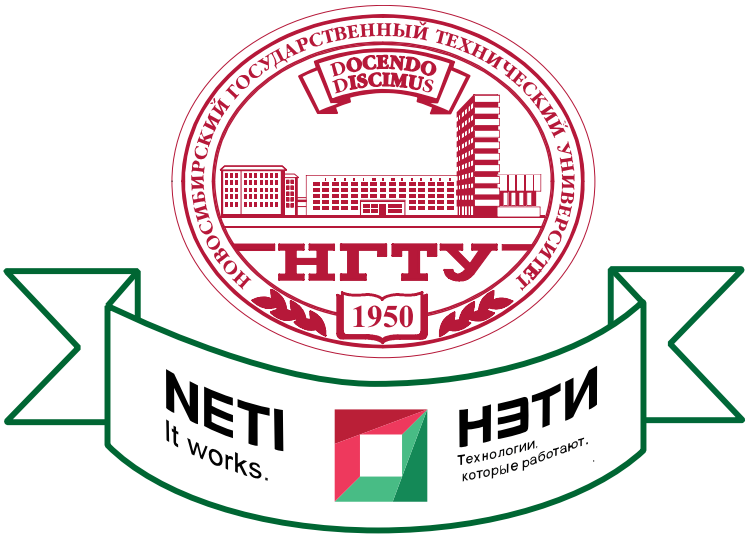
Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования

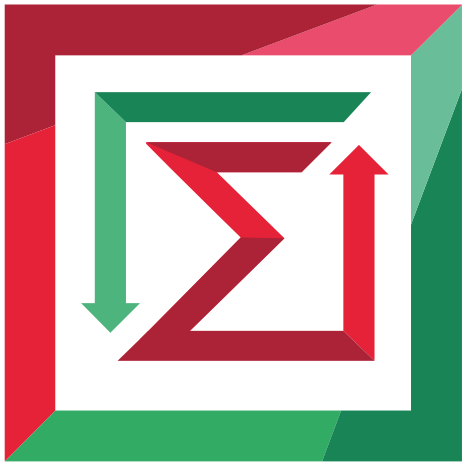
«Новосибирский государственный технический университет»



Кафедра теоретической и прикладной информатики

Лабораторная работа № 3

по дисциплине «Статистические методы анализа данных»



Факультет:	ПМИ
Группа:	ПМИ-02
Вариант:	6
Студент:	Сидоров Даниил,
	Дюков Богдан
Преподаватель:	Попов Александр Александрович.

Новосибирск

2026

1. Постановка задачи

1. Изменить модель регрессии, добавив в нее дополнительный регрессор, ранее не вошедший в состав модели, порождающей данные. Не генерируя новых данных, найти точечные оценки всех параметров расширенной модели. В дальнейшем при рассмотрении этой расширенной модели анализе должно быть показано, что параметр при дополнительном регрессоре незначим.

2. Построить доверительные интервалы для каждого параметра модели регрессии.

3. Проверить гипотезу о незначимости каждого параметра модели.

4. Проверить гипотезу о незначимости самой регрессии.

5. Рассчитать прогнозные значения для математического ожидания функции отклика для всего интервала действия одного из факторов, зафиксировав значения других факторов на границе или в центре области их определения.

6. По полученным в п. 5 прогнозным значениям построить графики прогнозных значений и доверительной полосы для математического ожидания функции отклика и для самого отклика.

7. Заново смоделировать исходные данные, увеличив мощность случайной помехи до 50…70 % от мощности полезного сигнала, и провести оценку параметров. Повторить пункты 3, 4 с новыми данными.

2. Ход работы

Добавление нового регрессора x_2^3

Старая модель:

$f(x_1, x_2, x_3) = (1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_1^2, x_2^2)$;

$\theta = (1, 2, 5, 4, 1.5, 2.5, 0.02, 0.01)^{\mathrm{T}}$;

$\eta(\mathrm{x}, \theta) = \theta^{\mathrm{T}} f(x_1, x_2, x_3)$.

