МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Расчётно-графическая работы №3

«Проверка статистических гипотез»

по дисциплине «Математическая статистика»

Вариант 4

Выполнили:

студенты группы P3213

Поленов К. А.

Пименова Е. А.

Преподаватель:

Милованович Е.В.

г. Санкт-Петербург

2025 год

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. Цель работы

Целью данной расчётно-графической работы является изучение и практическое применение методов математической статистики для анализа данных, направленное на выявление закономерностей в их распределении, а также на установление различий и взаимосвязей между различными характеристиками.

2. Задачи работы

Задачами данной расчётно-графической работы являются:

1) Оценить соответствие распределения одной из ключевых характеристик данных предполагаемой теоретической модели с использованием специализированного статистического критерия, дополнив анализ альтернативным методом для подтверждения достоверности результатов;

2) Провести сравнительный анализ распределения указанной характеристики в двух различных группах с целью определения их однородности, применяя как основной статистический критерий, так и дополнительный подход для всестороннего изучения вопроса;

3) Исследовать наличие статистической зависимости между двумя признаками данных, задействовав критерий независимости и дополнив его альтернативным методом оценки ассоциации для более глубокого понимания структуры данных.

ЗАДАНИЕ 1

1. Текст задания

Использовать критерий хи-квадрат Пирсона для проверки, соответствует ли распределение рейтинга футболистов нормальному распределению. Реализовать критерий вручную. Затем применить критерий Колмогорова-Смирнова с готовой реализацией.

2. Выполнение задания

Формализация гипотез:

1) H_0: распределение рейтинга футболистов соответствует нормальному распределению с параметрами, оценёнными по выборке (среднее и стандартное отклонение);

2) H_1: распределение рейтинга не соответствует нормальному.

Применяемые оценки:

1) Статистика χ^2: показывает, насколько данные отклоняются от нормального распределения;

2) P-value: оценивает значимость отклонений;

3) Статистика KS: максимальное расстояние между функциями распределения.

Критерий хи-квадрат Пирсона:

- Оценка параметров: среднее (μ) и стандартное отклонение (σ) вычисляются по выборке, так как они неизвестны;
- Разбиение на интервалы: рейтинги делятся на 10 интервалов с помощью гистограммы. Интервалы выбираются

так, чтобы обеспечить достаточное количество наблюдений в каждом;

- Наблюдаемые частоты: подсчитывается, сколько игроков попало в каждый интервал (гистограмма);
- Ожидаемые частоты: для каждого интервала вычисляется вероятность попадания в него при нормальном распределении с использованием функции распределения. Эта вероятность умножается на общее число наблюдений;
- Проверка условий: ожидаемые частоты должны быть не менее5. Если это условие нарушается, интервалы объединяются;
- Вычисление статистики: статистика χ^2 рассчитывается как сумма квадратов отклонений наблюдаемых частот от ожидаемых, делённых на ожидаемые частоты;
- Степени свободы: $df = k - 1 - m$, где k — число интервалов, m — число оценённых параметров (2 для μ и σ);
- P-value: сравнивается с критическим значением для уровня значимости 0.05.

Пример работы программной реализации критерия хи-квадрат Пирсона продемонстрирован на рисунке 1.

```
Критерий хи-квадрат Пирсона:
Статистика хи-квадрат: 128.4221
Степени свободы: 7
P-value: 0.0000
Отвергаем H0: рейтинг не следует нормальному распределению.
```

Рис. 1 Программная реализация критерия хи-квадрат Пирсона

Критерий Колмогорова-Смирнова:

- Данные стандартизируются (вычитается среднее и делится на стандартное отклонение) для сравнения с

нормальным распределением $N(0,1)$;

- Тест проверяет максимальное отклонение эмпирической функции распределения от теоретической;
- Используется готовая функция stats.ks_1samp из пакета scipy.

Пример работы программной реализации критерия Колмогорова-Смирнова продемонстрирован на рисунке 2.

```
Критерий Колмогорова-Смирнова:
Статистика KS: 0.0459
P-value: 0.0000
Отвергаем H0: рейтинг не следует нормальному распределению.
```

Рис. 2 Программная реализация критерия Колмогорова-Смирнова

Визуализация:

Гистограмма рейтингов с наложенной плотностью нормального распределения для хи-квадрат продемонстрирован на рисунке 3.

