```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
# Импортируем необходимые библиотеки
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns

# Загрузка данных
file_path = '/content/drive/My Drive/' + 'googleplaystore.csv'
df = pd.read_csv(file_path)
```

```
# Просмотр первых строк данных
df.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	10,000+	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	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August	1.2.4	4.0.3

Далее: Посмотреть рекомендованные графики New interactive sheet

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10841 non-null  object 
 1   Category        10841 non-null  object 
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object 
 4   Size            10841 non-null  object 
 5   Installs        10841 non-null  object 
 6   Type            10840 non-null  object 
 7   Price           10841 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10841 non-null  object 
 10  Last Updated    10841 non-null  object 
 11  Current Ver     10833 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

```
# Проверка наличия пропущенных значений
print(df.isnull().sum())
```

```
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
```

```
df.nunique()
```

	0
App	9660
Category	34
Rating	40
Reviews	6002
Size	462
Installs	22
Type	3
Price	93
Content Rating	6
Genres	120
Last Updated	1378
Current Ver	2832
Android Ver	33

dtype: int64

```
# Преобразование Rating в float64
df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')

# Преобразование Reviews в float64
df['Reviews'] = pd.to_numeric(df['Reviews'], errors='coerce')

# Преобразование Size в float64
df['Size'] = df['Size'].str.replace('M', '', regex=True)
df['Size'] = pd.to_numeric(df['Size'], errors='coerce')

# Преобразование Installs в float64 (удаляем ',' и '+')
df['Installs'] = df['Installs'].str.replace('[+,]', '', regex=True)
df['Installs'] = pd.to_numeric(df['Installs'], errors='coerce')

# Преобразование Price в float64 (удаляем '$')
df['Price'] = df['Price'].str.replace('$', '')
df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
```

```
df.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	159.0	19.0	10000.0	Free	0.0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967.0	14.0	500000.0	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live	ART_AND_DESIGN	4.7	87510.0	8.7	5000000.0	Free	0.0	Everyone	Art & Design	August	1.2.4	4.0.3

Далее: Посмотреть рекомендованные графики New interactive sheet

```
# Разделение на числовые и категориальные признаки
numeric_columns = ['Rating', 'Reviews', 'Size']
categorical_columns = ['Category', 'Type', 'Content Rating', 'Installs']

numeric_data = df[numeric_columns]
categorical_data = df[categorical_columns]

# Импутация числовых данных (среднее значение)
imp_numeric = SimpleImputer(missing_values=np.nan, strategy='mean')
numeric_data_imputed = imp_numeric.fit_transform(numeric_data)
numeric_data_imputed = imp_numeric.fit_transform(numeric_data)
numeric_data_imputed = pd.DataFrame(numeric_data_imputed, columns=numeric_columns)

# Импутация категориальных данных (наиболее частое значение)
imp_categorical = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
categorical_data_imputed = imp_categorical.fit_transform(categorical_data)
categorical_data_imputed = pd.DataFrame(categorical_data_imputed, columns=categorical_columns)

df[numeric_columns] = numeric_data_imputed
df[categorical_columns] = categorical_data_imputed


# Удаляем лишние столбцы
del df['App']
del df['Last Updated']
del df['Current Ver']
del df['Android Ver']
del df['Genres']
del df['Category']
print('done')
```

```
done
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Rating          10841 non-null  float64
 1   Reviews         10841 non-null  float64
 2   Size            10841 non-null  float64
 3   Installs        10841 non-null  object 
 4   Type            10841 non-null  object 
 5   Price           10840 non-null  float64
 6   Content Rating  10841 non-null  object 
dtypes: float64(4), object(3)
memory usage: 593.0+ KB
```

```
df.nunique()
```

```
                   0
Rating            41
Reviews         6002
Size             182
Installs          20
Type               3
Price             92
Content Rating     6

dtype: int64
```

