МИНОБРНАУКИ РОССИИ

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

РТУ МИРЭА

Институт Информационных Технологий
Кафедра прикладной математики

ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 4

«Корреляция, линейная регрессия и дисперсионный анализ»

по дисциплине

«Технологии и инструментарий анализа больших данных»

Выполнил студент группы ИВБО-03-21 — Лазарев А. В.

Принял преподаватель кафедры прикладной математики — Тетерин Н.Н.

Практическая работа выполнена « __ » _____ 2024 г.

«Зачтено» « __ » _____ 2024 г.

Москва 2024

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ 2

1 РЕШЕНИЕ ЗАДАЧ 3

1.1 Задача №1 3

1.2 Задача №1.1 3

1.3 Задача №3 3

1.4 Задача №2.1 4

1.5 Задача №2.2 5

1.6 Задачи №2.3 6

1.7 Задача №3.1 6

1.8 Задача №3.2 7

1.9 Задача №3.3 7

1.10 Задача №3.4 8

1.11 Задача №3.5 9

1.12 Задача №3.6 10

1 РЕШЕНИЕ ЗАДАЧ

1.1 Задача №1

Решение программы представлено на Рисунке 1.1.

```
[2]  x = np.array([80, 98, 75, 91, 78])
     y = np.array([100, 82, 105, 89, 102])
```

Рисунок 1.1 – Программа

1.2 Задача №1.1

Решение и результат программы представлены на Рисунке 1.2.

```
[ ]  np.corrcoef(x, y)[0, 1]

     -0.9999999999999998
```

Рисунок 1.2 – Программа и результат ее выполнения

1.3 Задача №3

Решение и результат программы представлены на Рисунке 1.3.

```
plt.grid(True)
plt.title('Рассеяние')
plt.xlabel('Улица')
plt.ylabel('Гараж')
plt.scatter(x, y, marker = 'o', color = 'crimson')
```

```
<matplotlib.collections.PathCollection at 0x7b933f83d210>
```

