|  |  |  |
| --- | --- | --- |
| ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ: |  |  |

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| доцент, к.т.н., доцент | / |  | / |  | / | В. В. Мышко |
| (должность, учёная степень, звание) |  | (подпись) |  | (дата защиты) |  | (инициалы, фамилия) |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

«Многофакторный регрессионный анализ»

ПО КУРСУ: «ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ»

|  |  |  |  |
| --- | --- | --- | --- |
| РАБОТУ ВЫПОЛНИЛ СТУДЕНТ: | 4134К | / | Самарин Д. В. |
|  | (номер группы) |  | (инициалы, фамилия) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | / |  | / | 18.02.2025 |
|  |  | (подпись студента) |  | (дата отчета) |

Цель

Целью работы является построение уравнения регрессии в виде линейного алгебраического полинома от двух переменных на основе экспериментальных данных, оценка адекватности модели по критерию Фишера, проверка значимости факторов регрессии по критерию Стьюдента, а также выполнение селекции факторов с последующей переоценкой модели в случае выявления незначимых переменных.

Задание на лабораторную работу

На основе заданного массива данных:

• построить уравнение регрессии в виде линейного алгебраического полинома от двух переменных;

• проверить адекватность уравнения регрессии;

• проверить значимость факторов регрессии.

Расчеты произвести в матричной форме.

Порядок выполнения задания:

1. Выполнить центрирование факторов (массив экспериментальных данных, таблица 5.1);

2. Составить матричное уравнение с вектором неизвестных оценок коэффициентов регрессии;

3. Найти оценки коэффициентов регрессии посредством решения матричного уравнения;

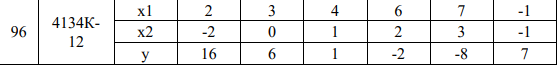
4. Проверить адекватность построенного уравнения регрессии экспериментальным данным по критерию Фишера при уровне значимости α = 0,05;

5. Выполнить селекцию факторов по критерию Стьюдента при таком же уровне значимости;

6. Повторно проверить адекватность уравнения регрессии после исключения незначимых факторов.

Ход работы

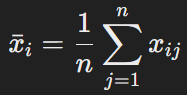
Вариант 96

****

Решение:

**Центрирование факторов:**

Для каждого фактора xi​ вычисляется среднее значение:



и затем каждый элемент преобразуется по формуле:



**Матричная форма модели:**

Модель регрессии задаётся уравнением:



где β0​ – интерсепт, β1​ и β2 – коэффициенты регрессии.

**Оценка коэффициентов (метод наименьших квадратов):**

В матричной форме решение нормального уравнения:

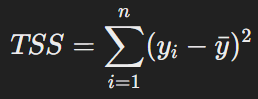


где X – дизайн-матрица, составленная из столбца единиц и центрированных факторов.

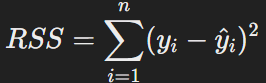
**Критерий Фишера:**

Для проверки адекватности модели вычисляются:

* Общая сумма квадратов (TSS):



* Остаточная сумма квадратов (RSS):



* Объяснённая сумма квадратов (ESS):



F-статистика вычисляется как:



где dfreg​ – число степеней свободы регрессии, dferror=n−(число параметров).

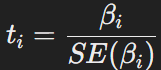
**Критерий Стьюдента:**

Стандартная ошибка коэффициента:



где MSE=RSS/dferror​.

t-статистика для коэффициента:



и соответствующий p-value рассчитывается как:



**Результаты выполнения работы**

В ходе работы был реализован алгоритм.

С работоспособностью алгоритма можно ознакомиться по ссылке ниже:

<https://fivelar.streamlit.app/>

**Исходные данные:**

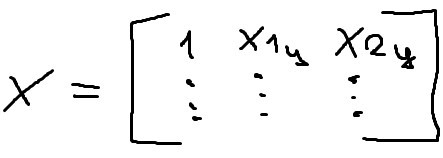
|  |
| --- |
| x1 = [ 2 3 4 6 7 -1]  x2 = [-2 0 1 2 3 -1]  y = [16 6 1 -2 -8 7] |

**Центрирование факторов:**

|  |
| --- |
| x1\_centered = [-1.5 -0.5 0.5 2.5 3.5 -4.5]  x2\_centered = [-2.5 -0.5 0.5 1.5 2.5 -1.5] |

**Построение модели и оценка коэффициентов:**

Составлена дизайн-матрица X:



Решением нормального уравнения получены оценки коэффициентов:

* β0 = 3.3333
* β1 = 1.1121
* β2 = −5.8362

Уравнение регрессии имеет вид:

y=3.3333+1.1121(x1​−3.5000)−5.8362(x2​−0.5000)

**Проверка адекватности модели (критерий Фишера):**

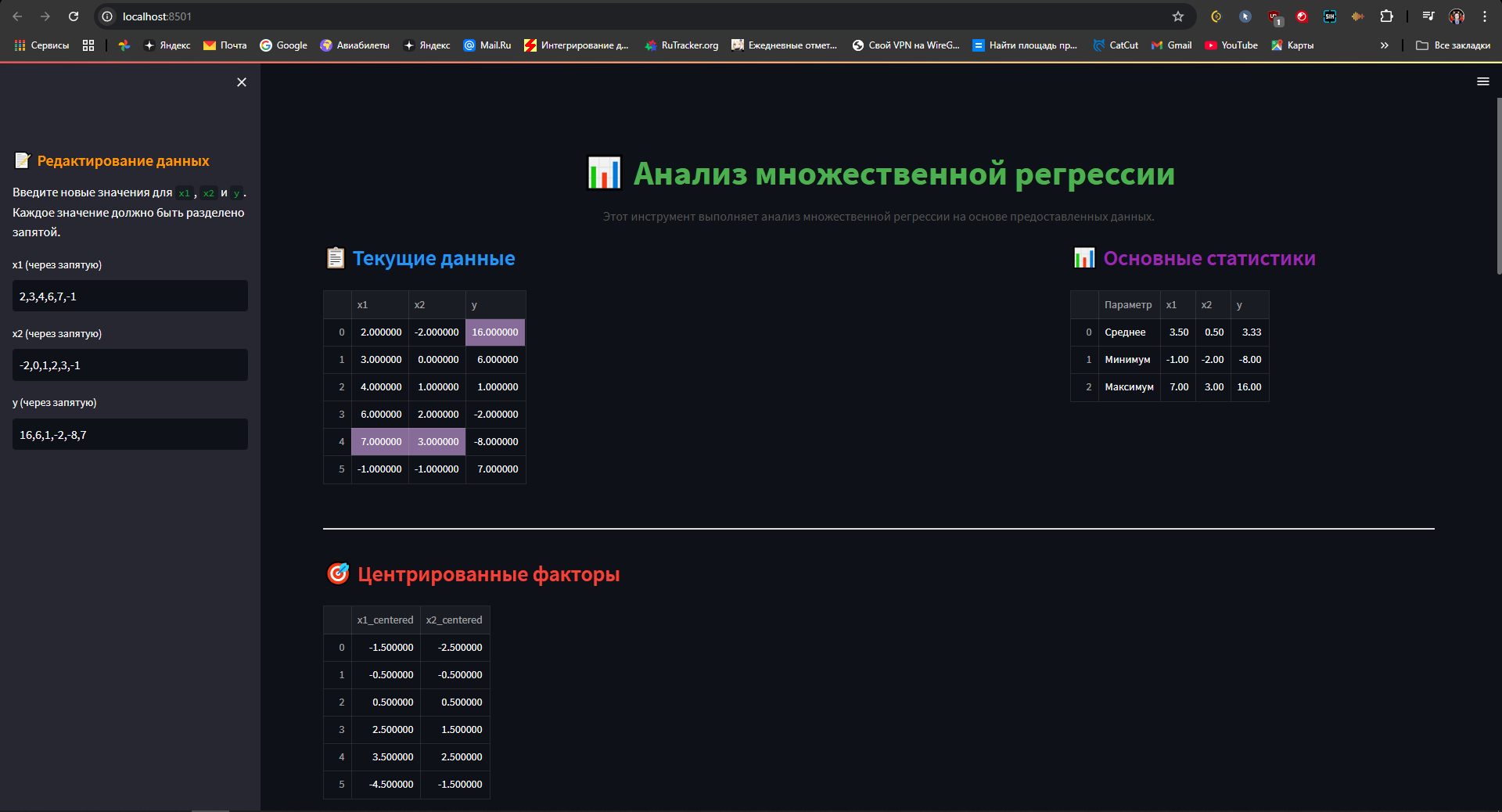
|  |
| --- |
| RSS = 0.9798850574712644  TSS = 343.33333333333337  ESS = 342.3534482758621  Степени свободы регрессии = 2 , степени свободы ошибок = 3  MSE = 0.3266283524904215  F статистика = 524.0718475073313  Критическое F (при α = 0.05) = 9.552094495921155  Модель адекватна на уровне значимости α = 0.05 |

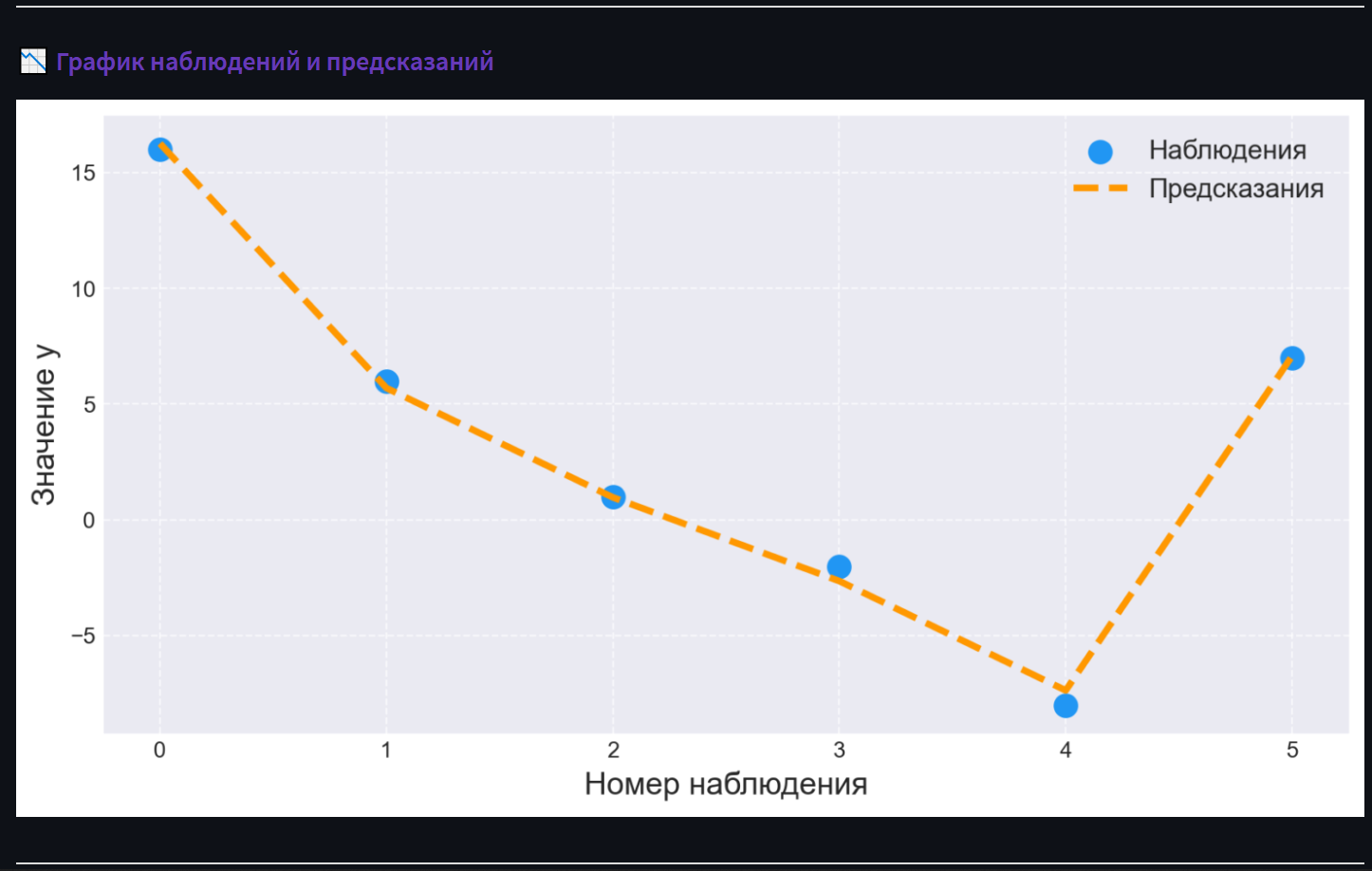
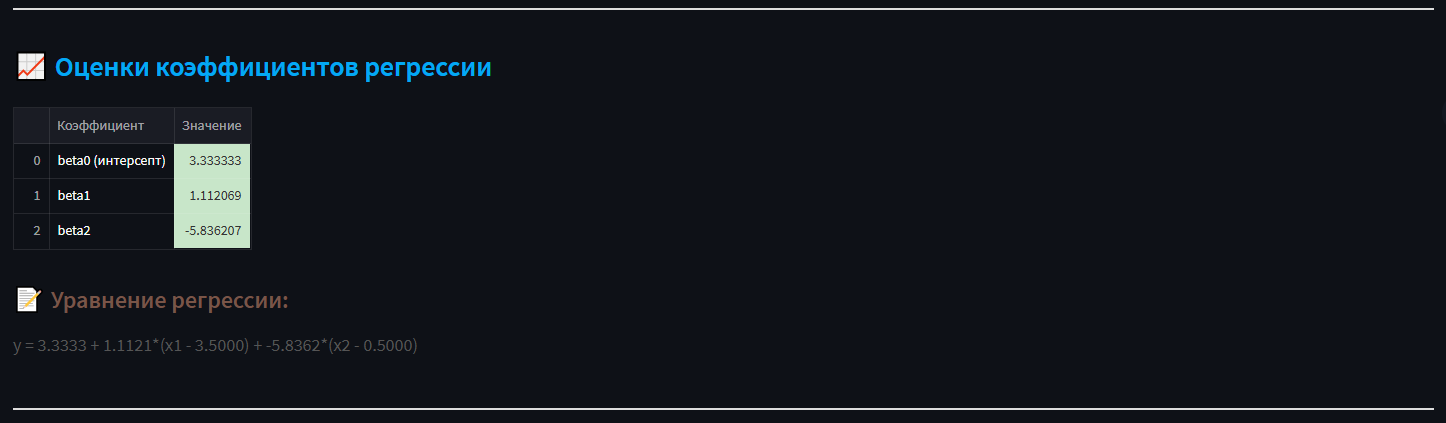
**Проверка значимости коэффициентов (критерий Стьюдента):**

