**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

Курсовая РАБОТА

**по дисциплине «Автоматизация схемотехнического проектирования»**

**Тема**: **Анализ экспериментальных данных**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1302 |  | Новиков Г.В. |
| Преподаватель |  | Боброва Ю.О. |

Санкт-Петербург

2025

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студент Новиков Г.В. | | |
| Группа 1302 | | |
| Тема работы: Анализ экспериментальных данных | | |
| Исходные данные:  Датасет, содержащий информацию о мобильных телефонах. | | |
| Содержание пояснительной записки:  «Содержание», «Введение», «График распределения параметров», «Медиана», «Графики распределения при медианном значении», «Среднее значение и стандартное отклонение», «Различия в классах», «Листинг», «Заключение», «Список использованных источников» | | |
| Предполагаемый объем пояснительной записки:  Не менее 15 страниц. | | |
| Дата выдачи задания: 03.02.2025 | | |
| Дата сдачи реферата: 24.03.2025 | | |
| Дата защиты реферата: 24.03.2025 | | |
| Студент гр. 1302 | |  | Новиков Г.В. | |
| Преподаватель | |  | Боброва Ю.О. | |

**Аннотация**

**Summary**

СОДЕРЖАНИЕ

[Введение 5](#_Toc192290886)

[1. График рапредения параметров 8](#_Toc192290887)

[2. Медиана 9](#_Toc192290888)

[2.1. Медиана параметра n 9](#_Toc192290889)

[2.2. Медиана параметра m 9](#_Toc192290890)

[3. Графики распределения при медианном значении 10](#_Toc192290891)

[3.1. График распределения параметров 10](#_Toc192290892)

[3.2. Гистограмма распределения, скатерограмма и боксплот параметров 10](#_Toc192290893)

[4. Среднее значение и стандартное отклонение 11](#_Toc192290894)

[5. Различия в классах 12](#_Toc192290895)

[5.1. Различия в классах по параметру n 12](#_Toc192290896)

[5.2. Различия в классах по параметру m 12](#_Toc192290897)

[Листинг 13](#_Toc192290898)

[Заключение 14](#_Toc192290899)

[Список использованных источников 15](#_Toc192290900)

Введение

Датасет содержит 1512 строк с характеристиками мобильных телефонов:

1. phone\_name — Название модели телефона.
2. brand — Бренд или производитель телефона.
3. os — Операционная система, установленная на телефоне.
4. inches — Диагональ экрана телефона в дюймах.
5. resolution — Разрешение экрана в формате "ширина x высота" (например, "1080x1920").
6. battery — Емкость аккумулятора телефона в миллиампер-часах (мАч).
7. battery\_type — Тип аккумулятора (например, "Li-Ion" или "Li-Po").
8. ram(GB) — Объем оперативной памяти (RAM) в гигабайтах (ГБ).
9. announcement\_date — Дата анонса телефона в формате "год-месяц-день".
10. weight(g) — Вес телефона в граммах (г).
11. storage(GB) — Объем встроенной памяти телефона в гигабайтах (ГБ).
12. video\_720p — Поддержка записи видео с разрешением 720p (HD).
13. video\_1080p — Поддержка записи видео с разрешением 1080p (Full HD).
14. video\_4K — Поддержка записи видео с разрешением 4K (Ultra HD).
15. video\_8K — Поддержка записи видео с разрешением 8K.
16. video\_30fps — Поддержка записи видео с частотой 30 кадров в секунду (fps).
17. video\_60fps — Поддержка записи видео с частотой 60 кадров в секунду (fps).
18. video\_120fps — Поддержка записи видео с частотой 120 кадров в секунду (fps).
19. video\_240fps — Поддержка записи видео с частотой 240 кадров в секунду (fps).
20. video\_480fps — Поддержка записи видео с частотой 480 кадров в секунду (fps).
21. video\_960fps — Поддержка записи видео с частотой 960 кадров в секунду (fps).
22. price(USD) — Цена телефона в долларах США (USD)

Рассматриваемый параметр – price(USD).

Задачи:

1. Построить график распределения параметров. Разделить выборку по классам. Подписать оси, добавить легенду и сетку на график.

2. Рассчитать медиану параметра n: для выборки в целом, для каждого класса отдельно

3. Рассчитать медиану параметра m: для выборки в целом, для каждого класса отдельно

4. Построить график распределения параметров для объектов выше медианного значения в выборке и ниже. Подписать оси, добавить легенду и сетку на график.

5. Построить гистограмму распределения, скатерограмму и боксплот параметров для объектов выше медианного значения в выборке и ниже. Подписать оси, добавить легенду и сетку на график.

6. Рассчитать среднее значение и стандартное отклонение для параметров: для всей выборки и для каждого класса отдельно.

7. Статистически оценить различия в классах по параметру n- совпадают ли выборки по этому параметру или нет.

8. Статистически оценить различия в классах по параметру m - совпадают ли выборки по этому параметру или нет.

1. Графики рапредения параметров

Задание: построить график распределения параметров. Разделить выборку по классам. Подписать оси, добавить легенду и сетку на график.

Для начала построим гистограмму для рассматриваемого параметра.

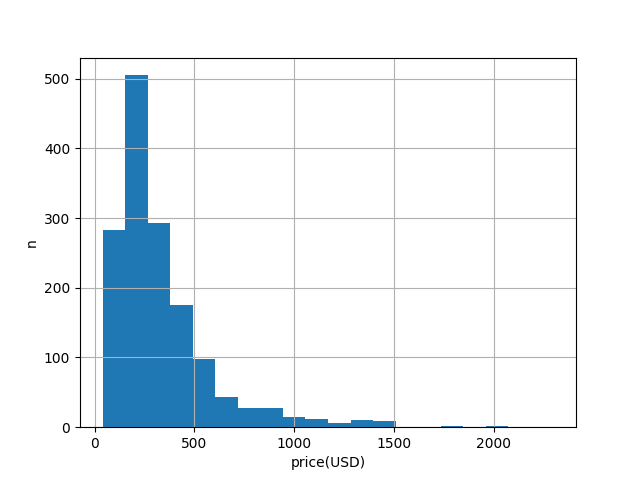


Рис. 1. Рассматриваемый параметр (price (USD))

Разделим выборку на 2 класса:

Класс 1: объекты со значением price(USD) больше или равным медиане.

Класс 2: объекты со значением price(USD) меньше медианы.

Медиана price(USD) равна 260.

Построим гистограммы для оставшихся параметров с разделением на классы. Для параметров brand, os, resolution, announcement\_date построены столбчатые диаграммы с 10 наиболее часто встречающимися значениями. Для параметра phone\_name график не был построен, так как для большинства строк значение уникально.

