|  |  |  |
| --- | --- | --- |
|  | |  |
| МИНОБРНАУКИ РОССИИ | |  |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** | |  |
|  | Институт информационных технологий (ИТ) | |
|  | Кафедра прикладной математики | |

|  |  |
| --- | --- |
| **Отчет по практической работе №4.** | |
| **по дисциплине** | |
| **«Технологии и инструментарий анализа больших данных»** | |
|  | |
| Выполнил студент группы ИКБО- 22-20 | Гологузов К.А. |
| Принял | Парамонов А.А. |

Москва 2023

Решение

1.1 Определить два вектора, представляющие собой число автомобилей, припаркованных в течении 5 рабочих дней у бизнес-центра на уличной стоянке и в подземном гараже.

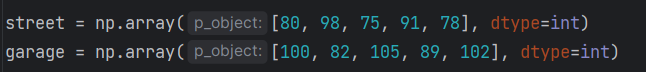


Рис. 1.1 – Два вектора

1.2. Построить диаграмму рассеяния для вышеупомянутых переменных.

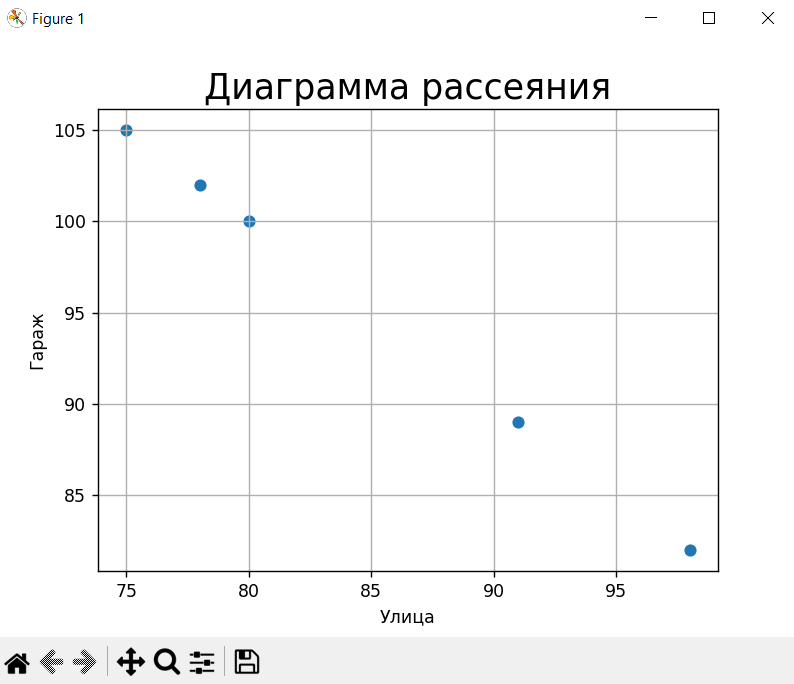


Рис. 1.2 – Диаграмма рассеяния

2. Найти и выгрузить данные. Вывести, провести предобработку и

описать признаки.

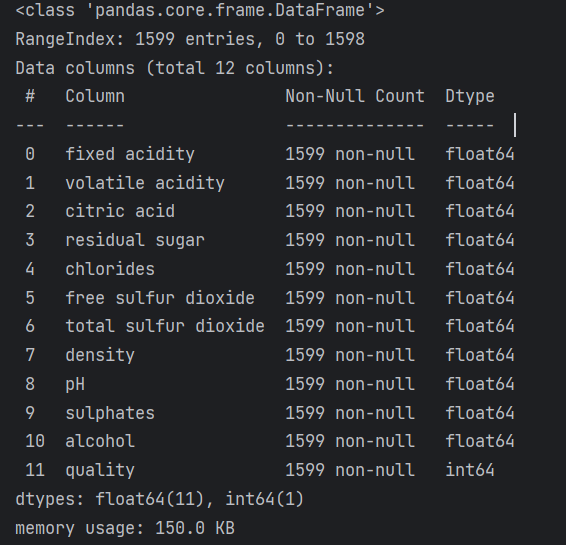


Рис. 2.1 – Вывод информации

2.1. Построить корреляционную матрицу по одной целевой переменной. Определить наиболее коррелирующую переменную, продолжить с ней работу в следующем пункте.

