In [1]:
```
import numpy as np
import pandas as pd
import matplotlib as mlp
import matplotlib.pyplot as plt
import plotly
import plotly.graph_objs as go
import plotly.express as px
import seaborn as sns
```

In [2]:
```
df = pd.read_csv(r"C:\Users\ho\Desktop\apps.csv")
df
```

Out[2]:

	Unnamed: 0	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	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	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	2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9654	10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	July 25, 2017	1.48	4.1 and up
9655	10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	July 6, 2018	1.0	4.1 and up
9656	10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
9657	10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	NaN	1,000+	Free	0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device
9658	10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	Lifestyle	July 25, 2018	Varies with device	Varies with device

9659 rows × 14 columns

In [3]:
```
df.duplicated().sum()
```

Out[3]:
```
0
```

In [4]:
```
apps = df.drop_duplicates()
print("Total numbers of Apps : ", apps[ 'App' ].count())
apps.head()
```

```
Total numbers of Apps :  9659
```

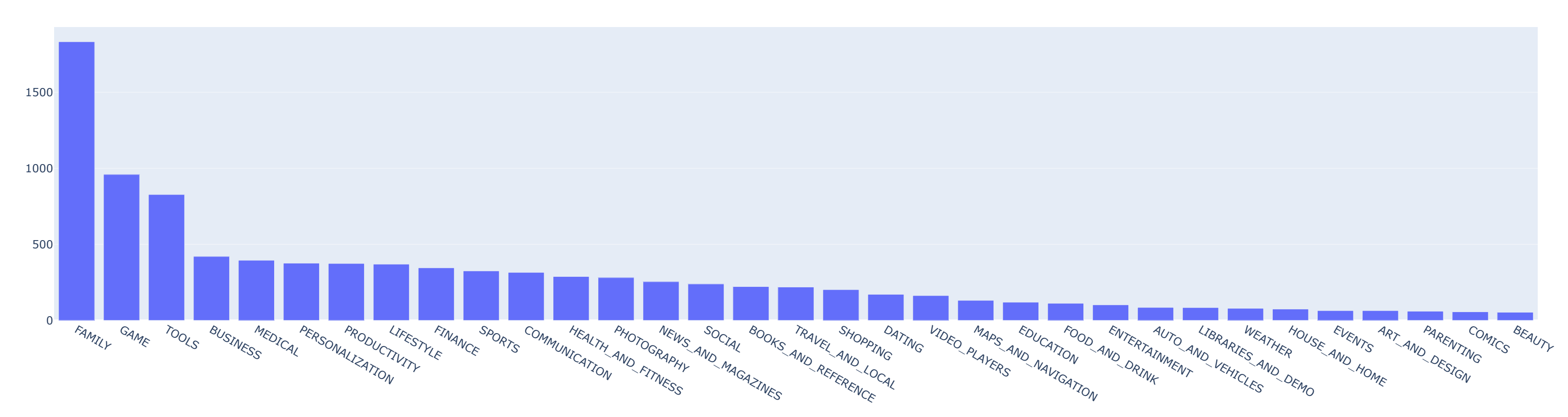
Out[4]:

	Unnamed: 0	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	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	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	2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

In [5]:
```
ch_remove = ('+' , '$' , 'M' , ',' )
col_clean = ('Size','Installs','Price')
for col in col_clean:
    for char in ch_remove:
        apps[col] = apps[col].str.replace(char, '')
    apps[col] = pd.to_numeric(apps[col])
```

In [6]:
```
n_categories = len(set(apps['Category']))
print('Number of categories = ', n_categories)
n_apps_in_category = apps['Category'].value_counts().sort_values(ascending=False)
data1 = go.Bar(
    x = n_apps_in_category.index,
    y = n_apps_in_category.values
)
fig = go.Figure(data1)
fig.show()
```

