МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационной безопасности

ОТЧЁТ

по лабораторной работе №1

по дисциплине «Машинное обучение»

ТЕМА: ИССЛЕДОВАНИЕ НАБОРА ДАННЫХ

Студент(ка) гр. 2323 _____ Миляев Н. И.

Преподаватель _____ Писарев И. А.

Санкт-Петербург

2024

ЗАДАНИЕ

НА ИССЛЕДОВАНИЕ НАБОРА ДАННЫХ ЛАБОРАТОРНУЮ

Студентка Миляев Н.И.

Группа 2323

Тема лабораторной: Исследование набора данных

Задание на лабораторную:

1. Создать Jupyter Notebook, переименовать его «Lab 1, № Группы, ФИО»

1. Выбор исследуемого датасета.

2. Для каждого датасета представить краткое его описание в вашем Jupyter Notebook:

- предметная область, источник данных, характер данных (реальные или имитационные)
- какие атрибуты представлены в датасете, их тип (числовой, строковый (категории)), что они обозначают
- есть ли описание задачи анализа, если есть - представить

3. Для каждого атрибута нужно определить:

- среднее значение, ско
- построить гистограмму распределения значений, определить есть ли выбросы
- есть ли пропущенные значение, сколько
- предложить вариант обработки пропущенных значений

4. Определить корреляцию между параметрами

- какие атрибуты высококоррелированы, определить характер корреляции
- какие атрибуты не имеют корреляцию
- постройте графики рассеивания (предпочтительнее матрицу графиков рассеивания)
- проанализируйте полученные результаты.

Отчет должен включать описания выполнения каждой подзадачи.

1 Часть

Для выполнения данной практической работы был выбран датасет с сайта: https://www.kaggle.com/datasets, который был предложен в Задании 1.

Из списка датасетов был выбран «**Top 1500 games on steam by revenue 09-09-2024**

» (https://www.kaggle.com/datasets/alicemtopcu/top-1500-games-on-steam-by-revenue-09-09-2024?resource=download). Этот набор данных содержит исчерпывающую информацию о 1500 лучших играх, выпущенных в Steam в период с 1 января 2024 года по 9 сентября 2024 года. Данные собраны из 30 отдельных файлов и объединены в один набор данных.

Этот набор данных содержит следующие атрибуты:

- *name*: Представляет собой название игры. (строковый)
- *releaseDate* : официальная дата выхода игры, указывающая, когда она стала доступна широкой публике. (в формате год-месяц)
- *copiesSold*: общее количество проданных единиц или копий игры. (числовой)
- *price:* первоначальная розничная цена игры на момент её выпуска. (числовой)
- *revenue :* сумма денег, вырученная от продаж игры. (числовой)
- *avgPlaytime :* средняя продолжительность, которую игроки потратили на игру. (числовой)
- *reviewScore :* оценка или рейтинг, выставленные игре на основе отзывов пользователей и критиков. (числовой)
- *publisherClass :* лассификация издателя, указывающая на то, является ли издатель AAA, AA или независимым. (категории)
- *publishers :* Название компании, ответственной за публикацию игры. (строковый)
- *developers :* Название (имена) команды разработчиков или компании, создавшей игру. (строковый)
- *steamId :* уникальный идентификатор, присвоенный игре Steam для отслеживания и управления. (числовой)

2 Часть

1. Средние значения и СКО для атрибутов были определены с помощью функций mean() и std().

```
import numpy as np
import pandas as pd
df = pd.read_csv('dataset.csv')
Avgdata = np.array(df['copiesSold'])
AvgdataMean = Avgdata.mean()
AvgdataSko = np.std(Avgdata)
print(AvgdataMean)
print(AvgdataSko)
```

```
141482.57
1132379.0120487728
```

Рис. 1.1 – Пример кода для определения среднего значения и СКО для ***copiesSold***

Средние значения для атрибутов:
− **copiesSold** : 141482.57
− **price** : 17.52
− **revenue** : 2632381.98
− **avgPlaytime** : 12.563
− **reviewScore**: 76.201

СКО для атрибутов:
− **copiesSold** : 1132379.012
− **price** : 12.642
− **revenue** : 27800967.997
− **avgPlaytime** : 21.535
− **reviewScore**: 24.311

2. Гистограммы распределения значений были построены с помощью функции .hist() и изменена с помощью функций: .grid(True); .title()

```
import matplotlib.pyplot as plt
plt.hist(df['price'], bins=50)
plt.grid(True)
plt.title('Game price')
```

Рис. 1.2 – Пример кода для построения гистограммы распределения значений для price

