МИНОБРНАУКИ РОССИИ

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

РТУ МИРЭА

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

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по дисциплине «Технологии и инструментарий анализа больших данных»

Практическая работа № 4

Студент группы *ИКБО-01-22 Прокопчук Роман Олегович* _____
(подпись)

Ассистент *Тетерин Николай Николаевич* _____
(подпись)

Отчёт представлен «__» ноября 2025 г.

Москва, 2025 г.

ЦЕЛЬ

Ознакомиться с инструментами на Python для построения корреляции, линейной регрессии и выполнения дисперсионного анализа.

ХОД РАБОТЫ

Задание

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

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

1.1. Найти и интерпретировать корреляцию между переменными «Улица» и «Гараж» (подсчитать корреляцию по Пирсону).

1.2. Построить диаграмму рассеяния для вышеупомянутых переменных.

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

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

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

2.3. Визуализировать регрессию на графике.

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

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

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.

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

3.4. Выполнить пост-хок тесты Тьюки и построить 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.

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

Ход работы

Код для выполнения задания 1 представлен на рисунке 1. Результат его работы – на рисунке 2, построенная диаграмма – на рисунке 3.

```
import matplotlib.pyplot as plt
import pandas as pd

data = {
    'День': ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница'],
    'Улица': [80, 98, 75, 91, 78],
    'Гараж': [100, 82, 105, 89, 102]
}

df = pd.DataFrame(data)

print("--- Исходные данные: ---")
print(df)
print("\n")

correlation = df['Улица'].corr(df['Гараж'])
abs_corr = abs(correlation)

print(f"Значение r = {correlation:.4f} (по модулю {abs_corr:.4f}).")

plt.figure(figsize=(8, 6))
plt.scatter(df['Улица'], df['Гараж'], color='crimson', marker='o')

plt.title('Диаграмма рассеяния: Уличная стоянка vs. Гараж')
plt.xlabel('Число автомобилей на уличной стоянке')
plt.ylabel('Число автомобилей в подземном гараже')

plt.grid(True)

plt.show()
```

