Министерство науки и высшего образования 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

2019

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, 2019 – 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 заседании кафедры «Информационные системы» (протокол № 1 от 30 августа 2019 г.)

Лабораторная работа №2.1
Корреляционный и регрессионный анализ данных. Создание набора данных.

Цель:

- исследовать возможности языка R для проведения корреляционного и регрессионного анализа данных;
- создание набора данных для проведения корреляционного и регрессионного анализа данных

Время: 2 часа

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

Краткие теоретические сведения

Различают два типа связей между различными явлениями и их признаками: функциональную или жестко детерминированную, с одной стороны, и статистическую или стохастически детерминированную - с другой. Строго определить различие этих типов связи можно тогда, когда они получают математическую формулировку.

Важнейшим частным случаем статистической связи является корреляционная связь.

При функциональной связи заданному значению фактора X соответствует строго определенные значение параметра Y, что свойственно строго детерминированным процессам (связь температуры и объема, давления и объема).

При корреляционной связи заданному значению фактора X может соответствовать множество возможных значений параметра Y.

Для изучения взаимосвязей используются корреляционный и регрессионный анализ.

1. Корреляционный анализ

Основной задачей корреляционного анализа является определение формы, направленности и тесноты взаимосвязи. При исследования корреляции используются графический и аналитический подходы.

Графический анализ начинается с построения корреляционного поля. Корреляционное поле (или диаграмма рассеяния) является графической зависимостью между результатами измерений двух признаков. Для ее построения исходные данные наносят на график, отображая каждую пару значений (*xi*,*yi*) в виде точки с координатами *xi* и *yi* в прямоугольной системе координат.

Визуальный анализ корреляционного поля позволяет сделать предположение о форме взаимосвязи двух исследуемых показателей. По **форме взаимосвязи** корреляционные зависимости принято разделять на линейные (рис. 1) и нелинейные (рис. 2).

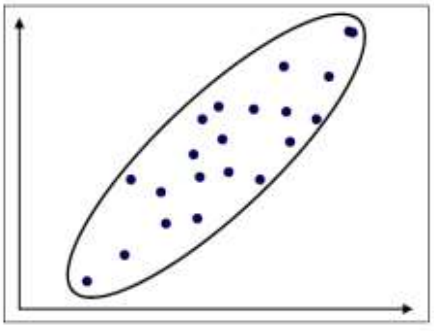

Рис 1. Линейная статистическая связь

При линейной зависимости огибающая корреляционного поля близка к эллипсу. Линейная взаимосвязь двух случайных величин состоит в том, что при увеличении одной случайной величины другая случайная величина имеет тенденцию возрастать (или убывать) по линейному закону.

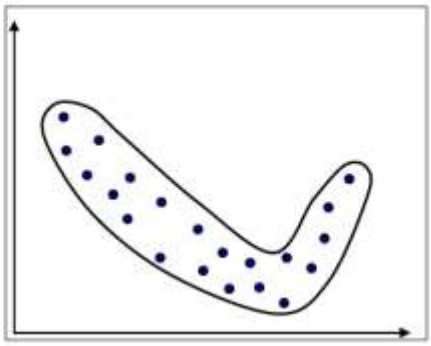

Рис 2. Нелинейная статистическая связь

Выявление формы статистической зависимости необходимо для выбора метода оценки тесноты (силы) взаимосвязи.

Направленность является положительной, если увеличение значения одного признака приводит к увеличению значения второго (рис. 3).

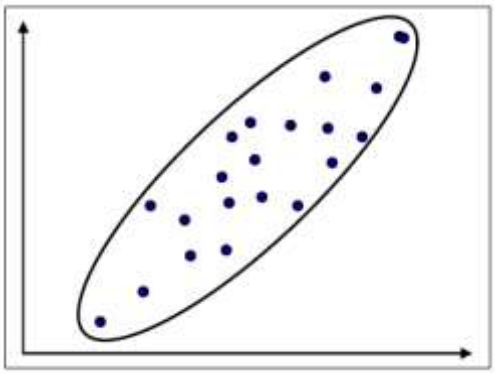

Рис 3. Положительная направленность

Направленность является отрицательной, если увеличение значения одного признака приводит к уменьшению значения второго (рис. 4).