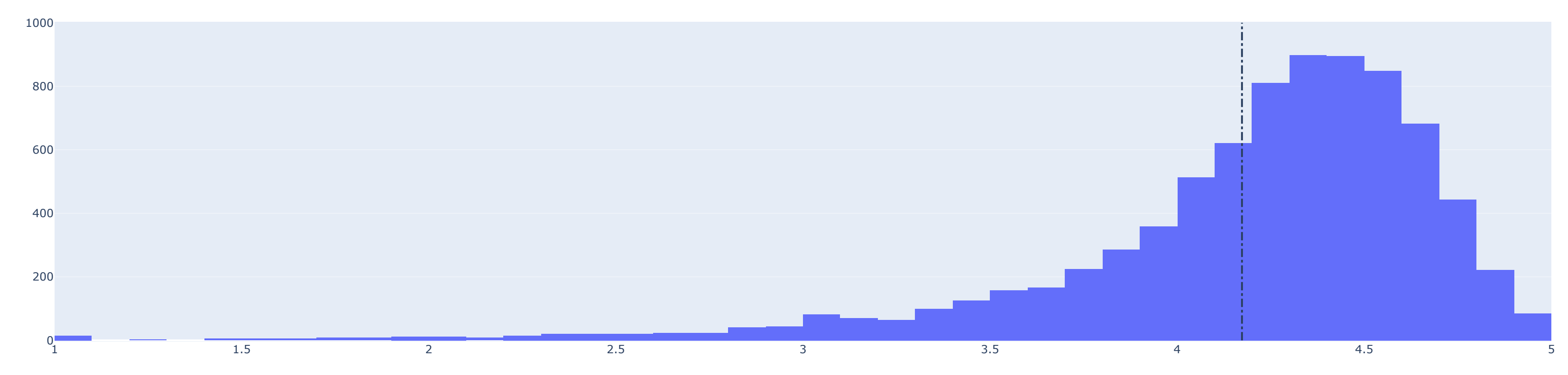
```
Number of categories =  33
```



