Лабораторная работа №2.1

Корреляционный и регрессионный анализ данных. Создание набора данных

2.1.1 Цель

− исследовать возможности языка R для проведения корреляционного и регрессионного анализа данных;

− создание набора данных для проведения корреляционного и регрессионного анализа данных

2.1.2 Порядок выполнения работы

1. Ознакомится с методическими указаниями

2. Исследовать основные функции и команды языка R, представленные в данной лабораторной работе

3. Выполнить все примеры.

4. Выполнить ввод данных с клавиатуры

5. Провести экспорт данных из текстового файла с разделителями

6. Выполнить экспорт данных из Excel.

7. Подобрать экспериментальные данные для анализа (пример данных представлен в Приложении А)

2.1.3 Ход работы

Были проанализированы примеры, представленные в методических указаниях.

Для создания таблицы используется функция data.frame() (листинг 1).

Листинг 1 – Создание таблицы

mydata <- data.frame(col1, col2, col3,…)

В листинге 1 col1, col2, col3,… это векторы любого типа (текстового, числового или логического), которые станут столбцами таблицы. Названия каждому столбцу можно присвоить при помощи функции names().

Пример 1 представлен в листинге 2.

Листинг 2 – Пример 1

patientID <- c(1, 2, 3, 4)

age <- c(25, 34, 28, 52)

diabetes <- c("Type1", "Type2", "Type1", "Type1")

status <- c("Poor", "Improved", "Excellent", "Poor")

patientdata <- data.frame(patientID, age, diabetes, status)

patientdata

Результат выполнения примера 1 представлен на рисунке 1.

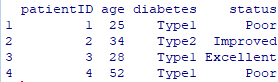


Рисунок 1 – Результат выполнения примера 1

Существует несколько способов обозначить элементы таблицы данных. Можно использовать индексы или можно указывать номера столбцов. Пример кода представлен в листингах 2 – 3.

Листинг 2 – Пример 2

patientdata [1:2]

Листинг 3 – Пример 3

patientdata [c("diabetes", "status")]

Результат выполнения примера 2 и примера 3 представлен на рисунках 2 – 3.

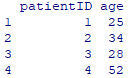


Рисунок 2 – Результат выполнения примера 2

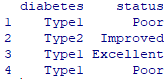


Рисунок 3 – Результат выполнения примера 3

Знак ‘$’ используется, чтобы обозначить определенную переменную в таблице данных. Пример кода представлен в листинге 4.

Листинг 4 – Пример 4

patientdata$age

Результат выполнения примера 4 представлен на рисунке 4.



Рисунок 4 – Результат выполнения примера 4

В R названия строк могут быть назначены при помощи параметра row.names функции создания таблицы данных. Пример кода представлен в листинге 5.

Листинг 5 – Пример 5

patientdata <- data.frame(patientID, age, diabetes, status, row.names=patientID)

patientdata

Результат выполнения примера 5 представлен на рисунке 5.

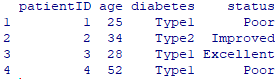


Рисунок 5 – Результат выполнения примера 5

Функция factor() сохраняет категориальные данные в виде вектора из целых чисел в диапазоне от одного до k (где k – число уникальных значений категориальной переменной) и в виде внутреннего вектора из цепочки символов (исходных значений переменной), соответствующим этим целым числам.

По умолчанию уровни фактора присваиваются значениям вектора в алфавитном порядке. Для упорядоченных факторов редко подходит алфавитный порядок уровней, предлагающийся по умолчанию. Установку по умолчанию можно изменить при помощи параметра levels.

Пример использования функции factor() представлены в листинге 6.

Листинг 6 – Пример 6

patientID <- c(1, 2, 3, 4)

age <- c(25, 34, 28, 52)

diabetes <- c("Type1", "Type2", "Type1", "Type1")

status <- c("Poor", "Improved", "Excellent", "Poor")

diabetes <- factor(diabetes)

status <- factor(status, order=TRUE)

patientdata <- data.frame(patientID, age, diabetes, status)

Получить информацию об объекте можно при помощи функции str(object) (листинг 7).