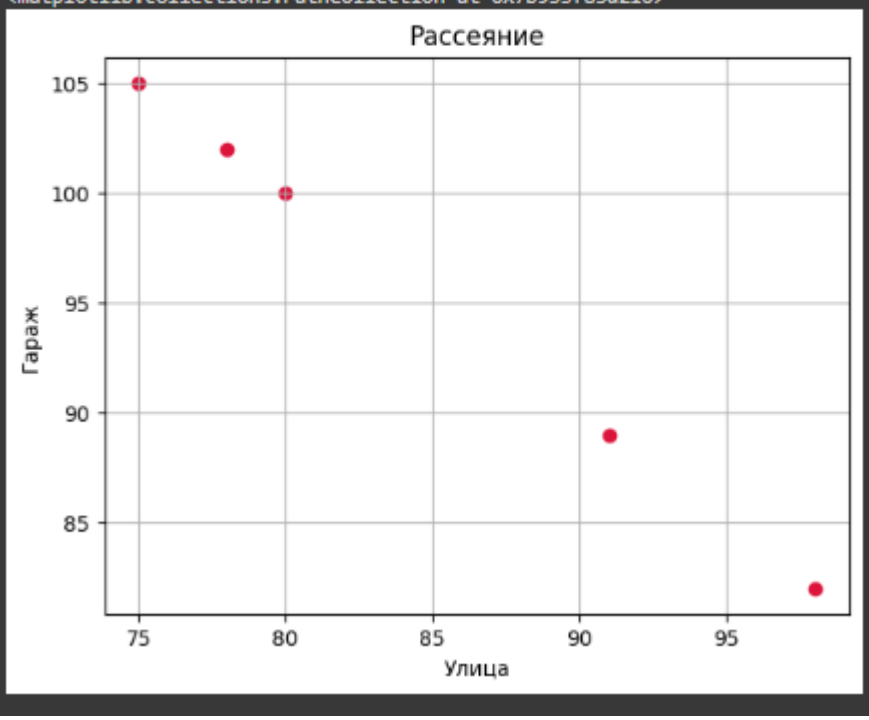

Рисунок 1.3 – Программа и результат ее выполнения

1.4 Задача №2.1

Решение и результат программы представлены на Рисунках 1.4, 1.5.

```
data_numeric = data.select_dtypes(include=['number'])
if 'cases' in data_numeric.columns:
    target_variable = 'cases'
else:
    raise ValueError("Целевая переменная 'cases' не найдена среди числовых данных")
correlation_matrix = data_numeric.corr()
correlations = correlation_matrix[target_variable].drop(target_variable)
most_correlated_variable = correlations.abs().idxmax()
max_correlation = correlations[most_correlated_variable]

print("Корреляционная матрица:")
print(correlation_matrix)
print(f"Наиболее коррелирующая переменная с '{target_variable}': {most_correlated_variable}, коэффициент корреляции: {max_correlation}")
```

Рисунок 1.4 – Программное решение

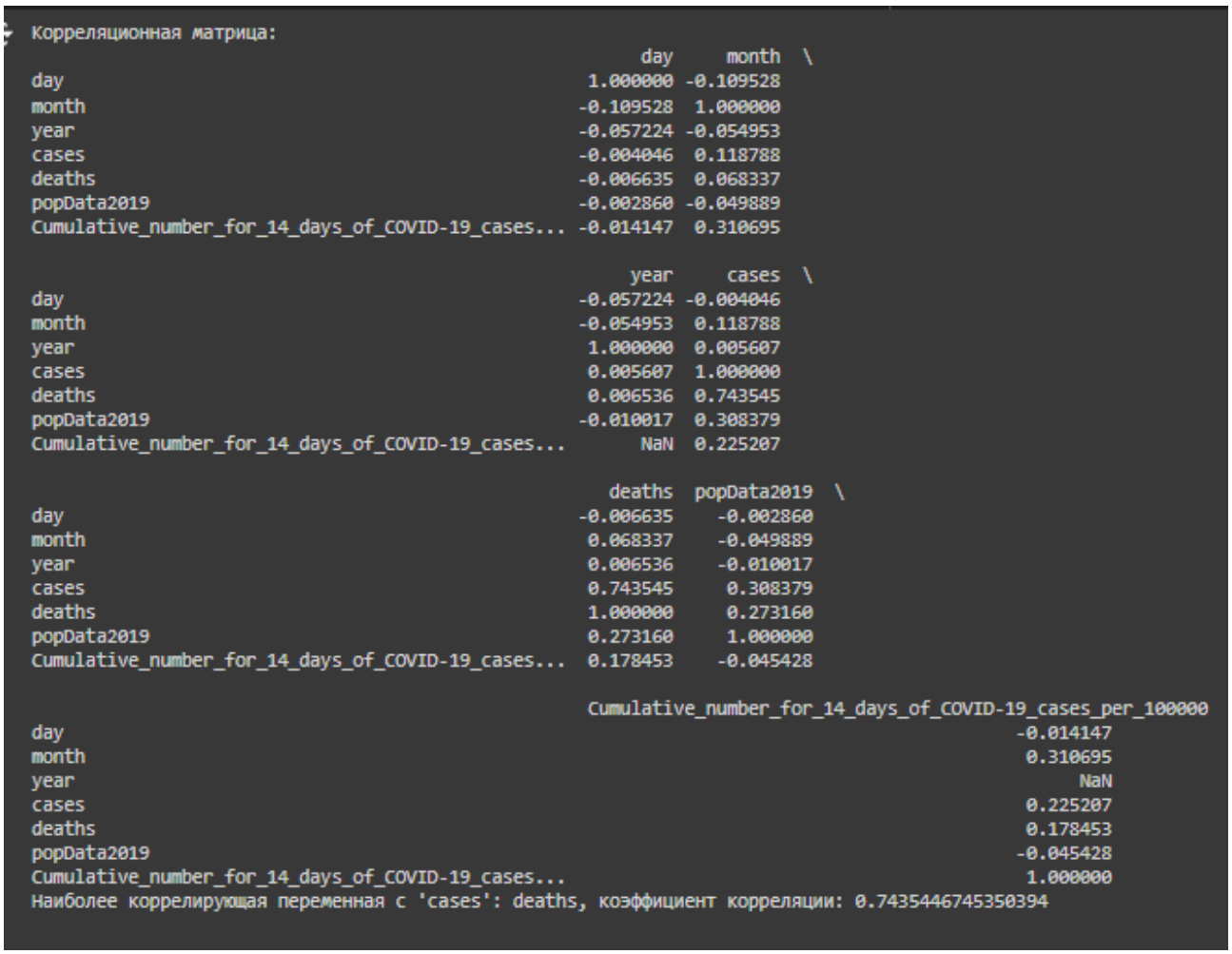

Рисунок 1.5 – Результат выполнения программы

1.5 Задача №2.2

Решение и результат программы представлены на Рисунке 1.6.

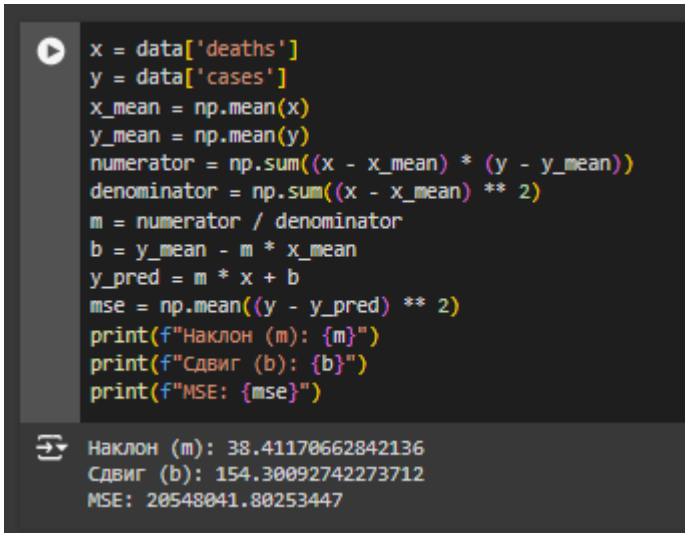

Рисунок 1.6 – Программа и результат ее выполнения

1.6 Задачи №2.3

Решение представлено на Рисунке 1.7, 1.8.

```
x = data['deaths']
y = data['cases']

x_mean = np.mean(x)
y_mean = np.mean(y)
m = np.sum((x - x_mean) * (y - y_mean)) / np.sum((x - x_mean) ** 2)
b = y_mean - m * x_mean

y_pred = m * x + b

plt.figure(figsize=(10, 6))
plt.scatter(x, y, color="blue", label="Данные")
plt.plot(x, y_pred, color="red", label="Линия регрессии")
plt.xlabel("Deaths")
plt.ylabel("Cases")
plt.title("Линейная регрессия между Cases и Deaths")
plt.legend()
plt.show()
```