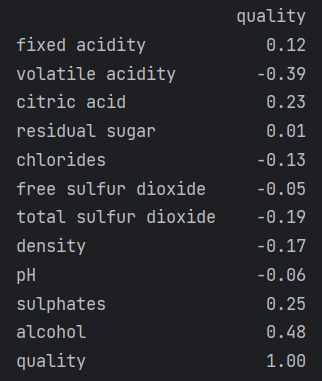


Рис. 2.2 – Корреляционная матрица по переменной quality

2.2. Реализовать регрессию вручную, отобразить наклон, сдвиг и MSE.

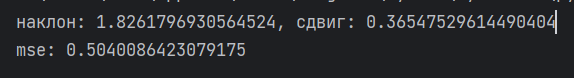


Рис. 2.3 – Результат регрессии

2.3. Визуализировать регрессию на графике.

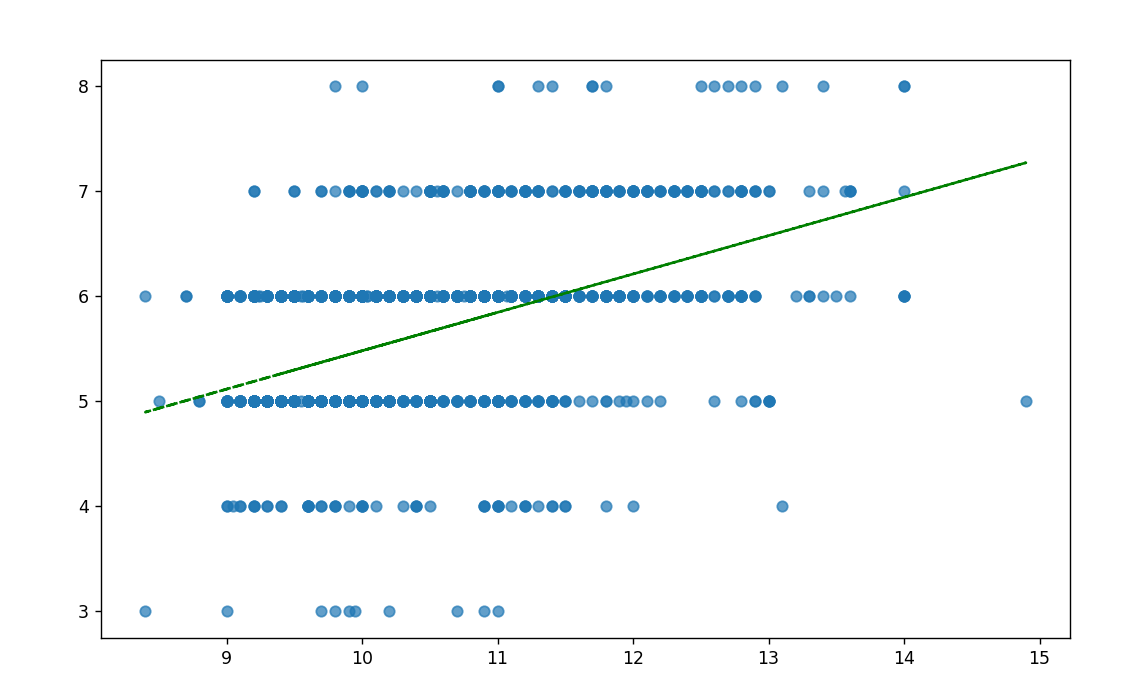


Рис. 2.4 – График регрессии

3. Загрузить данные: 'insurance.csv'. Вывести и провести предобработку. Вывести список уникальных регионов.

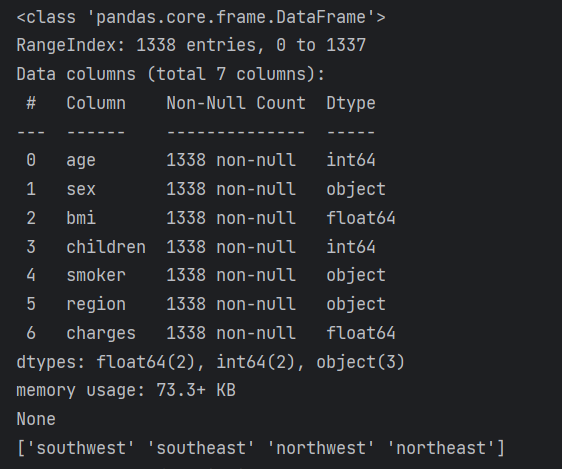


Рис. 3.1 – Данные

3.1. Выполнить однофакторный ANOVA тест, чтобы проверить влияние региона на индекс массы тела (BMI), используя первый способ, через библиотеку Scipy.



Рис. 3.2 – Результат однофакторного теста

3.2. Выполнить однофакторный ANOVA тест, чтобы проверить влияние региона на индекс массы тела (BMI), используя второй способ, с помощью функции anova\_lm() из библиотеки statsmodels.

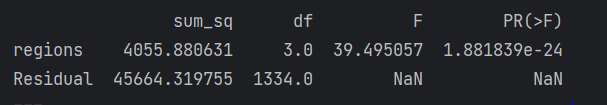


Рис. 3.3 – Результат теста с помощью функции anova\_lm()

3.3. С помощью t критерия Стьюдента перебрать все пары. Определить поправку Бонферрони. Сделать выводы.

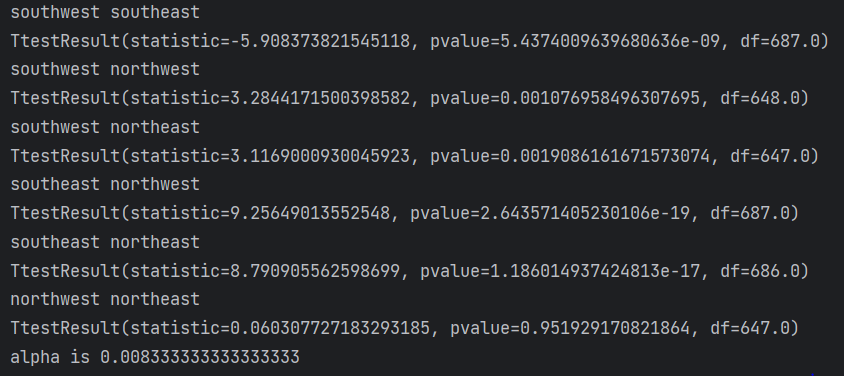


Рис. 3.4 – Результат перебора

Поправка Бонферрони равна 0.05/6 = 0.0083, следовательно статистически значимое влияние оказывают пары southwest – northwest и southwest – northeast.

3.4. Выполнить пост-хок тесты Тьюки и построить график.

