***Основы работы в среде R***

*R* — это научная некоммерческая программная среда[[1]](#footnote-1), развиваемая в рамках международного проекта GNU которая включает язык программирования для автоматизированной обработки данных и большой набор разнообразных функций для анализа данных, включая графические иллюстрации. Среда *R* применяется не только для первичного анализа, но также для математического моделирования высокого уровня. Отметим, что среда *R* может быть также использована в сочетании с другими программными средами. На базе *R* разработано большое количество многофункциональных пакетов, которые могут быть использованы для специализированного анализа данных.

В августе 1993 г. двое новозеландских учёных анонсировали свою новую разработку, которую они назвали *R*. По замыслу создателей (Robert Gentleman и Ross Ihaka), это должна была быть новая реализация языка *S*, отличающаяся от *S* некоторыми деталями, например, обращением с глобальными и локальными переменными, а также работой с памятью. Фактически, они создали не полный аналог *S*, а новое «ответвление» на «дереве *S*». Сначала проект развивался довольно медленно, но когда появилась платформа на базе интернета для написания дополнений и специализированных пакетов, всё большее количество людей стало переходить с *S* на *R*. Количество книг, написанных про *R* за последние годы, выросло в несколько раз, а количество пакетов уже приближается к полутора тысячам.

Идея центральной системы хранения и распространения пакетов CRAN известного как Comprehensive R Archive Network (http://cran.r-project.org/) была заимствована из TEX-сообщества (CTAN, или Comprehensive TeX Archive Network; аналогичной схемой пользуется и Perl-сообщество: CPAN или Comprehensive Perl Archive Network). Все три упомянутых проекта объединяет одно: стабильная база и множество дополнений.

Установка и обновление среды R

Для установки среды R необходимо:

1. Скачать установщик с сайта *http://cran.r-project.org/*

2) Выполнить установку согласно данным инструкциям.

В среде R отдельную команду можно набрать вручную, либо подготовить при помощи блокнота последовательность команд, а затем вставить эту последовательность в качестве функции:

> pr1<-function(n){

+ print("Hello World!")

+ }

> pr1(1)

[1] "Hello World!"

>

Любая строка в коде, начинающаяся с символа *#* означает комментарий. Узнать все доступные в текущем окружении имена можно с помощью функции *ls().* Удалить некоторые имена *x* и *x0* можно с помощью функции *rm(x, x0),* полностью очистить текущее окружение можно следующим способом *rm(list= ls()).* Выход из среды R осуществляется при помощи команды *q().*

Установка и обновление дополнительных пакетов

Среда R предлагает пользователям большой набор пакетов различных типов, позволяющих строить графики или выполнять классификационный либо регрессионный анализ. Их полный список доступен на сайте: [*http://cran.r-project.org/*](http://cran.r-project.org/)

Установку пакетов можно производить как вручную, загрузив файл с сайта, так и из консоли, выполнив определённые команды.

Для установки пакета - *install.packages(‘имя пакета’).*

Для подключения пакета к рабочей среде - *require(‘имя пакета’)* либо *library(‘имя пакета’).*

Например, следующая последовательность команд устанавливает пакет “*install*”, а затем обновляет версию R.

install.packages(installr)

require(installr)

updateR()

Установка Rcmdr

install.packages("Rcmdr", dependencies=TRUE)

где включение опции dependencies вызовет гарантированную установку полного комплекта остальных пакетов, которые могут потребоваться при обработке данных через меню Rcmdr.

### Практическое задание №1. Работа с пакетом R Commander

##### Работа с меню пакета R Commander

Удобным средством освоения вычислений в R для начинающего пользователя является R Commander – платформо-независимый графический интерфейс в стиле кнопочного меню, реализованный в пакете Rcmdr. Он позволяет осуществить большой комплект процедур статистического анализа, не прибегая к предварительному заучиванию функций на командном языке, однако невольно способствует этому, поскольку отображает все выполняемые инструкции в специальном окне.

Установить Rcmdr, как и любые другие расширения, можно из меню консоли R "Пакеты > Установить пакет", но лучше выполнив команду:

install.packages("Rcmdr", dependencies=TRUE)

где включение опции dependencies вызовет гарантированную установку полного комплекта остальных пакетов, которые могут потребоваться при обработке данных через меню Rcmdr.

