

**Министерство образования и науки Российской Федерации  
ФГАОУ ВО «Севастопольский государственный университет»**

**Институт информационных технологий  
и управления в технических системах**

**Лабораторная работа №2  
«Корреляционный и регрессионный анализ данных»**

по дисциплине «Интеллектуальный анализ данных»  
для студентов всех форм обучения направления подготовки  
09.03.02 «Информационные системы и технологии»



Севастополь  
**2017**

**Корреляционный и регрессионный анализ данных.** Методические указания к лабораторным занятиям по дисциплине «Интеллектуальный анализ данных» / Сост.: И.В. Дымченко, И.П. Шумейко, О.А. Сырых – Севастополь: Изд-во СевГУ, 2017 – 22 с.

Методические указания предназначены для проведения лабораторных работ по дисциплине «Интеллектуальный анализ данных». Целью методических указаний является помощь студентам в изучении возможностей системы RStudio. Излагаются практические сведения необходимые для выполнения лабораторной работы, требования к содержанию отчета.

Методические указания рассмотрены и утверждены на методическом семинаре и заседании кафедры «Информационные системы» (протокол № 1 от 29 августа 2016 г.)

## Лабораторная работа №2.2

### Корреляционный и регрессионный анализ данных. Работа с диаграммами

#### Цель:

- исследовать возможности языка R для проведения корреляционного и регрессионного анализа данных;
- исследовать возможности языка R для создания и изменения вида диаграмм

**Время:** 2 часа

**Лабораторное оборудование:** персональные компьютеры, выход в сеть Internet, RStudio.

### Краткие теоретические сведения

#### Работа с диаграммами

R – это изумительная программа для построения диаграмм. В стандартной интерактивной сессии создается диаграмма, вводом по одной команде и добавлением элементов диаграммы, пока не получится то, что необходимо.

#### Пример.

Рассмотрим следующие пять строк:

```
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
detach(mtcars)
```

Первая команда добавляет в траекторию поиска таблицу данных `mtcars` (встроенный набор данных). Вторая команда открывает окно графики и создает диаграмму рассеяния, на которой вес автомобиля отложен на горизонтальной оси, а расход топлива – на вертикальной. Третья команда добавляет регрессионную прямую. Четвертая команда добавляет название. Последняя команда удаляет таблицу данных из пути поиска. В R диаграммы обычно создаются в таком интерактивном стиле (см. рис.1).

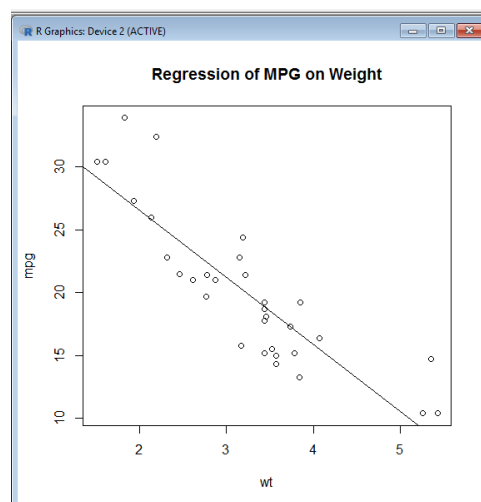


Рисунок 1.

Диаграммы можно сохранять при помощи программного кода или меню графического пользовательского интерфейса. Для сохранения диаграммы при помощи кода необходимо разместить создающие диаграмму команды между командами, которые назначают место вывода и закрывают вывод. Например, следующий программный код позволяет сохранить диаграмму в формате PDF под названием `mygraph.pdf` в текущей рабочей директории:

```
pdf("mygraph.pdf")
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
detach(mtcars)
dev.off()
```

В дополнение к pdf() можно использовать функции win.metafile(), png(), jpeg(), bmp(), tiff(), xfig() и postscript(), чтобы сохранять диаграммы в других форматах. Необходимо учитывать, что формат Windows metafile доступен только под Windows.

Способ сохранения диаграмм при помощи графического пользовательского интерфейса различается в зависимости от оперативной системы. Под Windows при активированном графическом устройстве нужно выбрать в меню Файл → Сохранить как, а затем в появившемся диалоге выбрать нужный формат графического файла и директорию для сохранения.