Оценки старой модели:

theta	theta_hat	sigma_squared	sigma_hat_squared	F
1	1,501426421	1,957897413	2,22151394	1,134642666
2	2,336565661			
5	5,062817987			
4	3,938056574			
1,5	1,354423461			
2,5	2,27845587			
0,02	-0,340012046			
0,01	0,026997102			

$F = \frac{\widehat{\sigma}^2}{\sigma^2} = 1.13 < F_T = 1{,}67;$

Модель является адекватной.

После добавления нового регрессора:

$f(x_1, x_2, x_3) = (1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_1^2, x_2^2, x_2^3);$

$\theta = (1, 2, 5, 4, 1.5, 2.5, 0.02, 0.01, 0)^{\mathrm{T}};$

$\eta(x, \theta) = \theta^{\mathrm{T}} f(x_1, x_2, x_3).$

Оценки расширенной модели:

theta	theta_hat	sigma_squared	sigma_hat_squared	F
1	1,501426421	1,957897413	2,311921867	1,180818694
2	2,336565661			
5	4,185311994			
4	3,938056574			
1,5	1,354423461			
2,5	2,27845587			
0,02	-0,340012046			
0,01	0,026997102			
0	0,961661846			

$F = \frac{\widehat{\sigma}^2}{\sigma^2} = 1.18 < F_T = 1{,}67;$

Модель является адекватной.

Построение доверительных интервалов

После преобразования формулы:

$$\left(\theta_j - \hat{\theta}_j\right)^{\mathrm{T}} \frac{1}{d_{jj}} \left(\theta_j - \hat{\theta}_j\right) \le \hat{\sigma}^2 F_{\alpha,1,n-m},\ d_{jj} = \left((X^T X)^{-1}\right)_{jj}$$

Получим двустороннее неравенство вида:

$$\hat{\theta}_j - t_{\alpha/2,f_R}\sigma\left(\hat{\theta}_j\right) \le \theta_j \le \hat{\theta}_j + t_{\alpha/2,f_R}\sigma\left(\hat{\theta}_j\right),$$

где $t_{\alpha/2,fR}$ - квантиль распределения Стьюдента, $fR = n - m = 24 - 9 = 15$, при уровне значимости 0.05. Квантиль получился равным ≈ 2.13. $\sigma\left(\hat{\theta}\right) = = \sqrt{\hat{\sigma}^2((X^T X)^{-1})_{jj}}$

left_interval	theta	theta_hat	right_interval
0,090534878	1	1,501426421	2,912317964
1,526348119	2	2,336565661	3,146783203
1,001767791	5	4,185311994	7,368856196
3,276516721	4	3,938056574	4,599596426
0,266319358	1,5	1,354423461	2,442527564
1,468238328	2,5	2,27845587	3,088673412
-1,743349994	0,02	-0,340012046	1,063325901
-1,457774198	0,01	0,026997102	1,511768402
-2,388585087	0	0,961661846	4,311908778

Проверка гипотезы о незначимости каждого параметра модели

Статистика в общем виде:

$$F = \frac{(RSS_H - RSS)/q}{RSS/(n-m)} = \frac{(A\theta - C)^{\mathrm{T}}\left[A(X^{\mathrm{T}}X)^{-1}A^{\mathrm{T}}\right]^{-1}(A\theta - C)}{q\hat{\sigma}^2}$$

Частный случай при H: $a^T\theta = 0$, где a^T- вектор-строка, в которой на j-м месте стоит единица, на остальных местах – нули. Тогда статистика:

$$F = \frac{(a^{\mathrm{T}}\hat{\theta})^{\mathrm{T}}(a^{\mathrm{T}}\hat{\theta})}{\hat{\sigma}^2 d_{jj}} = \frac{\left(\hat{\theta}_j\right)^2}{\hat{\sigma}^2 d_{jj}},$$

где d_{jj}- j-й диагональный элемент матрицы $(X^T X)^{-1}$. $F_{1,n-m} = 4{,}54$.