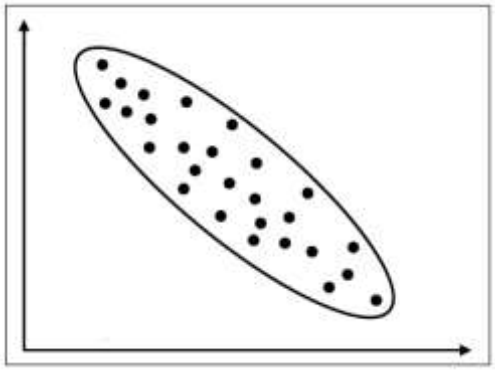

Рис 4. Отрицательная направленность

Теснота взаимосвязи может быть оценена качественно по ширине корреляционного поля – чем меньше его ширина, тем больше теснота и сильнее зависимость.

Количественная оценка тесноты взаимосвязи двух случайных величин осуществляется с помощью коэффициента корреляции *r*.

Коэффициент корреляции характеризует только линейную взаимосвязь

Направление (прямая или обратная) и сила (теснота) корреляционной связи характеризуется **коэффициентом линейной корреляции Пирсона** который рассчитывают по данным выборки *n* объектов по формуле

$$r_{xy} = \frac{n\sum xy - \sum x \sum y}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right] \cdot \left[n\sum y^2 - \left(\sum y\right)^2\right]}}$$

где *x* - значение факторного признака;

y - значение результативного признака;

n - число пар данных.

Коэффициент корреляции величина относительная; он принимает значение от минус единицы до плюс единицы, т.е. $-1 < r < 1$.

При $r > 0$ связь оценивается, как прямая, при $r < 0$ – обратная.

При $r = 0$ – связь отсутствует, при $|r| = 1$ – связь функциональная

Сила связи оценивается:

при $|r| < 0{,}3$ – как слабая,

при $0{,}3 < |r| < 0{,}7$ – умеренная,

при $|r| > 0{,}7$ – сильная.

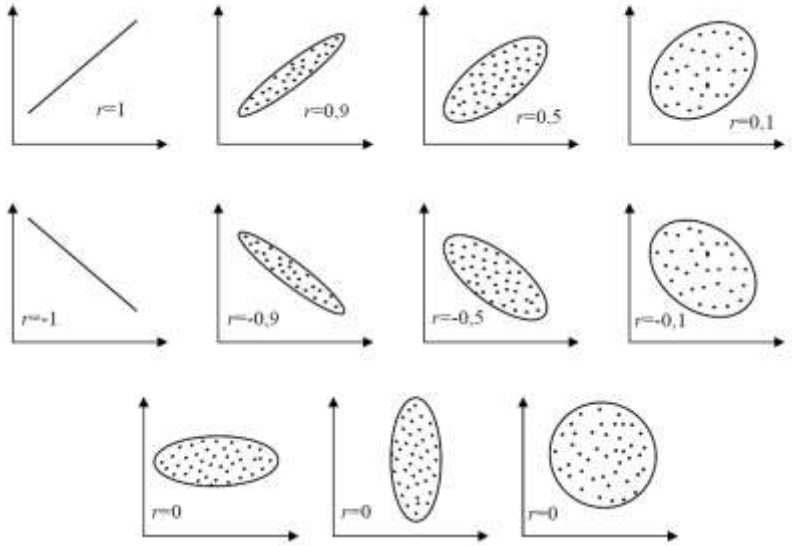

Рис 5. Корреляционные поля при различных значениях коэффициента корреляции.

2. Регрессионный анализ

В практических исследованиях возникает необходимость аппроксимировать (описать приблизительно) диаграмму рассеяния математическим уравнением. То есть зависимость между переменными величинами *Y* и *X* можно выразить аналитически с помощью формул и уравнений и графически в виде геометрического места точек в системе прямоугольных координат.

Основная особенность регрессионного анализа: при его помощи можно получить конкретные сведения о том, какую форму и характер имеет зависимость между исследуемыми переменными.

Под регрессией понимается функциональная зависимость между независимыми (объясняющими) переменными и средним значением зависимой (объясняемой) переменной, которая строится с целью предсказания этого среднего значения при фиксированных значениях переменной.

Этапы регрессионного анализа.

1. Формулировка задачи. На этом этапе формируются предварительные гипотезы о зависимости исследуемых явлений.

2. Определение зависимых и независимых (объясняющих) переменных.