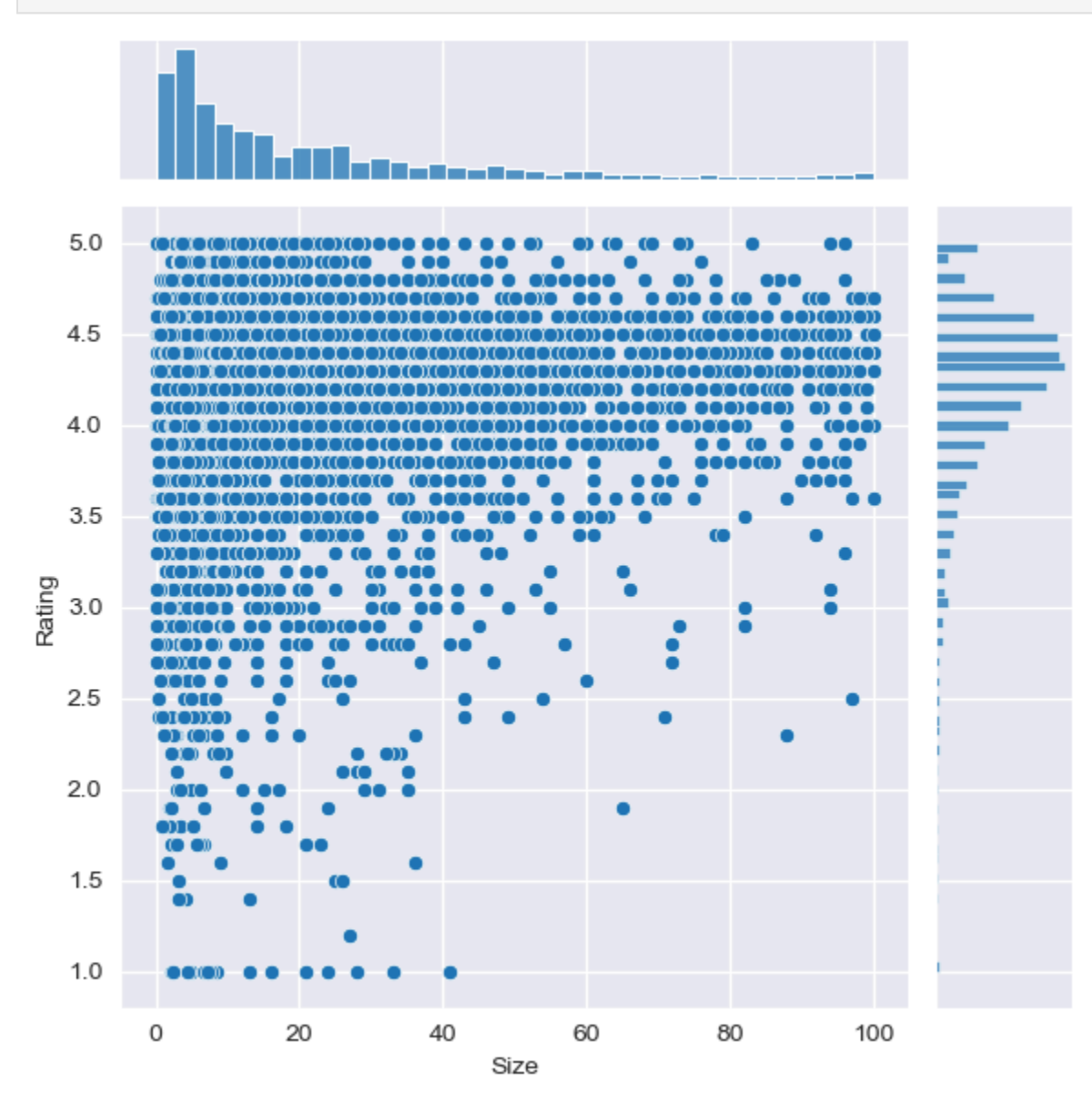
In [7]:
```
print("Avarage App rating = ", apps['Rating'].mean())
```

