

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
Национальный исследовательский технологический университет «МИСИС»
Институт информационных технологий и автоматизированных систем управления
Кафедра Бизнес-информатики и систем управления производством

Практическая работа №1

по дисциплине «Статистические методы анализа данных в принятии решений»
на тему «Подготовка и предварительный анализ данных»

Направление подготовки
38.03.05 Бизнес-информатика
Семестр 4

Выполнил:

Сычиков Владимир Андреевич

(ФИО студента)

ББИ-23-6

(№ группы)

26.02.2025

(дата сдачи)

Подпись:

Проверил:

(ФИО преподавателя)

(оценка)

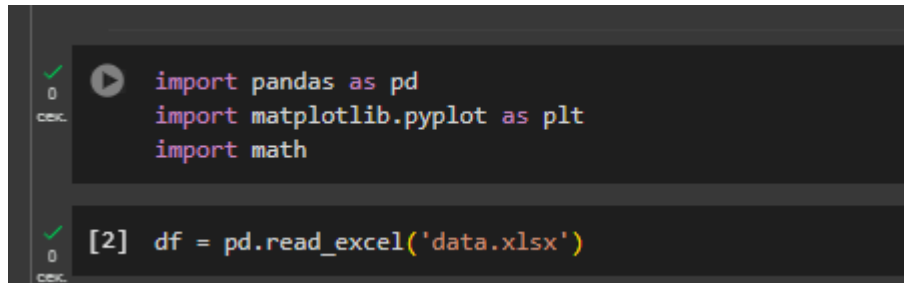
(дата проверки)

Подпись:

Москва – 2025

Ход работы:

Для начала работы я импортировал все необходимые библиотеки(pandas, matplotlib, math) с помощью команды import, а также задал элиасы для удобства обращения к библиотеке. Далее я подружил выборку и записал ее в датафрейм df.

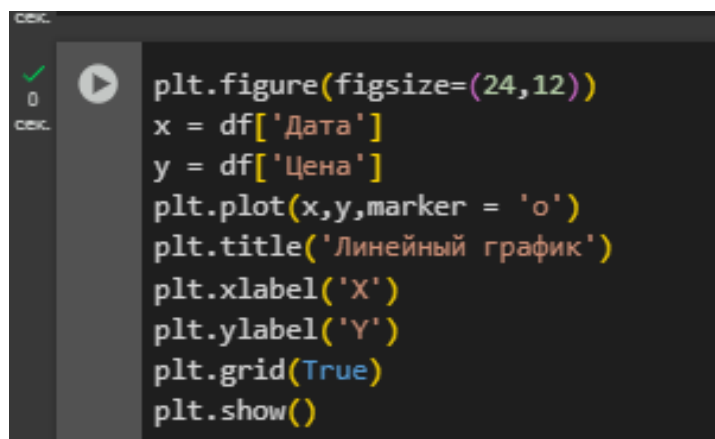


```
import pandas as pd
import matplotlib.pyplot as plt
import math

[2] df = pd.read_excel('data.xlsx')
```

Рисунок 1 - Импорт библиотек

Рисунок 2 описывает процесс создания линейного графика зависимости Даты от Цены. Для этого использовалась команда plt.figure(figsize=(24,12)) для создания области построения графика с заданными размерами. По оси x откладываются значения Даты, а по оси y — значения Цены. Данные берутся из предварительно загруженной выборки. Код сначала создает область для графика с помощью plt.figure и задает размер. Затем переменной x присваивается столбец 'Дата' из dataframe df, а переменной y - столбец 'Цена'. Функция plt.plot строит график по этим данным, marker='o' указывает, что точки на графике будут отмечены кружками. plt.title, plt.xlabel и plt.ylabel задают заголовок и подписи осей. plt.grid(True) добавляет сетку на график, а plt.show() отображает график.



```
plt.figure(figsize=(24,12))
x = df['Дата']
y = df['Цена']
plt.plot(x,y,marker = 'o')
plt.title('Линейный график')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)
plt.show()
```