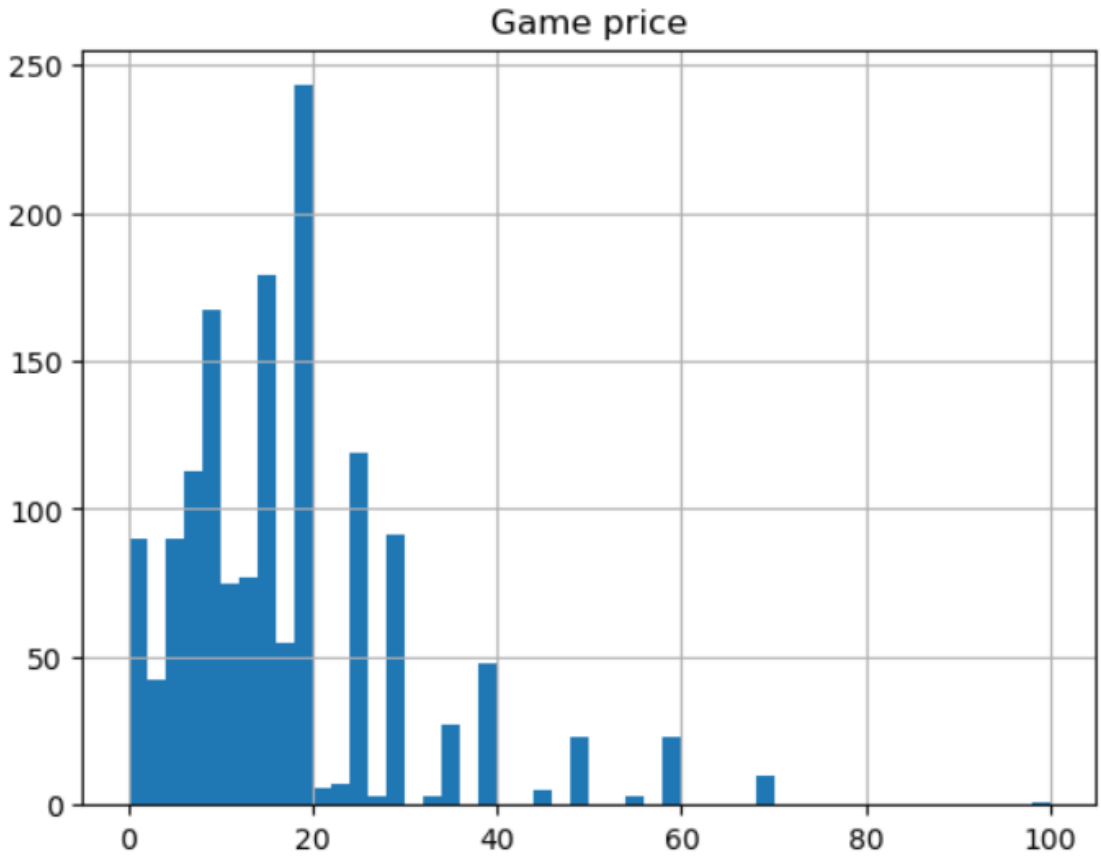


Рис. 1.3 – Гистограмма распределения значений для price

Наличие выбросов на гистограммах распределения значений определялось с помощью вызова функции detect_outliers_iqr().

Функция искала выбросы по алгоритму Межквартильного размаха:

1. Определение квартилей

Квартиль — это значение, которое разделяет набор данных на четыре равные части.

Первый квартиль(Q1) – это значение, ниже которого находится 25% данных.

Третий квартиль (Q3) – это значение, ниже которого находится 75% данных.

2. Вычисление IQR

Межквартильный размах (IQR) – это разница между третьим и первым квартилями.

IQR = Q3 - Q1

3. Определение границ для выявления выбросов

LowerBound = Q1 - 1.5 * IQR

UpperBound = Q3 + 1.5 * IQR

4. Поиск выбросов

Значения, которые меньше нижней границы или больше верхней границы.

Рис. 1.4 – Гистограмма распределения значений для price

```
Среднее значение цены:  76.20133333333334
СКО цены:  24.311330655112695
Выбросы:
0       99.99
1       59.99
2       49.99
4       59.99
10      49.99
        ...
1488    34.99
1490    69.99
1493    39.99
1497    34.99
1498    59.99
Name: price, Length: 140, dtype: float64
Количество выбросов, обнаруженных по методу IQR:  140
Количество пропущенных значений: 0
```