F_values	sign	F_quantile	Hypothesis
5,144829196	>	4,543077165	Не принимается
37,7835543	>	4,543077165	Не принимается
7,852067438	>	4,543077165	Не принимается
160,9911456	>	4,543077165	Не принимается
7,039116513	>	4,543077165	Не принимается
35,92758891	>	4,543077165	Не принимается
0,266694576	<	4,543077165	Принимается
0,001501984	<	4,543077165	Принимается
0,374318463	<	4,543077165	Принимается

В результате имеем то, что незначимы оказались параметры θ_6, θ_7, θ_8.

Проверка гипотезы о незначимости самой регрессии

Требуется установить, является ли регрессия с заданными регрессорами значимой, т. е. дает лучшее описание, чем модель среднего. Гипотеза H имеет вид: $\theta_0 = \theta_1 = \ldots = \theta_{m-1} = 0$. Тогда:

$$RSS = (Y - X\hat{\theta})^{\mathrm{T}}(Y - X\hat{\theta}) = (n-m)\hat{\sigma}^2,\ RSS_H = \left(Y - X\hat{\theta}_H\right)^{\mathrm{T}}\left(Y - X\hat{\theta}_H\right)$$

$$F = \frac{(RSS_H - RSS)/q}{RSS/(n-m)} < F_{\alpha,q,n-m},$$

где $F_{\alpha,q,n-m}$ – квантиль распределения Фишера, $q = m - 1 = 8$ при уровне значимости 0.05.

F_regression	sign	F_quantile_regression	Hypothesis
48,7394423	>	2,640796883	Не принимается

Расчет прогнозных значений для математического ожидания функции отклика

Рассчитаем прогнозные значения для математического ожидания функции отклика для всего интервала действия фактора x_2, зафиксировав значения фактора x_3на границе, а фактора x_1 в центре области определения. Формула расчета:

$$\eta(x, \hat{\theta}) = f^{\mathrm{T}}(x)\hat{\theta}$$

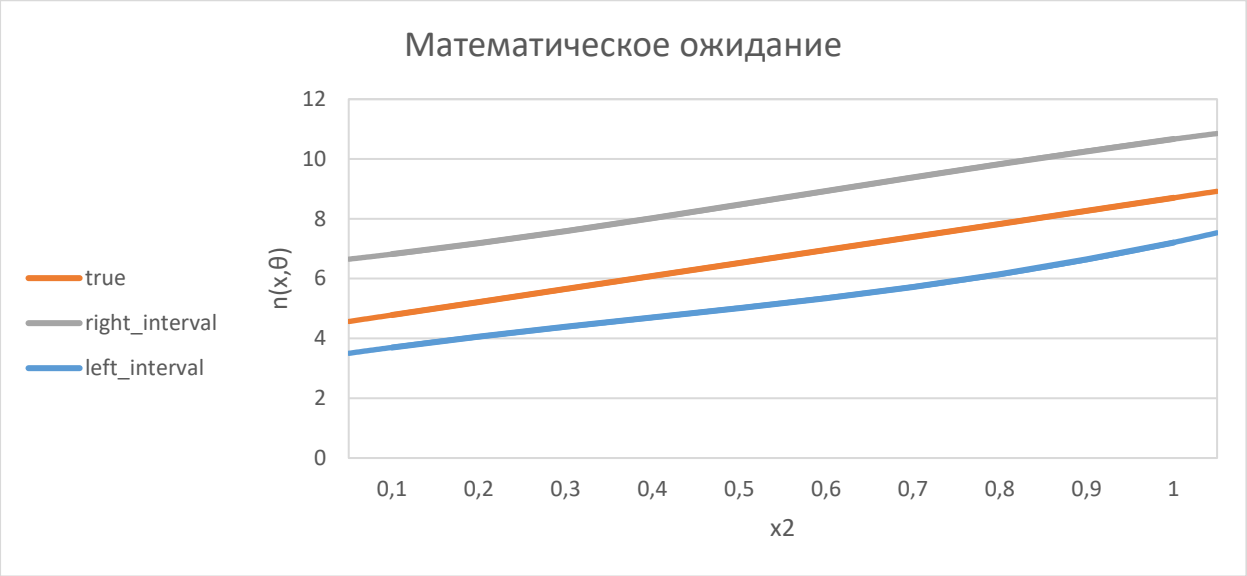
x1	x2	x3	$\eta(x, \hat{\theta})$
0	-1	1	0,319506258
0	-0,913043478	1	0,9086414
0	-0,826086957	1	1,458349395
0	-0,739130435	1	1,972424092
0	-0,652173913	1	2,454659341
0	-0,565217391	1	2,908848992
0	-0,47826087	1	3,338786893
0	-0,391304348	1	3,748266895
0	-0,304347826	1	4,141082848
0	-0,217391304	1	4,5210286
0	-0,130434783	1	4,891898002
0	-0,043478261	1	5,257484904
0	0,043478261	1	5,621583154
0	0,130434783	1	5,987986603
0	0,217391304	1	6,360489101
0	0,304347826	1	6,742884496
0	0,391304348	1	7,138966638
0	0,47826087	1	7,552529378
0	0,565217391	1	7,987366565
0	0,652173913	1	8,447272048
0	0,739130435	1	8,936039677
0	0,826086957	1	9,457463302
0	0,913043478	1	10,01533677
0	1	1	10,61345394