Рисунок 2 – Код для построения линейного графика 1

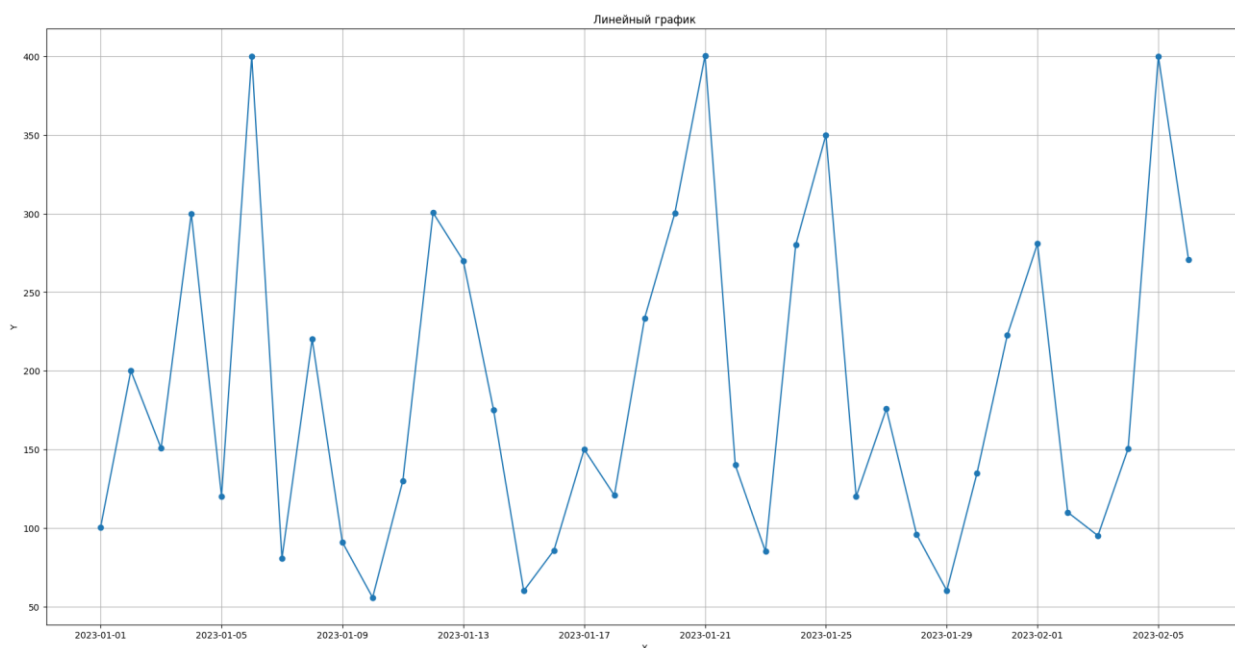


Рисунок 3 - Линейный график 1

График демонстрирует значительную волатильность значений с 1 января по 6 февраля 2023 года, характеризующуюся выраженными пиками и спадами без четкой восходящей или нисходящей тенденции.

На рисунке 4 код строит линейный график, отображающий зависимость между датой и количеством, для этого создается область для графика с помощью `plt.figure(figsize=(24,12))` и задается её размер, затем переменной `x` присваиваются значения из столбца 'Дата' фрейма данных `df`, которые будут отложены по горизонтальной оси графика, а переменной `y` - значения из столбца 'Количество', которые будут отложены по вертикальной оси графика, функция `plt.plot(x, y, marker='o')` строит линейный график, используя данные из `x` и `y`, параметр `marker='o'` указывает, что каждая точка данных будет отмечена кружочком, `plt.title('Линейный график')` устанавливает заголовок графика, а `plt.xlabel('X')` и `plt.ylabel('Y')` устанавливают подписи для горизонтальной и вертикальной осей соответственно, `plt.grid(True)` включает отображение сетки на графике, а `plt.show()` отображает построенный график.

```

plt.figure(figsize=(24,12))
x = df['Дата']
y = df['Количество']
plt.plot(x,y,marker = 'o')
plt.title('Линейный график')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)
plt.show()

