```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load the dataset
df = pd.read_csv('/content/InstagramThreads_Reviews-Data-1.csv')
```

```
#exploring the dataset
df.head()
```

	source	review_description	rating	review_date
0	Google Play	Meh. Not the greatest experience on a Chromebo...	2	7/8/2023 14:18
1	Google Play	Pretty good for a first launch!! Its easy to u...	3	7/19/2023 20:52
2	Google Play	For a brand new app, it's very well optimized....	3	7/6/2023 23:03
3	Google Play	Great app with a lot of potential! However, th...	3	7/10/2023 0:53
4	Google Play	The app is good, but it needs a lot of functio...	3	7/6/2023 16:57

Next steps: Generate code with df   View recommended plots

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32910 entries, 0 to 32909
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   source              32910 non-null  object
 1   review_description  32910 non-null  object
 2   rating              32910 non-null  int64
 3   review_date         32910 non-null  object
dtypes: int64(1), object(3)
memory usage: 1.0+ MB
```

```
df.describe()
```

	rating
count	32910.000000
mean	3.398481
std	1.751480
min	1.000000
25%	1.000000
50%	4.000000
75%	5.000000
max	5.000000

```
#Data preprocessing and cleaning
df.isnull().sum()
```

```
source                0
review_description    0
rating                0
review_date           0
dtype: int64
```

```
#dropping NA values if any
df.dropna(inplace=True)
```

```
df.describe()
```

	rating
count	32910.000000
mean	3.398481
std	1.751480
min	1.000000
25%	1.000000
50%	4.000000
75%	5.000000
max	5.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32910 entries, 0 to 32909
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   source              32910 non-null  object
 1   review_description  32910 non-null  object
 2   rating              32910 non-null  int64 
 3   review_date         32910 non-null  object
dtypes: int64(1), object(3)
memory usage: 1.0+ MB
```

```
#Convert 'review_date' to datetime

df['review_date'] = pd.to_datetime(df['review_date'])
```

```
df.head()
```

	source	review_description	rating	review_date
0	Google Play	Meh. Not the greatest experience on a Chromebo...	2	2023-07-08 14:18:00
1	Google Play	Pretty good for a first launch!! Its easy to u...	3	2023-07-19 20:52:00
2	Google Play	For a brand new app, it's very well optimized....	3	2023-07-06 23:03:00
3	Google	Great app with a lot of potential! However,	[illegible]	2023-07-10

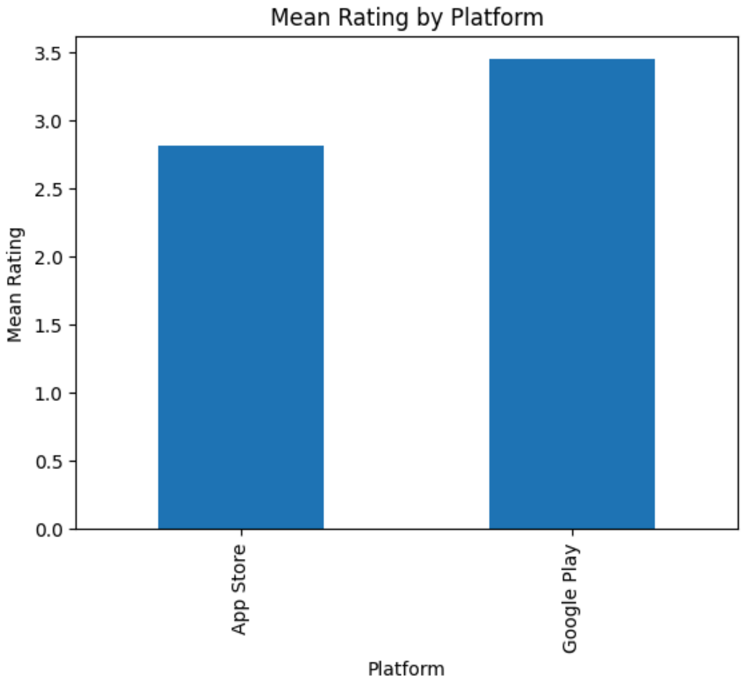
Next steps: Generate code with df View recommended plots

```
#exploritory data analysis

#Mean rating by platform
mean_rating_by_platform = df.groupby('source')['rating'].mean()
print(mean_rating_by_platform)


print('\n')

#median rating by platform
median_rating_by_platform = df.groupby('source')['rating'].median()
print(median_rating_by_platform)

print('\n')

#mode rating by platform
mode_rating_by_platform = df.groupby('source')['rating'].agg(pd.Series.mode)
print(mode_rating_by_platform)
```