Рисунок 1 – Код для выполнения задания 1

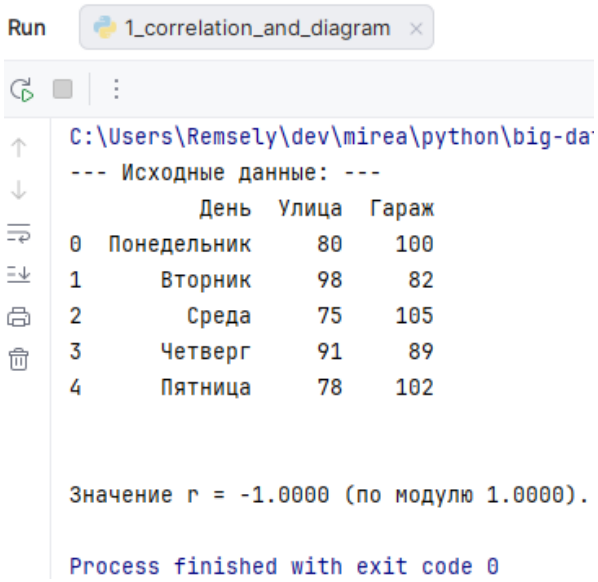

Рисунок 2 – Результат выполнения кода для задания 1

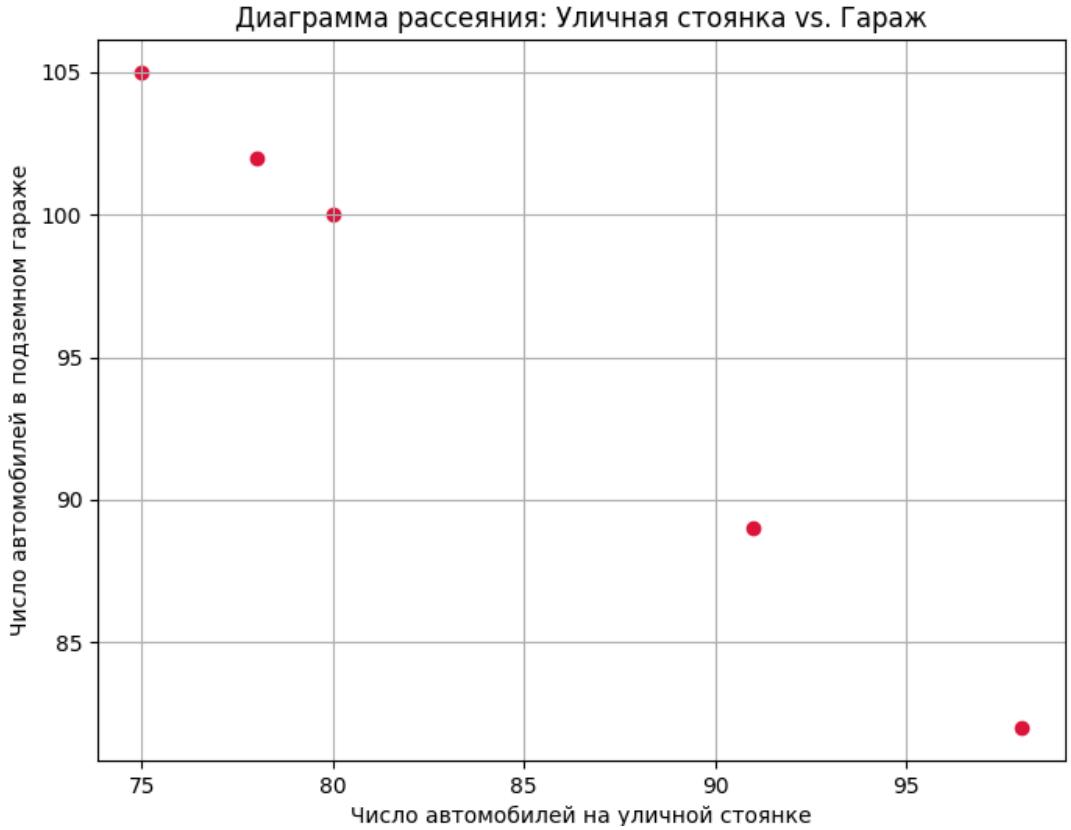

Рисунок 3 – Построенная для задания 1 диаграмма

Вывод: наблюдается идеальная отрицательная линейная зависимость – при росте числа машин на улице число машин в гараже падает на такую же величину.

Код для выполнения задания 2 представлен на рисунках 4-6. Результат его работы – на рисунке 7, построенная диаграмма – на рисунке 8.

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

df = pd.read_csv('data/insurance.csv')

print("--- Данные 'insurance.csv' ---")
print(df.head())
print("\n")

print(f"--- Пропущенные значения: ---\n{df.isnull().sum()}\n")

df['sex'] = df['sex'].map({'female': 0, 'male': 1})
df['smoker'] = df['smoker'].map({'no': 0, 'yes': 1})

df = pd.get_dummies(df, columns=['region'], drop_first=True)

print("--- Данные после предобработки (текст в числа): ---")
print(df.head())
print("\n")

# --- Задание 2.1 ---
corr_matrix = df.corr()

target_corr = corr_matrix['charges'].sort_values(ascending=False)

print(target_corr)
print("\n")

most_correlated_var = target_corr.index[1]
print(f"Наиболее коррелирующая переменная (кроме самой 'charges'): '{most_correlated_var}'")
```

Рисунок 4 – Код для преобразования данных и выполнения пункта 2.1

```python
# --- Задание 2.2: Реализация регрессии вручную (Градиентный спуск) ---
X = np.array(df[most_correlated_var])
y = np.array(df['charges'])


def mserror(X, y, w1, w0):  1 usage  new *
    y_pred = w1 * X + w0
    n = len(y)
    return np.sum((y - y_pred) ** 2) / n


def gr_mserror(X, y, w1, w0):  1 usage  new *
    y_pred = w1 * X + w0
    n = len(y)

    grad_w1 = (-2 / n) * np.sum(X * (y - y_pred))
    grad_w0 = (-2 / n) * np.sum(y - y_pred)

    return grad_w1, grad_w0


w1 = 0.0
w0 = 0.0
learning_rate = 0.1
n_iterations = 10000
eps = 0.0001

for i in range(n_iterations):
    grad_w1, grad_w0 = gr_mserror(X, y, w1, w0)

    next_w1 = w1 - learning_rate * grad_w1
    next_w0 = w0 - learning_rate * grad_w0

    if abs(next_w1 - w1) < eps and abs(next_w0 - w0) < eps:
        print(f"\nАлгоритм сошелся на итерации {i}")
        break

    w1, w0 = next_w1, next_w0

final_mse = mserror(X, y, w1, w0)

print("\n--- Результаты ручной регрессии: ---")
print(f"Наклон (w1): {w1:.2f}")
print(f"Сдвиг (w0): {w0:.2f}")
print(f"MSE (среднеквадратичная ошибка): {final_mse:.2f}")
print("\n")
```

Рисунок 5 – Код для выполнения пункта 2.2

```python
# --- Задание 2.3: Визуализация регрессии ---
plt.figure(figsize=(10, 6))
plt.scatter(X, y, alpha=0.3, label='Исходные данные')

