**УТВЕРЖДЕНО**
решением Ученого совета ФМИАТ
от 25 мая 2024 г., протокол № 5/24

Председатель _____ /Вояков М.А./
*расшифровка подписи*
21 мая 2024 г.

# РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

| | |
|---|---|
| Дисциплина | *Python для анализа данных* |
| Факультет | Математики, информационных и авиационных технологий |
| Кафедра | Прикладной математики |
| Курс | 3 |

Направление (специальность) 01.03.02. Прикладная математика и информатика
*код направления (специальности), полное наименование*

Направленность (профиль/специализация) Имитационное моделирование и анализ данных
*полное наименование*

Форма обучения очная

Дата введения в учебный процесс УлГУ: 1 сентября 2024 г.

Программа актуализирована на заседании кафедры: протокол №_____ от _____ 20_____г.
Программа актуализирована на заседании кафедры: протокол №_____ от _____ 20_____г.
Программа актуализирована на заседании кафедры: протокол №_____ от _____ 20_____г.

Сведения о разработчиках:

| ФИО | Кафедра | Должность,<br>ученая степень, звание |
|---|---|---|
| Савинов Ю.Г. | ПМ | Доцент, к.ф.м.н., доцент |

| СОГЛАСОВАНО |
|---|
| Заведующий кафедрой<br>/_____/ Бутов А.А.<br>*(Подпись)* *(ФИО)*<br>«21» мая 2024 г. |

## 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ:

В дисциплине изучаются библиотеки Python, необходимые для обработки и визуализации данных.

**Цель** дисциплины - формирование у студентов навыков, соответствующих видам профессиональной деятельности, необходимых для решения профессиональных задач.

**Задача** дисциплины – освоение обучающимися навыков работы с большими данными, их обработкой и визуализацией на современном языке программирования на примере Python.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП:

Дисциплина «Python для анализа данных» изучается в 6 семестре и относится к обязательной части дисциплин блока Б1.О направления подготовки 01.03.02. Прикладная математика и информатика. Дисциплина формирует практические навыки использования в профессиональной деятельности современных концепций и методов программирования.

Данная дисциплина базируется на входных знаниях, умениях, навыках и компетенциях студента, полученных им при изучении предшествующих учебных дисциплин: Программирование на языке Python, Базы данных, Алгебра и геометрия, Численные методы, Теория вероятностей, Статистика для анализа данных.

Результаты освоения дисциплины будут необходимы для дальнейшего процесса обучения в рамках поэтапного формирования компетенций при изучении последующих дисциплин (указаны в ФОС, пункт 1), а также для прохождения всех видов практик и государственной итоговой аттестации.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

| Код и наименование реализуемой компетенции | Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций |
|---|---|
| ОПК-2 – способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач | Знать: синтаксис и методы библиотек языка Python для анализа данных. Области их применения. Недостатки и ограничения библиотек Pandas, Numpy, Matplotlib. Возможности интеграции с другими языками программирования.<br>Уметь: разрабатывать эффективные модели анализа данных на языке Python.<br>Владеть (демонстрировать навыки и опыт деятельности): библиотеками Pandas, Numpy, Matplotlib языка Python. |
| ОПК-3 – способен применять и модифицировать математические модели для решения задач в области профессиональной деятельности | Знать: классификацию инструментов анализа данных, способы визуализации данных.<br>Уметь: визуализировать числовые и нечисловые данные, строить гистограммы, графики и диаграммы по различным данным.<br>Владеть: библиотеками визуализации данных Matplotlib, Seaborn и др. |

| | |
|---|---|
| ПК-7 – способен формировать суждения о значении и последствиях своей профессиональной деятельности с учетом социальных, профессиональных и этических позиций | Знать: основные возможности по нахождению и сбору данных.<br>Уметь: извлекать, собирать данные разных форматов, с веб-страниц с помощью средств и библиотек языка Python.<br>Владеть: навыками написания скриптов для сбора данных с веб-сайтов. |
| ПК-8 – способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения | Знать: основные виды представления данных.<br>Уметь: проводить предобработку данных (чистку, шкалирование, преобразование).<br>Владеть: методами предобработки данных. |

## 4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

**4.1. Объем дисциплины в зачетных единицах (всего)** __3_____

**4.2. Объем дисциплины по видам учебной работы (в часах)**

| Вид учебной работы | Количество часов (форма обучения) очная | |
|---|---|---|
| | Всего по плану | В т.ч. по семестрам |
| | | 6 |
| Контактная работа обучающихся с преподавателем в соответствии с УП | 54 | 54 |
| Аудиторные занятия: | | |
| • лекции | 18 | 18/18 |
| • семинары и практические занятия | 18 | 18/18 |
| • лабораторные работы, практикумы | 18 | 18/18 |
| Самостоятельная работа | 54 | 54 |
| Форма текущего контроля знаний и контроля самостоятельной работы: тестирование, контр. работа, коллоквиум, реферат и др.(не менее 2 видов) | | Выполнение лабораторных заданий, решение задач |
| Курсовая работа | | |
| Виды промежуточной аттестации (экзамен, зачет) | | зачет |
| Всего часов по дисциплине | 108 | 108 |

В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий в таблице через слеш указывается количество часов работы ЛЛС с обучающимися для проведения занятий в дистанционном формате с применением электронного обучения.

**4.3. Содержание дисциплины. Распределение часов по темам и видам учебной работы:**

Форма обучения __очная_____

| Название и разделов и тем | Всего | Виды учебных занятий | | | | | Форма текущего контроля знаний |
|---|---|---|---|---|---|---|---|
| | | Аудиторные занятия | | | Занятия в интерактивной форме | Самостоятельная работа | |
| | | лекции | Практические занятия, семинары | Лабораторные работы, практикумы | | | |
| **Тема 1. Введение в программирование на языке Python**. Синтаксис языка. Базовые типы данных: числа, строки, списки, кортежи, словари, множества. Функции. Итераторы и генераторы. Классы и объекты. Декораторы. Ввод-вывод. Обработка исключений. IPython, Jupyter Notebook. Подключение библиотек, создание собственных модулей. Элементы функционального программирования (lambda, map, zip, reduce, filter). Чтение и запись данных в текстовом формате. | 18 | 6 | 2 | 0 | 0 | 12 | Решение задач |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Тема 2. Обработка данных. Массивы и векторные вычисления.** Основы NumPy: многомерные массивы и векторные вычисления. Индексирование и вырезание. Универсальные функции: быстрые поэлементные операции над массивами. Обработка данных с применением массивов. Методы булевых массивов. Сортировка. Устранение дубликатов и другие теоретико-множественные операции. Файловый ввод-вывод массивов. Линейная алгебра. Генерация случайных чисел. | 30 | 4 | 4 | 6 | 0 | 14 | Решение задач. Лабораторная работа 1 |
| **Тема 3. Построение графиков и визуализация данных.** Визуализация данных в Python. Обзор библиотек: matplotlib, seaborn, plotly. Базовые типы визуализаций: графики, столбчатые диаграммы, гистограммы, точечные диаграммы (scatter plots), ящики с усами. Комбинирование различных графических элементов. Построение интерактивных диаграмм с помощью plotly. | 30 | 4 | 6 | 6 | 0 | 14 | Решение задач. Лабораторная работа 2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Тема 4. Специализированные библиотеки Python для анализа данных.** Введение в анализ табличных данных в Python. Пакет pandas. Объекты Series (последовательность) и DataFrame (таблица). Чтение-запись данных в различных форматах. Запросы к таблицам: выборка строк/столбцов по заданным критериям. Переформатирование данных: очистка, преобразование, слияние, изменение формы. Фильтрация отсутствующих данных. Агрегирование данных и групповые операции. Основы работы с временными рядами. | 30 | 4 | 6 | 6 | 0 | 14 | Решение задач. Лабораторная работа 3 |
| Итого | 108 | 18 | 18 | 18 | 0 | 54 | |

## 5. СОДЕРЖАНИЕ ДИСЦИЛИНЫ

**Тема 1. Введение в программирование на языке Python**. Синтаксис языка. Базовые типы данных: числа, строки, списки, кортежи, словари, множества. Функции. Итераторы и генераторы. Классы и объекты. Декораторы. Ввод-вывод. Обработка исключений. IPython, Jupyter Notebook. Подключение библиотек, создание собственных модулей. Элементы функционального программирования (lambda, map, zip, reduce, filter). Чтение и запись данных в текстовом формате.

**Тема 2. Обработка данных. Массивы и векторные вычисления.** Основы NumPy: многомерные массивы и векторные вычисления. Индексирование и вырезание. Универсальные функции: быстрые поэлементные операции над массивами. Обработка данных с применением массивов. Методы булевых массивов. Сортировка. Устранение дубликатов и другие теоретико-множественные операции. Файловый ввод-вывод массивов. Линейная алгебра. Генерация случайных чисел.

**Тема 3. Построение графиков и визуализация данных.** Визуализация данных в Python. Обзор библиотек: matplotlib, seaborn, plotly. Базовые типы визуализаций: графики, столбчатые диаграммы, гистограммы, точечные диаграммы (scatter plots), ящики с усами. Комбинирование различных графических элементов. Построение интерактивных диаграмм с помощью plotly.