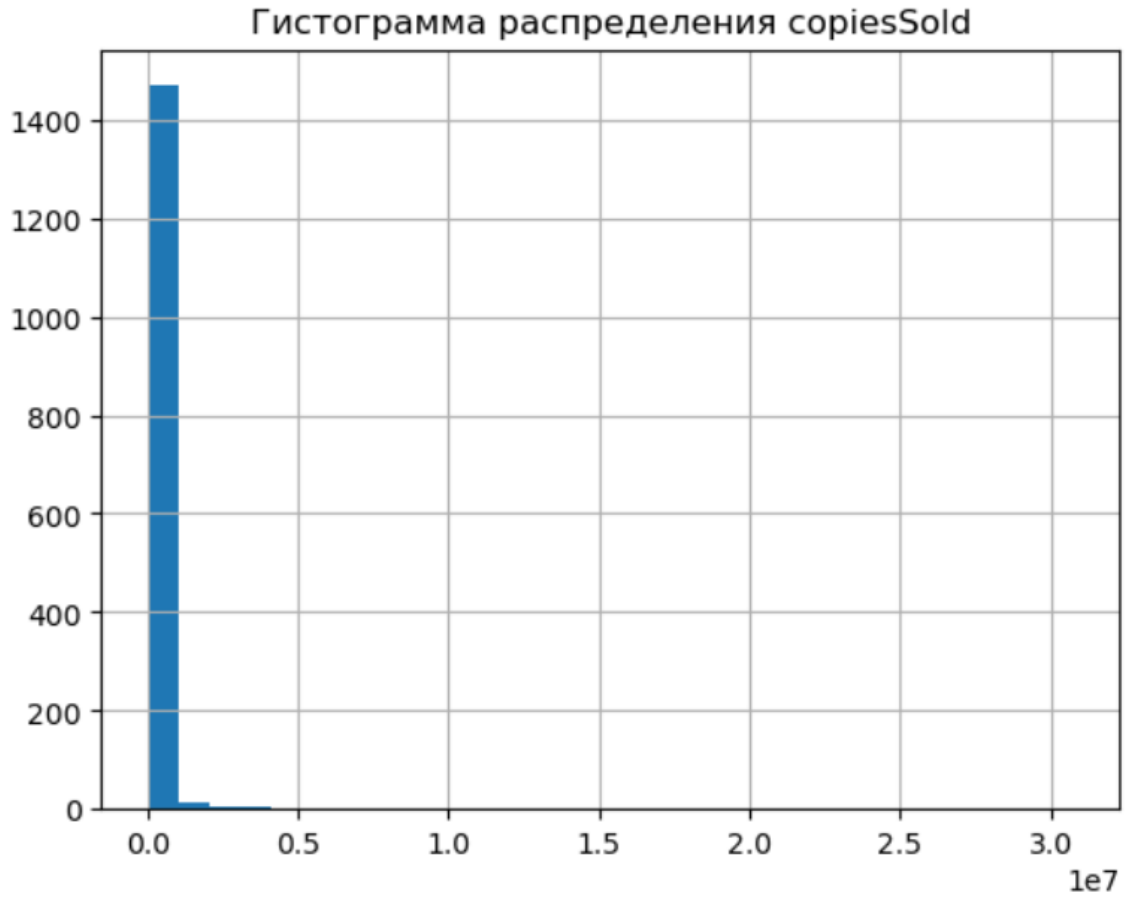


Рис. 1.5 – Гистограмма распределения значений для copiesSold

```
Среднее значение  copiesSold : 141482.57
СКО популярности  copiesSold : 1132379.0120487728
Выбросы:
0         165301
1         159806
2         214192
3         440998
4         141306
          ...
1495      452146
1496     2640903
1497      501474
1498      156339
1499      544144
Name: copiesSold, Length: 201, dtype: int64
Количество выбросов, обнаруженных по методу IQR:  201
Количество пропущенных значений:  0
```

Рис. 1.6 – Гистограмма распределения значений для revenue

```
Среднее значение  revenue : 2632381.9826861774
СКО популярности  revenue : 27800967.996853773
Выбросы:
0       8055097.0
1       7882151.0
2       7815247.0
3       7756399.0
4       7629252.0
          ...
1495    8739530.0
1496    8706135.0
1497    8641459.0
1498    8440898.0
1499    8125042.0
Name: revenue, Length: 223, dtype: float64
Количество выбросов, обнаруженных по методу IQR:  223
Количество пропущенных значений:  0
```