```

Рисунок 4 - Код для построения линейного графика 2

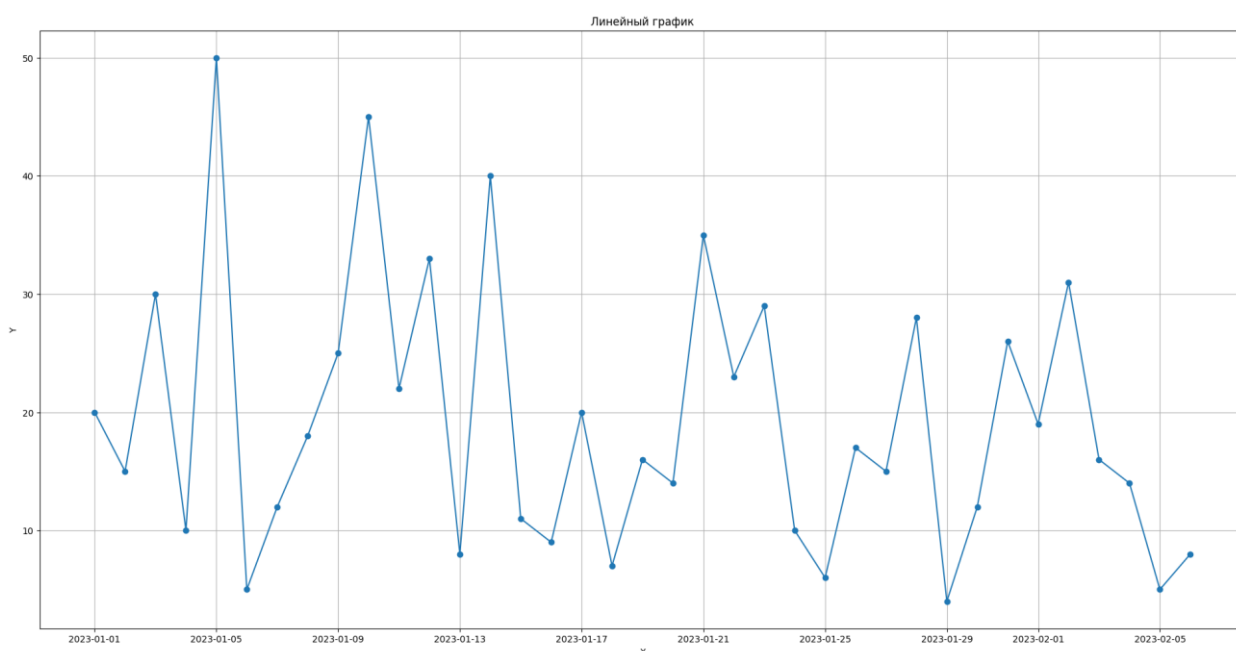


Рисунок 5 - Линейный график 2

Линейный график с 1 января по 6 февраля 2023 года показывает значительные колебания значений. Наблюдаются резкие пики и спады, особенно в начале периода. Четкой тенденции нет.

На рисунке 6 код создает диаграмму рассеяния, показывающую связь между количеством и ценой, начиная с создания области для графика заданного размера с помощью `plt.figure(figsize=(24,12))`, далее переменной `x` присваиваются значения из столбца 'Количество' фрейма данных `df`, а переменной `y` присваиваются значения из столбца 'Цена', функция `plt.scatter(x, y, color='blue')` создает диаграмму рассеяния, используя данные из `x` и `y`, при этом точки на графике отображаются синим цветом, заголовок графика не задан (`plt.title("")`), `plt.xlabel('X')` и `plt.ylabel('Y')` устанавливают подписи для горизонтальной и вертикальной осей соответственно, и `plt.grid(True)` включает отображение сетки, а `plt.show()` отображает график.

```

plt.figure(figsize=(24,12))
x = df['Количество']
y = df['Цена']
plt.scatter(x,y,color='blue')
plt.title('')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)
plt.show()