Запуск R Commander происходит при загрузке пакета Rcmdr через меню "Пакеты >Включить пакет" или командой

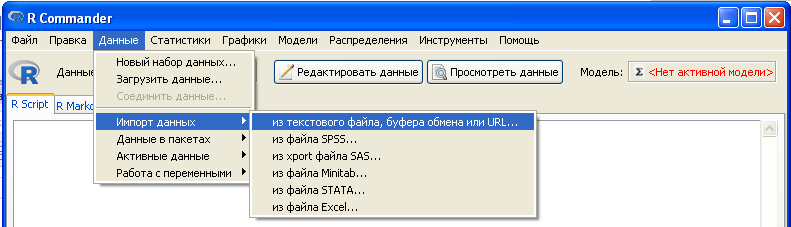
library(Rcmdr)

Работу в R Commander рассмотрим на примере корреляционного анализа данных по уровню зараженности двустворчатого моллюска *Dreissena polymorpha* инфузорией *Conchophthirus acuminatus* в трех озерах.

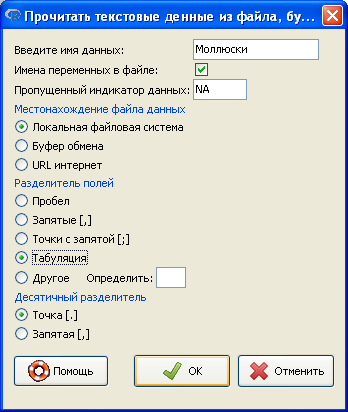
Таблица с исходными данными находится в файле Dreissena.txt,

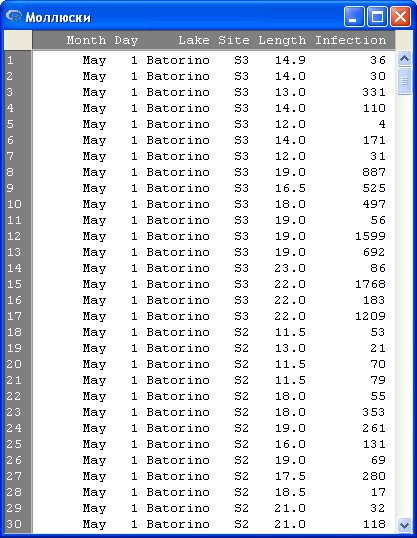
нас будут интересовать две переменные: длина раковины моллюска (ZMlength, мм) и число обнаруженных в моллюске инфузорий (CAnumber).

Первый этап – загрузка нового набора данных, и мы выбираем из меню "Импорт данных из текстового файла":

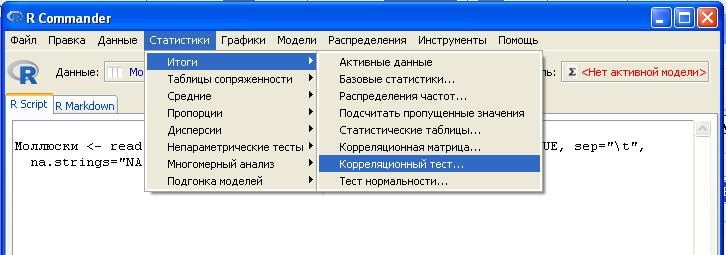


Чтобы убедиться в том, что наши данные загружены верно (или при необходимости их отредактировать), нажимаем кнопку "Посмотреть данные".

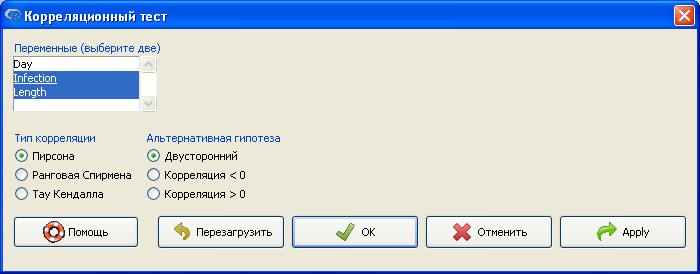
*Окно определения организации данных*

*Фрагмент загруженной таблицы*

На втором этапе в меню "Статистика" выбираем "Корреляционный тест":



Выбираем пару коррелируемых переменных и в Окне вывода получаем коэффициент корреляции Пирсона (R = 0.467), уровень достигнутой статистической значимости (p-value < 2.2e-16) и 95%-ные доверительные пределы.



