Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

# ОТЧЁТ

по лабораторной работе №2

КОРРЕЛЯЦИОННЫЙ И РЕГРЕССИОННЫЙ АНАЛИЗ ДАННЫХ

Выполнил:

ст. гр. ИС/б-21-2-о

Мовенко К. М.

Проверила:

Сырых О. А.

Севастополь

2024

# ЦЕЛЬ РАБОТЫ

Исследовать возможности языка R для проведения корреляционного и регрессионного анализа данных. Создание набора данных для проведения корреляционного и регрессионного анализа данных

# ЗАДАНИЕ

* 1. Ознакомится с методическими указаниями;
  2. Исследовать основные функции и команды языка R, представленные в данной лабораторной работе;
  3. Выполнить все примеры;
  4. Подобрать экспериментальные данные для анализа;
  5. Выполнить ввод данных с клавиатуры,
  6. Провести экспорт данных из текстового файла с разделителями;
  7. Выполнить экспорт данных из Excel.

# выполнение примеров

Прежде чем выбрать данные для анализа были выполнены тестовые примеры из методических указаний. Так была рассмотрена новая структура языка R – таблица данных.

Через заполнение численных и строковых векторов и функции data.frame() в окружение была перенесена таблица пациентов из методических указаний (рисунок 1). Можно заметить, что названия столбцов были взяты из имён соответствующих векторов.

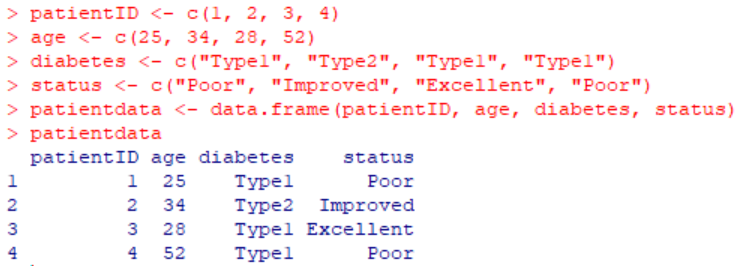


Рисунок 1 – Создание таблицы данных

Были опробованы два способа обращения к колонкам таблицы – через их индексы (нумерация с 0) и вектор с их названиями (рисунок 2).

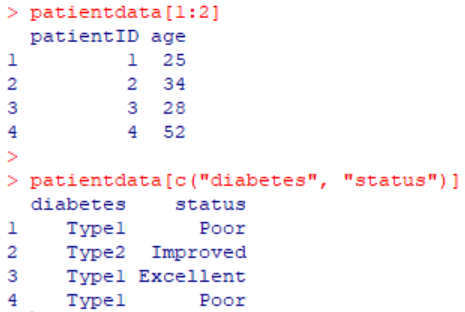


Рисунок 2 – Обращение к элементам таблицы данных

Также было протестировано обращение к столбцам таблицы с помощью символа $ (рисунок 3). При обращении через имя столбца он становится доступен в виде переменной векторного типа.

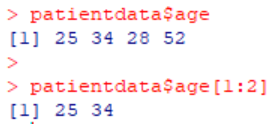


Рисунок 3 – Обращение к переменным таблицы

С помощью аргумента row.names значения столбца patientID были установлены в качестве обозначения строк при выводе данных и создания диаграмм (рисунок 4).

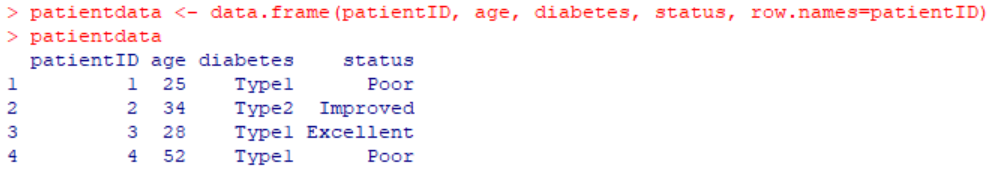


Рисунок 4 – Именование строк таблицы

Были рассмотрены категориальные данные таблицы. С помощью функции factor() вектор номинальных данных diabetes был сохранён в виде вектора целых чисел, которым соответствуют исходные значения (рисунок 5).

