Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Рубежный контроль №1
по дисциплине
«Методы машинного обучения»
на тему
«Методы обработки данных»

Выполнил:
студент группы ИУ5-21М
Попова И.А.

Москва — 2020 г.

РК №1

ИУ5-21М Попова И.А.

Тема: Методы обработки данных.

Номер варианта: 8

Номер задачи: 1

Номер набора данных, указанного в задаче: 8

Задача №1.

Для заданного набора данных постройте основные графики, входящие в этап разведочного анализа данных с использованием библиотек Matplotlib и Seaborn. В случае наличия пропусков в данных удалите строки или колонки, содержащие пропуски. Какие графики Вы построили и почему? Какие выводы о наборе данных Вы можете сделать на основании построенных графиков? Проведите корреляционный анализ. Сделайте выводы о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.

Для студентов групп ИУ5-21М, ИУ5И-21М - для пары произвольных колонок данных построить график "Диаграмма рассеяния".

Примечание: ответы на вопросы смотрите ниже.

Загрузка данных

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [2]:

```
data = pd.read_csv('googleplaystore.csv', sep=",")
```

In [3]:

```
data.head()
```

Out[3]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone

In [4]:

```
data.shape
```

Out[4]:

```
(10841, 13)
```

In [5]:

```
data.columns
```

Out[5]:

```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
       'Android Ver'],
      dtype='object')
```

In [6]:

```
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
App - 0
Category - 0
Rating - 1474
Reviews - 0
Size - 0
Installs - 0
Type - 1
Price - 0
Content Rating - 1
Genres - 0
Last Updated - 0
Current Ver - 8
Android Ver - 3
```

Так как данные содержат пропуски, в таком случае удалим колонки, содержащие пропуски.

In [7]:

```
data = data.dropna(axis='columns')
```

In [8]:

```
data.isnull().sum()
```

Out[8]:

```
App             0
Category        0
Reviews         0
Size            0
Installs        0
Price           0
Genres          0
Last Updated    0
dtype: int64
```

In [9]:

```
data.drop_duplicates(subset='App', inplace=True)
```

In [10]:

```
print('Number of apps in the dataset : ' , len(data))
data.sample(7)
```

```
Number of apps in the dataset :  9660
```

Out[10]:

	App	Category	Reviews	Size	Installs	Price	G
8336	DF-Server Mobile	PRODUCTIVITY	17	76M	100+	0	Produ
8719	DRAGON QUEST VIII	FAMILY	7812	27M	50,000+	$19.99	Role F
2078	Dr. Panda Town: Vacation	FAMILY	10366	78M	1,000,000+	0	Education;P
3224	GPS Status & Toolbox	TRAVEL_AND_LOCAL	149723	4.1M	10,000,000+	0	Travel &
8920	La citadelle du musulman	BOOKS_AND_REFERENCE	314	9.8M	50,000+	0	Bo Refe
10596	Free Florida DMV Test 2018	FAMILY	665	5.3M	50,000+	0	Edu
6280	BI Mobile	FINANCE	337	3.3M	10,000+	0	F

Cleaning data

In [11]:

```
data = data[data['Installs'] != 'Free']
data = data[data['Installs'] != 'Paid']

# - Installs : Remove + and ,
data['Installs'] = data['Installs'].apply(lambda x: x.replace('+', '') if '+' in str(x)
else x)
data['Installs'] = data['Installs'].apply(lambda x: x.replace(',', '') if ',' in str(x)
else x)
data['Installs'] = data['Installs'].apply(lambda x: int(x))
```

In [12]:

```
# - Size : Remove 'M', Replace 'k' and divide by 10^-3
#df['Size'] = df['Size'].fillna(0)

data['Size'] = data['Size'].apply(lambda x: str(x).replace('Varies with device', 'NaN') if 'Varies with device' in str(x) else x)

data['Size'] = data['Size'].apply(lambda x: str(x).replace('M', '') if 'M' in str(x) else x)
data['Size'] = data['Size'].apply(lambda x: str(x).replace(',', '') if 'M' in str(x) else x)
data['Size'] = data['Size'].apply(lambda x: float(str(x).replace('k', '')) / 1000 if 'k' in str(x) else x)


data['Size'] = data['Size'].apply(lambda x: float(x))
data['Installs'] = data['Installs'].apply(lambda x: float(x))

data['Price'] = data['Price'].apply(lambda x: str(x).replace('$', '') if '$' in str(x) else str(x))
data['Price'] = data['Price'].apply(lambda x: float(x))

data['Reviews'] = data['Reviews'].apply(lambda x: int(x))
```

In [13]:

```
data.head()
```

Out[13]:

	App	Category	Reviews	Size	Installs	Price	Genres	Last Updated
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	159	19.0	10000.0	0.0	Art & Design	January 7, 2018
1	Coloring book moana	ART_AND_DESIGN	967	14.0	500000.0	0.0	Art & Design;Pretend Play	January 15, 2018
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	87510	8.7	5000000.0	0.0	Art & Design	August 1, 2018
3	Sketch - Draw & Paint	ART_AND_DESIGN	215644	25.0	50000000.0	0.0	Art & Design	June 8 2018
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	967	2.8	100000.0	0.0	Art & Design;Creativity	June 20 2018

In [14]:

```
data.dtypes
```

Out[14]:

```
App              object
Category         object
Reviews           int64
Size            float64
Installs        float64
Price           float64
Genres           object
Last Updated     object
dtype: object
```

In [15]:

```
# Основные статистические характеристки набора данных
data.describe()
```

Out[15]:

	Reviews	Size	Installs	Price
count	9.659000e+03	8432.000000	9.659000e+03	9659.000000
mean	2.165926e+05	20.395289	7.777507e+06	1.099299
std	1.831320e+06	21.827542	5.375828e+07	16.852152
min	0.000000e+00	0.008500	0.000000e+00	0.000000
25%	2.500000e+01	4.600000	1.000000e+03	0.000000
50%	9.670000e+02	12.000000	1.000000e+05	0.000000
75%	2.940100e+04	28.000000	1.000000e+06	0.000000
max	7.815831e+07	100.000000	1.000000e+09	400.000000

Диаграмма рассеяния

Позволяет построить распределение двух колонок данных и визуально обнаружить наличие зависимости. Не предполагается, что значения упорядочены (например, по времени).