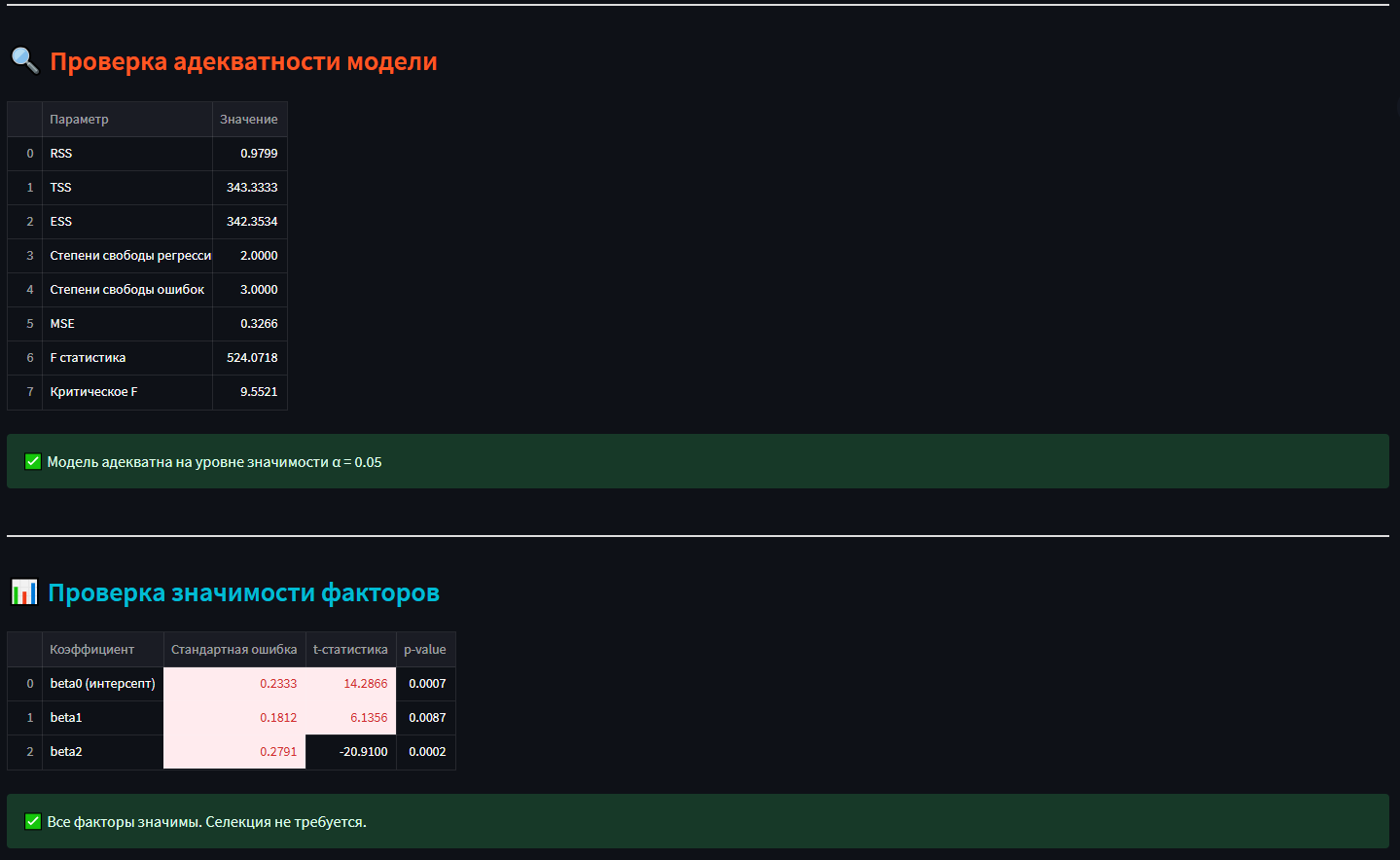
|  |
| --- |
| Стандартные ошибки коэффициентов: [0.23331965 0.18124721 0.27911062]  t-статистики коэффициентов: [ 14.28655213 6.13564721 -20.91001359]  p-value для коэффициентов: [0.00074316, 0.00870649, 0.00023925]  (Значимым считается фактор, если p-value < 0.05)  Незначимые факторы (индексы коэффициентов): [] |