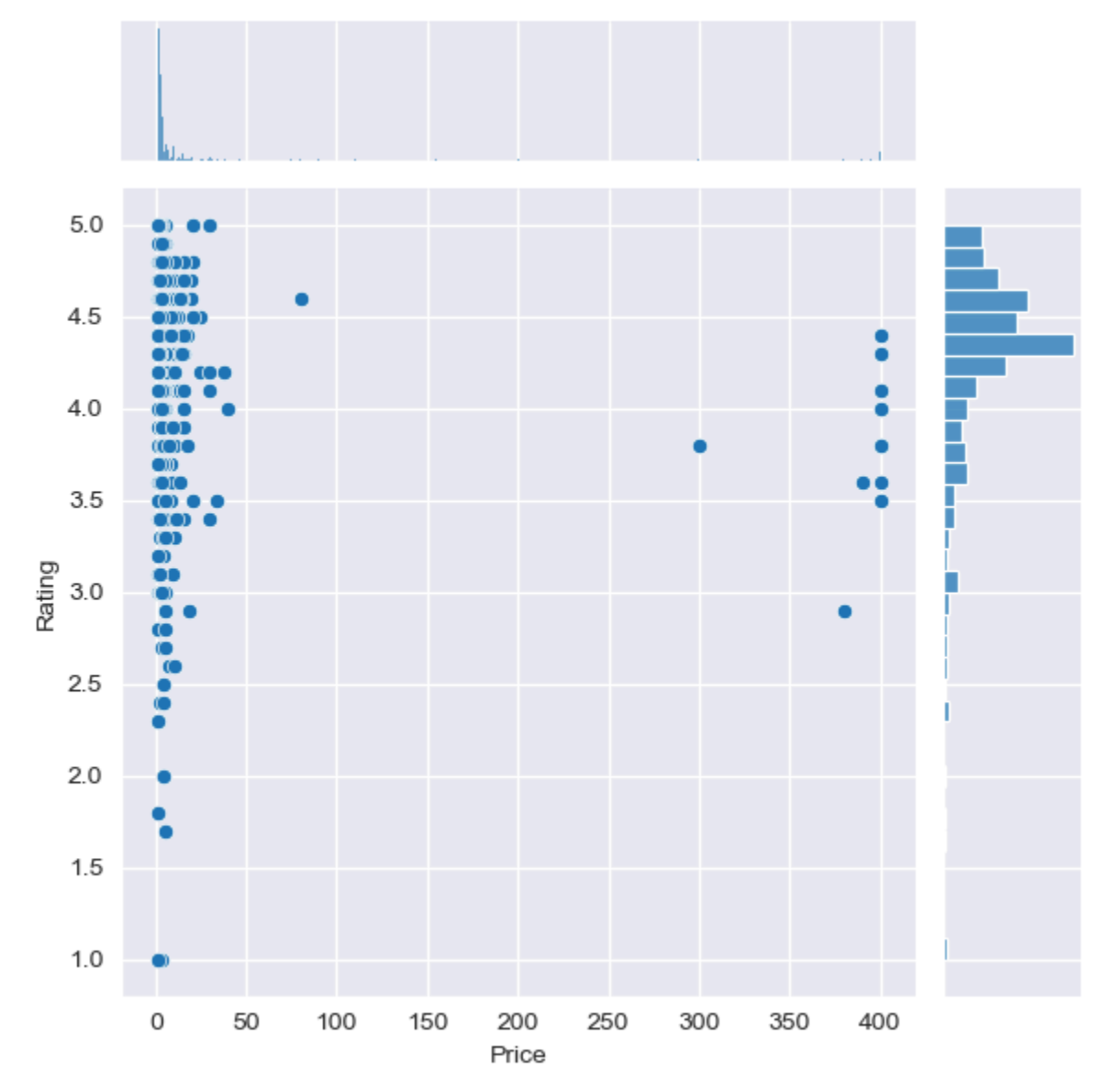
```
Avarage App rating =  4.173243045387994
```