Рисунок 1.7 – Программное решение

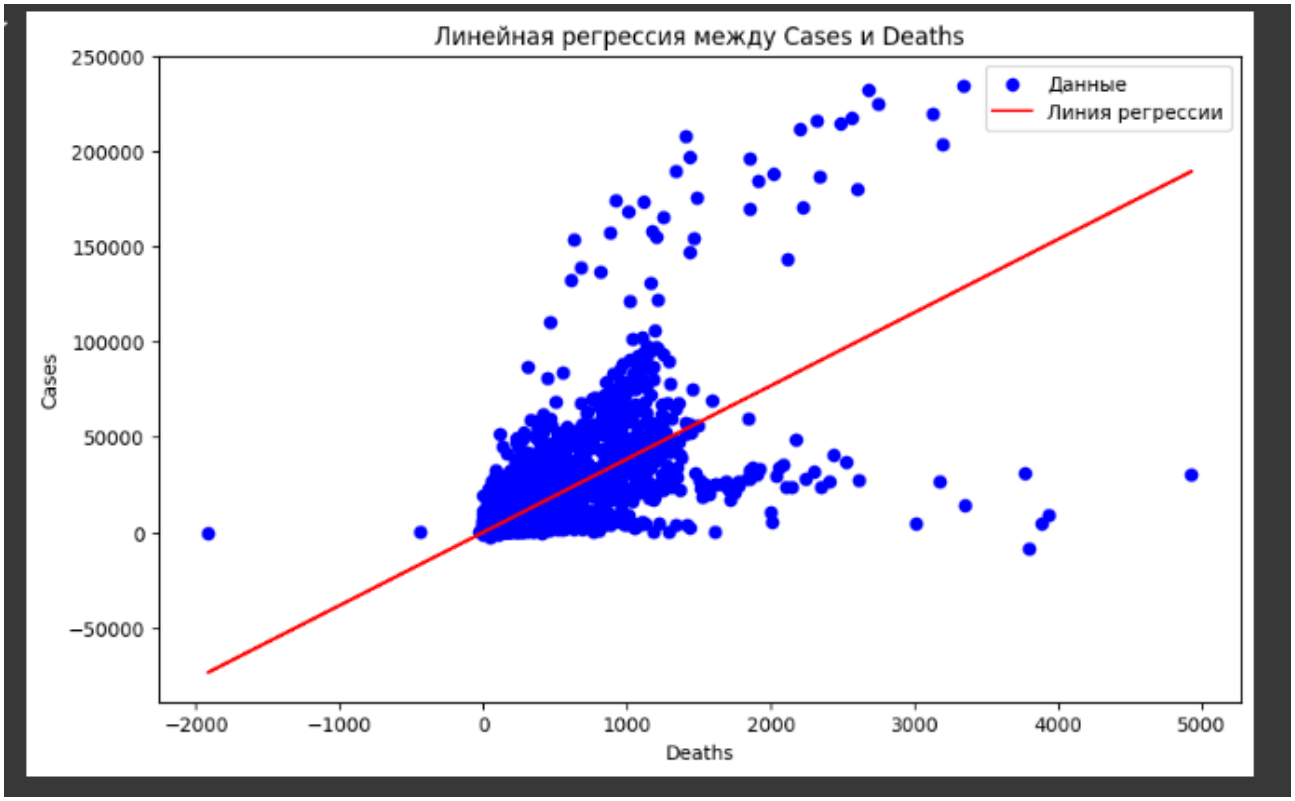

Рисунок 1.8 – Результат выполнения программы

1.7 Задача №3.1

Решение и результат программы представлены на Рисунке 1.9.

```
region_groups = [data[data['region'] == region]['bmi'] for region in data['region'].unique()]
anova_result = stats.f_oneway(*region_groups)
print(anova_result)
```

```
F_onewayResult(statistic=39.49505720170283, pvalue=1.881838913929143e-24)
```

Рисунок 1.9 – Программа и результат ее выполнения

1.8 Задача №3.2

Решение и результат программы представлены на Рисунке 1.10.

```
[6] model = ols('bmi ~ C(region)', data=data).fit()
    anova_results = anova_lm(model)
    print(anova_results)
```

	df	sum_sq	mean_sq	F	PR(>F)
C(region)	3.0	4055.880631	1351.960210	39.495057	1.881839e-24
Residual	1334.0	45664.319755	34.231124	NaN	NaN