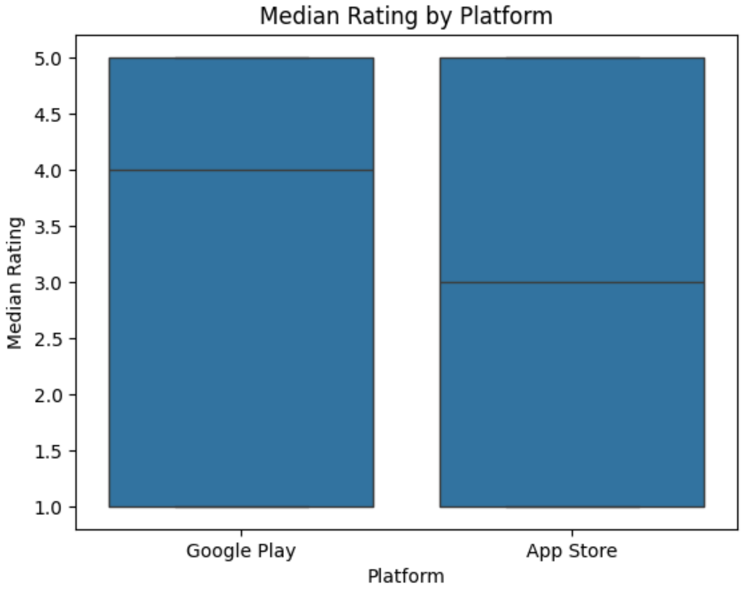
```
source
App Store      2.813258
Google Play    3.449521
Name: rating, dtype: float64


source
App Store      3.0
Google Play    4.0
Name: rating, dtype: float64
```

```
source
App Store      1
Google Play    5
Name: rating, dtype: int64
```

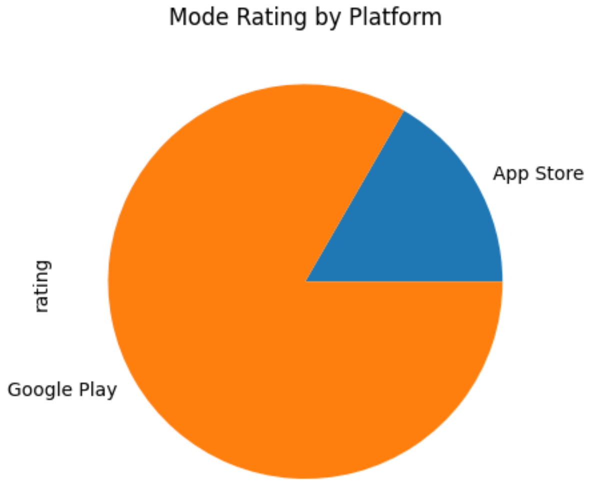
```
#visual represntation

import matplotlib.pyplot as plt
#bar graph for mean ratings


mean_rating_by_platform.plot(kind='bar')
plt.title('Mean Rating by Platform')
plt.xlabel('Platform')
plt.ylabel('Mean Rating')
plt.show()

#Mean rating is better on google play store
```



```
#boxplot for median rating by platform
sns.boxplot(x='source', y='rating', data=df)
plt.title('Median Rating by Platform')
plt.xlabel('Platform')
plt.ylabel('Median Rating')
plt.show()


#median is better on the google playstore
```



```
#pie chart for number of ratings per platform
mode_rating_by_platform.plot(kind='pie')
plt.title('Mode Rating by Platform')
plt.show()

#Google play has substantially better mode rating
```



```
#Number of reviews by platform
reviews_by_platform = df['source'].value_counts()
print(reviews_by_platform)
```

