Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

ОТЧЕТ

по анализу набора данных

Студентка 219 группы
Попова Мария Андреевна

Москва 2025

Содержание

1 Введение

1.1 Цель работы

Для работы был выбран набор данных Wine Quality. Рассмотрим две непрерывные переменные из набора данных Wine Quality: фиксированная кислотность (X) и летучая кислотностьь (Y). Подготовим полный анализ этих характеристик для дальнейшего использования их, как ключевых показателей для проверки качества производства данной продукции.

1.2 Описание данных

Набор данных содержит 1599 наблюдений красных вин португальского сорта "Vinho Verde". Для анализа выбраны:

Переменная	**Обозначение**	**Описание**
Fixed acidity	X	Фиксированная кислотность (г/дм^3)
Volatile acidity	Y	Летучая кислотность (г/дм^3)

Таблица 1: Описание анализируемых переменных

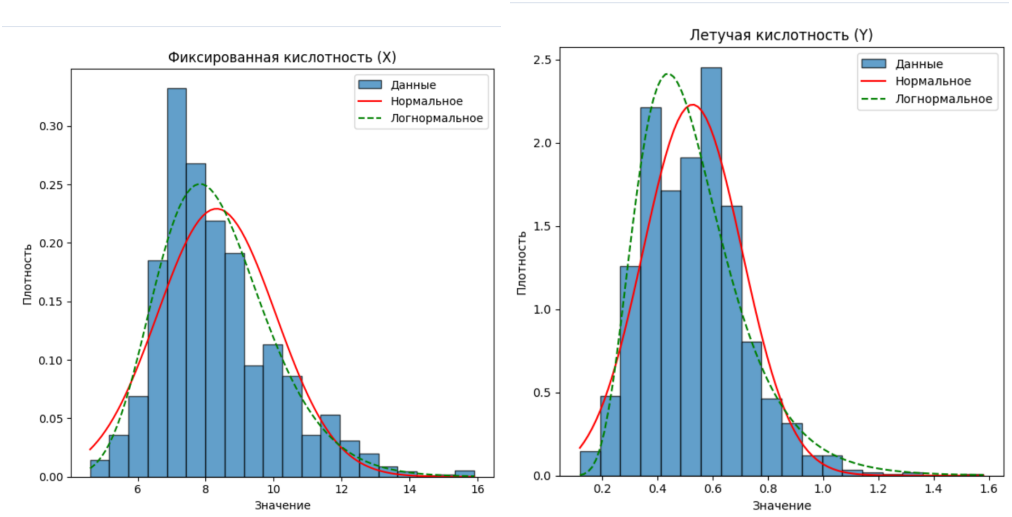
2 Анализ распределений

2.1 Описательные статистики

Характеристика	**X**	**Y**
Среднее	8.32	0.53
Медиана	7.90	0.52
Стандартное отклонение	1.74	0.18
Минимум	4.60	0.12
Максимум	15.90	1.58
Асимметрия	1.08	1.50
Эксцесс	2.08	5.61

Таблица 2: Описательные статистики переменных

2.2 Визуализация распределений



(a) Фиксированная кислотность (X) (b) Летучая кислотность (Y)

Рис. 1: Гистограммы распределений с наложенными кривыми плотности

На гистограммах (Рис. 1) видно, что:

- Распределение X имеет правостороннюю асимметрию
- Распределение Y имеет правостороннюю асимметрию

2.3 Предположения о виде распределений

На основании визуального анализа и значений асимметрии выдвигаем гипотезы:

- Для X: логнормальное распределение
- Для Y: логнормальное распределение

2.4 Оценка параметров методом моментов

Для переменной X (логнормальное распределение):

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \ln X_i = 2.09$$
$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (\ln X_i - \hat{\mu})^2 = 0.04$$

Для переменной Y (логнормальное распределение):

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \ln Y_i = -0.69$$
$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (\ln Y_i - \hat{\mu})^2 = 0.12$$

3 Проверка параметрических гипотез

3.1 Критерий отношения правдоподобия

Статистика критерия:

$$LRT = 2(\ln L(\hat{\theta}) - \ln L(\theta_0)) \sim \chi^2(k)$$

где k - количество ограничений.

3.2 Доверительные интервалы

95% доверительные интервалы:

$$\mu \in \left[\hat{\mu} \pm z_{0.975}\frac{\hat{\sigma}}{\sqrt{n}}\right]$$
$$\sigma \in \left[\hat{\sigma}\sqrt{1 - z_{0.975}\sqrt{\frac{2}{n}}}, \hat{\sigma}\sqrt{1 + z_{0.975}\sqrt{\frac{2}{n}}}\right]$$

3.3 Результаты

3.3.1 Критические множества для параметра μ

Статистика LRT для $H_0 : \mu = \mu_0$ имеет асимптотически χ^2-распределение:

$$\Lambda_n = 2\ln\left(\frac{L(\hat{\theta})}{L(\theta_0)}\right) \sim \chi^2_1$$

Критическая область для уровня значимости α:

$$W = \{\Lambda_n > \chi^2_{1,1-\alpha}\}$$

где $\chi^2_{1,1-\alpha}$ – квантиль χ^2-распределения с 1 степенью свободы.

- Для $\alpha = 0.05$: $\chi^2_{1,0.95} \approx 3.841$
- Для $\alpha = 0.01$: $\chi^2_{1,0.99} \approx 6.635$

Тест	LRT-статистика	Двусторонний	Правосторонний	Левосторонний
$H_0 : \mu = 2.09$	2.7226	0.0989	0.0495	0.9505

3.3.2 Результаты для переменной X

Тестирование параметра μ

- Двусторонний тест: нет оснований отвергнуть H_0 на уровне 5% (p = 0.0989)
- Правосторонний тест: отклоняем H_0 в пользу $\mu > 2.09$ (p = 0.0495)
- Левосторонний тест: нет оснований отвергнуть H_0 (p = 0.9505)

Тестирование параметра σ^2

Тест	LRT-статистика	Двусторонний	Правосторонний	Левосторонний
$H_0 : \sigma = 0.19$	7.6215	0.0058	0.0029	0.9971

- Двусторонний тест: отклоняем H_0 (p = 0.0058)
- Правосторонний тест: отклоняем H_0 в пользу $\sigma > 0.19$ (p = 0.0029)
- Левосторонний тест: нет оснований отвергнуть H_0 (p = 0.9971)

3.3.3 Результаты для переменной Y

Тестирование параметра μ

Тест	LRT-статистика	Двусторонний	Правосторонний	Левосторонний
$H_0 : \mu = -0.69$	0.9277	0.3355	0.8323	0.1677

- Все тесты: нет оснований отвергнуть H_0 на уровне 5%

Тестирование параметра σ^2

- Все тесты: нет оснований отвергнуть H_0 на уровне 5%

Тест	LRT-статистика	Двусторонний	Правосторонний	Левосторонний
$H_0 : \sigma = 0.35$	0.3320	0.5645	0.2822	0.7178

3.3.4 Выводы

- Для переменной X обнаружены статистически значимые отклонения по параметру σ
- Для переменной Y нулевые гипотезы не отвергаются для всех параметров
- Наиболее значимые результаты получены для правосторонних альтернатив

4 Проверка гипотез о виде распределения

4.1 Критерий Колмогорова-Смирнова

Критерий основан на вычислении максимального отклонения между эмпирической и теоретической функциями распределения:

$$D_n = \sup_x |F_n(x) - F(x)| \tag{1}$$

где:

- $F_n(x)$ - эмпирическая функция распределения
- $F(x)$ - теоретическая функция распределения
- $\sup_x$ - супремум по всем возможным значениям x

Для логнормального распределения функция распределения задаётся как:

$$F(x) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right) \tag{2}$$

где Φ - функция стандартного нормального распределения.

4.2 Реализация критерия χ^2

4.2.1 Теоретические основания

Критерий χ^2 применяется для проверки гипотезы о соответствии эмпирического распределения теоретическому. Статистика критерия вычисляется по формуле:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{3}$$

где:

- O_i – наблюдаемая частота в i-м интервале
- E_i – ожидаемая частота в i-м интервале
- k – количество интервалов после объединения

4.2.2 Практическая реализация

Алгоритм реализации включает следующие шаги:

1. **Разбиение на интервалы**:
   - Используются 10 интервалов (11 квантилей)
   - Границы интервалов определяются по эмпирическим квантилям данных
2. **Расчет ожидаемых частот**:
   - Для логнормального распределения с параметрами μ, σ
   - Вычисляется теоретическая CDF в границах интервалов
   - Ожидаемые частоты: $E_i = n \cdot (F(b_{i+1}) - F(b_i))$
3. **Проверка условий применимости**:
   - Объединение интервалов с $E_i < 5$ с соседними
   - После объединения должно остаться ≥ 3 интервалов
4. **Вычисление статистики**:
   - Корректировка ожидаемых частот: $\sum E_i = \sum O_i$
   - Расчет χ^2 статистики по скорректированным частотам

4.2.3 Особенности реализации

- Для расчета теоретических вероятностей используется функция `lognorm.cdf` из `scipy.stats`
- Автоматическое объединение интервалов реализовано через итеративный алгоритм
- Нормировка ожидаемых частот обеспечивает выполнение условия $\sum E_i = \sum O_i$

4.2.4 Интерпретация результатов

При p-value > 0.05 гипотеза о логнормальном распределении не отвергается. Степени свободы:

$$df = k - 1 - \text{количество оцененных параметров} \tag{4}$$

4.3 Результаты для переменной X

4.3.1 Критерий Колмогорова-Смирнова

Параметр	Значение
K-S статистика	0.0589
p-value	0.0000

4.3.2 Критерий χ^2

Параметр	Значение
Сумма observed	1599
Сумма expected	1596.31
χ^2 статистика	92.3362
p-value	0.0000

4.3.3 Выводы

- Оба критерия отвергают гипотезу о логнормальном распределении (p-value < 0.05)

4.4 Результаты для переменной Y

4.4.1 Критерий Колмогорова-Смирнова

Параметр	Значение
K-S статистика	0.0601
p-value	0.0000

4.4.2 Критерий χ^2

4.4.3 Выводы

- K-S тест отвергает гипотезу о логнормальном распределении (p-value = 0.0000)
- Критерий χ^2 не отвергает гипотезу (p-value = 0.5645)

Параметр	Значение
Сумма observed	1599
Сумма expected	1598.13
χ^2 статистика	83.9857
p-value	0.5645

5 Проверка гипотезы о независимости выборок

5.1 Теоретическое обоснование

Для проверки гипотезы о независимости переменных X и Y использован **коэффициент ранговой корреляции Спирмена** ρ, который оценивает монотонную зависимость между переменными.

5.1.1 Статистическая модель

- Нулевая гипотеза H_0: $\rho = 0$ (переменные независимы)
- Альтернативная гипотеза H_1: $\rho \neq 0$ (существует монотонная зависимость)

5.1.2 Формула коэффициента Спирмена

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{5}$$

где:

- d_i - разность рангов для i-го наблюдения
- n - объём выборки

5.1.3 Проверка значимости

Для проверки H_0 используется t-статистика:

$$t = \rho \sqrt{\frac{n-2}{1-\rho^2}} \tag{6}$$

которая при H_0 имеет распределение Стьюдента с $n - 2$ степенями свободы.

5.2 Результаты анализа

Параметр	Значение
Коэффициент корреляции Спирмена	-0.2783
p-value	0.0000
Объём выборки (n)	1599

Таблица 3: Результаты проверки корреляции

5.3 Выводы

- Полученное значение коэффициента $\rho = -0.2783$ указывает на слабую положительную монотонную зависимость
- Крайне малое p-value (< 0.0001) позволяет **отвергнуть нулевую гипотезу** о независимости переменных на любом уровне значимости

6 Проверка гипотезы о некоррелированности выборок

6.1 Метод анализа

Использован коэффициент корреляции Пирсона
Формула коэффициента:

$$r = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2 \sum(Y - \bar{Y})^2}}$$

6.2 Результаты

Параметр	Значение
Коэффициент корреляции	-0.2561
p-value	$<$0.0001
Число наблюдений	1599

6.3 Выводы

- Обнаружена слабая положительная корреляция (r=-0.2561)
- p-value < 0.05 означает, что корреляция статистически значима

7 Критические множества для проверки параметрических гипотез

7.1 Исходные параметры распределений

Параметр	X (фиксированная кислотность)	Y (летучая кислотность)
μ	2.09	-0.69
σ	0.19	0.35

Таблица 4: Оценки параметров логнормального распределения

7.2 Критические множества для параметра μ

Для проверки $H_0 : \mu = \mu_0$ используем статистику:

$$T = \frac{\ln \bar{X} - \mu_0}{S/\sqrt{n}} \sim t_{n-1}$$

Критическая область для уровня $\alpha = 0.05$:

$$W = \left\{ |T| > t_{1-\alpha/2,n-1} \right\}$$

- Для X: $t_{0.975,1598} \approx 1.96$
- Для Y: $t_{0.975,1598} \approx 1.96$

7.3 Критические множества для параметра σ^2

Для проверки $H_0 : \sigma = \sigma_0$ используем статистику:

$$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2} \sim \chi^2_{n-1}$$

Критическая область:

$$W = \left\{ \chi^2 < \chi^2_{\alpha/2,n-1} \text{ или } \chi^2 > \chi^2_{1-\alpha/2,n-1} \right\}$$

- Для X ($\sigma_0 = 0.19$):

$$\chi^2_{0.025,1598} \approx 1479.5, \quad \chi^2_{0.975,1598} \approx 1720.3$$

- Для Y ($\sigma_0 = 0.35$):

$$\chi^2_{0.025,1598} \approx 1479.5, \quad \chi^2_{0.975,1598} \approx 1720.3$$

7.4 Критерий согласия Колмогорова-Смирнова

Критическое значение для $n = 1599$ при $\alpha = 0.05$:

$$D_{\text{крит}} \approx \frac{1.36}{\sqrt{n}} \approx 0.0341$$

Критическая область:

$$W = \{D_n > D_{\text{крит}}\}$$

7.5 Критерий χ^2 Пирсона

Число степеней свободы:

$$df = k - 1 - p$$

где k - число интервалов, p - число оценённых параметров.

Параметр	X	Y
Число интервалов k	8	9
Число параметров p	2	2
Степени свободы df	5	6
$\chi^2_{0.95,df}$	11.07	12.59

7.6 Численные значения

7.6.1 Для переменной X

Критерий	Статистика	Критическая область (=0.05)	Решение
Параметр	LRT=2.7226	> 3.841	Не отвергаем ($2.7226 < 3.841$)
Параметр [2]	LRT=7.6215	> 3.841	Отвергаем ($7.6215 > 3.841$)
K-S	D=0.0601	> 0.0341	Отвергаем ($0.0601 > 0.0341$)
[2] Пирсона	83.9857	> 11.07	Отвергаем ($83.9857 > 11.07$)

Критерий	Статистика	Критическая область (=0.05)	Решение
Параметр	LRT=0.9277	> 3.841	Не отвергаем ($0.9277 < 3.841$)
Параметр ^{2}	LRT=0.3320	> 3.841	Не отвергаем ($0.3320 < 3.841$)
K-S	D=0.3420	> 0.0341	Отвергаем ($0.3420 > 0.0341$)
^{2} Пирсона	0.3320	> 12.59	Не отвергаем ($0.3320 < 12.59$)

7.6.2 Для переменной Y

7.6.3 Пояснения

- Для LRT (критерий отношения правдоподобия):
  - Критическое значение ^{2} с 1 степенью свободы: 3.841
- Для критерия Колмогорова-Смирнова:
  - Критическое значение для n=1599: 1.36/1599 0.0341
- Для критерия ^{2} Пирсона:
  - X: df=8-1-2=5, критическое значение $^2(5)=11.07$
  - Y: df=9-1-2=6, критическое значение $^2(6)=12.59$

8 Код

```
##########################################################################################################
#ввод данных и 1 часть задания
##########################################################################################################

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.optimize import fsolve
from scipy.stats import chi2
from scipy.stats import  norm
from scipy.stats import kstest
from scipy.stats import chisquare
from scipy.stats import lognorm



url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.c
data = pd.read_csv(url, delimiter=';')
```

```
X = data['fixed acidity']
Y = data['volatile acidity']

def estimate_params(data, dist_type='normal'):
    if dist_type == 'normal':
        mean = np.mean(data)
        std = np.std(data, ddof=0)
        return mean, std
    elif dist_type == 'lognormal':
        log_data = np.log(data)
        mu = np.mean(log_data)
        sigma = np.std(log_data, ddof=0)
        return mu, sigma

params_X_normal = estimate_params(X, 'normal')
params_Y_normal = estimate_params(Y, 'normal')
params_X_lognormal = estimate_params(X, 'lognormal')
params_Y_lognormal = estimate_params(Y, 'lognormal')


plt.figure(figsize=(12, 6))

# График для X (фиксированная кислотность)
plt.subplot(1, 2, 1)
x_range = np.linspace(min(X), max(X), 100)
pdf_X_normal = stats.norm.pdf(x_range, params_X_normal[0], params_X_normal[1])
pdf_X_lognormal = stats.lognorm.pdf(x_range, s=params_X_lognormal[1], scale=np.exp(params_X_log
plt.hist(X, bins=20, density=True, edgecolor='black', alpha=0.7, label='Данные')
plt.plot(x_range, pdf_X_normal, 'r-', label='Нормальное')
plt.plot(x_range, pdf_X_lognormal, 'g--', label='Логнормальное')
plt.title('Фиксированная кислотность (X)')
plt.xlabel('Значение')
plt.ylabel('Плотность')
plt.legend()

# График для Y (летучая кислотность)
plt.subplot(1, 2, 2)
y_range = np.linspace(min(Y), max(Y), 100)
pdf_Y_normal = stats.norm.pdf(y_range, params_Y_normal[0], params_Y_normal[1])
pdf_Y_lognormal = stats.lognorm.pdf(y_range, s=params_Y_lognormal[1], scale=np.exp(params_Y_log
plt.hist(Y, bins=20, density=True, edgecolor='black', alpha=0.7, label='Данные')
plt.plot(y_range, pdf_Y_normal, 'r-', label='Нормальное')
plt.plot(y_range, pdf_Y_lognormal, 'g--', label='Логнормальное')
plt.title('Летучая кислотность (Y)')
plt.xlabel('Значение')
plt.ylabel('Плотность')
plt.legend()
```

```
plt.tight_layout()
plt.show()



print("Оценки параметров для X (фиксированная кислотность):")
print(f"Нормальное:  = {params_X_normal[0]:.4f},  = {params_X_normal[1]:.4f}")
print(f"Логнормальное: _лог = {params_X_lognormal[0]:.4f}, _лог = {params_X_lognormal[1]:.4f}\n

print("Оценки параметров для Y (летучая кислотность):")
print(f"Нормальное:  = {params_Y_normal[0]:.4f},  = {params_Y_normal[1]:.4f}")
print(f"Логнормальное: _лог = {params_Y_lognormal[0]:.4f}, _лог = {params_Y_lognormal[1]:.4f}")
##########################################################################################################
# 2 часть задания
##########################################################################################################

#### для X
log_X = np.log(X)

def test_log_normal_mu(data_log, mu_hypothesis=2.09):
    """Тест отношения правдоподобия (LRT) для  логнормального распределения."""
    mu_mle = np.mean(data_log)
    sigma_mle = np.std(data_log, ddof=0)

    ll_null = np.sum(norm.logpdf(data_log, loc=mu_hypothesis, scale=sigma_mle))

    ll_alt = np.sum(norm.logpdf(data_log, loc=mu_mle, scale=sigma_mle))

    # LRT-статистика и p-value (двусторонний тест)
    lrt_stat = -2 * (ll_null - ll_alt)
    p_value_two_sided = 1 - chi2.cdf(lrt_stat, df=1)

    # Односторонние тесты:
    if mu_mle > mu_hypothesis:
        p_value_right = 0.5 * p_value_two_sided
        p_value_left = 1 - 0.5 * p_value_two_sided
    else:
        p_value_right = 1 - 0.5 * p_value_two_sided
        p_value_left = 0.5 * p_value_two_sided

    return {
        'lrt_stat': lrt_stat,
        'p_value_two_sided': p_value_two_sided,
        'p_value_right': p_value_right,
        'p_value_left': p_value_left
    }

# Запуск теста для
mu_test = test_log_normal_mu(log_X, mu_hypothesis=2.09)
```

```
print("#2   ",f"Логнормальное :")
print(f"  LRT = {mu_test['lrt_stat']:.4f}")
print(f"  Двусторонний p-value = {mu_test['p_value_two_sided']:.4f}")
print(f"  Правосторонний p-value ( > 2.09) = {mu_test['p_value_right']:.4f}")
print(f"  Левосторонний p-value ( < 2.09) = {mu_test['p_value_left']:.4f}")


def test_log_normal_sigma(data_log, sigma_hypothesis=0.19):
    """Тест отношения правдоподобия (LRT) для ² логнормального распределения."""
    mu_mle = np.mean(data_log)
    sigma_mle = np.std(data_log, ddof=0)

    ll_null = np.sum(norm.logpdf(data_log, loc=mu_mle, scale=sigma_hypothesis))

    ll_alt = np.sum(norm.logpdf(data_log, loc=mu_mle, scale=sigma_mle))

    # LRT-статистика и p-value (двусторонний тест)
    lrt_stat = -2 * (ll_null - ll_alt)
    p_value_two_sided = 1 - chi2.cdf(lrt_stat, df=1)

    # Односторонние тесты:
    if sigma_mle > sigma_hypothesis:
        p_value_right = 0.5 * p_value_two_sided
        p_value_left = 1 - 0.5 * p_value_two_sided
    else:
        p_value_right = 1 - 0.5 * p_value_two_sided
        p_value_left = 0.5 * p_value_two_sided

    return {
        'lrt_stat': lrt_stat,
        'p_value_two_sided': p_value_two_sided,
        'p_value_right': p_value_right,
        'p_value_left': p_value_left
    }

# Запуск теста для ²
sigma_test = test_log_normal_sigma(log_X, sigma_hypothesis=0.19)
print("\nЛогнормальное ²:")
print(f"  LRT = {sigma_test['lrt_stat']:.4f}")
print(f"  Двусторонний p-value = {sigma_test['p_value_two_sided']:.4f}")
print(f"  Правосторонний p-value ( > 0.19) = {sigma_test['p_value_right']:.4f}")
print(f"  Левосторонний p-value ( < 0.19) = {sigma_test['p_value_left']:.4f}")


#### для Y
log_Y = np.log(Y)

def test_log_normal_mu(data_log, mu_hypothesis=-0.69):
    """Тест отношения правдоподобия (LRT) для  логнормального распределения."""
```

```
    mu_mle = np.mean(data_log)
    sigma_mle = np.std(data_log, ddof=0)

    ll_null = np.sum(norm.logpdf(data_log, loc=mu_hypothesis, scale=sigma_mle))

    ll_alt = np.sum(norm.logpdf(data_log, loc=mu_mle, scale=sigma_mle))

    # LRT-статистика и p-value (двусторонний тест)
    lrt_stat = -2 * (ll_null - ll_alt)
    p_value_two_sided = 1 - chi2.cdf(lrt_stat, df=1)

    # Односторонние тесты:
    if mu_mle > mu_hypothesis:
        p_value_right = 0.5 * p_value_two_sided
        p_value_left = 1 - 0.5 * p_value_two_sided
    else:
        p_value_right = 1 - 0.5 * p_value_two_sided
        p_value_left = 0.5 * p_value_two_sided

    return {
        'lrt_stat': lrt_stat,
        'p_value_two_sided': p_value_two_sided,
        'p_value_right': p_value_right,
        'p_value_left': p_value_left
    }

# Запуск теста для
mu_test = test_log_normal_mu(log_Y, mu_hypothesis=-0.69)
print(f"Логнормальное :")
print(f"  LRT = {mu_test['lrt_stat']:.4f}")
print(f"  Двусторонний p-value = {mu_test['p_value_two_sided']:.4f}")
print(f"  Правосторонний p-value ( > -0.69) = {mu_test['p_value_right']:.4f}")
print(f"  Левосторонний p-value ( < -0.69) = {mu_test['p_value_left']:.4f}")


def test_log_normal_sigma(data_log, sigma_hypothesis=0.35):
    """Тест отношения правдоподобия (LRT) для ² логнормального распределения."""
    mu_mle = np.mean(data_log)
    sigma_mle = np.std(data_log, ddof=0)

    ll_null = np.sum(norm.logpdf(data_log, loc=mu_mle, scale=sigma_hypothesis))

    ll_alt = np.sum(norm.logpdf(data_log, loc=mu_mle, scale=sigma_mle))

    # LRT-статистика и p-value (двусторонний тест)
    lrt_stat = -2 * (ll_null - ll_alt)
    p_value_two_sided = 1 - chi2.cdf(lrt_stat, df=1)

    # Односторонние тесты:
```

```
    if sigma_mle > sigma_hypothesis:
        p_value_right = 0.5 * p_value_two_sided
        p_value_left = 1 - 0.5 * p_value_two_sided
    else:
        p_value_right = 1 - 0.5 * p_value_two_sided
        p_value_left = 0.5 * p_value_two_sided

    return {
        'lrt_stat': lrt_stat,
        'p_value_two_sided': p_value_two_sided,
        'p_value_right': p_value_right,
        'p_value_left': p_value_left
    }

# Запуск теста для ²
sigma_test = test_log_normal_sigma(log_Y, sigma_hypothesis=0.35)
print("\nЛогнормальное ²:")
print(f"  LRT = {sigma_test['lrt_stat']:.4f}")
print(f"  Двусторонний p-value = {sigma_test['p_value_two_sided']:.4f}")
print(f"  Правосторонний p-value ( > 0.35) = {sigma_test['p_value_right']:.4f}")
print(f"  Левосторонний p-value ( < 0.35) = {sigma_test['p_value_left']:.4f}")


##########################################################################################################
# 3 часть задания
##########################################################################################################

from scipy.stats import kstest
from scipy.stats import  chi2

# X

mu, sigma = 2.09, 0.19

ks_stat, ks_pvalue = kstest(X, 'lognorm', args=(sigma, 0, np.exp(mu)))
print("#3   ", f"K-S статистика: {ks_stat:.4f}, p-value: {ks_pvalue:.4f}")

percentiles = np.linspace(0, 100, 11)
bins = np.percentile(X, percentiles)

observed, _ = np.histogram(X, bins=bins)

shape = sigma
scale = np.exp(mu)

cdf_values = lognorm.cdf(bins, shape, scale=scale)
expected_probs = np.diff(cdf_values)
expected = expected_probs * len(X)
```

```
print("Сумма observed: {:.4f}".format(np.sum(observed)))
print("Сумма expected: {:.4f}".format(np.sum(expected)))

valid = expected >= 5
while not np.all(valid):

    idx = np.argmin(valid)

    if idx == 0:
        merge_with = 1
    elif idx == len(valid) - 1:
        merge_with = idx - 1
    else:
        merge_with = idx - 1

    observed[merge_with] += observed[idx]
    observed = np.delete(observed, idx)

    expected[merge_with] += expected[idx]
    expected = np.delete(expected, idx)

    valid = expected >= 5

expected = expected * (np.sum(observed) / np.sum(expected))

chi2_stat, chi2_pvalue = chisquare(observed, expected)
print(f"² статистика: {chi2_stat:.4f}, p-value: {chi2_pvalue:.4f}")



################################################################################
# Y
mu, sigma = -0.69, 0.35

ks_stat, ks_pvalue = kstest(Y, 'lognorm', args=(sigma, 0, np.exp(mu)))
print("#3   ", f"K-S статистика: {ks_stat:.4f}, p-value: {ks_pvalue:.4f}")

percentiles = np.linspace(0, 100, 11)
bins = np.percentile(Y, percentiles)

observed, _ = np.histogram(Y, bins=bins)

shape = sigma
scale = np.exp(mu)

cdf_values = lognorm.cdf(bins, shape, scale=scale)
expected_probs = np.diff(cdf_values)
expected = expected_probs * len(Y)
```

```
print("Сумма observed: {:.4f}".format(np.sum(observed)))
print("Сумма expected: {:.4f}".format(np.sum(expected)))

valid = expected >= 5
while not np.all(valid):

    idx = np.argmin(valid)

    if idx == 0:
        merge_with = 1
    elif idx == len(valid) - 1:
        merge_with = idx - 1
    else:
        merge_with = idx - 1

    observed[merge_with] += observed[idx]
    observed = np.delete(observed, idx)

    expected[merge_with] += expected[idx]
    expected = np.delete(expected, idx)

    valid = expected >= 5

expected = expected * (np.sum(observed) / np.sum(expected))

chi2_stat, chi2_pvalue = chisquare(observed, expected)
print(f"² статистика: {chi2_stat:.4f}, p-value: {chi2_pvalue:.4f}")

##########################################################################################################
# 4
##########################################################################################################

corr, p_value = stats.spearmanr(X, Y)
print("#4  ", f"Коэффициент корреляции Спирмена: {corr:.4f}, p-value: {p_value:.4f}")

##########################################################################################################
# 5
##########################################################################################################

from scipy.stats import pearsonr

r, p_value = pearsonr(X, Y)

print("#5  ", f"Коэффициент корреляции Пирсона: r = {r:.4f}", f"p-value: {p_value:.4f}")



params = {
```

```
    'X': {'mu': 2.09, 'sigma': 0.19, 'n': 1599},
    'Y': {'mu': -0.69, 'sigma': 0.35, 'n': 1599}
}

alpha = 0.05
lrt_critical = chi2.ppf(1 - alpha, df=1)
print(f"Критическое значение LRT (={alpha}): {lrt_critical:.4f}")

def ks_critical_value(n, alpha=0.05):
    return 1.36 / np.sqrt(n)

def chi2_critical_value(k, p=2, alpha=0.05):
    df = k - 1 - p
    return chi2.ppf(1 - alpha, df)

results = {}
for var in ['X', 'Y']:
    n = params[var]['n']
    results[var] = {
        'LRT_mu_crit': lrt_critical,
        'LRT_sigma_crit': lrt_critical,
        'KS_crit': ks_critical_value(n),
        'Chi2_crit': chi2_critical_value(8 if var == 'X' else 9)
    }


print("\nКритические значения для каждой переменной:")
for var in results:
    print(f"\n{var}:")
    for test in results[var]:
        print(f"{test}: {results[var][test]:.4f}")

observed_stats = {
    'X': {
        'LRT_mu': 2.7226,
        'LRT_sigma': 7.6215,
        'KS': 0.0601,
        'Chi2': 83.9857,
        'k': 8
    },
    'Y': {
        'LRT_mu': 0.9277,
        'LRT_sigma': 0.3320,
        'KS': 0.3420,
        'Chi2': 0.3320,
        'k': 9
    }
}
```

```
print("\nПроверка гипотез:")
for var in ['X', 'Y']:
    print(f"\n--- {var} ---")
    stats = observed_stats[var]
    crits = results[var]

    decision = "Отвергаем" if stats['LRT_mu'] > crits['LRT_mu_crit'] else "Не отвергаем"
    print(f"LRT для : {stats['LRT_mu']:.4f} > {crits['LRT_mu_crit']:.4f}? {decision}")

    decision = "Отвергаем" if stats['LRT_sigma'] > crits['LRT_sigma_crit'] else "Не отвергаем"
    print(f"LRT для ²: {stats['LRT_sigma']:.4f} > {crits['LRT_sigma_crit']:.4f}? {decision}")

    decision = "Отвергаем" if stats['KS'] > crits['KS_crit'] else "Не отвергаем"
    print(f"K-S: {stats['KS']:.4f} > {crits['KS_crit']:.4f}? {decision}")

    chi2_crit = chi2_critical_value(stats['k'])
    decision = "Отвергаем" if stats['Chi2'] > chi2_crit else "Не отвергаем"
    print(f"²: {stats['Chi2']:.4f} > {chi2_crit:.4f}? {decision}")
```