



**Министерство науки и высшего образования Российской
Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ

«Радиотехнический»

КАФЕДРА

ИУ-5 «Системы обработки информации и управления»

**Рубежный контроль №1 по курсу
Технологии машинного обучения**

**Тема работы: "Технологии разведочного анализа и обработки
данных."**

Выполнил: Лисин А. В.
Группа: РТ5-61Б

Дата
выполнения: «__» _____ 2021 г.

Подпись: _____

Проверил:

Дата
проверки: «__» _____ 2021 г.

Подпись: _____

Москва, 2021 г.

```
In [1]: 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.shape

Out[3]: (10841, 13)

In [4]: data.dtypes

Out[4]: App                object
Category                object
Rating                 float64
Reviews                object
Size                   object
Installs                object
Type                   object
Price                  object
Content Rating          object
Genres                  object
Last Updated            object
Current Ver             object
Android Ver             object
dtype: object
```

Как мы можем видеть, все значения кроме рейтинга являются объектами, хотя не должны (из-за миллионов в виде буквы М, запятых, разделяющих тысячи и плюсов в загрузках), что мы должны преобразовать.

```
In [5]: data.isnull().sum()

Out[5]: 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
dtype: int64

In [6]: for i in range(len(data['Reviews'])):
if data['Reviews'][i][-1] == 'M':
data['Reviews'][i] = str(int(float(data['Reviews'][i][0:-1])) * 1000000)

<ipython-input-6-28e3fda95573>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data['Reviews'][i] = str(int(float(data['Reviews'][i][0:-1])) * 1000000)

In [7]: data['Reviews'] = data['Reviews'].astype(str).astype(int)

In [8]: for i in range(len(data['Installs'])):
if data['Installs'][i] == 'Free':
data['Installs'][i] = 0
continue
data['Installs'][i] = data['Installs'][i].replace(",","")
data['Installs'][i] = data['Installs'][i].replace("+","")
data['Installs'][i] = str(int(float(data['Installs'][i])))

<ipython-input-8-9c6f0fc030bc>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data['Installs'][i] = data['Installs'][i].replace(",","")
<ipython-input-8-9c6f0fc030bc>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data['Installs'][i] = data['Installs'][i].replace("+","")
<ipython-input-8-9c6f0fc030bc>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data['Installs'][i] = str(int(float(data['Installs'][i])))
<ipython-input-8-9c6f0fc030bc>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data['Installs'][i] = 0

In [9]: data['Installs'] = data['Installs'].astype(str).astype(int)

In [10]: data.dtypes

Out[10]: App                object
Category                object
Rating                 float64
Reviews                int64
Size                   object
Installs                int64
Type                   object
Price                  object
Content Rating          object
Genres                  object
Last Updated            object
Current Ver             object
Android Ver             object
dtype: object

In [11]: data.head()
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10000	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500000	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5000000	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50000000	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100000	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

```
In [12]: data = data.dropna(axis=0, how='any')

In [13]: data.isnull().sum()

Out[13]: App                0
Category                0
Rating                 0
Reviews                0
Size                   0
Installs                0
Type                   0
Price                  0
Content Rating          0
Genres                  0
Last Updated            0
Current Ver             0
Android Ver             0
dtype: int64

In [14]: data.describe()