Графики прогнозных значений и доверительной полосы для математического ожидания функции отклика и для самого отклика

Интервальная оценка для истинного значения математического ожидания функции отклика:

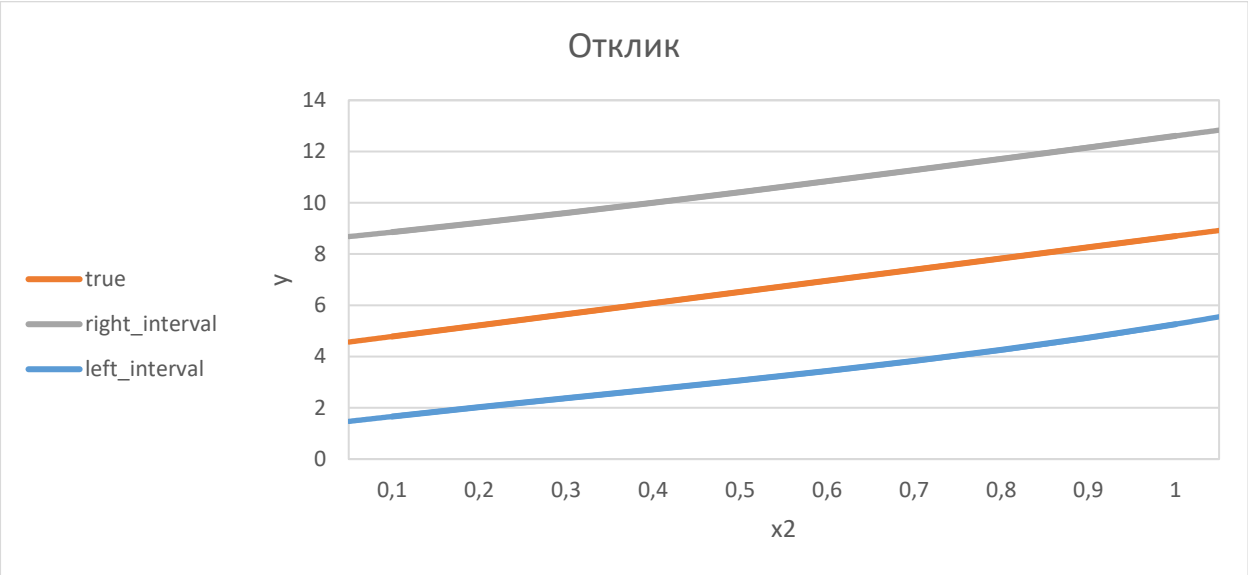
$$\eta(x,\ \hat{\theta}) - t_{\alpha/2,} f_R \sigma(\eta(x,\ \hat{\theta})) \le \eta(x,\ \theta) \le \eta(x,\ \hat{\theta}) + t_{\alpha/2,} f_R \sigma(\eta(x,\ \hat{\theta})),$$

где $t_{\alpha/2,fR}$ - квантиль распределения Стьюдента, $fR = n - m = 24 - 9 = 15$, при уровне значимости 0.05 и $\sigma\left(\eta\left(x, \hat{\theta}\right)\right) = \hat{\sigma}\sqrt{f^T(x) * (X^T X)^{-1} f(x)}$.



Для построения доверительного интервала для отклика будем использовать предыдущую формулу за тем исключением, что дисперсия оценки отклика будет определяться как:

$$\sigma^2(\hat{y}(x,\ \hat{\theta})) = \hat{\sigma}^2(1 + f^{\mathrm{T}}(x)(X^{\mathrm{T}}X)^{-1}f(x))$$



Моделирование с увеличенной мощностью случайной помехи

Пусть $\rho = 0.7$, тогда:

y	y_hat	y-y_hat	theta	theta_hat	sigma_squared	sigma_hat_squared
-11,4025911	-8,328386352	-3,074204752	1	2,876165873	27,41056378	32,36690614
0,138103151	-4,134044048	4,272147199	2	3,259313392		
2,139184299	-4,257471159	6,396655458	5	1,951716604		
-3,315422575	-0,063128855	-3,252293721	4	3,768228922		
-2,7135193	-3,341220145	0,627700846	1,5	0,955302469		
-6,789761971	0,853122159	-7,64288413	2,5	1,67105777		
-0,391516789	0,860860216	-1,252377005	0,02	-1,32704173		
8,980458625	5,05520252	3,925256105	0,01	0,073597333		
-11,27262693	-6,368391469	-4,904235465	0	3,598209149		
-1,382369624	1,168066375	-2,550435999				
-1,050578692	-1,657423621	0,606844929				
14,0758054	5,879034223	8,196771175				
-3,765407456	-0,110672978	-3,654734478				
10,25232098	7,425784866	2,826536119				
7,288035958	4,731460037	2,55657592				
9,19059568	12,26791788	-3,077322202				
-5,257635191	-7,062480046	1,804844856				
8,267977499	3,816093338	4,451884161				
-2,01991709	-1,711459544	-0,308457546				
-2,472406456	9,16711384	-11,6395203				
3,168408391	0,465790729	2,702617662				
16,48512809	11,34436411	5,140763981				
4,446745972	5,947976399	-1,501230426				
16,17564739	16,82654978	-0,650902392				

F
1,180818695

$$F = \frac{\hat{\sigma}^2}{\sigma^2} = 1.18 < F_T = 1{,}67;$$

Гипотеза о незначимости каждого параметра:

F_values	sign	F_quantile	Hypothesis
1,348536404	<	4,543077165	Принимается
5,251350867	>	4,543077165	Не принимается
0,121964608	<	4,543077165	Принимается
10,52893902	>	4,543077165	Не принимается
0,250128387	<	4,543077165	Принимается
1,380389739	<	4,543077165	Принимается
0,290179481	<	4,543077165	Принимается
0,00079731	<	4,543077165	Принимается
0,374318462	<	4,543077165	Принимается

Гипотеза о незначимости самой регрессии:

F_regression	sign	F_quantile_regression	Hypothesis
3,667903206	>	2,640796883	Не принимается