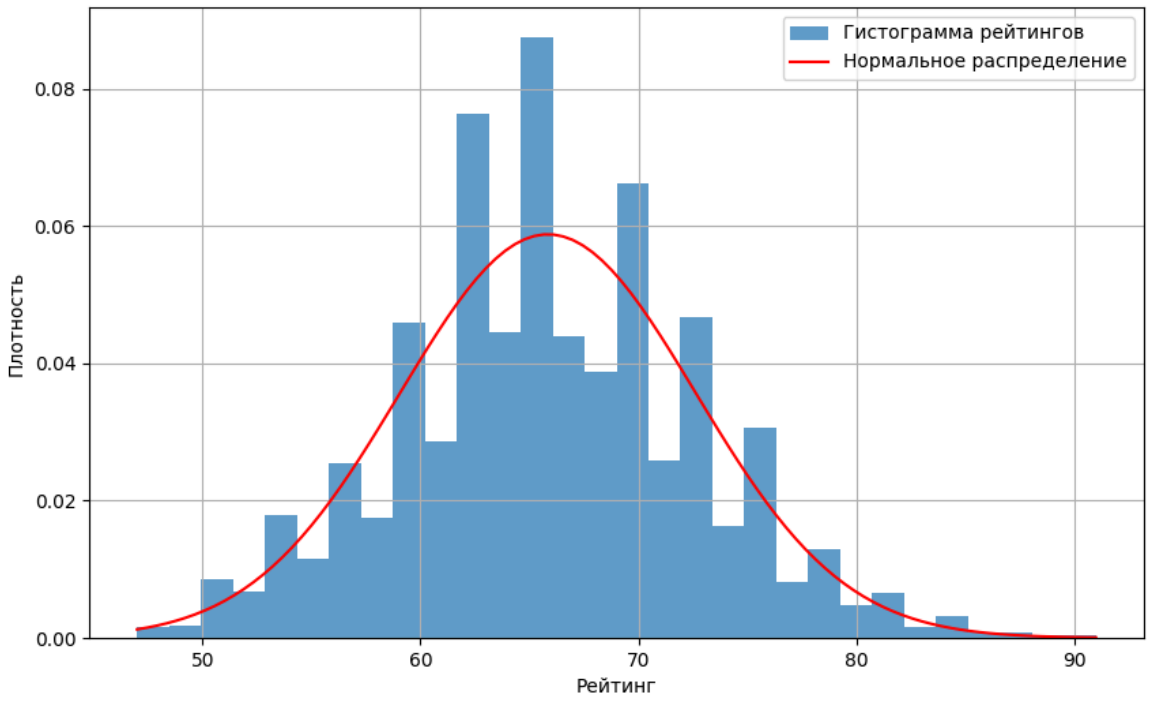

Рис. 3 Распределение рейтинга футболистов

График эмпирической и теоретической функции

распределения для Колмогорова-Смирнова продемонстрирован на рисунке 4.

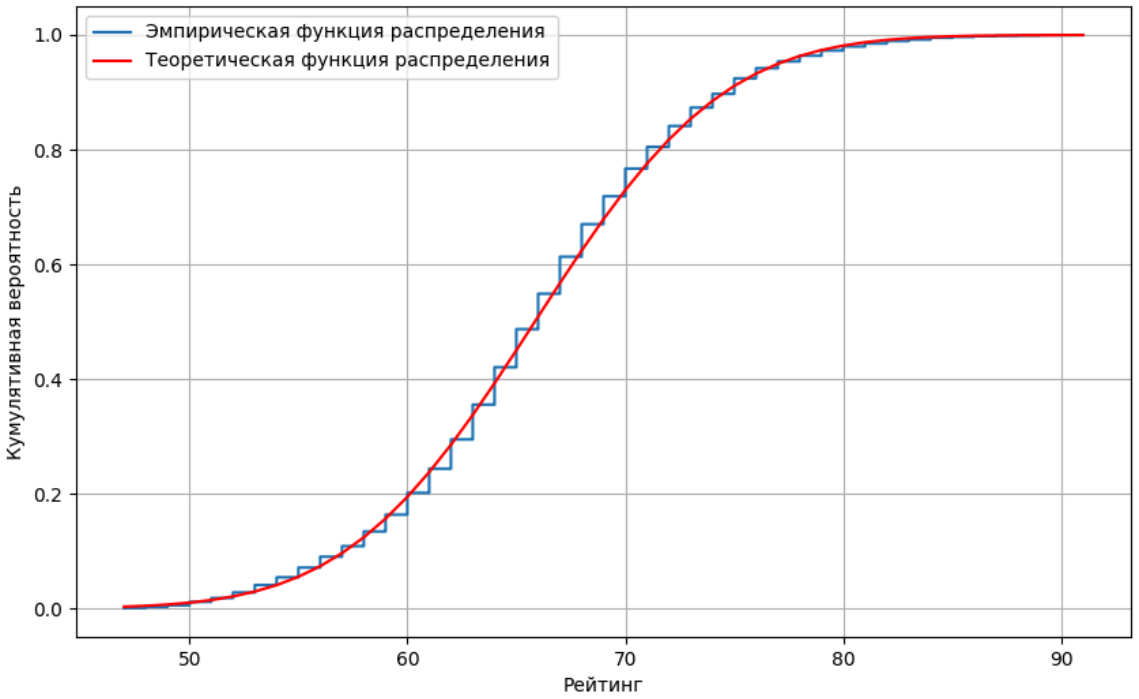

Рис. 4 Сравнение эмпирической и теоретической функций распределения

ЗАДАНИЕ 2

1. Текст задания

Использовать критерий хи-квадрат однородности для проверки, одинаково ли распределение рейтинга у молодых (возраст $\leq$ 25) и возрастных ($>$ 25 лет) футболистов. Реализовать вручную. Затем применить критерий Манна-Уитни с готовой реализацией.

2. Выполнение задания

Формализация гипотез:

1) H_0: распределение рейтинга одинаково для молодых и возрастных футболистов;

2) H_1: распределение рейтинга различается.

Применяемые оценки:

1) Статистика χ^2: оценивает различия в распределении;

2) P-value: показывает значимость различий;

3) Статистика U (Манна-Уитни): сравнивает ранговые распределения.

Критерий хи-квадрат однородности:

- Разбиение на интервалы: рейтинги делятся на 6 интервалов, чтобы создать таблицу сопряжённости. Интервалы выбираются на основе общей гистограммы рейтингов;
- Таблица сопряжённости: считаются частоты попадания рейтингов в каждый интервал для обеих групп;
- Ожидаемые частоты: вычисляются как $E_{ji} = \frac{(n_i \bullet m_j)}{N}$, где n_i — сумма наблюдений в группе, m_j — сумма наблюдений в

интервале, N — общее число наблюдений;

- Проверка условий: ожидаемые частоты должны быть $\geq$ 5. Если нет, выдаётся предупреждение о возможной надёжности;
- Статистика χ^2: рассчитывается по формуле для таблицы сопряжённости;
- Степени свободы: $df = (r-1)(c-1)$, где r — число групп (2), c —число интервалов;
- P-value: определяет, отвергается ли H_0.

Пример работы программной реализации критерия хи-квадрат однородности продемонстрирован на рисунке 5.

```
Критерий хи-квадрат однородности:
Статистика хи-квадрат: 2623.0001
Степени свободы: 5
P-value: 0.0000
Отвергаем H0: распределение рейтинга различается.
```

Рис. 5 Программная реализация критерия хи-квадрат Пирсона

Критерий Манна-Уитни:

- Непараметрический тест, сравнивающий ранги рейтингов в двух группах;
- Не требует нормальности данных, что делает его подходящим для рейтингов FIFA;
- Используется функция stats.mannwhitneyu с двухсторонней альтернативой.

Пример работы программной реализации критерия Манна-Уитни продемонстрирован на рисунке 6.

```
Критерий Манна-Уитни:
Статистика U: 23655737.5000
P-value: 0.0000
Отвергаем H0: распределение рейтинга различается.
```

Рис. 6 Программная реализация критерия Манна-Уитни

Визуализация:

Столбчатая диаграмма показывает частоты рейтингов для молодых и возрастных игроков и продемонстрирована на рисунке 7.