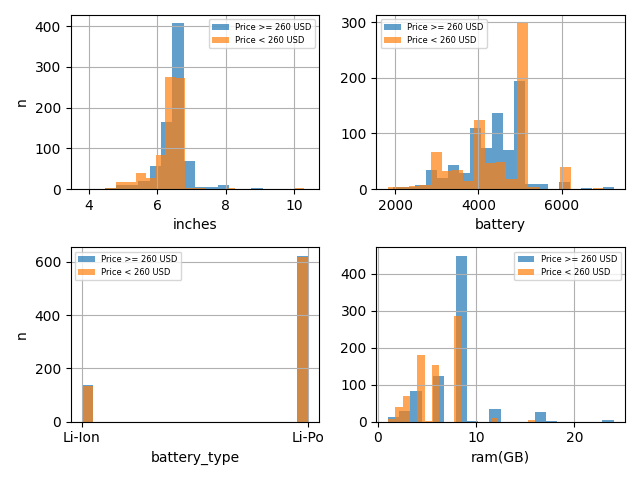


Рис. 2. Параметры

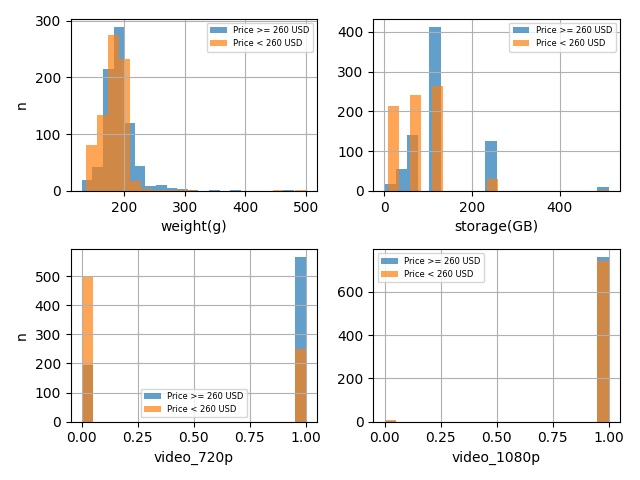


Рис. 3. Параметры

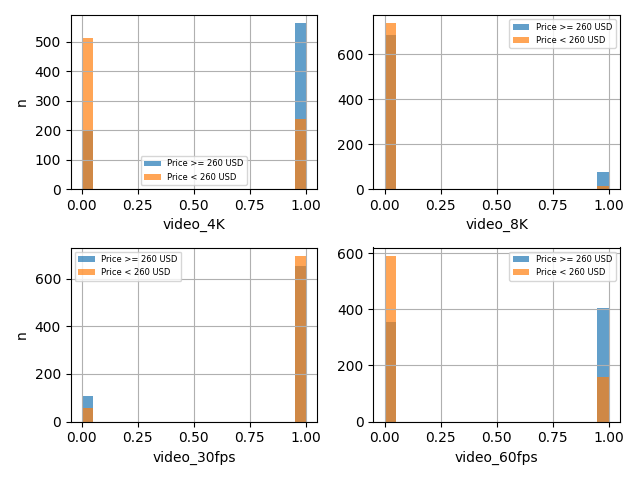


Рис. 4. Параметры

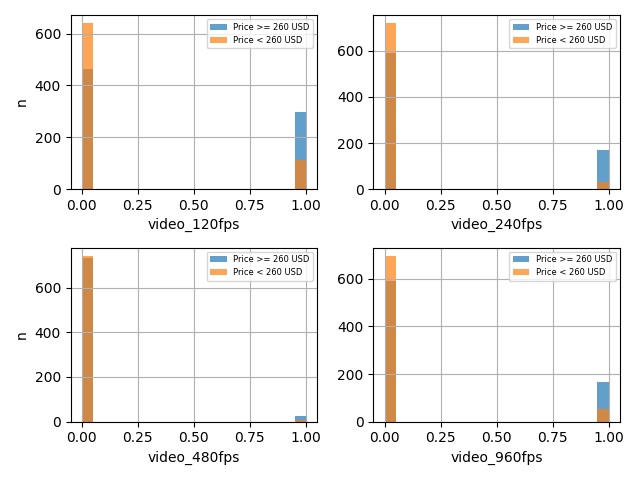


Рис. 5. Параметры

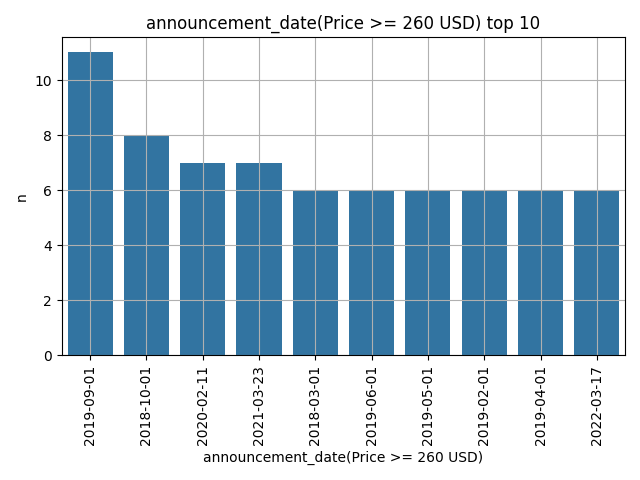


Рис. 6. Параметр

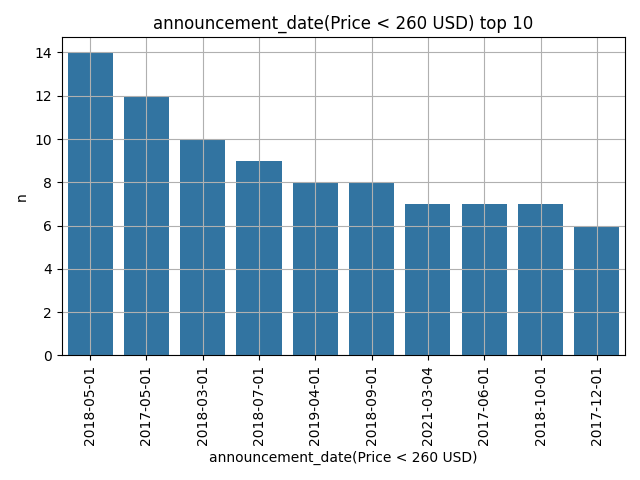


Рис. 7. Параметр

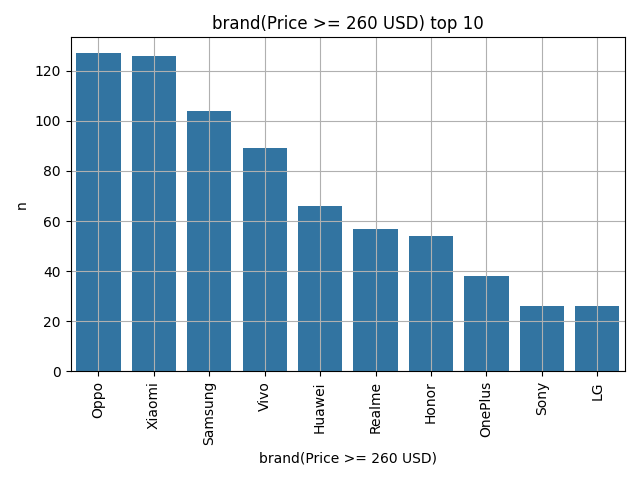


Рис. 8. Параметр

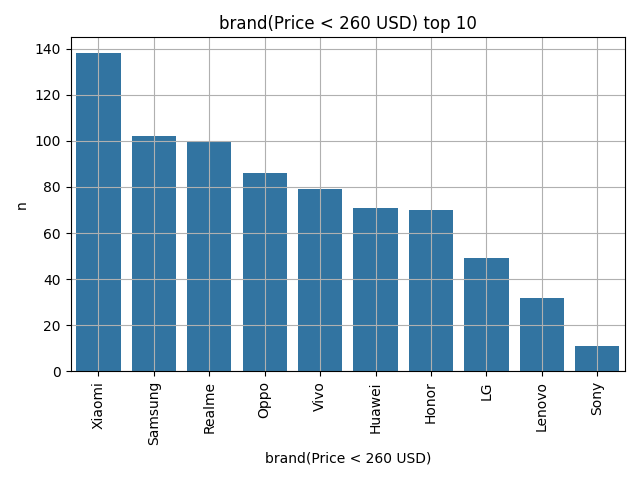


Рис. 9. Параметр

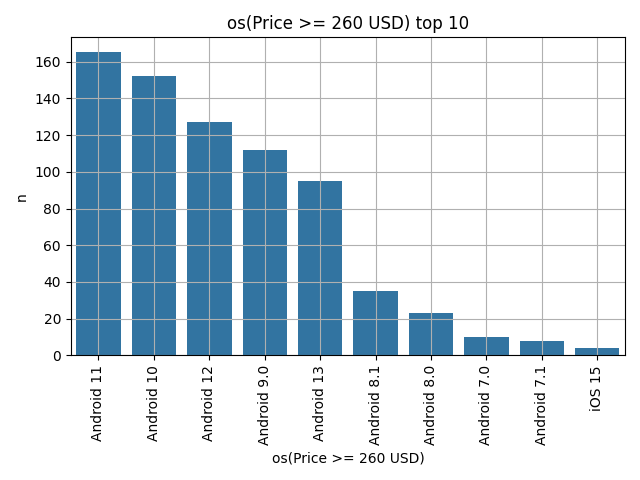


Рис. 10. Параметр

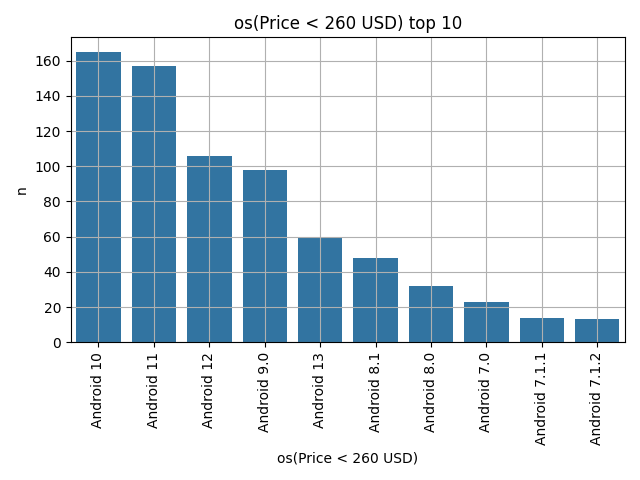


Рис. 11. Параметр

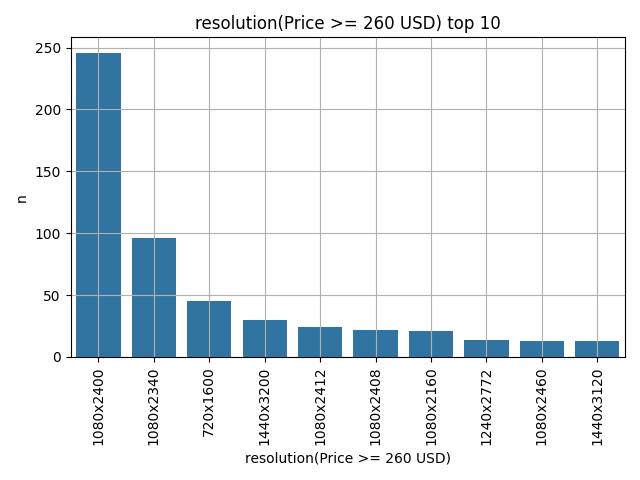


Рис. 12. Параметр

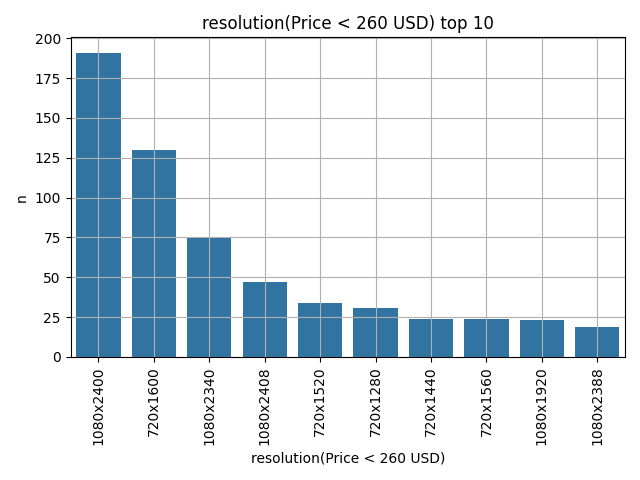


Рис. 13. Параметры

2. Медиана

2.1. Медиана параметра n

Задание: рассчитать медиану параметра n: для выборки в целом, для каждого класса отдельно.