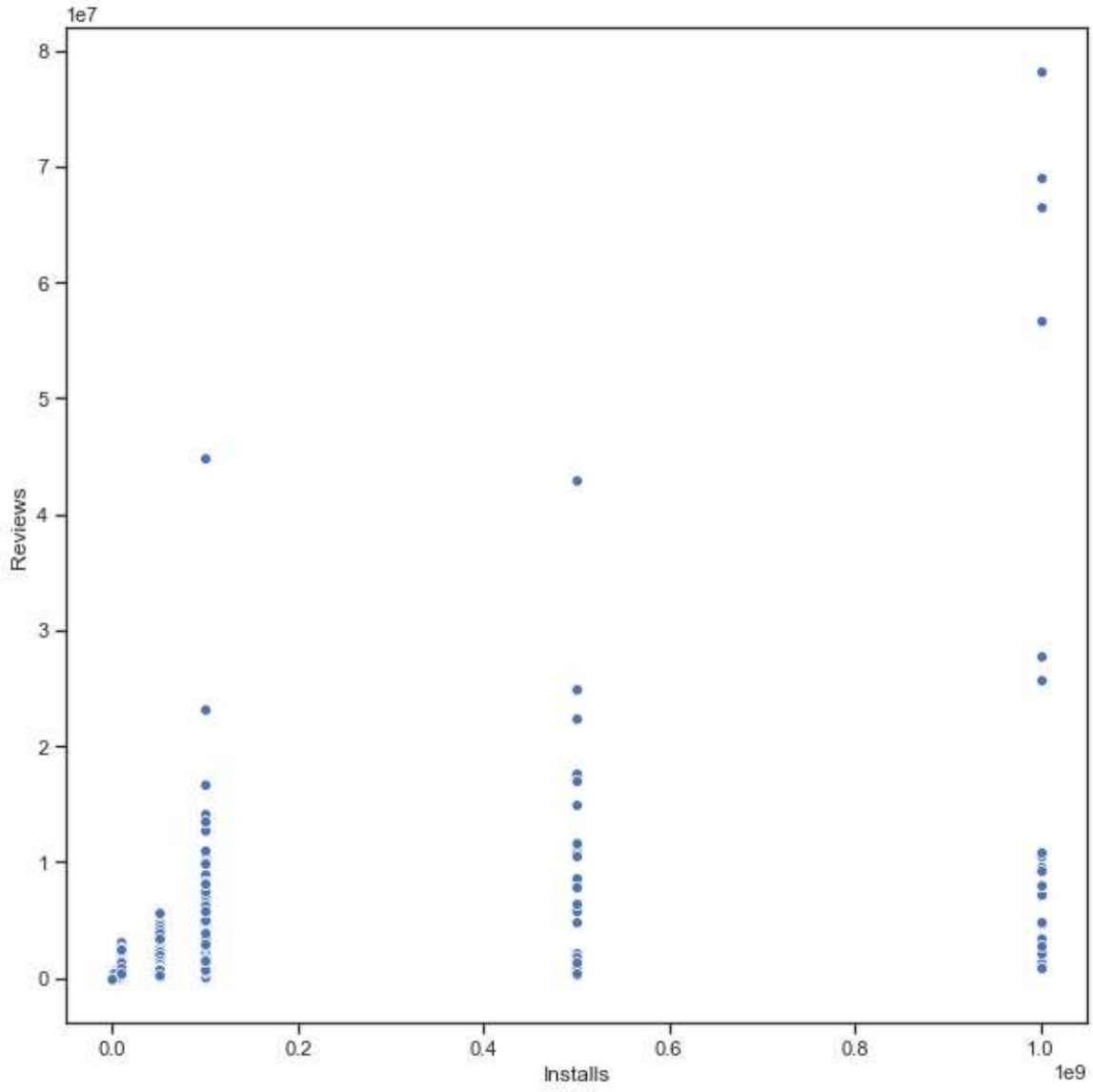
Построим диаграмму рассеяния для двух признаков - Installs и Reviews. По графику видно, что количество отзывов о приложении влияет на число установок. Но также распространены случаи, когда большое число установок приходится на приложения с небольшим количеством отзывов.