*Окно определения режима расчетов*

Pearson's product-moment correlation

data: Infection and Length

t = 11.496, df = 474, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.3935877 0.5343949

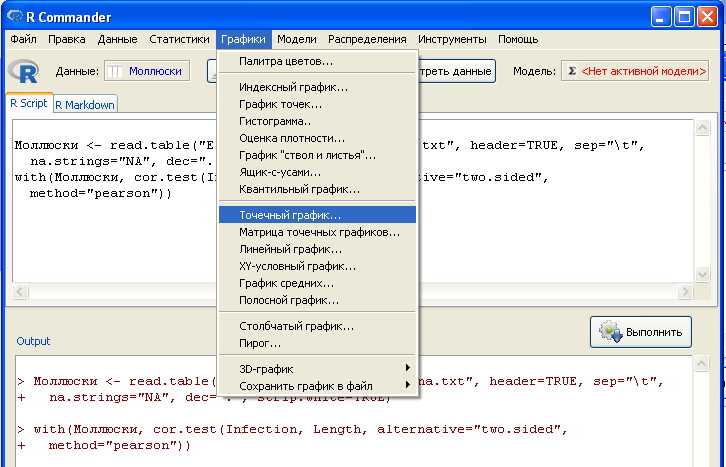
sample estimates:

cor

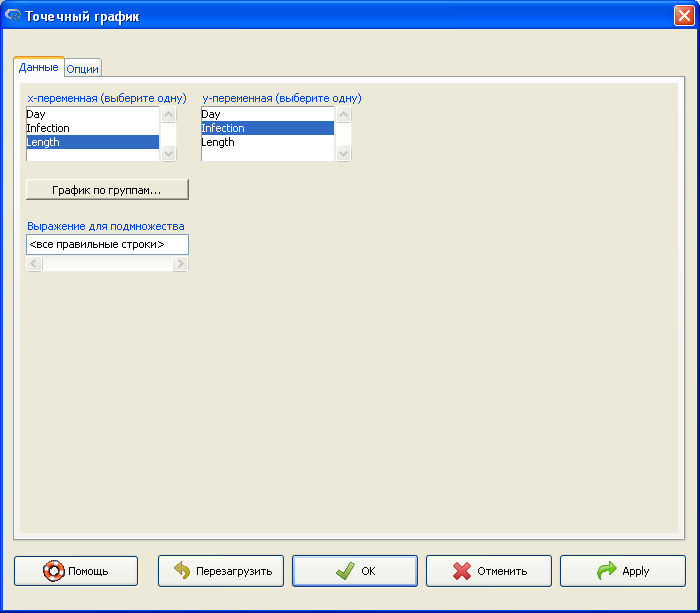
0.466946

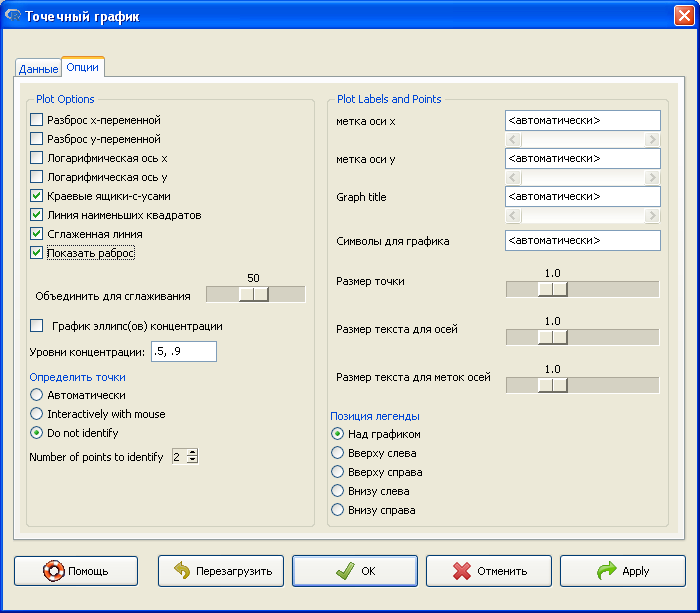
*Полученные результаты*

Полученные результаты легко скопировать из окна вывода через буфер обмена. Теперь получим графическое изображение корреляционной зависимости. Выберем точечный график (scatterplot) зависимости CAnumber от ZMlength и снабдим его краевыми диаграммами размахов, линией линейного тренда по методу наименьших квадратов (зеленым цветом), линией, сглаженной по методу локальной регрессии (красным цветом), представленной с доверительной областью (пунктир). Для каждого из трех озер (переменная Lake) экспериментальные точки будут представлены разными символами.



