Лабораторная работа №4
Решение статистических задач средствами Python.

Представление данных. Данные как таблица.
В общем случае мы рассматриваем данные как набор из нескольких наблюдений и объектов, описываемые различными признаками. Данные можно рассматривать как таблицу или матрицу, где столбцы – это признаки, а строки – объекты.
Хранить данные и управлять ими можно используя библиотеку pandas и такой класс в нем как DataFrame. В некоторой форме это аналог электронных таблиц в Python. В отличие от 2D массивов в numpy, DataFrame имеет именованные колонки, может содержать смешанные типы данных в колонках, имеет выборочную селекцию.
Создание DataFrame.
В качестве примера чтения DataFrame из csv-файла рассмотрим данные о размере мозга, его весе и уровнем IQ(Willerman et al. 1991), эти данные представляют собой смесь численных и категориальных значений.

```
import pandas
data = pandas.read_csv('brain_size.csv', sep=';', na_values=".")
print(data)
```

```
    Unnamed: 0  Gender  FSIQ  VIQ  PIQ  Weight  Height  MRI_Count
0            1  Female   133  132  124   118.0    64.5     816932
1            2    Male   140  150  124     NaN    72.5    1001121
2            3    Male   139  123  150   143.0    73.3    1038437
```

Обратите внимание, что если значение было пропущено, то ем устанавливается маркер NA (сокр. от английского not available). В случае если этого не сделать дальнейший анализ будет невозможен.
Создать DataFrame так же можно и из массива numpy:

```
import numpy as np
t = np.linspace(-6, 6, 20)
sin_t = np.sin(t)
cos_t = np.cos(t)

data1 = pandas.DataFrame({'t': t, 'sin': sin_t, 'cos': cos_t})
print(data1)
print(type(data1))
```

В pandas массивы можно также создавать из SQL, файлов excel и других форматов.
Задание 1. Реализуйте код, приведенный в примерах.

УПРАВЛЕНИЕ ДАННЫМИ.
DataFrame в pandas по своей структуре схожи с dataframe в R:

```
print(data.shape)    # 40 строк и 8 столюцов
print(data.columns)  # Названия столбцов
print(data['Gender'])  # Обращенеи к столбцам по именам
# Простейшая выборка
print(data[data['Gender'] == 'Female']['VIQ'].mean())
```

Для того, чтобы просмотреть большой массив, используется функция pandas.DataFrame.describe().
Функция groupby разделяет dataframe на по категориям

```
groupby_gender = data.groupby('Gender')
for gender, value in groupby_gender['VIQ']:
    print((gender, value.mean()))
```

Полученный объект groupby_gender позволяет выполнять различные операции над результирующими группами:

```
print(groupby_gender.mean())
```

Другие функции, которые можно выполнять отдельно над группами – это медиана, счетчик (используется для подсчета количества пропущенных значений в каждом наборе) или сумма. Функция groupby – это так называемая ленивая функция, когда вычисления происходят по мере необходимости результат.
Для построения коробчатых диаграмм используется функция boxplot.
Задание 2. Реализуйте код, приведенный в примере. Посчитайте среднее значение по признаку *VIQ* для всей группы, посчитайте количество женщин и мужчин, какое медианное значение у параметра *MRI* отдельно для мужчин и для женщин. Для объекта *groupby_gender* постройте коробчатые диаграммы по *'FSIQ'*, *'VIQ'*, *'PIQ'*.

Построение графиков
В библиотеке Pandas имеется набор средств для построения графиков (модуль pandas.tools.plotting использует библиотеку matplotlib) для проведения статистики данных в dataframe:

```
from pandas.tools import plotting
plotting.scatter_matrix(data[['Weight', 'Height', 'MRI_Count']])
```

Задание 3. Реализуйте код, приведенный в примере. Постройте матрицу рассеяния отдельно для мужчин и женщин. Как Вы думаете есть две подгруппы по полу или нет?

Статистическая проверка гипотез
Для самых простых статистических тестов используется библиотека scipy и ее модуль scipy.stats.
Функция scipy.stats.ttest_1samp() используя критерий Стьюдента проверяет среднее значение в выборке соответствует или нет теоретическому, эта функция возвращает t-статистику и p-значение.

```
from scipy import stats
stats.ttest_1samp(data['VIQ'], 0)
```

При p-значении равном 1.329e-28 можно утверждать, что среднее значение в VIQ не равно 0.
Как было показано выше среднее значение по признаку «VIQ» в мужской и женской популяции отличается. Для того, чтобы проверить значительно ли они отличаются друг от друга, используется двухсторонний тест Стьюдента, который реализован в функции scipy.stats.ttest_ind().

```
female_viq = data[data['Gender'] == 'Female']['VIQ']
male_viq = data[data['Gender'] == 'Male']['VIQ']
print(stats.ttest_ind(female_viq, male_viq))
```

PIQ, VIQ, и FSIQ дают три оценки IQ. Для того, чтобы проверить FISQ и PIQ значительно отличаются или нет необходимо использовать двухсторонний тест Стьюдента:

```
print(stats.ttest_ind(data['FSIQ'], data['PIQ']))
```

Недостатком этого метода является то, что этот метод не учитывает связи между наблюдениями. FSIQ и PIQ были измерены на одних и тех же людях. Таким образом, того чтобы устранить это влияние необходимо использовать двухвыборочный тест Стьюдента для зависимых выборок, который реализован в функции stats.ttest_rel:

```
print(stats.ttest_rel(data['FSIQ'], data['PIQ']))
```

Который в свою очередь эквивалентен одновыборочному тесту Стьюдента на разности между наблюдениями.

```
print(stats.ttest_1samp(data['FSIQ'] - data['PIQ'], 0))
```

Задание 4. Реализуйте код, приведенный в примерах. Проверьте разницу между весами у мужчин и женщин. Используйте непараметрическую статистику, чтобы проверить разницу между VIQ у мужчин и женщин.

Линейная регрессия.

Рассмотрим две группы наблюдений *x* и *y*, необходимо проверить гипотезу наблюдается ли линейная зависимость между *x* и *y*. Другими словами:

$$y = x * coef + intercept + e$$

где e – это шум. Для этого можно использовать статистическую модель:

1) Подобрать линейную модель
2) Использовать метод наименьших квадратов (МНК)

Во-первых, мы должны смоделировать наблюдения.

```
import numpy as np
x = np.linspace(-5, 5, 20)
np.random.seed(1)
# Добавляем шум с нормальным распределением
y = -5 + 3*x + 4 * np.random.normal(size=x.shape)
# создаем DataFrame
data = pandas.DataFrame({'x': x, 'y': y})
```

Затем необходимо определить МНК модель, для того чтобы посмотреть статистические результаты используется атрибут summary:

```
from statsmodels.formula.api import ols
model = ols("y ~ x", data).fit()
print(model.summary())
```

Рассмотрим данные из предыдущего примера, используя МНК можно сравнить *IQ* между мужчинами и женщинами. По умолчанию, statsmodels рассматривает категориальную переменную с K возможными значениями как K-1 'фиктивные' логические переменные (последний уровень поглощается термином перехвата).

```
data = pandas.read_csv('brain_size.csv', sep=';', na_values=".")
model = ols("VIQ ~ Gender + 1", data).fit()
print(model.summary())
```

«Пол» автоматически определяется как категориальная переменная, и, следовательно, каждое из его значений рассматривается как разные объекты. Так же целочисленные признаки дополнительно можно рассматривать как категориальные.

Для того, чтобы сравнить различные способы оценки *IQ*, необходимо создать периодическую таблицу – список *IQ*, где *IQ* – определяется как категориальная переменная. В этом случае мы получим те же значения для t-теста и соответствующие p-значения для эффекта типа iq:

```
data_fisq = pandas.DataFrame({'iq': data['FSIQ'], 'type': 'fsiq'})
data_piq = pandas.DataFrame({'iq': data['PIQ'], 'type': 'piq'})
data_long = pandas.concat((data_fisq, data_piq))
print(data_long)
model = ols("iq ~ type", data_long).fit()
print(model.summary())
```

Задание 5. Реализуйте код, приведенный в примере. Оцените параметры линейной модели, сравните их с теоретическими.

Визуализация данных.

Библиотека Seaborn комбинирует в себе как базовые статистические модели с построением DataFrame. Рассмотрим набор данных, который содержит в себе персональную информацию около о 500 человек.

```
import seaborn
names = [
    'EDUCATION: Number of years of education',
    'SOUTH: 1=Person lives in South, 0=Person lives elsewhere',
    'SEX: 1=Female, 0=Male',
    'EXPERIENCE: Number of years of work experience',
    'UNION: 1=Union member, 0=Not union member',
    'WAGE: Wage (dollars per hour)',
    'AGE: years',
    'RACE: 1=Other, 2=Hispanic, 3=White',
    'OCCUPATION: 1=Management, 2=Sales, 3=Clerical, 4=Service, 5=Professional, 6=Other',
    'SECTOR: 0=Other, 1=Manufacturing, 2=Construction',
    'MARR: 0=Unmarried,  1=Married',
]
short_names = [n.split(':')[0] for n in names]
data = pandas.read_csv('data2.txt', sep = '  ')
data.columns = short_names
```

Матрицу рассеяния можно использовать для отображения интуитивного взаимодействия между непрерывными переменными

```
seaborn.pairplot(data, vars=['WAGE', 'AGE', 'EDUCATION'],kind='reg')
```

Для построения категориальных переменных можно использовать различные оттенки

```
seaborn.pairplot(data, vars=['WAGE', 'AGE', 'EDUCATION'],
                 kind='reg', hue='SEX')
```

Регрессия, отображающая связь между переменными, например, заработная плата и образование, может быть построена с использованием seaborn.lmplot ():

```
seaborn.lmplot(y='WAGE', x='EDUCATION', data=data)
```

Задание 6. Реализуйте приведенный код.