МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра прикладной математики (ПМ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4

по дисциплине

«Технологии и инструментарий анализа больших данных»

Выполнил студент группы ИКБО-12-20 Колегов С. А.

Принял преподаватель Есипов Иван Владимирович

Практическая работа выполнена «__» ______ 2023 г.
(подпись студента)

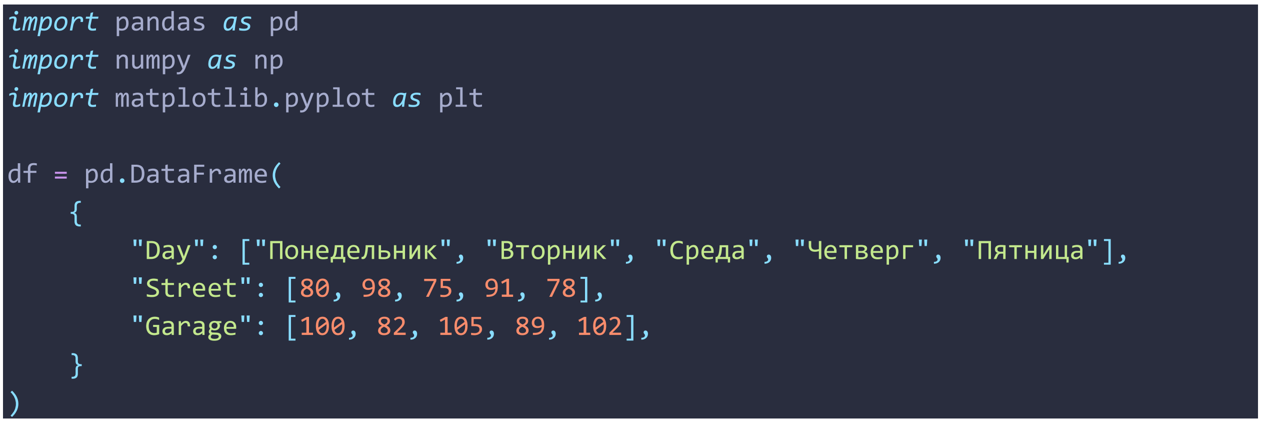
Зачтено «__» ______ 2023 г.

Москва 2023

ВЫПОЛНЕНИЕ РАБОТЫ

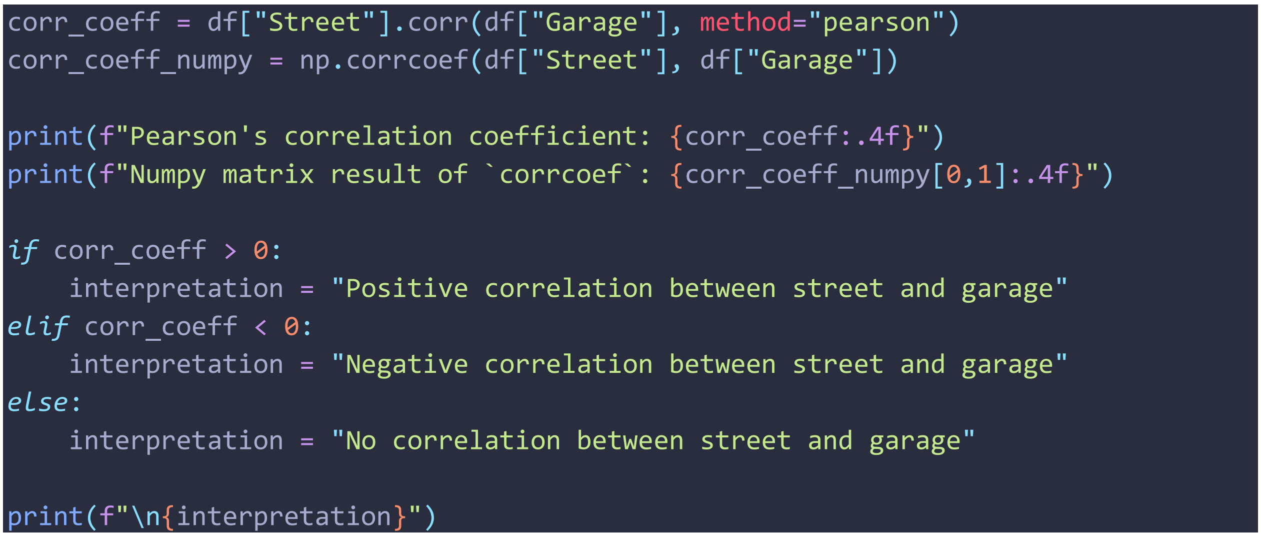
Часть 1

Определим дата-фрейм:

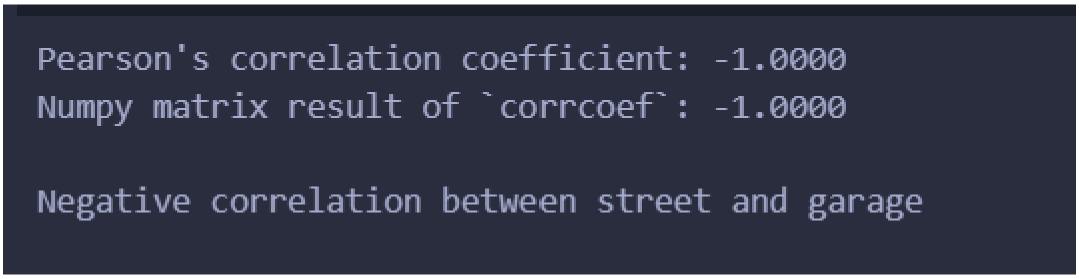


```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.DataFrame(
    {
        "Day": ["Понедельник", "Вторник", "Среда", "Четверг", "Пятница"],
        "Street": [80, 98, 75, 91, 78],
        "Garage": [100, 82, 105, 89, 102],
    }
)
```

Вычислим корреляционный коэф. Пирсона с помощью методов corr и numpy.corrcoef

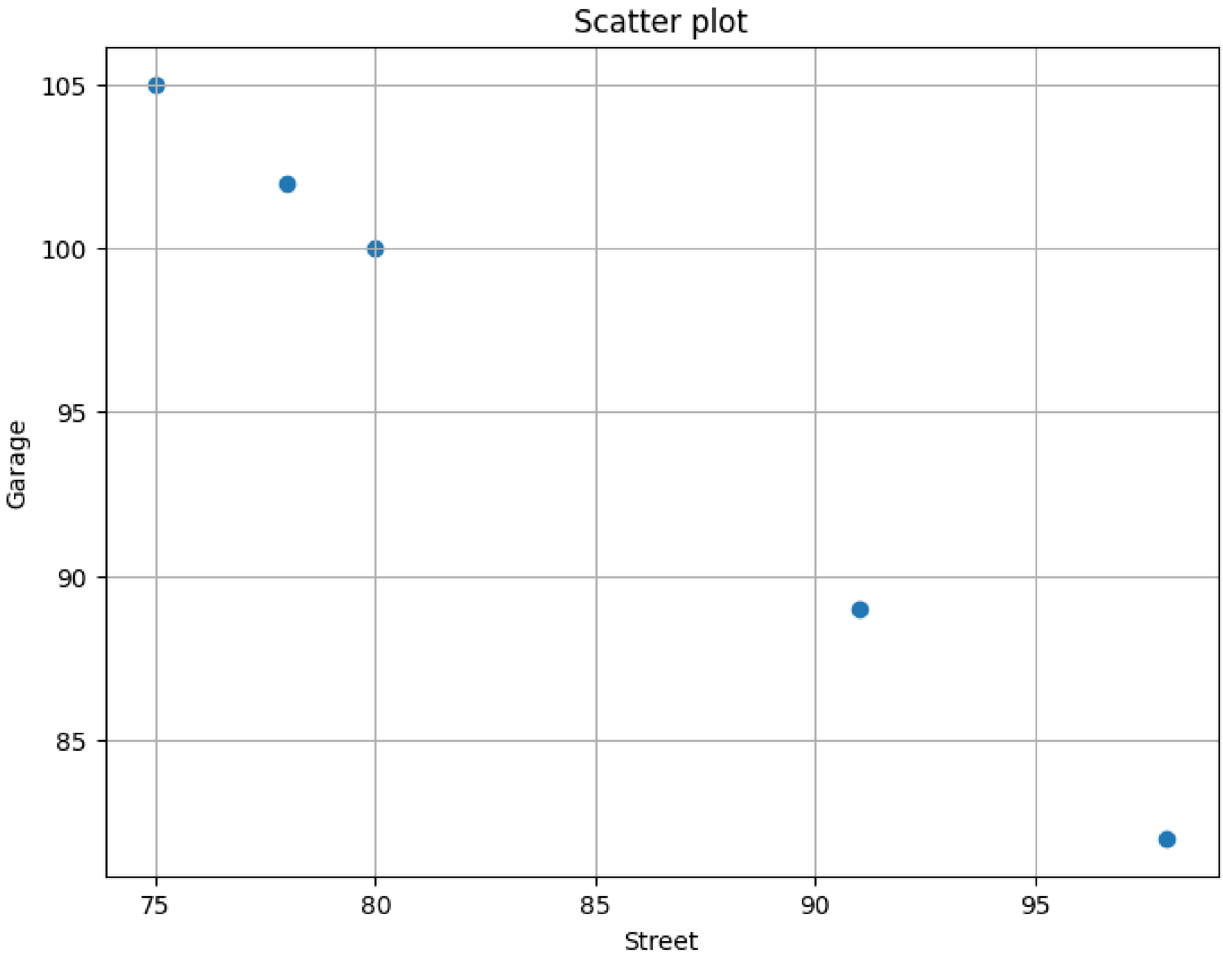


```python
corr_coeff = df["Street"].corr(df["Garage"], method="pearson")
corr_coeff_numpy = np.corrcoef(df["Street"], df["Garage"])

print(f"Pearson's correlation coefficient: {corr_coeff:.4f}")
print(f"Numpy matrix result of `corrcoef`: {corr_coeff_numpy[0,1]:.4f}")

if corr_coeff > 0:
    interpretation = "Positive correlation between street and garage"
elif corr_coeff < 0:
    interpretation = "Negative correlation between street and garage"
else:
    interpretation = "No correlation between street and garage"

print(f"\n{interpretation}")
```