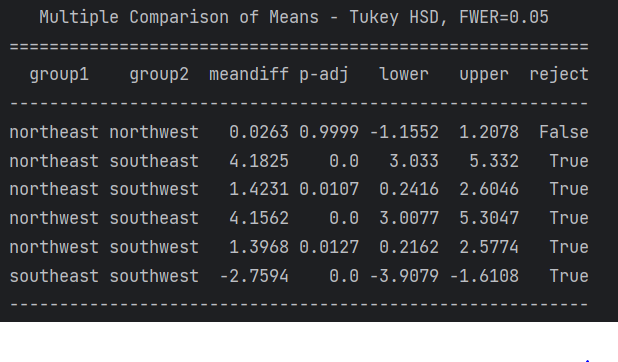


Рис. 3.5 – пост-хок тесты Тьюки

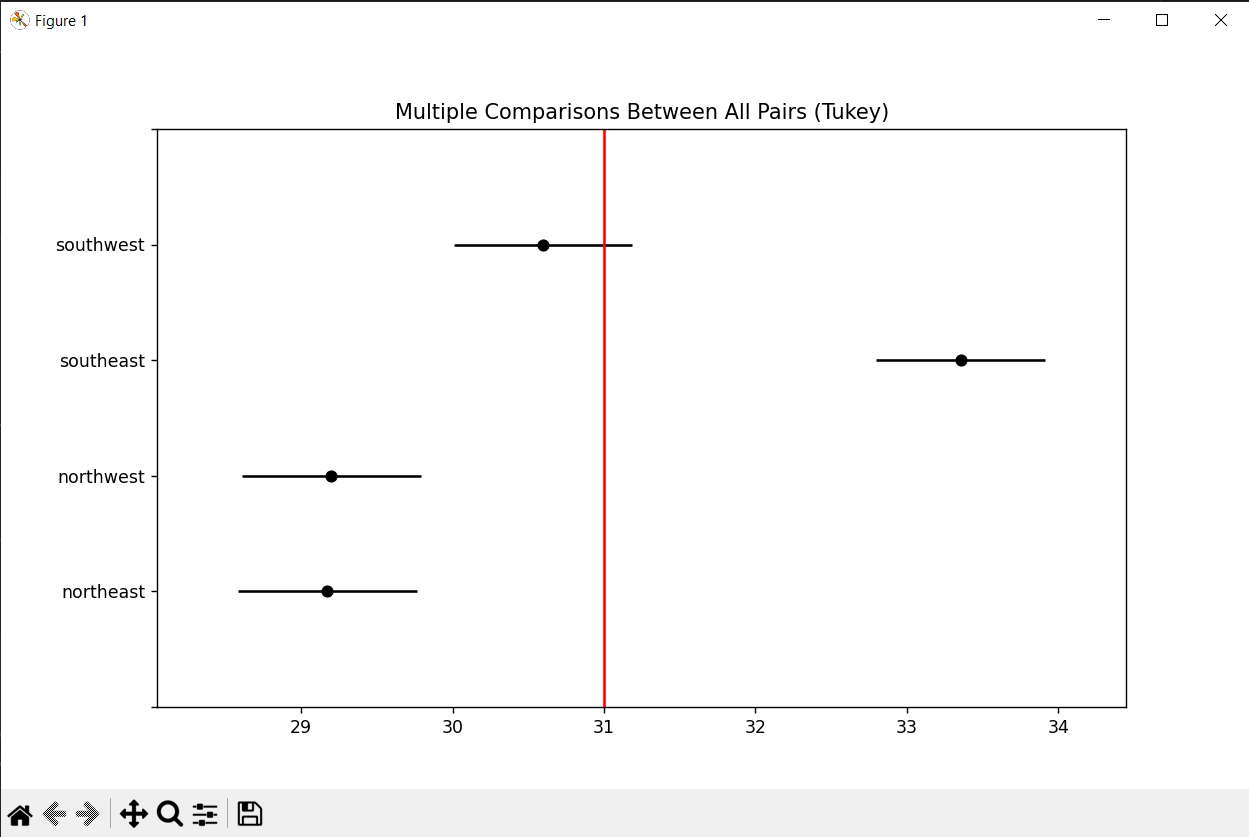


Рис. 3.6 – График

3.5. Выполнить двухфакторный ANOVA тест, чтобы проверить влияние региона и пола на индекс массы тела (BMI), используя функцию anova\_lm() из библиотеки statsmodels.