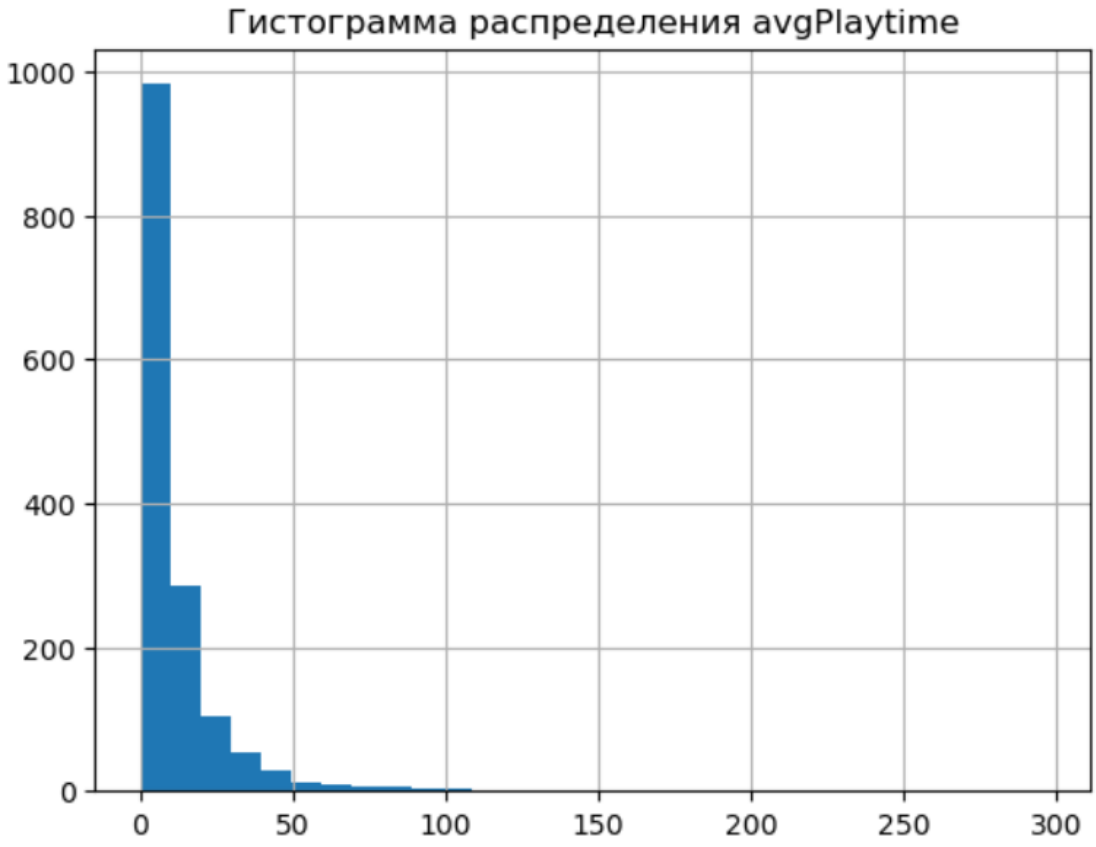


Рис. 1.7 – Гистограмма распределения значений для avgPlaytime

```
Среднее значение  avgPlaytime : 12.56270363676686
СКО популярности  avgPlaytime : 21.534990641302613
Выбросы:
0       42.365140
1       29.651061
4       34.258496
5       95.697813
7       41.418885
          ...
1489    33.544343
1493    91.461899
1494    31.078359
1496    31.974027
1499    30.160995
Name: avgPlaytime, Length: 147, dtype: float64
Количество выбросов, обнаруженных по методу IQR:  147
Количество пропущенных значений:  0
```

Рис. 1.8 – Гистограмма распределения значений для reviewScore

```
Среднее значение  reviewScore : 76.20133333333334
СКО популярности  reviewScore : 24.311330655112695
Выбросы:
10      0
13      0
16      0
34      0
40      0
       ..
1458    0
1464    0
1478    0
1483    0
1491    0
Name: reviewScore, Length: 118, dtype: int64
Количество выбросов, обнаруженных по методу IQR:  118
Количество пропущенных значений:  0
```

Количество пропущенных значений определялось с помощью функций .isnull().sum())

```
print("Количество пропущенных значений: ",df[i].isnull().sum())
```

Рис. 1.9 –Пример кода для поиска пропущенных значений для

Пропущенные значения для каждого атрибута:

− copiesSold : 0

− price : 0

− revenue : 0

− avgPlaytime : 0

− reviewScore : 0

В данном варианте датасета нет пропущенных значений, следовательно нечего обрабатывать. Однако при наличии пропущенных значений их можно было бы удалить с помощью функции .dropna().

3 Часть

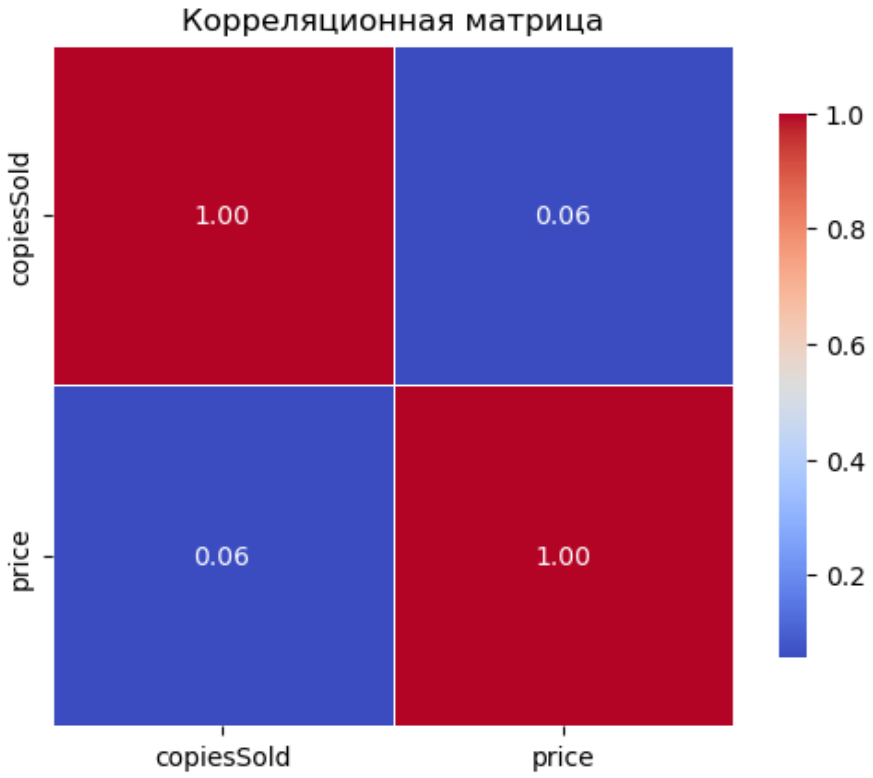
1. Корреляция между параметрами была определена с помощью функции correlation = df[].corr(df[]). Для упрощения понимания была создана и выведена в графическом виде матрица корреляции выбранных атрибутов.

```
colon = ['copiesSold', 'revenue', 'avgPlaytime',  'reviewScore']
for i in colon:
    correlation = df[i].corr(df['price'])
    subset = df[[i, 'price']]
    correlation_matrix = subset.corr()
    print(correlation_matrix)

    sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', square=True, linewidths=.5, cbar_kws={"shrink": .8})
    plt.title('Корреляционная матрица')
    plt.show()
```