```

Рисунок 6 - Код для построения графика рассеяния 1

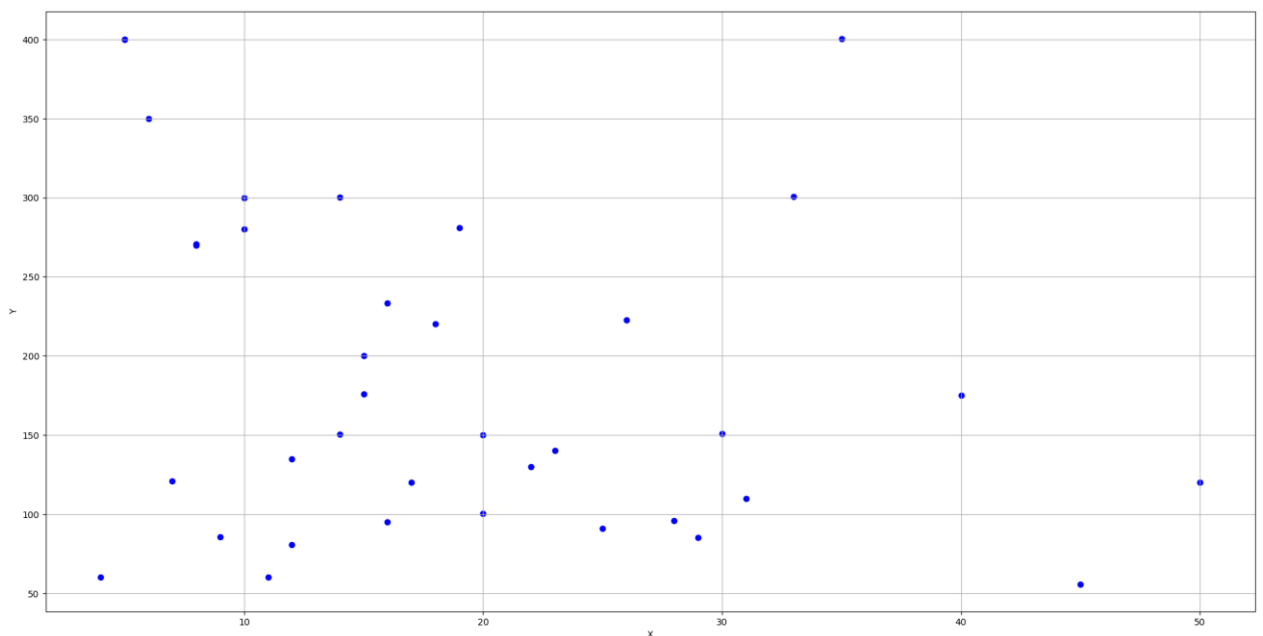


Рисунок 7 - График рассеяния 1

Диаграмма рассеяния показывает слабо выраженную зависимость между переменными X и Y. Точки распределены хаотично по всей области графика, что не позволяет выделить явную тенденцию или закономерность.

На рисунке 8 код создает гистограмму, отображающую распределение цен из датафрейма df, сначала вычисляется оптимальное количество бинов (столбцов) для гистограммы с использованием формулы Стерджеса: $x = 1 + (3.322 * \mathit{math.log}(37))$, где 37 вероятно является количеством наблюдений, затем с помощью функции `plt.hist(df['Цена'], bins = int(x), color='orange', edgecolor='black')` создается гистограмма, используя данные из столбца 'Цена' датафрейма df, количество бинов задается вычисленным значением x, гистограмма заполняется оранжевым цветом, а границы столбцов делаются черными, `plt.title('Гистограмма')` устанавливает заголовок графика, `plt.xlabel('Значения')` и `plt.ylabel('Частота')` задают подписи для горизонтальной и вертикальной осей соответственно, и наконец, `plt.grid(True)` добавляет сетку на график, а `plt.show()` отображает созданную гистограмму.

```
x = 1 + (3.322*math.log(37))
print(x)
plt.hist(df['Цена'], bins = int(x), color='orange',edgecolor='black')
plt.title('Гистограмма')
plt.xlabel('Значения')
plt.ylabel('Частота')
plt.grid(True)
plt.show()
```