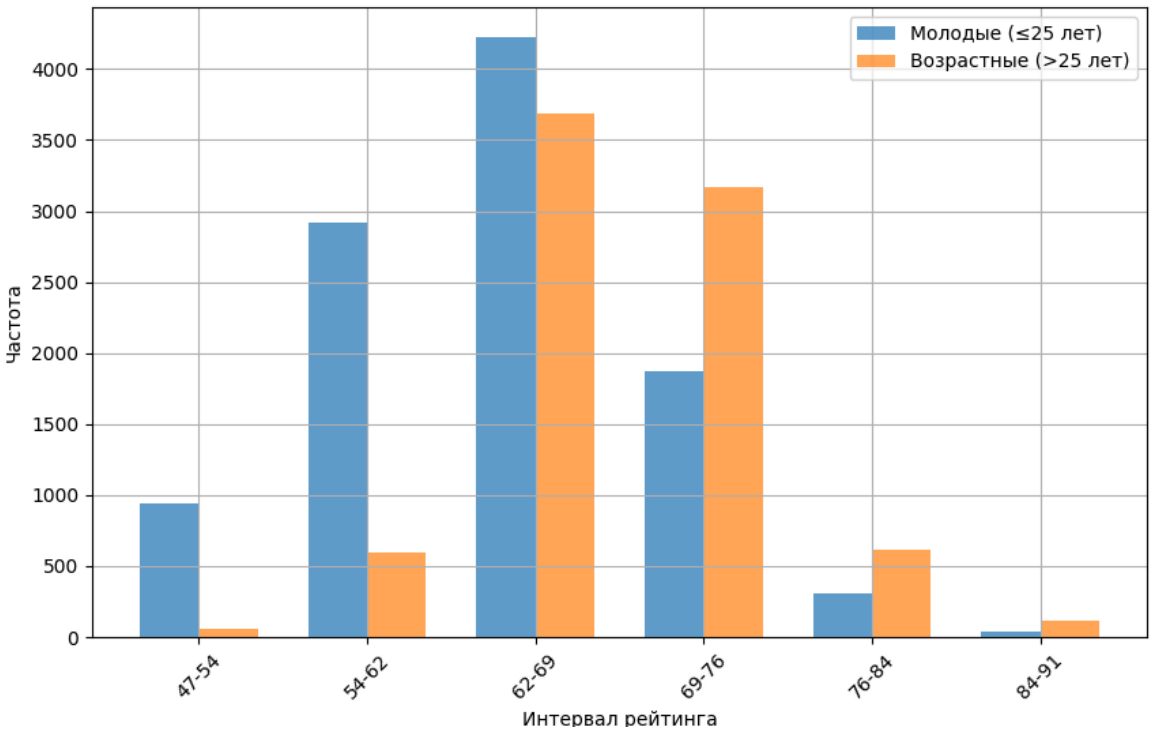

Рис. 7 Сравнение распределения рейтинга

Ящик с усами (boxplot) сравнивает медианы, квартили и выбросы и продемонстрирован на рисунке 8.