X_line = np.array([0, 1])
y_line = w1 * X_line + w0

plt.plot( *args: X_line, y_line, color='red', linewidth=3,
         label=f'Модель: y = {w1:.2f}*x + {w0:.2f}')

plt.title(f'Линейная регрессия: {most_correlated_var} vs. charges')
plt.xlabel(most_correlated_var.capitalize())
plt.ylabel('Charges (Стоимость страховки)')
plt.legend()
plt.grid(True)
plt.show()
```

Рисунок 6 – Код для выполнения пункта 2.3

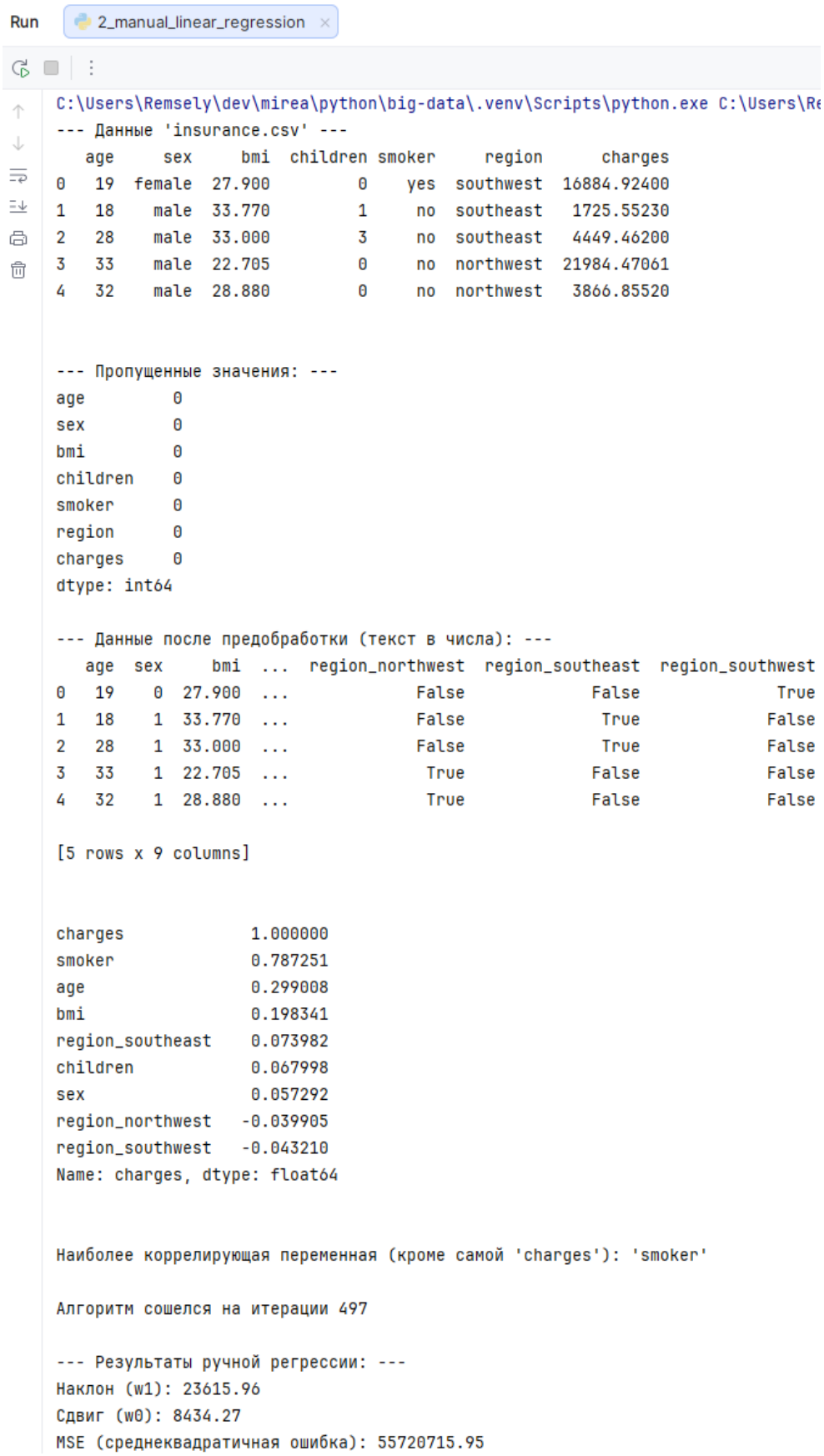

Рисунок 7 – Результат работы кода для задания 2

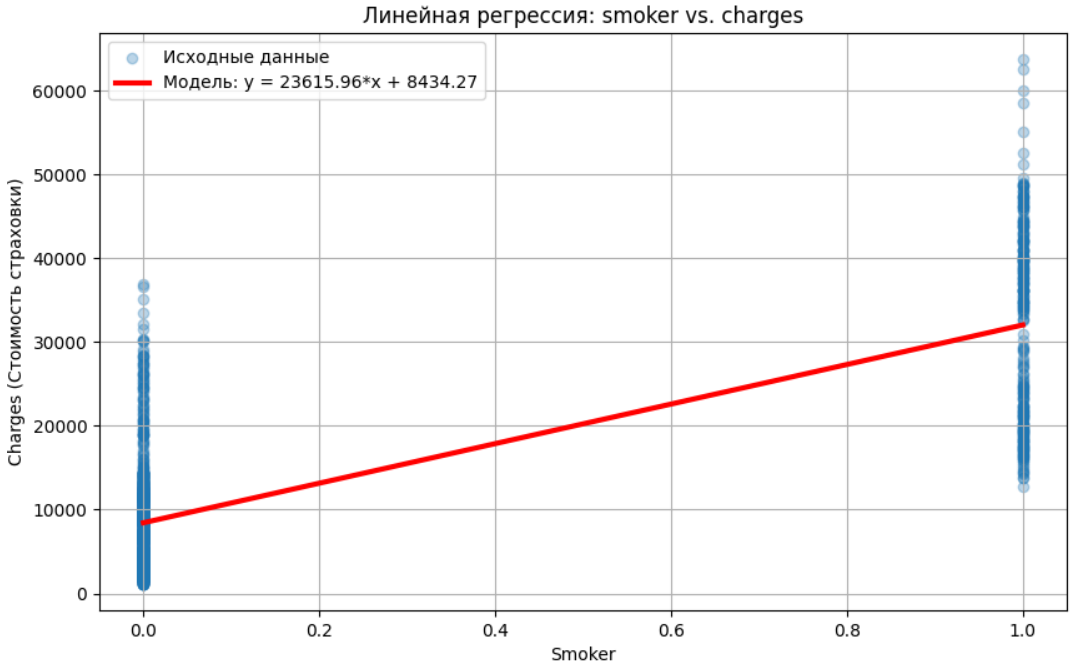

Рисунок 8 – Построенная диаграмма линейной регрессии

Вывод: курение – доминирующий фактор стоимости страховки в наборе (сильная корреляция и значительный прирост прогноза).

Код для выполнения задания 3 представлен на рисунках 9-13. Результат его работы – на рисунках 14-16, построенная диаграммы – на рисунках 17-18.