*Выбор типа графика*





*Режимы графических опций*



*График, скопированный из графического окна R Commander*

Как эквивалент всем нажатиям кнопок меню R Commander, в окне скриптов появляются инструкции языка R. В нашем случае они имеют следующий вид:

> Моллюски <- read.table("E:/Магистры/ПР№1/Dreissena.txt", header=TRUE, sep="\t", + na.strings="NA", dec=".", strip.white=TRUE)

> with(Моллюски, cor.test(Infection, Length, alternative="two.sided", + method="pearson"))

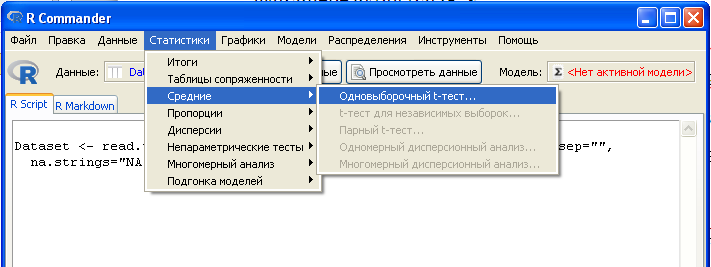
> scatterplot(Infection~Length, reg.line=lm, smooth=TRUE, spread=TRUE, + boxplots='xy', span=0.5, ellipse=FALSE, levels=c(.5, .9), data=Моллюски)

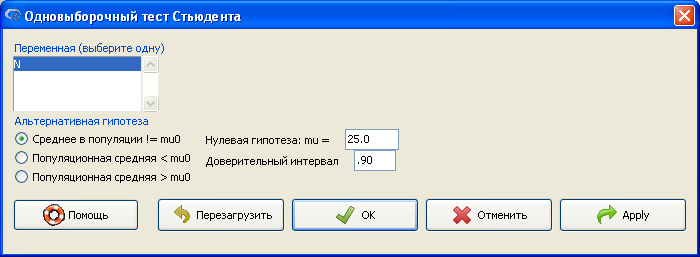
Сам скрипт или выводимые результаты (а также и то, и другое вместе) можно сохранить в файлах и в любой момент повторить. Тот же самый результат можно получить без запуска R Commander, загрузив сохраненный файл через консоль R.

По большому счету, не зная конструкций языка R (или просто не желая отягощать свою память их запоминанием), с использованием Rcmdr можно выполнить обработку данных с использованием почти всех базовых статистических методов. Здесь представлены параметрические и непараметрические тесты, методы подгонки различных непрерывных и дискретных распределений, анализ многомерных таблиц сопряженности, одномерный и многомерный дисперсионный анализ, метод главных компонент и кластеризация, различные формы обобщенных регрессионных моделей и др. Достоин тщательного изучения развитый аппарат анализа и тестирования полученных моделей.

Рассмотрим пример обработки одномерных данных.

Аналогичным образом необходимо загрузить данные файла nybirths.dat. Выполним одновыборочный t-тест.





Результат работы:

One Sample t-test

data: N

t = 0.33153, df = 167, p-value = 0.7407

alternative hypothesis: true mean is not equal to 25

90 percent confidence interval:

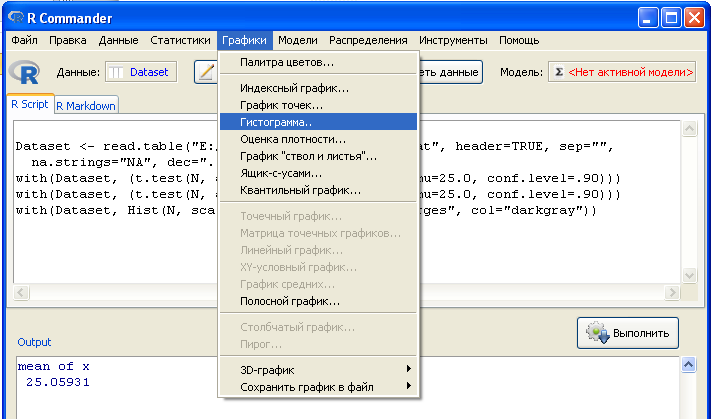
24.76341 25.35521

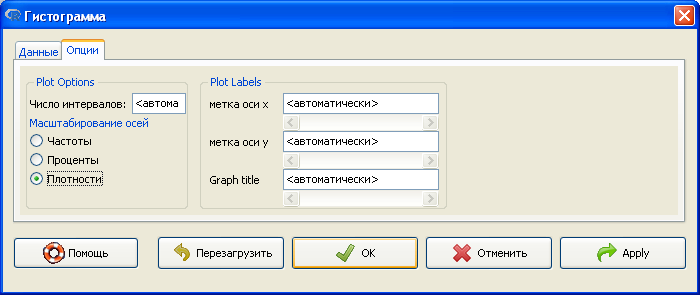
sample estimates:

mean of x

25.05931

Построим гистограмму для набора данных.