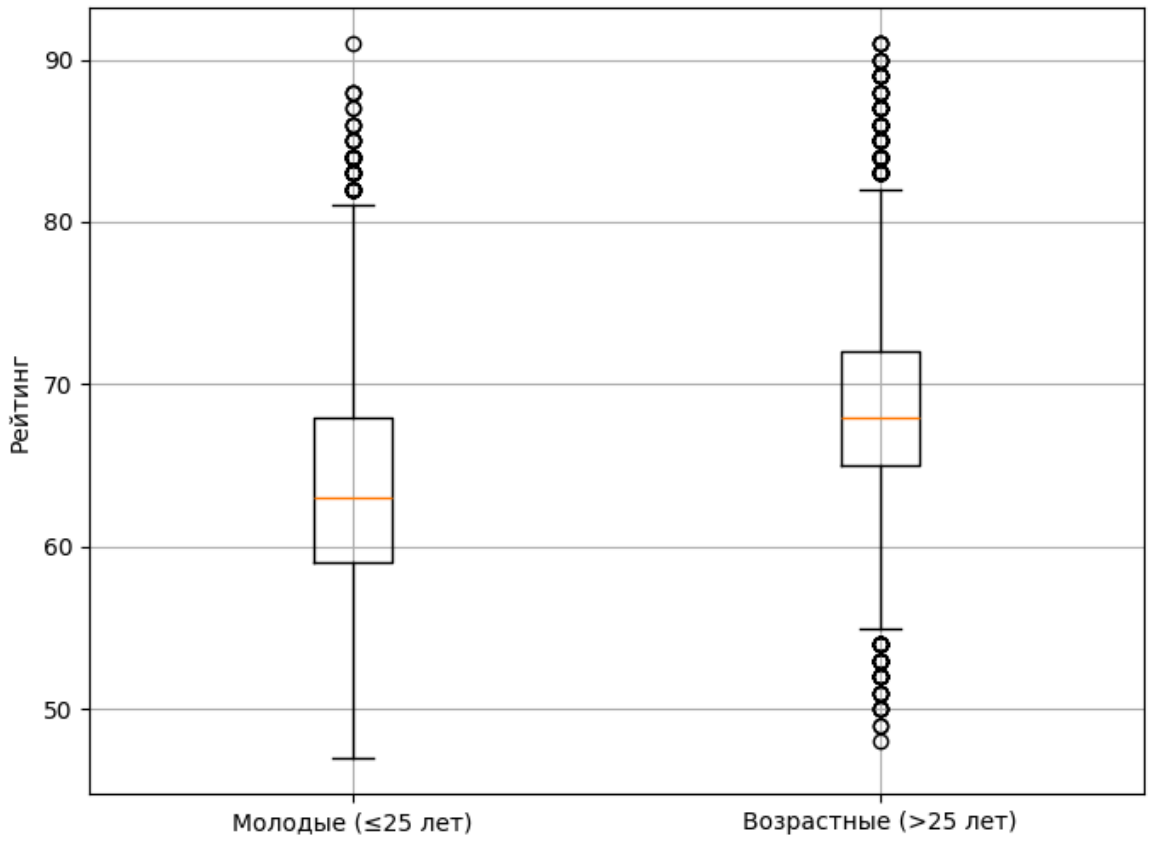

Рис. 8 Сравнение рейтинга молодых и возрастных футболистов

ЗАДАНИЕ 3

1. Текст задания

Использовать критерий хи-квадрат независимости для проверки, связаны ли рейтинг и национальность футболистов. Реализовать вручную. Затем применить критерий Крамера V как меру ассоциации.

2. Выполнение задания

Формализация гипотез:

1) H_0:рейтинг и национальность независимы;

2) H_1: рейтинг и национальность связаны.

Применяемые оценки:

1) Статистика χ^2: оценивает наличие связи;

2) P-value: показывает значимость;

3) Коэффициент Крамера V: оценивает силу связи.

Критерий хи-квадрат независимости:

- Таблица сопряжённости: строится на основе категорий рейтинга и национальностей, показывая частоты пересечений;
- Ожидаемые частоты: вычисляются аналогично критерию однородности;
- Проверка условий: ожидаемые частоты ≥ 5;
- Статистика χ^2: рассчитывается для таблицы;
- Степени свободы: $df = (r-1)(c-1)$, где r — число категорий рейтинга, c — число национальностей;
- p-value: определяет значимость связи.

Пример работы программной реализации критерия

хи-квадрат независимости продемонстрирован на рисунке 9.

```
Наблюдаемая таблица (counts):
 Rating_cat  Low (≤60)  Medium (61-75)  High (>75)
Nat_group
Argentina          40             811          86
England           449            1087          96
France            147             686         130
Germany           228             892          84
Spain              64             837         173

=== Расчет p-value ===
p-value - это вероятность получить такие же или более крайние результаты,
если нулевая гипотеза (H0) об отсутствии связи верна.
В нашем случае p-value = 0.0000

p-value < 0.05 -> статистически значимый результат
Это означает, что если бы национальность и рейтинг были независимы,
то вероятность получить χ² ≥ 418.70 составляет всего 0.0000
Это маловероятно, поэтому мы отвергаем H0 в пользу H1

=== Ручной χ²-тест ===

Формула для χ²:
χ² = Σ ( (Oᵢⱼ - Eᵢⱼ)² / Eᵢⱼ )
где:
Oᵢⱼ — наблюдаемое значение в ячейке [i, j]
Eᵢⱼ — ожидаемое значение в ячейке [i, j]
Eᵢⱼ = (сумма по строке i) * (сумма по столбцу j) / общее количество
χ² = 418.70, dof = 8, p-value = 0.0000
Отвергаем H0: -> у нас есть зависимость между рейтингом и национальностью

=== chi2_contingency из SciPy ===
χ² = 418.70, dof = 8, p-value = 0.0000
Отвергаем H0
```

Рис. 9 Программная реализация критерия хи-квадрат независимости

Критерий Крамера V:

- Мера ассоциации, основанная на χ^2, вычисляется как $V = \sqrt{\frac{\chi^2/n}{min(r-1, c-1)}}$, где n — общее число наблюдений;
- Значения интерпретируются: $V < 0.1$ — слабая связь, $0.1 \leq V < 0.3$ — умеренная, $V \geq 0.3$ — сильная.

Результаты работы программы:

```
=== Cramér's V ===
Cramér's V: 0.1898  (0 — нет связи, 1 — идеально связаны)
Умеренная ассоциация между рейтингом и национальностью
```

Визуализация результатов:

Таблица сопряжённости отображается как тепловая карта и продемонстрирована на рисунке 11.