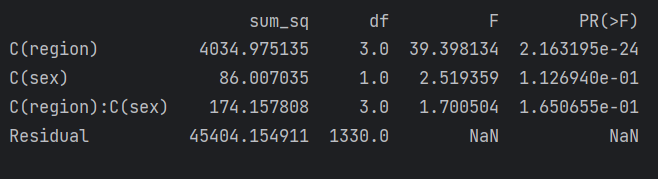


Рис. 3.7 – Результат двухфакторного теста

3.6. Выполнить пост-хок тесты Тьюки и построить график.

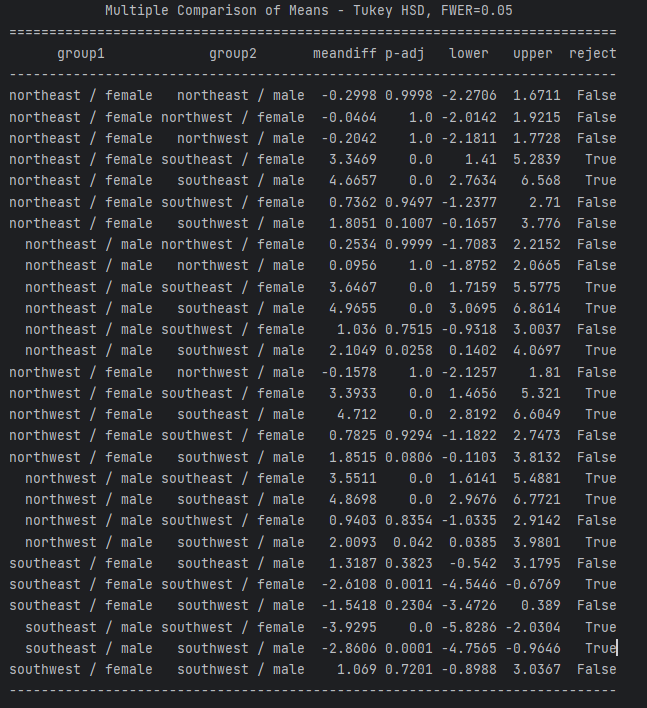


Рис. 3.8 – Пост-хок тест

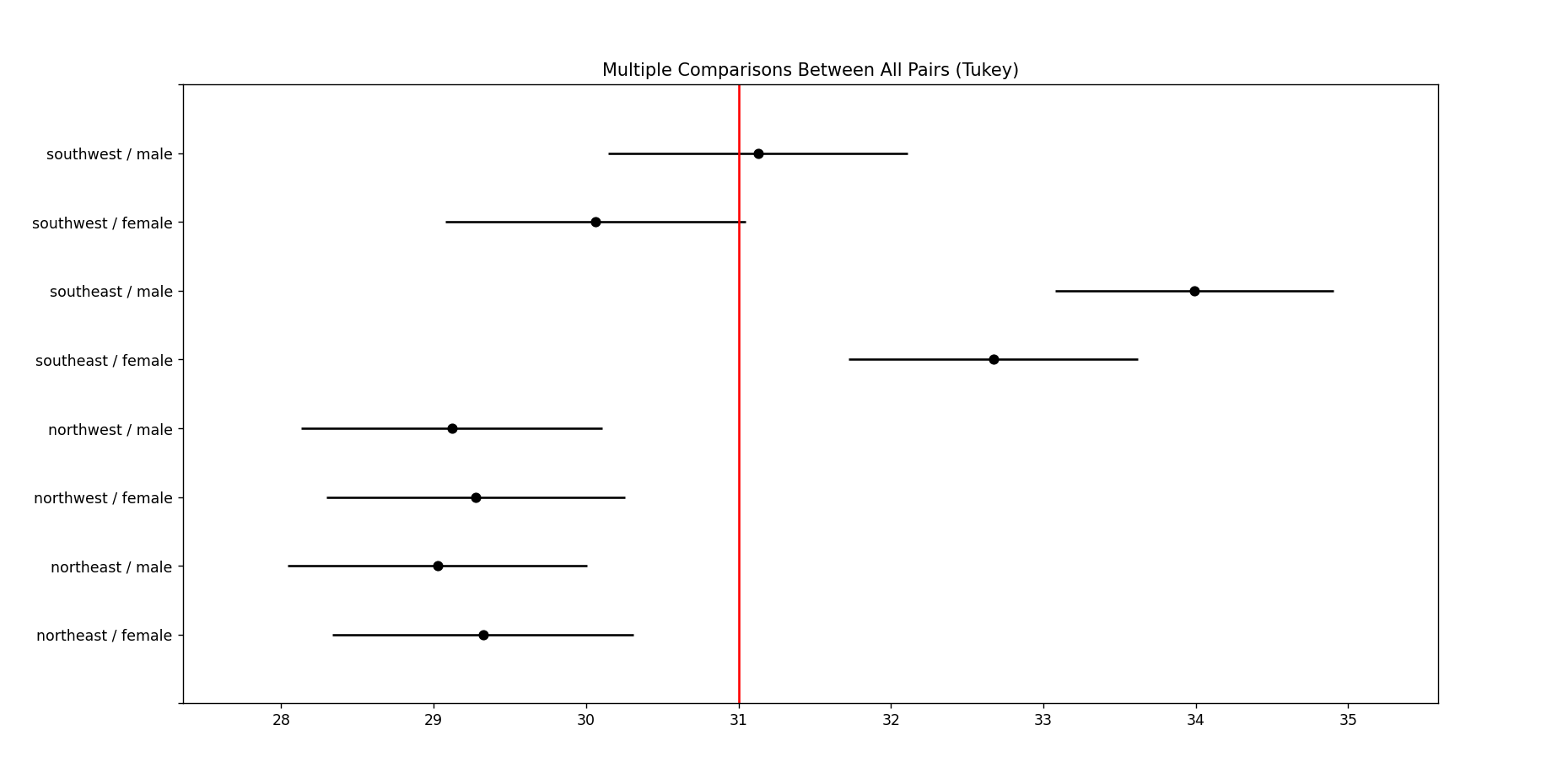


Рис. 3.9 – График

ПРИЛОЖЕНИЕ

Приложение 1 – 1.py

import numpy as np  
import matplotlib.pyplot as plt  
  
  
street = np.array([80, 98, 75, 91, 78], dtype=int)  
garage = np.array([100, 82, 105, 89, 102], dtype=int)  
  
corrcoef = np.corrcoef(street, garage)[0,1]  
print(corrcoef)  
  
plt.grid(True)  
plt.title("Диаграмма рассеяния",fontsize=20)  
plt.xlabel("Улица")  
plt.ylabel("Гараж")  
plt.scatter(street, garage)  
plt.show()

Приложение 2 – 2.py

import numpy as np  
import pandas as pd  
from sklearn.linear\_model import LinearRegression  
import os  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv("../second\_practice/winequality-red.csv")  