3. Код программы

```
import numpy as np

import pandas as pd

from scipy.stats import t

from scipy.stats import f

np.random.seed(56)


# Генерация комбинаций факторов

def generate_combinations(x1_levels, x2_levels, x3_levels):

    x1_list = []

    x2_list = []

    x3_list = []


    for i in x1_levels:

        for j in x2_levels:

            for k in x3_levels:

                x1_list.append(i)

                x2_list.append(j)

                x3_list.append(k)


    return map(np.array, [x1_list, x2_list, x3_list])


# Сохранение датафрейма df в файл csv с именем filename

def save_to_csv(df, filename):

    df.to_csv(filename, index=False, encoding='utf-8-sig')


# Получение дисперсии шума

def get_sigma_squared(u):

    # Вычисление мощности сигнала

    omega_squared = np.dot(u - np.mean(u), u - np.mean(u)) / (len(u) - 1)

    # Доля от мощности сигнала

    rho = 0.7

    # Вычисление дисперсии шума

    return rho * omega_squared


# Получение ошибки

def get_noise(u, sigma_squared):

    return np.random.normal(0, np.sqrt(sigma_squared), len(u))


# ------------------ЗАДАНИЕ 1----------------------#
```

```
# Определение параметров
theta = np.array([1, 2, 5, 4, 1.5, 2.5, 0.02, 0.01, 0])

# Определение уровней для каждого фактора
x1_levels = np.array([-1, 0, 1])
x2_levels = np.array([-1, -0.33, 0.33, 1])
x3_levels = np.array([-1, 1])

# Генерация комбинаций факторов
x1, x2, x3 = generate_combinations(x1_levels, x2_levels, x3_levels)

# Вычисление дисперсии шума
sigma_squared = 1.957897413

# Зашумленный отклик
y = [
-7.421921045,
-2.139324492,
-2.090986381,
-1.350575114,
-1.69529944,
-0.586504844,
0.650083657,
5.35306572,
-8.867318951,
-0.362126435,
-3.687216239,
6.217390024,
-1.994746844,
7.613558733,
3.420614445,
9.791004984,
-8.710567529,
4.429908371,
-4.660703112,
4.743967873,
-0.130615528,
12.95403137,
3.408641288,
16.0689259
]
```

```
# Создание матрицы X
X = np.column_stack((np.ones(len(x1)), x1, x2, x3, x1*x2, x1*x3, x1**2, x2**2, x2**3))

# Вычисление вектора параметров модели theta_hat
theta_hat = np.linalg.inv(X.T @ X) @ X.T @ y

y_hat = X @ theta_hat

# Вычисление вектора остатков
e_hat = y - y_hat

# Вычисление несмещенной оценки
sigma_hat_squared = e_hat.T @ e_hat / (len(y) - len(theta))

# Вычисление F-статистики
F=sigma_hat_squared/sigma_squared

# Сохранение всех датафреймов в файлы csv
save_to_csv(pd.concat([
    pd.DataFrame({'y': y}),
    pd.DataFrame({'y_hat': y_hat}),
    pd.DataFrame({'y-y_hat': y-y_hat}),
    pd.DataFrame({'theta': theta}),
    pd.DataFrame({'theta_hat': theta_hat}),
    pd.DataFrame({'sigma_squared': [sigma_squared]}),
    pd.DataFrame({'sigma_hat_squared': [sigma_hat_squared]}),
    pd.DataFrame({'F': [F]})], axis=1), 'Task1.csv')

# ------------------ЗАДАНИЕ 2----------------------#
# Вычисление степеней свободы
f_R = len(y) - len(theta)

# Вычисление квантиля распределения Стьюдента
t_quantile = t.ppf(1 - 0.05 / 2, f_R)

# Вычисление дисперсии оценок параметров
var_theta_hat = np.diag(np.linalg.inv(X.T @ X)) * sigma_hat_squared