Новая диаграмма, которая создается при помощи команды высокого уровня, такой как plot(), hist() или boxplot(), обычно заменяет предыдущую диаграмму. Для создания более одной диаграммы есть несколько способов:

1. Открыть новое графическое устройство, перед тем как создавать новую диаграмму  
dev.new()  
команды для построения диаграммы 1  
dev.new()  
команды для построения диаграммы 2  
и т.д.

Каждая новая диаграмма будет появляться в последнем открытом окне.

2. Можно получить доступ к нескольким диаграммам сразу через пользовательский интерфейс. Под Windows эта операция состоит из двух этапов. После того как открыто первое окно графики, необходимо выбрать в меню История команд → Запись. Затем использовать пункты меню Предыдущий и Следующий для перемещения между созданными диаграммами.

3. Можно использовать функции dev.new(), dev.next(), dev.prev(), dev.set() и dev.off() для одновременного открытия нескольких окон графики и выбора необходимой диаграммы.

### Пример.

В табл. 1. Представлен набор данных, который описывает реакцию пациента на два лекарства в пяти дозировках.

Таблица 1. Реакция пациента на два лекарства в пяти дозировках

Дозировка	Реакция на лекарство А	Реакция на лекарство В
20	16	15
30	20	18
40	27	25
45	40	31
60	60	40

Эти данные можно ввести при помощи следующего программного кода:

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
```

Простой линейный график, изображающий зависимость реакции пациента от дозы лекарства А, можно создать так:

```
plot(dose, drugA, type="b")
```

`plot()` – это функция общего назначения, которая строит диаграммы в R. В этом случае `plot(x, y, type="b")` располагает  $x$  на горизонтальной оси, а  $y$  – на вертикальной, изображает точки с координатами  $(x, y)$  и соединяет их линиями.

Параметр `type="b"` означает, что на графике должны быть показаны и точки, и линии. Получившийся график показан на рис. 2.

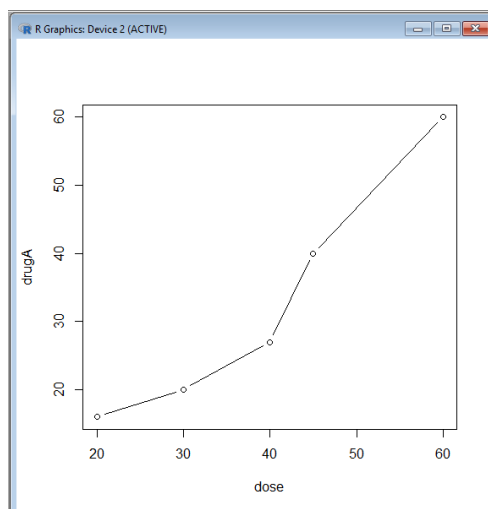


Рисунок 2. График зависимости реакции пациента от дозы лекарства A

### Графические параметры

Многие характеристики диаграмм (шрифты, цвета, оси, названия) можно изменять при помощи опций, которые называются «графические параметры».

Один способ назначить эти параметры – использовать функцию `par()`. Значения параметров, заданные таким способом, будут действовать на протяжении всей сессии, пока вы не измените их. Формат применения функции таков: `par(название параметра=значение, название параметра=значение, ...)`. Функция `par()` без аргументов выводит на экран действующие значения графических параметров. Добавление аргумента `no.readonly=TRUE` позволяет увидеть только те графические параметры, которые можно изменять.

Например для обозначения отдельных пациентов нужно использовать заполненный треугольник вместо пустого кружка и соединить символы пунктирной линией, а не сплошной. Это можно сделать при помощи следующего программного кода:

```
opar <- par(no.readonly=TRUE)
par(lty=2, pch=17)
plot(dose, drugA, type="b")
par(opar)
```

Получившийся график показан на рис. 3.

Первая команда создает копию текущих параметров. Вторая команда назначает тип линии – пунктирная (`lty=2`) вместо сплошной по умолчанию и тип символа – заполненный треугольник (`pch=17`). Затем создается график и восстанавливаются исходные значения параметров.

Можно использовать столько функций `par()`, сколько нужно, так что команда может быть также записана в виде

