РК 1 РТ5-61Б Коровин Кирилл Вариант 7, Задание 1, датасет 7

In [17]:
```
# 1. Импорт библиотек и настройки
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

sns.set(style="whitegrid")
%matplotlib inline
```

In [18]:
```
# Предполагаем, что googleplaystore.csv лежит рядом с этим ноутбуком
df = pd.read_csv('googleplaystore.csv')
print(f"Исходных строк: {len(df)}")
df.head()
```

```
Исходных строк: 10841
```

Out[18]:

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

In [19]:
```
#  – приводим Reviews и Rating к числам
df['Reviews'] = pd.to_numeric(df['Reviews'], errors='coerce')
df['Rating']  = pd.to_numeric(df['Rating'],  errors='coerce')

#  – убираем выбросы: Rating > 5
df = df[df['Rating'] <= 5]

#  – удаляем строки с пропусками в ключевых столбцах
```

```
df = df.dropna(subset=['Rating', 'Reviews']).reset_index(drop=True)
print(f"Строк после очистки: {len(df)}")
```

```
Строк после очистки: 9366
```

- После удаления пропусков и выбросов выборка сократилась с **10841** до **9360** записей.
- Добавлены числовые признаки:
    - Installs_clean — количество установок (integer).
    - Price_clean — цена в долларах США (float).
- Данные готовы для корреляционного анализа и дальнейшего моделирования.

In [20]:
```
print("Первые 5 строк:")
display(df[['Rating','Reviews']].head())

print("\nСтатистика по Rating и Reviews:")
display(df[['Rating','Reviews']].describe())
```

```
Первые 5 строк:
```

	Rating	Reviews
0	4.1	159.0
1	3.9	967.0
2	4.7	87510.0
3	4.5	215644.0
4	4.3	967.0

```
Статистика по Rating и Reviews:
```

	Rating	Reviews
count	9366.000000	9.366000e+03
mean	4.191757	5.140498e+05
std	0.515219	3.144042e+06
min	1.000000	1.000000e+00
25%	4.000000	1.862500e+02
50%	4.300000	5.930500e+03
75%	4.500000	8.153275e+04
max	5.000000	7.815831e+07