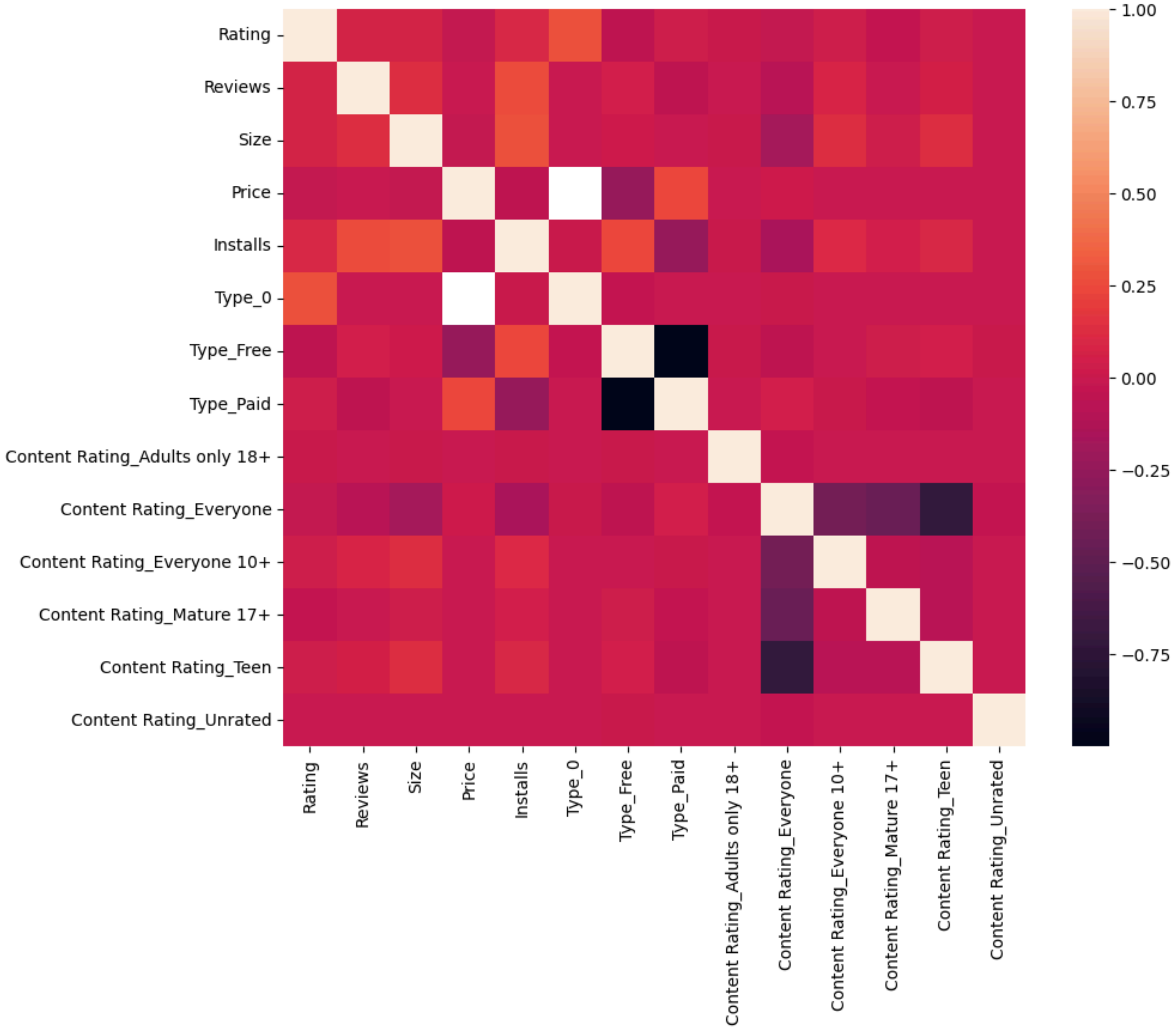
```
# Кодирование категориальных признаков
label_columns = ['Installs']
onehot_columns = ['Type', 'Content Rating']

# Последовательное кодирование labelEncoder
for i in label_columns:
  lb = LabelEncoder()
  result = lb.fit_transform(df[i])
  enc_df = pd.DataFrame(result, columns=[i])
  df = pd.concat([df.drop(columns=[i]), enc_df], axis=1)

# Кодирование onehot_columns встроенной функцией pandas
df = pd.get_dummies(df, columns=onehot_columns, drop_first=False)


# Корреляционный анализ
correlation_matrix = df.corr()

# Визуализация корреляционной матрицы
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix)
plt.show()
```