Рисунок 8 - Код для построения гистограммы 1

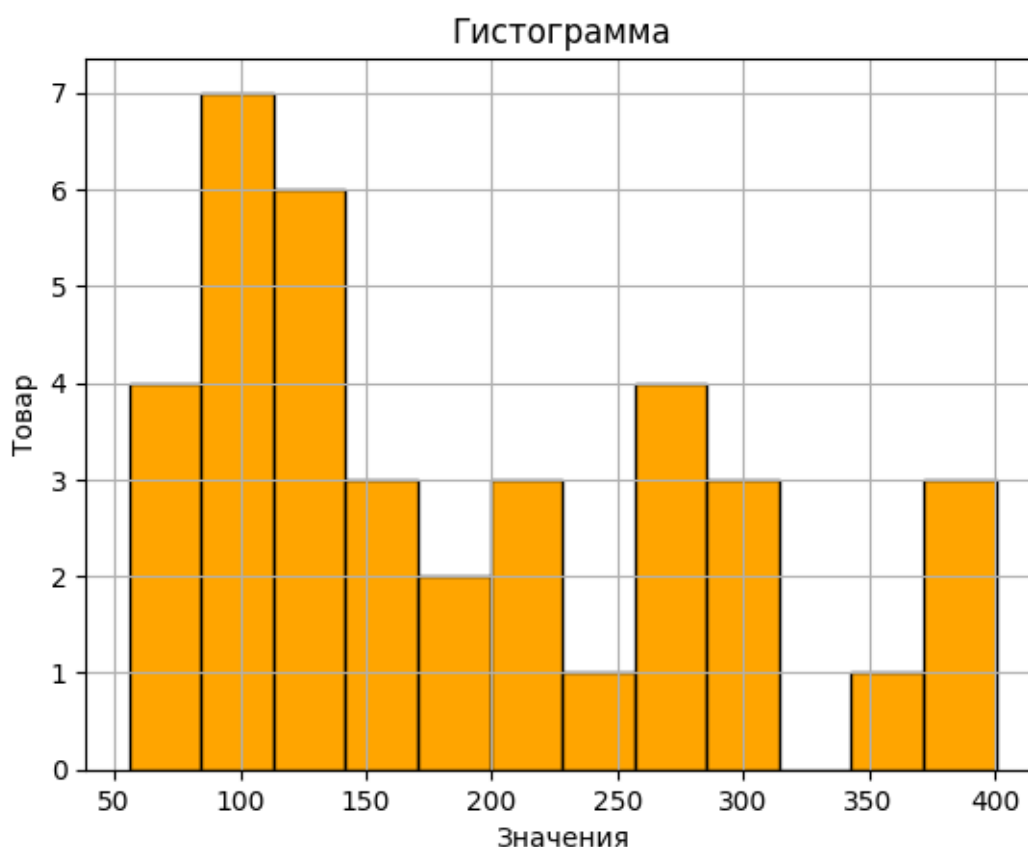


Рисунок 9 - Гистограмма 1

Гистограмма показывает, что значения данных (на оси X) распределены неравномерно. Наблюдается несколько пиков частоты (на оси Y, ошибочно обозначенной как "Товар"), что указывает на наличие нескольких модальных значений. Распределение не является ни нормальным, ни равномерным.

На рисунке 10 код создает гистограмму накопленных частот, отражающую кумулятивное распределение цен из датафрейма df. Сначала вычисляется приблизительное количество бинов для гистограммы, используя формулу Стерджеса: $x = 1 + (3.22 * \log(37))$, где 37, вероятно, представляет собой количество данных. Затем, с помощью функции `plt.hist(df['Цена'], bins = int(x), cumulative = True, color = 'green', edgecolor = 'black')`, создается гистограмма, используя данные из столбца 'Цена' датафрейма df.

Параметр `cumulative = True` указывает, что будет построена гистограмма накопленных частот, `bins` определяет количество интервалов (столбцов), `color = 'green'` задает зеленый цвет для заполнения столбцов, а `edgecolor = 'black'` устанавливает черный цвет для границ столбцов. Далее, `plt.title('Гистограмма накопленной части')` устанавливает заголовок графика, `plt.xlabel('Значение')` и `plt.ylabel('Накопленная частота')` задают подписи для горизонтальной и вертикальной осей соответственно. Наконец, `plt.grid(True)` добавляет сетку на график для облегчения визуального анализа, и `plt.show()` отображает созданную гистограмму накопленных частот.

```
x = 1 + (3.22*math.log(37))
plt.hist(df['Цена'], bins = int(x), cumulative = True, color = 'green', edgecolor = 'black')
plt.title('Гистограмма накопленной части')
plt.xlabel('Значение')
plt.ylabel('Накопленная частота')
plt.grid(True)
plt.show()
```

Рисунок 10 - Код для построения гистограммы 2

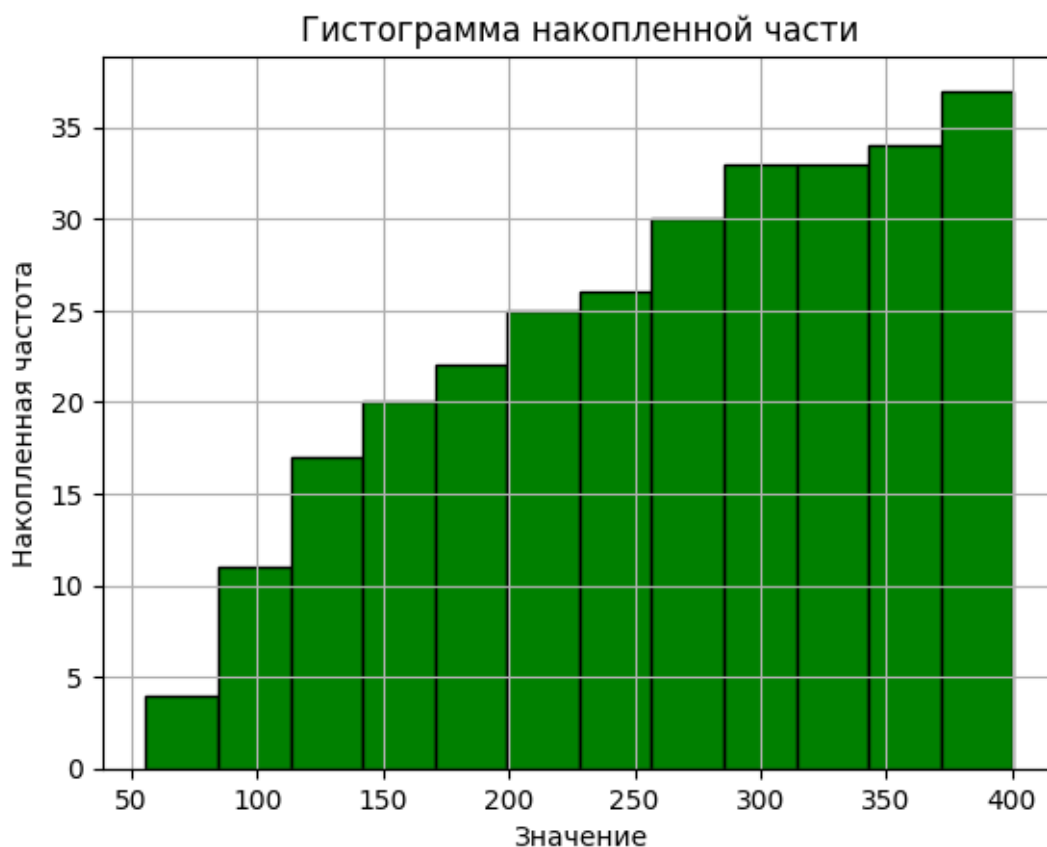


Рисунок 11 - Гистограмма 2

Гистограмма накопленной части показывает, как накапливается частота значений. Заметен постепенный рост накопленной частоты с увеличением значений, что говорит о том, что большинство значений распределено в диапазоне от 50 до 400. Наиболее резкий

рост наблюдается в начале, что указывает на более высокую концентрацию меньших значений.

На рисунке 12 код генерирует гистограмму, визуализирующую распределение цен, где по оси Y указаны "Товар" (что, вероятно, не совсем корректно, учитывая, что это гистограмма частот). Сначала вычисляется количество столбцов гистограммы (bins) по формуле Стерджеса: $x = 1 + (3.322 * \log(37))$, где 37, вероятно, размер выборки. Это значение выводится на экран с помощью `print(x)`. Затем с помощью `plt.hist(df['Цена'], bins = int(x), color='orange', edgecolor='black')` строится гистограмма на основе столбца 'Цена' из датафрейма `df`, с количеством столбцов, равным `x`, оранжевым цветом заполнения и черными границами. `plt.title('Гистограмма')` устанавливает заголовок графика, `plt.xlabel('Значения')` подписывает горизонтальную ось (ось X), а `plt.ylabel('Товар')` ошибочно подписывает вертикальную ось (ось Y) как "Товар" – корректнее было бы "Частота" или "Количество". `plt.grid(True)` добавляет сетку для улучшения читаемости, и, наконец, `plt.show()` отображает полученную гистограмму. Важно отметить, что подпись оси Y в данном случае вводит в заблуждение, так как гистограмма отображает частоту встречаемости значений, а не категорию товара.