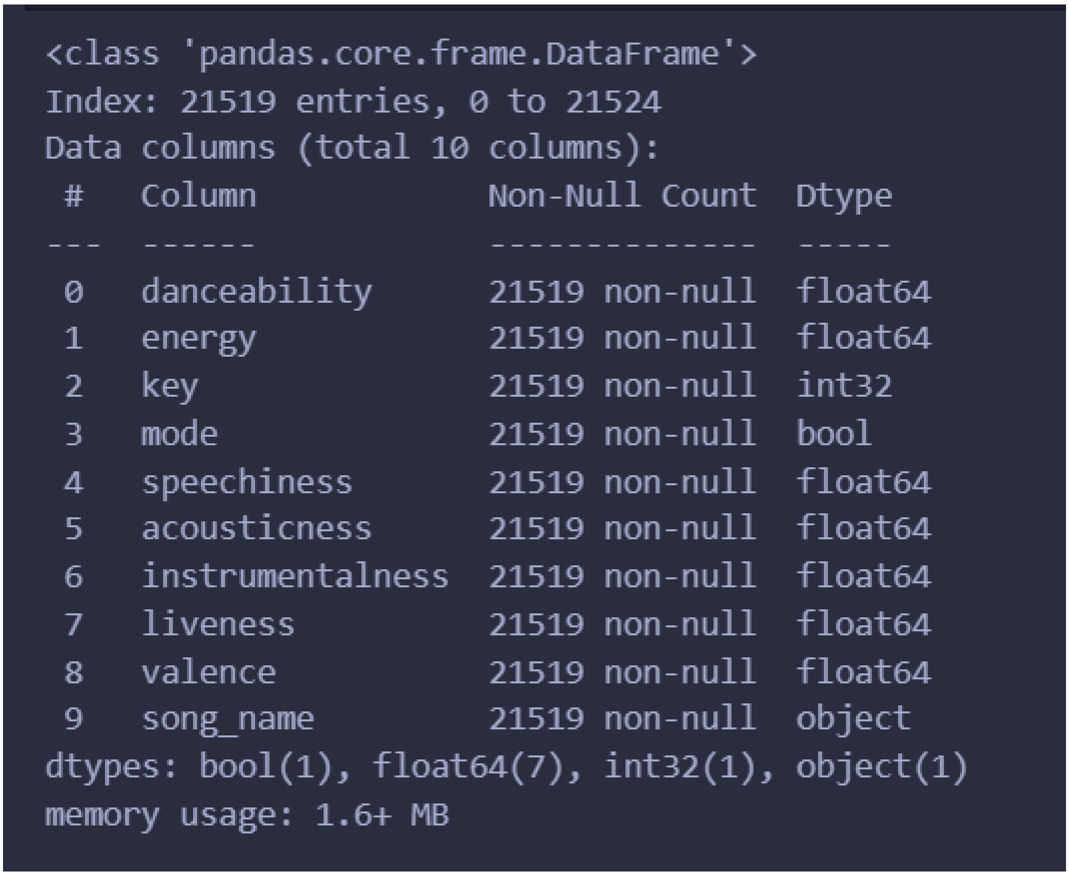
```
Pearson's correlation coefficient: -1.0000
Numpy matrix result of `corrcoef`: -1.0000

Negative correlation between street and garage
```

Построим график рассеивания:


Scatter plot
Garage
105
100
95
90
85
Street
75
80
85
90
95

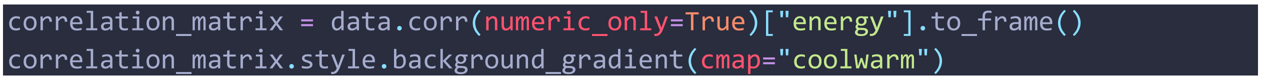
Часть 2

Для второй части был выбран датасет треков Spotity со следующими параметрами:

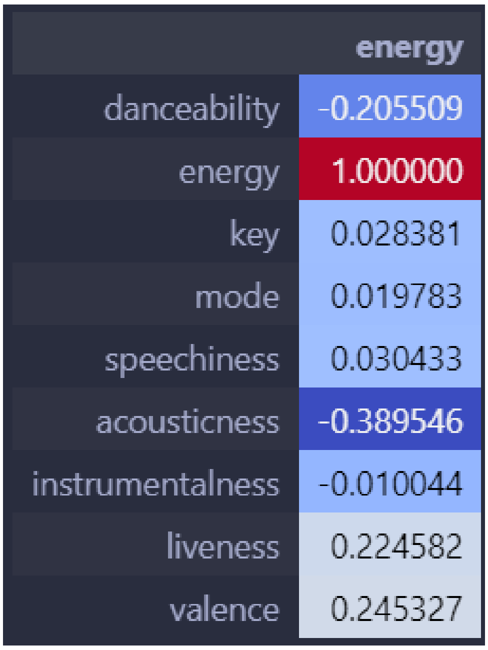


```
<class 'pandas.core.frame.DataFrame'>
Index: 21519 entries, 0 to 21524
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   danceability      21519 non-null  float64
 1   energy            21519 non-null  float64
 2   key               21519 non-null  int32  
 3   mode              21519 non-null  bool   
 4   speechiness       21519 non-null  float64
 5   acousticness      21519 non-null  float64
 6   instrumentalness  21519 non-null  float64
 7   liveness          21519 non-null  float64
 8   valence           21519 non-null  float64
 9   song_name         21519 non-null  object 
dtypes: bool(1), float64(7), int32(1), object(1)
memory usage: 1.6+ MB
```

Построим корреляционную матрицу для параметра «energy»:



```
correlation_matrix = data.corr(numeric_only=True)["energy"].to_frame()
correlation_matrix.style.background_gradient(cmap="coolwarm")
```



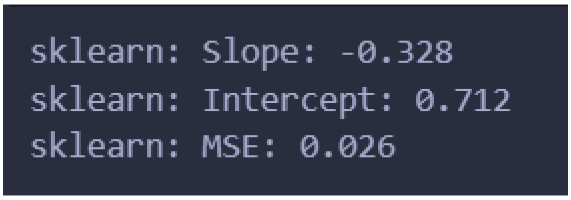
	energy
danceability	-0.205509
energy	1.000000
key	0.028381
mode	0.019783
speechiness	0.030433
acousticness	-0.389546
instrumentalness	-0.010044
liveness	0.224582
valence	0.245327

Выберем параметр «acousticness» как самый коррелирующий с «energy»:

```
X = np.array(data[["acousticness"]], type(float))
y = np.array(data["energy"], type(float))
```

Посчитаем наклон, сдвиг и MSE с помощью sklearn:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

model = LinearRegression()
model.fit(X, y)

print(f"sklearn: Slope: {model.coef_[0]:.3f}")
print(f"sklearn: Intercept: {model.intercept_:.3f}")
print(f"sklearn: MSE: {mean_squared_error(model.predict(X), y):.3f}")
```

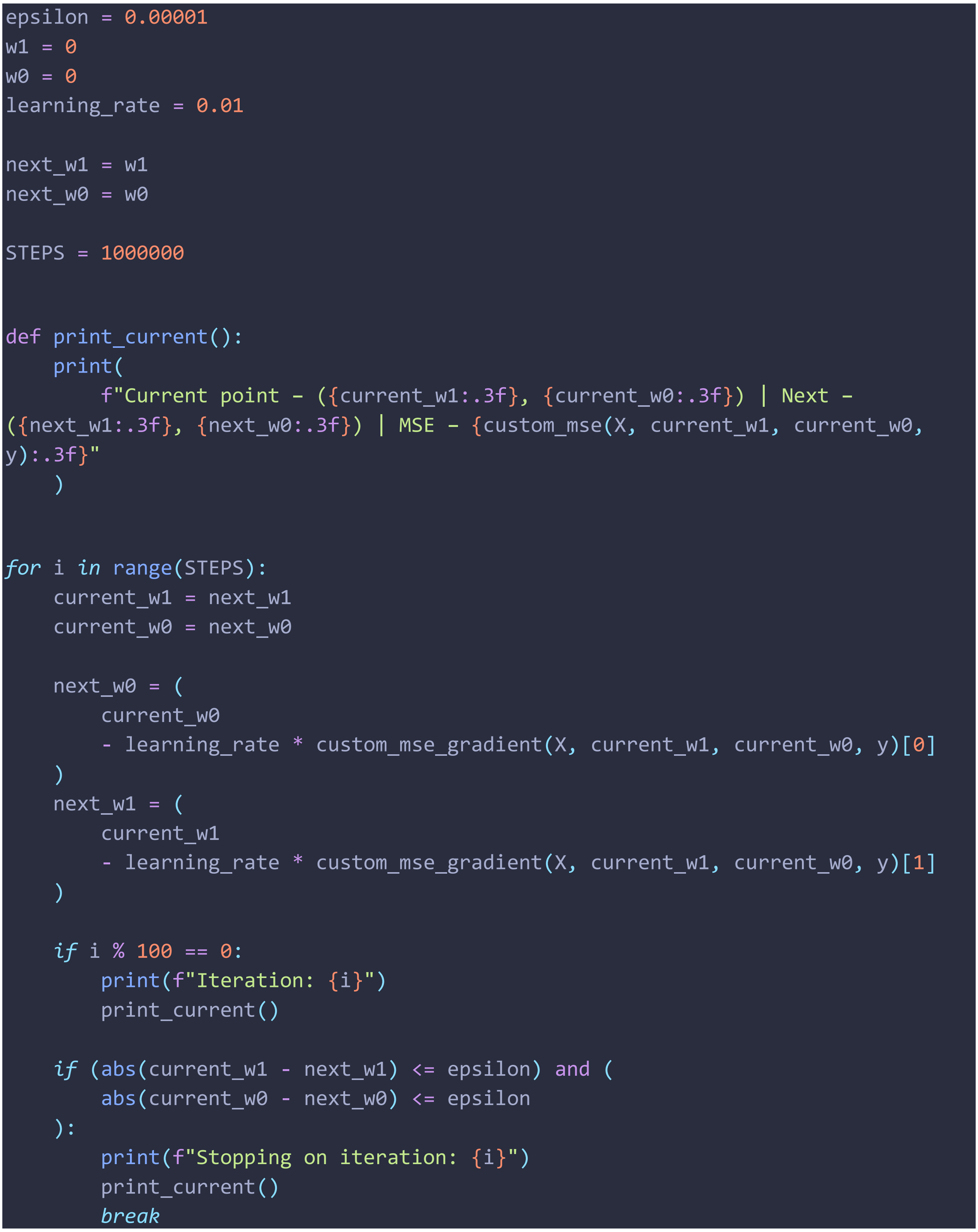


```
sklearn: Slope: -0.328
sklearn: Intercept: 0.712
sklearn: MSE: 0.026
```

Определим функции для ручного нахождения MSE и поиска градиента функции MSE.

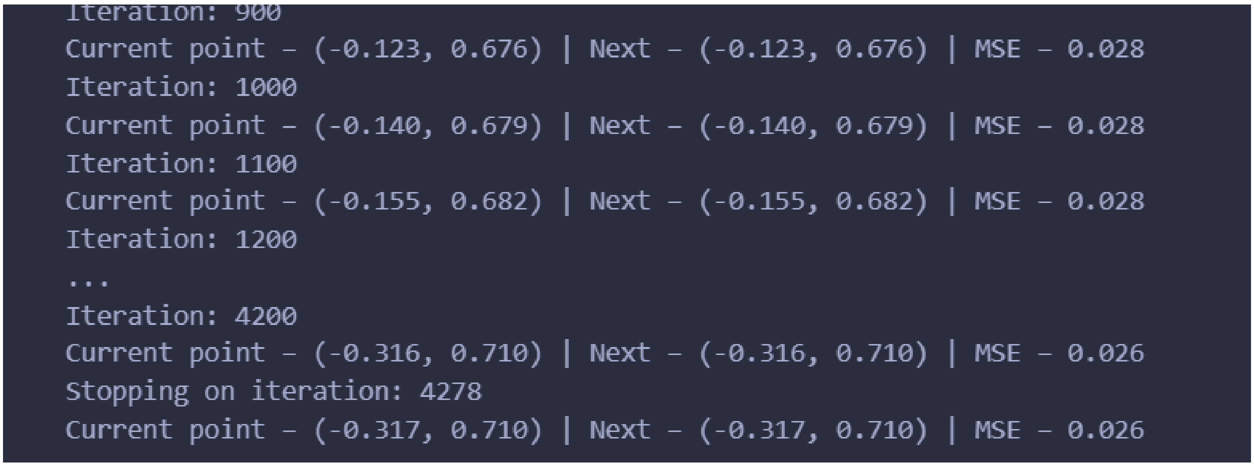
```
def custom_mse(X, w1, w0, y):
    y_pred = w1 * X[:, 0] + w0
    y_pred_len = len(y_pred)
    return np.sum((y - y_pred) ** 2) / y_pred_len


def custom_mse_gradient(X, w1, w0, y):
    y_pred = w1 * X[:, 0] + w0
    x_len = len(X)
    return np.array(
        [
            np.sum(y - y_pred) * -1 * 2 / x_len,
            np.sum((y - y_pred) * -X[:, 0]) * 2 / x_len,
        ]
    )
```

Посчитаем с помощью цикла сдвиг, наклон и ошибку:



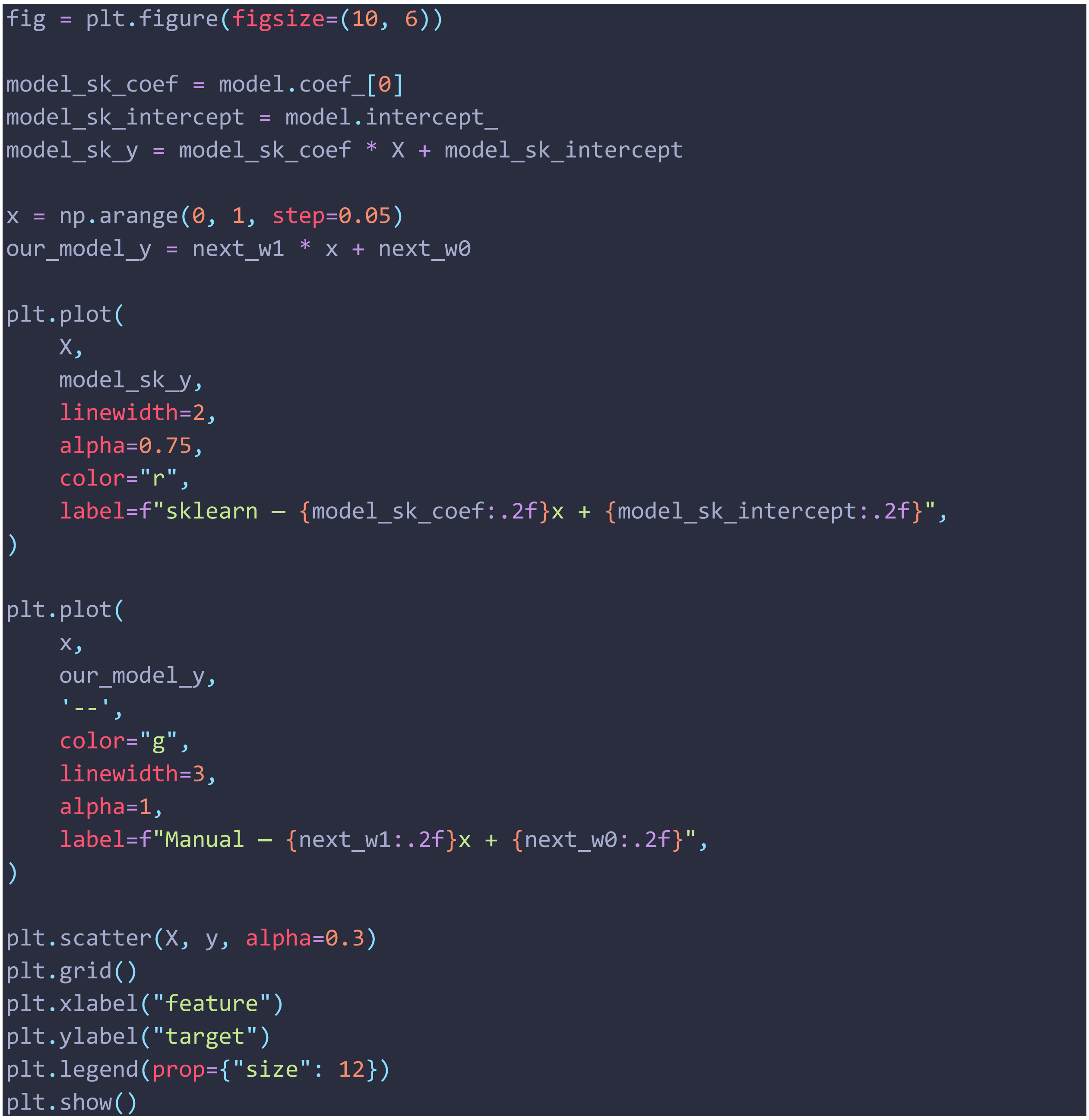
```
epsilon = 0.00001
w1 = 0
w0 = 0
learning_rate = 0.01

next_w1 = w1
next_w0 = w0

STEPS = 1000000


def print_current():
    print(
        f"Current point – ({current_w1:.3f}, {current_w0:.3f}) | Next –
({next_w1:.3f}, {next_w0:.3f}) | MSE – {custom_mse(X, current_w1, current_w0,
y):.3f}"
    )


for i in range(STEPS):
    current_w1 = next_w1
    current_w0 = next_w0

    next_w0 = (
        current_w0
        - learning_rate * custom_mse_gradient(X, current_w1, current_w0, y)[0]
    )
    next_w1 = (
        current_w1
        - learning_rate * custom_mse_gradient(X, current_w1, current_w0, y)[1]
    )

    if i % 100 == 0:
        print(f"Iteration: {i}")
        print_current()

    if (abs(current_w1 - next_w1) <= epsilon) and (
        abs(current_w0 - next_w0) <= epsilon
    ):
        print(f"Stopping on iteration: {i}")
        print_current()
        break
```

Количество пройденных итераций – 4278:

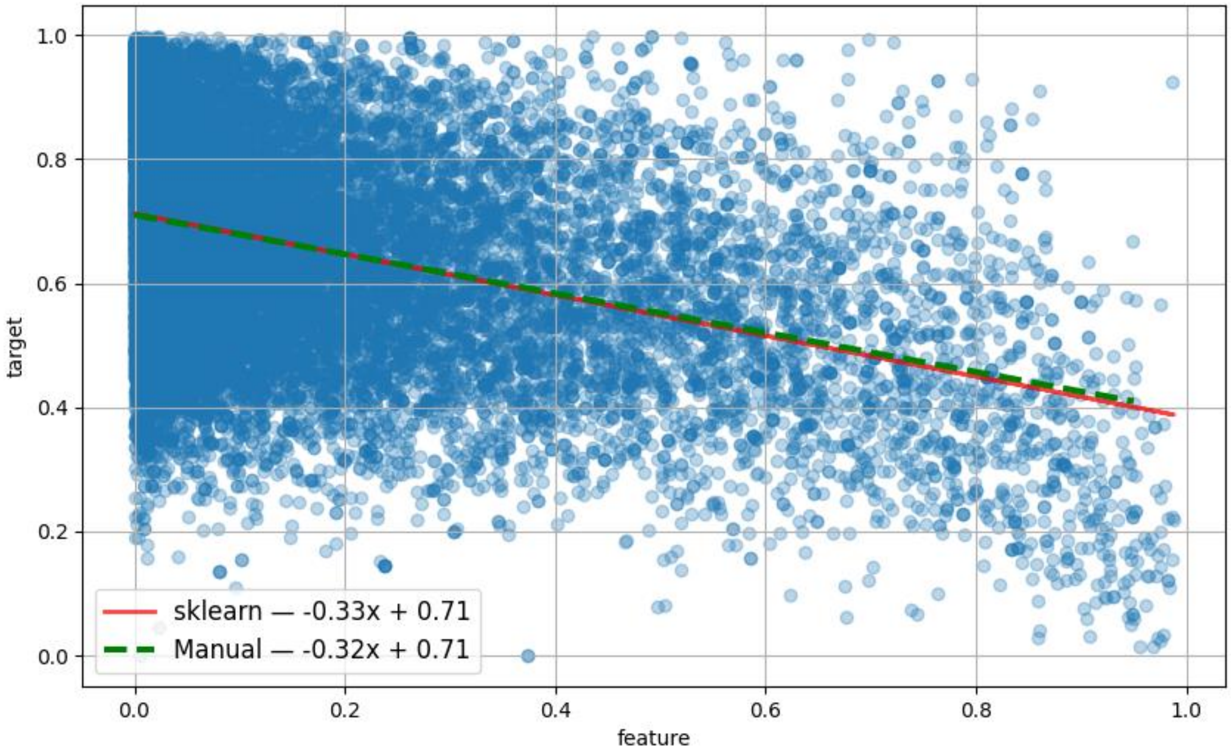


```
    Iteration: 900
    Current point – (-0.123, 0.676) | Next – (-0.123, 0.676) | MSE – 0.028
    Iteration: 1000
    Current point – (-0.140, 0.679) | Next – (-0.140, 0.679) | MSE – 0.028
    Iteration: 1100
    Current point – (-0.155, 0.682) | Next – (-0.155, 0.682) | MSE – 0.028
    Iteration: 1200
    ...
    Iteration: 4200
    Current point – (-0.316, 0.710) | Next – (-0.316, 0.710) | MSE – 0.026
    Stopping on iteration: 4278
    Current point – (-0.317, 0.710) | Next – (-0.317, 0.710) | MSE – 0.026
```

«Ручной» наклон – -0.317, сдвиг – 0.710, MSE – 0.026. Визуализация с графиками:

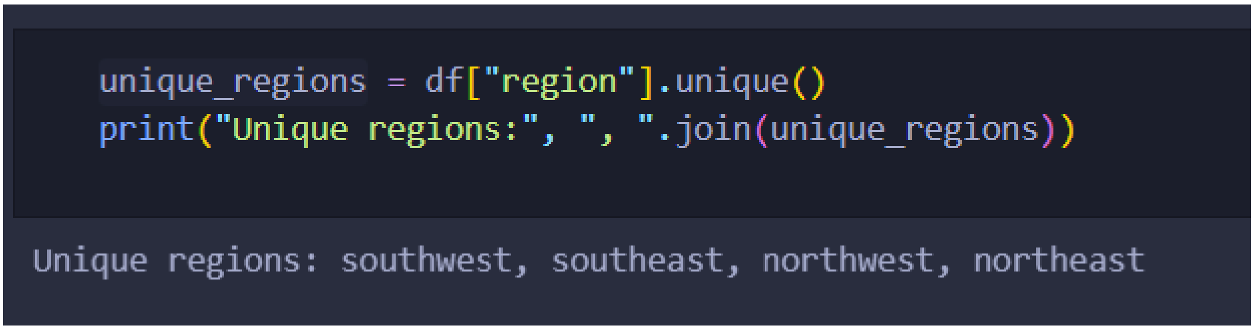


```
fig = plt.figure(figsize=(10, 6))

model_sk_coef = model.coef_[0]
model_sk_intercept = model.intercept_
model_sk_y = model_sk_coef * X + model_sk_intercept

x = np.arange(0, 1, step=0.05)
our_model_y = next_w1 * x + next_w0

plt.plot(
    X,
    model_sk_y,
    linewidth=2,
    alpha=0.75,
    color="r",
    label=f"sklearn — {model_sk_coef:.2f}x + {model_sk_intercept:.2f}",
)

plt.plot(
    x,
    our_model_y,
    '--',
    color="g",
    linewidth=3,
    alpha=1,
    label=f"Manual — {next_w1:.2f}x + {next_w0:.2f}",
)

plt.scatter(X, y, alpha=0.3)
plt.grid()
plt.xlabel("feature")
plt.ylabel("target")
plt.legend(prop={"size": 12})
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
target
0.0
0.2
0.4
0.6
0.8
1.0
feature
sklearn — -0.33x + 0.71
Manual — -0.32x + 0.71

Часть 3

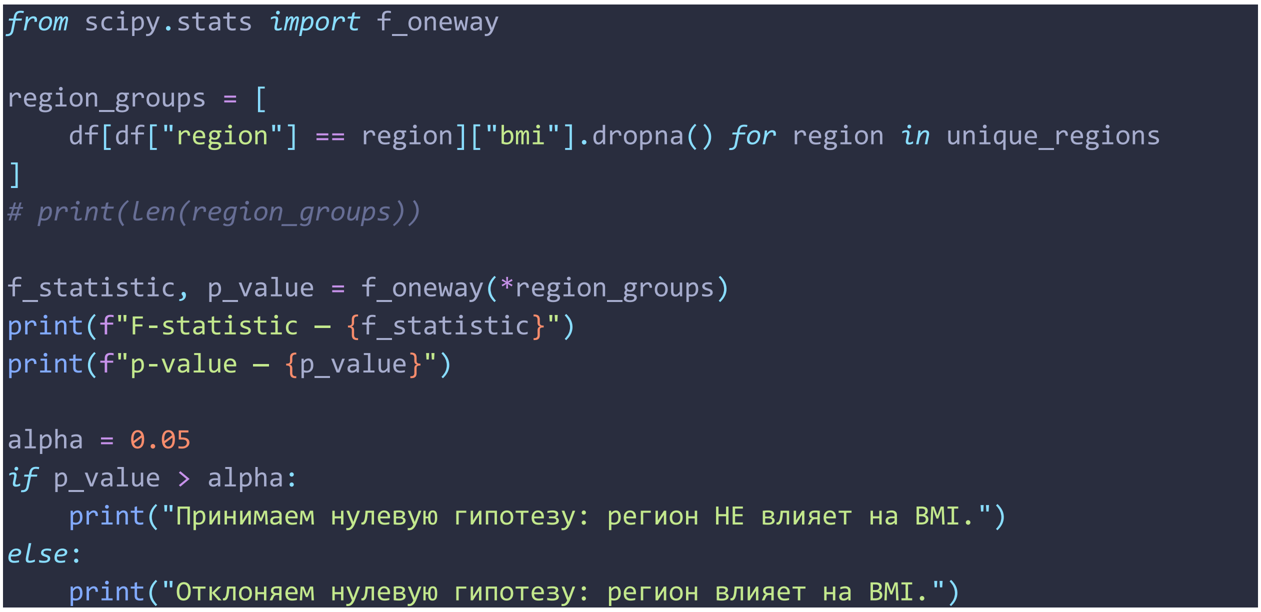
Найдём список уникальных регионов:



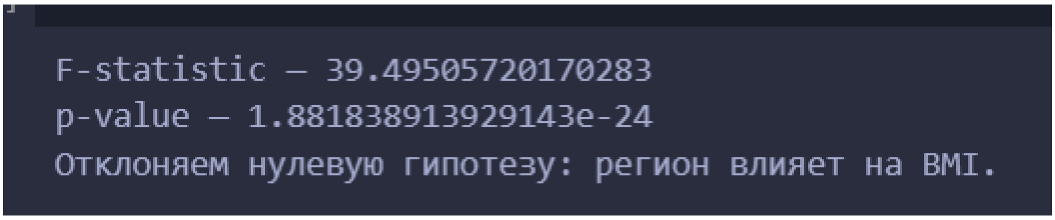
```python
unique_regions = df["region"].unique()
print("Unique regions:", ", ".join(unique_regions))
```

```
Unique regions: southwest, southeast, northwest, northeast
```

Выполним однофакторный ANOVA-тест с помощью Scipy:

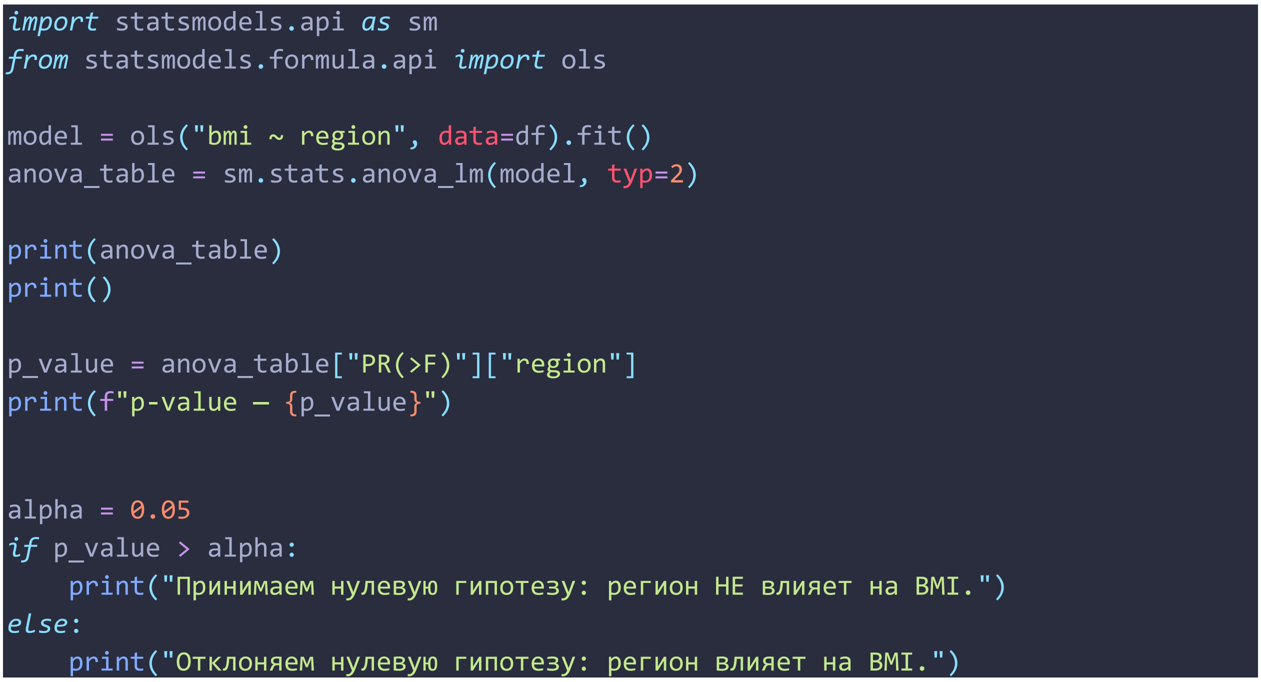


```python
from scipy.stats import f_oneway

region_groups = [
    df[df["region"] == region]["bmi"].dropna() for region in unique_regions
]
# print(len(region_groups))

f_statistic, p_value = f_oneway(*region_groups)
print(f"F-statistic – {f_statistic}")
print(f"p-value – {p_value}")

alpha = 0.05
if p_value > alpha:
    print("Принимаем нулевую гипотезу: регион НЕ влияет на BMI.")
else:
    print("Отклоняем нулевую гипотезу: регион влияет на BMI.")
```

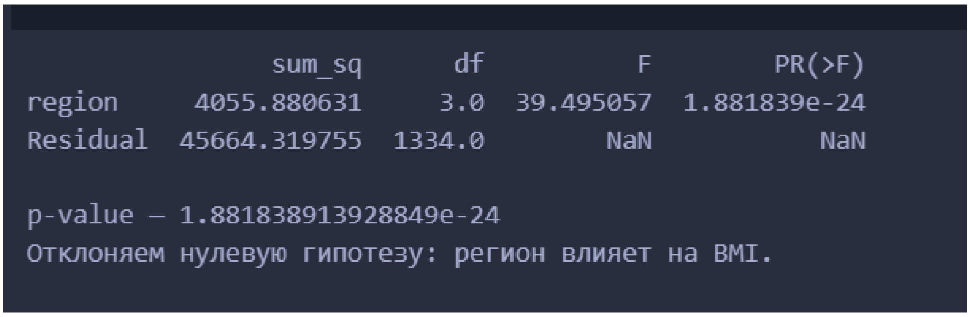


```
F-statistic – 39.49505720170283
p-value – 1.881838913929143e-24
Отклоняем нулевую гипотезу: регион влияет на BMI.
```

Выполним однофакторный ANOVA с помощью statsmodels.anova_lm:

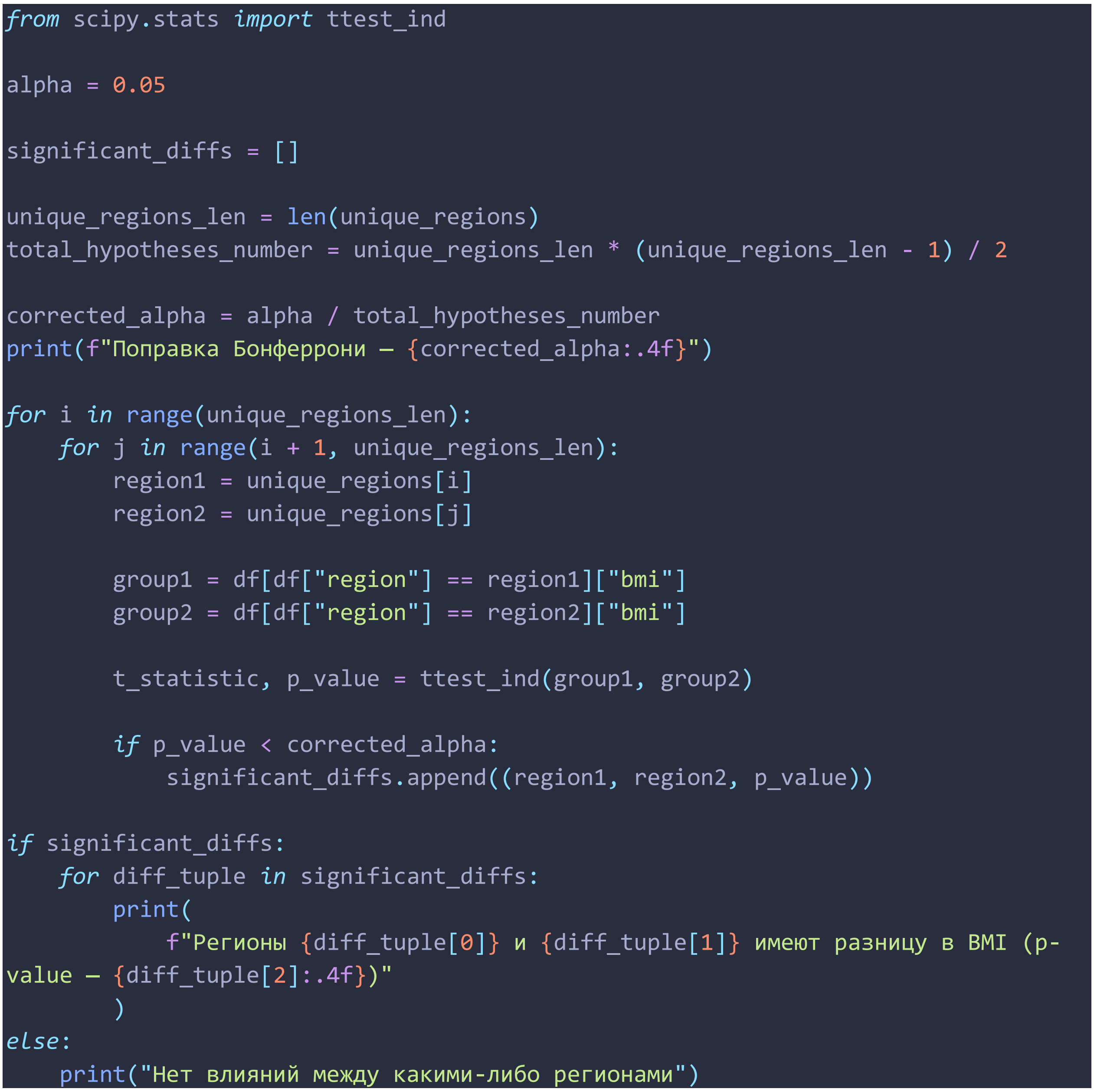


```
import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols("bmi ~ region", data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)

print(anova_table)
print()

p_value = anova_table["PR(>F)"]["region"]
print(f"p-value — {p_value}")


alpha = 0.05
if p_value > alpha:
    print("Принимаем нулевую гипотезу: регион НЕ влияет на BMI.")
else:
    print("Отклоняем нулевую гипотезу: регион влияет на BMI.")
```

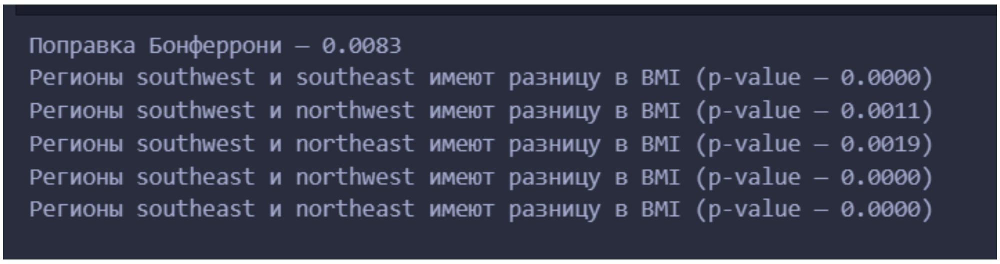


```
                sum_sq      df          F        PR(>F)
region     4055.880631     3.0  39.495057  1.881839e-24
Residual  45664.319755  1334.0        NaN           NaN

p-value — 1.881838913928849e-24
Отклоняем нулевую гипотезу: регион влияет на BMI.
```

С помощью t-критерия Стьюдента переберём все пары, определим поправку Бонферрони:

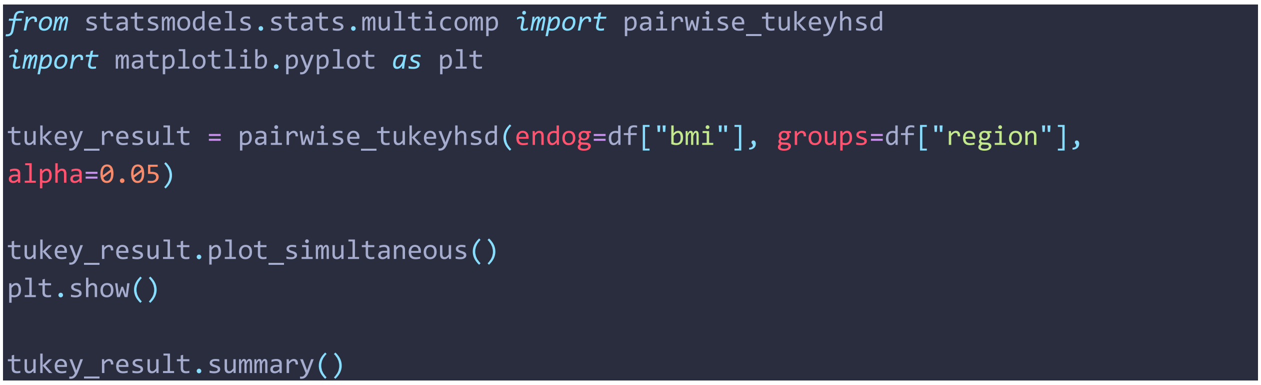


```
from scipy.stats import ttest_ind

alpha = 0.05

significant_diffs = []

unique_regions_len = len(unique_regions)
total_hypotheses_number = unique_regions_len * (unique_regions_len - 1) / 2

corrected_alpha = alpha / total_hypotheses_number
print(f"Поправка Бонферрони — {corrected_alpha:.4f}")

for i in range(unique_regions_len):
    for j in range(i + 1, unique_regions_len):
        region1 = unique_regions[i]
        region2 = unique_regions[j]

        group1 = df[df["region"] == region1]["bmi"]
        group2 = df[df["region"] == region2]["bmi"]

        t_statistic, p_value = ttest_ind(group1, group2)

        if p_value < corrected_alpha:
            significant_diffs.append((region1, region2, p_value))

if significant_diffs:
    for diff_tuple in significant_diffs:
        print(
            f"Регионы {diff_tuple[0]} и {diff_tuple[1]} имеют разницу в BMI (p-
value — {diff_tuple[2]:.4f})"
        )
else:
    print("Нет влияний между какими-либо регионами")
```

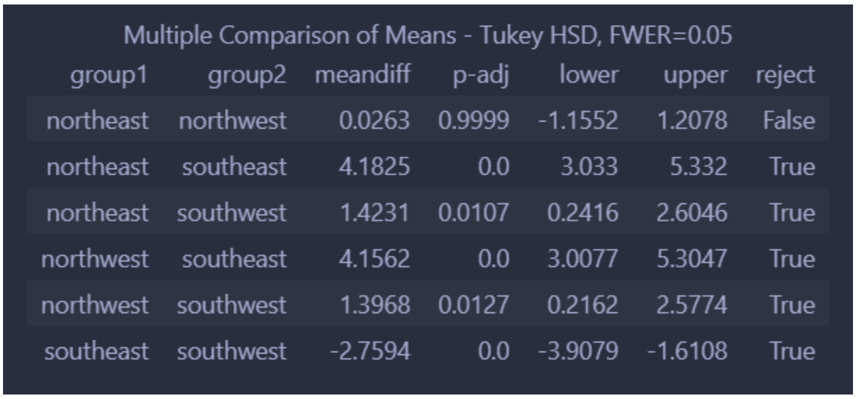


```
Поправка Бонферрони — 0.0083
Регионы southwest и southeast имеют разницу в BMI (p-value — 0.0000)
Регионы southwest и northwest имеют разницу в BMI (p-value — 0.0011)
Регионы southwest и northeast имеют разницу в BMI (p-value — 0.0019)
Регионы southeast и northwest имеют разницу в BMI (p-value — 0.0000)
Регионы southeast и northeast имеют разницу в BMI (p-value — 0.0000)
```

Выполним пост-хок тесты Тьюки и построим график:

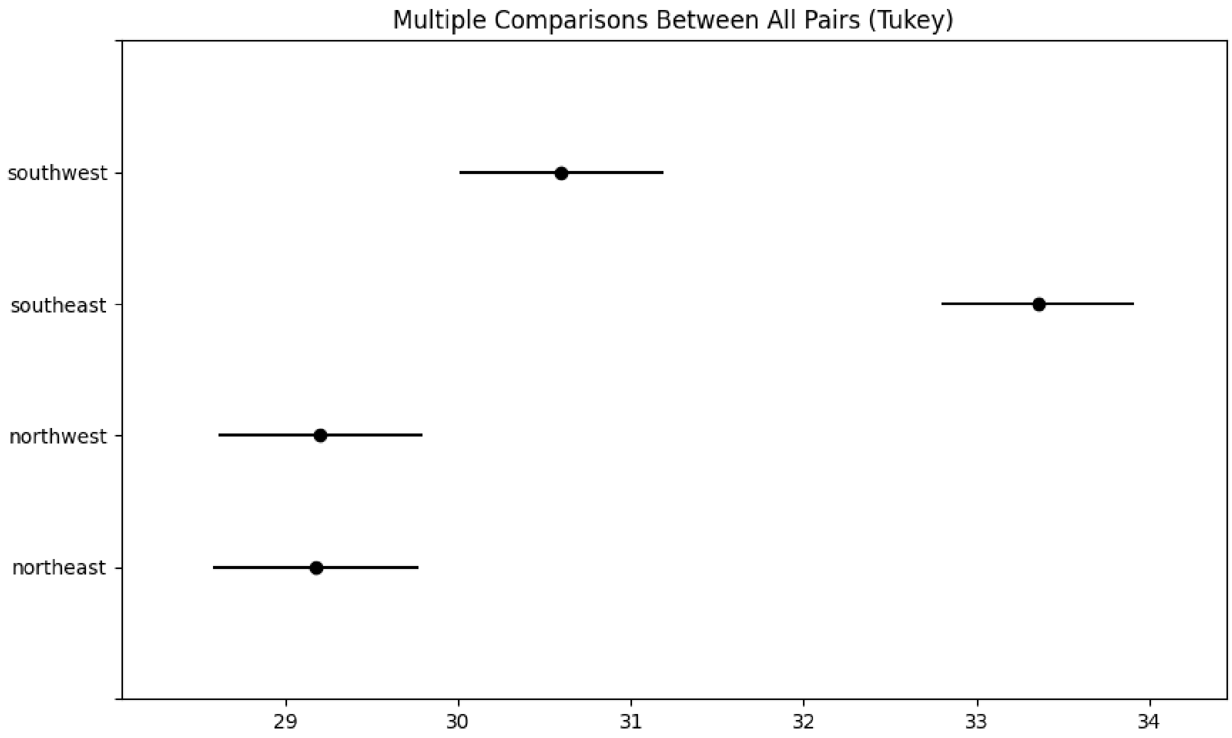


```python
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import matplotlib.pyplot as plt

tukey_result = pairwise_tukeyhsd(endog=df["bmi"], groups=df["region"],
alpha=0.05)

tukey_result.plot_simultaneous()
plt.show()

tukey_result.summary()
```

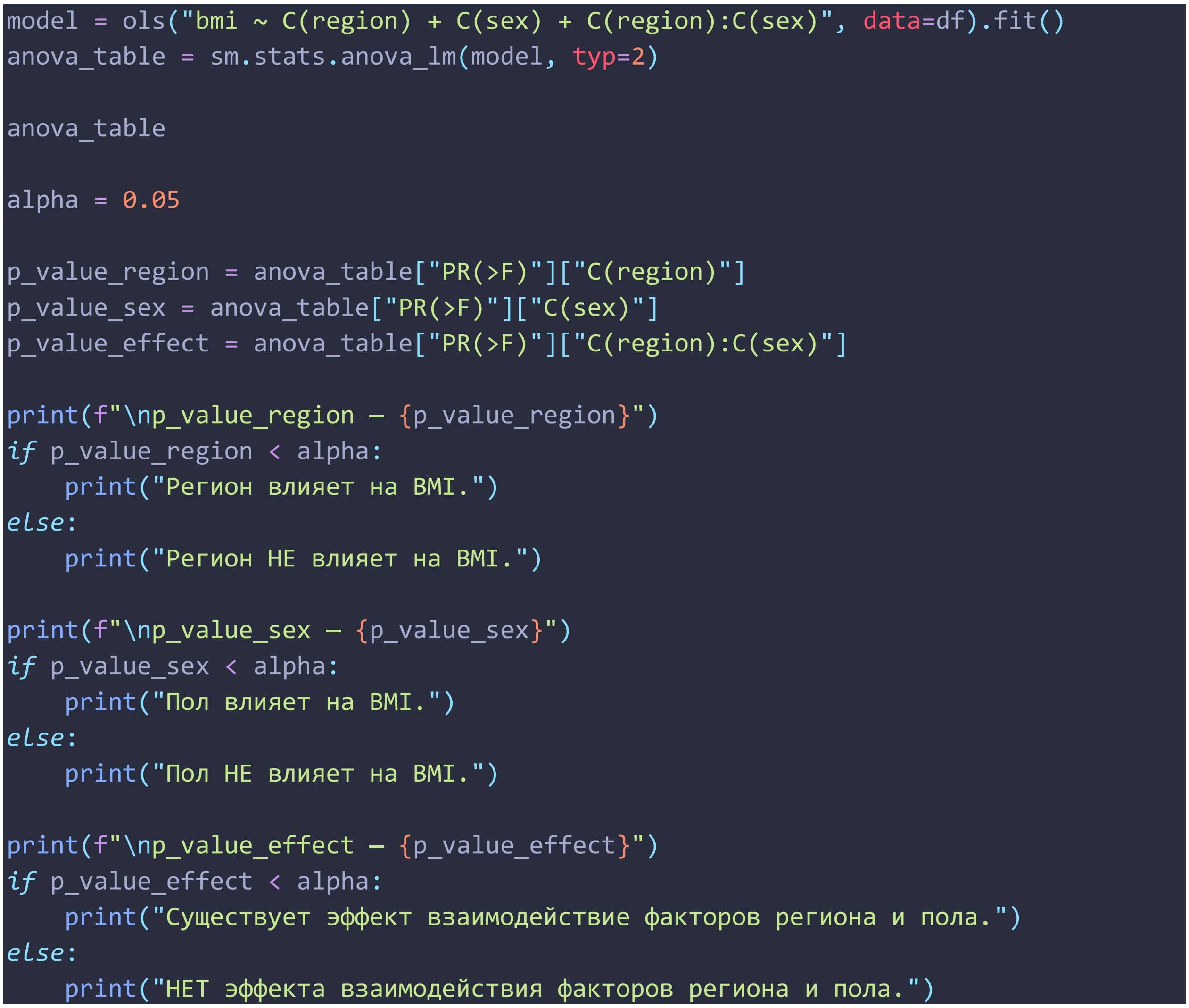


Multiple Comparison of Means - Tukey HSD, FWER=0.05

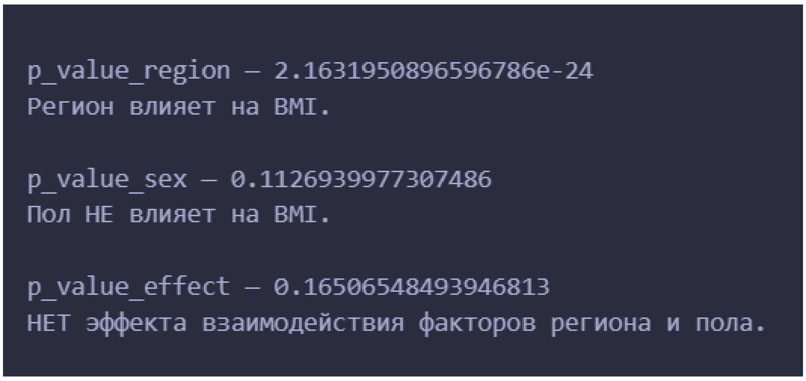
group1	group2	meandiff	p-adj	lower	upper	reject
northeast	northwest	0.0263	0.9999	-1.1552	1.2078	False
northeast	southeast	4.1825	0.0	3.033	5.332	True
northeast	southwest	1.4231	0.0107	0.2416	2.6046	True
northwest	southeast	4.1562	0.0	3.0077	5.3047	True
northwest	southwest	1.3968	0.0127	0.2162	2.5774	True
southeast	southwest	-2.7594	0.0	-3.9079	-1.6108	True



Выполним двухфакторный ANOVA-тест, чтобы проверить влияние региона и пола на индекс массы тела:

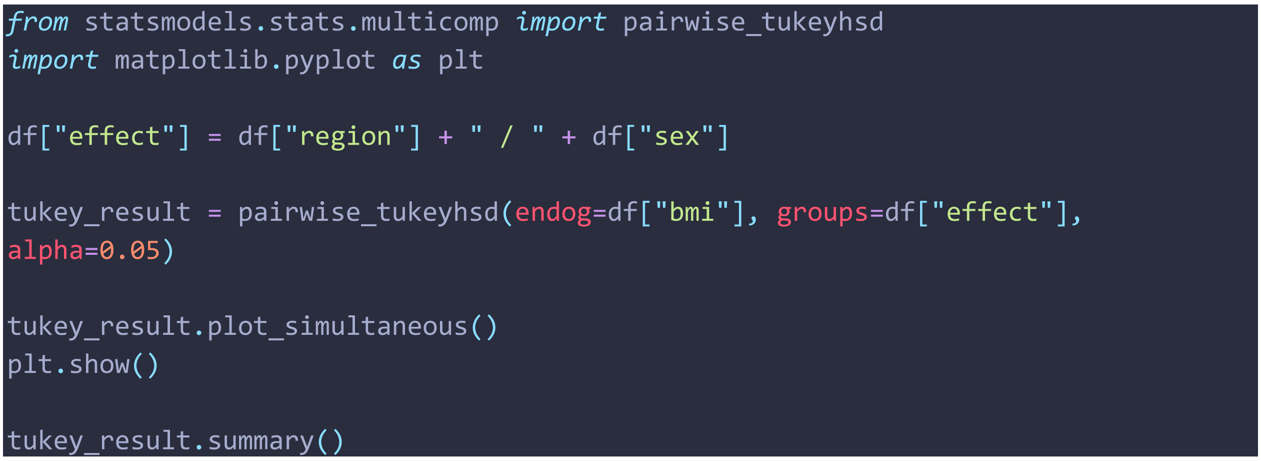


```
model = ols("bmi ~ C(region) + C(sex) + C(region):C(sex)", data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)

anova_table

alpha = 0.05

p_value_region = anova_table["PR(>F)"]["C(region)"]
p_value_sex = anova_table["PR(>F)"]["C(sex)"]
p_value_effect = anova_table["PR(>F)"]["C(region):C(sex)"]

print(f"\np_value_region — {p_value_region}")
if p_value_region < alpha:
    print("Регион влияет на BMI.")
else:
    print("Регион НЕ влияет на BMI.")

print(f"\np_value_sex — {p_value_sex}")
if p_value_sex < alpha:
    print("Пол влияет на BMI.")
else:
    print("Пол НЕ влияет на BMI.")

print(f"\np_value_effect — {p_value_effect}")
if p_value_effect < alpha:
    print("Существует эффект взаимодействие факторов региона и пола.")
else:
    print("НЕТ эффекта взаимодействия факторов региона и пола.")
```