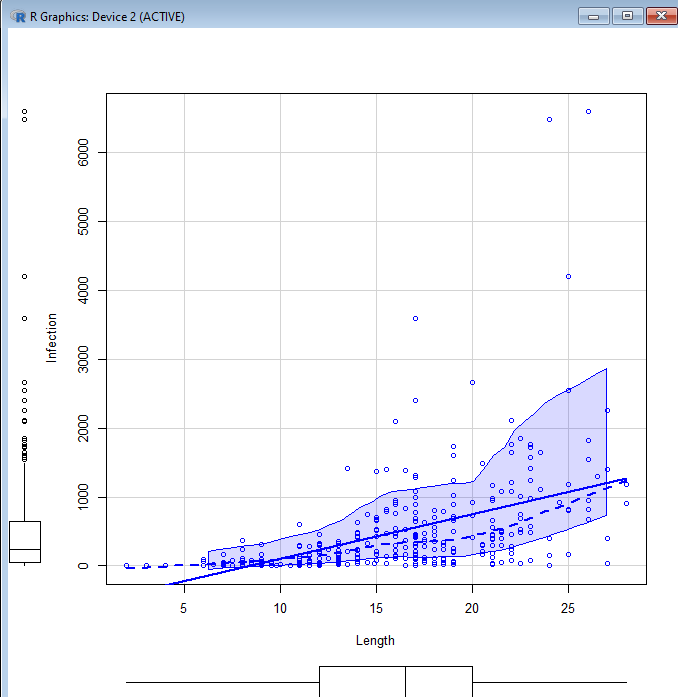
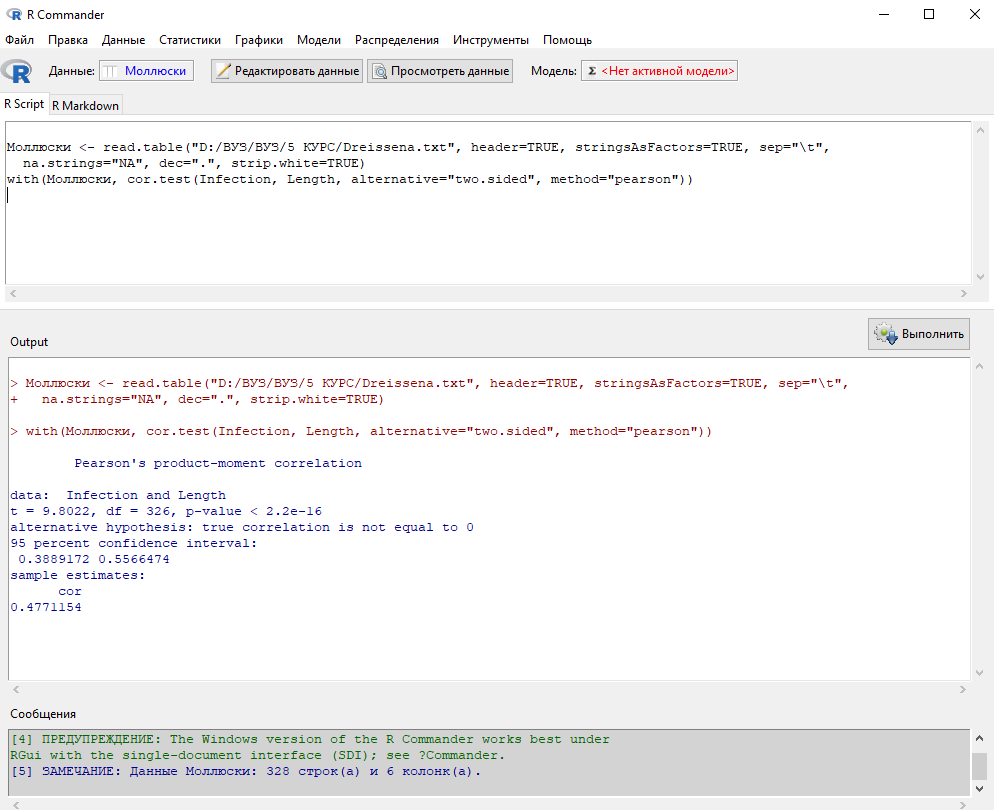


Результат:

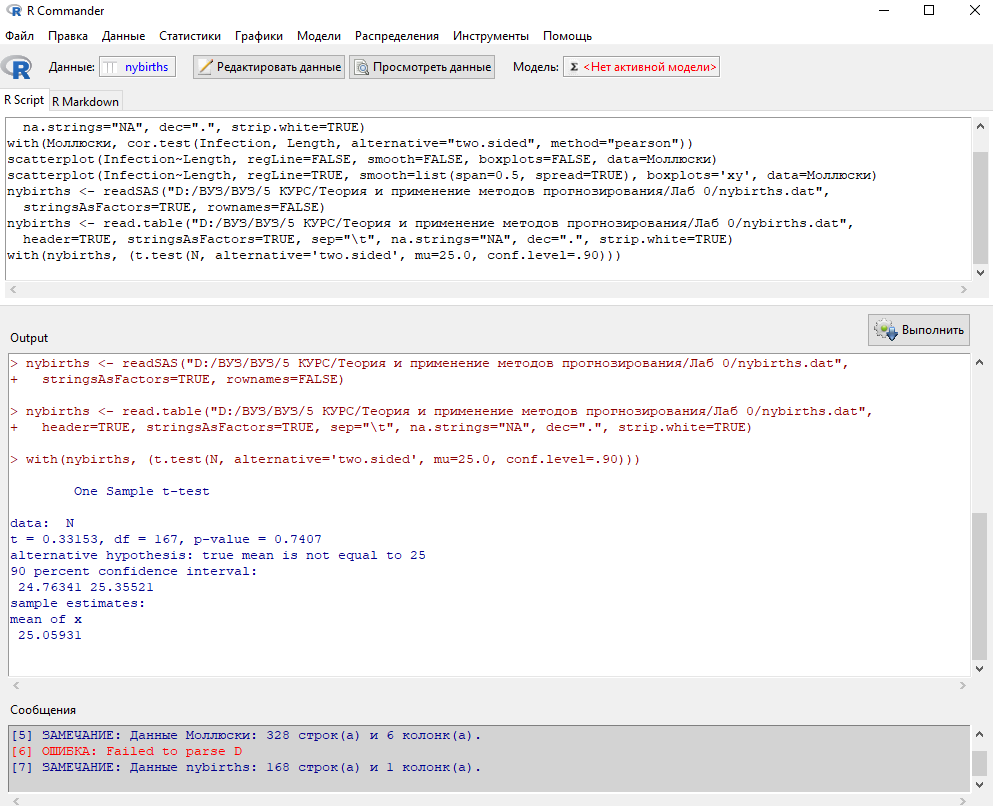


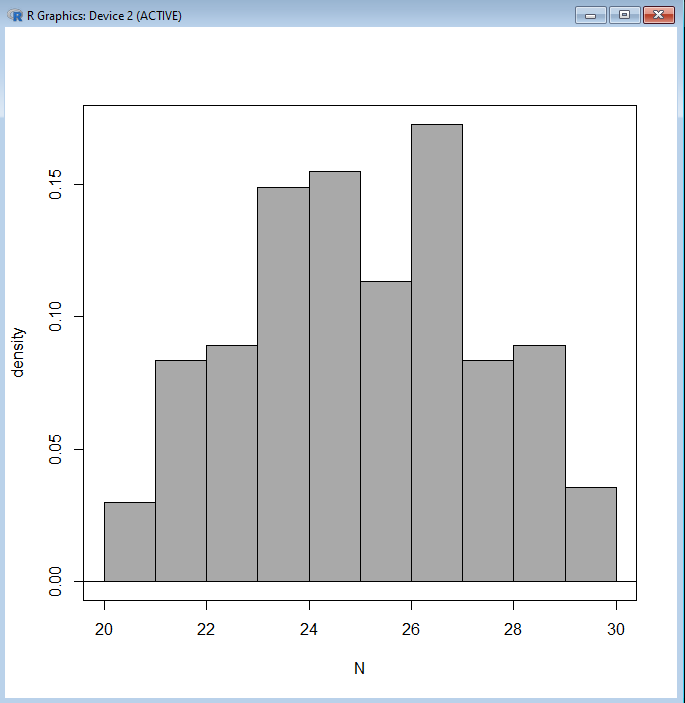
##### Задание

1. Выполнить описанные выше действия с тестовым набором данных Dreissena.txt



1. Выполнить описанные выше действия с тестовым набором данных nybirths.dat.