```python
import warnings

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

warnings.filterwarnings('ignore')

df = pd.read_csv('data/insurance.csv')

print("--- Данные 'insurance.csv' ---")
print(df.head())
print("\n")

unique_regions = df['region'].unique()
print(f"--- Уникальные регионы в датасете: {unique_regions} ---")
print("\n")

# --- Задание 3.1 ---
regions = df['region'].unique()
bmi_by_region = []
for region in regions:
    bmi_by_region.append(df[df['region'] == region]['bmi'])

f_stat, p_value = stats.f_oneway(*bmi_by_region)

print(f"F-статистика: {f_stat:.4f}")
print(f"P-значение: {p_value:.4f}")

alpha = 0.05
if p_value < alpha:
    print(f"Вывод: P-значение ({p_value:.4f}) < {alpha}.")
    print("-> Существуют статистически значимые различия в BMI между регионами.")
else:
    print(f"Вывод: P-значение ({p_value:.4f}) > {alpha}, не можем отклонить нулевую гипотезу.")
    print("-> Статистически значимых различий в BMI между регионами не найдено.")
print("\n")
```

Рисунок 9 – Код для выполнения пункта 3.1

```python
# --- Задание 3.2 ---
model = ols( formula: 'bmi ~ C(region)', data=df).fit()
anova_table = sm.stats.anova_lm( *args: model, type=2)

print(anova_table)

# --- Задание 3.3 ---
pairs = []
for i in range(len(regions)):
    for j in range(i + 1, len(regions)):
        pairs.append((regions[i], regions[j]))

print(f"Всего {len(pairs)} пар для сравнения.")

alpha = 0.05
bonferroni_alpha = alpha / len(pairs)
print(f"Уровень значимости (alpha): {alpha}")
print(f"Новый уровень значимости (поправка Бонферрони): {alpha} / {len(pairs)} = {bonferroni_alpha:.4f}\n")

print("Результаты t-тестов:")
for r1, r2 in pairs:
    group1 = df[df['region'] == r1]['bmi']
    group2 = df[df['region'] == r2]['bmi']

    t_stat, p_value = stats.ttest_ind(group1, group2)

    print(f"  Пара: {r1} vs {r2}")
    print(f"    P-значение: {p_value:.4f}")

    if p_value < bonferroni_alpha:
        print(f"    Вывод: ЗНАЧИМО ({p_value:.4f} < {bonferroni_alpha:.4f})")
    else:
        print(f"    Вывод: НЕ ЗНАЧИМО ({p_value:.4f} > {bonferroni_alpha:.4f})")
print("\n")
```

Рисунок 10 – Код для выполнения пунктов 3.2 и 3.3

```python
# --- Задание 3.4 ---
print("Задание 3.4\n")

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

print(tukey_result)

print("Построение графика для теста Тьюки...")
tukey_result.plot_simultaneous()
plt.title("Тест Тьюки (Tukey HSD) для BMI по Регионам")
plt.show()
```

Рисунок 11 – Код для выполнения пункта 3.4