Медиана была рассчитана для параметра inches.



Рис. 14. Медиана параметра inches.

2.2. Медиана параметра m

Задание: рассчитать медиану параметра m: для выборки в целом, для каждого класса отдельно.

Медиана была рассчитана для параметра ram(GB).



Рис. 15. Медиана параметра ram(GB).

3. Графики распределения при медианном значении

3.1. График распределения параметров

Задание: построить график распределения параметров для объектов выше медианного значения в выборке и ниже. Подписать оси, добавить легенду и сетку на график.

Данные разделены относительно медианы параметра ram(GB).

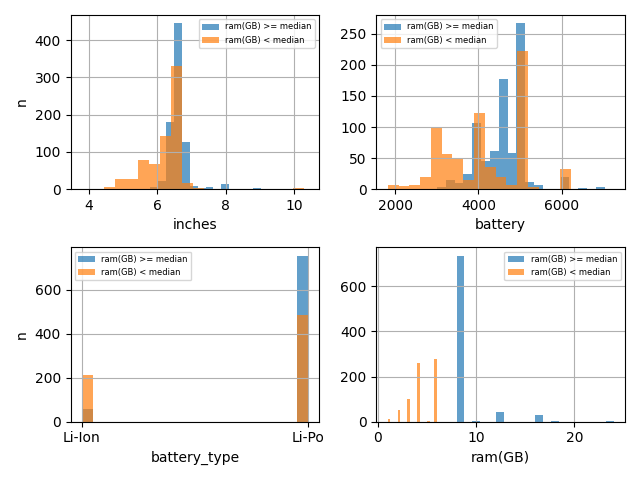


Рис. 16. Параметры

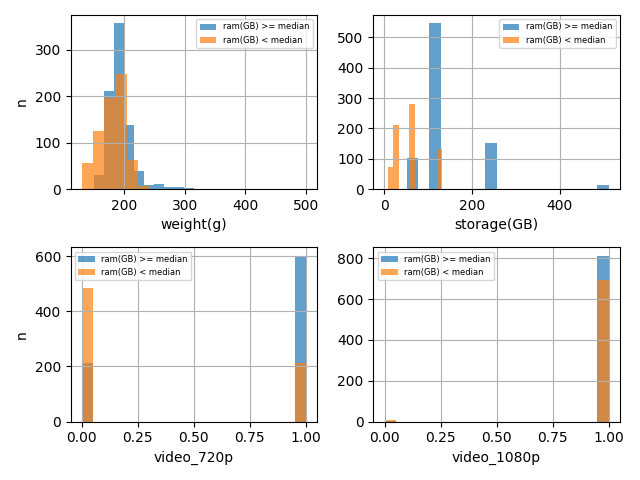


Рис. 17. Параметры

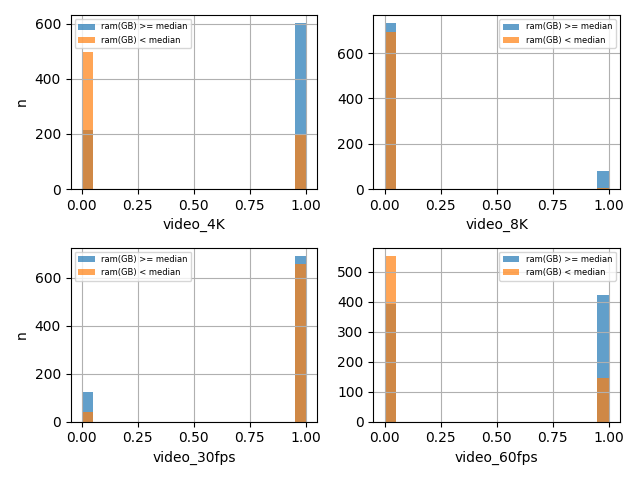


Рис. 18. Параметры

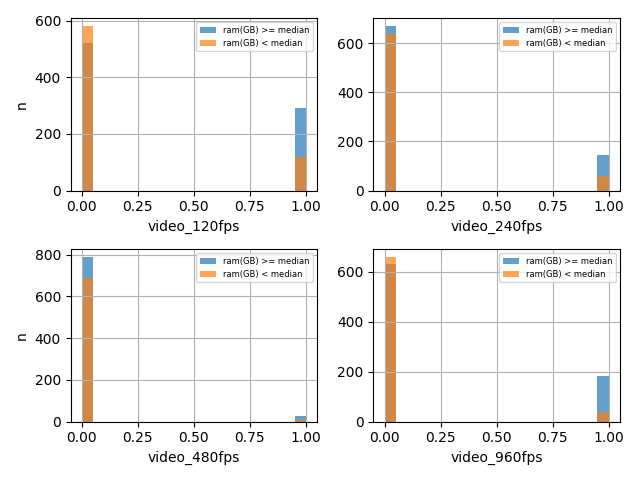


Рис. 19. Параметры

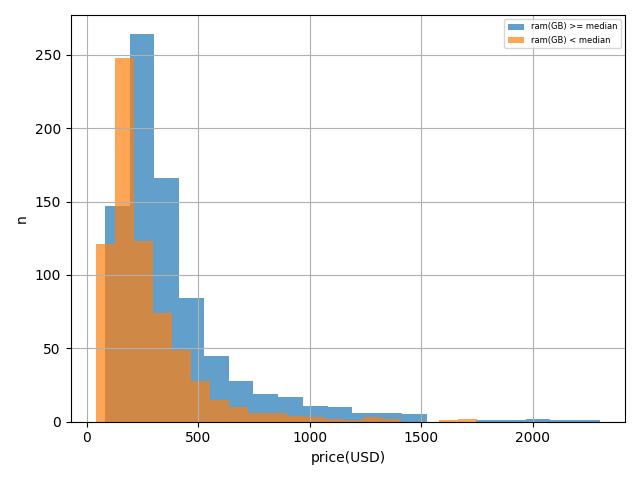


Рис. 20. Параметр price(USD)