Листинг 7 – Пример использования str(object)

str(patientdata)

Результат использования функции str(patientdata) представлен на рисунке 6.

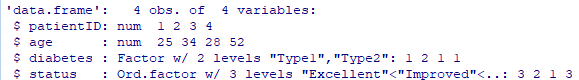


Рисунок 6 – Результат использования функции str(patientdata)

Получить статистику по объекту можно при помощи функции summary(object) (листинг 8).

Листинг 8 – Пример использования summary(object)

summary(patientdata)

Результат использования функции summary(patientdata) представлен на рисунке 7.

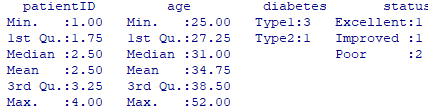


Рисунок 7 – Результат использования функции summary(patientdata)

Списки – это самый сложный тип данных в R. Фактически список – это упорядоченный набор объектов (компонентов). Список может объединять разные (возможно, не связанные между собой) объекты под одним именем. К примеру, список может представлять собой сочетание векторов, матриц, таблиц данных и даже других списков. Примеры использования списков представлены в листинге 9.

Листинг 9 – Пример 7

g <- “My First List”

h <- c(25, 26, 18, 39)

j <- matrix(1:10, nrow=5)

k <- c(“one”, “two”, “three”)

mylist <- list(title=g, ages=h, j, k)

Полученный список отобразили в консоли (рисунки 8 – 10)

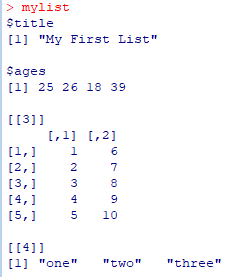


Рисунок 8 – Отображение всего списка



Рисунок 9 – Отображение второго объекта списка



Рисунок 10 – Отображение объекта списка по имени

Ввод данных с клавиатуры.

Самый простой способ введения данных – это ввод с клавиатуры. Функция edit() откроет текстовый редактор, куда можно внести свои данные. Для ввода данных необходимо:

1. Создать пустую таблицу данных (или матрицу), указав названия и типы переменных;

2. Открыть текстовый редактор с этим объектом, ввести экспериментальные данные и сохранить результат в виде объекта с данными.

Пример использования данного механизма представлен в листинге 10. Окно редактора представлено на рисунке 11.

Листинг 10 – Пример 8

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

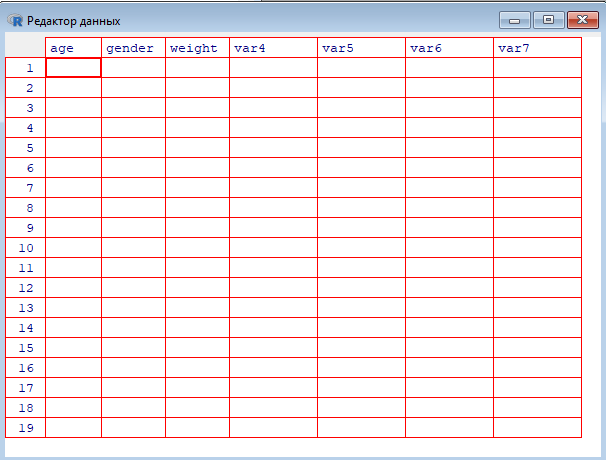


Рисунок 11 – Окно редактора

Импорт данных из текстового файла с разделителями.

Импорт данных из текстовых файлов с разделителями возможен при помощи команды read.table(), функции, которая сохраняет данные в виде таблицы. mydataframe <- read.table(file, header=логичское\_значение, sep=”разделитель”, row.names=”название”) где file – это ASCII файл с разделителями, header – это логическое значение, определяющее, содержит ли первая строка названия переменных (TRUE – да, FALSE – нет), sep указывает, каким символом разделены элементы данных, а row.names – необязательный параметр, для указания столбца (столбцов), в котором содержатся названия строк. Пример кода представлен в листинге 11.

Листинг 11 – Пример 9

grades <- read.table(“studentgrades.csv”, header=TRUE, sep=”,”, row.names=”STUDENTID”)

grades

Результат выполнения примера 9 представлен на рисунке 12.

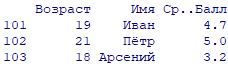


Рисунок 12 – Результат выполнения примера 9

Импорт данных из Excel.

Лучший способ прочесть файл в формате Excel – это сохранить его в формате текстового файла с разделителями и импортировать в R, как это описано выше. Функция read.xlsx() осуществляет импорт нужного листа XLSX-файла в таблицу данных. Проще всего использовать эту функцию по такой схеме: read.xlsx(file, n), где file – это путь к файлу книги Excel, а n – число листов, которые нужно импортировать. Пример кода представлен в листинге 12.

Листинг 12 – Пример 10

library(xlsx)

workbook <- “c:/myworkbook.xlsx”

mydataframe <- read.xlsx(workbook, 1)

mydataframe

Результат выполнения примера 10 представлен на рисунке 13.



Рисунок 13 – Результат выполнения примера 10

Лабораторная работа № 2.2

Корреляционный и регрессионный анализ данных. Работа с диаграммами

2.2.1 Цель:

− исследовать возможности языка R для проведения корреляционного и регрессионного анализа данных;

− исследовать возможности языка R для создания и изменения вида диаграмм

2.2.2 Порядок выполнения работы:

1. Ознакомится 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, продемонстрированный в Листинг 1.

Листинг 1 – Код программы

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.

2.2.3. Ход работы

Работа с диаграммами.

В стандартной интерактивной сессии R создается диаграмма, вводом по одной команде и добавлением элементов диаграммы, пока не получится то, что необходимо. Пример кода создания диаграммы представлен в листинге 1.

Листинг 1 – Пример 1

attach(mtcars)

plot(wt, mpg)

abline(lm(mpg~wt))

title(“Regression of MPG on Weight”)

detach(mtcars)

Результат работы примера 1 представлен на рисунке 1.

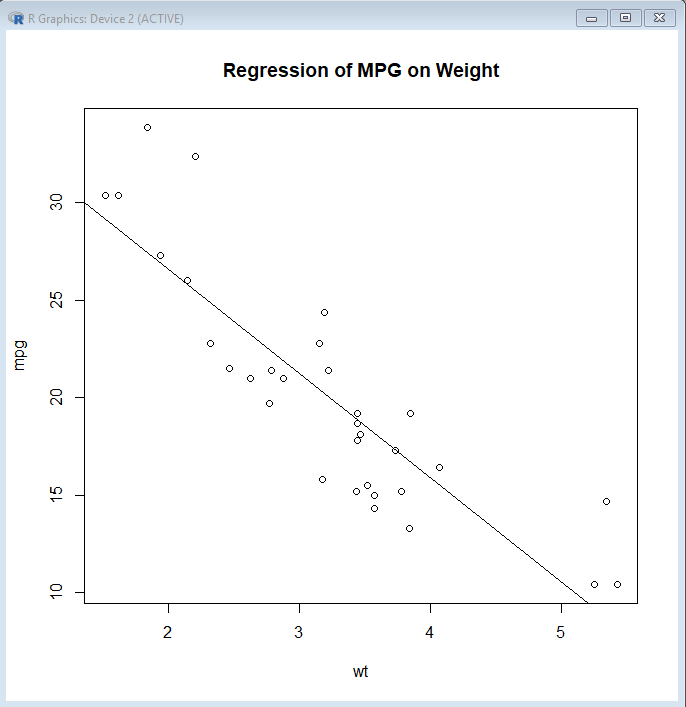


Рисунок 1 – Результат работы примера 1