**Тема 4. Специализированные библиотеки Python для анализа данных.** Введение в анализ табличных данных в Python. Пакет pandas. Объекты Series (последовательность) и DataFrame (таблица). Чтение-запись данных в различных форматах. Запросы к таблицам: выборка строк/столбцов по заданным критериям. Переформатирование данных: очистка,

преобразование, слияние, изменение формы. Фильтрация отсутствующих данных. Агрегирование данных и групповые операции. Основы работы с временными рядами.

## 6. ТЕМЫ ПРАКТИЧЕСКИХ И СЕМИНАРСКИХ ЗАНЯТИЙ

Данный вид работы не предусмотрен УП.

## 7. ЛАБОРАТОРНЫЕ РАБОТЫ, ПРАКТИКУМЫ

**Тема 2.**

*Лабораторная работа 1 «ПЕРВИЧНЫЙ АНАЛИЗ ДАННЫХ»*

Цель работы: изучение программных средств для организации рабочего места специалиста по анализу данных и машинному обучению.

Основные задачи:

– получение программного доступа к данным, содержащимся в источниках различного типа;

– выполнение предварительного анализа данных и получение обобщенных характеристик наборов данных;

– исследование простых методов визуализации данных;

– изучение основных библиотек Python для работы с данными.

Индивидуальное задание

1. Подберите набор данных и согласуйте свой выбор с преподавателем.

2. Проведите первичный анализ данных. В результате анализа данных студент должен предоставить следующую информацию о наборе данных:

- Описание набора данных, пояснения, позволяющие лучше понять природу данных. Назначение набора данных и возможные модели, которые можно построить на основе данного набора данных (практические задачи, решаемые с использованием данного обучающего набора данных). Описание каждого признака и его тип.

- Форма набора данных: количество элементов набора, количество признаков, количество пропущенных значений, среднее значение отдельных признаков, максимальные и минимальные значения отдельных признаков и прочие показатели. Предположения, которые можно сделать, проведя первичный анализ.

- Графические представления, позволяющие судить о неоднородности исследуемого набора данных. Построение графиков желательно произвести по нескольким проекциям.

Содержание отчета и его форма

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы сдается преподавателю в бумажной или электронной форме (по согласованию).

Методические указания

Необходимо организовать подготовку данных для построения моделей (классификации, кластеризации и др.).

1. В качестве данных выбрать набор данных из базы https://archive.ics.uci.edu/datasets. При выборе набора данных необходимо согласовать свой выбор с другими студентами группы и преподавателем, так как работа над одинаковыми наборами данных недопустима.

Для примера разобран классический набор данных об ирисах Фишера. Необходимо скачать набор данных из репозитория Center for Machine Learning and Intelligent Systems (необходим только один текстовый файл с данными измерений):
http://archive.ics.uci.edu/dataset/53/iris
Это, пожалуй, самый известный набор данных, с которого многие начинают исследование алгоритмов машинного обучения. Данный набор данных предназначен для построения модели классификации. Данные о 150 экземплярах ириса, по 50 экземпляров из трёх видов – Ирис щетинистый (Iris setosa), Ирис виргинский (Iris virginica) и Ирис разноцветный (Iris versicolor). Для каждого экземпляра измерялись четыре характеристики (в сантиметрах):
1) длина наружной доли околоцветника (sepal length);
2) ширина наружной доли околоцветника (sepal width);
3) длина внутренней доли околоцветника (petal length);
4) ширина внутренней доли околоцветника (petal width).

На основании этого набора данных требуется построить правило классификации, определяющее вид растения по данным измерений. Это задача многоклассовой классификации, так как имеется три класса – три вида ириса.

2. Подключение библиотеки NumPy и загрузка данных.

```
import numpy as np
data_path = "../datasets/iris/iris.data"
data = np.genfromtxt(data_path, delimiter=",")
print(data)
```

```
[[5.1 3.5 1.4 0.2 nan]
 [4.9 3.  1.4 0.2 nan]
 [4.7 3.2 1.3 0.2 nan]
 [4.6 3.1 1.5 0.2 nan]
 [5.  3.6 1.4 0.2 nan]
```

Метод genfromtxt() возвращает массив numpy (тип numpy.ndarray). Чтобы получить первые 10 строк таблицы можно использовать срез data[:10], также мы выводим тип (type) переменной и форму (shape).

```
print ( "Data type : ", type(data) )
print ( "Data shape : ", data.shape )
print ( data[:10] )
```

```
Data type :  <class 'numpy.ndarray'>
Data shape :  (150, 5)
[[ 5.1  3.5  1.4  0.2  nan]
 [ 4.9  3.   1.4  0.2  nan]
 [ 4.7  3.2  1.3  0.2  nan]
 [ 4.6  3.1  1.5  0.2  nan]
 [ 5.   3.6  1.4  0.2  nan]
 [ 5.4  3.9  1.7  0.4  nan]
 [ 4.6  3.4  1.4  0.3  nan]
 [ 5.   3.4  1.5  0.2  nan]
 [ 4.4  2.9  1.4  0.2  nan]
 [ 4.9  3.1  1.5  0.1  nan]]
```

Из представленного фрагмента видно, что data – это двумерный массив размером 150x5, или можно сказать, что это одномерный массив, каждый элемент которого также одномерный массив размером 5 элементов. Следует обратить внимание, что пятый столбец содержит неопределенные значения NaN (не число). Чтобы получить значения пятого столбца можно указать типы столбцов при загрузке данных.

```
dt = np.dtype("f8, f8, f8, f8, U30")
data2 = np.genfromtxt("iris.data", delimiter=",", dtype=dt)
print(data2.shape)
print(type(data2))
print(type(data2[0]))
print(type(data2[0][4]))
print(data2[:10])
```

```
(150,)
<class 'numpy.ndarray'>
<class 'numpy.void'>
<class 'numpy.str_'>
[(5.1, 3.5, 1.4, 0.2, 'Iris-setosa') (4.9, 3.0, 1.4, 0.2, 'Iris-setosa')
 (4.7, 3.2, 1.3, 0.2, 'Iris-setosa') (4.6, 3.1, 1.5, 0.2, 'Iris-setosa')
 (5.0, 3.6, 1.4, 0.2, 'Iris-setosa') (5.4, 3.9, 1.7, 0.4, 'Iris-setosa')
 (4.6, 3.4, 1.4, 0.3, 'Iris-setosa') (5.0, 3.4, 1.5, 0.2, 'Iris-setosa')
 (4.4, 2.9, 1.4, 0.2, 'Iris-setosa') (4.9, 3.1, 1.5, 0.1, 'Iris-setosa')]
```

Также можно использовать функцию  Pandas read_table(), задав самостоятельно названия. столбцов.

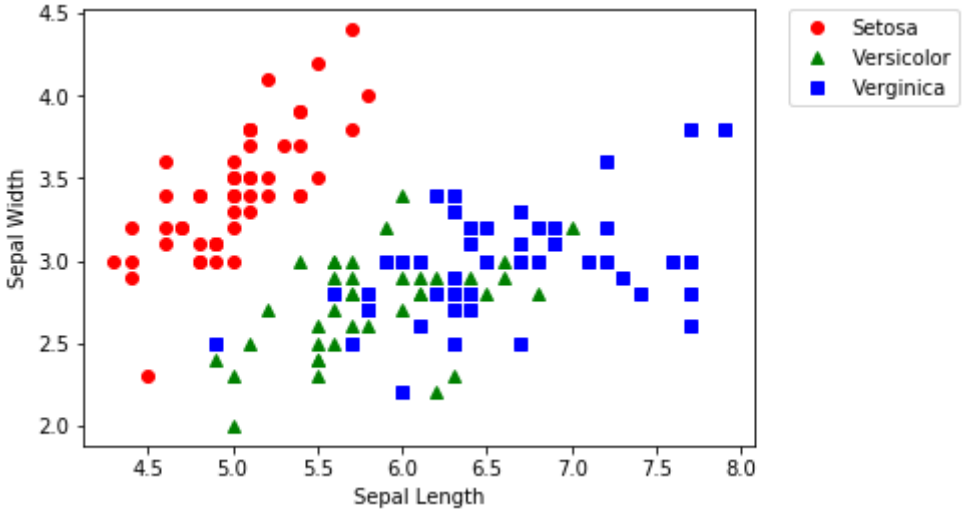
In [19]:
```
import pandas as pd
import numpy as np

data_source = 'iris.data'
d = pd.read_table(data_source, delimiter=',',
                  header=None,
                  names=['sepal_length','sepal_width',
                         'petal_length','petal_width','answer'])
d.head()
```