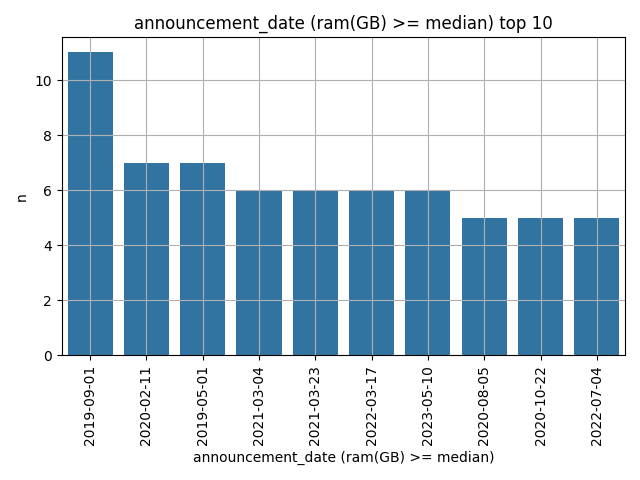


Рис. 21. Параметр

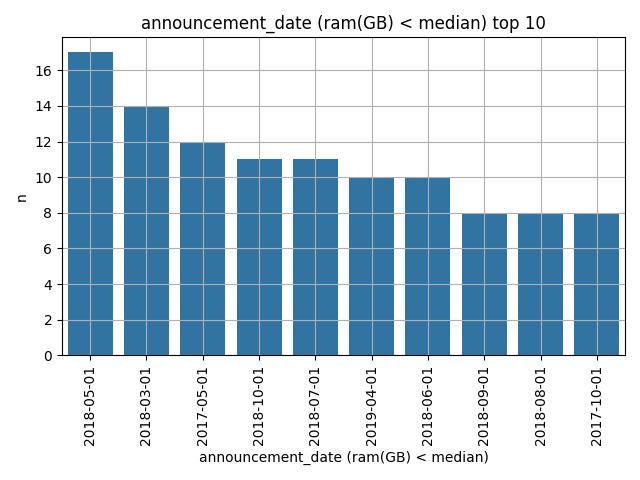


Рис. 22. Параметр

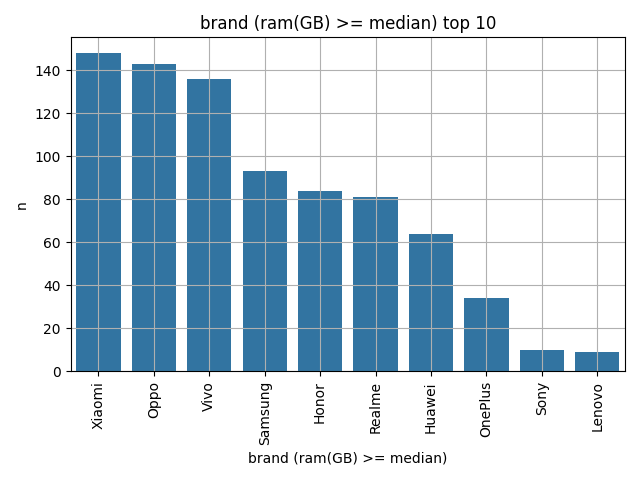


Рис. 23. Параметр

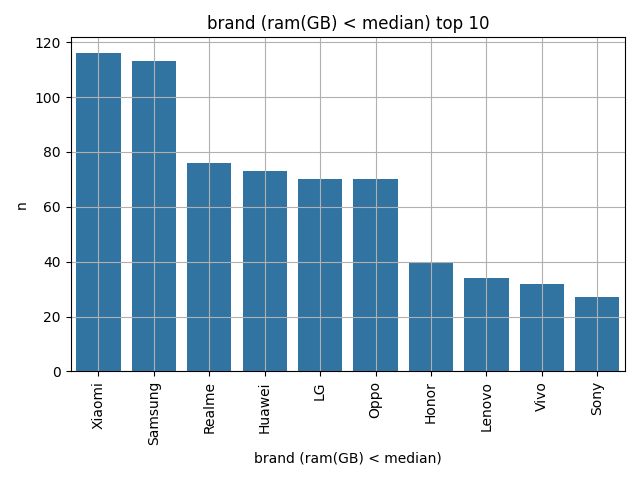


Рис. 24. Параметр

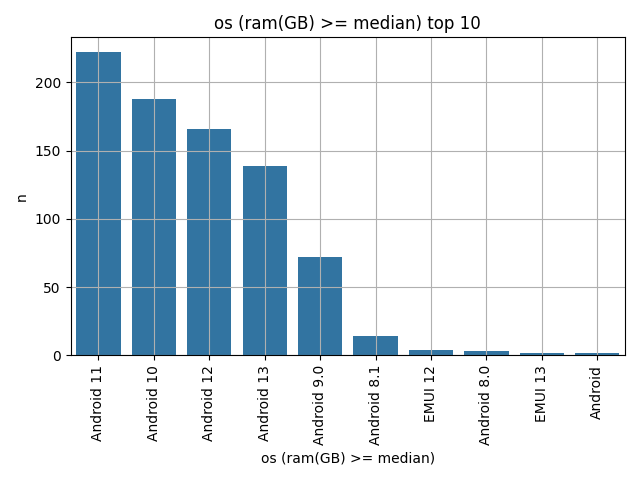


Рис. 25. Параметр

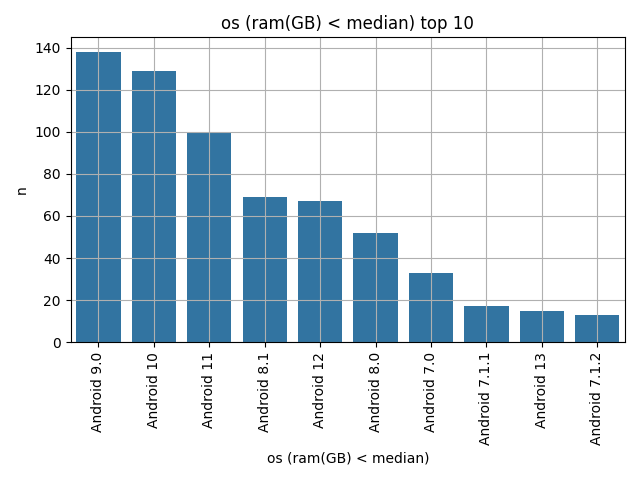


Рис. 26. Параметр

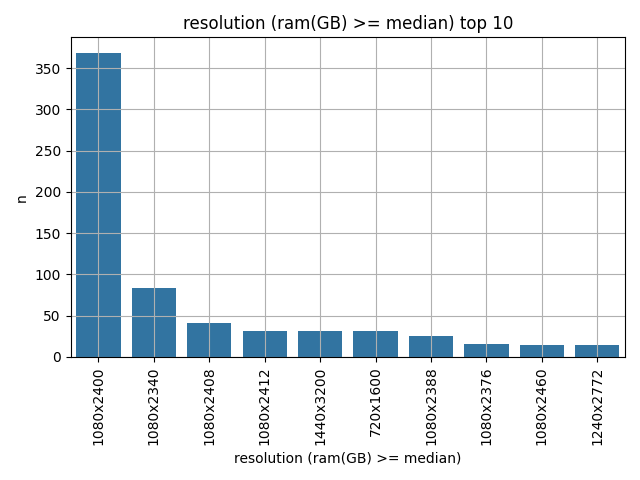


Рис. 27. Параметр

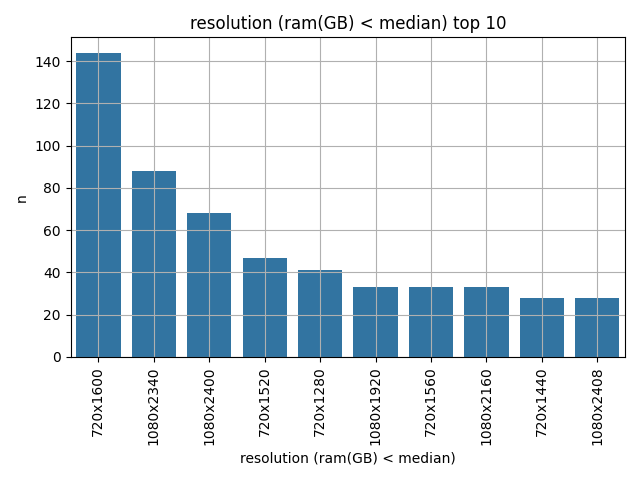


Рис. 28. Параметр

3.2. Гистограмма распределения, скатерограмма и боксплот параметров

Задание: построить гистограмму распределения, скатерограмму и боксплот параметров для объектов выше медианного значения в выборке и ниже. Подписать оси, добавить легенду и сетку на график.