# print(df.info())  
# print(df.isnull().sum())  
  
corr\_matrix = df.corr().quality.to\_frame().round(2)  
corr\_matrix.style.background\_gradient(cmap='coolwarm')  
# print(corr\_matrix)  
y = df["quality"].values  
x = df["alcohol"].values  
  
X = df[["alcohol"]]  
# y = df["quality"]  
# x = np.array(X, type(float))  
# y = np.array(y, type(float))  
  
def mserror(x, w1, w0, y):  
 y\_pred = w1 \* x + w0  
 return np.sum((y - y\_pred)\*\*2) / len(y\_pred)  
  
  
def gr\_mserror(x, w1, w0, y):  
 y\_pred = w1 \* x + w0  
 return np.array([2 / len(x) \* np.sum((y-y\_pred)) \* -1, 2 / len(x) \*  
 np.sum((y - y\_pred) \* -x)])  
  
  
eps = 0.000001  
w1 = 0  
w0 = 0  
learning\_rate = 0.001  
next\_w1 = w1  
next\_w0 = w0  
n = 1000000  
for i in range(n):  
 cur\_w1 = next\_w1  
 cur\_w0 = next\_w0  
  
 next\_w0 = cur\_w0 - learning\_rate \* gr\_mserror(x, cur\_w1, cur\_w0, y)[0]  
 next\_w1 = cur\_w1 - learning\_rate \* gr\_mserror(x, cur\_w1, cur\_w0, y)[1]  
  