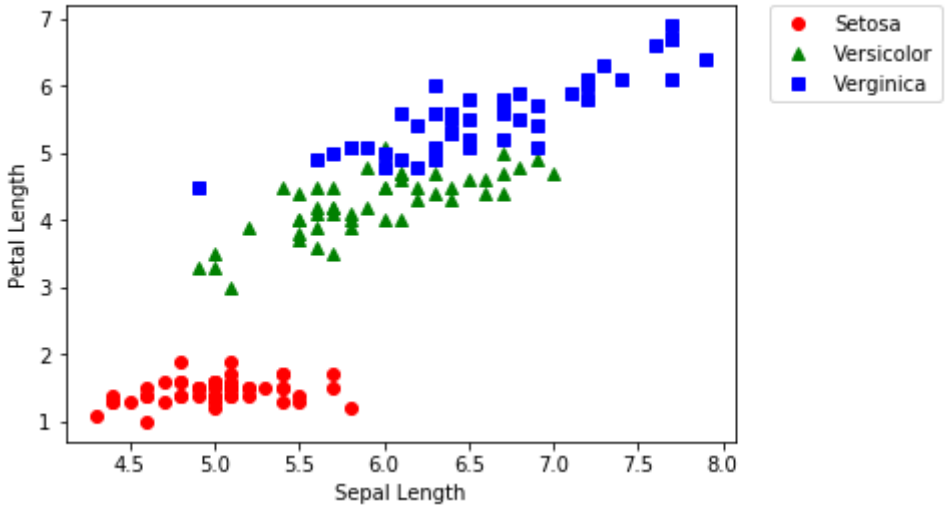
Out[19]:

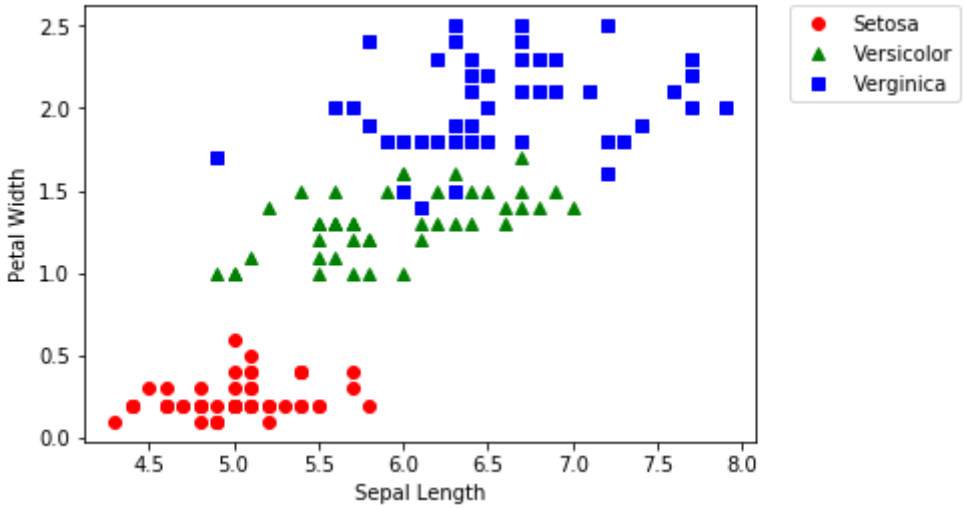
| | sepal_length | sepal_width | petal_length | petal_width | answer |
| --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

## 3. Построение графиков с использованием Matplotlib

```
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

# Данные из отдельных столбцов
sepal_length = [] # Sepal Length
sepal_width = [] # Sepal Width
petal_length = [] # Petal Length
petal_width = [] # Petal Width

# Выполняем обход всей коллекции data2
for dot in data2:
    sepal_length.append(dot[0])
    sepal_width.append(dot[1])
    petal_length.append(dot[2])
    petal_width.append(dot[3])

# Строим графики по проекциям данных
# Учитываем, что каждые 50 типов ирисов идут последовательно
plt.figure(1)
setosa, = plt.plot(sepal_length[:50], sepal_width[:50], 'ro', label='Setosa')
versicolor, = plt.plot(sepal_length[50:100], sepal_width[50:100], 'g^', label='Versicolor')
virginica, = plt.plot(sepal_length[100:150], sepal_width[100:150], 'bs', label='Verginica')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')

plt.figure(2)
setosa, = plt.plot(sepal_length[:50], petal_length[:50], 'ro', label='Setosa')
versicolor, = plt.plot(sepal_length[50:100], petal_length[50:100], 'g^', label='Versicolor')
virginica, = plt.plot(sepal_length[100:150], petal_length[100:150], 'bs', label='Verginica')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Sepal Length')
plt.ylabel('Petal Length')

plt.figure(3)
setosa, = plt.plot(sepal_length[:50], petal_width[:50], 'ro', label='Setosa')
versicolor, = plt.plot(sepal_length[50:100], petal_width[50:100], 'g^', label='Versicolor')
virginica, = plt.plot(sepal_length[100:150], petal_width[100:150], 'bs', label='Verginica')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Sepal Length')
plt.ylabel('Petal Width')

plt.show()
```




Из графиков уже хорошо видно, что множество Setosa хорошо отделимо, а множества Versicolor и Verginica представляют собой множества, разделение которых является непростой задачей. Следует помнить, что цель первичного исследования данных – получение представления о структуре и природе данных, а не построение модели предсказания, классификации и т.п.

**Тема 3.**

*Лабораторная работа 2 «ВИЗУАЛИЗАЦИЯ ДАННЫХ»*

Цель работы: изучение программных средств для визуализации наборов данных.

Основные задачи:

– установка и настройка matplotlib, seaborn;

–изучение основных типов графиков библиотеки matplotlib;

–изучение основных типов графиков библиотеки seaborn;

– получение навыков анализа данных по визуальным представлениям данных.

Индивидуальное задание

1. Подберите набор данных на ресурсе https://archive.ics.uci.edu/datasets и согласуйте свой выбор с преподавателем.

2. Проведите первичный анализ данных. Особое внимание следует уделить графическому представлению распределений признаков, визуализации взаимосвязей, позволяющие судить о наборе данных. Построение графиков желательно произвести по нескольким проекциям. При анализе данных использовать как можно более разнообразные типы графиков.

Содержание отчета и его форма

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и результаты, полученные в ходе её выполнения.

Отчет о выполнении лабораторной работы сдается преподавателю в бумажной или электронной форме (по согласованию).

Методические указания

Выполним в качестве примера анализ набора данных «Предсказание ухода клиента». Данный набор данных используется в качестве учебного набора при изучении методов прогнозирования. Набор представляет собой данные об активности клиентов телекоммуникационной компании (количество часов разговоров, видеозвонков, ночные и дневные разговоры и прочие). Набор данных подходит для обучения моделей логистической регрессии, моделей классификации (CNN, kNN, Logic tree). Набор данных можно получить на портале Kaggle https://www.kaggle.com/datasets/keyush06/telecom-churncsv

1. Подключение библиотек и загрузка данных

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
data_path = "../datasets/telecom_churn/telecom_churn.csv"
data = pd.read_csv(data_path)
data.head(10)
# data.columns
```

| | State | Account length | Area code | International plan | Voice mail plan | Number vmail messages | Total day minutes | Total day calls | Total day charge | Total eve minutes | Total eve calls | Total eve charge | Total night minutes | Total night calls | Total night charge | T minu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | KS | 128 | 415 | No | Yes | 25 | 265.1 | 110 | 45.07 | 197.4 | 99 | 16.78 | 244.7 | 91 | 11.01 | |
| 1 | OH | 107 | 415 | No | Yes | 26 | 161.6 | 123 | 27.47 | 195.5 | 103 | 16.62 | 254.4 | 103 | 11.45 | |
| 2 | NJ | 137 | 415 | No | No | 0 | 243.4 | 114 | 41.38 | 121.2 | 110 | 10.30 | 162.6 | 104 | 7.32 | |
| 3 | OH | 84 | 408 | Yes | No | 0 | 299.4 | 71 | 50.90 | 61.9 | 88 | 5.26 | 196.9 | 89 | 8.86 | |
| 4 | OK | 75 | 415 | Yes | No | 0 | 166.7 | 113 | 28.34 | 148.3 | 122 | 12.61 | 186.9 | 121 | 8.41 | |
| 5 | AL | 118 | 510 | Yes | No | 0 | 223.4 | 98 | 37.98 | 220.6 | 101 | 18.75 | 203.9 | 118 | 9.18 | |
| 6 | MA | 121 | 510 | No | Yes | 24 | 218.2 | 88 | 37.09 | 348.5 | 108 | 29.62 | 212.6 | 118 | 9.57 | |
| 7 | MO | 147 | 415 | Yes | No | 0 | 157.0 | 79 | 26.69 | 103.1 | 94 | 8.76 | 211.8 | 96 | 9.53 | |
| 8 | LA | 117 | 408 | No | No | 0 | 184.5 | 97 | 31.37 | 351.6 | 80 | 29.89 | 215.8 | 90 | 9.71 | |
| 9 | WV | 141 | 415 | Yes | Yes | 37 | 258.6 | 84 | 43.96 | 222.0 | 111 | 18.87 | 326.4 | 97 | 14.69 | |

Рассмотрим основные признаки, представленный в наборе. Загрузим набор данных с использованием pandas и выведем признаки набора данных. Набор данных telecom_churn.csv содержит большое количество признаков. Для детального изучения воспользуемся методом info() класса DataFrame.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3333 entries, 0 to 3332
Data columns (total 20 columns):
State                     3333 non-null object
Account length            3333 non-null int64
Area code                 3333 non-null int64
International plan        3333 non-null object
Voice mail plan           3333 non-null object
Number vmail messages     3333 non-null int64
Total day minutes         3333 non-null float64
Total day calls           3333 non-null int64
Total day charge          3333 non-null float64
Total eve minutes         3333 non-null float64
Total eve calls           3333 non-null int64
Total eve charge          3333 non-null float64
Total night minutes       3333 non-null float64
Total night calls         3333 non-null int64
Total night charge        3333 non-null float64
Total intl minutes        3333 non-null float64
Total intl calls          3333 non-null int64
Total intl charge         3333 non-null float64
Customer service calls    3333 non-null int64
Churn                     3333 non-null bool
dtypes: bool(1), float64(8), int64(8), object(3)
memory usage: 498.1+ KB
```

Графики, используемые при анализе данных, делят не по библиотекам, с использованием которых они строятся, а по типам признаков, для анализа которых предназначены графики.

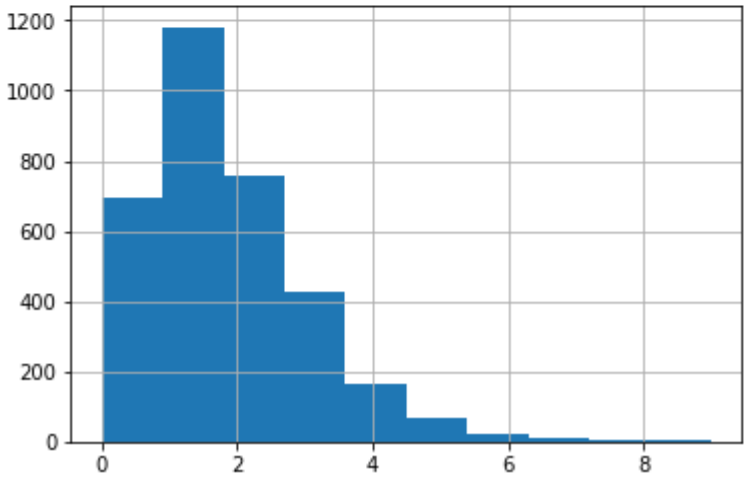
2. Визуализация количественных признаков

Для представления распределения простого количественного признака подходит обычная гистограмма, содержащаяся во всех библиотеках. Для построения гистограммы вызывается метод **hist()** класса DataFtrame.

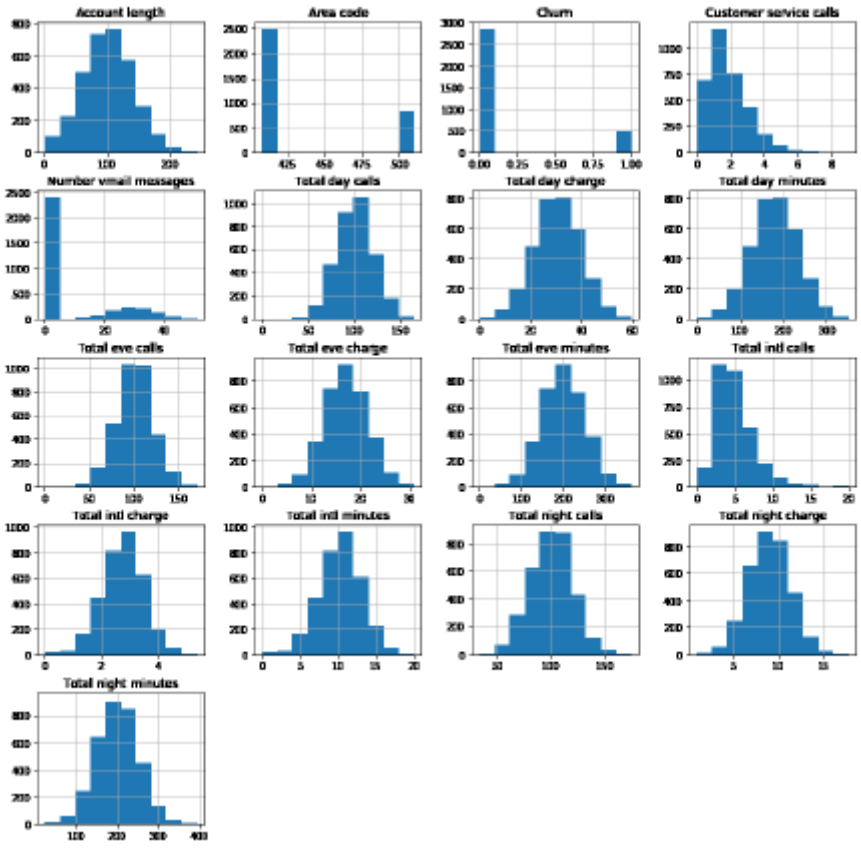
```
data.columns
```

```
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day minutes',
       'Total day calls', 'Total day charge', 'Total eve minutes',
       'Total eve calls', 'Total eve charge', 'Total night minutes',
       'Total night calls', 'Total night charge', 'Total intl minutes',
       'Total intl calls', 'Total intl charge', 'Customer service calls',
       'Churn'],
      dtype='object')
```

```
# Применение pandas для визуализации данных
# Pandas работает как настройка над matplotlib
data['Customer service calls'].hist();
```

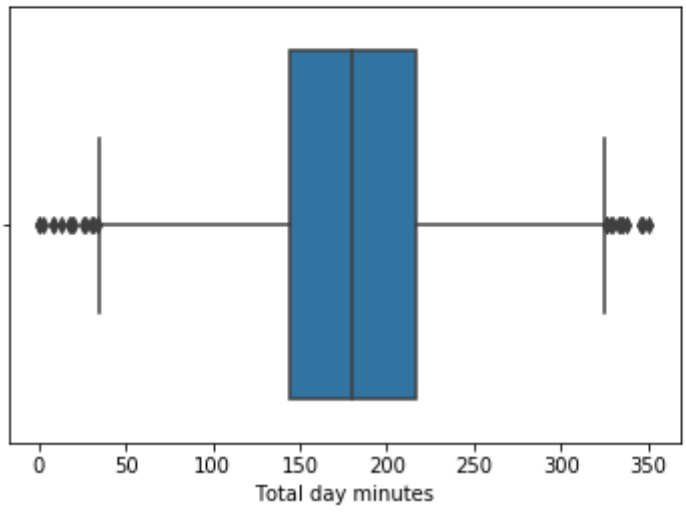



На самом деле используется метод из библиотеки matplotlib. Метод hist() можно использовать для построения гистограмм по нескольким признакам. При этом неколичественные признаки игнорируются.




Один из эффективных типов графиков для анализа количественных признаков – это «ящик с усами» (boxplot). На рисунке показан код и реализованный график.

```
# использование Seaborn
# Построение диаграммы типа "ящик с усами"
# по диаграмме можно определить медиану, квартили,
# интерквартильный размах, выбросы
sns.boxplot(data['Total day minutes']);
```



Для анализа нескольких признаков графики boxplot также эффективны. На рисунке представлен код и результат построения графиков для анализа трех штатов с максимальным объемом дневных звонков.

```
top_data = data[['State','Total day minutes']]
top_data = top_data.groupby('State').sum()
top_data = top_data.sort_values('Total day minutes',ascending=False)
top_data = top_data[:3].index.values
sns.boxplot(y='State',
            x='Total day minutes',
            data=data[data.State.isin(top_data)], palette='Set2');
```

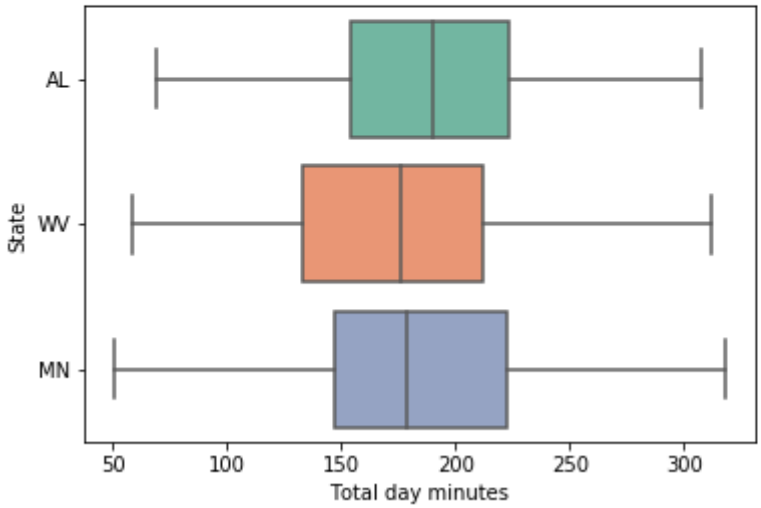
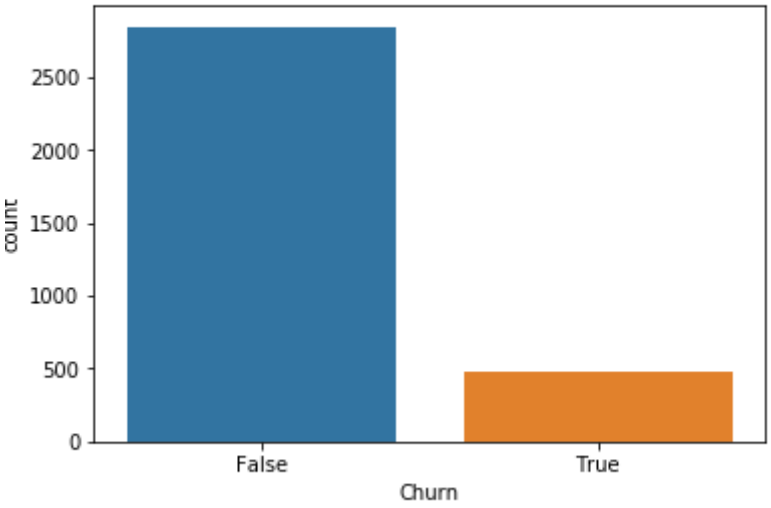



График boxplot состоит из коробки, усов и точек. Коробка показывает интерквартильный размах распределения, то есть соответственно 25% (первая квартиль, $Q1$) и 75% ($Q3$) перцентили. Черта внутри коробки обозначает медиану распределения (можно получить с использованием метода median() в pandas и numpy). Усы отображают весь разброс точек кроме выбросов, то есть минимальные и максимальные значения, которые попадают в промежуток ($Q1-1{,}5\cdot IQR$, $Q3+1{,}5\cdot IQR$), где $IQR=Q3-Q1$ – интерквартильный размах. Точками на графике обозначаются выбросы (outliers), то есть те значения, которые не вписываются в промежуток значений, заданный усами графика.

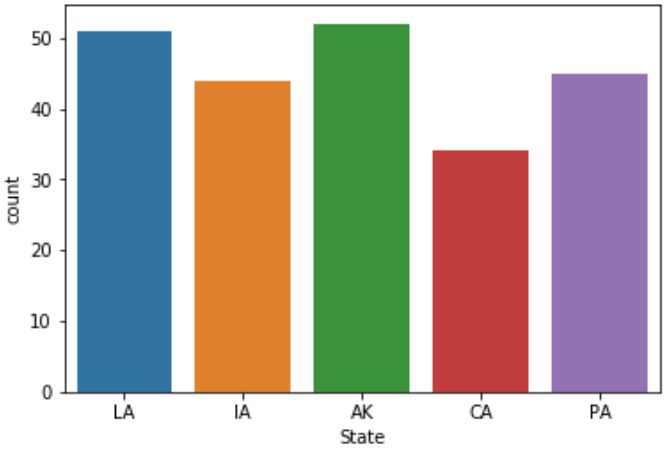
3. Категориальные признаки

Типичным категориальным признаком в анализируемом наборе данных является «Штат» (State). Под категориальный признак подходит также «Отказ» (Churn) (хотя он является логическим). На рисунке представлены графики типа countplot() из библиотеки seaborn, которые строят гистограммы, но не по сырым данным, а по рассчитанному количеству разных значений признака.

```
sns.countplot(data['Churn']);
```




```
# гистограмма "популярных" штатов
sns.countplot(data[data['State'].isin(data['State'].value_counts().tail(5).index)]['State']);
```



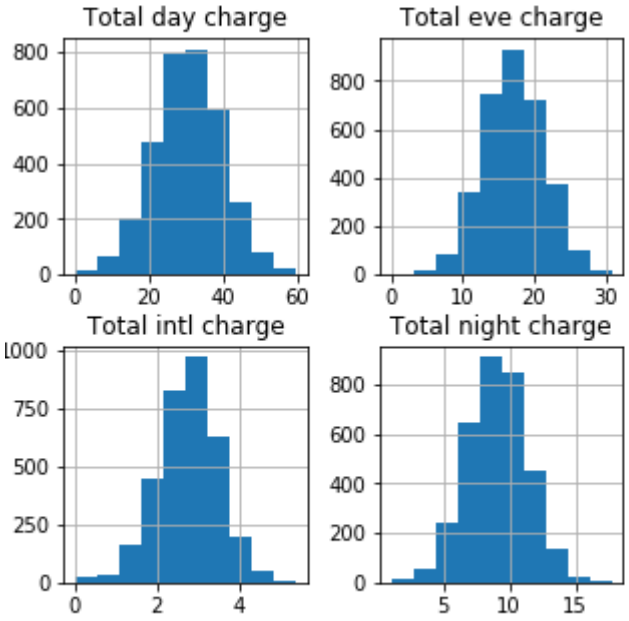

4. Визуализация соотношения количественных признаков

Одним из вариантов визуализации соотношения количественных признаков является диаграмма по нескольким признакам. Рассмотрим пример демонстрирующий сравнениераспределений показателей, связанных с финансовыми затратами клиентов. Упрощенно, можно сказать, что это все показатели, содержащие подстроку «charge» в имени показателя. На рисунке представлен код для отбора требуемых показателей. После отбора интересующих показателей можно построить диаграммы для сравнения.

```
# Отбор числовых признаков, содержащих слово 'charge'
feats = [f for f in data.columns if 'charge' in f]
len(feats)
# feats=['Total day calls', 'Total day charge']
```

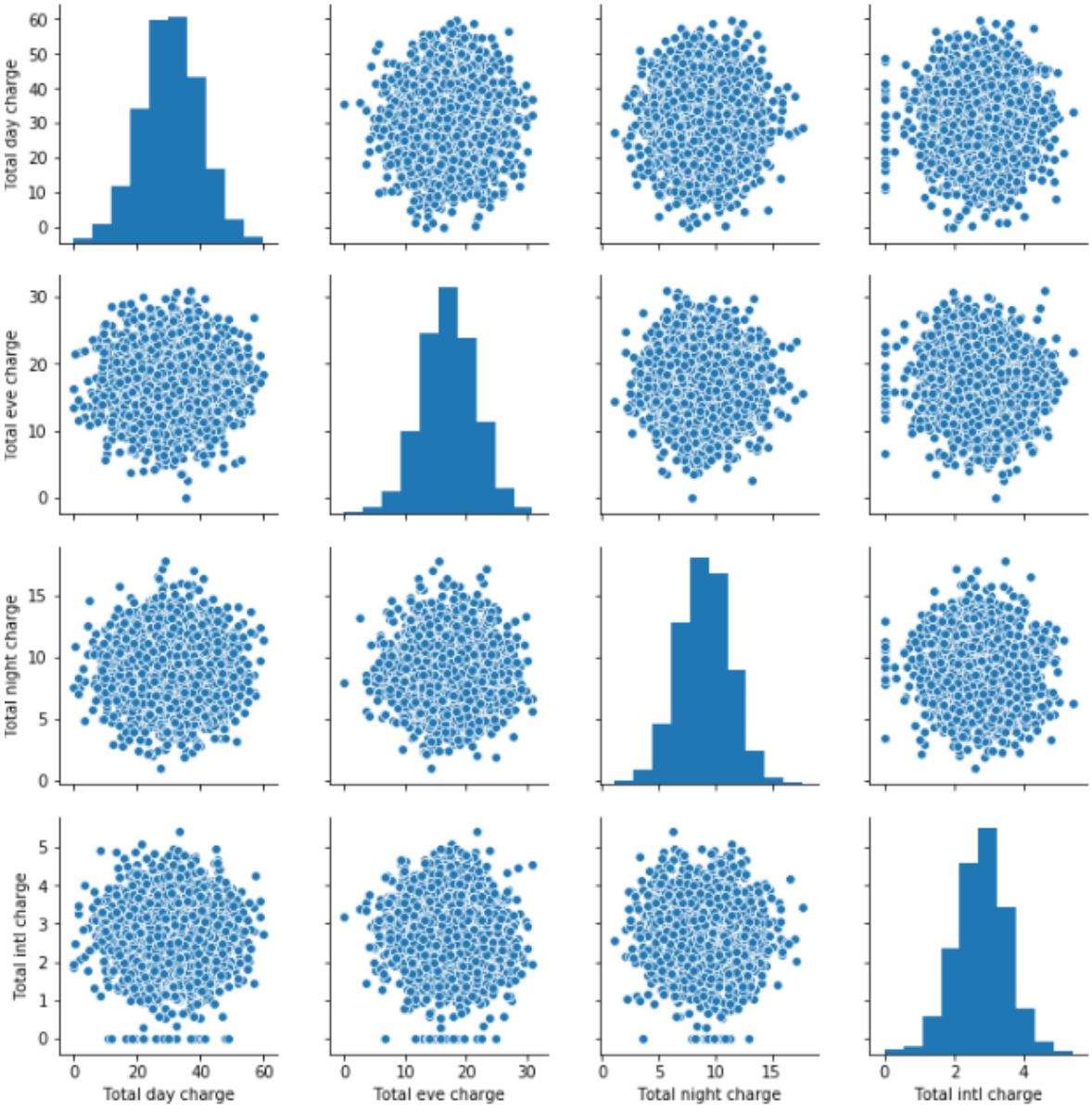
```
4
```

```
# строим отдельные гистограммы
# для нескольких признаков
data[feats].hist(figsize=(5,5));
```



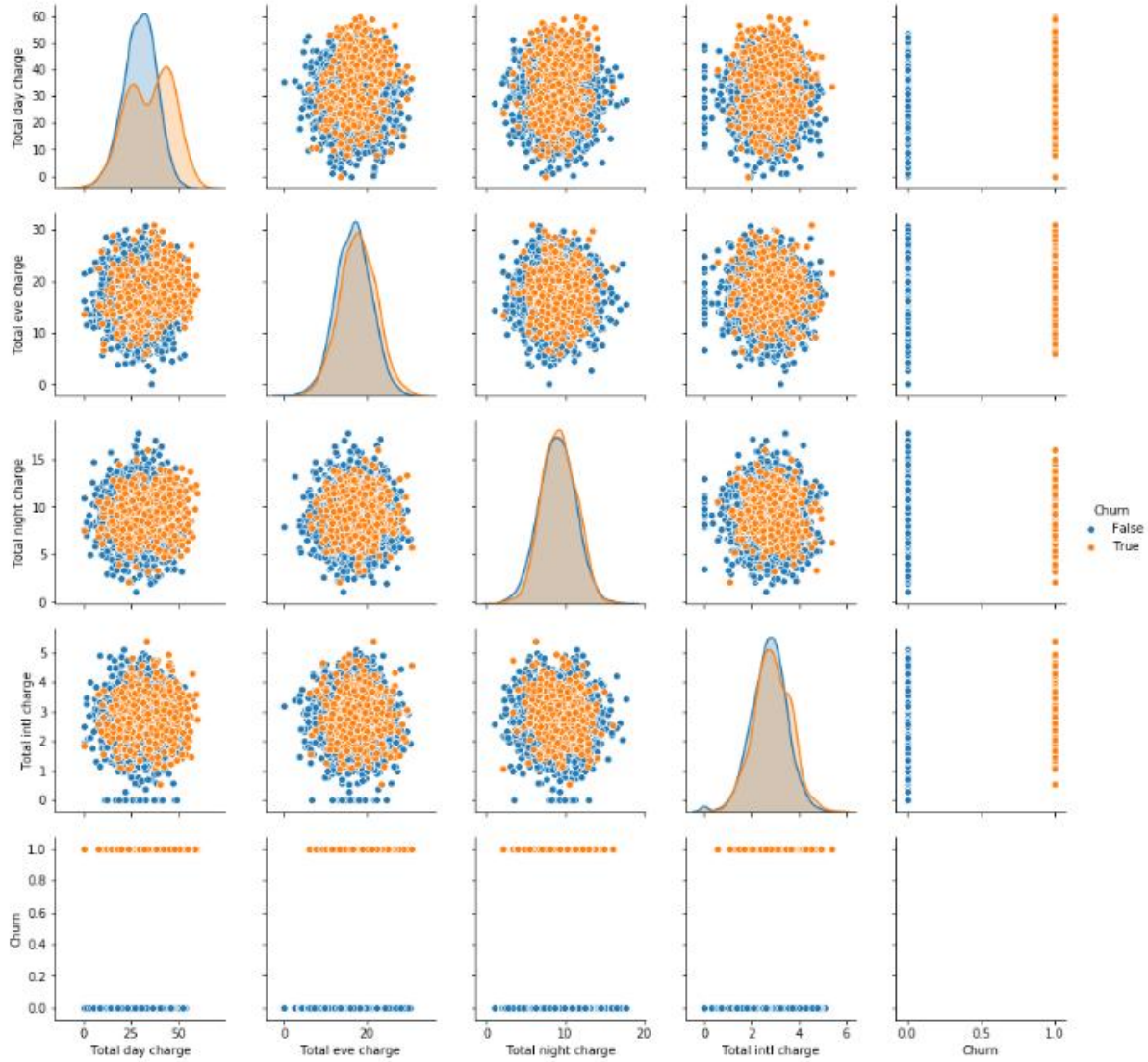

Часто используют попарное сравнение признаков для обеспечения широкого взгляда на набор данных. На диагональных графиках рисунка представлены гистограммы распределения отдельного признака, на внедиаганальных позициях – попарные распределения.

```
# Попарное распределение признаков
# Применение Seaborn
sns.pairplot(data[feats]);
```



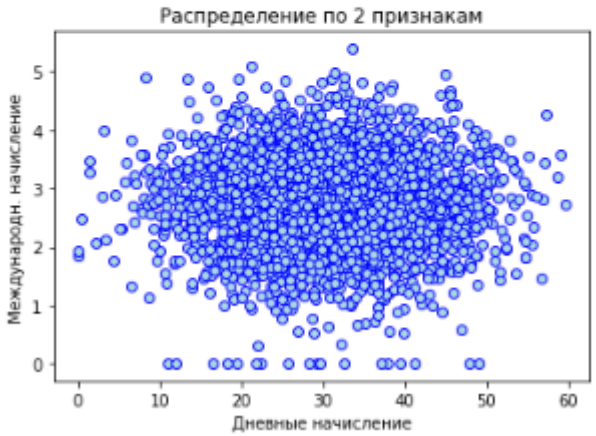

Можно реализовать более сложные графики. Например, если требуется добавить к существующим признакам, целевой признак Churn (количество отказов) и раскрасить разные типы элементов, то можно воспользоваться попарными распределениями, но с отображением подмножеств отказов.

```
sns.pairplot(data[feats + ['Churn']], hue='Churn');
```



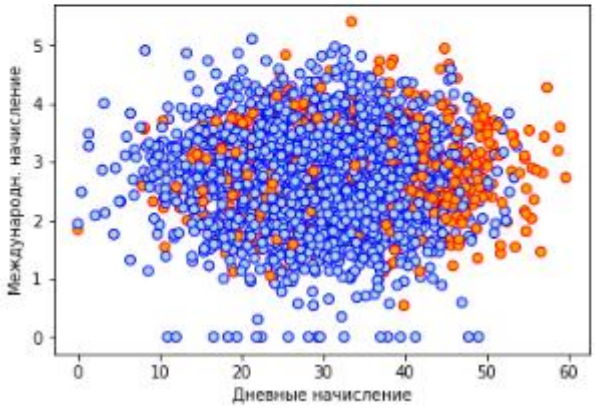

Графика scatter библиотеки matplotlib, предназначенного для вывода множества точек.

```
plt.scatter(data['Total day charge'],
            data['Total intl charge'],
            color='lightblue', edgecolors='blue')
plt.xlabel('Дневные начисление')
plt.ylabel('Международн. начисление')
plt.title('Распределение по 2 признакам');
```




Более тонкой настройки параметров графика.

```
# Раскрашивание данных
# Цвет в зависимости от ухода клиента
c = data['Churn'].map({False: 'lightblue', True: 'orange'})
edge_c = data['Churn'].map({False: 'blue', True: 'red'})
# Настройка графика
plt.scatter(data['Total day charge'], data['Total intl charge'],
            color=c, edgecolors=edge_c
           )
plt.xlabel('Дневные начисление')
plt.ylabel('Международн. начисление');
```



```
# Раскраска лояльных и ушедших клиентов,
# добавление легенды

# Ушедшие клиенты
data_churn = data[data['Churn']]
# Оставшиеся клиенты
data_loyal = data[~data['Churn']]

plt.scatter(data_churn['Total day charge'],
            data_churn['Total intl charge'],
            color='orange',
            edgecolors='red',
            label='Ушли'
           )
plt.scatter(data_loyal['Total day charge'],
            data_loyal['Total intl charge'],
            color='lightblue',
            edgecolors='blue',
            label='Остались'
           )
plt.xlabel('Дневные начисление')
plt.ylabel('Международн. начисление')
plt.title('Распределение клиентов')
plt.legend();
```

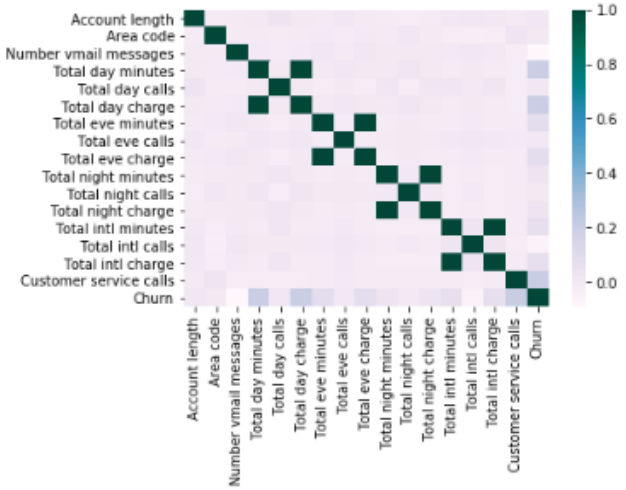

В реальных задачах машинного обучения при первичном анализе данных необходимо выявить корреляции признаков обучающей выборки. В пакете Pandas имеется встроенный инструмент для этого – метод **corr()** класса DataFrame. На рисунке показан фрагмент вывода этой функции.

```
# Применяется функция corr() из Pandas
data.corr()
```

| | Account length | Area code | Number vmail messages | Total day minutes | Total day calls | Total day charge | Total eve minutes | Total eve calls | Total eve charge | Total night minutes | Total night calls | Tot nig char |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Account length** | 1.000000 | -0.012463 | -0.004628 | 0.006216 | 0.038470 | 0.006214 | -0.006757 | 0.019260 | -0.006745 | -0.008955 | -0.013176 | -0.0089 |
| **Area code** | -0.012463 | 1.000000 | -0.001994 | -0.008264 | -0.009646 | -0.008264 | 0.003580 | -0.011886 | 0.003607 | -0.005825 | 0.016522 | -0.0058 |
| **Number vmail messages** | -0.004628 | -0.001994 | 1.000000 | 0.000778 | -0.009548 | 0.000776 | 0.017562 | -0.005864 | 0.017578 | 0.007681 | 0.007123 | 0.0076 |
| **Total day minutes** | 0.006216 | -0.008264 | 0.000778 | 1.000000 | 0.006750 | 1.000000 | 0.007043 | 0.015769 | 0.007029 | 0.004323 | 0.022972 | 0.0043 |
| **Total day calls** | 0.038470 | -0.009646 | -0.009548 | 0.006750 | 1.000000 | 0.006753 | -0.021451 | 0.006462 | -0.021449 | 0.022938 | -0.019557 | 0.0229 |
| **Total day charge** | 0.006214 | -0.008264 | 0.000776 | 1.000000 | 0.006753 | 1.000000 | 0.007050 | 0.015769 | 0.007036 | 0.004324 | 0.022972 | 0.0043 |

Полученная матрица имеет размер 17×17. Это незначительный размер (в реальных задачах машинного обучения размеры матриц корреляции имеют порядки $10^6$–$10^{10}$ и более), но даже для матрицы рассматриваемого набора данных проанализировать корреляцию признаков вручную – трудоемкая задача. Например, можно использовать скрипты, для выделения больших коэффициентов корреляции. Но лучше использовать специальный тип графика – heatmap.

```
sns.heatmap(data.corr(), cmap=plt.cm.PuBuGn);
```




Из карты heatmap видно, что некоторые признаки коррелируют: например сильная корреляция в парах (total day charge, total day minutes), (total night charge, total night minutes). Из таких пар можно удалить один признак

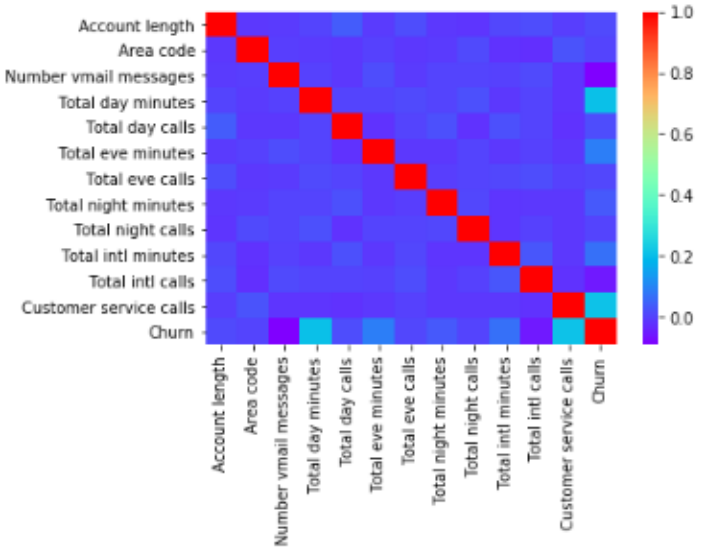
Коррелирующие признаки обычно удаляются и не рассматриваются в процессе обучения.

```
# Удаление коррелирующих признаков
data_uncorr = data.drop(feats, axis=1)
data_uncorr.columns
```

```
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day minutes',
       'Total day calls', 'Total eve minutes', 'Total eve calls',
       'Total night minutes', 'Total night calls', 'Total intl minutes',
       'Total intl calls', 'Customer service calls', 'Churn'],
      dtype='object')
```

Перестраиваем heatmap без коррелирующих признаков

```
sns.heatmap(data_uncorr.corr(), cmap=plt.cm.rainbow);
```




**Тема 4.**

*Лабораторная работа 3 «ПРЕДОБРАБОТКА ДАННЫХ»*

Цель: Ознакомиться с методами предобработки данных

Основные задачи:

– загрузка и проверка данных

– шкалирование данных

Индивидуальное задание

1. Подберите набор данных на ресурсе https://archive.ics.uci.edu/datasets и согласуйте свой выбор с преподавателем.

2. Проведите предобработку данных.

Содержание отчета и его форма

1. Номер и название лабораторной работы; задачи лабораторной работы.

2. Реализация каждого пункта подраздела «Индивидуальное задание» с приведением исходного кода программы, диаграмм и графиков для визуализации данных.

3. Экранные формы (консольный вывод) и листинг программного кода с комментариями, показывающие порядок выполнения лабораторной работы, и

результаты, полученные в ходе её выполнения.

4. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы сдается преподавателю в бумажной или электронной форме (по согласованию).

Методические указания

1. Загрузка данных. Загрузить любой датасет по ссылке: https://archive.ics.uci.edu/datasets
Ниже пример для данных https://www.kaggle.com/andrewmvd/heart-failure-clinical-data
Данные представлены в виде csv таблицы.

2. Создать Python скрипт. Загрузить датасет в датафрейм, и исключить бинарные признаки и признак времени.

```
import pandas as pd
import numpy as np

df = pd.read_csv('heart_failure_clinical_records_dataset.csv')

df = df.drop(columns =
['anaemia','diabetes','high_blood_pressure','sex','smoking','time','DEATH_EV
ENT'])

print(df) #Вывод датафрейма с данными для лаб. работы. Должно быть 299
наблюдений и 6 признаков
```

3. Построить гистограммы признаков

```
import matplotlib.pyplot as plt

n_bins = 20

fig, axs = plt.subplots(2,3)

axs[0, 0].hist(df['age'].values, bins = n_bins)
axs[0, 0].set_title('age')

axs[0, 1].hist(df['creatinine_phosphokinase'].values, bins = n_bins)
axs[0, 1].set_title('creatinine_phosphokinase')

axs[0, 2].hist(df['ejection_fraction'].values, bins = n_bins)
axs[0, 2].set_title('ejection_fraction')

axs[1, 0].hist(df['platelets'].values, bins = n_bins)
axs[1, 0].set_title('platelets')

axs[1, 1].hist(df['serum_creatinine'].values, bins = n_bins)
axs[1, 1].set_title('serum_creatinine')
```

```
axs[1, 2].hist(df['serum_sodium'].values, bins = n_bins)
axs[1, 2].set_title('serum_sodium')

plt.show()
```

4. На основании гистограмм определите диапазоны значений для каждого из признаков, а также возле какого значения лежит наибольшее количество наблюдений.

5. Так как библиотека Sklearn работает с NumPy массива, то преобразуйте датафрейм к двумерному массиву NumPy, где строка соответствует наблюдению, а столбец признаку

```
data = df.to_numpy(dtype='float')
```

6. Стандартизация данных

6.1. Подключите модуль Sklearn. Настройте стандартизацию на основе первых 150 наблюдений
используя StandardScaler

```
from sklearn import preprocessing

scaler = preprocessing.StandardScaler().fit(data[:150,:])
```

6.2. Стандартизуйте все данные

```
data_scaled = scaler.transform(data)
```

6.3. Постройте гистограммы стандартизированных данных

```
fig, axs = plt.subplots(2,3)

axs[0, 0].hist(data_scaled[:,0], bins = n_bins)
axs[0, 0].set_title('age')

axs[0, 1].hist(data_scaled[:,1], bins = n_bins)
axs[0, 1].set_title('creatinine_phosphokinase')

axs[0, 2].hist(data_scaled[:,2], bins = n_bins)
axs[0, 2].set_title('ejection_fraction')

axs[1, 0].hist(data_scaled[:,3], bins = n_bins)
axs[1, 0].set_title('platelets')

axs[1, 1].hist(data_scaled[:,4], bins = n_bins)
axs[1, 1].set_title('serum_creatinine')

axs[1, 2].hist(data_scaled[:,5], bins = n_bins)
axs[1, 2].set_title('serum_sodium')

plt.show()
```

6.4. Сравните данные до и после стандартизации. Опишите, что изменилось и почему.

6.5. Рассчитайте мат. ожидание и СКО до и после стандартизации. На основании этих значений

выведите для каждого признака формулы по которым они стандартизировались.

6.6. Сравните значений из формул с полями mean_ и var_ объекта scaler

6.7. Проведите настройку стандартизации на всех данных и сравните с результатами настройки на основании 150 наблюдений.

7. Приведение к диапазону

7.1. Приведите данные к диапазону используя MinMaxScaler

```
min_max_scaler = preprocessing.MinMaxScaler().fit(data)
data_min_max_scaled = min_max_scaler.transform(data)
```

7.2. Постройте гистограммы для признаков и сравните с исходными данными

7.3. Через параметры MinMaxScaler определите минимальное и максимальное значение в данных для каждого признака.

7.4. Аналогично трансформируйте данные используя MaxAbsScaler и RobustScaler. Постройте гистограммы. Определите к какому диапазону приводятся данные.

7.5. Напишите функцию, которая приводит все данные к диапазону [-5; 10].

8. Нелинейные преобразования

8.1. Приведите данные к равномерному распределению используя QuantileTransformer

```
quantile_transformer = preprocessing.QuantileTransformer(n_quantiles = 100,
random_state=0).fit(data)
data_quantile_scaled = quantile_transformer.transform(data)
```

8.2. Постройте гистограммы и сравните с исходными данными
8.3. Определите, как и на что влияет значение параметра n_quantiles
8.4. Приведите данные к нормальному распределению передав в QuantileTransformer параметр output_distribution='normal'
8.5. Постройте гистограммы и сравните с исходными данными
8.6. Самостоятельно приведите данные к нормальному распределению используя PowerTransformer

9. Дискретизация признаков
9.1. Проведите дискретизацию признаков, используя KBinsDiscretizer, на следующее количество диапазонов:
age - 3
creatinine_phosphokinase - 4
ejection_fraction - 3
platelets - 10
serum_creatinine - 2
serum_sodium - 4
9.2. Постройте гистограммы. Объясните полученные результаты.
9.3. Через параметр bin_edges_ выведите диапазоны каждого интервала для каждого признака.

## 8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

Данный вид работы не предусмотрен УП.

## 9. ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЧЕТУ

1. Язык Python и особенности его стиля программирования. Интерактивный режим Python.
2. Синтаксис и управляющие конструкции языка Python. Переменные, значения и их типы. Типы данных в Python.
3. Встроенные операции и функции. Основные алгоритмические конструкции.
4. Условный оператор. Множественное ветвление.
5. Циклы и счетчики.
6. Определение функций. Параметры и аргументы. Вызовы функций. Оператор возврата. Конструкции *args, **kwargs.
7. Списки, кортежи и словари.
8. Операторы общие для всех типов последовательностей.
9. Специальные операторы и функции для работы со списками. Срезы.
10. Работа со словарями. Методы словарей.
11. Случайные числа random, randrange, choice.
12. Функции обработки строк  join, replace, split.
13. Стандартная библиотека и pip. Модули и пакеты в Python. Основные стандартные модули. Программа дисциплины "Язык Python и анализ данных.
14. Импортирование модулей. Создание собственных модулей и их импортирование. Специализированные модули и приложения.
15. Файлы и исключения. Работа с внешними источниками данных.
16. Исключения, обработка исключений, вызов исключений (try-except-finally).

17. Утверждения (assert). Открытие, чтение, запись. (open, инструкция with).
18. Работа с текстовыми файлами, xml и csv - файлами.
19. Функциональное программирование. Лямбда-функции.
20. Использование функций map, filter, reduce, zip.
21. Генераторы, декораторы, рекурсия.
22. Модификация функций с помощью декораторов.
23. Итерируемые объекты. Использование генераторов (yield).
24. Наука о данных и Python. Библиотеки: NumPy, pandas, matplotlib, SciPy.
25. Основы NumPy: массивы и векторные вычисления.
26. Инструменты визуализации данных для Python.
27. Введение в API библиотеки matplotlib.
28. Библиотека pandas. Введение в структуры данных pandas.
29. Объекты Dataframe и Series.
30. Визуализация данных в pandas. Seaborn.
31. Агрегирование данных и групповые операции.

## 10. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩИХСЯ

Форма обучения: очная

| Название разделов и тем | Вид самостоятельной работы | Объем в часах | Форма контроля |
| --- | --- | --- | --- |
| **Тема 1. Введение в программирование на языке Python** | Проработка учебного материала, решение задач, подготовка к сдаче зачета. | 12 | Проверка домашнего задания, зачет. |
| **Тема 2. Обработка данных. Массивы и векторные вычисления** | Проработка учебного материала, решение задач, подготовка к сдаче лабораторной работы, подготовка к сдаче зачета. | 14 | Проверка домашнего задания, проверка лабораторной работы, зачет. |
| **Тема 3. Построение графиков и визуализация данных** | Проработка учебного материала, решение задач, подготовка к сдаче лабораторной работы, подготовка к сдаче зачета. | 14 | Проверка домашнего задания, проверка лабораторной работы, зачет. |
| **Тема 4. Специализированные библиотеки Python для анализа данных** | Проработка учебного материала, решение задач, подготовка к сдаче лабораторной работы, подготовка к сдаче зачета. | 14 | Проверка домашнего задания, проверка лабораторной работы, зачет. |

## 11. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### а) Список рекомендуемой литературы

**основная**

1. Златопольский Д.М., Основы программирования на языке Python [Электронный ресурс]: учебник / Златопольский Д. М. - М. : ДМК Пресс, 2017. - 284 с. - ISBN 978-5-97060-552-3 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785970605523.html
2. Федоров, Д. Ю. Программирование на языке высокого уровня python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 161 с. — (Бакалавр. Прикладной курс). — ISBN 978-5-534-10971-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://www.biblio-online.ru/bcode/437489

**дополнительная**

1. Лучано Рамальо, Python. К вершинам мастерства [Электронный ресурс] / Лучано Рамальо - М. : ДМК Пресс, 2016. - 768 с. - ISBN 978-5-97060-384-0 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785970603840.html
2. Маккинли У., Python и анализ данных [Электронный ресурс] / Уэс Маккинли - М. : ДМК Пресс, 2015. - 482 с. - ISBN 978-5-97060-315-4 - Режим доступа: http://www.studentlibrary.ru/book/ISBN9785970603154.html

**учебно-методическая (разработанная НПР, реализующими ОПОП ВО)**

1. **Бурмистрова** В. Г. Python для анализа данных : методические указания для самостоятельной работы студентов бакалавриата направления 01.03.02 Прикладная математика и информатика / В. Г. **Бурмистрова** ; УлГУ, ФМИиАТ. - 2023. - Неопубликованный ресурс. - URL: http://lib.ulsu.ru/MegaPro/Download/MObject/15469 . - Режим доступа: ЭБС УлГУ. - Текст : электронный.

Согласовано:

_____/_____/_____/_____
Должность сотрудника научной библиотеки ФИО подпись дата

### б) Программное обеспечение

Для образовательного процесса студенту необходимо рабочее место с ПК с установленным следующим программным обеспечением: операционная среда ОС Windows/Linux; MS Office системы программирования на языке, Python3.

### в) Профессиональные базы данных, информационно-справочные системы

**1. Электронно-библиотечные системы:**

1.1. Цифровой образовательный ресурс IPRsmart : электронно-библиотечная система : сайт / ООО Компания «Ай Пи Ар Медиа». - Саратов, [2024]. – URL: http://www.iprbookshop.ru. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.2. Образовательная платформа ЮРАЙТ : образовательный ресурс, электронная библиотека : сайт / ООО Электронное издательство ЮРАЙТ. – Москва, [2024]. - URL: https://urait.ru. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.3. База данных «Электронная библиотека технического ВУЗа (ЭБС «Консультант студента») : электронно-библиотечная система : сайт / ООО Политехресурс. – Москва, [2024]. – URL: https://www.studentlibrary.ru/cgi-bin/mb4x. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.4. Консультант врача. Электронная медицинская библиотека : база данных : сайт / ООО Высшая школа организации и управления здравоохранением-Комплексный медицинский консалтинг. – Москва, [2024]. – URL: https://www.rosmedlib.ru. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.5. Большая медицинская библиотека : электронно-библиотечная система : сайт / ООО Букап. – Томск, [2024]. – URL: https://www.books-up.ru/ru/library/ . – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.6. ЭБС Лань : электронно-библиотечная система : сайт / ООО ЭБС Лань. – Санкт-Петербург, [2024]. – URL: https://e.lanbook.com. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.7. ЭБС Znanium.com : электронно-библиотечная система : сайт / ООО Знаниум. - Москва, [2024]. - URL: http://znanium.com . – Режим доступа : для зарегистрир. пользователей. - Текст : электронный.

**2. КонсультантПлюс** [Электронный ресурс]: справочная правовая система. / ООО «Консультант Плюс» - Электрон. дан. - Москва : КонсультантПлюс, [2024].

**3. Базы данных периодических изданий:** eLIBRARY.RU: научная электронная библиотека : сайт / ООО «Научная Электронная Библиотека». – Москва, [2024]. – URL: http://elibrary.ru. – Режим доступа : для авториз. пользователей. – Текст : электронный

**4. Федеральная государственная информационная система «Национальная электронная библиотека»** : электронная библиотека : сайт / ФГБУ РГБ. – Москва, [2024]. – URL: https://нэб.рф. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

**5. Российское образование** : федеральный портал / учредитель ФГАУ «ФИЦТО». – URL: http://www.edu.ru. – Текст : электронный.

**6. Электронная библиотечная система УлГУ** : модуль «Электронная библиотека» АБИС Мега-ПРО / ООО «Дата Экспресс». – URL: http://lib.ulsu.ru/MegaPro/Web. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

Согласовано:

_____/_____/_____/_____

Должность сотрудника УИТиТ ФИО подпись дата

## 12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ:

Аудитории для выполнения лабораторных работ и практикумов, для проведения текущего контроля и промежуточной аттестации.

Аудитории укомплектованы специализированной мебелью, учебной доской. Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной инфромационно-образовательной среде, электронно-библиотечной системе. Перечень оборудования, используемого в учебном процессе, указывается в соответствии со сведениями о материально-техническом обеспечении и оснащенности образовательного процесса, размещенными на официальном сайте УлГУ в разделе «Сведения об образовательной организации».

## 13. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ДЛЯ ОБУЧАЮЩИХСЯ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей:

– для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации;

– для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации;

– для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации;

В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий, организация работы ППС с обучающимися с ОВЗ и инвалидами предусматривается в электронной информационно-образовательной среде с учетом их индивидуальных психофизических особенностей.

| Разработчик | _____ | доцент | Савинов Ю.Г. |
| --- | --- | --- | --- |
| | подпись | должность | ФИО |