```python
# --- Задание 3.5 ---
model_2way = ols( formula: 'bmi ~ C(region) + C(sex) + C(region):C(sex)', data=df).fit()
anova_table_2way = sm.stats.anova_lm( *args: model_2way, type=2)

print(anova_table_2way)

print("\n--- Выводы по двухфакторному ANOVA: ---")
p_region = anova_table_2way.loc['C(region)', 'PR(>F)']
p_sex = anova_table_2way.loc['C(sex)', 'PR(>F)']
p_interaction = anova_table_2way.loc['C(region):C(sex)', 'PR(>F)']

if p_region < 0.05:
    print(f"1. 'region': {p_region:.4f} (< 0.05) - ЗНАЧИМО. Регион влияет на BMI.")
else:
    print(f"1. 'region': {p_region:.4f} (> 0.05) - НЕ ЗНАЧИМО.")

if p_sex < 0.05:
    print(f"2. 'sex': {p_sex:.4f} (< 0.05) - ЗНАЧИМО. Пол влияет на BMI.")
else:
    print(f"2. 'sex': {p_sex:.4f} (> 0.05) - НЕ ЗНАЧИМО.")

if p_interaction < 0.05:
    print(f"3. 'interaction': {p_interaction:.4f} (< 0.05) - ЗНАЧИМО. Эффект региона зависит от пола.")
else:
    print(f"3. 'interaction': {p_interaction:.4f} (> 0.05) - НЕ ЗНАЧИМО.")
print("\n")
```

Рисунок 12 – Код для выполнения пункта 3.5

```python
# --- Задание 3.6 ---
df['region_sex_group'] = df['region'] + '_' + df['sex']

print("--- Новые группы для теста (первые 5): ---")
print(df['region_sex_group'].head())
print("\n")

tukey_2way_result = (
    pairwise_tukeyhsd(
        endog=df['bmi'],
        groups=df['region_sex_group'],
        alpha=0.05
    )
)

print(tukey_2way_result)