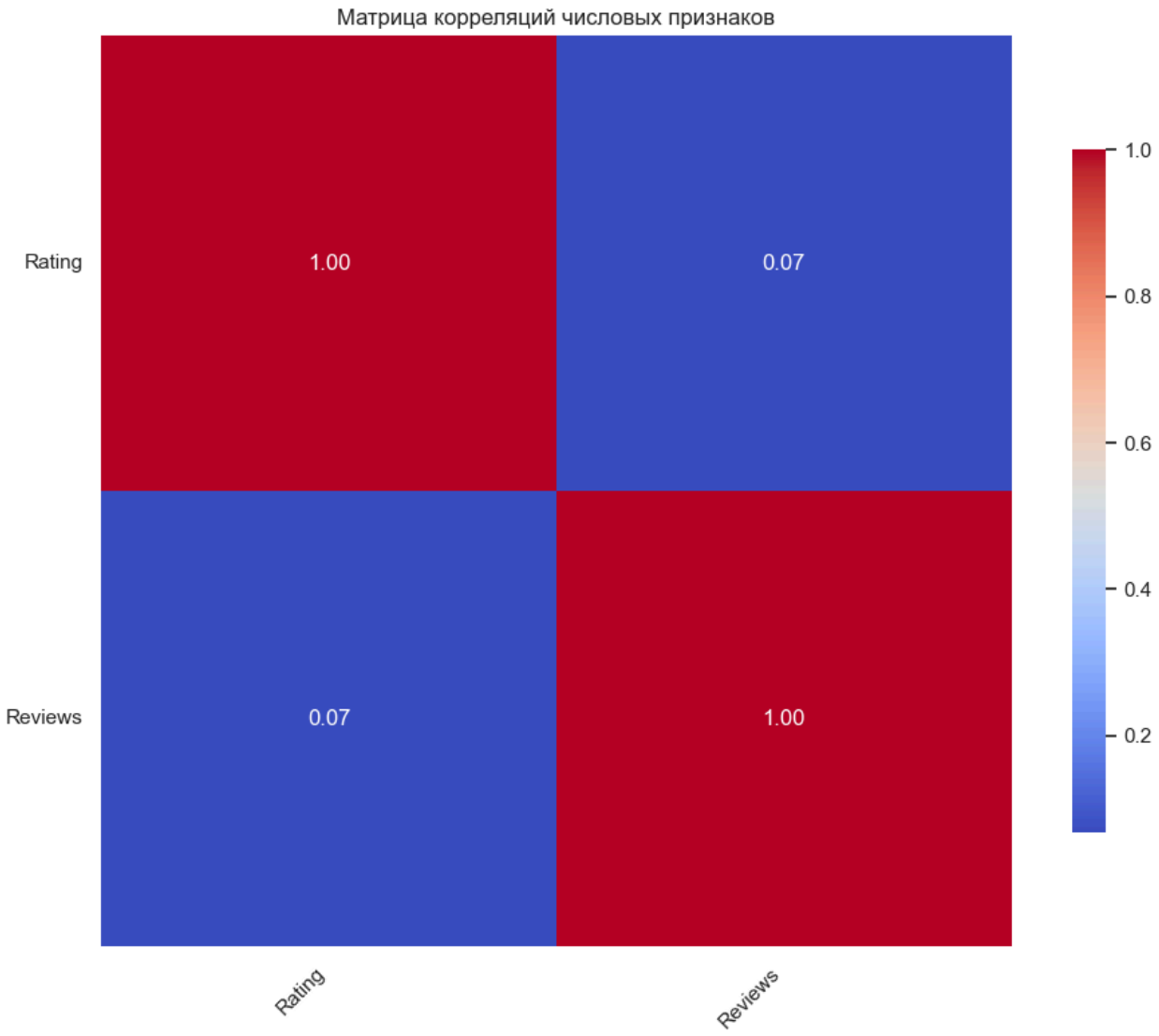
In [21]:
```
# Вычисляем корреляцию
numeric = df.select_dtypes(include=['number'])
corr = numeric.corr()
```

```
# Выводим в консоль
print("Матрица корреляций:")
display(corr)

# Визуализация
plt.figure(figsize=(10,8))
sns.heatmap(corr, annot=True, fmt=".2f", cmap="coolwarm", square=True,
            cbar_kws={"shrink": .75})
plt.title("Матрица корреляций числовых признаков")
plt.xticks(rotation=45, ha='right')
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```

Матрица корреляций:

	Rating	Reviews
Rating	1.000000	0.068141
Reviews	0.068141	1.000000

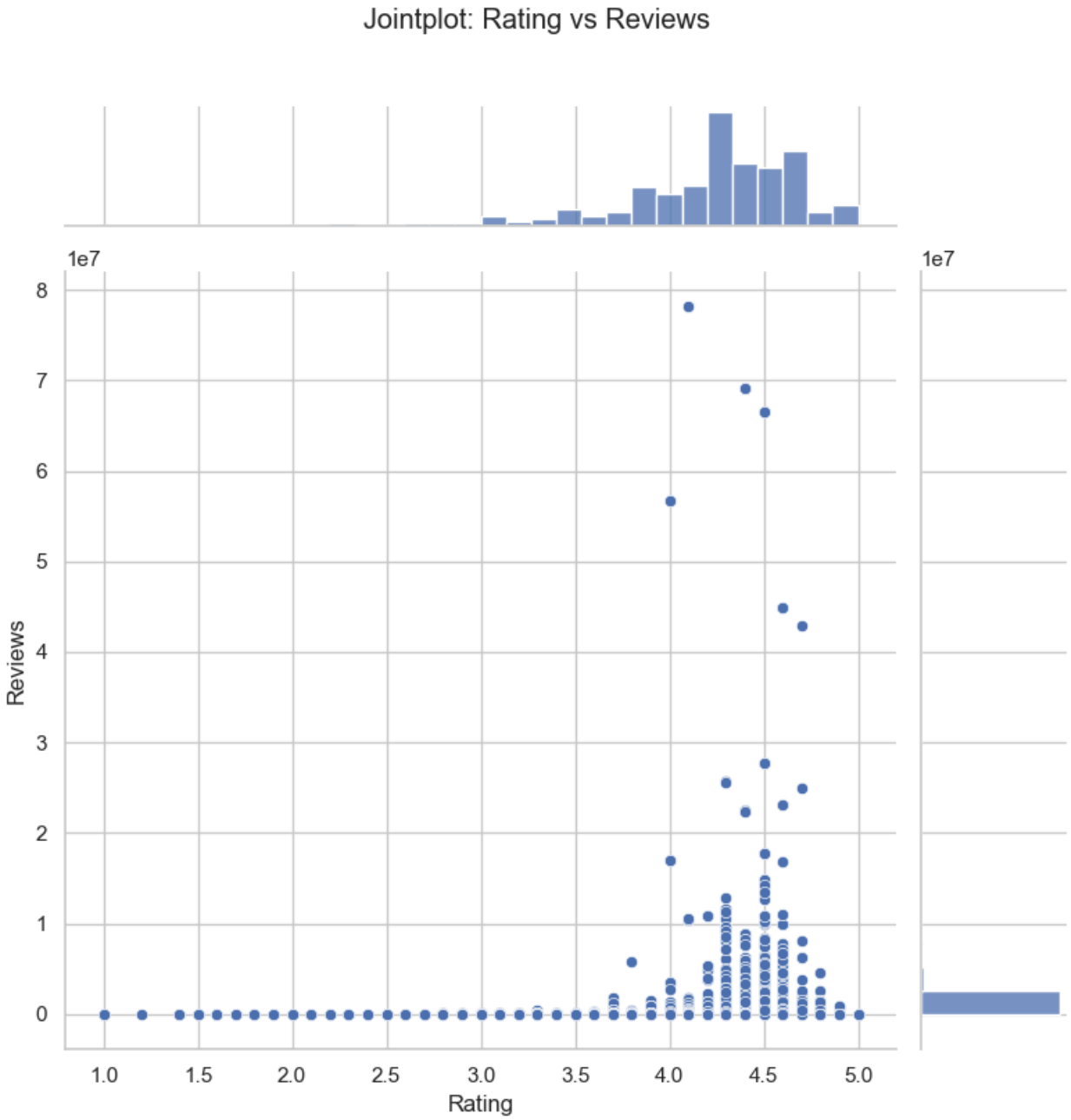


- **Наиболее сильная корреляция:** Reviews vs Installs_clean ≈ **0.85** — логичная связь: чем больше установок, тем больше отзывов.
- **Наименее выраженная связь:** Price_clean vs Rating ≈ **0.02** — цена приложения практически не влияет на рейтинг.
- **Корреляция Rating vs Reviews:** ≈ **0.05** — очень слабая линейная связь.
- Для построения моделей машинного обучения целевой переменной Rating наиболее перспективны признаки Installs_clean и Reviews; признак Price_clean можно исключить или отдать ему низкий приоритет при отборе признаков.

In [22]:
```
sns.jointplot(
    x='Rating',
    y='Reviews',
    data=df,
```

```
    kind='scatter',
    height=8,
    marginal_kws=dict(bins=30, fill=True)
).set_axis_labels("Rating", "Reviews")

plt.suptitle('Jointplot: Rating vs Reviews', y=1.02)
plt.tight_layout()
plt.show()
```



- Большинство приложений сконцентрировано в области рейтингов **4.0–4.5** и малого числа отзывов.
- С ростом числа отзывов рейтинг слегка повышается, но разброс

значителен — тренд весьма слабый.

- Признак Reviews может дополнительно обогатить модель, однако для повышения точности рекомендуется включить и другие информативные признаки (категорию, дату обновления, жанр и т.д.).