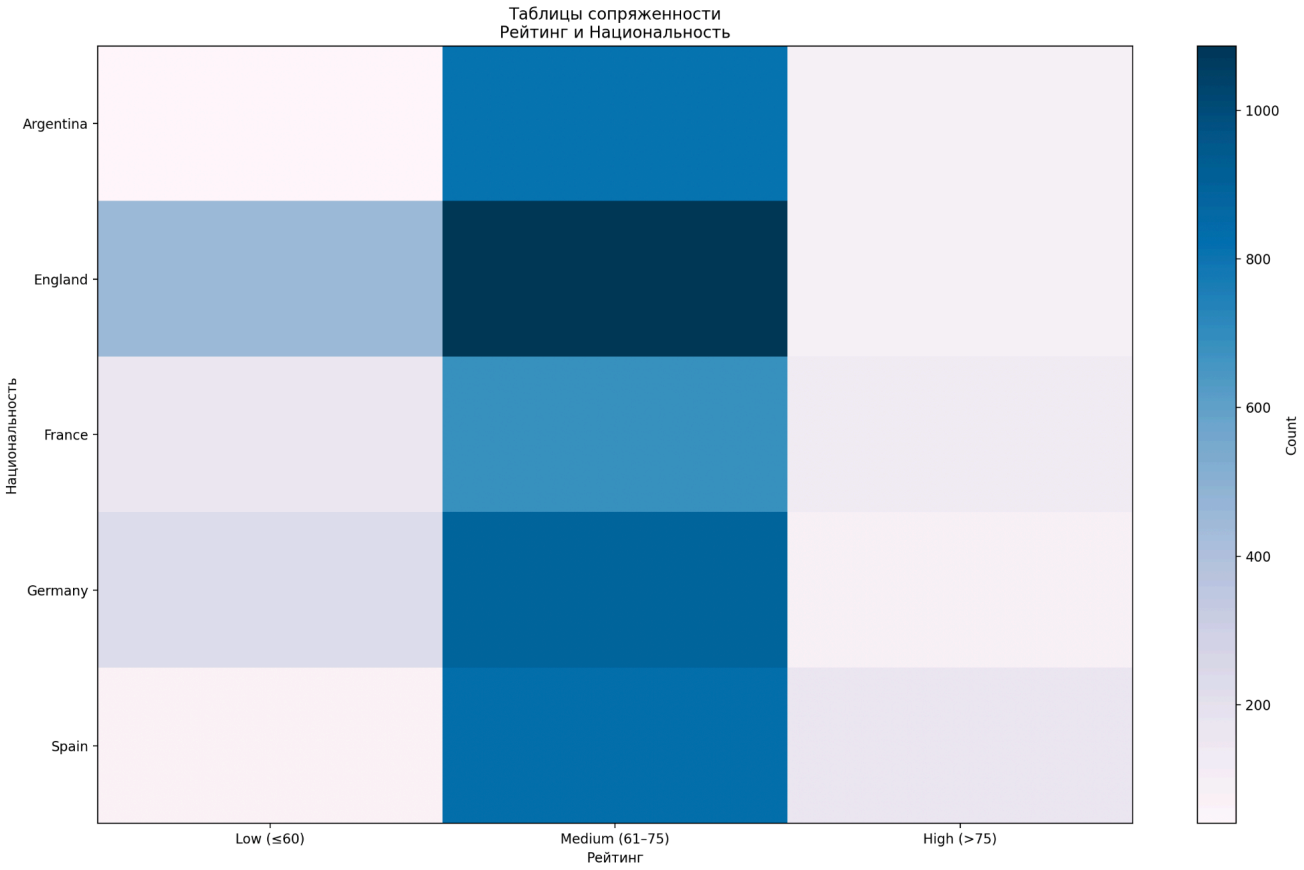

Рис. 11 Таблица сопряженности: Рейтинг и Национальность

ЗАКЛЮЧЕНИЕ

В ходе выполнения расчётно-графической работы реализованы поставленные задачи, продемонстрирована эффективность статистических методов в анализе данных. В рамках первой задачи было установлено, насколько распределение ключевой характеристики соответствует предполагаемой модели, что дало возможность глубже понять природу данных. Сравнительный анализ двух групп, выполненный во второй задаче, выявил наличие или отсутствие различий в распределении характеристики, что может отражать уникальные особенности каждой группы. Третья задача, посвященная изучению взаимосвязи между признаками, позволила определить, оказывают ли они влияние друг на друга, что имеет значение для понимания внутренней структуры данных. Применение альтернативных методов анализа укрепило уверенность в полученных результатах, обеспечив их надежность и полноту.