Так как все факторы значимы (p-value всех коэффициентов меньше 0.05), селекция не требуется.

Результат работы программы:







**Анализ полученных результатов:**

Проведённый многофакторный регрессионный анализ позволяет оценить, как два фактора – x1 и x2 – влияют на зависимую переменную y. Для повышения стабильности оценок и уменьшения мультиколлинеарности перед построением модели были выполнены следующие шаги:

**Центрирование факторов:**

Вычитание среднего значения из каждого наблюдения для факторов x1​ и x2​ позволяет устранить влияние различий в масштабах переменных. Это важно, так как центрирование приводит к тому, что интерсепт (β0​) интерпретируется как значение y при средних значениях факторов, а коэффициенты при переменных отражают отклонение от среднего. В нашем случае центрированные значения показали, что отклонения варьируются как в положительную, так и в отрицательную сторону, что обеспечивает сбалансированное распределение данных вокруг центра.

**Оценка коэффициентов регрессии:**

Составленная дизайн-матрица X, включающая константу и центрированные факторы, используется для решения нормального уравнения. В результате получены следующие оценки коэффициентов:

* β0​=3.3333
* β1=1.1121
* β2=−5.8362

Это уравнение регрессии:

y=3.3333+1.1121(x1​−3.5000)−5.8362(x2​−0.5000)

Интерсепт здесь отражает среднее значение y при том, что x11​ и x2​ принимают свои средние значения. Коэффициент β1​ показывает, что при увеличении x1​ на единицу (относительно среднего), y увеличивается примерно на 1.1121 единиц, при прочих равных условиях. Отрицательное значение β2 говорит о том, что при увеличении x2​ на единицу (также относительно среднего) y уменьшается на 5.8362 единиц.

**Проверка адекватности модели по критерию Фишера:**

Для проверки того, насколько построенная модель способна объяснить изменчивость зависимой переменной, были рассчитаны следующие показатели:

* TSS (Total Sum of Squares) отражает общую изменчивость исходных значений y вокруг их среднего.
* RSS (Residual Sum of Squares) измеряет ту часть изменчивости, которая не объясняется моделью, то есть разницу между наблюдаемыми значениями и значениями, предсказанными моделью.
* ESS (Explained Sum of Squares) – это разница между TSS и RSS, показывающая, какая часть общей изменчивости объясняется моделью.

В нашем случае:

* TSS = 343.3333
* RSS = 0.9799
* ESS = 342.3534

Полученные значения свидетельствуют о том, что модель объясняет подавляющую часть вариации y (ESS почти равна TSS). Далее, при расчёте F-статистики:



где число степеней свободы регрессии dfreg=2 (так как два фактора) и степени свободы ошибок dferror=3. Критическое значение F при уровне значимости α=0.05 составляет 9.55. Поскольку вычисленная F-статистика существенно превышает критическое значение, можно заключить, что модель является статистически значимой и адекватно описывает зависимость y от факторов x1​ и x2​.

**Проверка значимости отдельных коэффициентов (критерий Стьюдента):**

Для каждого коэффициента регрессии были вычислены стандартные ошибки, t-статистика и соответствующие p-value. Стандартные ошибки для коэффициентов оказались достаточно малы ([0.2333, 0.1812, 0.2791]), что привело к высоким значениям t-статистик:

* Для β0​ t = 14.29 (p ≈ 0.00074)
* Для β1​ t = 6.14 (p ≈ 0.00871)
* Для β2​ t = -20.91 (p ≈ 0.00024)

Все полученные p-value значительно меньше уровня значимости 0.05, что позволяет утверждать, что все коэффициенты являются статистически значимыми. Это означает, что оба фактора x1 и x2​ вносят существенный вклад в объяснение изменчивости y. Соответственно, селекция факторов не требуется.

Полученные результаты демонстрируют, что построенная модель имеет высокую объяснительную способность – практически вся вариация зависимой переменной y объясняется выбранными факторами. Высокая F-статистика и значимость каждого коэффициента подтверждают надежность модели и корректность проведённых вычислений. Центрирование факторов способствовало улучшению интерпретируемости модели, поскольку позволило интерпретировать интерсепт как значение y при средних значениях x1​ и x2​.

**Вывод:**

Выполнив лабораторную работу, можно сделать вывод, что модель успешно описывает исследуемую зависимость, и все этапы анализа выполнены корректно. Результаты позволяют использовать данную модель для дальнейших предсказаний и анализа влияния факторов на результат.

Листинг кода:

|  |
| --- |
| import numpy as np  import scipy.stats as stats  import streamlit as st  import pandas as pd  import matplotlib.pyplot as plt  from matplotlib import style  # Настройка страницы Streamlit  st.set\_page\_config(      page\_title="Множественная регрессия",      layout="wide",      initial\_sidebar\_state="expanded"  )  # Стиль для графиков  style.use("seaborn-v0\_8-darkgrid")  # Заголовок с иконкой  st.markdown("""  <h1 style="text-align: center; color: #4CAF50;">      📊 Анализ множественной регрессии  </h1>  <p style="text-align: center; font-size: 16px; color: #555;">      Этот инструмент выполняет анализ множественной регрессии на основе предоставленных данных.  </p>  """, unsafe\_allow\_html=True)  # Исходные данные по умолчанию  default\_data = {      "x1": [2, 3, 4, 6, 7, -1],      "x2": [-2, 0, 1, 2, 3, -1],      "y": [16, 6, 1, -2, -8, 7]  }  # Боковая панель для редактирования данных  with st.sidebar:      st.markdown("""      <h2 style="color: #FF9800;">📝 Редактирование данных</h2>      """, unsafe\_allow\_html=True)      st.markdown("Введите новые значения для `x1`, `x2` и `y`. Каждое значение должно быть разделено запятой.")      x1\_input = st.text\_input("x1 (через запятую)", value=",".join(map(str, default\_data["x1"])))      x2\_input = st.text\_input("x2 (через запятую)", value=",".join(map(str, default\_data["x2"])))      y\_input = st.text\_input("y (через запятую)", value=",".join(map(str, default\_data["y"])))  # Преобразование ввода в массивы NumPy  try:      x1 = np.array(list(map(float, x1\_input.split(","))))      x2 = np.array(list(map(float, x2\_input.split(","))))      y = np.array(list(map(float, y\_input.split(","))))      if len(x1) != len(x2) or len(x1) != len(y):          st.error("Ошибка: Длины массивов x1, x2 и y должны совпадать!")          st.stop()  except ValueError:      st.error("Ошибка: Введите числовые значения, разделенные запятыми!")      st.stop()  # Главная область: Отображение текущих данных  col1, col2 = st.columns([2, 1])  with col1:      st.markdown("""      <h3 style="color: #2196F3;">📋 Текущие данные</h3>      """, unsafe\_allow\_html=True)      data = pd.DataFrame({"x1": x1, "x2": x2, "y": y})      st.dataframe(data.style.highlight\_max(axis=0, color="#876c99"))  with col2:      st.markdown("""      <h3 style="color: #9C27B0;">📊 Основные статистики</h3>      """, unsafe\_allow\_html=True)      stats\_data = pd.DataFrame({          "Параметр": ["Среднее", "Минимум", "Максимум"],          "x1": [np.mean(x1), np.min(x1), np.max(x1)],          "x2": [np.mean(x2), np.min(x2), np.max(x2)],          "y": [np.mean(y), np.min(y), np.max(y)]      })      styled\_stats\_data = stats\_data.style.format({          "x1": "{:.2f}",          "x2": "{:.2f}",          "y": "{:.2f}"      }).set\_table\_styles([          {"selector": "th", "props": [("background-color", "#E3F2FD"), ("color", "#333")]}      ])      st.dataframe(styled\_stats\_data)  # Центрирование факторов  mean\_x1 = np.mean(x1)  mean\_x2 = np.mean(x2)  x1\_centered = x1 - mean\_x1  x2\_centered = x2 - mean\_x2  st.markdown("""  <hr style="border: 1px solid #ddd;">  <h3 style="color: #F44336;">🎯 Центрированные факторы</h3>  """, unsafe\_allow\_html=True)  centered\_data = pd.DataFrame({"x1\_centered": x1\_centered, "x2\_centered": x2\_centered})  st.dataframe(centered\_data.style.bar(color="#FF9800"))  # Построение модели  n = len(y)  X = np.column\_stack((np.ones(n), x1\_centered, x2\_centered))  k = 2  # число факторов  XtX = X.T @ X  XtX\_inv = np.linalg.inv(XtX)  beta\_hat = XtX\_inv @ (X.T @ y)  st.markdown("""  <hr style="border: 1px solid #ddd;">  <h3 style="color: #03A9F4;">📈 Оценки коэффициентов регрессии</h3>  """, unsafe\_allow\_html=True)  coefficients = pd.DataFrame({      "Коэффициент": ["beta0 (интерсепт)", "beta1", "beta2"],      "Значение": beta\_hat  })  styled\_coefficients = coefficients.style.map(      lambda x: "background-color: #C8E6C9; color: #333;" if isinstance(x, (int, float)) and abs(x) > 1 else "",      subset=["Значение"]  )  st.dataframe(styled\_coefficients)  # Уравнение регрессии  equation = f"y = {beta\_hat[0]:.4f} + {beta\_hat[1]:.4f}\*(x1 - {mean\_x1:.4f}) + {beta\_hat[2]:.4f}\*(x2 - {mean\_x2:.4f})"  st.markdown(f"<h4 style='color: #795548;'>📝 Уравнение регрессии:</h4><p style='font-size: 18px; color: #555;'>{equation}</p>", unsafe\_allow\_html=True)  # График наблюдений и предсказаний  st.markdown("""  <hr style="border: 1px solid #ddd;">  <h3 style="color: #673AB7;">📉 График наблюдений и предсказаний</h3>  """, unsafe\_allow\_html=True)  