Скатерограмма и боксплоты были построены по каждому столбцу относительно медианы столбца ram(GB).

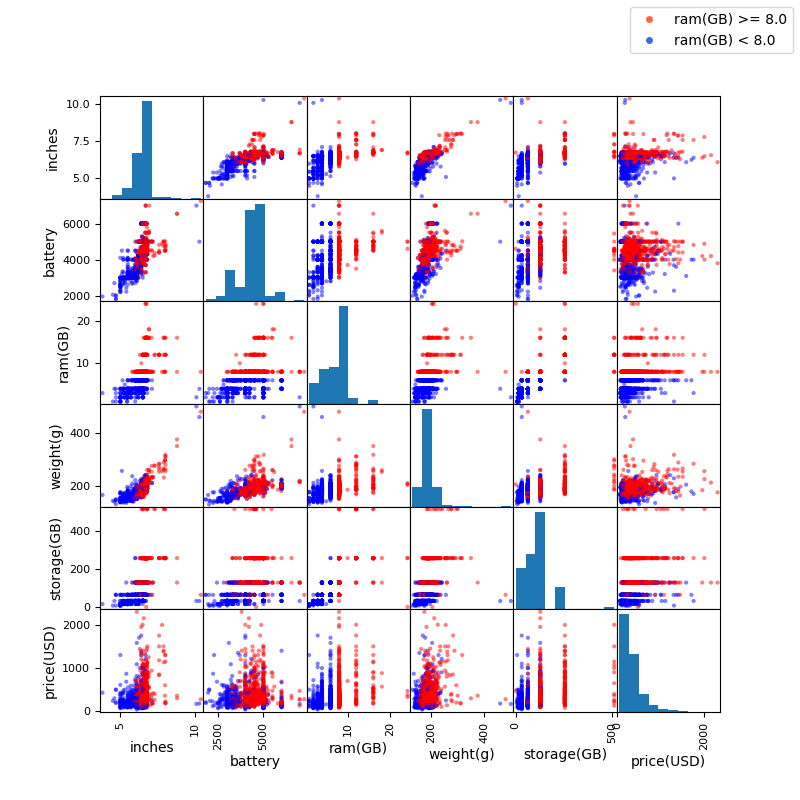


Рис. 29. Скатерограммы

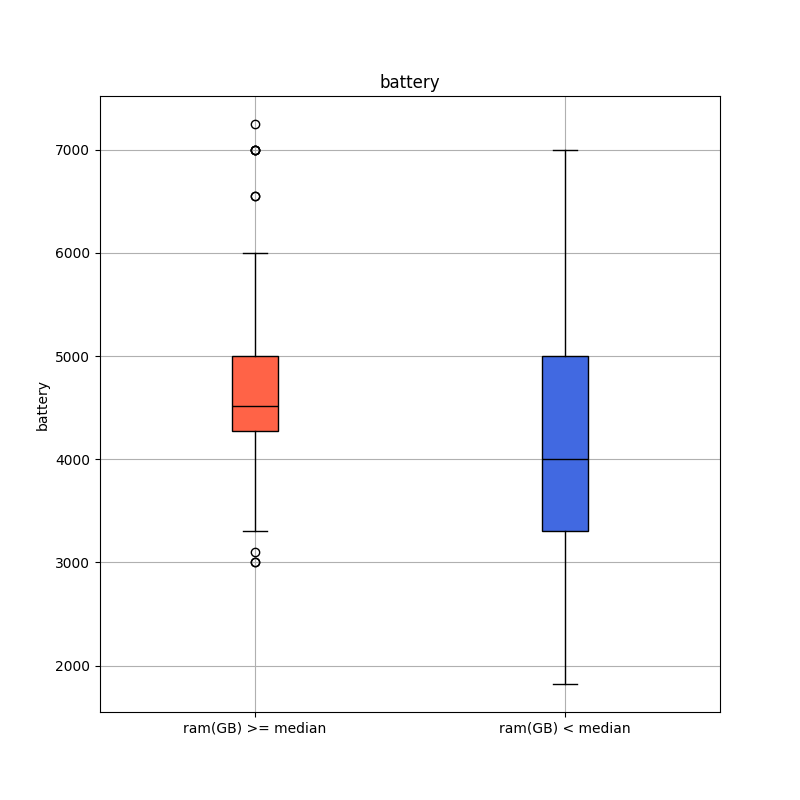


Рис. 30. Боксплот

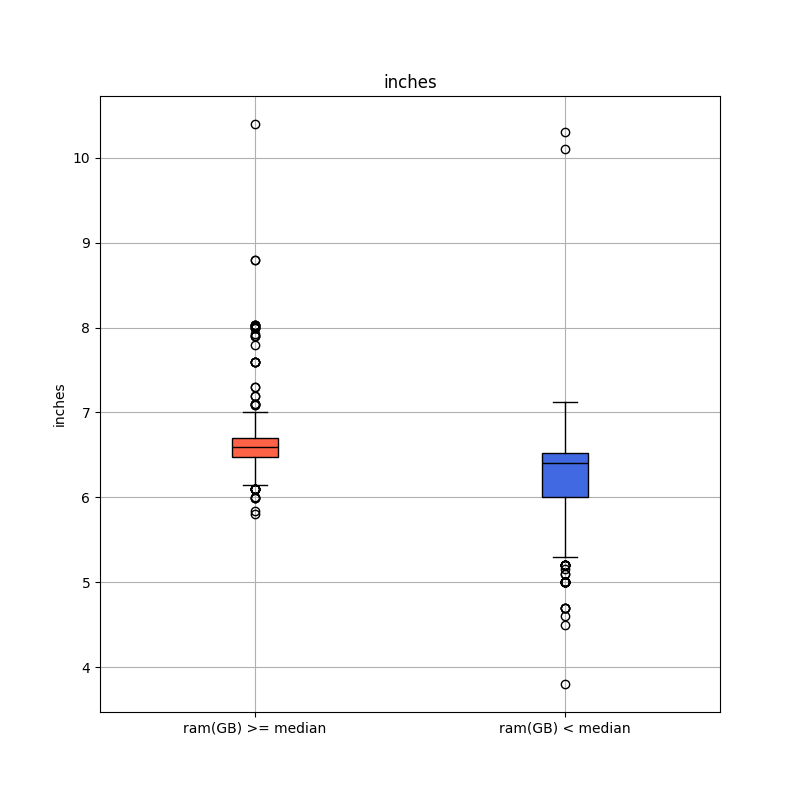


Рис. 31. Боксплот

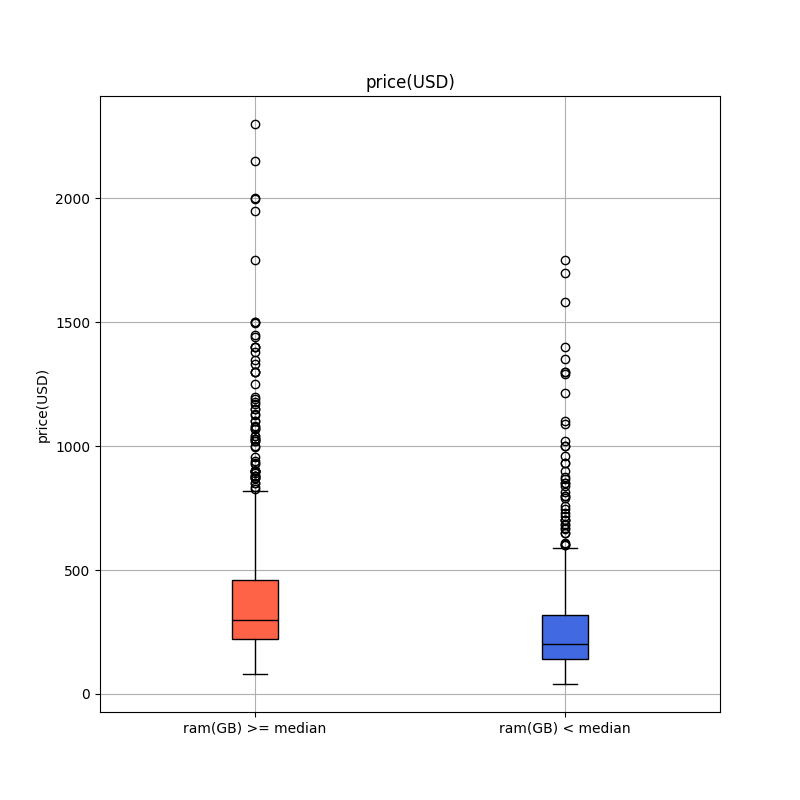


Рис. 32. Боксплот

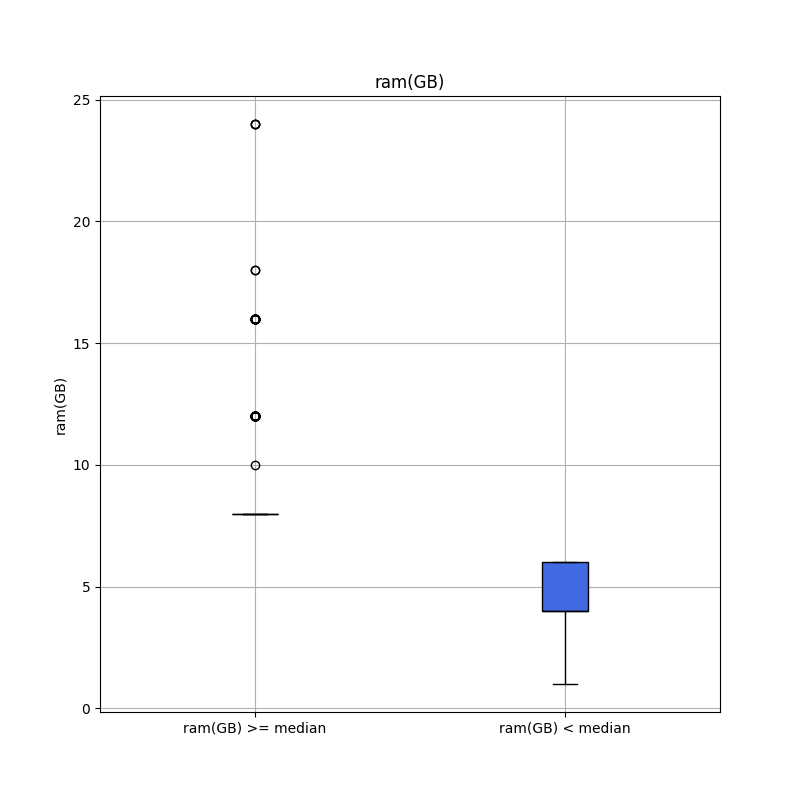


Рис. 33. Боксплот

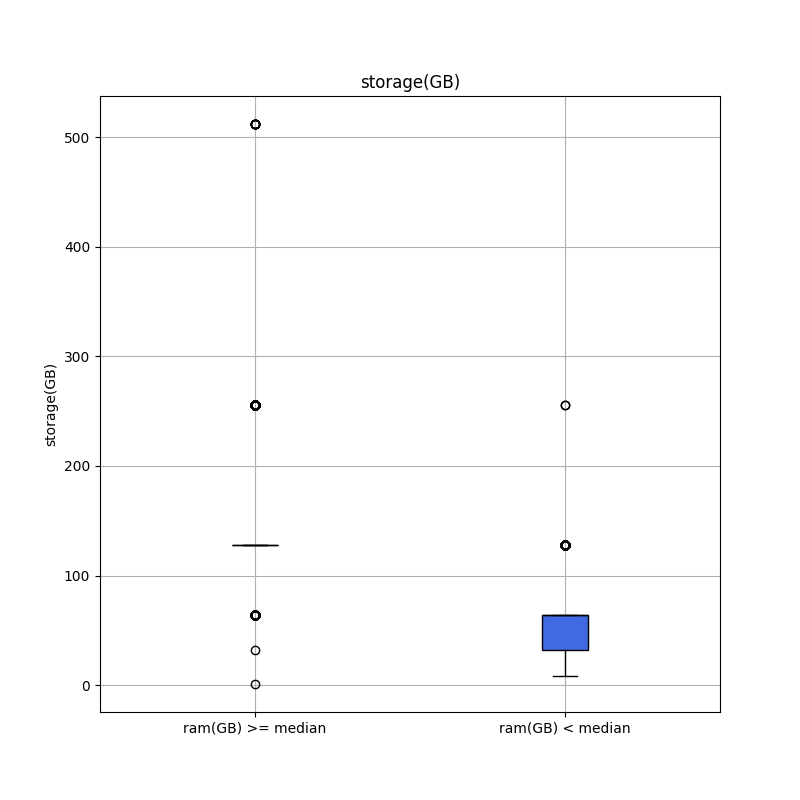


Рис. 34. Боксплот

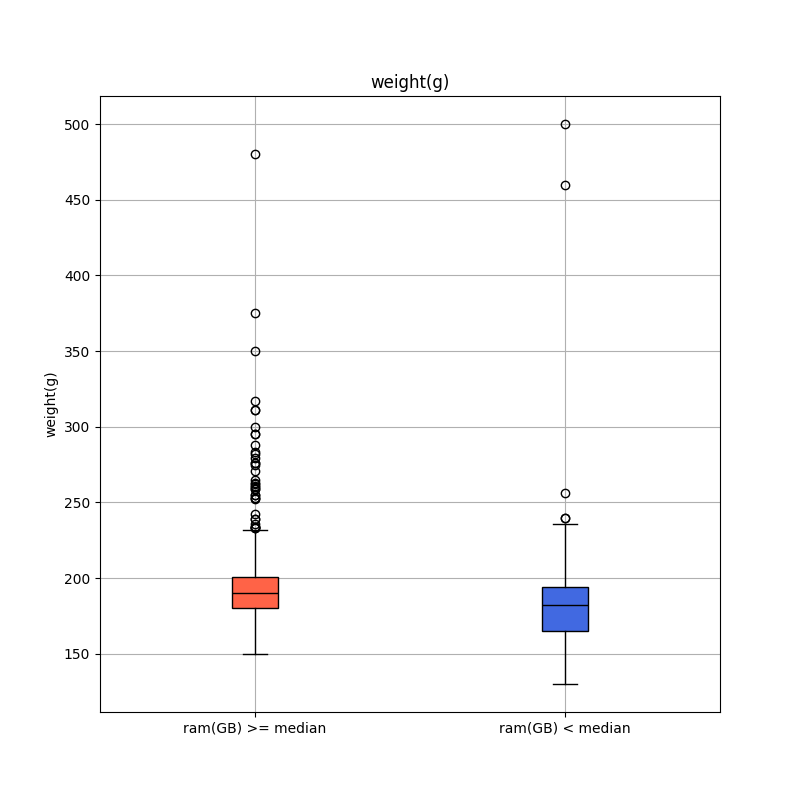


Рис. . Боксплот

4. Среднее значение и стандартное отклонение

Задание: рассчитать среднее значение и стандартное отклонение для параметров: для всей выборки и для каждого класса отдельно.

Данные разделены относительно медианы параметра ram(GB).

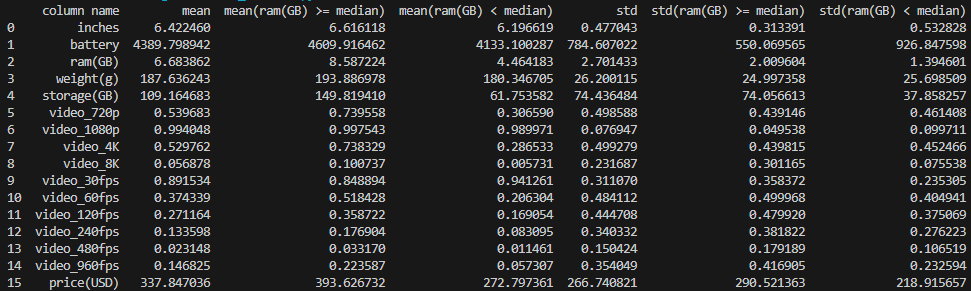


Рис. 36. Средние значения и стандартные отклонения

5. Различия в классах

5.1. Различия в классах по параметру n

Задание: статистически оценить различия в классах по параметру n- совпадают ли выборки по этому параметру или нет.

Для оценки использовался Т-тест для независимых выборок.

P-значение: основной показатель, который помогает определить, есть ли значимые различия между группами. Если p-значение меньше уровня значимости (обычно 0.05), то различия считаются значимыми.

Сначала была проведена проверка на нормальность распределения с помощью теста Шапиро-Уилка (если p-значение незначим, то распределение нормальное), затем была проведен тест Левина для проверки гипотезы о равенстве дисперсий (если p-значение большое, то дисперсии равны). Если дисперсии равны, то параметр equal\_var в Т-тесте будет равен True, иначе False. Если p-значение Т-теста меньше 0.05, то различия значительны. Также выводятся значения среднего, стандартного отклонения, медианы и дисперсии для каждой выборки.