Рисунок 1.10 –Программа и результат ее выполнения

1.9 Задача №3.3

Решение и результат программы представлены на Рисунке 1.11.

```python
regions = data['region'].unique()
region_pairs = list(combinations(regions, 2))
alpha = 0.05
k = len(region_pairs)
alpha_bonferroni = alpha / k
for region1, region2 in region_pairs:
    group1 = data[data['region'] == region1]['bmi']
    group2 = data[data['region'] == region2]['bmi']
    t_stat, p_value = stats.ttest_ind(group1, group2)
    print(f"Сравнение регионов: {region1} и {region2}")
    print(f"t-статистика: {t_stat:.4f}, p-значение: {p_value:.4f}")
    if p_value < alpha_bonferroni:
        print(f"Результат значим при уровне значимости {alpha_bonferroni:.4f} (с поправкой Бонферрони)")
    else:
        print(f"Результат незначим при уровне значимости {alpha_bonferroni:.4f} (с поправкой Бонферрони)")
    print("-" * 50)
```

```
Сравнение регионов: southwest и southeast
t-статистика: -5.9084, p-значение: 0.0000
Результат значим при уровне значимости 0.0083 (с поправкой Бонферрони)
--------------------------------------------------
Сравнение регионов: southwest и northwest
t-статистика: 3.2844, p-значение: 0.0011
Результат значим при уровне значимости 0.0083 (с поправкой Бонферрони)
--------------------------------------------------
Сравнение регионов: southwest и northeast
t-статистика: 3.1169, p-значение: 0.0019
Результат значим при уровне значимости 0.0083 (с поправкой Бонферрони)
--------------------------------------------------
Сравнение регионов: southeast и northwest
t-статистика: 9.2565, p-значение: 0.0000
Результат значим при уровне значимости 0.0083 (с поправкой Бонферрони)
--------------------------------------------------
Сравнение регионов: southeast и northeast
t-статистика: 8.7909, p-значение: 0.0000
Результат значим при уровне значимости 0.0083 (с поправкой Бонферрони)
--------------------------------------------------
Сравнение регионов: northwest и northeast
t-статистика: 0.0603, p-значение: 0.9519
Результат незначим при уровне значимости 0.0083 (с поправкой Бонферрони)
--------------------------------------------------
```

Рисунок 1.11 – Программа и результат ее выполнения

1.10 Задача №3.4

Решение и результат программы представлены на Рисунке 1.12, 1.13.

```python
tukey = pairwise_tukeyhsd(endog=data['bmi'], groups=data['region'], alpha=0.05)
print(tukey)
fig = tukey.plot_simultaneous(figsize=(8, 6))
plt.title("Результаты пост-хок теста Тьюки для индекса массы тела (BMI) между регионами")
plt.xlabel("Разность средних значений BMI")
plt.grid(True)
plt.show()
```

Рисунок 1.12 – Программа

```
   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
-----------------------------------------------------
```