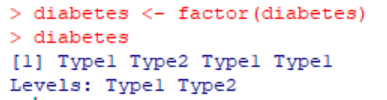


Рисунок 5 – Преобразование в фактор

Затем был рассмотрен вектор порядковых данных status. Для него была проведена похожая операция, однако благодаря аргументу ordered теперь данные можно сравнивать (рисунок 6).

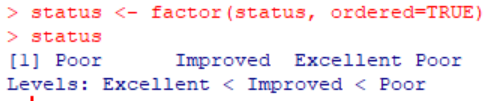


Рисунок 6 - Преобразование в упорядоченный фактор

По умолчанию данные упорядочиваются в алфавитном порядке, однако в данном случае имеет смысл изменить порядок в обратную сторону. С помощью аргумента levels порядок был выбран вручную (рисунок 7).

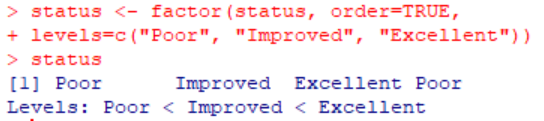


Рисунок 7 – Ручное упорядочивание элементов фактора

Из полученных факторов и созданных ранее векторов была заново построена известная таблица данных (рисунок 8).

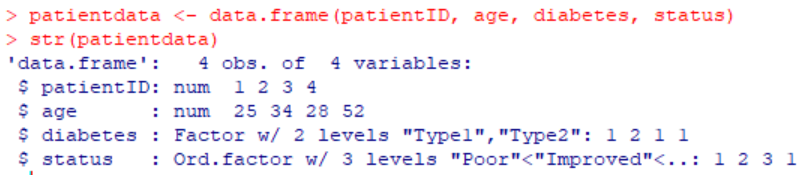


Рисунок 8 – Построение таблицы из факторов

Далее были рассмотрены списки. Для примера был создан список, состоящий из числового и строкового векторов, строки и матрицы (рисунок 9). Некоторым элементам списка было дано название.

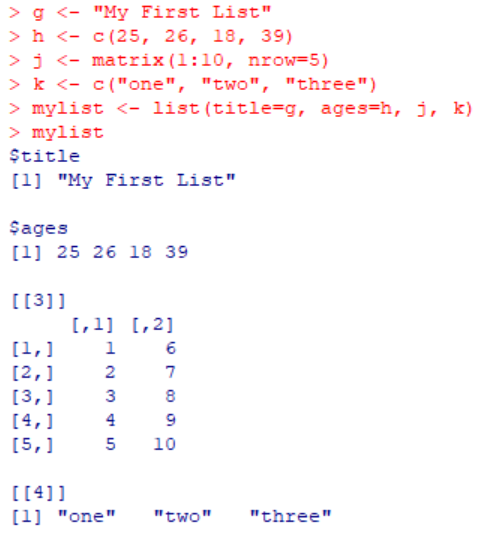


Рисунок 9 – Создание и вывод списка

Далее был протестирован ввод данных таблицы с клавиатуры. Для этого была создана пустая таблица mydata с тремя столбцами. С помощью команды edit() была открыта и заполнена её копия (Рисунок 10).

mydata <- data.frame(age=numeric(0), gender=character(0), weight=numeric(0))

mydata <- edit(mydata)

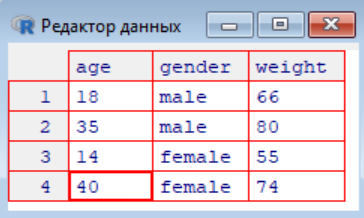


Рисунок 10 – Ручное заполнение таблицы данных

Затем было опробовано заполнение таблицы из файла. Для этого был создан csv-файл с табличными данными, которые были перенесены в R через функцию read.table() (рисунок 11).