Формула для т-теста:

где и – выборки, s – дисперсии, n – объем выборки.

Для столбца Age относительно последнего столбца (0 – Negative и 1 – Positive) (различия значительны):

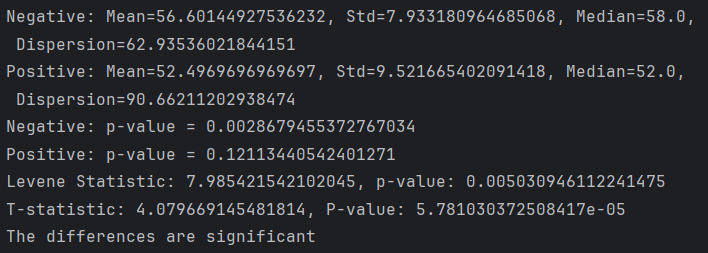
**

Рис. 37. Т-тест для столбца Age.

5.2. Различия в классах по параметру m

Задание: статистически оценить различия в классах по параметру m - совпадают ли выборки по этому параметру или нет.

Для столбца CholesterolLevel относительно последнего столбца (0 – Negative и 1 – Positive) (различия не значительны):

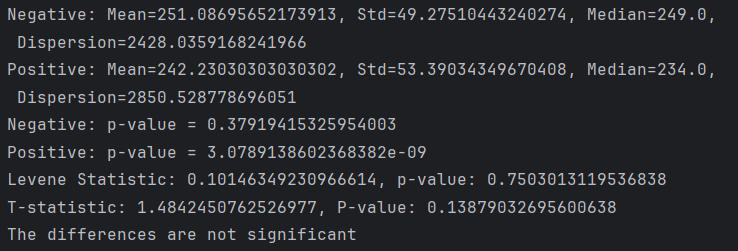


Рис. 38. Т-тест для столбца CholesterolLevel.

Листинг

1. Построить график распределения параметров. Разделить выборку по классам. Подписать оси, добавить легенду и сетку на график.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('Heart\_Attack.csv')

columns = df.columns[:-1].tolist()

last\_col = df.columns[-1]

for col in columns:

plt.hist(df[df[last\_col] == 0][col], bins=20, alpha=0.7, label='Negative')

plt.hist(df[df[last\_col] == 1][col], bins=20, alpha=0.7, label='Positive')

plt.title(f'График распределения {col}')

plt.xlabel(f'{col} value')

plt.ylabel('Number of objects')

plt.legend()

plt.grid(True)

plt.savefig('hist\_' + str(col) + '.png')

plt.show()

1. Рассчитать медиану параметра n: для выборки в целом, для каждого класса отдельно.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('Heart\_Attack.csv')

name\_col = 'Age'

col = df[name\_col]

# positive = df[df.iloc[:, 13] == 1]

# negative = df[df.iloc[:, 13] == 0]

negative = df[df['Disease'] == 0]

positive = df[df['Disease'] == 1]

neg\_col = negative[name\_col]

pos\_col = positive[name\_col]

med\_col = np.median(col) # медиана с использованием numpy

print('Numpy Median: ', med\_col)

med\_col\_neg = np.median(neg\_col)

med\_col\_pos = np.median(pos\_col)

print('Numpy Median Negative: ', med\_col\_neg)

print('Numpy Median Positive: ', med\_col\_pos, '\n')

col\_sort = sorted(col)

len\_col = len(col\_sort)

if len\_col % 2 == 0:

el1 = col\_sort[len\_col // 2 - 1]

el2 = col\_sort[len\_col // 2]

rez = (el1 + el2) / 2

print('Manual Median: ', rez)

else:

el = col\_sort[len\_col // 2]

print('Manual Median: ', el)

neg\_sort = sorted(neg\_col)

len\_neg = len(neg\_sort)

pos\_sort = sorted(pos\_col)

len\_pos = len(pos\_sort)

if len\_neg % 2 == 0:

el1 = neg\_sort[len\_neg // 2 - 1]

el2 = neg\_sort[len\_neg // 2]

rez = (el1 + el2) / 2

print('Manual Median Negative: ', rez)

elif len\_neg % 2 != 0:

el = neg\_sort[len\_neg // 2]

print('Manual Median Negative: ', el)

if len\_pos % 2 == 0:

el1 = pos\_sort[len\_pos // 2 - 1]

el2 = pos\_sort[len\_pos // 2]

rez = (el1 + el2) / 2

print('Manual Median Positive: ', rez)

elif len\_pos % 2 != 0:

el = pos\_sort[len\_pos // 2]

print('Manual Median Positive: ', el)

1. Рассчитать медиану параметра m: для выборки в целом, для каждого класса отдельно.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('Heart\_Attack.csv')

#name\_col = 'Age'

name\_col = 'CholesterolLevel'

col = df[name\_col]

# positive = df[df.iloc[:, 13] == 1]

# negative = df[df.iloc[:, 13] == 0]

negative = df[df['Disease'] == 0]

positive = df[df['Disease'] == 1]

neg\_col = negative[name\_col]

pos\_col = positive[name\_col]

med\_col = np.median(col) # медиана с использованием numpy

print('Numpy Median: ', med\_col)

med\_col\_neg = np.median(neg\_col)

med\_col\_pos = np.median(pos\_col)

print('Numpy Median Negative: ', med\_col\_neg)

print('Numpy Median Positive: ', med\_col\_pos, '\n')

col\_sort = sorted(col)

len\_col = len(col\_sort)

if len\_col % 2 == 0:

el1 = col\_sort[len\_col // 2 - 1]

el2 = col\_sort[len\_col // 2]

rez = (el1 + el2) / 2

print('Manual Median: ', rez)

else:

el = col\_sort[len\_col // 2]

print('Manual Median: ', el)

neg\_sort = sorted(neg\_col)

len\_neg = len(neg\_sort)

pos\_sort = sorted(pos\_col)

len\_pos = len(pos\_sort)

if len\_neg % 2 == 0:

el1 = neg\_sort[len\_neg // 2 - 1]

el2 = neg\_sort[len\_neg // 2]

rez = (el1 + el2) / 2

print('Manual Median Negative: ', rez)

elif len\_neg % 2 != 0:

el = neg\_sort[len\_neg // 2]

print('Manual Median Negative: ', el)

if len\_pos % 2 == 0:

el1 = pos\_sort[len\_pos // 2 - 1]

el2 = pos\_sort[len\_pos // 2]

rez = (el1 + el2) / 2

print('Manual Median Positive: ', rez)

elif len\_pos % 2 != 0:

el = pos\_sort[len\_pos // 2]

print('Manual Median Positive: ', el)

1. Построить график распределения параметров для объектов выше медианного значения в выборке и ниже. Подписать оси, добавить легенду и сетку на график.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('Heart\_Attack.csv')

name\_col = 'Age'

colum = df[name\_col]

med\_col = int(np.median(colum)) # медиана с использованием numpy

print('Numpy Median: ', med\_col)

columns = df.columns[:].tolist()

i = 0

for col in columns:

plt.hist(df[df[name\_col] <= med\_col][col], bins=20, alpha=0.7, label='Меньше медианы')

plt.hist(df[df[name\_col] >= med\_col][col], bins=20, alpha=0.7, label='Больше медианы')

plt.title(f'График распределения {col}')

plt.xlabel(f'{col} value')

plt.ylabel('Number of objects')

plt.legend()

plt.grid(True)

plt.savefig(f'{i}\_hist\_mediana\_' + str(col) + '.png')

plt.show()

i += 1

1. Построить гистограмму распределения, скатерограмму и боксплот параметров для объектов выше медианного значения в выборке и ниже. Подписать оси, добавить легенду и сетку на график.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from matplotlib.lines import Line2D

df = pd.read\_csv('Heart\_Attack.csv')

name\_col = 'Age'

colum = df[name\_col]

med\_col = int(np.median(colum)) # медиана с использованием numpy

print('Numpy Median: ', med\_col)

columns = ['Age', 'RestingBloodPressure', 'CholesterolLevel', 'HighestBeatsPerMinute', 'DepressionST']

columns1 = ['Age', 'Gender', 'ChestPain', 'FastingBloodSugar', 'RestingECG', 'ExerciseInducedAngina']

columns2 = ['Age', 'PeakSTSegmentSlope', 'NumberOfLargeVessels', 'AbsorptionWaist', 'Disease']

legend\_elements = \

[Line2D([0], [0], marker='o', color='w', markerfacecolor='tomato', markersize=6, label=f'{name\_col} ≤ {med\_col}'),