3. Сбор статистических данных. Данные должны быть собраны для каждой из переменных, включенных в регрессионную модель (гипотеза).

4. Формулировка гипотезы о форме связи (простая или множественная, линейная или нелинейная).

5. Определение функции регрессии (заключается в расчете численных значений параметров уравнения регрессии)

6. Оценка точности регрессионного анализа.

7. Интерпретация полученных результатов. Полученные результаты регрессионного анализа сравниваются с предварительными гипотезами. Оценивается корректность и правдоподобие полученных результатов.

8. Предсказание неизвестных значений зависимой переменной.

При помощи регрессионного анализа возможно решение задачи прогнозирования и классификации.

Характер и форма зависимости между переменными могут образовывать следующие разновидности регрессии:

- положительная линейная регрессия (выражается в равномерном росте функции);
- положительная равноускоренно возрастающая регрессия;
- положительная равнозамедленно возрастающая регрессия;
- отрицательная линейная регрессия (выражается в равномерном падении функции);
- отрицательная равноускоренно убывающая регрессия;
- отрицательная равнозамедленно убывающая регрессия.

3. Возможности языка R для проведения корреляционного и регрессионного анализа данных

3.1. Создание набора данных

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.

Пример:

Таблица 1. Набор данных о пациентах

Порядковый номер пациента (PatientID)	**Дата поступления: месяц/день/год (AdmDate)**	**Возраст (Age)**	**Тип диабета (Diabetes)**	**Состояние (Status)**
1	10/15/2019	25	Type1	Плохое (Poor)
2	11/01/2019	34	Type2	Улучшившееся (Improved)
3	10/21/2019	28	Type1	Превосходное (Excellent)
4	10/28/2019	52	Type1	Плохое (Poor)

3.1.1. Структура данных

Необходимо различать **структуру** набора данных и **типы** данных, которые его составляют.

R работает с самыми разными структурами данных, включая

- скаляры,

– векторы,
– массивы данных,
– таблицы данных
– списки.

Они различаются типами данных, способом создания, сложностью устройства, а также способом обозначать и извлекать их отдельные элементы. Эти структуры данных схематически изображены на рис. 6

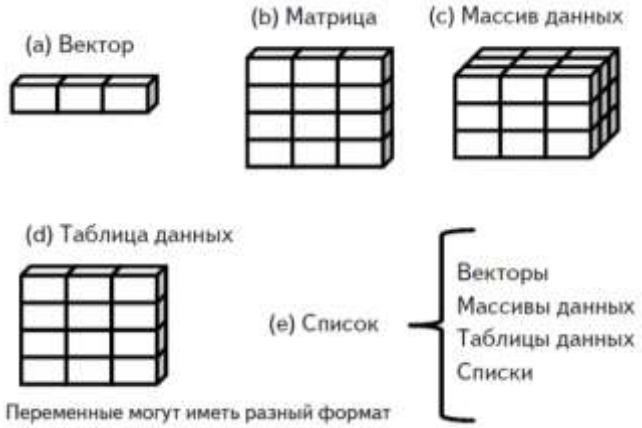

Рис. 6 Типы структуры данных в R

Существует несколько присущих только R терминов.

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 как с ним обращаться).

Таблица данных (data frame) – это тип структуры данных в R, аналогичный тому виду, в котором хранятся данные в обычных статистических программах (Столбцы – это переменные, а строки – это наблюдения. В одной таблице данных могут содержаться переменные разных типов (например, числовые и текстовые). Таблицы данных – это основной тип структуры данных.

Факторы – это номинальные или порядковые переменные. В R они хранятся и обрабатываются особым образом.

Таблица 1 будет прочитана в R как таблица данных.

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

Переменные PatientID, AdmDate и Age будут прочитаны R как числовые, а Diabetes и Status – как текстовые.

Таблица данных (data frame) – это более широко используемый по сравнению с матрицей объект, поскольку разные столбцы могут содержать разные типы данных (числовой, текстовый и т. д.). Таблица данных – это самая часто используемая структура данных в R.

Набор данных про пациентов (табл. 1) состоит из числовых и текстовых данных. Эти данные нужно представить в виде таблицы данных, а не матрицы, поскольку здесь есть данные разных типов.