```
par(lty=2)
par(pch=17)
```

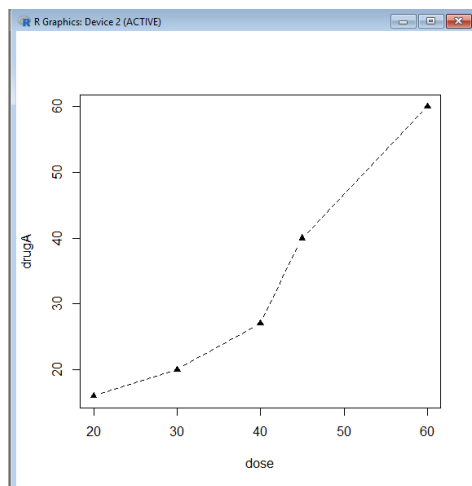


Рисунок 3. График зависимости реакции пациента от дозы лекарства А с измененными типом линии и символами.

Второй способ задать графические параметры – это включить записи типа название параметра=значение внутрь графической функции высокого уровня. В этом случае заданные параметры будут действовать только для конкретной диаграммы. Можно было бы построить тот же график при помощи следующего программного кода:

```
plot(dose, drugA, type="b", lty=2, pch=17)
```

Не во всех графических функциях высокого уровня можно изменять все возможные графические параметры.

**Познакомьтесь со справкой по каждой функции для построения диаграмм (например: ?plot, ?hist или ?boxplot), чтобы узнать, какие графические параметры можно назначать таким образом.**

### Символы и линии

Графические параметры можно использовать для того, чтобы указывать тип символов и линий на диаграммах. Соответствующие параметры перечислены в табл. 2.

Таблица 2. Параметры для указания типов символов и линий

Параметр	Описание
pch	Определяет тип символа (см. рис.4)
cex	Определяет размер символа. cex – это число, обозначающее, как символы должны быть масштабированы по отношению к размеру по умолчанию. 1 = размер по умолчанию, 1.5 – на 50% крупнее, 0.5 – на 50% мельче и т. д.
lty	Определяет тип линии (см. рис. 5)
lwd	Определяет толщину линии по сравнению с толщиной линии по умолчанию (1). Например, lwd=2 делает линию в два раза толще, чем по умолчанию

Параметр pch= определяет тип символов, которые используются на диаграмме. Возможные значения приведены на рис. 4.

Для символов с 21 по 25 можно отдельно указывать цвет контура (border=) и заполнения (bg=).

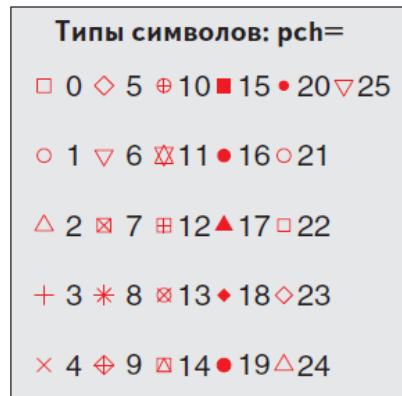


Рисунок 4. Символы, назначаемые при помощи параметра pch.

Параметр lty= применяется для обозначения нужного типа линии. Значения параметра показаны на рис. 5.

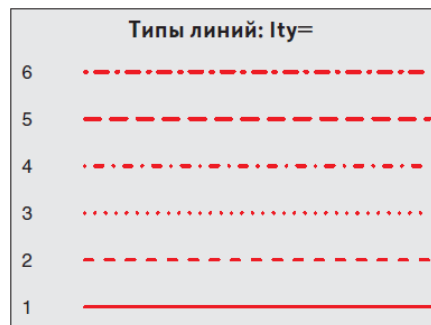


Рисунок 5. Типы линий, назначаемые при помощи параметра lty.

Программный код, объединяющий все эти параметры, `plot(dose, drugA, type="b", lty=3, lwd=3, pch=15, cex=2)` создаст график, на котором точечная линия в три раза шире, чем по умолчанию, соединяет наблюдения, представленные в виде заполненных квадратов в два раза большего размера, чем по умолчанию.

Результат представлен на рис. 6.

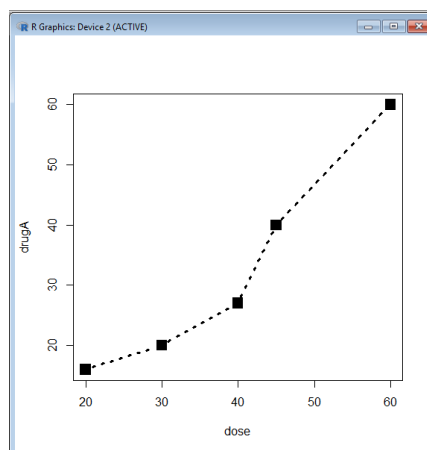


Рисунок 6. График зависимости реакции пациента от дозы лекарства A с измененными типом и шириной линии, а также типом и размером символов.

### Цвета

В R есть несколько связанных с цветами параметров. В табл. 3 приведены некоторые из самых распространенных.

Таблица 3. Параметры для назначения цвета

Параметр	Описание
col	Цвет элементов на графике. Для некоторых функций (таких как <code>lines</code> и <code>pie</code> ) можно указывать вектор из значений, которые используются по очереди. Например, если <code>col=c("red", "blue")</code> и изображены три линии, первая будет красной, вторая – синей и третья – красной.
col.axis	Цвет значений осей
col.lab	Цвет подписей осей
col.main	Цвет заголовков
col.sub	Цвет подзаголовков
fg	Цвет графика
bg	Цвет фона

В R цвета можно обозначать номером, названием, в шестнадцатеричной системе, а также в системах RGB или HSV. Например, `col=1`, `col="white"`, `col="#FFFFFF"`, `col=rgb(1,1,1)` и `col=hsv(0,0,1)` – взаимозаменяемые способы обозначить белый цвет. Функция `rgb()` определяет цвета по значениям красного, зеленого и синего, а `hsv()` основана на значениях оттенка и насыщенности.

Функция `colors()` выводит на экран список всех доступных цветов.

В R также реализован ряд функций, которые позволяют создавать векторы из близких цветов. К таким функциям относятся `rainbow()`, `heat.colors()`, `terrain.colors()`, `topo.colors()` и `cm.colors()`. Например, `rainbow(10)` создает 10 соседних "радужных" цветов. Оттенки серого создаются функцией `gray()`.

В этом случае задаются оттенки серого в виде вектора чисел от 0 до 1. Команда `gray(0:10/10)` создаст 10 оттенков серого.

Пример: чтобы увидеть, как это работает можно запустить программный код