In [16]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Installs', y='Reviews', data=data)
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2112e4b7a58>
```

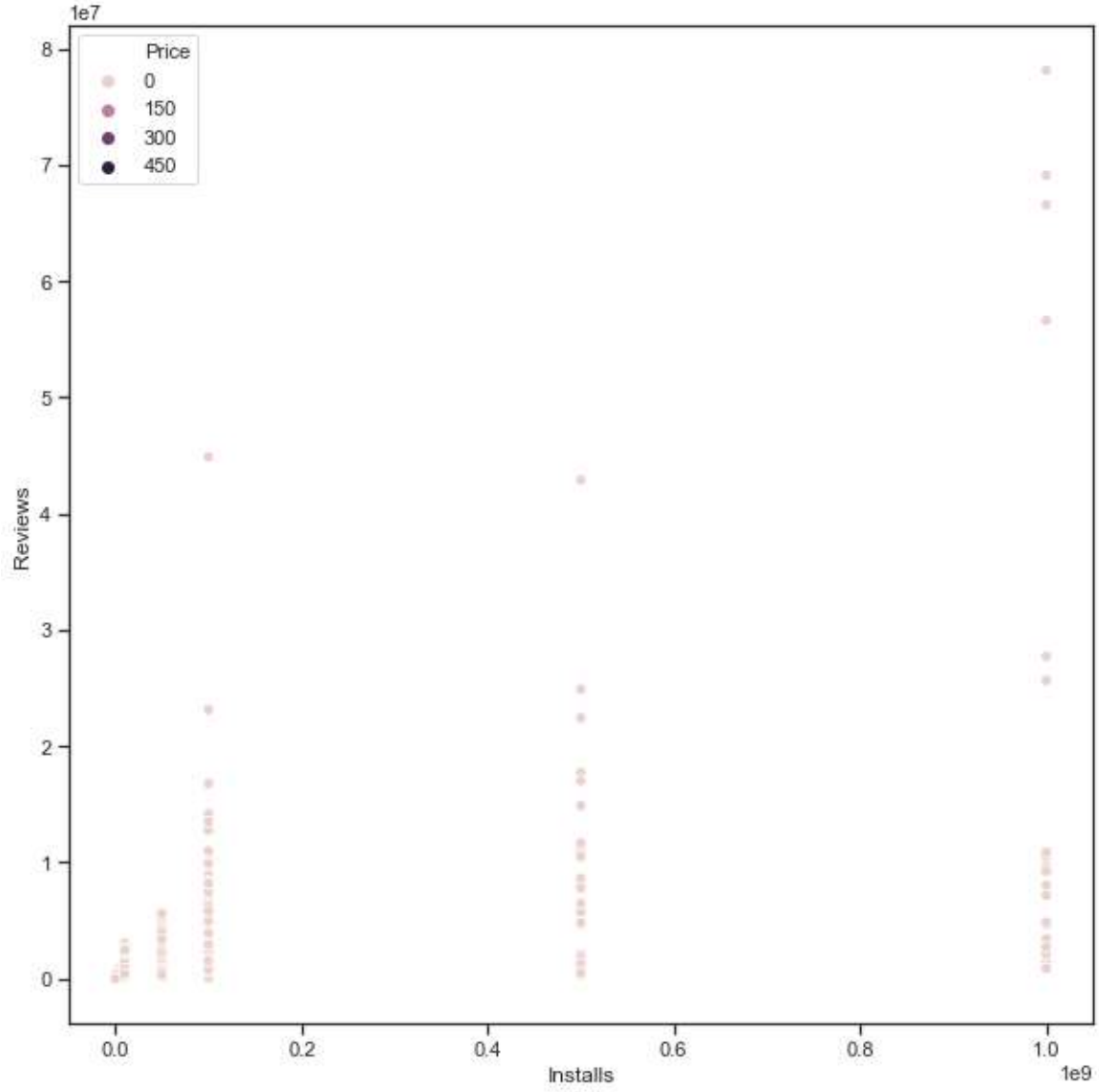


In [17]:

```
# Посмотрим насколько на эту зависимость влияет целевой признак.
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Installs', y='Reviews', data=data, hue='Price')
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21132093ac8>
```



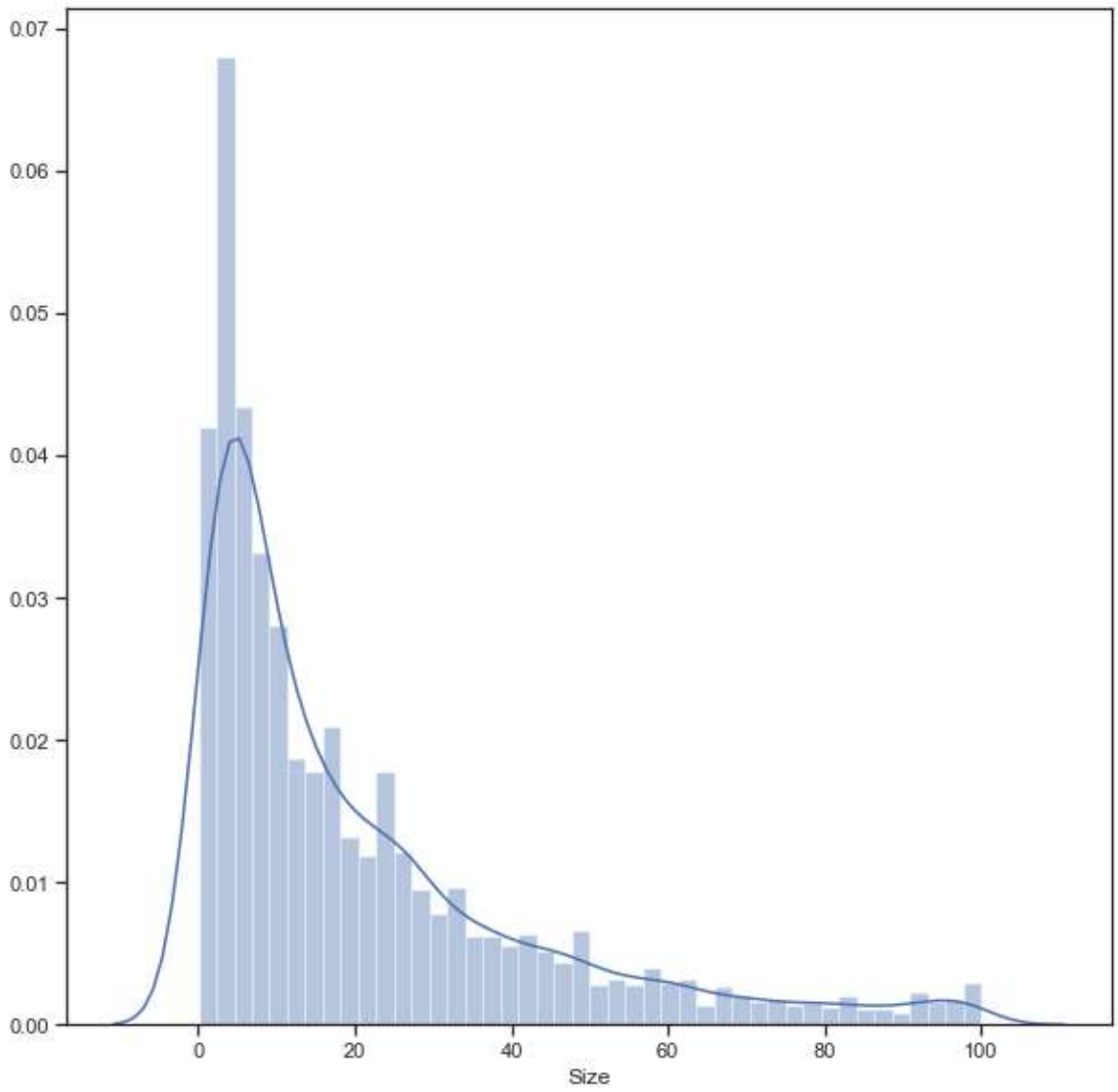
Гистограмма

In [18]:

```
# Определение наиболее вероятного значения признака Size
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['Size'])
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21131ee9da0>
```



In [19]:

```
data['Size'].median()
```

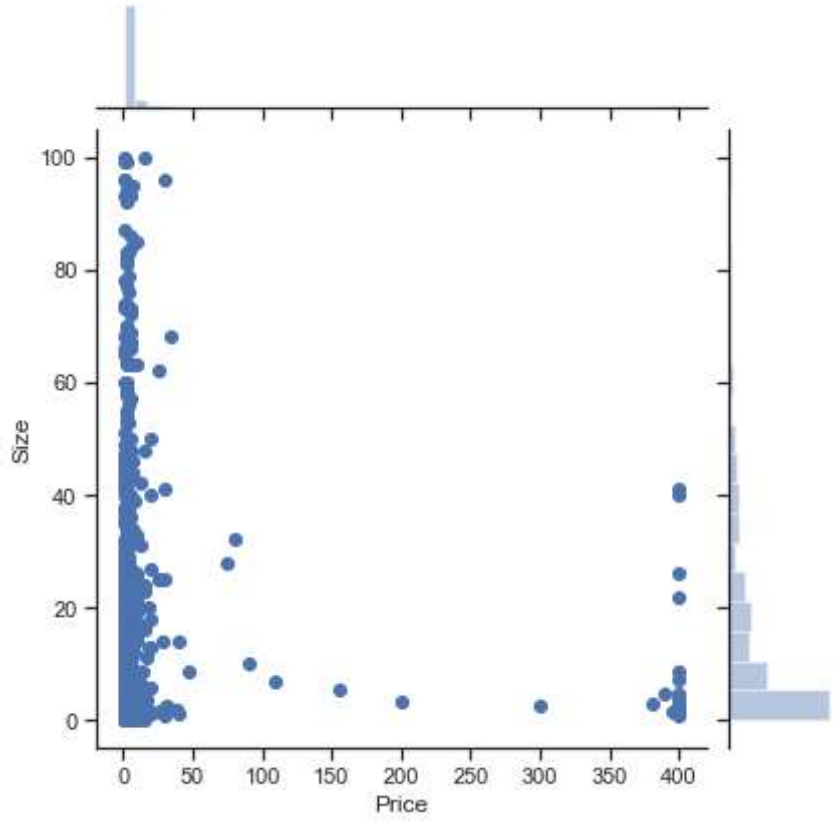
Out[19]:

```
12.0
```

Joinplot

In [20]:

```
# joinplot отображает зависимость цены от размера приложения
paid_apps = data[data.Price > 0]
p = sns.jointplot( "Price", "Size", paid_apps)
```



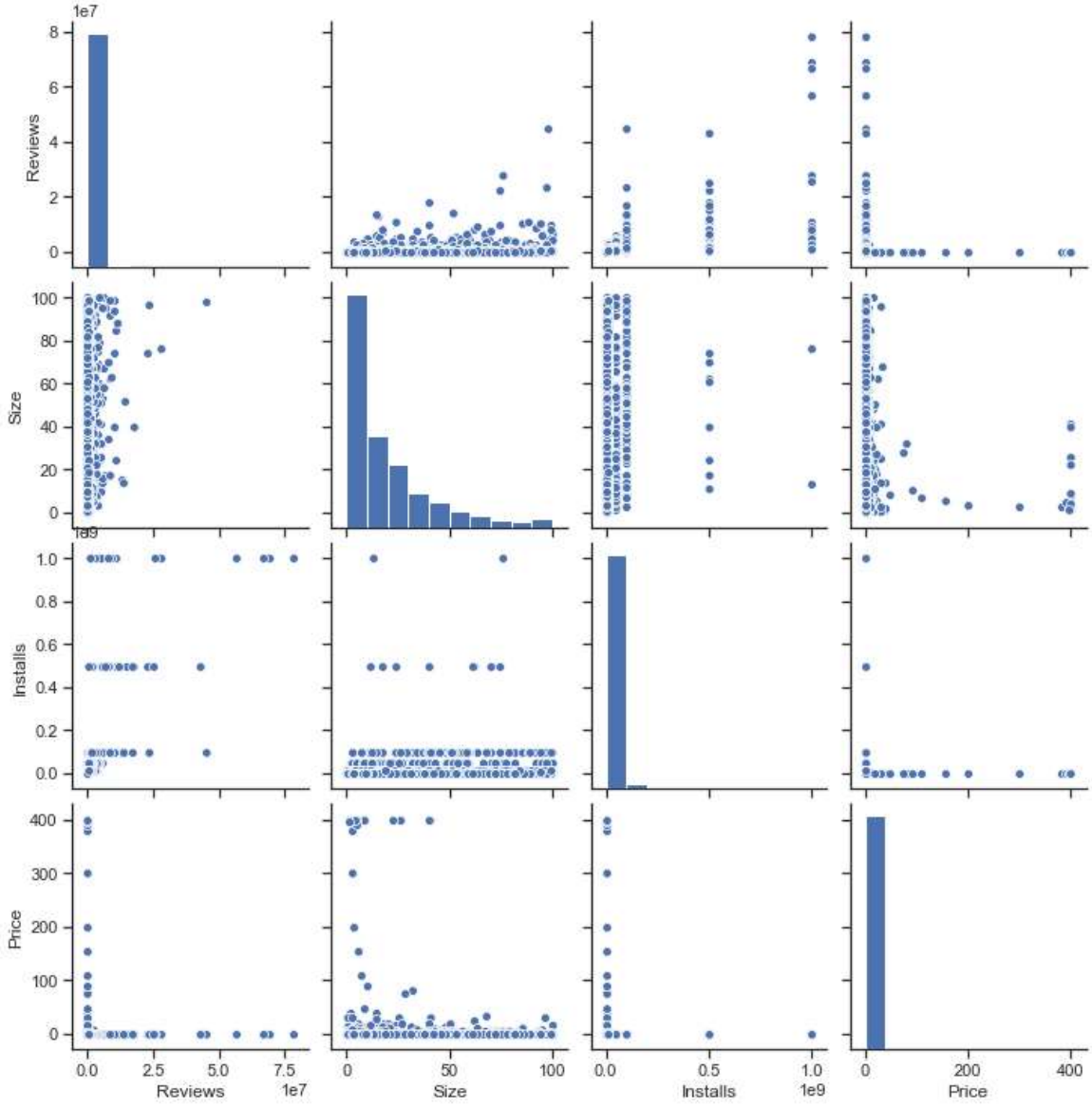
Парные диаграммы

In [21]:

```
# Парные диаграммы по признакам датасета
sns.pairplot(data)
```

Out[21]:

```
<seaborn.axisgrid.PairGrid at 0x21134657550>
```



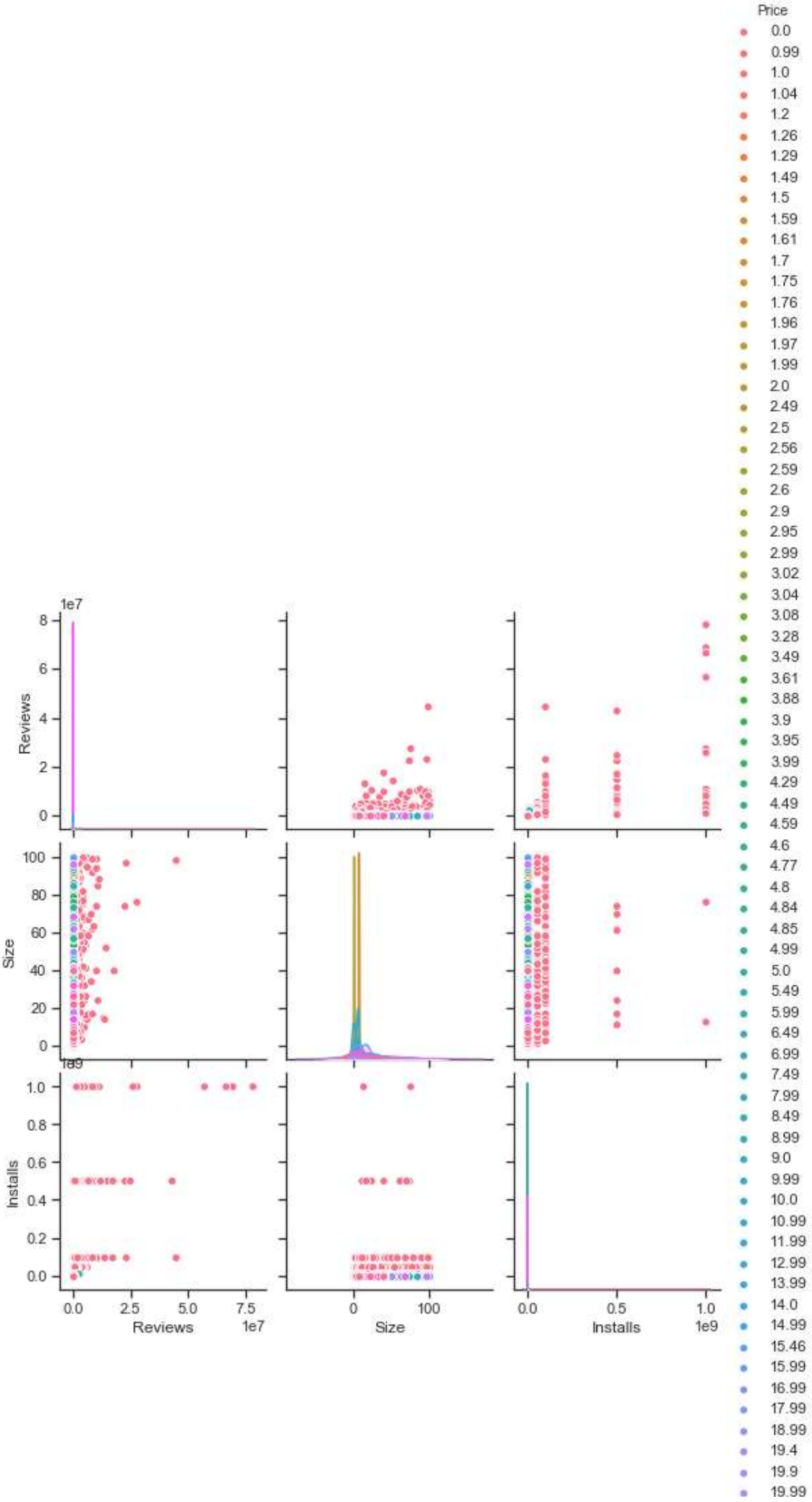
In [22]:

```
# Группировка по значениям признака Price
sns.pairplot(data, hue="Price")
```

```
c:\users\innap\miniconda3\lib\site-packages\seaborn\distributions.py:288:
UserWarning: Data must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
```

Out[22]:

```
<seaborn.axisgrid.PairGrid at 0x21134e91e10>
```


Price
0.0
0.99
1.0
1.04
1.2
1.26
1.29
1.49
1.5
1.59
1.61
1.7
1.75
1.76
1.96
1.97
1.99
2.0
2.49
2.5
2.56
2.59
2.6
2.9
2.95
2.99
3.02
3.04
3.08
3.28
3.49
3.61
3.88
3.9
3.95
3.99
4.29
4.49
4.59
4.6
4.77
4.8
4.84
4.85
4.99
5.0
5.49
5.99
6.49
6.99
7.49
7.99
8.49
8.99
9.0
9.99
10.0
10.99
11.99
12.99
13.99
14.0
14.99
15.46
15.99
16.99
17.99
18.99
19.4
19.9
19.99
1e7
Reviews
Size
1e9
Installs

- 24.99
- 25.99
- 28.99
- 29.99
- 30.99
- 33.99
- 37.99
- 39.99
- 46.99
- 74.99
- 79.99
- 89.99
- 109.99
- 154.99
- 200.0
- 299.99
- 379.99
- 389.99
- 394.99
- 399.99
- 400.0

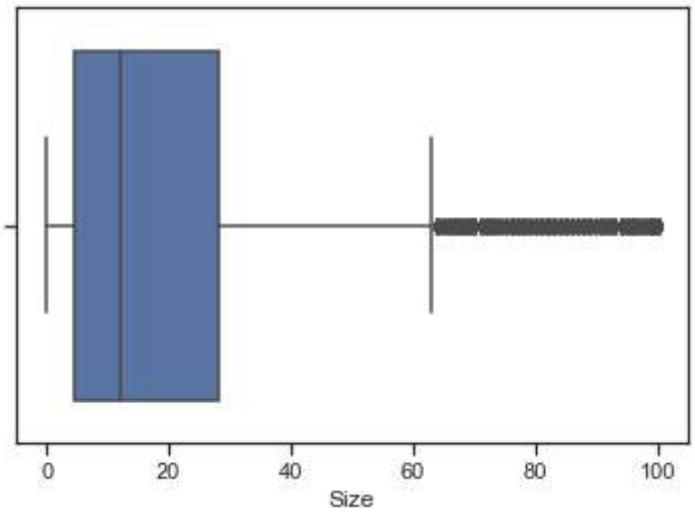
Ящик с усами

In [23]:

```
# Одномерное распределение вероятности
sns.boxplot(x=data['Size'])
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21137548cf8>
```

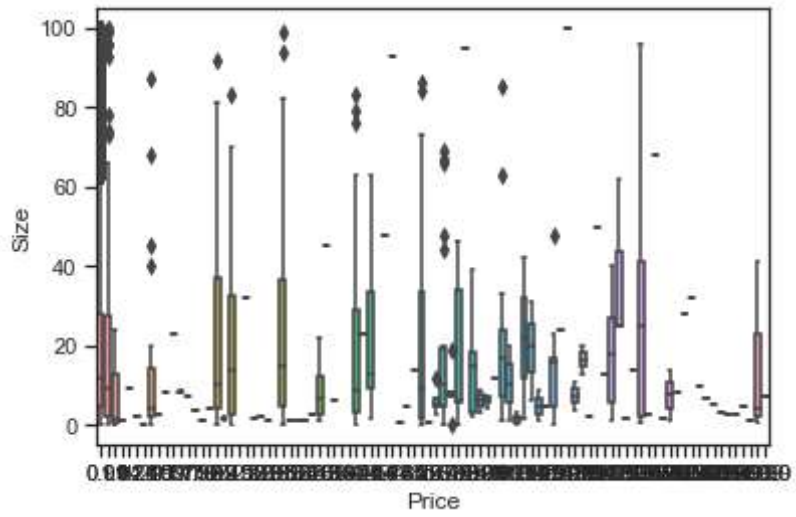


In [24]:

```
# Распределение параметра Size сгруппированные по Price.
sns.boxplot(x='Price', y='Size', data=data)
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21136c7b7b8>
```



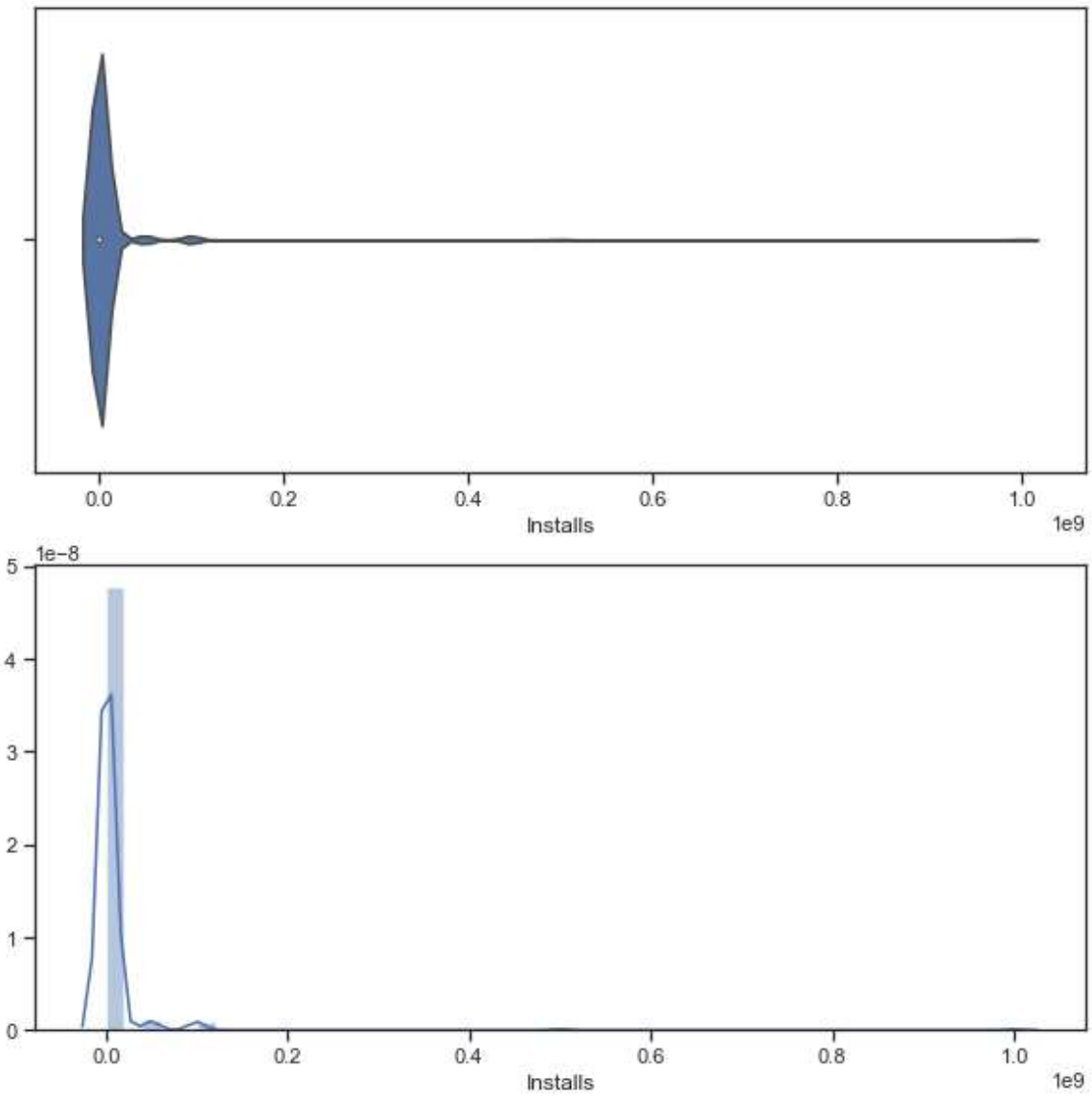
Violin plot

In [25]:

```
# По краям графика отображаются распределения плотности признака Installs
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['Installs'])
sns.distplot(data['Installs'], ax=ax[1])
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21137c3ccc0>
```

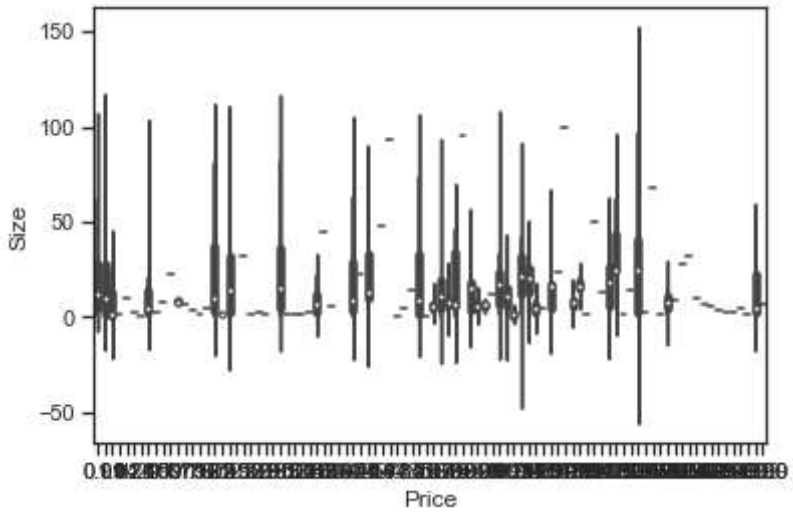


In [26]:

```
# Распределение параметра Size сгруппированные по Price.
sns.violinplot(x='Price', y='Size', data=data)
```

Out[26]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21137cfc898>
```



Корреляционный анализ

In [27]:

```
data.corr()
```

Out[27]:

	Reviews	Size	Installs	Price
Reviews	1.000000	0.179321	0.625165	-0.007598
Size	0.179321	1.000000	0.134291	-0.022439
Installs	0.625165	0.134291	1.000000	-0.009405
Price	-0.007598	-0.022439	-0.009405	1.000000

In [28]:

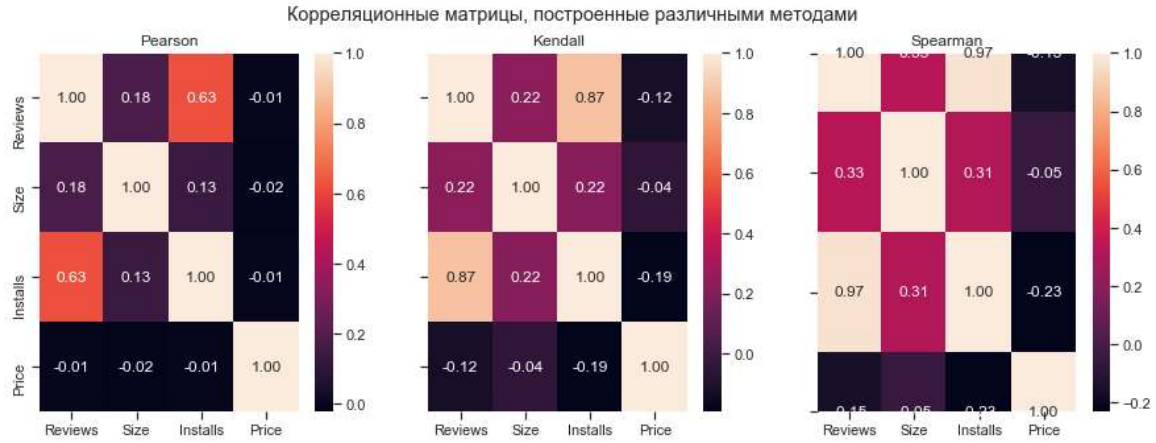
```
data.corr(method='spearman')
```

Out[28]:

	Reviews	Size	Installs	Price
Reviews	1.000000	0.329949	0.967707	-0.150713
Size	0.329949	1.000000	0.310458	-0.049035
Installs	0.967707	0.310458	1.000000	-0.232029
Price	-0.150713	-0.049035	-0.232029	1.000000

In [29]:

```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(15,5))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```



Вывод

Проведите корреляционный анализ. Сделайте выводы о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.

Корреляционная матрица содержит коэффициенты корреляции между всеми парами признаков.

На основе корреляционной матрицы можно сделать следующие выводы:

1. Целевой признак Price слишком слабо коррелирует с остальными признаками. Больше всего с признаком Size(-0.05 Spearman), что достаточно логично, так как на цену влияет сложность приложения. Данный признак можно оставить в модели.
2. Признаки Installs и Reviews коррелируют друг с другом. Можно оставить в модели один из этих признаков, к примеру, Reviews. Этот признак сильнее коррелирован с целевым признаком.

Какие графики Вы построили и почему?

В ходе выполнения РК1 был проведен разведочный анализ данных приложений с Google Play Store. Были исследованы основные характеристики датасета, а также проведено визуальное исследование данных в результате которого были построены графики: диаграмма рассеяния, гистограммы распределения, joinplot(Комбинация гистограмм и диаграмм рассеивания), парные диаграммы, диаграмма "ящик с усами" и графики violin plot.

Диаграмма рассеивания позволяет построить распределение двух колонок данных и визуально обнаружить наличие зависимости. В данном случае исследовалась взаимосвязь между признаками - Installs и Reviews, чтобы определить влияние числа отзывов на количество установок приложения.

Гистограмма распределения позволяет оценить плотность вероятности распределения данных. При помощи гистограммы было исследовано распределение признака Size(размер приложения). По гистограмме частот можно предположить, что признак описывается законом, близким к нормальному, и имеет наиболее вероятное значение, лежащее в пределах 12-20MB.

Joinplot - комбинация гистограмм и диаграмм рассеивания. С помощью этой гистограммы исследовалась взаимосвязь между признаками - Installs и Reviews. По графику видно, что количество отзывов о приложении влияет на число установок. Но также распространены случаи, когда большое число установок приходится на приложения с небольшим количеством отзывов.

Парные диаграммы представляют комбинацию гистограмм и диаграмм рассеивания для всего набора данных. Вывод содержит множество диаграмм рассеивания и гистограмм распределения по каждой паре признаков. Таким образом парная диаграмма обобщает все ранее построенные графики.

Диаграмма "ящик с усами" показывает одномерное распределение вероятности. Построен график по признаку Size(размер приложения). На графике показаны наблюдаемый минимум - 0MB, максимум - 60MB, нижний квартиль - примерно 5MB, верхний квартиль - 25MB, медиану - 12MB и выбросы - более 60MB.

На графиках violin plot по краям отображаются распределения плотности. При помощи данного вида графиков исследовался признак Installs(число установок). Вместе с гистограммой график показывает, что наибольшее значение вероятоности приходится примерно на 0,1 * 10^6 установок.

Какие выводы о наборе данных Вы можете сделать на основании построенных графиков?

На основании построенных графиков можно сделать вывод о том, что пользователи отдают предпочтение наиболее легким приложениям(требующего небольшого объема памяти - Size). Чем ниже цена, тем более чаще приложение скачивают пользователи. Приложения с низкой стоимостью(либо бесплатные) имеют больше отзывов. Одними из самых дорогих приложений являются приложения категории Medical и Family, однако присутствуют выбросы, где цена достаточно высокая для приложений категории Game, Lifestyle, Finance. Также размер приложения влияет на цену, что обусловлено сложностью реализации.

In [ ]: