}n_1)\Gamma(\frac{1}{2}n_2)} \left(\frac{n_1}{n_2}\right)^{\frac{1}{2}n_1} x^{\frac{1}{2}n_1-1} \times$ $\times \left(1 + \frac{n_1}{n_2}x\right)^{-\frac{1}{2}(n_1+n_2)}$ $n_1 > 0, n_2 > 0$	f	<code>df1 = n_1,</code> <code>df2 = n_2</code>
Гипергео- метрическое	$\frac{C_m^k C_{n-r}^{r-k}}{C_{m+n}^r}$ $m, n = 1, 2, \dots,$ $r = 1, 2, \dots, m+n,$ $k = 0, 1, \dots, m$	hyper	<code>m = m,</code> <code>n = n,</code> <code>r = r</code>				
Равномерное	$(b-a)^{-1}, a < x < b$ $a < b$	unif	<code>min = a,</code> <code>max = max</code>				
Экспоненциальное	$\lambda e^{-\lambda x}, x \geq 0$ $\lambda > 0$	exp	<code>lambda = lambda</code>				
Нормальное	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	norm	<code>mean = mean,</code> <code>sd = sd</code>				

Генерация данных

Приставка **r** означает генерирование выборки значений, обязательный аргумент — объем выборки.
Приставка **d** означает вычисление плотности распределения для непрерывных распределений или вероятности значения для дискретных распределений. Обязательный аргумент — точка или массив точек, в которых вычисляется плотность. Приставка **p** соответствует вычислению функции распределения. Обязательный аргумент — точка или массив точек, в которых вычисляется функция распределения.
Приставка **q** означает вычисление квантиля распределения. Обязательный аргумент — вероятность или массив вероятностей, для которых вычисляются квантили.

Пример:

```
n=rnorm(10)
```

```
> n
```

```
[1] -0.5673446 -1.1319980 -0.4145146 1.6568926 1.8034000 -0.3728487 -0.1046350 1.5705039
```

```
[9] -0.1435875 -1.3417850
```

Вопросы по теме

1. Почему R наиболее распространённый язык для разработки программ анализа данных?
2. Охарактеризуйте объекты в R?
3. Какие функции вы знаете для работы с объектами?
4. Как ввести данные в R?
5. Как генерировать данные в R?
6. Что такое пакеты и как с ними работать?
7. Как реализовать ввод и вывод в R?

Литература



Роберт И. Кабаков R в действии. Анализ и визуализация данных в программе R /
пер. с англ. Полины А. Волковой. – М.: ДМК Пресс, 2014. – 588 с.: ил. ISBN 978-5-
947060-077-1