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

|  |  |
| --- | --- |
| **Отчет по четвертой практической работе.** | |
| **по дисциплине** | |
| **«Технологии и инструментарий анализа больших данных»** | |
|  | |
| Выполнил студент группы ИКБО- 22-20 | Никулин К.В. |
| Принял | Парамонов А.А. |

Москва 2023 г.

СОДЕРЖАНИЕ

[1. КОРРЕЛЯЦИЯ 3](#_Toc150361538)

[1.1. Выгрузка данных 3](#_Toc150361539)

[1.2. Нахождение корреляции 3](#_Toc150361540)

[1.3. Диаграмма рассеяния 4](#_Toc150361541)

[2. ЛИНЕЙНАЯ РЕГРЕССИЯ 5](#_Toc150361542)

[2.1. Выгрузка данных 5](#_Toc150361543)

[2.2. Корреляционная матрица 6](#_Toc150361544)

[2.3. Реализация линейной регрессии 6](#_Toc150361545)

[2.4. Визуализация линейной регрессии 7](#_Toc150361546)

[3. ДИСПЕРСИОННЫЙ АНАЛИЗ 8](#_Toc150361547)

[3.1. Выгрузка данных 8](#_Toc150361548)

[3.2. Однофакторный ANOVA тест – библиотека Scipy 9](#_Toc150361549)

[3.3. Однофакторный ANOVA тест – библиотека statsmodels 9](#_Toc150361550)

[3.4. T-Критерий Стьюдента и поправка Бонферрони 10](#_Toc150361551)

[3.5. Пост-хок тесты Тьюки 11](#_Toc150361552)

[3.6. Двухфакторный ANOVA тест 11](#_Toc150361553)

[3.7. Пост-хок тесты Тьюки 12](#_Toc150361554)

1. КОРРЕЛЯЦИЯ
   1. Выгрузка данных

Данные представляют собой число автомобилей, припаркованных в течении 5 рабочих дней у бизнес-центра на уличной стоянке и в подземном гараже.

Таблица 1 – данные о припаркованных автомобилях

|  |  |  |
| --- | --- | --- |
| День | Улица | Гараж |
| Понедельник | 80 | 100 |
| Вторник | 98 | 82 |
| Среда | 75 | 105 |
| Четверг | 91 | 89 |
| Пятница | 78 | 102 |

* 1. Нахождение корреляции



**Рисунок 1 – Нахождение корреляции**

Коэффициент корреляции практически равен –1, что означает сильную отрицательную корреляцию.

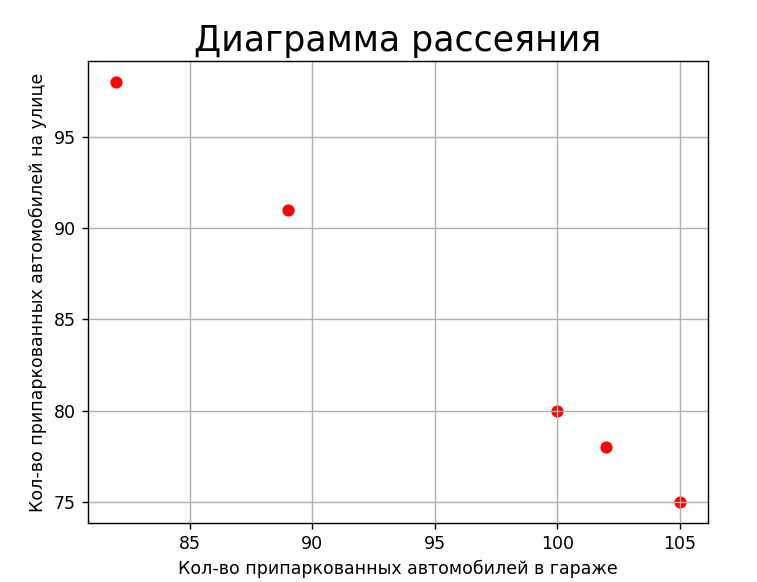
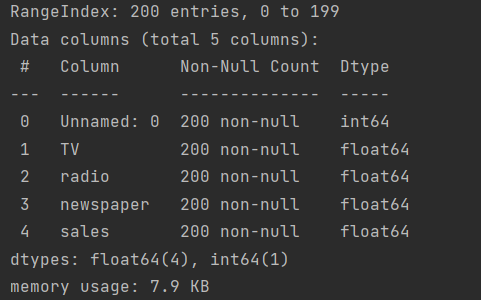


Рисунок 2 – Диаграмма рассеяния

1. ЛИНЕЙНАЯ РЕГРЕССИЯ
   1. Выгрузка данных

Используемый датасет Advertising.csv представляет собой данные о рекламах, имеет 5 параметров и 200 строк.