Рис. 2.1 – Пример кода для расчета корреляции и вывода в матричном виде атрибута цены к остальным атрибутам

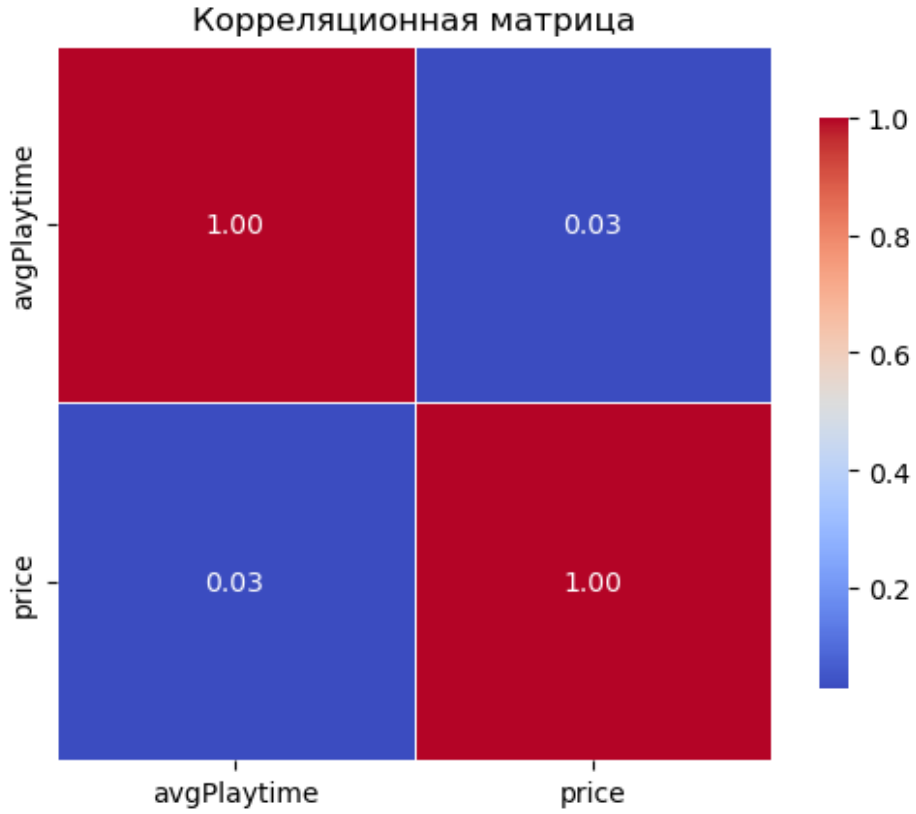
Для данной практической работы были рассмотрены корреляции между ценой игры со ледующими атрибутами: 'copiesSold', 'revenue', 'avgPlaytime', 'reviewScore'.