```
n <- 10
mycolors <- rainbow(n)
pie(rep(1, n), labels=mycolors, col=mycolors)
mygrays <- gray(0:n/n)
pie(rep(1, n), labels=mygrays, col=mygrays)
```

### Характеристики текста

Графические параметры также используются для определения размера, шрифта и стиля текста. Параметры, определяющие размер шрифта, приведены в табл. 4. Параметры, при помощи которых можно указать тип шрифта, перечислены в табл. 5.

Таблица 4. Параметры, определяющие размер шрифта

Параметр	Описание
cex	Число, определяющее, как отображаемый на диаграмме текст будет масштабирован относительно размера по умолчанию (1). 1.5 – на 50% больше, 0.5 – на 50% меньше и т. д.
cex.axis	Размер значений на осях по отношению к cex
cax.lab	Размер названий осей по отношению к cex
cex.main	Размер заголовков по отношению к cex
cex.sub	Размер подзаголовков по отношению к cex

Например, на всех диаграммах, созданных после команды

```
par(font.lab=3, cex.lab=1.5, font.main=4, cex.main=2)
```



в 1.5 раза более крупные, чем по умолчанию, подписи осей будут выделены курсивом, а названия будут в два раза крупнее, чем по умолчанию, и еще выделены полужирным курсивом.

Таблица 5. Параметры, определяющие семейство, размер и стиль шрифта.

Параметр	Описание
font	Число, которое определяет шрифт для текста на диаграмме. 1 = обычный, 2 = полужирный, 3 = курсив, 4 = полужирный курсив, 5 = символы (в кодировке Adobe)
font.axis	Шрифт значений на осях
font.lab	Шрифт для подписей по осям
font.main	Шрифт для заголовков
font.sub	Шрифт для подзаголовков
ps	Размер точки в шрифте (приблизительно 0.3 мм)
family	Семейство шрифтов. Стандартные значения – serif, sans и mono

Размер и стиль шрифта установить просто, тогда как с семейством шрифтов дело обстоит немного сложнее. Это происходит потому, что отображение serif, sans и mono зависит от устройства. Например, под Windows mono отображается как TT Courier New, serif – как TT Times New Roman, а sans – как TT Arial (TT обозначает шрифт типа True Type). Если вы удовлетворены таким отображением семейств шрифтов, то можете использовать параметры типа family="serif", чтобы добиться желаемого результата. Если вы не удовлетворены, вам нужно создать новую систему соответствий. Под Windows можете назначать эти соответствия при помощи функции windowsFont().

Например, после выполнения команды

```
windowsFonts (
A=windowsFont("Arial Black"),
B=windowsFont("Bookman Old Style"),
C=windowsFont("Comic Sans MS")
)
```

можно использовать A, B и C как названия семейств шрифтов.

В этом случае par(family="A") назначит шрифт Arial Black

### Размеры диаграммы и полей

Можно определять размер диаграммы и полей при помощи параметров, приведенных в табл. 6.

Таблица 6. Параметры для определения размеров диаграммы и полей

Параметр	Описание
pin	Размер диаграммы (ширина, высота) в дюймах
mai	Числовой вектор, задающий размеры полей, где параметры с (низ, лево, верх, право) измеряются в дюймах
mar	Числовой вектор, задающий размеры полей, где с (низ, лево, верх, право) измеряются в числе строк. По умолчанию это с(5, 4, 4, 2) + 0.1

Команда par(pin=c(4,3), mai=c(1,.5, 1, .2)) позволяет создавать диаграммы размером 4 дюйма в ширину и 3 дюйма в высоту с шириной полей сверху и снизу по одному дюйму, слева 0.5 дюйма и справа 0.2 дюйма.

**Пример.** Представленный ниже программный код позволяет получить диаграммы, показанные на рис. 7

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
opar <- par(no.readonly=TRUE)
par(pin=c(2, 3))
par(lwd=2, cex=1.5)
par(cex.axis=.75, font.axis=3)
plot(dose, drugA, type="b", pch=19, lty=2, col="red")
plot(dose, drugB, type="b", pch=23, lty=6, col="blue",
bg="green")
par(opar)
```

Сначала вводятся данные в виде векторов, затем сохраняются текущие графические параметры (чтобы можно было восстановить их позднее). Затем происходит изменение графических параметров так, чтобы графики были 2 дюйма в ширину и 3 дюйма в высоту. Кроме того, линии будут в два раза шире, а символы – в 1.5 раза крупнее, чем по умолчанию.

Значения на осях даны курсивом, их размер составляет 75% от размера по умолчанию. Затем создан первый график с заполненными красными кружками и пунктирными линиями. Второй график содержит зеленые ромбы с синей каймой и синие линии. И восстановление исходных графических параметров.

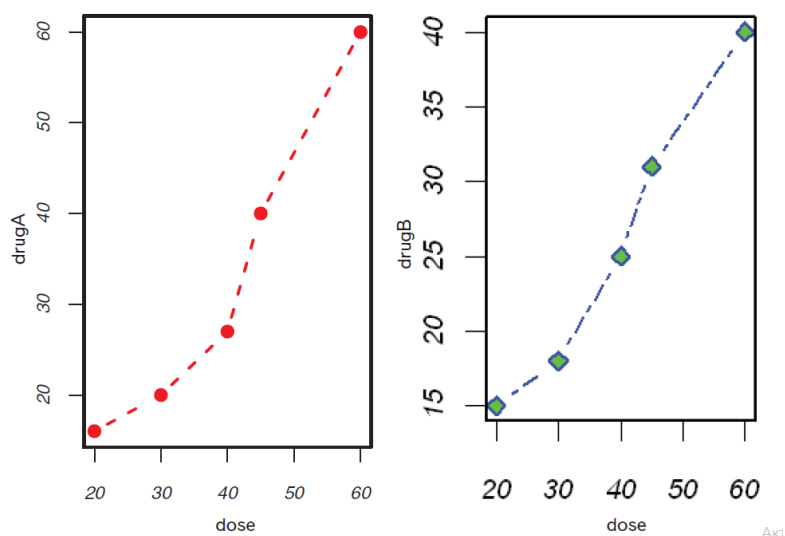


Рисунок 7. Линейный график зависимости реакции пациента от дозы лекарств А и В

Параметры, заданные при помощи функции `par()`, применяются к обоим графикам, а параметры, назначенные «внутри» функций `plot()`, действуют только для соответствующего графика.