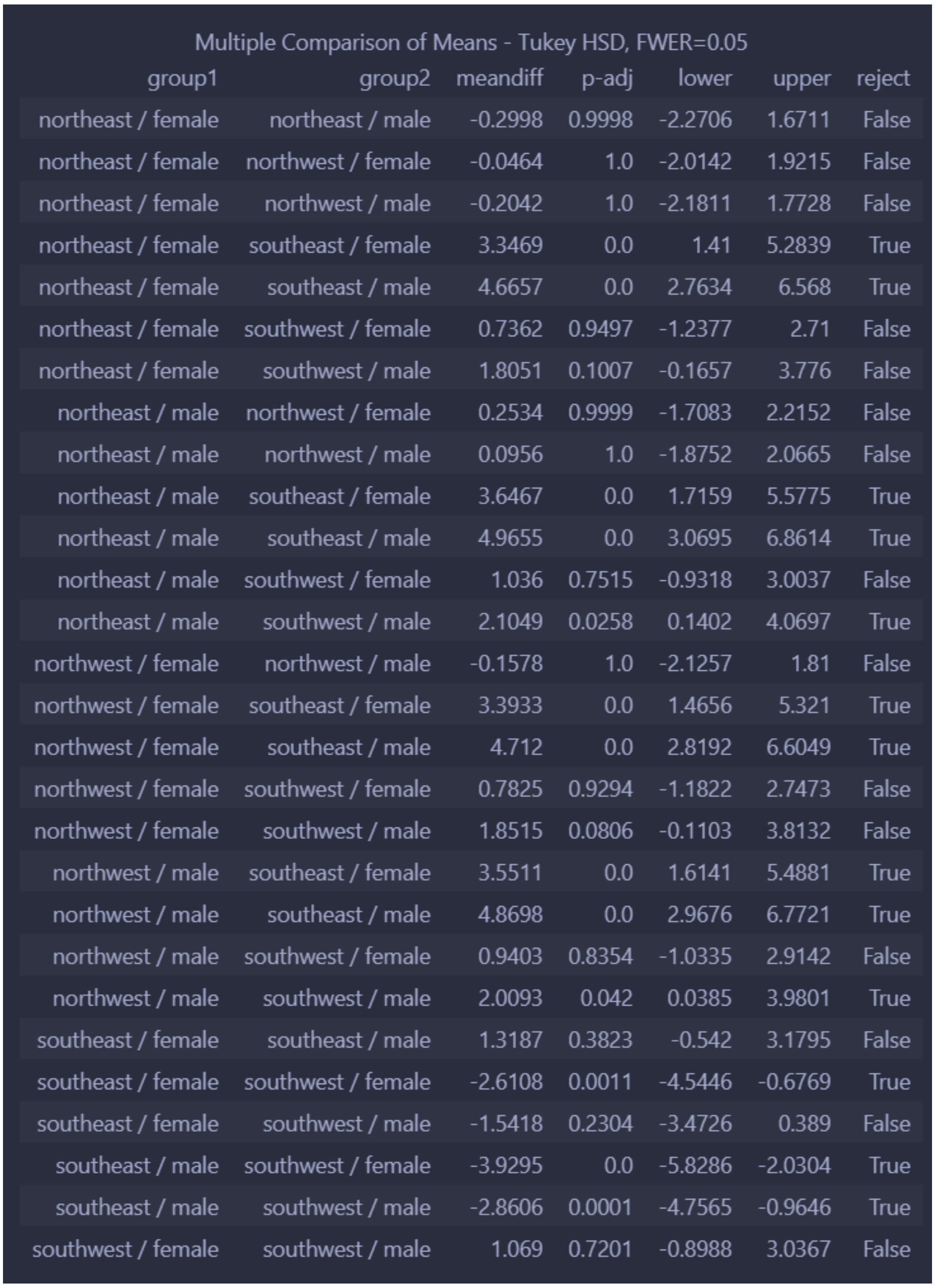


```

p_value_region — 2.1631950896596786e-24
Регион влияет на BMI.

p_value_sex — 0.1126939977307486
Пол НЕ влияет на BMI.

p_value_effect — 0.16506548493946813
НЕТ эффекта взаимодействия факторов региона и пола.
```

Выполним пост-хок тесты Тьюки и построить график:

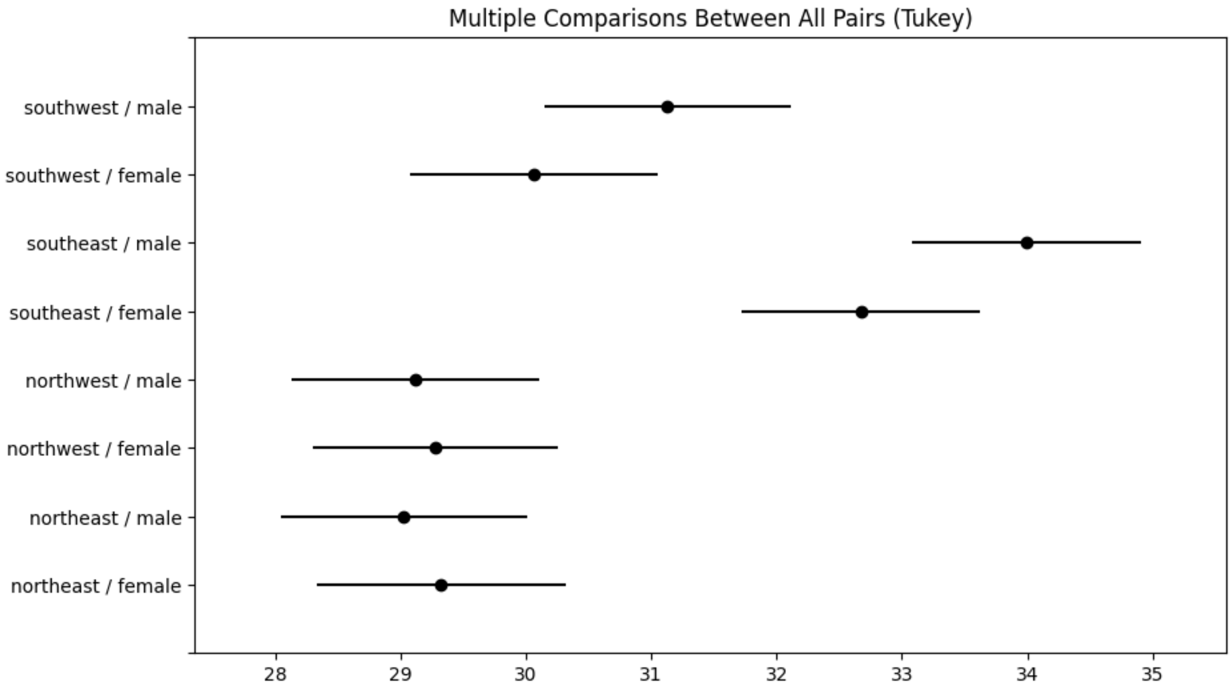


```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import matplotlib.pyplot as plt

df["effect"] = df["region"] + " / " + df["sex"]

tukey_result = pairwise_tukeyhsd(endog=df["bmi"], groups=df["effect"],
alpha=0.05)

tukey_result.plot_simultaneous()
plt.show()

tukey_result.summary()
```



Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
northeast / female	northeast / male	-0.2998	0.9998	-2.2706	1.6711	False
northeast / female	northwest / female	-0.0464	1.0	-2.0142	1.9215	False
northeast / female	northwest / male	-0.2042	1.0	-2.1811	1.7728	False
northeast / female	southeast / female	3.3469	0.0	1.41	5.2839	True
northeast / female	southeast / male	4.6657	0.0	2.7634	6.568	True
northeast / female	southwest / female	0.7362	0.9497	-1.2377	2.71	False
northeast / female	southwest / male	1.8051	0.1007	-0.1657	3.776	False
northeast / male	northwest / female	0.2534	0.9999	-1.7083	2.2152	False
northeast / male	northwest / male	0.0956	1.0	-1.8752	2.0665	False
northeast / male	southeast / female	3.6467	0.0	1.7159	5.5775	True
northeast / male	southeast / male	4.9655	0.0	3.0695	6.8614	True
northeast / male	southwest / female	1.036	0.7515	-0.9318	3.0037	False
northeast / male	southwest / male	2.1049	0.0258	0.1402	4.0697	True
northwest / female	northwest / male	-0.1578	1.0	-2.1257	1.81	False
northwest / female	southeast / female	3.3933	0.0	1.4656	5.321	True
northwest / female	southeast / male	4.712	0.0	2.8192	6.6049	True
northwest / female	southwest / female	0.7825	0.9294	-1.1822	2.7473	False
northwest / female	southwest / male	1.8515	0.0806	-0.1103	3.8132	False
northwest / male	southeast / female	3.5511	0.0	1.6141	5.4881	True
northwest / male	southeast / male	4.8698	0.0	2.9676	6.7721	True
northwest / male	southwest / female	0.9403	0.8354	-1.0335	2.9142	False
northwest / male	southwest / male	2.0093	0.042	0.0385	3.9801	True
southeast / female	southeast / male	1.3187	0.3823	-0.542	3.1795	False
southeast / female	southwest / female	-2.6108	0.0011	-4.5446	-0.6769	True
southeast / female	southwest / male	-1.5418	0.2304	-3.4726	0.389	False
southeast / male	southwest / female	-3.9295	0.0	-5.8286	-2.0304	True
southeast / male	southwest / male	-2.8606	0.0001	-4.7565	-0.9646	True
southwest / female	southwest / male	1.069	0.7201	-0.8988	3.0367	False


Multiple Comparisons Between All Pairs (Tukey)
southwest / male
southwest / female
southeast / male
southeast / female
northwest / male
northwest / female
northeast / male
northeast / female
28
29
30
31
32
33
34
35

ВЫВОД

В ходе выполнения данной практической работы были выполнены задачи по выявлению корреляций, проведены тесты ANOVA, пост-хок тесты Тьюки, построены диаграммы рассеивания и визуализирована регрессия на графике.