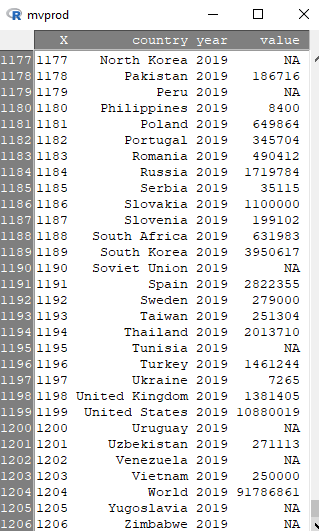


1. Изучить основные функции пакета R Commander и определить, какие из них могут быть полезны для решения реальной прикладной задачи.

|  |
| --- |
| which.max(x) — индекс элемента с максимальным значением  which.min(x) — индекс элемента с минимальным значением  rev(x) — реверсирует порядок элементов  sort(x) — сортирует элементы объекта по возрастанию  cut(x,breaks) — делит вектор на равные интервалы  match(x, y) — ищет элементы x, которые есть в y  which(x == a) — возвращает порядковые элементы x, которые равны a  na.omit(x) — исключает отсутствующие значения объекта  na.fail(x) — бросает исключение, если объект содержит отсутствующие значения  unique(x) — исключает из объекта повторяющиеся элементы  table(x) — создаёт таблицу с количеством повторений каждого уникального элемента  subset(x, …) — возвращает подмножество элемента, которое соответствует заданному условию  sample(x, size) — возвращает случайный набор размера size из элементов x  replace(x, list, values) — заменяет значения x c индексами из list значениями из values  append(x, values) — добавляет элементы values в вектор x  plot(x) — график x  plot(x, y) — график зависимости y от x  hist(x) — гистограмма  barplot(x) — столбчатая диаграмма  dotchart(x) — диаграмма Кливленда  pie(x) — круговая диаграмма  boxplot(x) — график типа “коробочки с усами”  sunflowerplot(x, y) — то же, что и plot(), однако точки с одинаковыми координатами изображаются в виде “ромашек”, количество лепестков у которых пропорционально количеству таких точек  coplot(x˜y | z) — график зависимости y от x для каждого интервала значений z  interaction.plot(f1, f2, y) — если f1 и f2 — факторы, эта фукнция создаст график со средними значениями y в соответствии со значениями f1 (по оси х) и f2 (по оси у, разные кривые)  matplot(x, y) — график зависимости столбцов y от столбцов x  fourfoldplot(x) — изображает (в виде частей окружности) связь между двумя бинарными переменными в разных совокупностях  assocplot(x) — график Кохена-Френдли  mosaicplot(x) — мозаичный график остатков лог-линейной регрессии  pairs(x) — если х - матрица или таблица данных, эта функция изобразит диаграммы рассеяния для всех возможных пар переменных из х  plot.ts(x), ts.plot(x) — изображает временной ряд  qqnorm(x) — квантили  qqplot(x, y) — график зависимости квантилей y от квантилей х  contour(x, y, z) — выполняет интерполяцию данных и создает контурный график  filled.contour(x, y, z) — то же, что contour(), но заполняет области между контурами определёнными цветами  image(x, y, z) — изображает исходные данные в виде квадратов, цвет которых определяется значениями х и у  persp(x, y, z) — то же, что и image(), но в виде трехмерного графика  stars(x) — если x — матрица или таблица данных, изображает график в виде “звезд” так, что каждая строка представлена “звездой”, а столбцы задают длину сегментов этих “звезд”  symbols(x, y, …) — изображает различные символы в соответствии с координатами  termplot(mod.obj) — изображает частные эффекты переменных из регрессионной модели  points(x, y) — рисование точек  lines(x, y) — рисование линии  text(x, y, labels, …) — добавление текстовой надписи  mtext(text, side=3, line=0, …) — добавление текстовой надписи  segments(x0, y0, x1, y1) — рисование отрезка  arrows(x0, y0, x1, y1, angle= 30, code=2) — рисование стрелочки  abline(a,b) — рисование наклонной прямой  abline(h=y) — рисование вертикальной прямой  abline(v=x) — рисование горизонтальной прямой  abline(lm.obj) — рисование регрессионной прямой  rect(x1, y1, x2, y2) — рисование прямоугольника  polygon(x, y) — рисование многоугольника  legend(x, y, legend) — добавление легенды  title() — добавление заголовка  axis(side, vect) — добавление осей  rug(x) — рисование засечек на оси X  locator(n, type = “n”, …) — возвращает координаты на графике, в которые кликнул пользователь  sd(x) — стандартное отклонение  var(x) — дисперсия  cor(x) — корреляционная матрица  var(x, y) — ковариация между x и y  cor(x, y) — линейная корреляция между x и y  aov(formula) — дисперсионный анализ  anova(fit,…) — дисперсионный анализ для подогнанных моделей fit  density(x) — ядерные плотности вероятностей  binom.test() — точный тест простой гипотезы о вероятности успеха в испытаниях Бернулли  pairwise.t.test() — попарные сравнения нескольки независимых или зависимых выборок  prop.test() — проверка гипотезы о том, что частоты какого-либо признака равны во всех анализируемых группах  t.test() — тест Стьюдента  rnorm(n, mean=0, sd=1) — нормальное распределение  rexp(n, rate=1) — экспоненциальное распределение  rgamma(n, shape, scale=1) — гамма-распределение  rpois(n, lambda) — распределение Пуассона  rweibull(n, shape, scale=1) — распределение Вейбулла  rcauchy(n, location=0, scale=1) — распределение Коши  rbeta(n, shape1, shape2) — бета-распределение  rt(n, df) — распределение Стьюдента  rf(n, df1, df2) — распределение Фишера  rchisq(n, df) — распределение Пирсона  rbinom(n, size, prob) — биномиальное распределение  rgeom(n, prob) — геометрическое распределение  rhyper(nn, m, n, k) — гипергеометрическое распределение  rlogis(n, location=0, scale=1) — логистическое распределение  rlnorm(n, meanlog=0, sdlog=1) — логнормальное распределение  rnbinom(n, size, prob) — отрицательное биномиальное распределение  runif(n, min=0, max=1) — равномерное распределение  sin(x), cos(x), tan(x), asin(x), acos(x), atan(x), atan2(y, x), log(x), log(x, base), log10(x), exp(x) — элементарные математические функции  min(x), max(x) — минимальный и максимальный элементы объекта  range(x) — вектор из минимального и максимального элемента объекта  pmin(x, y), pmax(x, y) — возвращают вектор с минимальными (максимальными) для каждой пары x[i], y[i]  sum(x) — сумма элементов объекта  prod(x) — произведение элементов объекта  diff(x) — возвращает вектор из разниц между соседними элементами  mean(x) — среднее арифметическое элементов объекта  median(x) — медиана (средний элемент) объекта  weighted.mean(x, w) — средневзвешенное объекта x (w определяет веса)  round(x, n) — округляет x до n знаков после запятой  cumsum(x), cumprod(x), cummin(x), cummax(x) — кумулятивные суммы, произведения, минимумы и максимумы вектора x (i-ый элемент содержит статистику по элементам x[1:i])  union(x, y), intersect(x, y), setdiff(x,y), setequal(x,y), is.element(el,set) — операции над множествами: объединение, пересечение, разность, сравнение, принадлежность  Re(x), Im(x), Mod(x), Arg(x), Conj(x) — операции над комплексными числами: целая часть, мнимая часть, модуль, аргумент, сопряжённое число  fft(x), mvfft(x) — быстрое преобразование Фурье  choose(n, k) — количество сочетаний  rank(x) — ранжирует элементы объекта |