### Добавление текста, настройка параметров осей и условных обозначений

Для многих графических функций высокого уровня (например, `plot`, `hist`, `boxplot`) возможен контроль не только графических параметров, но и параметров осей и надписей. К примеру, при помощи приведенного ниже программного кода можно разместить на диаграмме заголовок (`main`), подзаголовок (`sub`) и подписи осей (`xlab`, `ylab`), а также задать диапазон значений на осях (`xlim`, `ylim`). Результат представлен на рис. 8.

```
plot(dose, drugA, type="b",
```

```
col="red", lty=2, pch=2, lwd=2,
main="клинические испытания прпарата А",
sub="это вымышленные данные",
xlab="Доза", ylab="Эффект от препарата",
xlim=c(0, 60), ylim=c(0, 70))
```

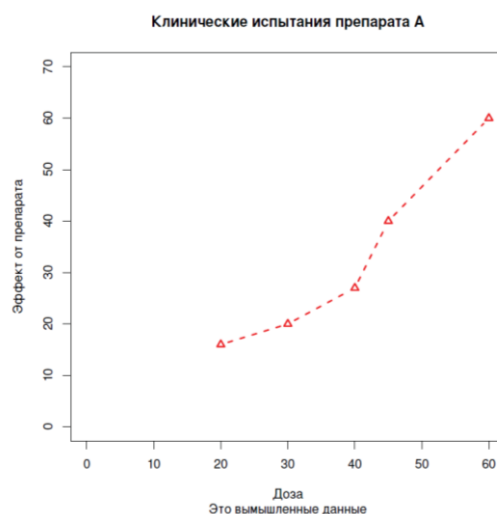


Рисунок 8. Линейный график зависимости реакции пациента от дозы лекарства А с заголовком, подзаголовком и модифицированными осями

Некоторые графические функции высокого уровня по умолчанию выводят надписи и подписи на диаграммах. От них можно избавиться, указав `ann=FALSE` как один из аргументов команд `plot()` или `par()`.

### Заголовки

Для размещения заголовков и подписей осей на диаграмме используется функция `title()`. Формат ее применения таков:

```
title(main="Мой_заголовок", sub="подзаголовок",
xlab="подпись_по_оси_x", ylab="подпись_по_оси_y")
```

Графические параметры (такие как размер и тип шрифта, ориентация и цвет текста) также можно задать при помощи функции `title()`.

**Пример:** следующий программный код позволяет получить диаграмму с красным заголовком, синим подзаголовком и зелеными подписями по осям, размер которых на 25% меньше, чем по умолчанию:

```
title(main=" Мой_заголовок ", col.main="red",
sub="мой подзаголовок ", col.sub="blue",
xlab="моя_подпись_по_оси_x ", ylab="моя_подпись_по_оси_Y",
col.lab="green", cex.lab=0.75)
```

### Оси

Вместо осей, создаваемых на диаграммах по умолчанию, можно создать оси по своему усмотрению, используя функцию `axis()`. Формат ее применения таков (все параметры описаны в табл. 7):

```
axis(side, at=, labels=, pos=, lty=, col=, las=, tck=, ...).
```

Таблица. 3.7. Параметры осей

Параметр	Описание
----------	----------

side	Цифра, определяющая, с какой стороны диаграммы рисовать ось (1 = низ, 2 = лево, 3 = верх, 4 = право)
at	Числовой вектор, который задает положение делений на осях
labels	Текстовый вектор, который содержит подписи под делениями осей (если вектор не задан, используются значения вектора at)
pos	Координата оси (то есть значение другой оси, в котором первая ось пересекает ее)
lty	Тип линии
col	Цвет линии и делений оси
las	Положение подписей делений по отношению к оси (0 = параллельно, 2 = перпендикулярно)
tck	Длина деления оси, выражается в виде доли от длины диаграммы (отрицательное число означает положение деления снаружки от рамки диаграммы, положительное число – внутри рамки диаграммы, 0 – отсутствие делений, 1 – сетка); значение по умолчанию –0.01

При создании осей нужно предотвратить появление осей, которые создаются по умолчанию графической функцией высокого уровня. Аргумент `axes=FALSE` подавляет создание всех осей (даже рамки вокруг диаграммы, если не добавлен аргумент `frame.plot=TRUE`). Аргументы `xaxt="n"` и `yaxt="n"` отменяют создание x- и y-осей соответственно (при этом рамка без делений остается).

## Задание и порядок выполнения лабораторной работы №2.2

1. Ознакомиться с методическими указаниями,
2. Исследовать основные функции и команды языка R, представленные в данной лабораторной работе
3. Выполнить все примеры.
4. Реализовать код и прописать комментарии к каждому действию

```
x <- c(1:10)
y <- x
z <- 10/x
opar <- par(no.readonly=TRUE)
par(mar=c(5, 4, 4, 8) + 0.1)
plot(x, y, type="b",
pch=21, col="red",
 yaxt="n", lty=3, ann=FALSE)
lines(x, z, type="b", pch=22, col="blue", lty=2)
axis(2, at=x, labels=x, col.axis="red", las=2)
axis(4, at=z, labels=round(z, digits=2),
 col.axis="blue", las=2, cex.axis=0.7, tck=-.01)
mtext("y=1/x", side=4, line=3, cex.lab=1, las=2, col="blue")
title("Пример осей",
 xlab="значение переменной X",
 ylab="Y=X")
par(opar)
```

5. Изучить самостоятельно добавление Легенды к диаграмме
6. По экспериментальным данным провести построение 3 - 4 различных диаграмм
7. Используя изученные функции и команды провести модификацию построенных графиков.

### Содержание отчета

Отчет по выполняемой лабораторной работе выполняется каждым студентом индивидуально на листах формата А4 в рукописном или машинном варианте исполнения и должен содержать:

- название работы;
- цель и задачи исследований;
- набор экспериментальных данных;
- построенные диаграммы;
- программный код с комментариями;
- выводы по работе.

### Контрольные вопросы

1. Принцип построения диаграмм в языке R.
2. Функции изменения графических параметров диаграмм
3. Команды Добавление текста,
4. Команды настройки параметров осей
5. Команды добавления условных обозначений.

## Библиография

1. Алексей Шипунов и др. Наглядная статистика. Используем R! – М.: ДМК Пресс, 2014. – 298 с. [Электронный ресурс]. Режим доступа: <http://ashipunov.info/shipunov/school/books/rbook.pdf>.
2. Зарядов И.С. Введение в статистический пакет R: типы переменных, структуры данных, чтение и запись информации, графика. М.: Издательство Российского университета дружбы народов, 2010. – 207 с.
3. 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.
4. Официальный сайт RStudio. Режим доступа: <https://www.rstudio.com>.
5. Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных [Электронный ресурс]. Режим доступа: <http://machinelearning.ru>.
6. Мاستицкий С.Э., Шитиков В.К. (2014) Статистический анализ и визуализация данных с помощью R. – Электронная книга. Режим доступа: <http://r-analytics.blogspot.com>

## **Организация защиты и критерии оценивания выполнения лабораторных работ**

К защите представляется отчет, включающий в себя результаты выполнения лабораторной работы, выполненный согласно правилам и единый титульный лист, на котором отмечаются результаты выполнения заданий.

К отчетам прилагается электронный носитель, содержащий папки с исполняемыми файлами, файлами отчетов и презентациями (если требуется в задании) созданных в ходе выполнения лабораторных работ.

На проверку теоретической подготовки, проводимой по контрольным вопросам, отводится 5–6 минут.