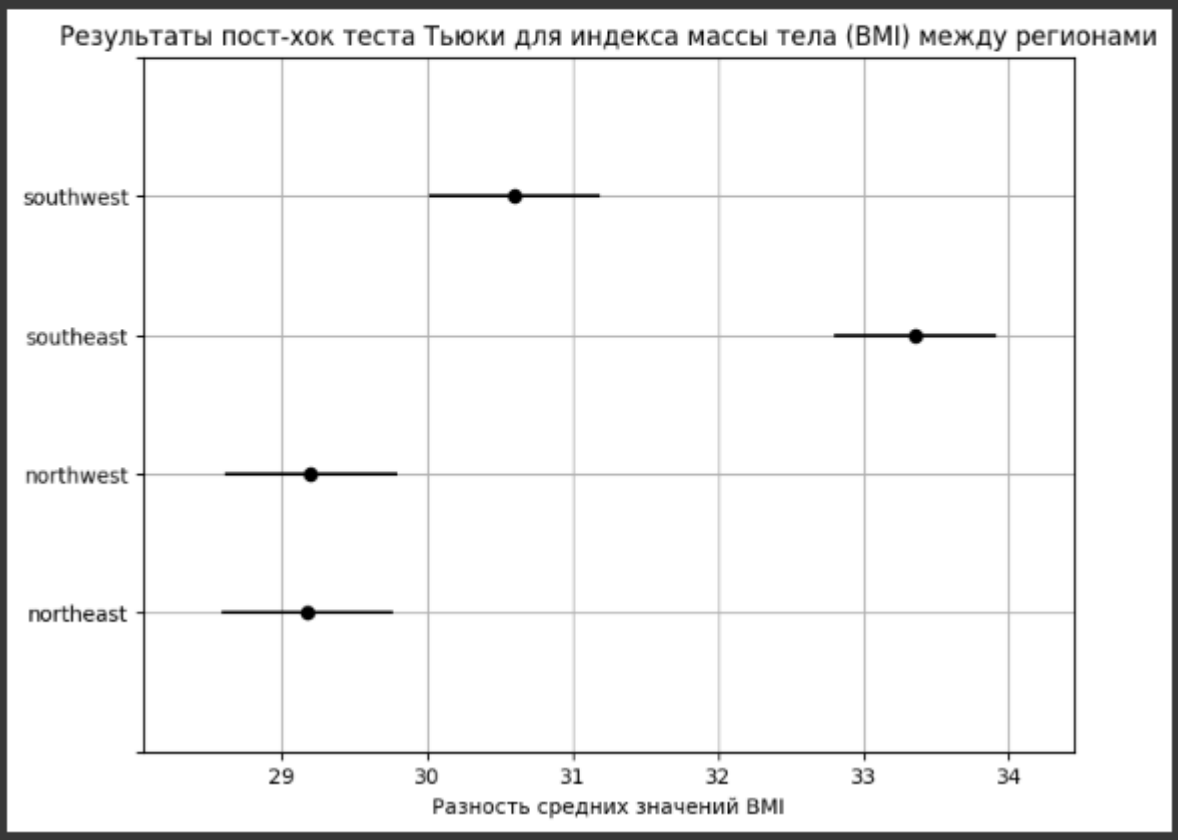

Рисунок 1.12 – Результат выполнения

1.11 Задача №3.5

Решение и результат программы представлены на Рисунке 1.13.

```
[9] print(data.isnull().sum())
    data['region'] = data['region'].astype('category')
    data['sex'] = data['sex'].astype('category')
    formula = 'bmi ~ C(region) + C(sex) + C(region):C(sex)'
    model = ols(formula, data=data).fit()
    anova_results = anova_lm(model, typ=2)
    print(anova_results)
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
                     sum_sq      df          F        PR(>F)
C(region)       4034.975135     3.0  39.398134  2.163195e-24
C(sex)            86.007035     1.0   2.519359  1.126940e-01
C(region):C(sex)  174.157808     3.0   1.700504  1.650655e-01
Residual        45404.154911  1330.0        NaN           NaN
```

Рисунок 1.13 – Программа и результат

1.12 Задача №3.6

Решение и результат программы представлены на Рисунке 1.14, 1.15.

```
tukey = pairwise_tukeyhsd(endog=data['bmi'], groups=data['region'].astype(str) + data['sex'].astype(str), alpha=0.05)
print(tukey)
tukey.plot_simultaneous()
plt.title('Tukey HSD Test Results')
plt.show()
```

