МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ИМЕНИ Н.Э. БАУМАНА

Факультет "Информатика и системы управления"

Кафедра "Системы обработки информации и управления"

Лабораторная работа № X

**"Знакомство с языком программирования R для анализа данных "**

Москва — 2022

**Оглавление**

[I. Загрузка данных в R 3](#_Toc121005534)

[II. Визуализация данных c помощью R 6](#_Toc121005535)

[III. Прогнозирование временных рядов в R 7](#_Toc121005536)

[IV. Классификация и кластеризация в R 8](#_Toc121005537)

[V. Нейронные сети в R 11](#_Toc121005538)

**Немного о языке программирования R**

**R** — язык программирования, созданный специально для статистического анализа данных. У языка R чёткая сфера применения — статистические вычисления, анализ данных и машинное обучение. R — не только язык для анализа данных, но и целая рабочая среда, куда уже встроены готовые методы статистического анализа и инструменты для визуализации.

Обычно на R не пишут классические сервисы с интерфейсом, кнопками и текстом, к которым все привыкли. Чаще всего программы на R представляют собой код, выполнение которого выводит какой-то определённый результат или графики. С помощью языка R можно:

● **Очистить и обработать данные**. Например, для обработки массива информации о популярности разных товаров на R можно написать программу, которая сгруппирует данные о покупках по товарам, удалит дубли и подготовит таблицу для дальнейшей аналитики.

● **Провести статистический тест**. Посчитать среднюю продолжительность и увидеть, есть ли статистически значимая разница между несколькими показателями.

● **Объединять данные из разных таблиц**. Взять таблицы разных форматов, собрать из них данные и обрабатывать все как единый файл.

● **Отрисовать интерактивный график**. Распределить данные, отрегулировать параметры.

● **Анализировать регрессионные модели**. Выявить отношения между переменными — например, как доход магазина зависит от разных факторов.

● **Провести другую математическую операцию**. Объединить многомерные массивы, спрогнозировать величину, распознать текст. Для большинства задач есть готовые библиотеки, либо можно написать собственный код, используя широкий математический инструментарий.

Скачать среду разработки, посмотреть документацию и т.п. можно на официальном сайте [r-project.org](https://www.r-project.org/)

**Подготовка к выполнению лабораторной работы**

1. Установите программное обеспечение [R](https://cloud.r-project.org/).
2. Если вы хотите работать в ПО 20 века, то выполните пункт 3 и перейдите к пункту 7, если мы идете в ногу со временем, то перейдите сразу к пункту 4
3. Откройте программу R. Перед вами появится консоль. Нажмите на «Файл -> Открыть скрипт». Выберите «required\_packages.R»
4. Установите VSCode. Он потребуется для более удобного выполнения лабораторной работы.
5. Запустите VSCode и откройте папку «lab\_R». Для этого, в верхней панели выберите «Файл -> Открыть папку…». В левой части программы должно появиться дерево папки. Если этого не произошло, нажмите «ctrl+B»
6. Так же вам потребуются два расширения для VSCode. Вкладку с магазином расширений можно найти на боковой панели. Нам понадобятся: «R» и «Excel Viewer». Далее нужно задать «Excel Viewer» как редактор по умолчанию для расширений «.csv» и «.xlsx». Для этого нажмите ПКМ по любому файлу «.csv» в дереве проекта, далее «открыть с помощью… –> настройка редактора по умолчанию для “\*.csv” –> Excel Viewer». Проделайте те же действия для файла «.xlsx»
7. Откройте файл «required\_packages.R». В данном файле находятся все необходимые дополнительные пакеты для лабораторной работы. Нажмите на кнопку запуска в правом верхнем углу (Или третью кнопку слева, если вы решили вернуться во времена windows XP🙃). Сначала вас спросят про secure CRAN mirrors. Выберите «0-cloud». После, если у вас windows, вас спросят два вопроса о расположении папки с пакетами. Начнется установка.

**Ход выполнения лабораторной работы**

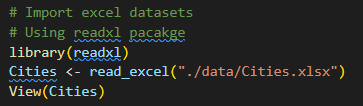
1. **Загрузка данных в R**

Откройте папку «part\_1», файл «main\_for\_students.R».

В этой части лабораторной мы будем изучать загрузку данных в программу из различных источников, а также поработаем с data frame.

1. **Классический вариант загрузки данных (.csv and .txt файлы)**
   1. Первым делом нужно указать рабочее пространство. В самом файле есть подсказки
   2. В файле уже указаны некоторые способы загрузки разных файлов из локального хранилища

****

****

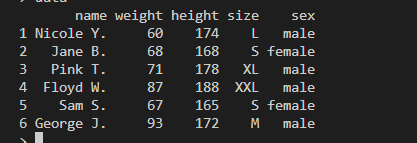
* 1. Ваша задача импортировать файл «Population.txt», записать в переменную и отобразить его на экране. Используйте для этого утилиту read.table и view. При импорте файла нужно учесть, что бы первая строка стала заголовком таблицы, а также нужно указать разделитель ‘\t’. Напишите код на ~34 строке. Руководство по каждой утилите можно найти, наведясь на нее и нажав на соответствующую ссылку.