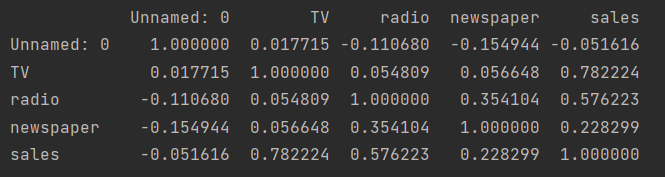


**Рисунок 3 – датасет Advertising.csv**

* 1. Корреляционная матрица

Построим корреляционную матрицу.

По корреляционной матрице (рис. 4) видно, что наибольшую корреляцию имеет параметр TV с параметром sales.



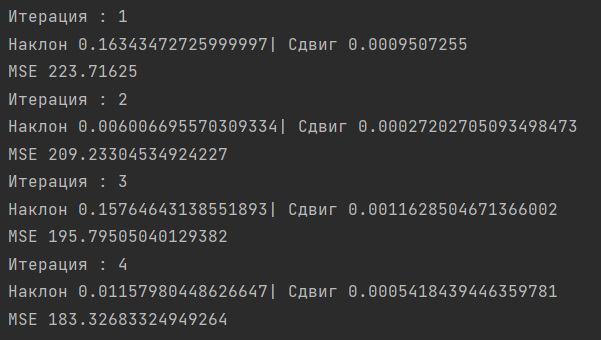
**Рисунок 4 – Корреляционная матрица**

* 1. Визуализация линейной регрессии

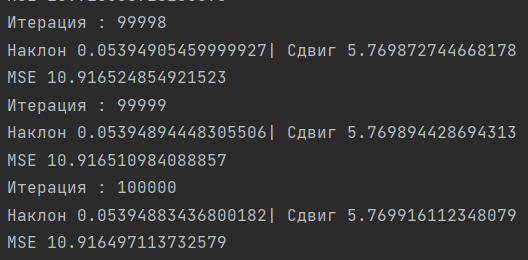
Отобразим модель линейной регрессии на графике после обучения (при 100 000 итераций и коэффициентом обучения 0.0000339).

Также выведем коэффициенты b1(наклон), b0(смещения) и mse(среднеквадратическая ошибка).

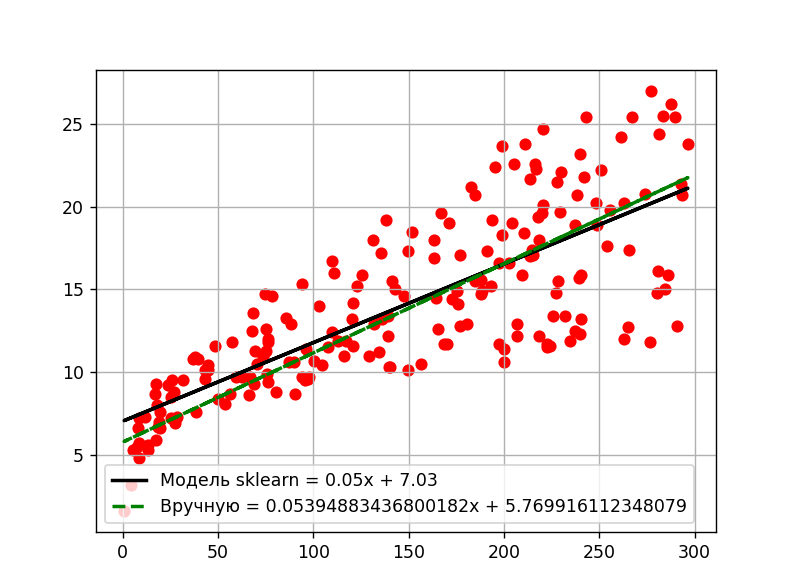
Для сравнения результатов также построим модель линейной регрессии от sklearn.



**Рисунок 5 – Первые итерации**

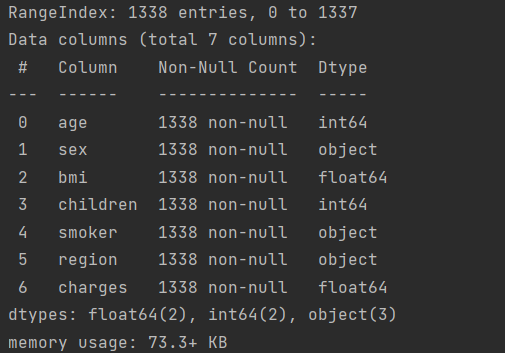


**Рисунок 6 – Последние итерации**



**Рисунок 7 – Визуализация линейной регрессии**

1. ДИСПЕРСИОННЫЙ АНАЛИЗ
   1. Выгрузка данных



**Рисунок 8 – данные insurance.csv**

Отобразим уникальные значения параметра region:



**Рисунок 9 – уникальные значения параметра region**

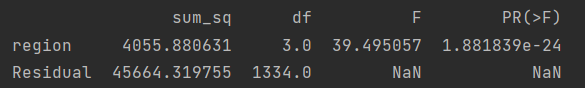
* 1. Однофакторный ANOVA тест – библиотека Scipy



**Рисунок 10 – Однофакторный ANOVA test Scipy**

Поскольку pvalue меньше 0.05 фактор регион оказывает статистически значимое влияние на фактор bmi

* 1. Однофакторный ANOVA тест – библиотека statsmodels

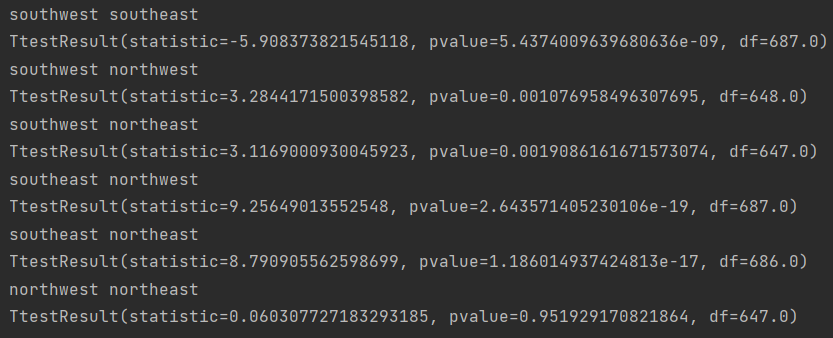


**Рисунок 11 – Однофакторный ANOVA test statsmodels**

Получаем значение pvalue такое-же как и в предыдущем тесте.

* 1. T-Критерий Стьюдента и поправка Бонферрони

Переберем все пары с помощью Т-критерия Стьюдента:

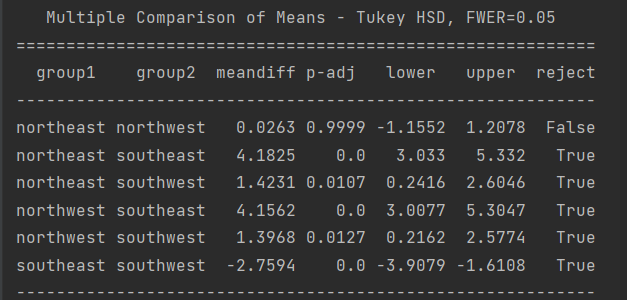


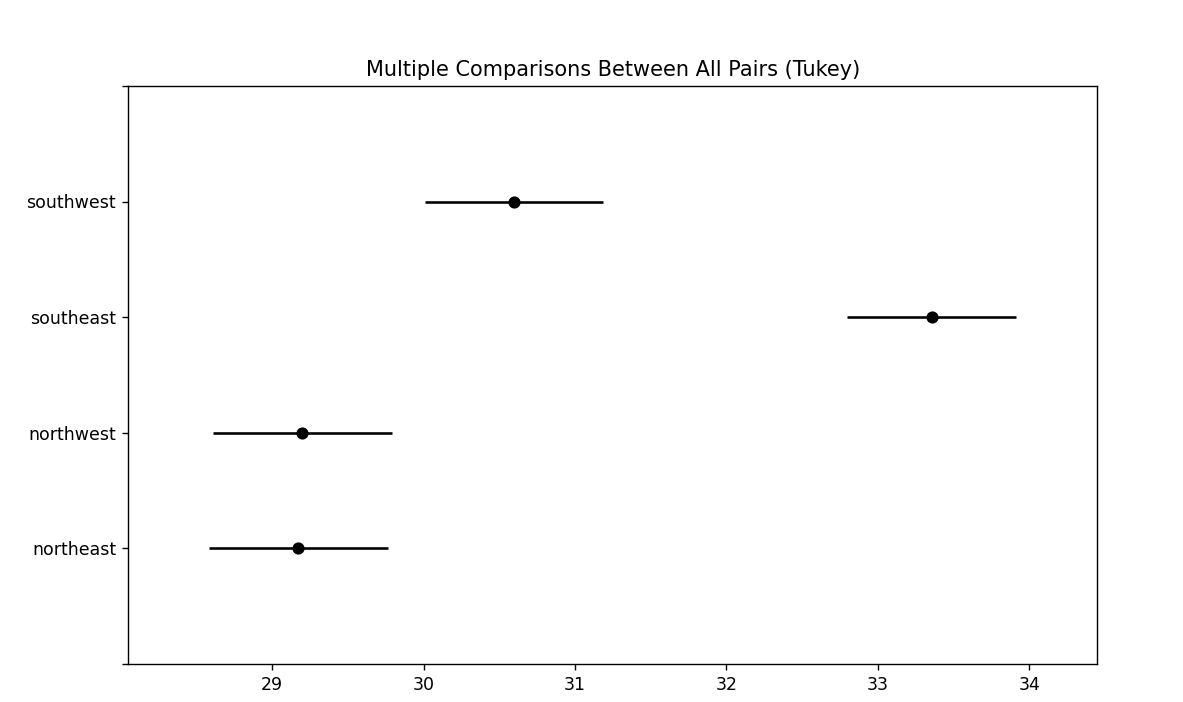
**Рисунок 12 – Т-Критерий Стьюдента**

Поправка Банферони = 0.05 / 6 = 0.008 (так как имеем 6 пар).

Значение pvalue меньше 0.008 во всех парах кроме последней, что значит только последняя пара не оказывает статистически важного влияния.

* 1. Пост-хок тесты Тьюки





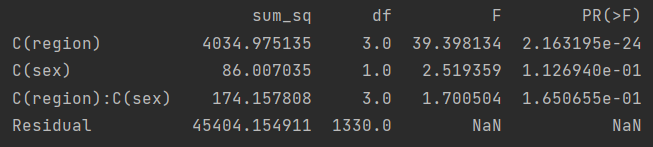
**Рисунок 13 – Пост-хок тесты Тьюки**

Пост-хок тесты Тьюки соответствуют полученным данным в предыдущих расчетах.

Только значение группы northeast и northwest не оказывает статистически важного влияния.

* 1. Двухфакторный ANOVA тест

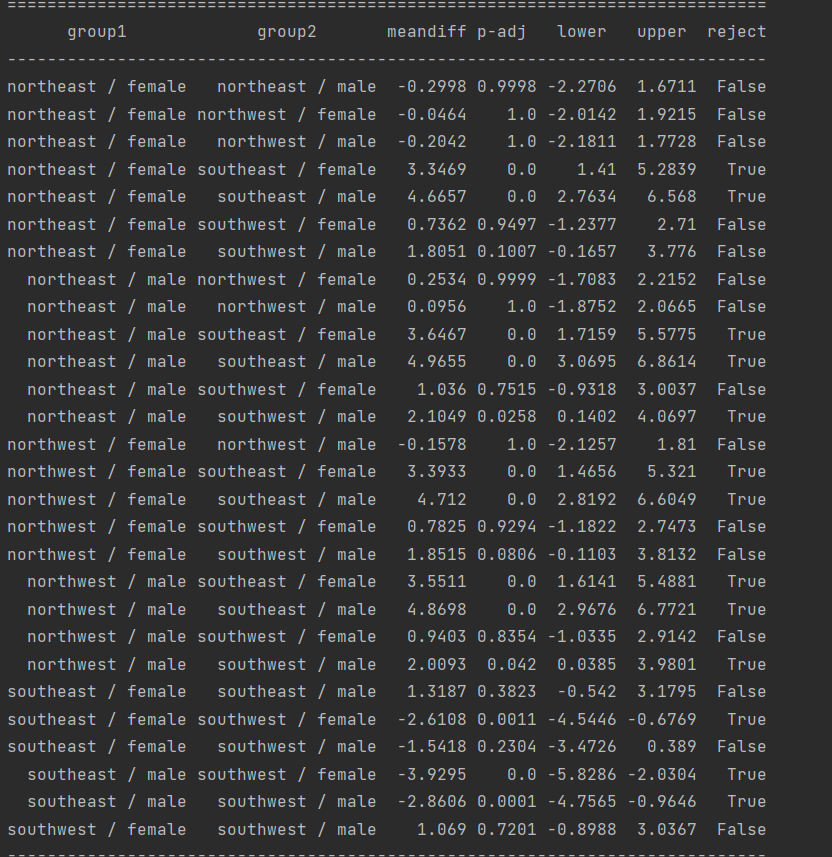
Выполним двухфакторный ANOVA тест чтобы проверить влияние региона и пола на индекс массы (bmi).