	revenue	price
revenue	1.000000	0.162521
price	0.162521	1.000000

	avgPlaytime	price
avgPlaytime	1.000000	0.029053
price	0.029053	1.000000

Рис. 2.2 – Корреляция между атрибутами: price и copiesSold, price и revenue



	reviewScore	price
reviewScore	1.000000	-0.035025
price	-0.035025	1.000000

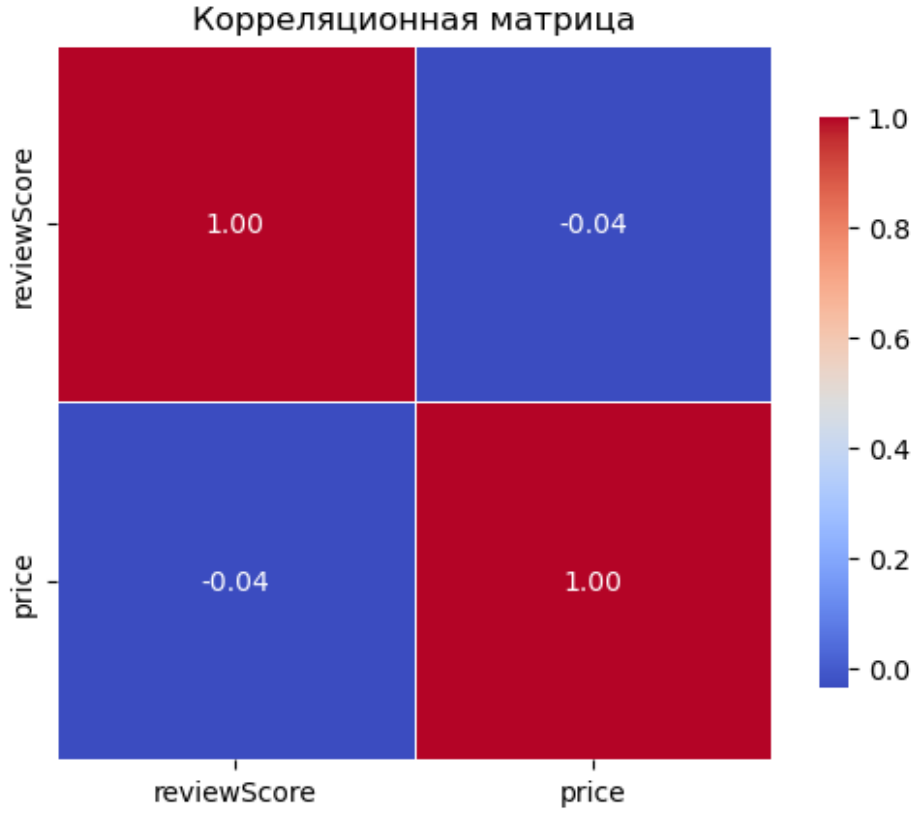


Рис. 2.3 – Корреляция между атрибутами: price и avgPlaytime, price и reviewScore

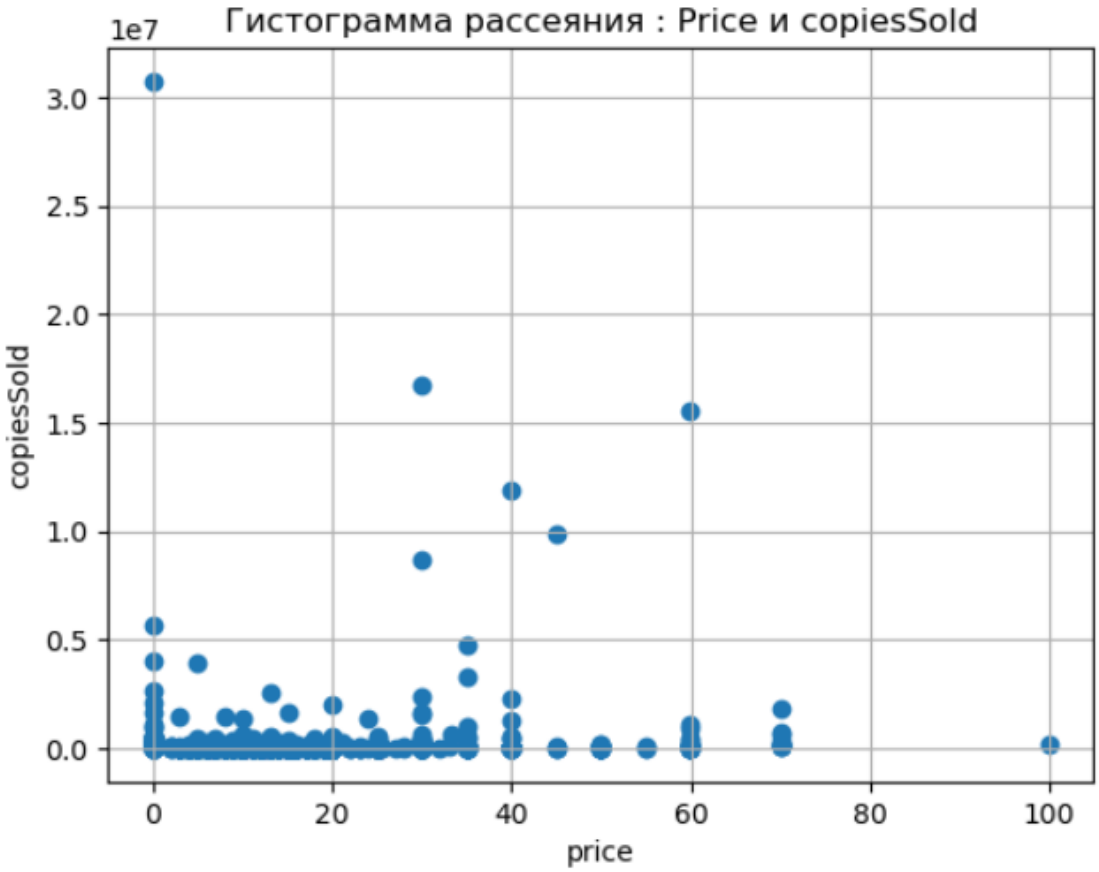
1. Сильной корреляции между всеми парами атрибутов не было не было замечено. Скорее всего на это повлияло наличие нулевых значений в графе цена, а так же видимо цена не сильно влияет на остальные показатели игры.