# Вычисление стандартного отклонения оценок параметров
std_theta_hat = np.sqrt(var_theta_hat)
```

```
# Вычисление доверительных интервалов
left_interval = theta_hat - t_quantile * std_theta_hat
right_interval = theta_hat + t_quantile * std_theta_hat

save_to_csv(pd.DataFrame({
    'left_interval': left_interval,
    'theta': theta,
    'theta_hat': theta_hat,
    'right_interval': right_interval
}), 'Task2.csv')

# ------------------ЗАДАНИЕ 3----------------------#
# Вычисление квантиля распределения Фишера
F_quantile = f.ppf(1 - 0.05, 1, f_R)

# Вычисление F-статистики для каждого параметра
F_values = (theta_hat ** 2) / (sigma_hat_squared * np.diag(np.linalg.inv(X.T @ X)))

# Проверка гипотезы для каждого параметра
hypothesis_results = F_values < F_quantile

save_to_csv(pd.DataFrame({
    'F_values': F_values,
    'sign': ['<' if result else '>' for result in hypothesis_results],
    'F_quantile': [F_quantile] * len(F_values),
    'Hypothesis': ['Принимается' if result else 'Не принимается' for result in hypothesis_results]
}), 'Task3.csv')

# ------------------ЗАДАНИЕ 4----------------------#
# Вычисление RSS
RSS = e_hat.T @ e_hat

# Вычисление RSS_H
RSS_H = np.sum((y - np.mean(y)) ** 2)

# Вычисление степеней свободы
q = len(theta) - 1

# Вычисление F-статистики
F_regression = ((RSS_H - RSS) / q) / (RSS / (len(y) - len(theta)))
```

```
# Вычисление квантиля распределения Фишера

F_quantile_regression = f.ppf(1 - 0.05, q, len(y) - len(theta))


save_to_csv(pd.DataFrame({

    'F_regression': F_regression,

    'sign': ['<' if F_regression < F_quantile_regression else '>'],

    'F_quantile_regression': F_quantile_regression,

    'Hypothesis': ['Принимается' if F_regression < F_quantile_regression else 'Не принимается']

}), 'Task4.csv')


# ------------------ЗАДАНИЕ 5----------------------#

# Зафиксированные значения факторов

x1_fixed = 0

x3_fixed = 1


# Диапазон значений для x2

x2_range = np.linspace(-1, 1, 24)


f_x = np.column_stack((np.ones(len(x2_range)), x1_fixed * np.ones(len(x2_range)), x2_range, x3_fixed *
np.ones(len(x2_range)),

                       x1_fixed * x2_range, x1_fixed * x3_fixed * np.ones(len(x2_range)), (x1_fixed **
2) * np.ones(len(x2_range)),

                       x2_range ** 2, x2_range ** 3))


# Вычисление прогнозных значений

u_pred = f_x @ theta_hat


save_to_csv(pd.DataFrame({

    'x1': x1_fixed,

    'x2': x2_range,

    'x3': x3_fixed,

    'u_pred': u_pred

}), 'Task5.csv')


# ------------------ЗАДАНИЕ 6----------------------#

u_true = f_x @ theta

y_pred = u_pred


left_interval_pred = u_pred - t_quantile * np.sqrt(sigma_hat_squared * np.diag(f_x @ np.linalg.inv(X.T @
X) @ f_x.T))

right_interval_pred = u_pred + t_quantile * np.sqrt(sigma_hat_squared * np.diag(f_x @ np.linalg.inv(X.T
@ X) @ f_x.T))
```

```
sigma_hat_squared_unbiased = e_hat.T @ e_hat / (len(y) - len(theta))


left_interval_pred2 = y_pred - t_quantile * np.sqrt(sigma_hat_squared_unbiased * (1 + np.diag(f_x @ np.linalg.inv(X.T @ X) @ f_x.T)))

right_interval_pred2 = y_pred + t_quantile * np.sqrt(sigma_hat_squared_unbiased * (1 + np.diag(f_x @ np.linalg.inv(X.T @ X) @ f_x.T)))




save_to_csv(pd.DataFrame({

    'left_interval': left_interval_pred,

    'u_true': u_true,

    'right_interval': right_interval_pred

}), 'Task6_Graphs1.csv')


save_to_csv(pd.DataFrame({

    'left_interval': left_interval_pred2,

    'y_true': u_true,

    'right_interval': right_interval_pred2

}), 'Task6_Graphs2.csv')


# ------------------ЗАДАНИЕ 7----------------------#

# Задание 7-1

# Вычисление истинного отклика без шума

u = theta[0] + theta[1]*x1 + theta[2]*x2 + theta[3]*x3 \

    + theta[4]*x1*x2 + theta[5]*x1*x3 \

    + theta[6]*x1**2 + theta[7]*x2**2 + theta[8]*x2**3


# Вычисление дисперсии шума, учитывая, что мощность случайной помехи = 70% от мощности полезного сигнала

sigma_squared = get_sigma_squared(u)


# Вычисление ошибки

e = get_noise(u, sigma_squared)


# Зашумленный отклик

y = u + e


# Вычисление вектора параметров модели theta_hat

theta_hat = np.linalg.inv(X.T @ X) @ X.T @ y


y_hat = X @ theta_hat


# Вычисление вектора остатков
```

```
e_hat = y - y_hat

# Вычисление несмещенной оценки
sigma_hat_squared = e_hat.T @ e_hat / (len(y) - len(theta))

# Вычисление F-статистики
F=sigma_hat_squared/sigma_squared

# Сохранение всех датафреймов в файлы csv
save_to_csv(pd.concat([
    pd.DataFrame({'y': y}),
    pd.DataFrame({'y_hat': y_hat}),
    pd.DataFrame({'y-y_hat': y-y_hat}),
    pd.DataFrame({'theta': theta}),
    pd.DataFrame({'theta_hat': theta_hat}),
    pd.DataFrame({'sigma_squared': [sigma_squared]}),
    pd.DataFrame({'sigma_hat_squared': [sigma_hat_squared]}),
    pd.DataFrame({'F': [F]})], axis=1), 'Task7_1.csv')

# Задание 7-2
# Вычисление квантиля распределения Фишера
F_quantile = f.ppf(1 - 0.05, 1, f_R)

# Вычисление F-статистики для каждого параметра
F_values = (theta_hat ** 2) / (sigma_hat_squared * np.diag(np.linalg.inv(X.T @ X)))

# Проверка гипотезы для каждого параметра
hypothesis_results = F_values < F_quantile

save_to_csv(pd.DataFrame({
    'F_values': F_values,
    'sign': ['<' if result else '>' for result in hypothesis_results],
    'F_quantile': [F_quantile] * len(F_values),
    'Hypothesis': ['Принимается' if result else 'Не принимается' for result in hypothesis_results]
}), 'Task7_2.csv')

# Задание 7-3
# Вычисление RSS
RSS = e_hat.T @ e_hat

# Вычисление RSS_H
```

```
RSS_H = np.sum((y - np.mean(y)) ** 2)

# Вычисление степеней свободы
q = len(theta) - 1

# Вычисление F-статистики
F_regression = ((RSS_H - RSS) / q) / (RSS / (len(y) - len(theta)))

# Вычисление квантиля распределения Фишера
F_quantile_regression = f.ppf(1 - 0.05, q, len(y) - len(theta))

save_to_csv(pd.DataFrame({
    'F_regression': F_regression,
    'sign': ['<' if F_regression < F_quantile_regression else '>'],
    'F_quantile_regression': F_quantile_regression,
    'Hypothesis': ['Принимается' if F_regression < F_quantile_regression else 'Не принимается']
}), 'Task7_3.csv')
```