В справочнике можно спокойно прочитать о работе утилиты и посмотреть примеры использования. Не обязательно прописывать все аргументы для команды, указанные в справочнике

* 1. Для выполнения, последовательно скопируйте команды в терминал и выполните ее

1. **Загрузка данных из других пакетов R**
   1. Будем загружать данные из пакета Mcomp. Измените dataset c M1 на M3
2. **Загрузка данных из интернета**
   1. Теперь загрузим данные из интернет-источника.
   2. Для этого нам понадобиться библиотека RCurl. Импортируете ее перед командой getURL
   3. Вы можете выбрать любой другой [dataset](https://people.sc.fsu.edu/~jburkardt/data/csv/csv.html)
3. **Получение свойств данных**
   1. Для получения свойства загруженных данных воспользуемся следующими командами:
      * head(x) — посмотреть начальные строки объекта
      * tail(x) — посмотреть последние строки объекта
      * dput(x) — получить представление объекта в R-синтаксисе
      * summary(object) — общая информация об объекте
      * str(object) — внутренняя структура объекта
      * dim(x) — размерности объекта
      * dimnames(x) — имена размерностей объекта
      * names(x) — имена объекта
      * length(x) — число элементов
   2. Попробуйте выполнить все эти команды.
4. **Работа с Data Frame**
   1. Сначала создадим таблицу с помощью data.frame из векторов, которые будут обозначать наши столбцы. Каждую строку нужно последовательно выполнить в терминале. Должна получиться таблица:



* 1. Вам необходимо выполнить ряд заданий по анализу данной таблицы

1. **Визуализация данных c помощью R**

Откройте папку «part\_2», файл «main\_for\_students.R».

В этой части лабораторной мы научимся строить диаграммы с помощью сторонней библиотеки «ggplot».

Постройте следующие диаграммы, последовательно выполняя команды из файла:

* + - Scatter Plot (диаграмма рассеяния)
    - Line Graph (Линейный график )
    - Bar Graph (столбчатая диаграмма)
    - Histogram (гистограмма)
    - Box Plot (ящики с усами)

Для Box Plot самостоятельно постройте диаграммы с помощью «ggplot» по аналогии с предыдущими диаграммами

1. **Прогнозирование временных рядов в R**

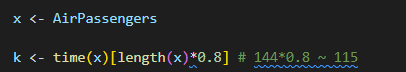
Откройте папку «part\_3», файл «main\_for\_students.R».

В этой части лабораторной мы посмотрим на прогнозирование временных рядов с помощью библиотеки «forecast».

1. Подключим библиотеки «forecast» и «ggplot»



1. Используем встроенный dataset AirPassengers, переименуем в X для удобства использования и найдем дату, которая разделит наш dataset в пропорции 80/20



1. Создадим обучающее и тестовое множества в пропорции 80/20



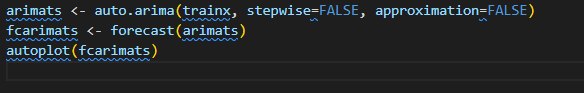
1. Используем метод Benchmark forecasting - naive для прогнозирования и визуализируем с помощью autoplot()



1. Теперь попробуем метод Exponential smoothing и также визуализируем его



1. Последний метод – Arima- очень мощный статистический метод для прогнозирования временных рядов. Auto.arima выполняется не моментально, так что не волнуйтесь, если у вас перестанет работать терминал и просто подождите. После построения временного ряда вы увидите существенную разницу по сравнению с другими методами.

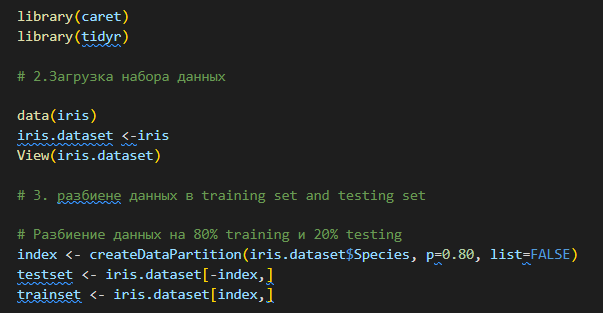


1. **Классификация и кластеризация в R**

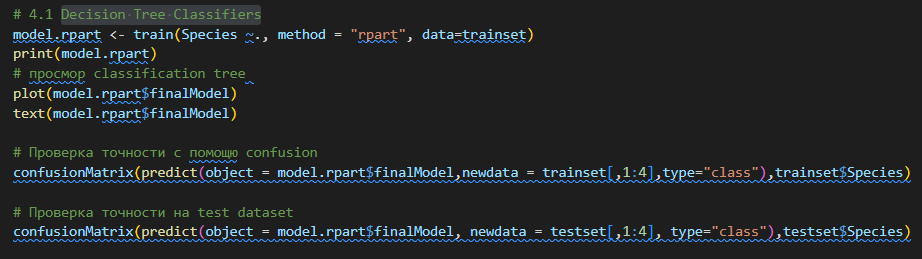
Откройте папку «part\_4», файл «main\_for\_students.R».