print("--- Построение графика для теста Тьюки... ---")
tukey_2way_result.plot_simultaneous()
plt.title("Тест Тьюки (Tukey HSD) для BMI по (Регион + Пол)")
plt.show()
```

Рисунок 13 – Код для выполнения пункта 3.6

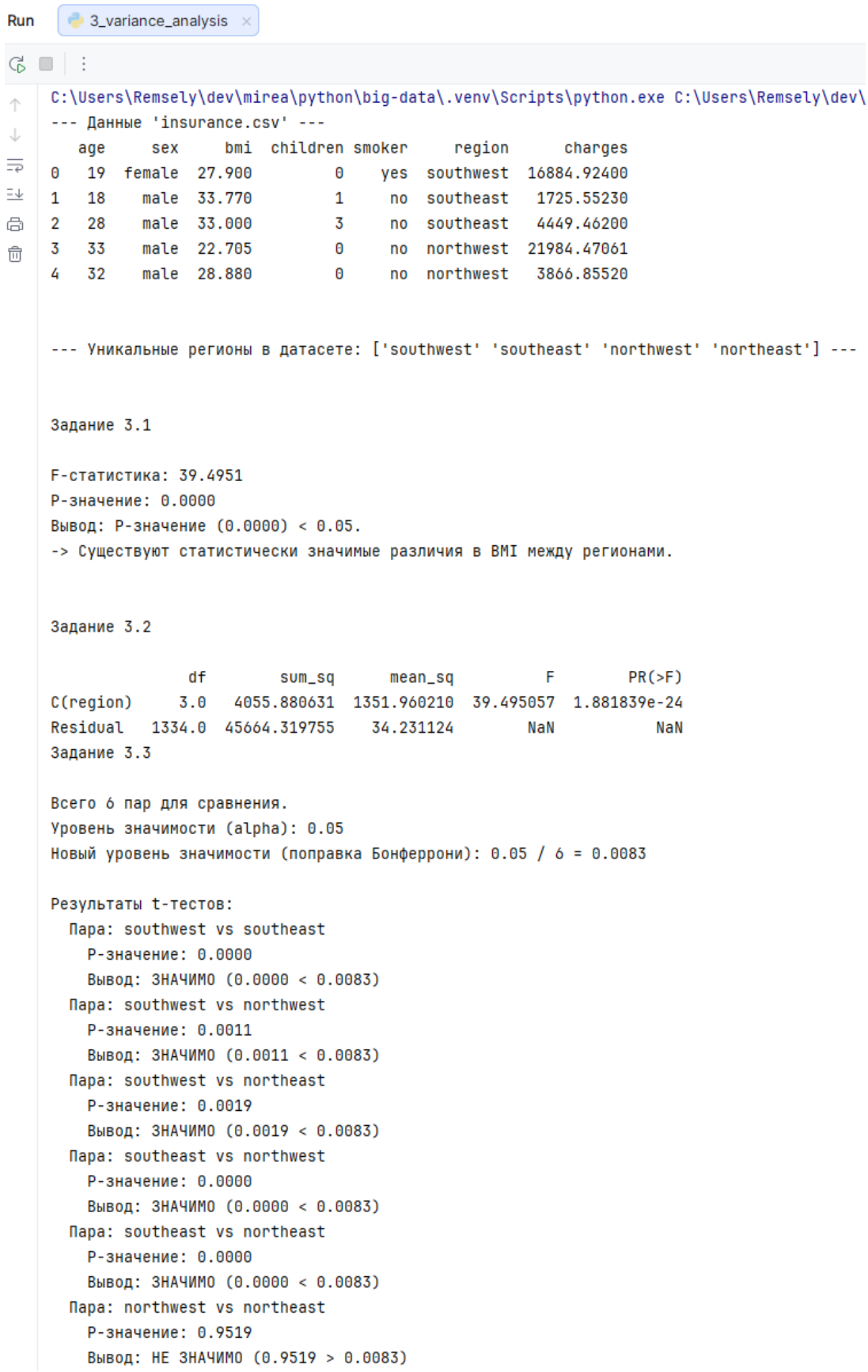

Рисунок 14 – Результат работы программы для пунктов 3.1-3.3

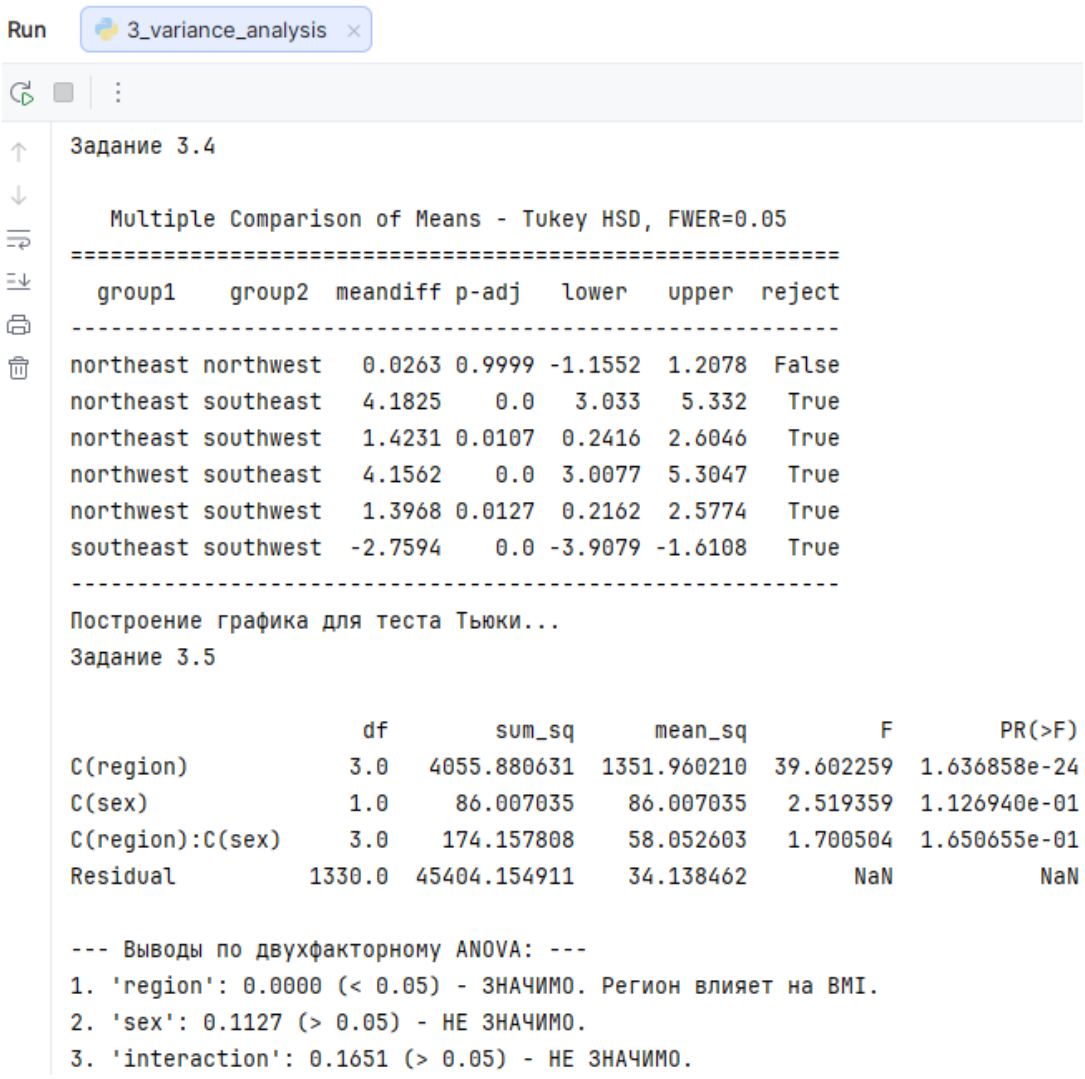

Рисунок 15 – Результат работы программы для пунктов 3.4-3.5

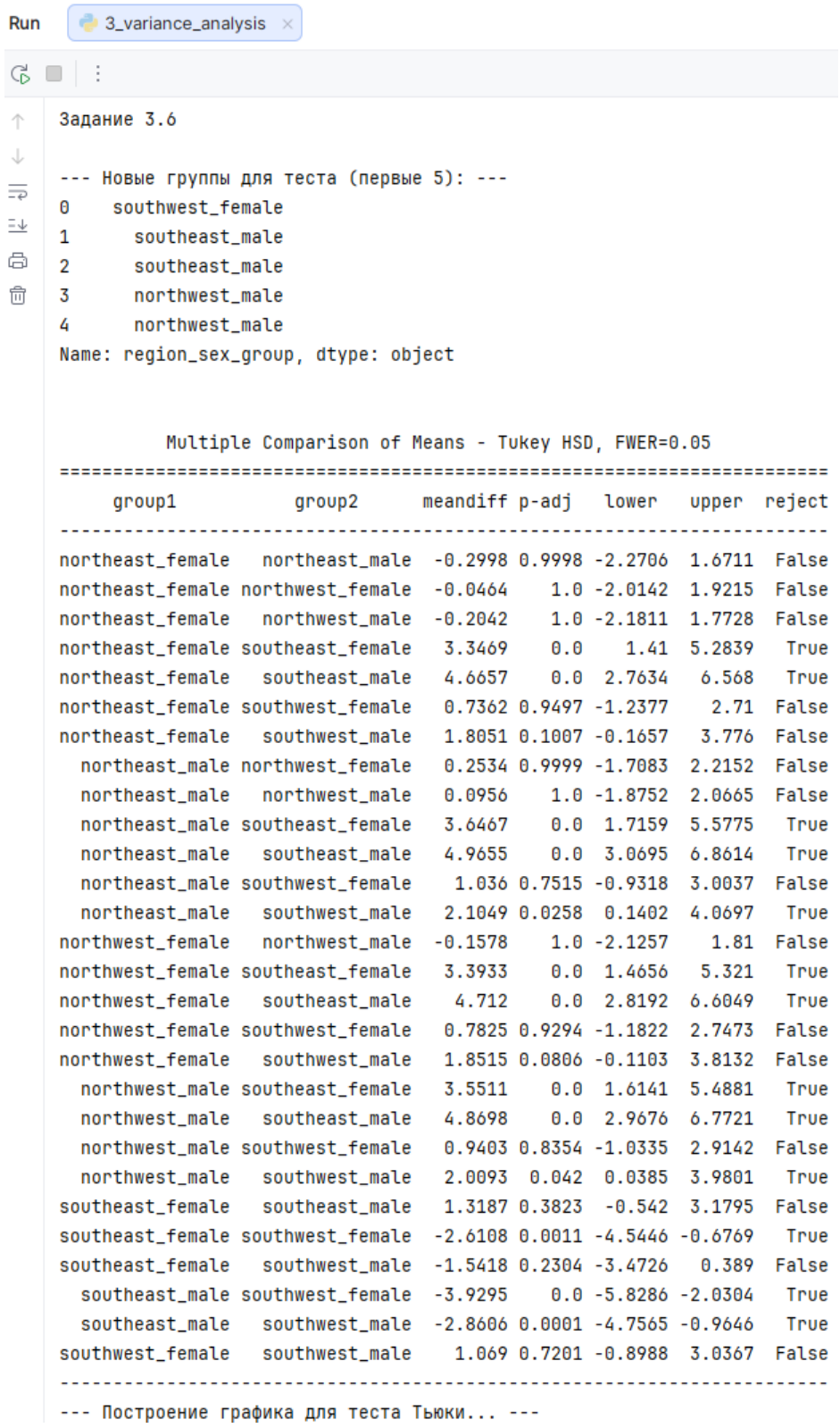

Рисунок 16 – Результат работы программы для пункта 3.6