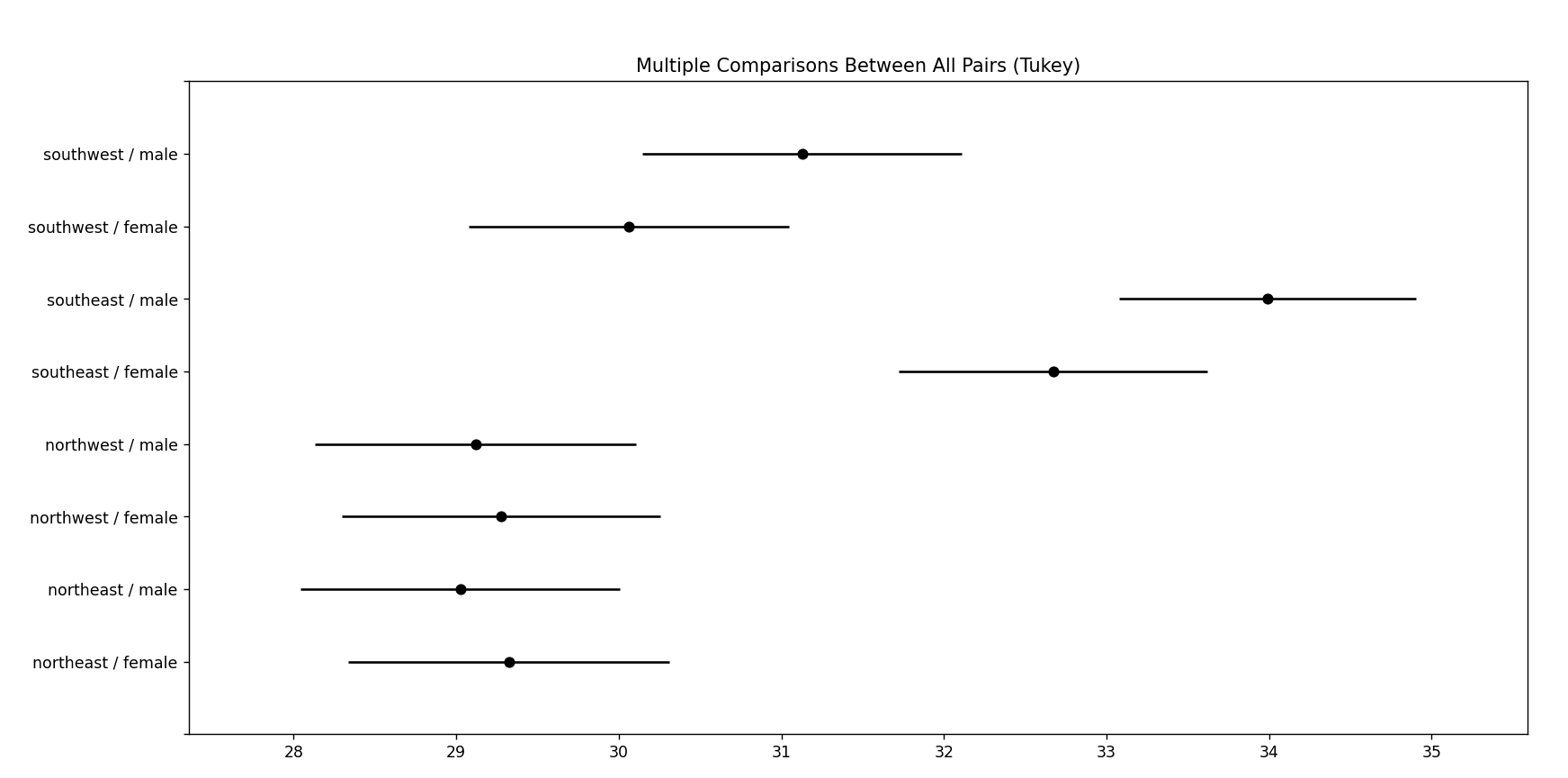
**Рисунок 14 – Двухфакторный ANOVA тест**

Все значения p-value <0.05, соответственно bmi зависит и от region и от sex.