In [8]:
```
data = [go.Histogram(
        x = apps['Rating'],
        xbins = {'start': 1, 'size': 0.1, 'end' : 5}
)]
avg_rating = apps['Rating'].mean()
# Vertical dashed line to indicate the average app rating
layout = {'shapes': [{
            'type' :'line',
            'x0': avg_rating,
            'y0': 0,
            'x1': avg_rating,
            'y1': 1000,
            'line': { 'dash': 'dashdot'}
        }]
        }
plotly.offline.iplot({'data': data, 'layout': layout})
```



In [10]:
```
sns.set_style("darkgrid")
import warnings
warnings.filterwarnings("ignore")
large_categories = apps.groupby('Category').filter(lambda x: len(x) >= 250).reset_index() #Relation between app category and app price
paid_app = apps[apps['Type'] == 'Paid']
plt1 = sns.jointplot( x = 'Size', y = 'Rating', data = large_categories)
plt2 = sns.jointplot( x = 'Price', y = 'Rating', data = paid_app)
```



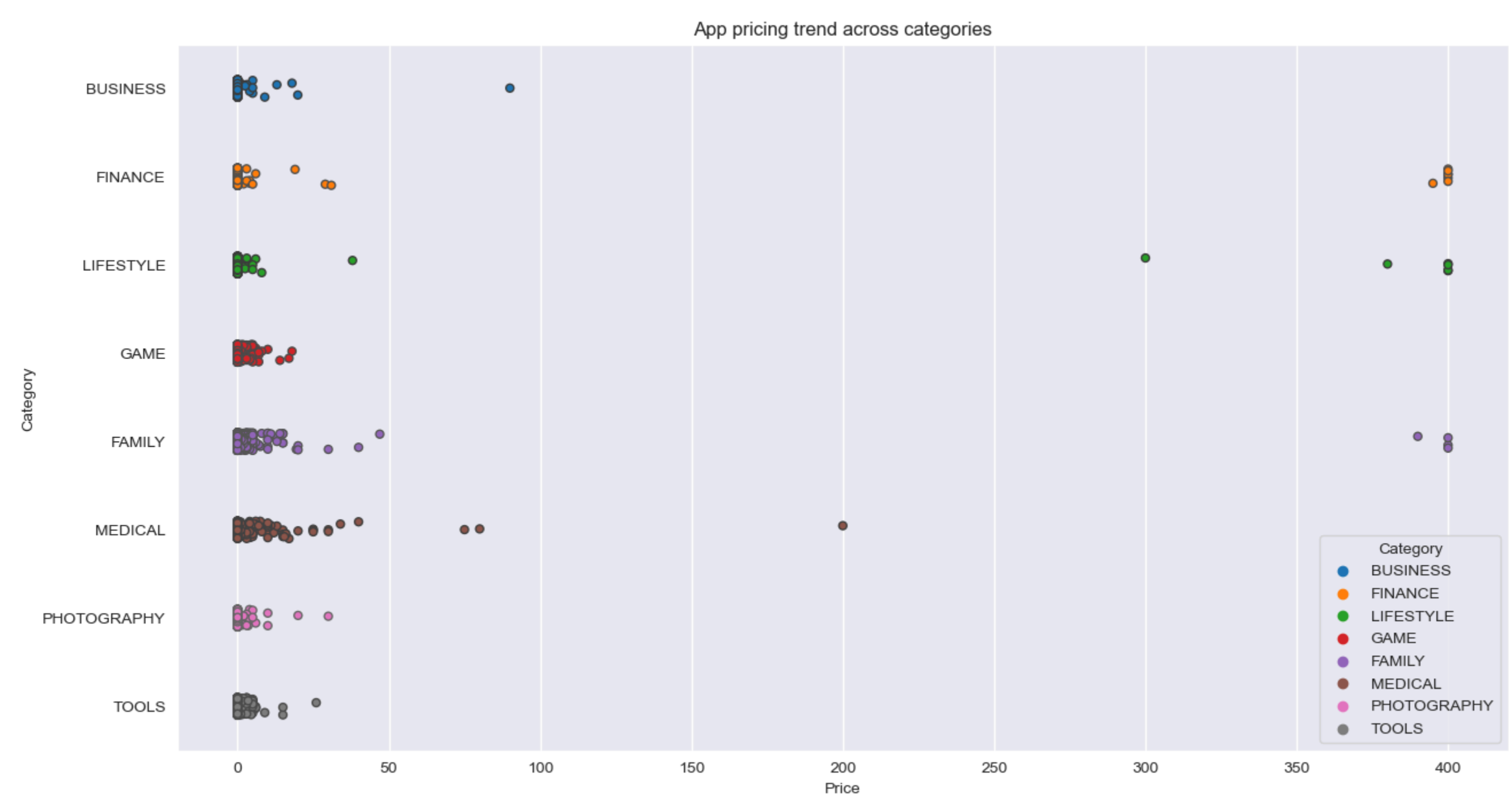


Relation between app category and app price

In [11]:
```
fig, ax = plt.subplots()
fig.set_size_inches(15, 8)
popular_app_cats = apps[apps.Category.isin(['GAME', 'FAMILY', 'PHOTOGRAPHY', 'MEDICAL', 'TOOLS', 'FINANCE', 'LIFESTYLE', 'BUSINESS'])]
ax = sns.stripplot(x='Price', y='Category', data = popular_app_cats,  jitter=True, linewidth=1, hue='Category')
ax.set_title('App pricing trend across categories')
apps_above_200 = popular_app_cats[['Category', 'App', 'Price']][popular_app_cats['Price'] > 200]
apps_above_200
```