Таблица данных создается при помощи функции `data.frame()`:

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

где – 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:

Создание таблицы данных (табл 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
```

Каждый столбец должен содержать данные только одного типа, при этом в одной таблице данных могут быть столбцы с данными разного типа.

Существует несколько способов обозначить элементы таблицы данных. Можно использовать индексы или можно указывать номера столбцов.

Пример:

Обозначение элементов таблицы данных

1)

```
> patientdata [1:2]
  patientID age
1         1  25
2         2  34
3         3  28
4         4  52
```

2)

```
> patientdata [c("diabetes", "status")]
  diabetes    status
1    Type1      Poor
2    Type2  Improved
3    Type1 Excellent
4    Type1      Poor
```

Знак $ используется, чтобы обозначить определенную переменную в таблице данных.

```
> patientdata$age
[1] 25 34 28 52
```

R названия строк могут быть назначены при помощи параметра row.names функции создания таблицы данных. Например, программный код

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

назначает patientID переменной, которая будет использоваться для обозначения строк при выводе данных и создании диаграмм в R.

Факторы

Переменные бывают номинальными, порядковыми или непрерывными.

Номинальные переменные – это категориальные данные, которые невозможно упорядочить. Переменная Diabetes – это пример номинальных данных. Даже если обозначить Type 1 (тип 1) единицей, а Type 2 (тип 2) – двойкой, все равно эти цифры нельзя будет сравнивать в терминах «больше – меньше».

Порядковые данные можно упорядочить, но не оценить количественно. Переменная Status – хороший пример порядковых данных. Понятно, что у больного с плохим (poor)

самочувствием дела идут не так хорошо, как у больного, чье состояние улучшилось (improved), но не ясно, насколько.

Непрерывные переменные могут принимать любое значение в пределах определенного диапазона. Их значения можно упорядочить и понять, насколько одно из них больше другого.

Возраст, выраженный в годах, является непрерывной переменной и может принимать такие значения, как 14.5 или 22.8, а также любые значения между этими двумя.

Категориальные (номинальные и порядковые) данные называются в R **факторами**. Факторы очень важны в R, поскольку они определяют, как данные будут проанализированы и графически представлены.

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.

Например есть вектор

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

Команда `diabetes <- factor(diabetes)` преобразует этот вектор в (1, 2, 1, 1) и устанавливает внутреннее соответствие 1=Type1 и 2=Type2 (присвоение числовых значений происходит в алфавитном порядке). Любой анализ, который будете проводиться с вектором `diabetes`, будет воспринимать эту переменную как номинальную и выбирать статистические методы, подходящие для этого типа данных.

При работе с векторами, которые представлены порядковыми данными, для функции `factor`() нужно добавлять параметр `ordered=TRUE`. Примененная к вектору

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

команда `status <- factor(status, ordered=TRUE)` преобразует этот вектор в вид (3, 2, 1, 3) и установит внутреннее соответствие как 1=Excellent, 2=Improved, 3=Poor. Во время любой обработки этого вектора он будет воспринят как порядковая переменная с применением соответствующих статистических методов.

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`.

Например,

```
status <- factor(status, order=TRUE,
levels=c("Poor", "Improved", "Excellent"))
```

присвоит уровни значениям вектора следующим образом: 1=Poor, 2=Improved, 3=Excellent.

Пример: Использование факторов

```
> 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)
```

Сначала вводятся данные как векторы. Затем указывается, что `diabetes` – это фактор, а `status` – это упорядоченный фактор. Потом данные объединяются в таблицу.

```
> str(patientdata)
'data.frame': 4 obs. of 4 variables:
```

```
$ patientID: num 1 2 3 4
$ age : num 25 34 28 52
$ diabetes : Factor w/ 2 levels "Type1","Type2": 1 2 1 1
$ status : Ord.factor w/ 3 levels "Excellent"<"Improved"<..: 3 2
1 3
> summary(patientdata)
patientID age diabetes status
Min. :1.00 Min. :25.00 Type1:3 Excellent:1
1st Qu.:1.75 1st Qu.:27.25 Type2:1 Improved :1
Median :2.50 Median :31.00 Poor :2
Mean :2.50 Mean :34.75
3rd Qu.:3.25 3rd Qu.:38.50
Max. :4.00 Max. :52.00
```

Функция `str(object)` выводит информацию об объекте (в нашем случае это таблица данных).

Ясно видно, что `diabetes` – это фактор, а `status` – это упорядоченный фактор; также указано, как он закодирован внутри программы.

Обратите внимание, что функция `summary()` обрабатывает переменные по-разному. Для непрерывной переменной age вычислены минимум (minimum, Min.), максимум (maximum, Max.), среднее (Mean) и квартили (first and third quartiles: 1st Qu., 3rd Qu.) (Квартили – это числа, которые делят набор данных на четыре равные части (четверти)), а для категориальных переменных diabetes и status подсчитана частота встречаемости каждого значения.

Списки

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.

Список можно создать при помощи функции `list()`:

`mylist <- list(объект 1, объект 2, …),`

где объекты – это любые структуры данных

Объектам в списке можно присваивать имена:

`mylist <- list(name1= объект 1, name2= объект 2, …).`

Пример. Создание списка

1) Создание списка:

```
>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)
```

2) Вывод списка на экран

```
> mylist
$title
[1] "My First List"
$ages
[1] 25 26 18 39
[[3]]
[,1] [,2]
[1,] 1 6
[2,] 2 7
```

```
[3,] 3 8
[4,] 4 9
[5,] 5 10
[[4]]
[1] "one" "two" "three"
```

3) Вывод на экран второго объекта списка

```
> mylist[[2]]
[1] 25 26 18 39
> mylist[["ages"]]
[[1] 25 26 18 39
```

В данном примере создаете список из четырех компонентов: тестовая строка, числовой вектор, матрица и текстовый вектор. В виде списка можно сохранять любое число объектов.

Можно обозначать элементы списка, указав их номер или название внутри двойных квадратных скобок. В данном случае и `mylist[[2]]` и `mylist[["ages"]]` обозначают один и тот же числовой вектор из четырех элементов.

Списки – это важный тип структуры данных в R по двум причинам. Во-первых, они позволяют без труда упорядочить и вызвать на экран разрозненную информацию. Во-вторых, результаты выполнения многих команд представляют собой списки.

В этом случае пользователь извлекает из таких списков нужную информацию.

3.1.2. Ввод данных

Обычно аналитики сталкиваются с данными, которые поступают из разных источников и в разных форматах. Задача состоит в том, чтобы импортировать данные в программу, проанализировать их и представить отчет о результатах. В R реализованы разные способы импорта данных.

На рис. 7 видно, что в R можно вводить данные с клавиатуры, импортировать из текстовых файлов, из Microsoft Excel и Access, из распространенных статистических программ, специализированных форматов, а также из разных систем управления базами данных.

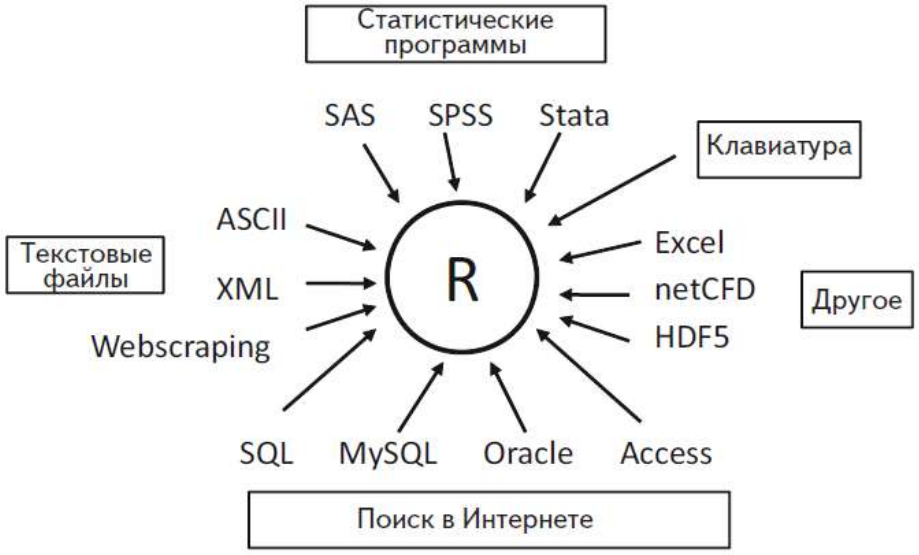

Рис. 7. Источники, из которых можно импортировать данные в R

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: необходимо создать таблицу данных с названием `mydata` с тремя переменными: `age` (возраст, числовая), `gender` (пол, текстовая) и `weight` (вес, числовая). Затем открыть текстовый редактор, внестие данные и сохранить результат.

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

Функция `edit()` работает с копией объекта. Если не присвоить результат ее работы какому-либо объекту, все изменения пропадут!