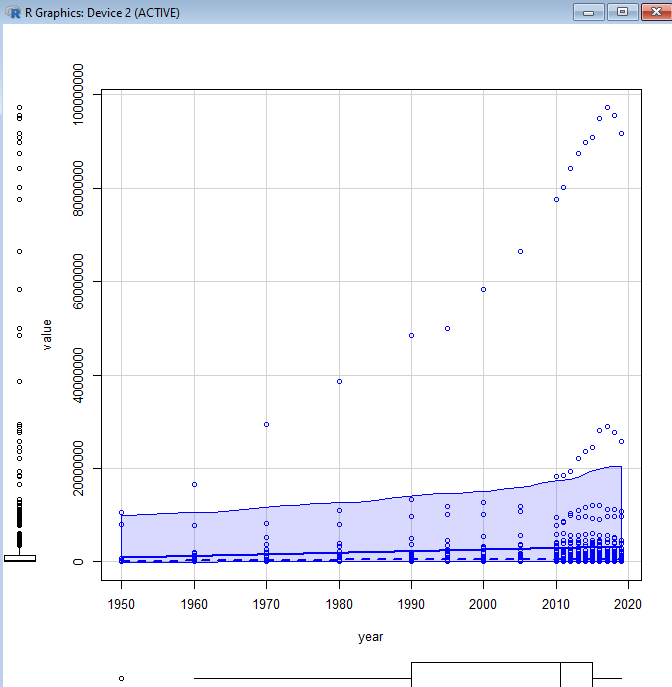
1. Подготовить фрагмент данных из реальной задачи.

Производство автомобилей по странам, 1950–2019 гг.



1. Выполнить базовые функции для выбранного фрагмента данных.

Увеличение производства автомобилей со временем:



##### Содержание отчёта

1. Фрагмент данных, использованных для выполнения работы. Описание реальной задачи, которую планируется решить.

2. «Скриншоты» полученных результатов.

3. Выводы о проделанной работе. В выводах обязательно должен присутствовать ответ на вопрос: «Какие функции пакета R Commander можно применить к своей задаче»

1. http://cran.r-project.org [↑](#footnote-ref-1)