2. По матрицам корреляции нетрудно заметить, что все представленные атрибуты имеют определенную корреляцию.

3. Были построены графики рассеивания с помощью функции .scatter(). Они были изменены с помощью функций: .xlabel(); .ylabel(); .grid(True); .title().

```
colon = ['copiesSold', 'revenue', 'avgPlaytime',  'reviewScore']
for i in colon:
    plt.scatter(df['price'],df[i]);
    plt.xlabel('price')
    plt.ylabel(i)
    plt.grid(True)
    title = "Гистограмма рассеяния : Price и " + i
    plt.title(title)
    plt.show()
```

Рис. 2.4 – Пример кода для построения графика рассеивания атрибута цены к остальным атрибутам



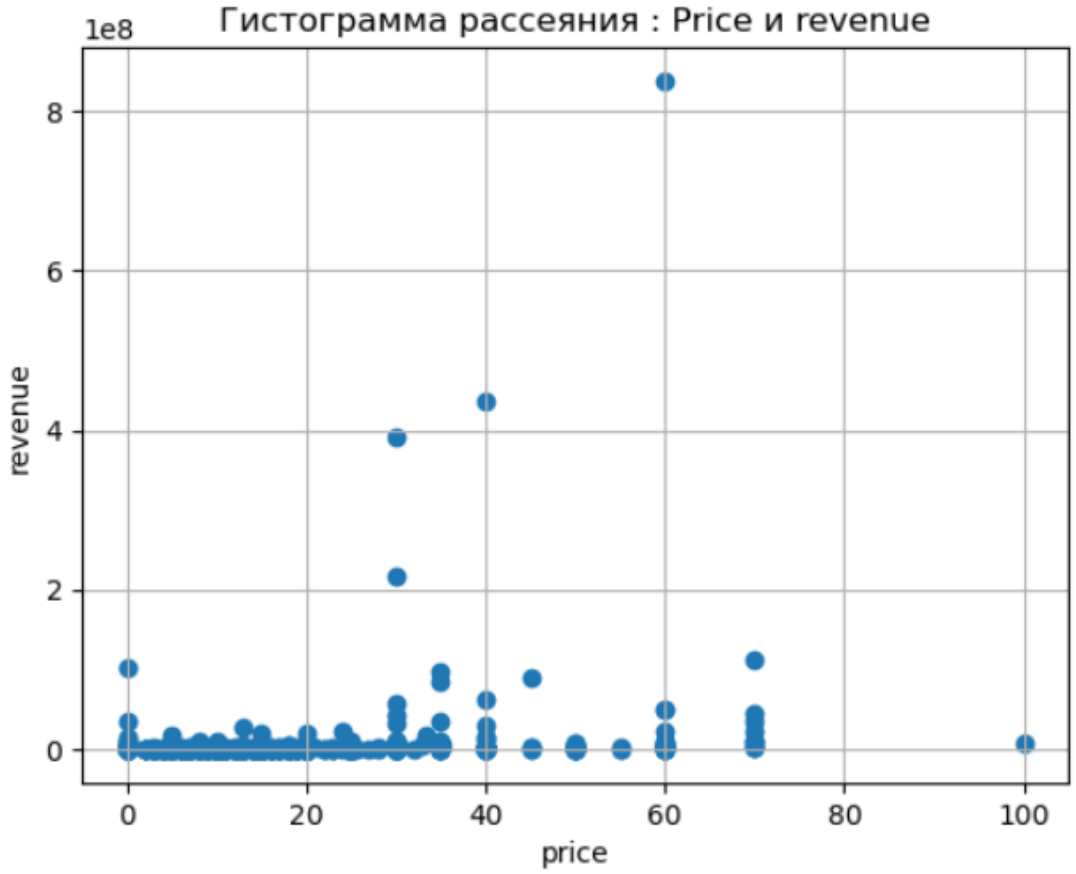


Рис. 2.8 – График рассеяния для атрибутов: price и copiesSold, price и revenue

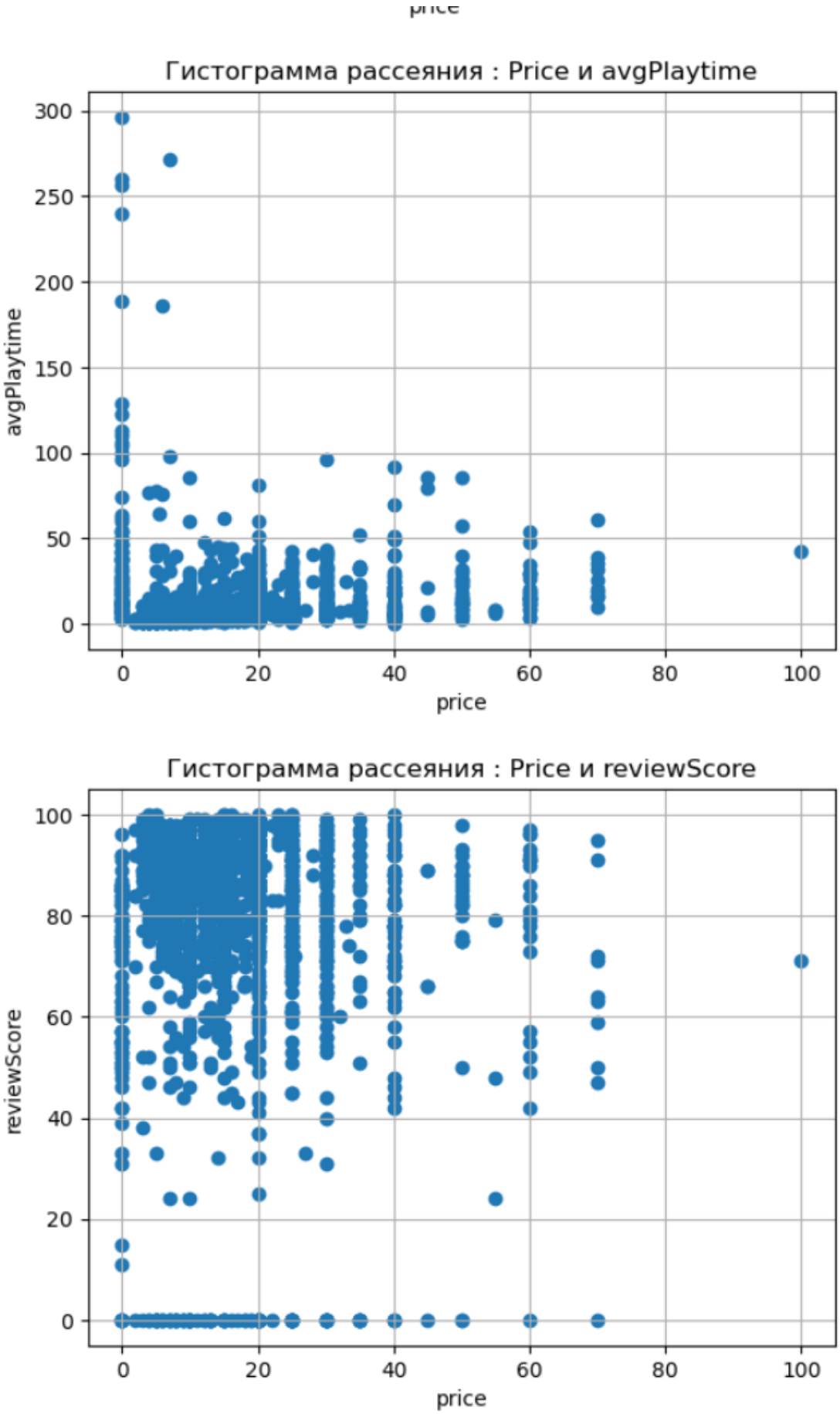


Рис. 2.9 – График рассеяния для атрибутов: price и avgPlaytime, price и reviewScore

ЗАКЛЮЧЕНИЕ

В данной работе были изучены:

- Наборы данных;
- Jupyter Notebook;
- Построение гистограмм на Python;
- Поиск пропущенных значений на Python;
- Определение корреляции на Python;
- Построение графиков рассеивания на Python; Были построены и вычислены:
- среднее значение, ско
- гистограмма распределения значений
- наличие выбросов
- пропущенные значения, сколько

По построенным графикам, гистограммам и матрицам были сделаны и записаны соответствующие выводы.

С помощью написания лабораторной работы стало ясно: как выявить, проанализировать, оформить зависимость и графическое представление данной зависимости между атрибутами. По выбранному датасету можно сказать об определенной степени зависимости популярности языков программирования, основанной на возможности совместного использования нескольких языков для одного проекта или насколько выбранный язык программирования подходит как для конкретной нишевой задачи, так и для разнообразных задач. Именно эти факторы и определяют популярность языка программирования.

Для выполнения практической работы была использована программа Jupyter-ноутбук; язык программирования Python; библиотеки: matplotlib, numpy, pandas.

Были изучены ссылки, представленные в методических указаниях к лабораторной работе.

Написание практической работы помогло: усвоить информацию о наборах данных и взаимодействиями с ними, разобраться с необходимыми библиотеками для построения графиков, гистограмм и матриц на Python.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. https://www.analyticsvidhya.com/blog/2021/02/an-intuitive-guide-tovisualization-in-python/
2. https://stepik.org/lesson/8086/step/1?unit=1365
3. https://medium.com/swlh/identify-outliers-with-pandas-statsmodels-andseaborn-2766103bf67c