Результат работы функции `edit()` под Windows показан на рис. 8.

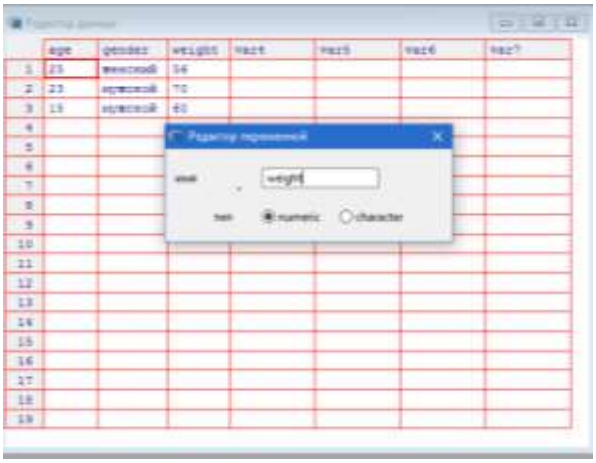

Рис. 8. Редактирование данных при помощи встроенного текстового редактора под Windows

Щелкая по названиям столбцов, можно изменить название и тип соответствующей переменной. Можно добавлять дополнительные переменные, щелкая на названия неиспользованных столбцов.

После закрытия текстового редактора, результаты сохраняются в виде выбранного объекта (в данном случае объект mydata).

Повторное введение функции `mydata <- edit(mydata)` позволяет редактировать введенные данные и добавлять новые.

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=логическое_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, программный код

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

позволяет прочесть файл с разделителями-запятыми, который называется studentgrades.csv, из текущей рабочей директории и сохранить его в виде таблицы данных с

названием grades. В этом файле названия переменных содержались в первой строке, а названия строк – в столбце с названием STUDENTID.

использование параметра `sep` позволяет импортировать файлы с любыми символами в качестве разделителей.

По умолчанию текстовые переменные преобразуются в факторы. Такое преобразование можно заблокировать разными способами. Добавление параметра `stringsAsFactors=FALSE` не позволит преобразовывать в факторы все текстовые переменные. В качестве альтернативы можно использовать параметр `colClasses` для того, чтобы указать формат (например, логический, числовой, текстовый, фактор) каждого столбца.

У функции `read.table()` есть много дополнительных параметров, при помощи которых можно контролировать импорт данных. Подробнее об этом можно прочесть, выполнив команду `help(read.table)`.

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. Под Windows для доступа к файлам Excel также можно использовать пакет RODBC. В первой строке электронной таблицы должны содержаться названия переменных (столбцов).

Прежде всего необходимо скачать и установить пакет RODBC.