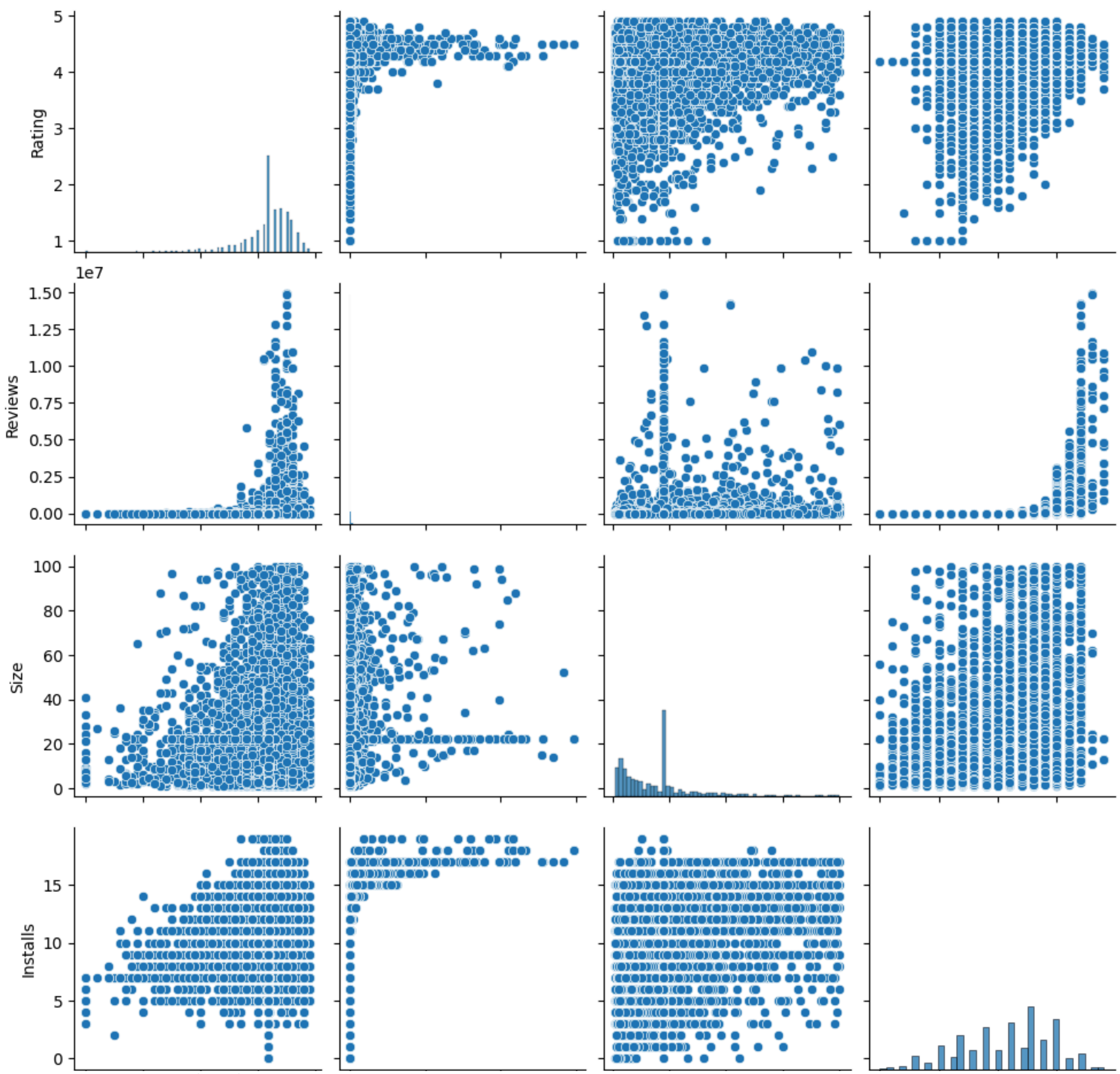


```
# Из парной диаграммы обнаружил аномалии в Rating и reviews
df = df[df['Rating'] < 5]
df = df[df['Reviews'] < 15_000_000]

# Выбор числовых признаков для парных диаграмм
numeric_columns = ['Rating', 'Reviews', 'Size', 'Installs']

numeric_data = df[numeric_columns]

# Построение парных диаграмм
sns.pairplot(numeric_data)
plt.show()
```



Не удается связаться с сервисом reCAPTCHA. Проверьте подключение к Интернету и перезагрузите страницу.