A screenshot of a Jupyter Notebook cell with a dark background. On the left, there is a play button icon. The code is written in a light-colored font. It calculates the number of bins using the Sturges formula, prints the result, and then uses plt.hist to create an orange histogram with black outlines. The plot is titled 'Гистограмма', the x-axis is labeled 'Значения', and the y-axis is labeled 'Товар'. A grid is enabled, and the plot is shown.

```
x = 1 + (3.322*math.log(37))
print(x)
plt.hist(df['Цена'], bins = int(x), color='orange', edgecolor='black')
plt.title('Гистограмма')
plt.xlabel('Значения')
plt.ylabel('Товар')
plt.grid(True)
plt.show()
```

Рисунок 12 - Код для построения гистограммы 3

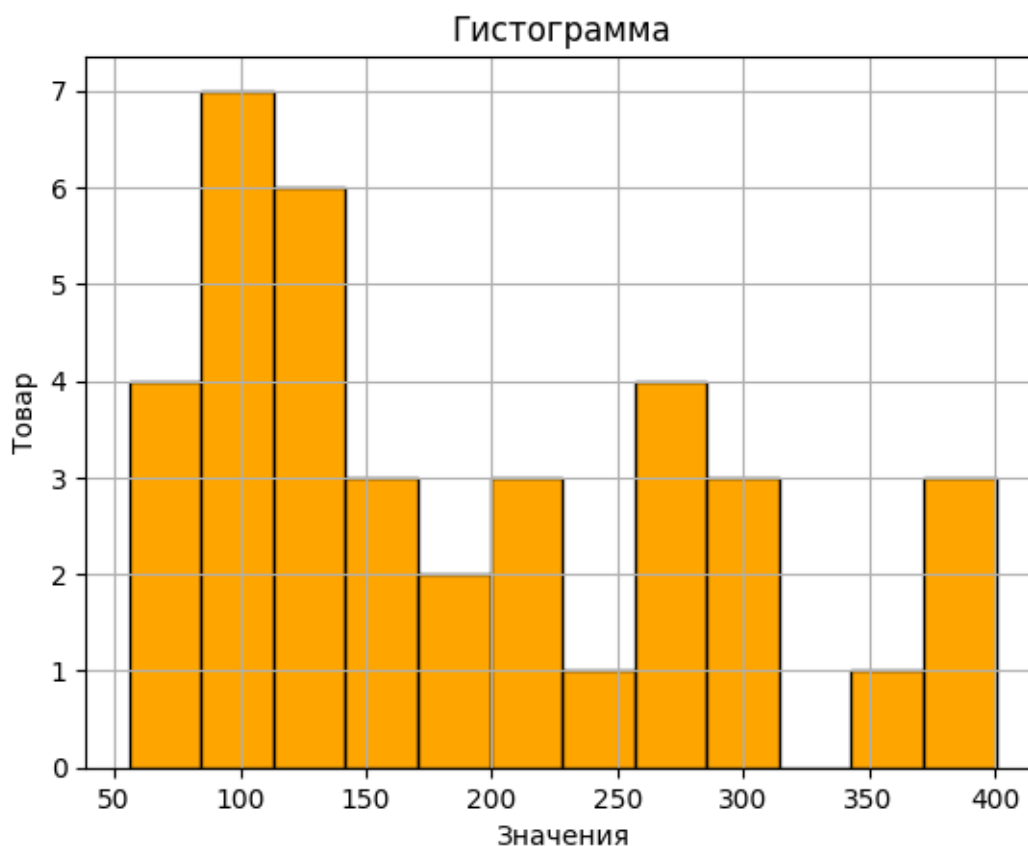


Рисунок 13 - Гистограмма 3

Гистограмма показывает, что значения имеют несколько модальных пиков, преимущественно в диапазоне 75-150. Распределение не является нормальным. Наименьшее количество "товаров" (ось Y) имеет значения в диапазоне 350-400.

Вывод:

Проведенная работа была направлена на визуализацию и предварительный анализ данных посредством создания линейных графиков, диаграмм рассеяния и гистограмм с использованием Python и библиотек matplotlib, pandas, а также math для расчета оптимального количества бинов для гистограмм, в результате чего удалось визуально исследовать взаимосвязи между переменными, оценить характер распределения данных и выявить наличие волатильности и отсутствия четких тенденций.