y\_pred = X @ beta\_hat  fig, ax = plt.subplots(figsize=(10, 5))  ax.scatter(range(len(y)), y, label="Наблюдения", color="#2196F3", s=100)  ax.plot(range(len(y)), y\_pred, label="Предсказания", color="#FF9800", linestyle="--", linewidth=3)  ax.set\_xlabel("Номер наблюдения", fontsize=14)  ax.set\_ylabel("Значение y", fontsize=14)  ax.legend(fontsize=12)  ax.grid(True, linestyle="--", alpha=0.7)  st.pyplot(fig)  # Проверка адекватности модели  st.markdown("""  <hr style="border: 1px solid #ddd;">  <h3 style="color: #FF5722;">🔍 Проверка адекватности модели</h3>  """, unsafe\_allow\_html=True)  residuals = y - y\_pred  RSS = np.sum(residuals\*\*2)  TSS = np.sum((y - np.mean(y))\*\*2)  ESS = TSS - RSS  df\_reg = k  df\_error = n - (k + 1)  MSE = RSS / df\_error  F\_stat = (ESS / df\_reg) / MSE  alpha = 0.05  F\_crit = stats.f.ppf(1 - alpha, df\_reg, df\_error)  fisher\_results = pd.DataFrame({      "Параметр": ["RSS", "TSS", "ESS", "Степени свободы регрессии", "Степени свободы ошибок", "MSE", "F статистика", "Критическое F"],      "Значение": [RSS, TSS, ESS, df\_reg, df\_error, MSE, F\_stat, F\_crit]  })  styled\_fisher\_results = fisher\_results.style.format({      "Значение": "{:.4f}"  # Форматируем только числовой столбец  }).set\_table\_styles([      {"selector": "th", "props": [("background-color", "#FFF3E0"), ("color", "#333")]}  ])  st.dataframe(styled\_fisher\_results)  if F\_stat > F\_crit:      st.success("✅ Модель адекватна на уровне значимости α = 0.05")  else:      st.error("❌ Модель неадекватна на уровне значимости α = 0.05")  # Проверка значимости факторов  st.markdown("""  <hr style="border: 1px solid #ddd;">  <h3 style="color: #00BCD4;">📊 Проверка значимости факторов</h3>  """, unsafe\_allow\_html=True)  SE\_beta = np.sqrt(np.diag(MSE \* XtX\_inv))  t\_stats = beta\_hat / SE\_beta  p\_values = [2 \* (1 - stats.t.cdf(np.abs(t), df=df\_error)) for t in t\_stats]  student\_results = pd.DataFrame({      "Коэффициент": ["beta0 (интерсепт)", "beta1", "beta2"],      "Стандартная ошибка": SE\_beta,      "t-статистика": t\_stats,      "p-value": p\_values  })  # Применяем форматирование только к числовым столбцам  styled\_student\_results = (      student\_results.style      .format({          "Стандартная ошибка": "{:.4f}",          "t-статистика": "{:.4f}",          "p-value": "{:.4f}"      })      .map(          lambda x: "background-color: #FFEBEE; color: #D32F2F;"                    if isinstance(x, (int, float)) and x > 0.05 else "",          subset=["Стандартная ошибка", "t-статистика", "p-value"]  # Применяем только к числовым столбцам      )  )  st.dataframe(styled\_student\_results)  insignificant = []  for i, p in enumerate(p\_values):      if p > alpha and i > 0:  # Пропускаем интерсепт          insignificant.append(i)  if insignificant:      st.warning(f"⚠️ Незначимые факторы (индексы коэффициентов): {insignificant}")  else:      st.success("✅ Все факторы значимы. Селекция не требуется.")  # Селекция факторов  if insignificant:      cols = [0] + [i for i in range(1, len(beta\_hat)) if i not in insignificant]      X\_new = X[:, cols]      XtX\_new = X\_new.T @ X\_new      XtX\_inv\_new = np.linalg.inv(XtX\_new)      beta\_hat\_new = XtX\_inv\_new @ (X\_new.T @ y)      st.subheader("Новая модель после исключения незначимых факторов")      new\_coefficients = pd.DataFrame({          "Коэффициент": [f"beta{i}" for i in cols],          "Значение": beta\_hat\_new      })      st.dataframe(new\_coefficients)      y\_pred\_new = X\_new @ beta\_hat\_new      residuals\_new = y - y\_pred\_new      RSS\_new = np.sum(residuals\_new\*\*2)      df\_error\_new = n - X\_new.shape[1]      MSE\_new = RSS\_new / df\_error\_new      TSS\_new = np.sum((y - np.mean(y))\*\*2)      ESS\_new = TSS\_new - RSS\_new      df\_reg\_new = X\_new.shape[1] - 1      F\_stat\_new = (ESS\_new / df\_reg\_new) / MSE\_new if df\_reg\_new > 0 else np.nan      F\_crit\_new = stats.f.ppf(1 - alpha, df\_reg\_new, df\_error\_new) if df\_reg\_new > 0 else np.nan      fisher\_results\_new = pd.DataFrame({          "Параметр": ["RSS", "TSS", "ESS", "Степени свободы регрессии", "Степени свободы ошибок", "MSE", "F статистика", "Критическое F"],          "Значение": [RSS\_new, TSS\_new, ESS\_new, df\_reg\_new, df\_error\_new, MSE\_new, F\_stat\_new, F\_crit\_new]      })      st.dataframe(fisher\_results\_new)      if F\_stat\_new > F\_crit\_new:          st.success("✅ Новая модель адекватна на уровне значимости α = 0.05")      else:          st.error("❌ Новая модель неадекватна на уровне значимости α = 0.05") |