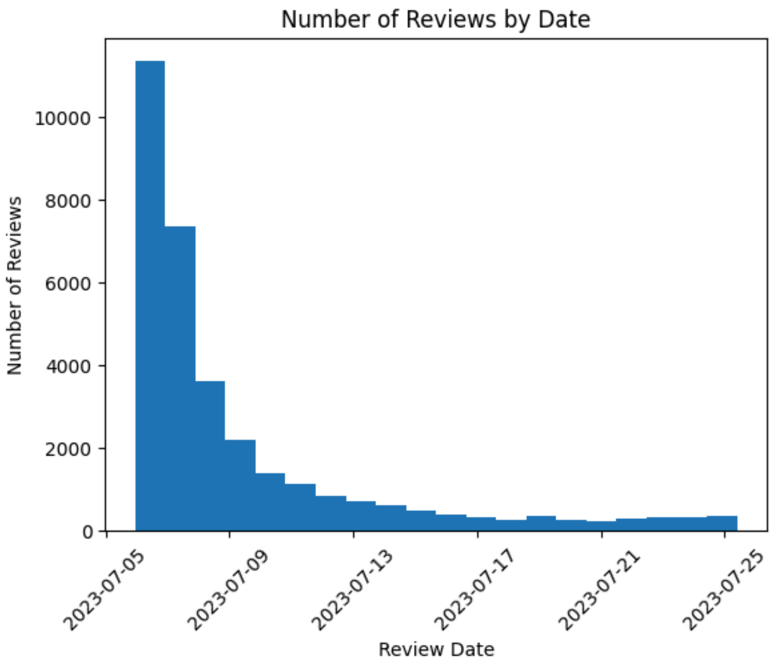
```
source
Google Play    30270
App Store       2640
Name: count, dtype: int64
```

```
#Mean rating over time

mean_rating_over_time = df.groupby('review_date')['rating'].mean()
print(mean_rating_over_time)
'''plt.plot(mean_rating_over_time)
plt.title('Mean Rating Over Time')
plt.xlabel('Date')
plt.ylabel('Mean Rating')
plt.show()'''
```

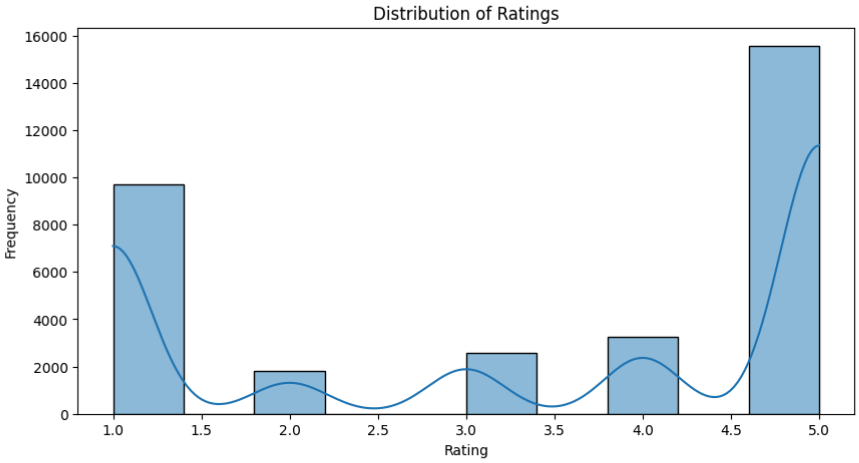
```
review_date
2023-07-05 22:53:00    3.0
2023-07-05 22:54:00    5.0
2023-07-05 22:56:00    5.0
2023-07-05 22:59:00    4.0
2023-07-05 23:00:00    5.0
                      ... 
2023-07-25 09:21:00    1.0
2023-07-25 09:32:00    2.0
2023-07-25 09:35:00    1.0
2023-07-25 09:37:00    5.0
2023-07-25 09:42:00    5.0
Name: rating, Length: 11749, dtype: float64
'plt.plot(mean_rating_over_time)\nplt.title('Mean Rating Over Time')\nplt.xlabel('Da
te')\nplt.ylabel('Mean Rating')\nplt.show()'
```

```
#Number of reviews over time
reviews_over_time = df['review_date'].value_counts()
print(reviews_over_time.head())
print(reviews_over_time.tail())

print('\n')

#visualization

plt.hist(df['review_date'],bins = 20)

plt.xlabel('Review Date')
plt.ylabel('Number of Reviews')
plt.title('Number of Reviews by Date')


# Rotate x-axis labels for better readability
plt.xticks(rotation=45)
plt.show()
```

```
review_date
2023-07-06 16:43:00    22
2023-07-06 13:35:00    20
2023-07-06 16:49:00    20
2023-07-06 13:36:00    19
2023-07-06 14:22:00    19
Name: count, dtype: int64
review_date
2023-07-16 13:52:00    1
2023-07-10 03:35:00    1
2023-07-16 14:50:00    1
2023-07-14 04:26:00    1
2023-07-17 06:39:00    1
Name: count, dtype: int64
```