* 1. Пост-хок тесты Тьюки



**Рисунок 15 – Пост-хок тесты Тьюки**



**Рисунок 16 – графическое отображение пост-хок тестов**

Приложение 1 – Код Корреляция

import matplotlib.pyplot as plt  
import numpy as np  
  
garage = np.array([100, 82, 105, 89, 102])  
street = np.array([80, 98, 75, 91, 78])  
print(f"Коэффициент корреляции {np.corrcoef(street, garage)[0, 1]}")  
  
plt.grid(True)  
plt.title("Диаграмма рассеяния", fontsize=20)  
plt.xlabel("Кол-во припаркованных автомобилей в гараже")  
plt.ylabel("Кол-во припаркованных автомобилей на улице")  
plt.scatter(garage, street, marker='o', color="red")  
plt.show()

Приложение 2 – Код линейная регрессия

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.linear\_model import LinearRegression  
  
  
def regression(df, epochs, learning\_rate):  
 x = np.array(df[['TV']])  
 y = np.array(df['sales'])  
 b0, b1 = 0, 0  
 for i in range(epochs):  
 f = b1 \* x[:, 0] + b0  
 gr\_b1 = 2 / len(x) \* np.sum((y - f) \* (-x[:, 0]))  
 gr\_b0 = 2 / len(x) \* np.sum(y - f) \* (-1)  
 b1 = b1 - learning\_rate \* gr\_b1  
 b0 = b0 - learning\_rate \* gr\_b0  
 mse = np.sum((y - f) \*\* 2) / len(x)  
 print(f"Итерация : {i + 1}")  
 print(f"Наклон {b1}| Сдвиг {b0}")  
 print(f"MSE {mse}")  
 # sklearn  
 model = LinearRegression()  
 model.fit(x, y)  
 model\_a = model.coef\_[0]  
 model\_b = model.intercept\_  
 model\_pred = model\_a \* x + model\_b  
  
 plt.scatter(x, y, marker='o', color="red")  
 plt.plot(x, model\_pred, linewidth=2, color='black', label=f'Модель sklearn = {model\_a:.2f}x + {model\_b:.2f}')  
 plt.plot(x, f, '--g', linewidth=2, label=f'Вручную = {b1}x + {b0}')  
 plt.grid(True)  
 plt.legend()  
 plt.show()  
 return  
  
  
def main():  
 df = pd.read\_csv("Advertising.csv")  
 print(df.info())  
 print(df.corr())  
 regression(df, 100000, 0.0000339)  
 return  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Приложение 3 – Код дисперсионный анализ

import matplotlib.pyplot as plt  
import numpy as np  
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  
  
  
def main():  
 df = pd.read\_csv("insurance.csv")  
 print(df.info())  
 regions = df.region.unique()  
 print(regions)  
  
 # # scipy ANOVA test 1  
 groups = df.groupby('region').groups  
 southwest = df['bmi'][groups['southwest']]  
 southeast = df['bmi'][groups['southeast']]  
 northwest = df['bmi'][groups['northwest']]  
 northeast = df['bmi'][groups['northeast']]  
 # print(stats.f\_oneway(southwest, southeast, northwest, northeast))  
  
 # # anova\_lm ANOVA test 2  
 # model = ols('bmi ~ region', data=df).fit()  
 # anova\_result = sm.stats.anova\_lm(model, typ=2)  
 # print(anova\_result)  
  
 # t-Student 3  
 region\_pairs = []  
 for reg1 in range(3):  
 for reg2 in range(reg1 + 1, 4):  
 region\_pairs.append((regions[reg1], regions[reg2]))  
 # for reg1, reg2 in region\_pairs:  
 # print(reg1, reg2)  
 # print(stats.ttest\_ind(df['bmi'][groups[reg1]], df['bmi'][groups[reg2]]))  
  
 # #post-hok 4  
 tukey = pairwise\_tukeyhsd(endog=df['bmi'], groups=df['region'], alpha=0.05)  
 tukey.plot\_simultaneous()  
 # print(tukey.summary())  
 # plt.show()  
  
 # ANOVA 5  
 model = ols('bmi ~ C(region) + C(sex) + C(region):C(sex)', data=df).fit()  
 anova\_result = sm.stats.anova\_lm(model, typ=2)  
 # print(anova\_result)  
  
 # post-hok 6  
 df['combination'] = df.region + " / " + df.sex  
 tukey = pairwise\_tukeyhsd(endog=df['bmi'], groups=df['combination'], alpha=0.05)  
 tukey.plot\_simultaneous()  
 print(tukey.summary())  
 # plt.show()  
 return  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()