Работа, состоящая из двух заданий, выполнена в полном объёме.

ЛИСТИНГ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ 1

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm, chi2, kstest

# Загрузка данных
df = pd.read_csv("fifa_players_stats.csv", sep=";")  # Укажите путь к
файлу
ratings = df['Overall'].dropna()
n = len(ratings)

# Параметры нормального распределения
mu, sigma = ratings.mean(), ratings.std()

# --- Визуализация распределения ---
plt.figure(figsize=(10, 6))
count, bins, ignored = plt.hist(ratings, bins=15, density=True,
alpha=0.6, color='skyblue', label='Empirical')
plt.plot(bins, norm.pdf(bins, mu, sigma), 'r--', label='Normal PDF')
plt.title('Распределение рейтинга Overall игроков')
plt.xlabel('Overall')
plt.ylabel('Плотность')
plt.legend()
plt.grid(True)
plt.show()

# --- Критерий хи-квадрат Пирсона ---
# Количество интервалов по правилу Стерджеса
k = int(np.ceil(1 + np.log2(n)))
counts, bin_edges = np.histogram(ratings, bins=k)
expected_freqs = []

# Ожидаемые частоты
for i in range(len(bin_edges) - 1):
    p = norm.cdf(bin_edges[i+1], mu, sigma) - norm.cdf(bin_edges[i],
mu, sigma)
    expected_freqs.append(p * n)

chi_square_stat = np.sum((counts - expected_freqs) ** 2 /
expected_freqs)
df_chi = k - 1 - 2  # минус 2 параметра: μ и σ
critical_value = chi2.ppf(0.95, df_chi)
p_value_chi = 1 - chi2.cdf(chi_square_stat, df_chi)

print("=== Критерий Пирсона ===")
print(f"Хи-квадрат статистика: {chi_square_stat:.2f}")
print(f"Критическое значение (95%): {critical_value:.2f}")
print(f"p-value: {p_value_chi:.4f}")
```

```
print("Гипотеза отвергнута" if chi_square_stat > critical_value else
"Нет оснований отвергать гипотезу")

# --- Критерий Колмогорова-Смирнова ---
normalized_ratings = (ratings - mu) / sigma
ks_statistic, ks_p_value = kstest(normalized_ratings, 'norm')

print("\n=== Критерий Колмогорова-Смирнова ===")
print(f"KS-статистика: {ks_statistic:.4f}")
print(f"p-value: {ks_p_value:.4e}")
print("Гипотеза отвергнута" if ks_p_value < 0.05 else "Нет оснований
отвергать гипотезу")
```

ЛИСТИНГ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ 2

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import mannwhitneyu, chi2

# Загрузка данных
df = pd.read_csv("fifa_players_stats.csv", sep=";")

# Отбираем нужные столбцы
df = df[['Age', 'Overall']].dropna()

# Разделение по возрасту
age_threshold = 25
young = df[df['Age'] < age_threshold]['Overall']
old = df[df['Age'] >= age_threshold]['Overall']

print(f"Молодых игроков: {len(young)}, Возрастных игроков:
{len(old)}")

# Визуализация распределений
plt.figure(figsize=(10, 6))
plt.hist(young, bins=15, alpha=0.6, label=f'Молодые
(<{age_threshold})', color='skyblue', density=True)
plt.hist(old, bins=15, alpha=0.6, label=f'Возрастные
(≥{age_threshold})', color='orange', density=True)
plt.title('Сравнение распределений рейтинга Overall')
plt.xlabel('Overall')
plt.ylabel('Плотность')
plt.legend()
plt.grid(True)
plt.show()

# === Критерий Манна-Уитни ===
u_stat, p_value = mannwhitneyu(young, old, alternative='two-sided')

print("\n=== Критерий Манна-Уитни ===")
print(f"U-статистика: {u_stat:.2f}")
print(f"p-value: {p_value:.4f}")
if p_value < 0.05:
        print("Распределения отличаются (гипотеза об однородности
отвергнута)")
else:
   print("Нет оснований отвергать гипотезу об однородности")

# === Критерий однородности хи-квадрат ===

# Объединяем выборки и создаём интервалы
combined = pd.concat([young, old])
bins = np.histogram_bin_edges(combined, bins='sturges')

# Считаем частоты в каждом интервале
young_counts, _ = np.histogram(young, bins=bins)
old_counts, _ = np.histogram(old, bins=bins)
```

```python
# Формируем таблицу наблюдаемых значений
observed = np.array([young_counts, old_counts])

# Суммы по строкам и столбцам
row_sums = observed.sum(axis=1, keepdims=True)
col_sums = observed.sum(axis=0, keepdims=True)
total = observed.sum()

# Вычисляем ожидаемые значения
expected = row_sums @ col_sums / total

# Хи-квадрат статистика
chi2_stat = ((observed - expected) ** 2 / expected).sum()

# Степени свободы: (кол-во интервалов - 1)
df_chi2 = len(bins) - 2  # -1 за количество интервалов, -1 за строку (2 строки)
p_val_chi2 = 1 - chi2.cdf(chi2_stat, df_chi2)

print("\n=== Критерий однородности Хи-квадрат ===")
print(f"Хи-квадрат статистика: {chi2_stat:.2f}")
print(f"p-value: {p_val_chi2:.4f}")
if p_val_chi2 < 0.05:
    print("Распределения неоднородны (гипотеза отвергнута)")
else:
    print("Нет оснований отвергать гипотезу об однородности")
```