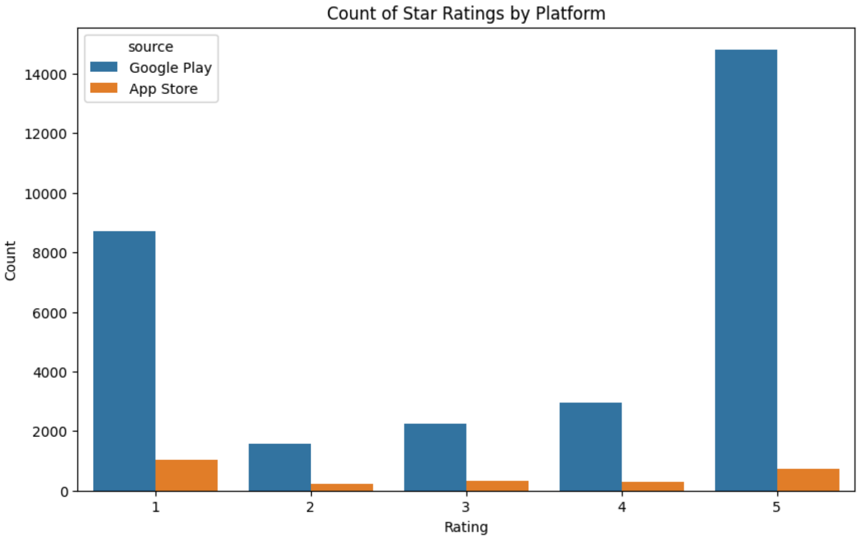
```
# Distribution of ratings
plt.figure(figsize=(10, 5))
sns.histplot(df['rating'], bins=10, kde=True)
plt.title('Distribution of Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```



```
#generating word cloud

from wordcloud import WordCloud
# Randomly select 30-50 reviews
sample_reviews = df['review_description'].sample(50).values

# Generate
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(' '.join(sample_reviews))

# Display
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

app
Twitter
Instagram
account
Good
Threads
use
see
delete
easy
insta
cheap
cool
follow
copy
Will
Nice
really
Best
alternative
Thank
version
Super
reason
Great
Perfect
Facebook
followers
following
personal
complicated
giving
technical
love
anyway
working
data
credit
available
notice
post
keep
problem
tons
ya
ha
clear
getting
Crashing
better
addictive
page
suggestions
Bug
information
change
diwlond
score
fitting
Bank
Android
comments
musky
idiots
photo
mack
idea
hi
slow
bhi
minds
Galaxy
adding
rival
tried
raha
chalta
got
bhai
krogga
kro
nhi
Bakvas
RIP
aasa
Access
lets
koi
mat
isn
star
bird
Isa
ka
General
mag
plz
desktop
days
reinstall
attachments

```
#difference in the reviews by stars per platform
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='rating', hue='source')
plt.title('Count of Star Ratings by Platform')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()
```


Count of Star Ratings by Platform
source
🟦 Google Play
🟧 App Store
Count
Rating