```
install.packages("RODBC")
```

Теперь можно использовать следующий программный код для импорта данных:

```
library(RODBC)
channel <- odbcConnectExcel("myfile.xls")
mydataframe <- sqlFetch(channel, "mysheet")
odbcClose(channel)
```

Здесь `myfile.xls` – это файл Excel, `mysheet` – это название нужного листа из рабочей книги Excel, `channel` – это вспомогательный объект RODBC, созданный функцией `odbcConnectExcel()`, и `mydataframe` – это получившаяся таблица данных. Этот пакет можно также использовать для импорта данных из Microsoft Access. Подробности изложены в файле справки: `help(RODBC)`.

В Excel с 2007 используются файлы формата XLSX, которые фактически представляют собой сжатый набор XML-файлов. Для импорта электронных таблиц в этом формате можно использовать пакет `xlsx`.

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 2007, а `n` – число листов, которые нужно импортировать. Например, под Windows программный код

```
library(xlsx)
workbook <- "c:/myworkbook.xlsx"
mydataframe <- read.xlsx(workbook, 1)
```

импортирует первый лист книги `myworkbook.xlsx`, хранящейся на диске C:, и сохраняет его в виде таблицы данных `mydataframe`. Пакет `xlsx` может не только импортировать листы XLSX-файлов. Он также может создавать файлы этого формата и управлять ими.

Задание и порядок выполнения лабораторной работы №2.1

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 (пример данных представлен в Приложении А)
5. Выполнить ввод данных с клавиатуры,
6. Провести экспорт данных из текстового файла с разделителями,
7. Выполнить экспорт данных из Excel.

Контрольные вопросы

1. Функциональная связь.
2. Статистическая связь.
3. Корреляционная связь.
4. Корреляционный анализ.
5. Корреляционное поле.
6. Корреляционный анализ: форма зависимости.
7. Корреляционный анализ: направленность взаимосвязи.
8. Корреляционный анализ: теснота (сила) взаимосвязи.
9. Коэффициент корреляции. Его свойства.
10. Регрессионный анализ.
11. Этапы регрессионного анализа.

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. Оффициальный 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. Мастицкий С.Э., Шитиков В.К. (2014) Статистический анализ и визуализация данных с помощью R. – Электронная книга. Режим доступа: http://r-analytics.blogspot.com

Требования к содержанию и оформлению отчетов

Отчеты по лабораторным работам оформляются согласно правилам оформления принятым на кафедре, ГОСТам и ЕСКД.

Основные правила по оформлению отчетной документации:

Параметры страницы: А4 (21×29,7), ориентация – книжная (допускается использовать альбомную ориентацию страницы для выполнения схем и таблиц).

Поля: левое – 25 (30) мм, верхнее – 20 мм, нижнее – 20 мм, правое – 10 мм.

Нумерация страницы – вверху, по центру. Нумерация ведется с титульного листа, номер на титульном листе не ставиться.

Шрифт Times New Roman, кегль 14, интервал – полуторный.

Заголовки разделов: абзацный отступ – 0, выравнивание по центру, шрифт – жирный, буквы прописные, нумерация – арабскими цифрами, точка в конце номера и названия раздела не ставиться.

Заголовки подразделов (допускается три уровня, например, 1.1., 1.1.1.): абзацный отступ – 1,25 см, выравнивание по ширине, шрифт – жирный, точка в конце названия подраздела не ставиться.

Основной текст: абзацный отступ – 1,25 см, выравнивание по ширине, шрифт – обычный.

Нумерация рисунков и таблиц – сквозная внутри раздела (например, в разделе 1 – рисунок 1.1, рисунок 1.2 и т.д., или таблица 1.1, таблица 1.2 и т.д.).

Рисунки помещаются после упоминания их в тексте и имеют подпись, размещаемую под рисунком без абзацного отступа и имеющую выравнивание по центру и точку на конце названия (например, Рисунок 1.1 – Название.).

Таблицы размещаются после ссылки на них в тексте. Название приводится над таблицей, без абзацного отступа с выравниванием по левому краю, без точки на конце названия (например, (Таблица 2.2 – Название).

Допускается выносить рисунки и таблицы в Приложения. В этом случае ссылка должна содержать номер приложения (например: рисунок А.1 Приложения А или таблица Б.1 Приложения Б).

Основная часть должна содержать ссылки на используемую литературу или информационные источники, список которых приводится после раздела Выводы и перед Приложениями. Ссылка заключается в квадратные скобки (например – [1], [5,7], [3-6].