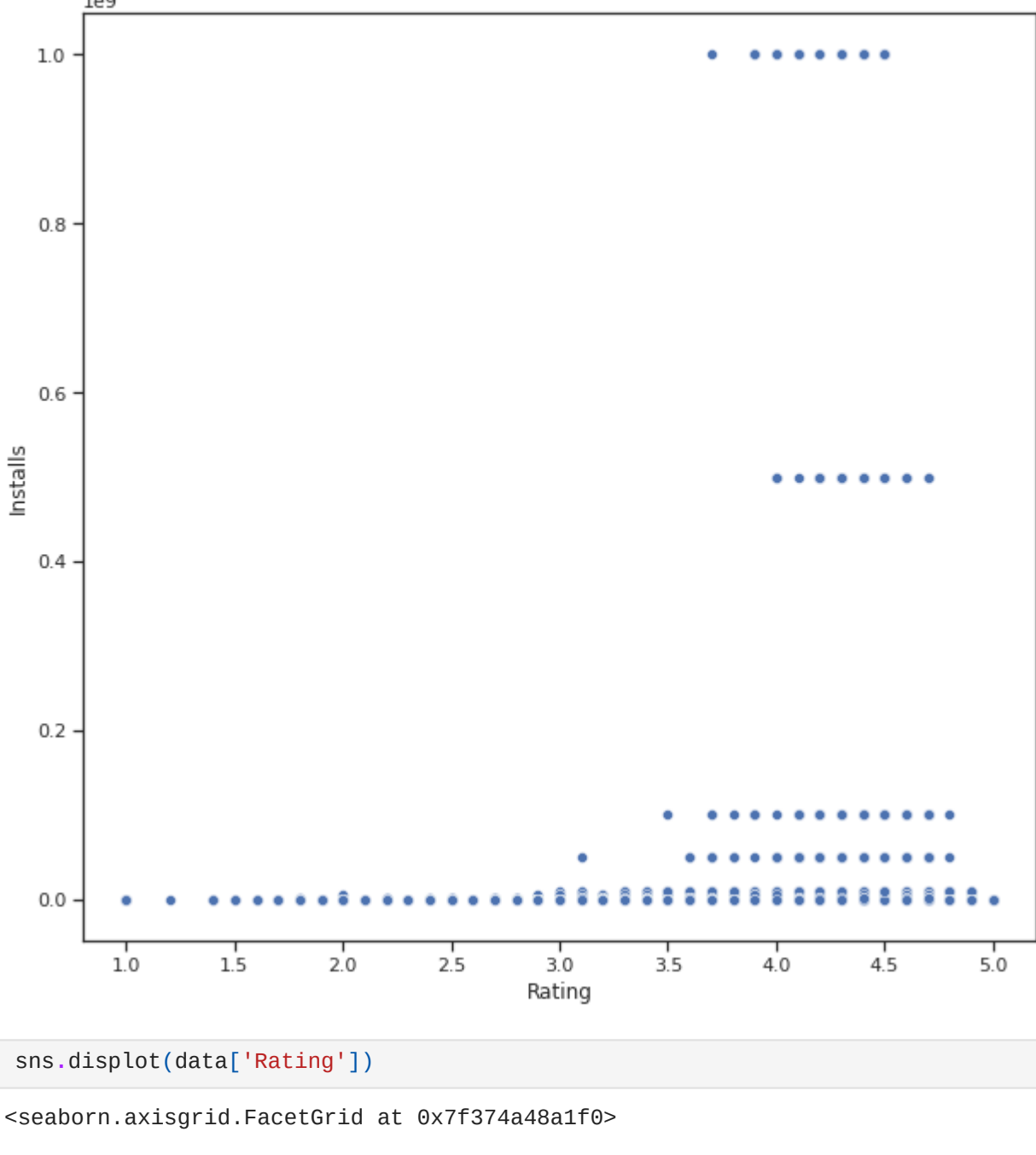
Out[14]:
```

	Rating	Reviews	Installs
count	9360.000000	9.360000e+03	9.360000e+03
mean	4.191838	5.143767e+05	1.790875e+07
std	0.515263	3.145023e+06	9.126637e+07
min	1.000000	1.000000e+00	1.000000e+00
25%	4.000000	1.867500e+02	1.000000e+04
50%	4.300000	5.955000e+03	5.000000e+05
75%	4.500000	8.162750e+04	5.000000e+06
max	5.000000	7.815831e+07	1.000000e+09

Как мы видим теперь основные значения, а именно ревью и загрузки в правильном виде, а значит можем перейти к корреляционному анализу.

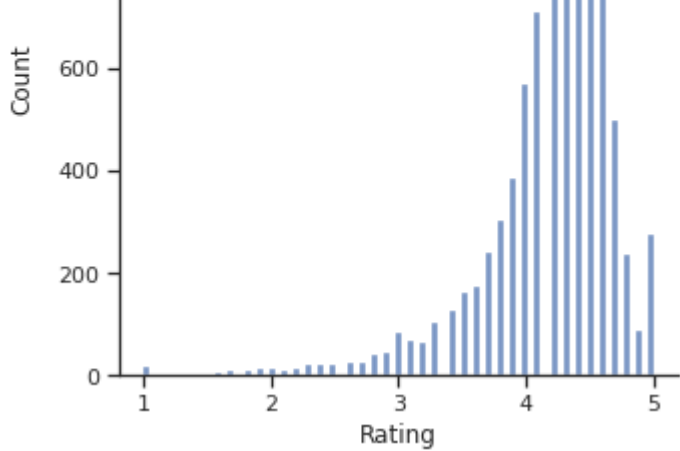
```
In [15]: fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Rating', y='Installs', data=data)

Out[15]: <AxesSubplot:xlabel='Rating', ylabel='Installs'>
```



```
In [16]: sns.displot(data['Rating'])

Out[16]: <seaborn.axisgrid.FacetGrid at 0x7f374a48a1f0>
```



```
In [17]: data.corr()