 # print(f"iter {i}")  
 # print(f"curr point {cur\_w1, cur\_w0}")  
  
 if (abs(cur\_w1 - next\_w1) <= eps) and (abs(cur\_w0 - next\_w0) <= eps):  
 break  
  
model = LinearRegression()  
  
model.fit(X, y)  
  
# print(model.coef\_, model.intercept\_)  
  
fig = plt.figure(figsize=(10, 6))  
  
  
our\_y = next\_w1 \* X + next\_w0

Продлжение – 2.py

print(f"наклон: {next\_w0}, сдвиг: {next\_w1}")  
print(f"mse: {mserror(x, next\_w1, next\_w0, y)}")  
x = np.arange(0, 40)  
model\_a = model.coef\_[0]  
model\_b = model.intercept\_  
sklearn\_y = model\_a \* X + model\_b  
# plt.plot(X, sklearn\_y, color="r")  
plt.plot(X, our\_y, "--g")  
plt.scatter(X, y, alpha=0.7)  
plt.show()

Приложение 3 – 3.py

import matplotlib.pyplot as plt  
import pandas as pd  
import scipy.stats as stats  
import statsmodels.api as sm  
from statsmodels.formula.api import ols  
from statsmodels.stats.multicomp import pairwise\_tukeyhsd  
  
  
df = pd.read\_csv("insurance.csv")  
print(df.info())  
uniques = df.region.unique()  
print(uniques)  
  
frame = pd.DataFrame({"regions": df.region.values, "bmi": df.bmi.values})  
groups = frame.groupby("regions").groups  
southwest = (df["bmi"].loc[df["region"] == uniques[0]])  
southeast = df["bmi"].loc[df["region"] == uniques[1]]  
northwest = df["bmi"].loc[df["region"] == uniques[2]]  
northeast = df["bmi"].loc[df["region"] == uniques[3]]  
  
f\_oneway = stats.f\_oneway(southwest, southeast, northwest, northeast)  
print(f\_oneway)  
print("===")  
model = ols("bmi ~ regions", data= frame).fit()  
anova\_result = sm.stats.anova\_lm(model, typ= 2)  
print(anova\_result)  
print("===")  
reg\_pairs = []  
for reg1 in range(3):  
 for reg2 in range(reg1 + 1, 4):  
 reg\_pairs.append((uniques[reg1], uniques[reg2]))  
  
count = 0  
for reg1, reg2 in reg\_pairs:  
 count += 1  
 print(reg1, reg2)  
 print(stats.ttest\_ind(frame.bmi[groups[reg1]], frame.bmi[groups[reg2]]))  
  
alpha = 0.05 / count  
  
print(f"alpha is {alpha}")  
print("===")  
  
tukey = pairwise\_tukeyhsd(endog=df.bmi, groups=df.region, alpha=0.05)  
tukey.plot\_simultaneous()  
plt.vlines(x=31, ymin=-10, ymax=10, color='red')  
print(tukey.summary())  
plt.show()  
  
frame = pd.DataFrame({'region': df.region, 'bmi': df.bmi, 'sex': df.sex})  
model = ols('bmi ~ C(region) + C(sex) + C(region):C(sex)', data=frame).fit()  
anova = sm.stats.anova\_lm(model, typ=2)  
print(anova)  
  
df['combination'] = df.region + ' / ' + df.sex  
  
tukey = pairwise\_tukeyhsd(endog=df['bmi'], groups=df['combination'], alpha=0.05)  
tukey.plot\_simultaneous()  
plt.vlines(x=31, ymin=-10, ymax=10, color='red')  
print(tukey.summary())  
plt.show()