В этой части лабораторной мы изучим инструменты для классификации и кластеризации dataset`ов

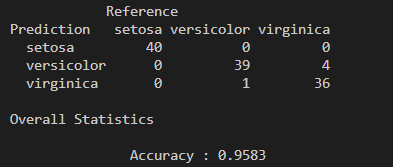
1. Подключим библиотеки, загрузим dataset Iris и разобьем его на тренировочные и проверочные данные



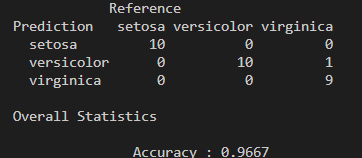
1. Проведем классификацию с помощью «Decision Tree Classifiers» и проверим точность классификации на тренировочных и проверочных данных

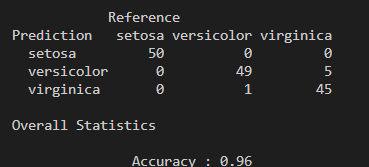


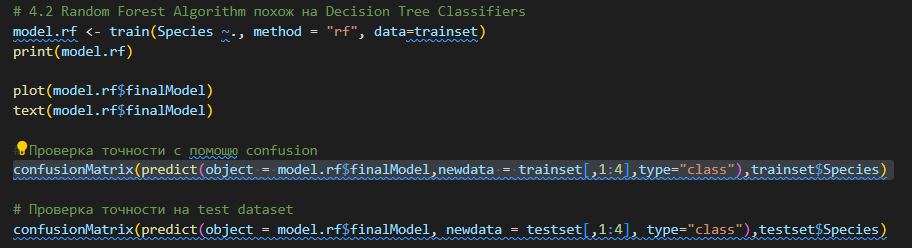
Получили, что наша модель классифицирует данные с точностью 95.8% на тренировочных данных:



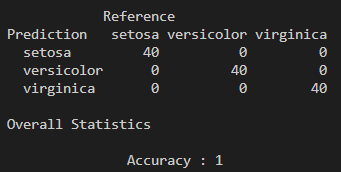
С точностью 96.7% на проверочном множестве:



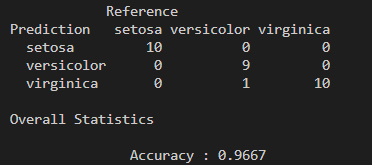
И, с точностью 96% на всем множестве данных: 

1. Теперь проведем классификацию с помощью «Random Forest Algorithm». Этот алгоритм более точно производит классификацию, что мы сейчас и пок

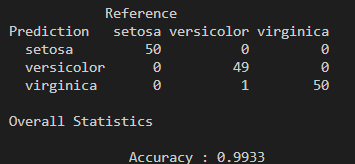
Тренировочное множество – 100%:



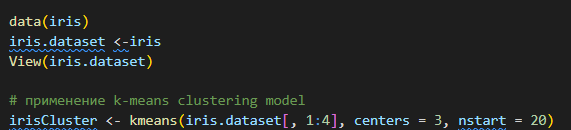
Проверочное множество – 96.7%:



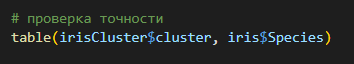
Все множество – 99.3%:

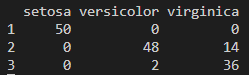


1. Теперь проведем кластеризацию того же dataset`а iris. Повторно загрузим dataset iris и применим k-means модель кластеризации:



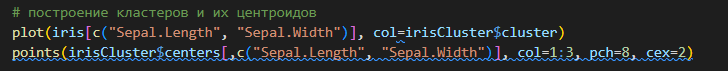
1. Посмотрим на таблицу соответствия кластеров и цветов:





Видно, что во второй кластер попало много цветов virginica, что говорит о том, что цветы versicolor и virginica достаточно похожи.

1. В заключении визуализируем нашу кластеризацию

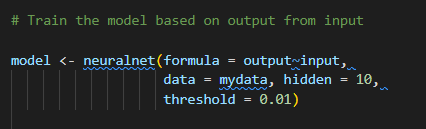


1. **Нейронные сети в R**

Откройте папку «part\_5», файл «main\_for\_students.R».

Последним заданием будет написание нейронной сети для операции сложения.

1. Сначала нужно добавить обучающие данные с помощью data.frame и назвать mydata. Вектора назвать «input\_1», «input\_2» и «output». Вам дан пример с данными для возведения во вторую степень. Вам нужно дополнить входные данные, изменить параметры функции neuralink (п.2) и функцию вывода.
2. После создания обучающих данных, нужно обучить нашу НС



1. Теперь визуализируйте результат с помощью plot()
2. Выведем результат в виде наглядной таблицы