Out[17]:
```

	Rating	Reviews	Installs
Rating	1.000000	0.068133	0.051337
Reviews	0.068133	1.000000	0.641605
Installs	0.051337	0.641605	1.000000

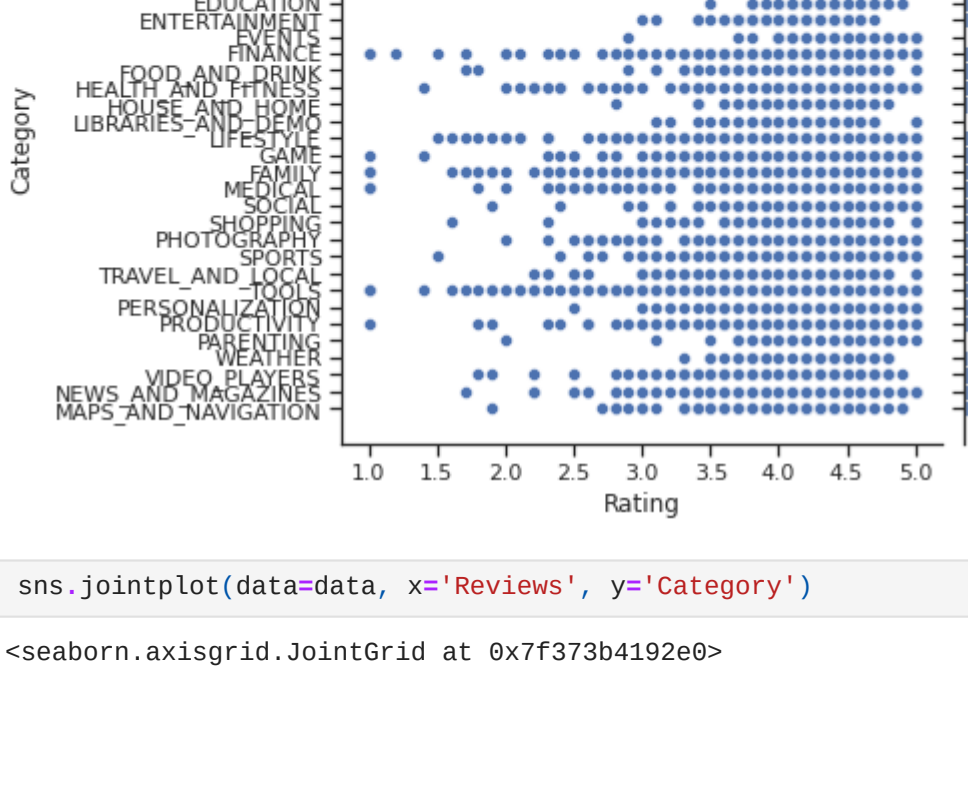
```
In [18]: sns.heatmap(data.corr(), annot = True, fmt = '.6f')
```



Сильнее всего отзывы коррелируют с количеством установок, при этом рейтинг не коррелирует практически ни с чем.

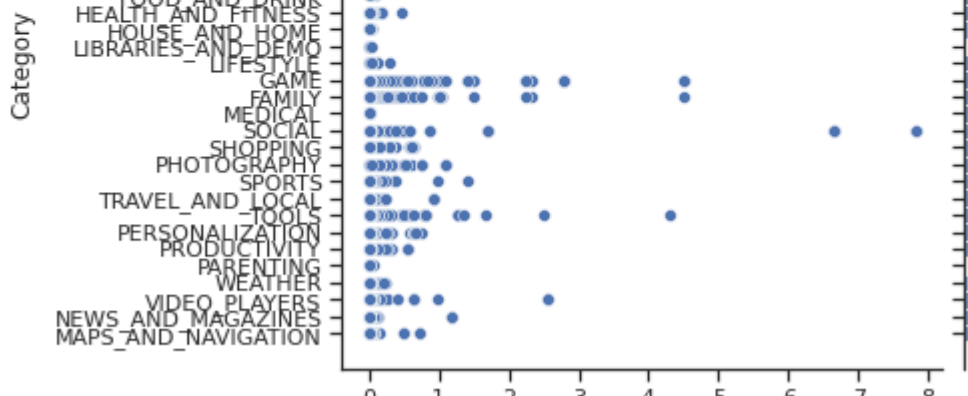
```
In [19]: sns.jointplot(data=data, x='Rating', y='Category')
```

```
Out[19]: <seaborn.axisgrid.JointGrid at 0x7f373b745b50>
```



```
In [20]: sns.jointplot(data=data, x='Reviews', y='Category')
```

```
Out[20]: <seaborn.axisgrid.JointGrid at 0x7f373b4192e0>
```



```
In [ ]:
```