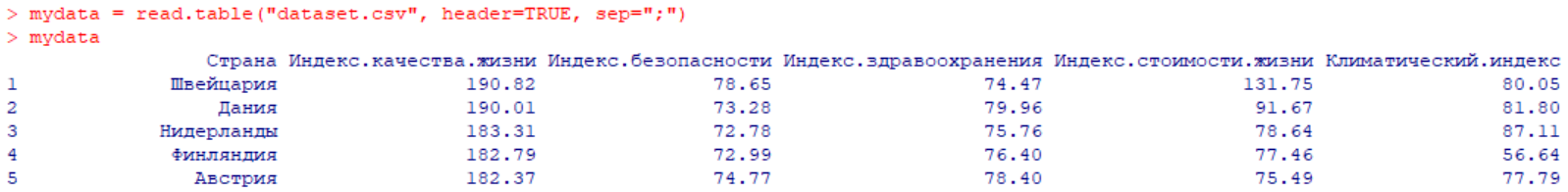


Рисунок 11 – Чтение таблицы из файла

Для чтения Exel-файлов был опробован пакет RODBC, однако из-за несовместимости с Windows 64-bit вместо него был использован пакет readxl (рисунок 12).

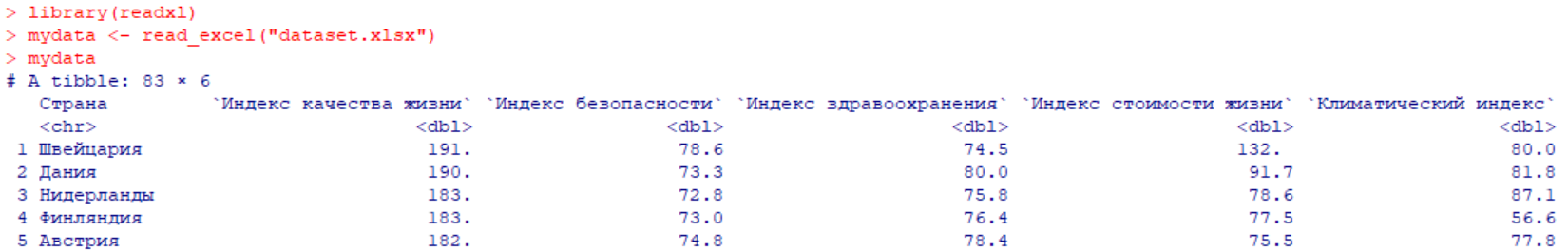


Рисунок 12 – Чтение Excel-таблицы

Было протестировано построение диаграмм. С помощью указанных ниже команд была построена диаграмма рассеяния, отображающая зависимость веса автомобилей от их расхода топлива (рисунок 13). Набор данных mtcars – встроенный.

attach(mtcars)

plot(wt, mpg)

abline(lm(mpg~wt))

title(“Regression of MPG on Weight”)

detach(mtcars)

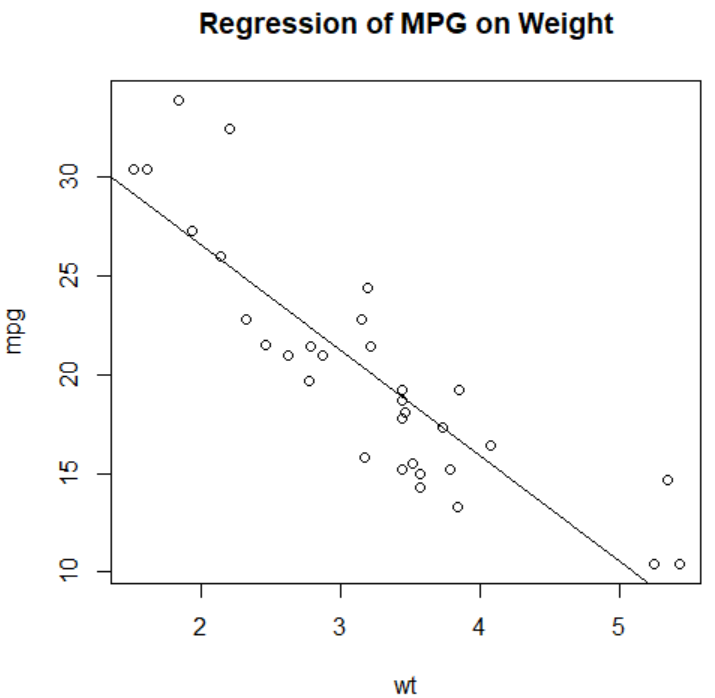


Рисунок 13 – Диаграмма с корреляционной прямой

С помощью команды pdf() тот же самый график был распечатан в новом PDF-файле mygraph.pdf. Рисунок диаграммы идентичен рисунку 13.

pdf("mygraph.pdf")

attach(mtcars)

plot(wt, mpg)

abline(lm(mpg~wt))

title("Regression of MPG on Weight")

detach(mtcars)

dev.off()

В окружение была перенесена таблица из методических указаний, описывающая реакцию пациента на два лекарства в пяти дозировках (рисунок 14).

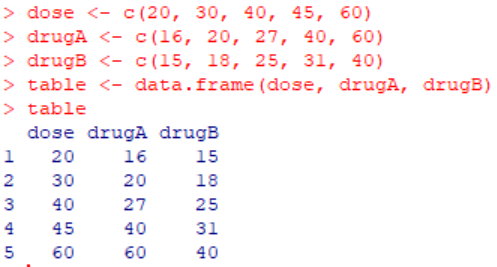


Рисунок 14 – Занесение таблицы

По этой таблице с помощью команды plot() была построена диаграмма зависимости реакции пациента на препарат A от дозы (рисунок 15). На графике показаны и точки, и линии.

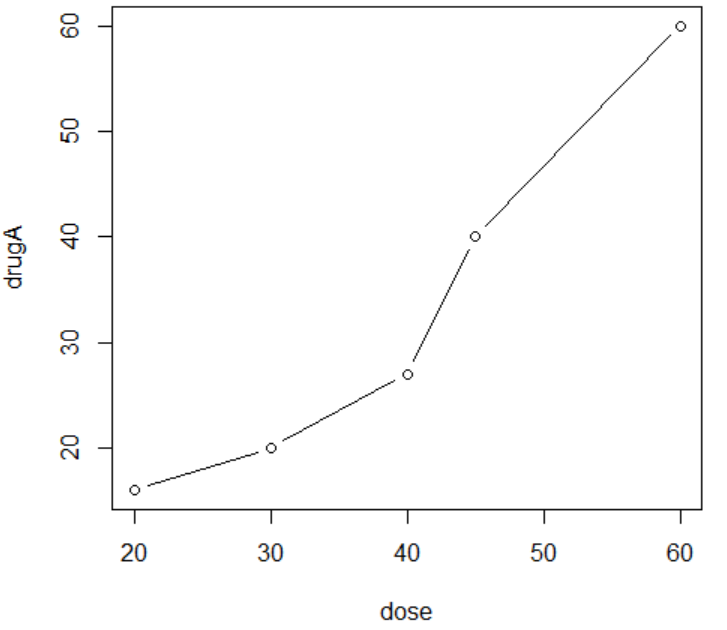


Рисунок 15 – Диаграмма зависимости параметров

Далее были рассмотрены графические параметры. С помощью команды par() точки были заменены на треугольники, а сплошные линии – пунктиром (рисунок 16).

opar <- par(no.readonly=TRUE)

par(lty=2, pch=17)

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

par(opar)

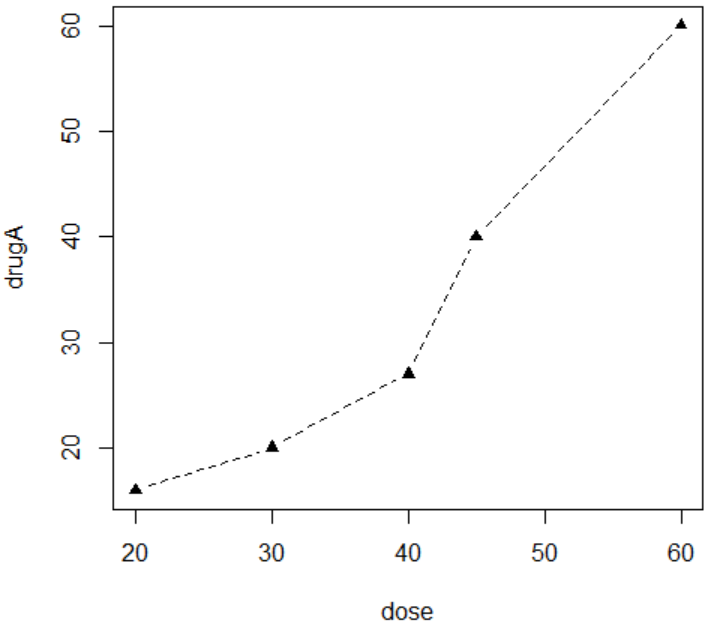


Рисунок 16 – Диаграмма с изменёнными параметрами

Тот же самый эффект был достигнут при указании значений параметров напрямую в команде plot().

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

Были исследованы значения некоторых параметров функции plot(): pch (вид символа), cex (размер символа), lty (тип линии) и lwd (толщина линии). Диаграмма была перестроена с указанием каждого параметра (рисунок 17).

plot(dose, drugA, type="b", pch=11, lty=3, lwd=3, cex=1.5)

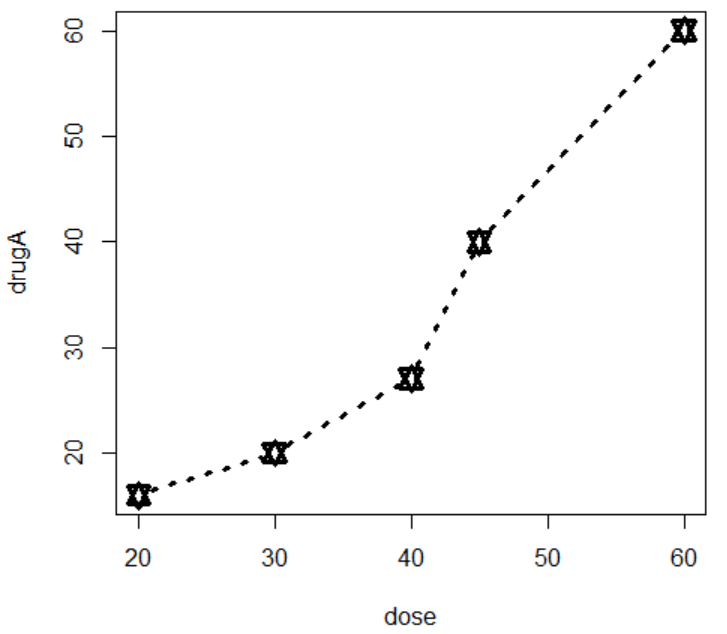


Рисунок 17 – Диаграмма с новыми значениями параметров

Были рассмотрены параметры, отвечающие за цвет, форматы представления цвета и функции, стоящие цветовые вектора. С помощью функций rainbow() и gray() были построены круговые диаграммы радужного (рисунок 18) и серого (рисунок 19) спектров.

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)

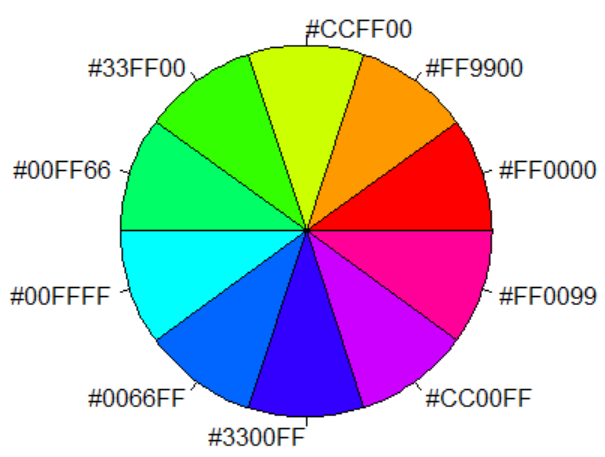


Рисунок 18 – Диаграмма цветов радуги

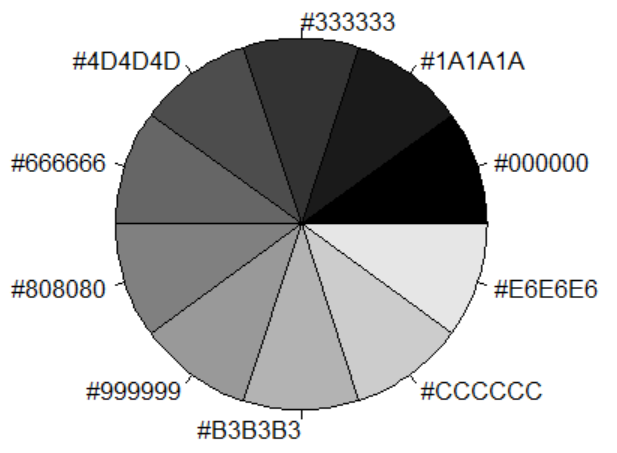


Рисунок 19 – Диаграмма оттенков серого

Были рассмотрены подробнее параметры, отвечающие за размер текста – как в целом, так и конкретно на осях, заголовке и подзаголовках (рисунок 20).

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

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

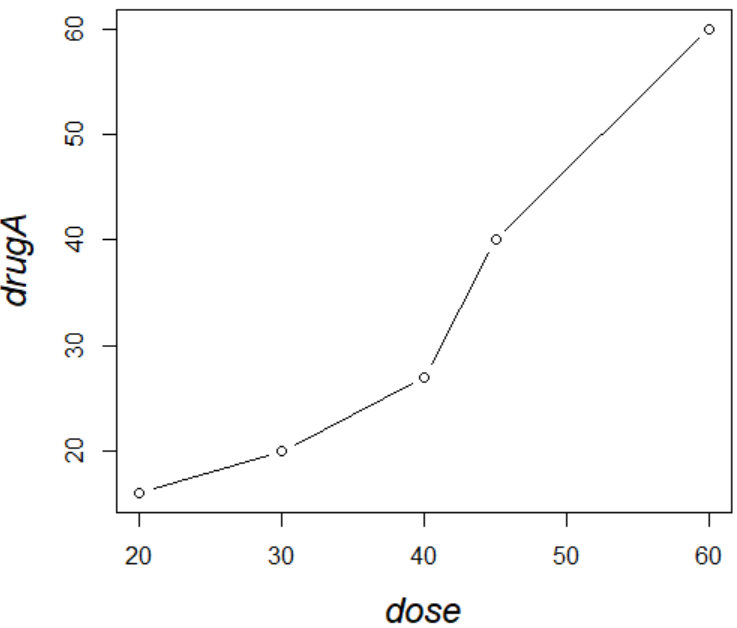


Рисунок 20 – Вывод графика с изменёнными параметрами размера символов

Были исследованы параметры pin и mai. С их помощью, а также указанием других параметров, была выведена суженная версия указанного ранее графика (рисунок 21).

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, drugB, type="b", pch=23, lty=6, col="blue", bg="green")

par(opar)

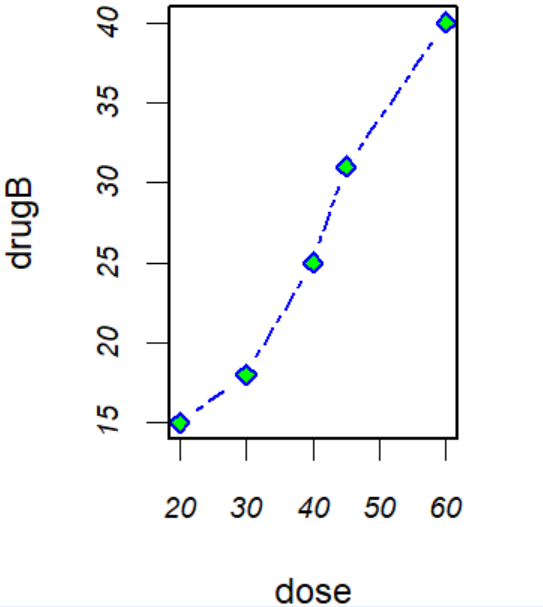


Рисунок 21 – Применение параметра pin на графике

С помощью параметров main, sub, xlab и ylab график был выведен вместе с соответствующим заголовком, подзаголовком и подписями осей (рисунок 22).

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

col="red", lty=2, pch=2, lwd=2,

main="Клинические испытания препарата A",

sub="Это вымышленные данные",

xlab="Доза", ylab="Эффект от препарата",

xlim=c(0, 60), ylim=c(0, 70))

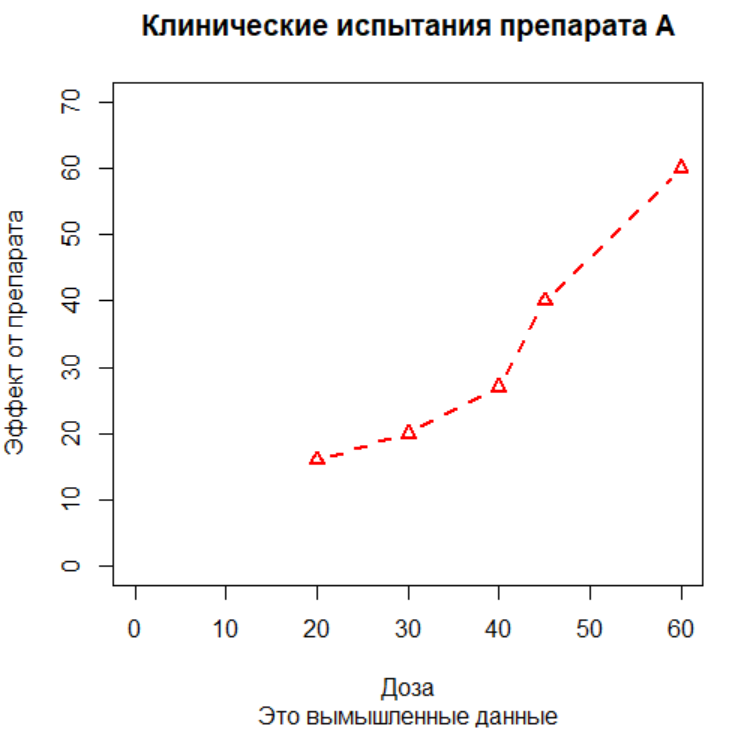


Рисунок 22 – Добавление подписей к графику

Для вывода подписей была также опробована функция title(). В ней при указании текста и параметров к графику были добавлены разноцветные подписи и заголовки (рисунок 23).

plot(dose, drugA, type="b", ann=FALSE)

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)

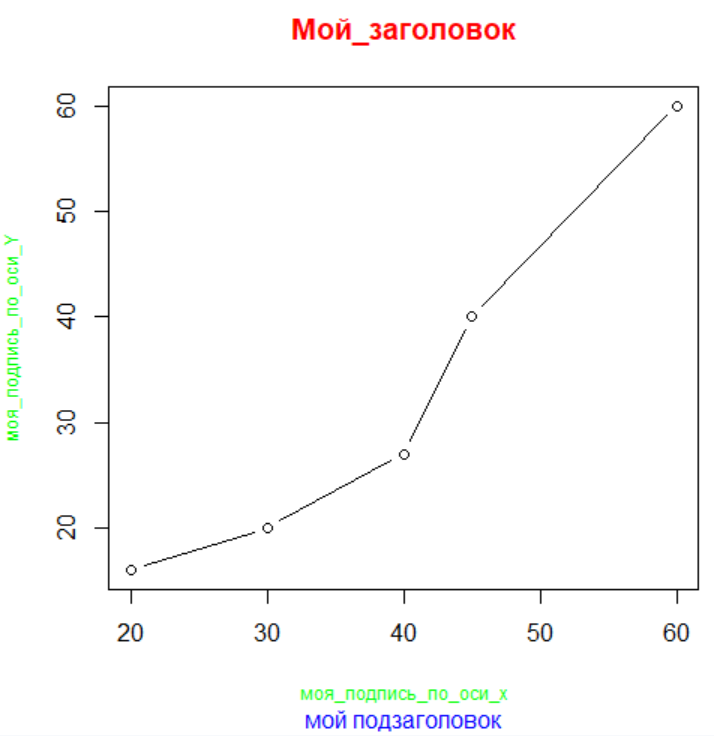


Рисунок 23 – Добавление цветных заголовков

Был реализован код, указанный в методических указаниях, который выводит в одном окне графики линейной и нелинейной функций (рисунок 24). В коде прописаны различные параметры отображения графиков: цвета, размеры, вид осей, заголовки и подписи и др. Каждая строчка кода была снабжена комментарием.

x <- c(1:10) # значения x

y <- x # y = x

z <- 10/x # z = 10/x

opar <- par(no.readonly=TRUE) # сохранить параметры по умолчанию

par(mar=c(5, 4, 4, 8) + 0.1) # новые размеры полей графиков

plot(x, y, type="b", # вывод графика y=x

pch=21, col="red", # символ - красная точка

yaxt="n", lty=3, ann=FALSE) # пунктир, убрать подписи и ось Y

lines(x, z, type="b", pch=22, # вывод графика z=10/x квадратами

col="blue", lty=2) # цвет синий, пунктир

axis(2, at=x, labels=x, # левая ось, все значения x

col.axis="red", las=2) # подписи красные, перпендикулярные оси

axis(4, at=z, labels=round(z, digits=2), # правая ось, значения z

col.axis="blue", las=2, # подписи синие, перпендикулярные

cex.axis=0.7, tck=-.01) # размер подписей и длина делений

mtext("y=1/x", side=4, line=3, # надпись возле правой оси

cex.lab=1, las=2, col="blue") # перпендикулярность и синий цвет

title("Пример осей", # заголовок графика

xlab="значение переменной X", # подпись оси x

ylab="Y=X") # подпись оси y

par(opar) # возвращение исходных параметров в системе

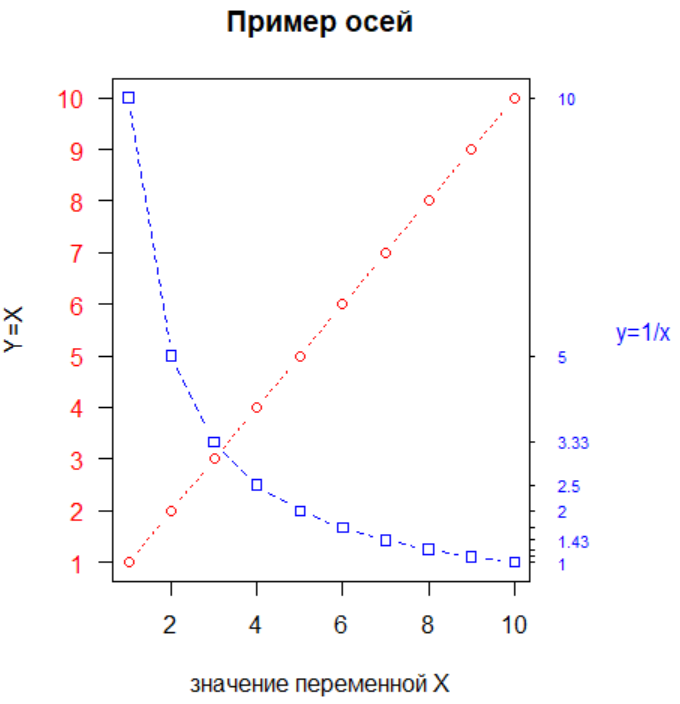


Рисунок 24 – Работа с графиками и их параметрами

СВЯЗЬ ОТ 0.6 ДО 1, 2-3 ПАРЫ

# ХОД РАБОТЫ

# ВЫВОД