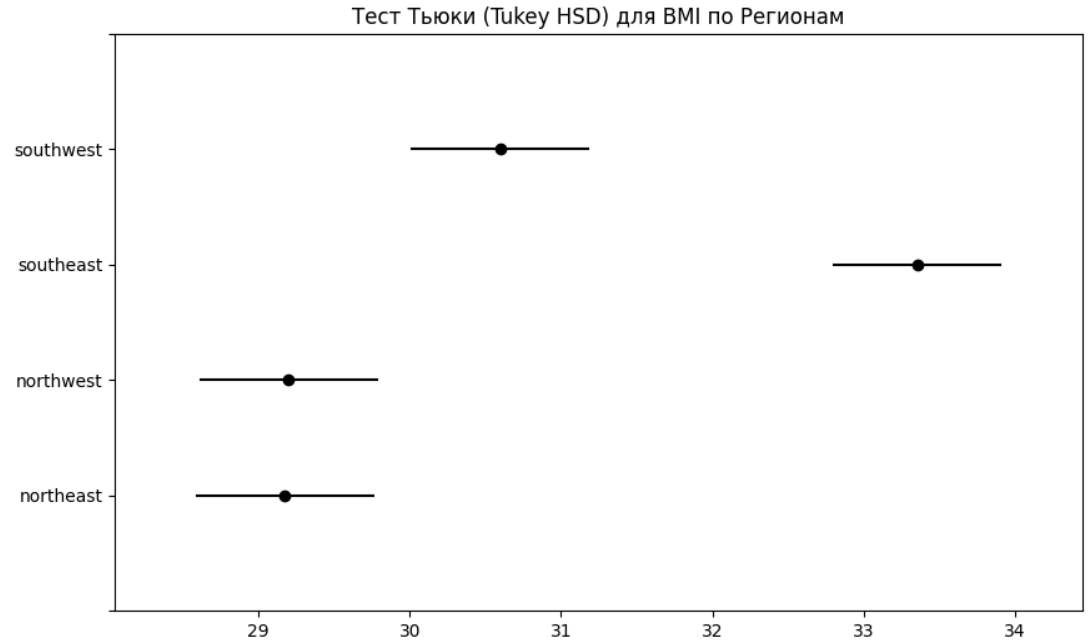

Рисунок 17 – Построенный график для BMI по регионам

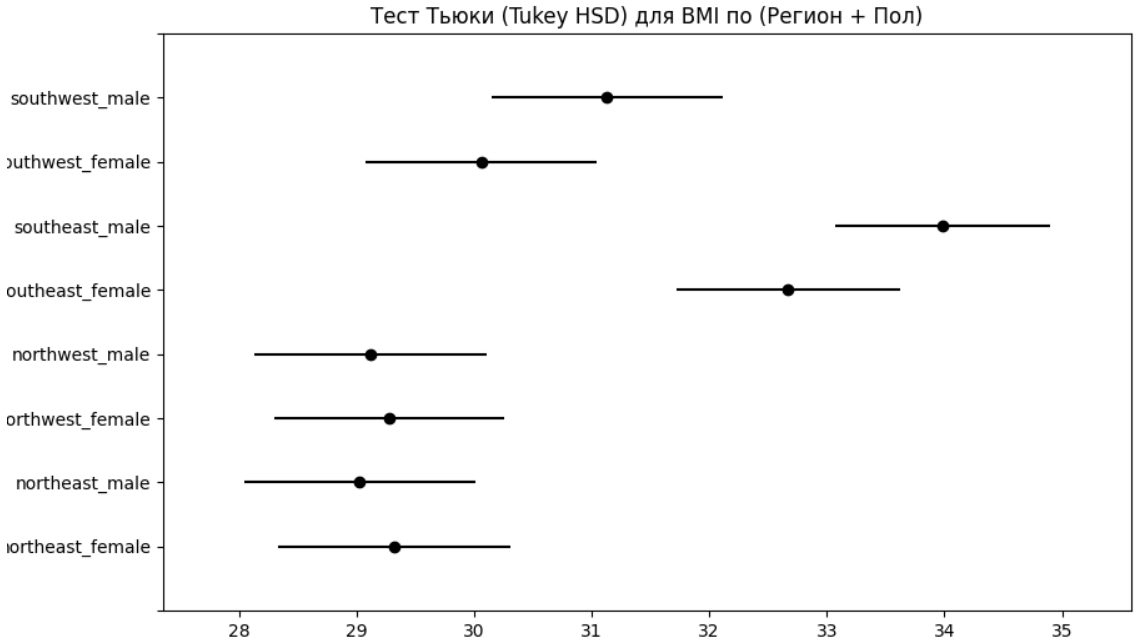

Рисунок 18 – Построенный график для BMI по регионам и полу

Выводы:

- Между регионами существуют статистически значимые различия в BMI.
- Фактор region значим ($p < 0.05$).
- Все пары регионов значимы, кроме northwest vs northeast. Ранг средних BMI: southeast < southwest < (northwest ≈ northeast).
- Тест Тьюки подтверждает – southeast ниже всех; southwest выше southeast, но ниже northwest и northeast; northwest ≈ northeast (различий нет).

- Двухфакторный ANOVA: region значим, sex не значим, взаимодействие region:sex не значимо.
- Пост-хок по группам region+sex: различия формируются регионом; внутри каждого региона различий между полами нет. Сочетания с southeast сильно отличаются от большинства других групп.

ВЫВОД

В ходе выполненной практической работы проведено ознакомление с инструментами на Python для построения корреляции, линейной регрессии и выполнения дисперсионного анализа.