Line2D([0], [0], marker='o', color='w', markerfacecolor='royalblue', markersize=6, label=f'{name\_col} > {med\_col}')]

### SCATTER

colors = df[name\_col].apply(lambda x: 'red' if x <= med\_col else 'blue')

for i, col\_group in enumerate([columns, columns1, columns2]):

pd.plotting.scatter\_matrix(df[col\_group], color=colors, figsize=(10, 10))

plt.figlegend(handles=legend\_elements, loc='upper right', fontsize=10)

plt.savefig(f'{i}\_scatter.png')

plt.show()

### BOXPLOT

under\_med = df[df[name\_col] <= med\_col]

above\_med = df[df[name\_col] > med\_col]

labels = ['Under median', 'Above median']

color\_ = ['tomato', 'royalblue']

for col in columns:

under\_col = under\_med[col]

above\_col = above\_med[col]

data = [under\_col, above\_col]

fig, ax = plt.subplots()

ax.set\_ylabel(col)

bp = ax.boxplot(data, patch\_artist = True, tick\_labels = labels)

for patch, color in zip(bp['boxes'], color\_):

patch.set\_facecolor(color)

plt.title(f"Box plot {col}")

plt.figlegend(handles=legend\_elements, loc='upper right', fontsize=10)

plt.grid()

plt.savefig(f'boxplot\_{col}.png')

plt.show()

1. Рассчитать среднее значение и стандартное отклонение для параметров: для всей выборки и для каждого класса отдельно.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from tabulate import tabulate

df = pd.read\_csv('Heart\_Attack.csv')

table\_data = []

name\_col = df.columns[:].tolist()

for name in name\_col:

col = df[name]

negative = df[df['Disease'] == 0]

positive = df[df['Disease'] == 1]

neg\_col = negative[name]

pos\_col = positive[name]

len\_col = len(col)

sum\_col = sum(col)

print(f'{name}:')

mean\_col = np.mean(col) # среднее с использованием numpy

print(' Numpy Mean: ', mean\_col)

manual\_mean = sum\_col / len\_col

print('Manual Mean: ', manual\_mean)

mean\_col\_neg = np.mean(neg\_col)

mean\_col\_pos = np.mean(pos\_col)

print('Numpy Mean Negative: ', mean\_col\_neg)

print('Numpy Mean Positive: ', mean\_col\_pos)

manual\_neg\_mean = sum(neg\_col) / len(neg\_col)

print('Manual Mean Negative: ', manual\_neg\_mean)

manual\_pos\_mean = sum(pos\_col) / len(pos\_col)

print('Manual Mean Positive: ', manual\_pos\_mean)

difference = []

difference\_neg = []

difference\_pos = []

for i in col:

difference.append((i - manual\_mean) \*\* 2)

for i in neg\_col:

difference\_neg.append((i - manual\_neg\_mean) \*\* 2)

for i in pos\_col:

difference\_pos.append((i - manual\_pos\_mean) \*\* 2)

manual\_std\_col = np.sqrt(sum(difference) / len\_col)

manual\_std\_neg = np.sqrt(sum(difference\_neg) / len(neg\_col))

manual\_std\_pos = np.sqrt(sum(difference\_pos) / len(pos\_col))

std\_col = np.std(col) # стандартное отклонение с использованием numpy

print(' Numpy STD: ', std\_col)

print('Manual STD: ', manual\_std\_col)

std\_col\_neg = np.std(neg\_col)

std\_col\_pos = np.std(pos\_col)

print('Numpy STD Negative: ', std\_col\_neg)

print('Numpy STD Positive: ', std\_col\_pos)

print('Manual STD Negative: ', manual\_std\_neg)

print('Manual STD Positive: ', manual\_std\_pos, '\n')

table\_data.append([name, mean\_col, mean\_col\_neg, mean\_col\_pos, std\_col, std\_col\_neg, std\_col\_pos])

df\_table = pd.DataFrame(table\_data, columns=['Colum', 'Mean', 'Mean Negative', 'Mean Positive', 'STD', 'STD Negative', 'STD Positive'])

print(tabulate(df\_table, headers='keys', tablefmt='pretty'))

1. Статистически оценить различия в классах по параметру n- совпадают ли выборки по этому параметру или нет.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

from scipy.stats import levene

df = pd.read\_csv('Heart\_Attack.csv')

name\_col = 'Age'

#name\_col = 'CholesterolLevel'

col = df[name\_col]

negative = df[df['Disease'] == 0]

positive = df[df['Disease'] == 1]

neg\_col = negative[name\_col]

pos\_col = positive[name\_col]

mean\_col\_neg = np.mean(neg\_col)

mean\_col\_pos = np.mean(pos\_col)

std\_col\_neg = np.std(neg\_col)

std\_col\_pos = np.std(pos\_col)

med\_col\_neg = np.median(neg\_col)

med\_col\_pos = np.median(pos\_col)

dis\_col\_neg = np.var(neg\_col)

dis\_col\_pos = np.var(pos\_col)

print(f'Negative: Mean={mean\_col\_neg}, Std={std\_col\_neg}, Median={med\_col\_neg}, Dispersion={dis\_col\_neg}')

print(f'Positive: Mean={mean\_col\_pos}, Std={std\_col\_pos}, Median={med\_col\_pos}, Dispersion={dis\_col\_pos}')

stat, p = stats.shapiro(neg\_col)

print(f'Negative: p-value = {p}')

stat, p = stats.shapiro(pos\_col)

print(f'Positive: p-value = {p}')

lev = levene(neg\_col, pos\_col)

print(f'Levene Statistic: {lev.statistic}, p-value: {lev.pvalue}')

if lev.pvalue >= 0.05:

eq = True

else:

eq = False

t\_stat, p\_value = stats.ttest\_ind(neg\_col, pos\_col, equal\_var=eq)

print(f"T-statistic: {t\_stat}, P-value: {p\_value}")

if p\_value < 0.05:

print('The differences are significant')

else:

print('The differences are not significant')

1. Статистически оценить различия в классах по параметру m - совпадают ли выборки по этому параметру или нет.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

from scipy.stats import levene

df = pd.read\_csv('Heart\_Attack.csv')

#name\_col = 'Age'

name\_col = 'CholesterolLevel'

col = df[name\_col]

negative = df[df['Disease'] == 0]

positive = df[df['Disease'] == 1]

neg\_col = negative[name\_col]

pos\_col = positive[name\_col]

mean\_col\_neg = np.mean(neg\_col)

mean\_col\_pos = np.mean(pos\_col)

std\_col\_neg = np.std(neg\_col)

std\_col\_pos = np.std(pos\_col)

med\_col\_neg = np.median(neg\_col)

med\_col\_pos = np.median(pos\_col)

dis\_col\_neg = np.var(neg\_col)

dis\_col\_pos = np.var(pos\_col)

print(f'Negative: Mean={mean\_col\_neg}, Std={std\_col\_neg}, Median={med\_col\_neg}, Dispersion={dis\_col\_neg}')

print(f'Positive: Mean={mean\_col\_pos}, Std={std\_col\_pos}, Median={med\_col\_pos}, Dispersion={dis\_col\_pos}')

stat, p = stats.shapiro(neg\_col)

print(f'Negative: p-value = {p}')

stat, p = stats.shapiro(pos\_col)

print(f'Positive: p-value = {p}')

lev = levene(neg\_col, pos\_col)

print(f'Levene Statistic: {lev.statistic}, p-value: {lev.pvalue}')

if lev.pvalue >= 0.05:

eq = True

else:

eq = False

t\_stat, p\_value = stats.ttest\_ind(neg\_col, pos\_col, equal\_var=eq)

print(f"T-statistic: {t\_stat}, P-value: {p\_value}")

if p\_value < 0.05:

print('The differences are significant')

else:

print('The differences are not significant')

Заключение

В ходе выполнения курсовой работы были изучены .

Список использованных источников

1. Kaggle датасет «Heart Attack» URL: <https://www.kaggle.com/datasets/pritsheta/heart-attack>
2. Википедия «T-критерий Стьюдента» URL: <https://ru.wikipedia.org/wiki/T-%D0%BA%D1%80%D0%B8%D1%82%D0%B5%D1%80%D0%B8%D0%B9_%D0%A1%D1%82%D1%8C%D1%8E%D0%B4%D0%B5%D0%BD%D1%82%D0%B0>