Рисунок 1.14 – Программа

```
northeastfemale   northwestmale  -0.2042    1.0 -2.1811  1.7728  False
northeastfemale southeastfemale   3.3469    0.0    1.41  5.2839   True
northeastfemale   southeastmale   4.6657    0.0  2.7634   6.568   True
northeastfemale southwestfemale   0.7362 0.9497 -1.2377    2.71  False
northeastfemale   southwestmale   1.8051 0.1007 -0.1657   3.776  False
  northeastmale northwestfemale   0.2534 0.9999 -1.7083  2.2152  False
  northeastmale   northwestmale   0.0956    1.0 -1.8752  2.0665  False
  northeastmale southeastfemale   3.6467    0.0  1.7159  5.5775   True
  northeastmale   southeastmale   4.9655    0.0  3.0695  6.8614   True
  northeastmale southwestfemale    1.036 0.7515 -0.9318  3.0037  False
  northeastmale   southwestmale   2.1049 0.0258  0.1402  4.0697   True
northwestfemale   northwestmale  -0.1578    1.0 -2.1257    1.81  False
northwestfemale southeastfemale   3.3933    0.0  1.4656   5.321   True
northwestfemale   southeastmale    4.712    0.0  2.8192  6.6049   True
northwestfemale southwestfemale   0.7825 0.9294 -1.1822  2.7473  False
northwestfemale   southwestmale   1.8515 0.0806 -0.1103  3.8132  False
  northwestmale southeastfemale   3.5511    0.0  1.6141  5.4881   True
  northwestmale   southeastmale   4.8698    0.0  2.9676  6.7721   True
  northwestmale southwestfemale   0.9403 0.8354 -1.0335  2.9142  False
  northwestmale   southwestmale   2.0093  0.042  0.0385  3.9801   True
southeastfemale   southeastmale   1.3187 0.3823  -0.542  3.1795  False
southeastfemale southwestfemale  -2.6108 0.0011 -4.5446 -0.6769   True
southeastfemale   southwestmale  -1.5418 0.2304 -3.4726   0.389  False
  southeastmale southwestfemale  -3.9295    0.0 -5.8286 -2.0304   True
  southeastmale   southwestmale  -2.8606 0.0001 -4.7565 -0.9646   True
southwestfemale   southwestmale    1.069 0.7201 -0.8988  3.0367  False
-----------------------------------------------------------------------
```

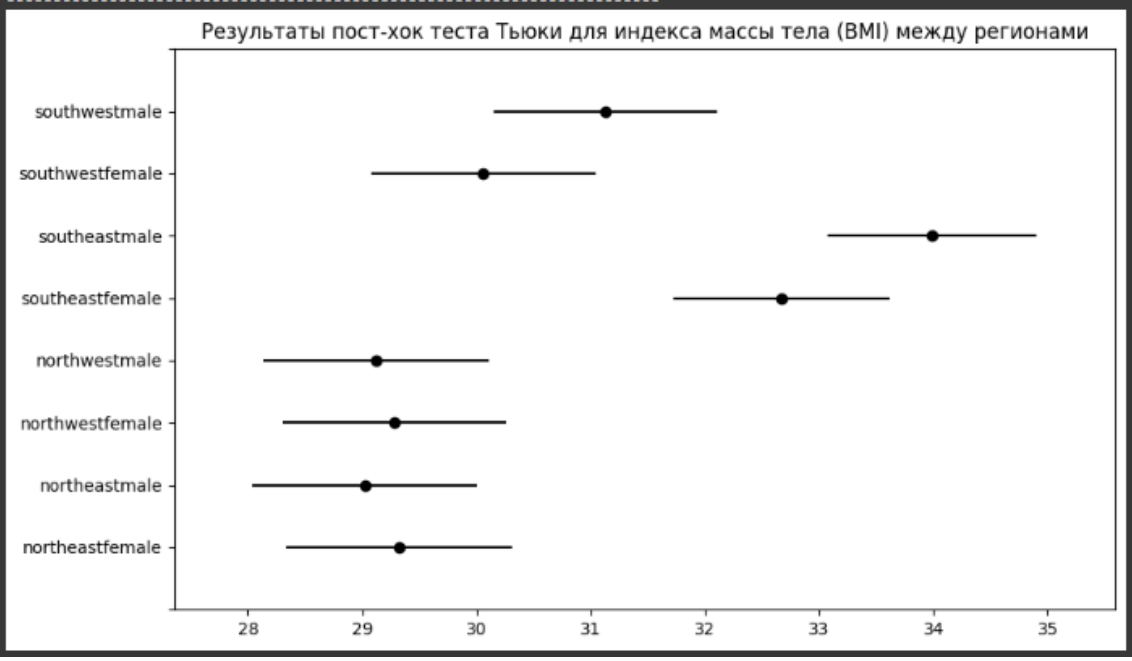

Рисунок 1.15 – Результат выполнения программы