*Степень усвоения теоретического материала оценивается по следующим критериям:*

- **оценка «отлично» выставляется, если:**
  - последовательно, четко, связно, обоснованно и безошибочно с использованием принятой терминологии изложен учебный материал, выделены главные положения, ответ подтвержден конкретными примерами, фактами;
  - самостоятельно и аргументировано сделан анализ, обобщение, выводы, установлены межпредметные (на основе ранее приобретенных знаний) и внутрипредметные связи, творчески применены полученные знания в незнакомой ситуации;
  - самостоятельно и рационально используются справочные материалы, учебники, дополнительная литература, первоисточники; применяется система условных обозначений при ведении записей, сопровождающих ответ; используются для доказательства выводы из наблюдений и опытов, ответ подтверждается конкретными примерами;
  - допускает не более одного недочета, который легко исправляется по требованию преподавателя.
- **оценка «хорошо» ставится, если:**
  - дан полный и правильный ответ на основе изученных теорий; допущены незначительные ошибки и недочеты при воспроизведении изученного материала, определения понятий, неточности при использовании научных терминов или в выводах и обобщениях из наблюдений и опытов; материал излагает в определенной логической последовательности;
  - самостоятельно выделены главные положения в изученном материале; на основании фактов и примеров проведено обобщение, сделаны выводы, установлены внутрипредметные связи.
  - допущены одна негрубая ошибка или не более двух недочетов, которые исправлены самостоятельно при требовании или при небольшой помощи преподавателя; в основном усвоил учебный материал.
- **оценка «удовлетворительно» ставится, если:**
  - усвоено основное содержание учебного материала, но имеются пробелы в усвоении материала, не препятствующие дальнейшему изучению; материал излагает несистематизированно, фрагментарно, не всегда последовательно;
  - показана недостаточная сформированность отдельных знаний и умений; выводы и обобщения аргументируются слабо, в них допускаются ошибки;
  - допущены ошибки и неточности в использовании научной терминологии, даются недостаточно четкие определения понятий; в качестве доказательства не используются выводы и обобщения из наблюдений, фактов, опытов или допущены ошибки при их изложении;

- обнаруживается недостаточное понимание отдельных положений при воспроизведении текста учебника (записей, первоисточников) или неполные ответы на вопросы преподавателя, с допущением одной – двух грубых ошибок.
- **оценка «неудовлетворительно» ставится, если:**
  - не усвоено и не раскрыто основное содержание материала; не сделаны выводы и обобщения;
  - не показано знание и понимание значительной или основной части изученного материала в пределах поставленных вопросов или показаны слабо сформированные и неполные знания и неумение применять их к решению конкретных вопросов и задач по образцу;
  - при ответе (на один вопрос) допускается более двух грубых ошибок, которые не могут быть исправлены даже при помощи преподавателя;
  - не даются ответы ни на один из поставленных вопросов.

Оценка выполнения лабораторных работ проводится по следующим критериям
- **оценка «отлично» ставится, если студент:**
  - творчески планирует выполнение работы;
  - самостоятельно и полностью использует знания программного материала;
  - правильно и аккуратно выполняет задание;
  - умеет пользоваться литературой и различными информационными источниками;
  - выполнил работу без ошибок и недочетов или допустил не более одного недочета
- **оценка «хорошо» ставится, если студент:**
  - правильно планирует выполнение работы;
  - самостоятельно использует знания программного материала;
  - в основном правильно и аккуратно выполняет задание;
  - умеет пользоваться литературой и различными информационными источниками;
  - выполнил работу полностью, но допустил в ней: не более одной негрубой ошибки и одного недочета или не более двух недочетов.
- **оценка «удовлетворительно» ставится, если студент:**
  - допускает ошибки при планировании выполнения работы;
  - не может самостоятельно использовать значительную часть знаний программного материала;
  - допускает ошибки и неаккуратно выполняет задание;
  - затрудняется самостоятельно использовать литературу и информационные источники;
  - правильно выполнил не менее половины работы или допустил:
    - не более двух грубых ошибок или не более одной грубой и одной негрубой ошибки и одного недочета;
    - не более двух– трех негрубых ошибок или одной негрубой ошибки и трех недочетов;
    - при отсутствии ошибок, но при наличии четырех–пяти недочетов.
- **оценка «неудовлетворительно» ставится, если студент:**
  - не может правильно спланировать выполнение работы;
  - не может использовать знания программного материала;
  - допускает грубые ошибки и неаккуратно выполняет задание;
  - не может самостоятельно использовать литературу и информационные источники;
  - допустил число ошибок недочетов, превышающее норму, при которой может быть выставлена оценка «3»;
  - если правильно выполнил менее половины работы;
  - не приступил к выполнению работы;
  - правильно выполнил не более 10% всех заданий.