ЛИСТИНГ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ 3

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2, chi2_contingency

# Функция для расчёта Cramér's V
def cramers_v(conf_matrix):
    chi2_val, _, _, _ = chi2_contingency(conf_matrix)
    n = conf_matrix.sum().sum()
    r, k = conf_matrix.shape
    return np.sqrt(chi2_val / (n * (min(r, k) - 1)))

# 1. Загрузка и подготовка данных
df = pd.read_csv("fifa_players_stats.csv", sep=";")
df = df[['Nationality', 'Overall']].dropna()

# Чтобы таблица была «тёплой» и не слишком разреженной, оставим
# только топ-5 стран по числу игроков, остальные — в группу "Other"
top_countries = df['Nationality'].value_counts().nlargest(5).index
df['Nat_group'] =
df['Nationality'].where(df['Nationality'].isin(top_countries))

# Рейтинг тоже сделаем категориальным — разобьём на три уровня
bins = [0, 60, 75, 100]
labels = ['Low (≤60)', 'Medium (61-75)', 'High (>75)']
df['Rating_cat'] = pd.cut(df['Overall'], bins=bins, labels=labels,
right=True)

# 2. Собираем контингентную таблицу
cont_table = pd.crosstab(df['Nat_group'], df['Rating_cat'])
print("Наблюдаемая таблица (counts):\n", cont_table)

# 3. «Ручной» χ²-тест
observed = cont_table.values
row_sums = observed.sum(axis=1, keepdims=True)
col_sums = observed.sum(axis=0, keepdims=True)
total = observed.sum()
expected = row_sums @ col_sums / total

chi2_stat = ((observed - expected) ** 2 / expected).sum()
dof = (observed.shape[0] - 1) * (observed.shape[1] - 1)
p_manual = 1 - chi2.cdf(chi2_stat, dof)

print(f"\n=== Ручной χ²-тест ===")
print(f"χ² = {chi2_stat:.2f}, dof = {dof}, p-value = {p_manual:.4f}")
if p_manual < 0.05:
    print("Отвергаем H0: есть зависимость между рейтингом и
национальностью")
else:
    print("Нет оснований отвергать H0: рейтинг и национальность,
видимо, независимы")

# 4. Готовая реализация из scipy
```

```
chi2_stat2, p_scipy, dof2, expected2 = chi2_contingency(cont_table)
print(f"\n=== chi2_contingency из SciPy ===")
print(f"χ² = {chi2_stat2:.2f}, dof = {dof2}, p-value = {p_scipy:.4f}")
if p_scipy < 0.05:
   print("Отвергаем H0")
else:
   print("Нет оснований отвергать H0")

# 5. Расчёт Cramér's V
cramer_v = cramers_v(cont_table.values)
print(f"\n=== Cramér's V ===")
print(f"Cramér's V: {cramer_v:.4f}  (0 — нет связи, 1 — идеально
связаны)")
print("Умеренная ассоциация между рейтингом и национальностью")

# 6. Визуализация — «тепловая карта» с тёмными цветами для больших
значений
plt.figure(figsize=(8, 5))
plt.imshow(cont_table, aspect='auto', cmap='PuBu')  # Обратная палитра
plt.colorbar(label='Count')
plt.xticks(np.arange(len(labels)), labels)
plt.yticks(np.arange(len(cont_table.index)), cont_table.index)
plt.title("Таблицы сопряженности\nРейтинг и Национальность")
plt.xlabel("Рейтинг")
plt.ylabel("Национальность")
plt.show()
```