Приложения обозначаются русскими заглавными буквами в порядке их следования (Приложение А, Приложение Б). Слово «Приложение…» выравнивается по центру без абзацного отступа и имеет жирный шрифт, прописные буквы. Название приложение располагается на следующей строке, без абзацного отступа, выравнивание по центру, шрифт – жирный, первая буква прописная, остальные – строчные.

По завершению изучения курса у студента должен быть сформировать набор отчетов (Приложение №1), сведенных в единый документ и имеющий единый титульный лист (Приложение №2), на котором отражаются результаты прохождения этапов изучения дисциплины.

Каждый раздел этого документа является отчетом по выполнению соответствующей лабораторной работы (обязательные разделы и правила выполнения отчетов представлены в Приложении 1).

Сформированный документ, с отметками о выполнении всех лабораторных работ обязателен для представления на итоговом контроле и является подтверждением о допуске к итоговому контролю.

К отчету прилагается папка с файлами – результатами выполнения лабораторной работы (данная папка должна так же находится на сетевом диске в папке проектов изучаемой дисциплины), название папки ГПА_фамилия.

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, отводиться 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 систему 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 ошибку 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 их 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.

Приложение А

Таблица А.1.

Индексы характеристик качества жизни

	Индекс ожидаемой продол–ти при рождении	Индекс уровня образования	Индекс уровня бедности	Индекс уровня безработицы	Индекс реального ВВП на душу населения	Индекс общест – го развития
Российская Федерация	0,70	0,86	0,76	0,87	0,144	0,667
Тюменская область	0,71	0,91	0,85	0,86	0,555	0,777
Самарская область	0,71	0,88	0,82	0,91	0,193	0,703
Мурманская область	0,73	0,92	0,82	0,79	0,208	0,694
Республика Татарстан	0,73	0,85	0,80	0,89	0,157	0,685
Республика Коми	0,69	0,90	0,79	0,82	0,216	0,683
Республика Якутия	0,67	0,91	0,68	0,86	0,296	0,683
Магаданская область	0,69	0,96	0,66	0,82	0,245	0,675
Хабаровский край	0,66	0,91	0,75	0,88	0,172	0,674
Пермская область	0,69	0,84	0,79	0,87	0,164	0,671
Белгородская область	0,74	0,81	0,80	0,89	0,115	0,671
Липецкая область	0,72	0,81	0,81	0,89	0,119	0,670
Московская область	0,70	0,88	0,73	0,90	0,134	0,669
Камчатская область	0,66	0,94	0,67	0,82	0,250	0,668
Нижегородская область	0,70	0,82	0,79	0,91	0,125	0,669
Ульяновская область	0,72	0,82	0,81	0,89	0,097	0,667
Красноярский край	0,65	0,86	0,77	0,84	0,205	0,665
Ярославская область	0,70	0,83	0,76	0,89	0,137	0,663
Томская область	0,69	0,86	0,73	0,85	0,174	0,661
Иркутская область	0,65	0,88	0,73	0,86	0,167	0,657
Вологодская область	0,69	0,82	0,75	0,87	0,159	0,658
Кемеровская область	0,66	0,84	0,77	0,88	0,138	0,658
Ростовская область	0,71	0,86	0,79	0,84	0,078	0,656
Тульская область	0,68	0,83	0,79	0,88	0,093	0,655
Респ. Башкортостан	0,71	0,84	0,72	0,87	0,137	0,655
Воронежская область	0,73	0,81	0,75	0,91	0,083	0,657
Челябинская область	0,71	0,85	0,73	0,88	0,112	0,656
Свердловская область	0,69	0,86	0,67	0,90	0,152	0,654
Курская область	0,70	0,81	0,74	0,90	0,110	0,652
Калужская область	0,69	0,85	0,74	0,90	0,087	0,653
Омская область	0,72	0,83	0,75	0,85	0,118	0,654
Республика Карелия	0,68	0,85	0,77	0,83	0,128	0,652
Орловская область	0,71	0,83	0,74	0,87	0,098	0,650
Оренбургская область	0,70	0,84	0,73	0,87	0,113	0,651
Новгородская область	0,67	0,80	0,81	0,85	0,111	0,648
Рязанская область	0,70	0,82	0,70	0,93	0,095	0,649
Краснодарский край	0,71	0,84	0,75	0,84	0,092	0,646
Удмуртская Республика	0,71	0,86	0,68	0,87	0,107	0,645
Владимирская область	0,69	0,84	0,72	0,88	0,085	0,643
Волгоградская область	0,71	0,86	0,69	0,85	0,100	0,642
Саратовская область	0,71	0,86	0,70	0,84	0,091	0,640
Приморский край	0,68	0,89	0,66	0,85	0,125	0,641
Сахалинская область	0,65	0,91	0,62	0,83	0,188	0,640

Костромская область	0,68	0,80	0,73	0,89	0,097	0,639
Смоленская область	0,68	0,82	0,77	0,84	0,092	0,640
Тамбовская область	0,71	0,79	0,74	0,87	0,070	0,636
Тверская область	0,67	0,83	0,69	0,89	0,095	0,635
Калининградская область	0,68	0,88	0,70	0,83	0,079	0,634
Амурская область	0,67	0,87	0,67	0,83	0,126	0,633
Архангельская область	0,68	0,87	0,62	0,85	0,134	0,631
Ленинградская область	0,69	0,84	0,65	0,85	0,110	0,628
Брянская область	0,70	0,80	0,73	0,84	0,070	0,628
Астраханская область	0,69	0,85	0,66	0,84	0,091	0,626
Новосибирская область	0,71	0,84	0,58	0,86	0,112	0,620
Ставропольский край	0,72	0,84	0,62	0,84	0,092	0,622