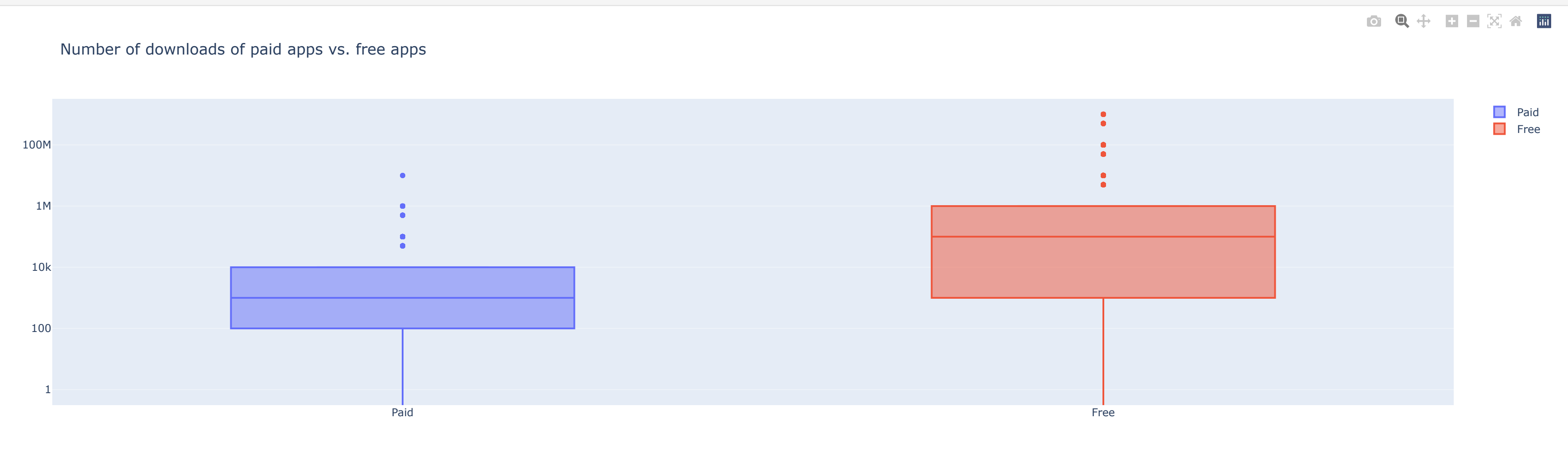
Out[11]:

	Category	App	Price
3327	FAMILY	most expensive app (H)	399.99
3465	LIFESTYLE	💎 I'm rich	399.99
3469	LIFESTYLE	I'm Rich - Trump Edition	400.00
4396	LIFESTYLE	I am rich	399.99
4398	FAMILY	I am Rich Plus	399.99
4399	LIFESTYLE	I am rich VIP	299.99
4400	FINANCE	I Am Rich Premium	399.99
4401	LIFESTYLE	I am extremely Rich	379.99
4402	FINANCE	I am Rich!	399.99
4403	FINANCE	I am rich(premium)	399.99
4406	FAMILY	I Am Rich Pro	399.99
4408	FINANCE	I am rich (Most expensive app)	399.99
4410	FAMILY	I Am Rich	389.99
4413	FINANCE	I am Rich	399.99
4417	FINANCE	I AM RICH PRO PLUS	399.99
8763	FINANCE	Eu Sou Rico	394.99
8780	LIFESTYLE	I'm Rich/Eu sou Rico/أنا غني/我很有錢	399.99



Popularity of paid apps vs free apps

In [12]:
```
tr0 = go.Box(
    y=apps['Installs'][apps['Type'] == 'Paid'],
    name = 'Paid'
)
tr1 = go.Box(
    y=apps['Installs'][apps['Type'] == 'Free'],
    name = 'Free'
)
layout = go.Layout(
    title = "Number of downloads of paid apps vs. free apps",
    yaxis = dict(
        type = 'log',
        autorange = True
    )
)
data = [tr0, tr1]
plotly.offline.iplot({'data': data, 'layout': layout})
```



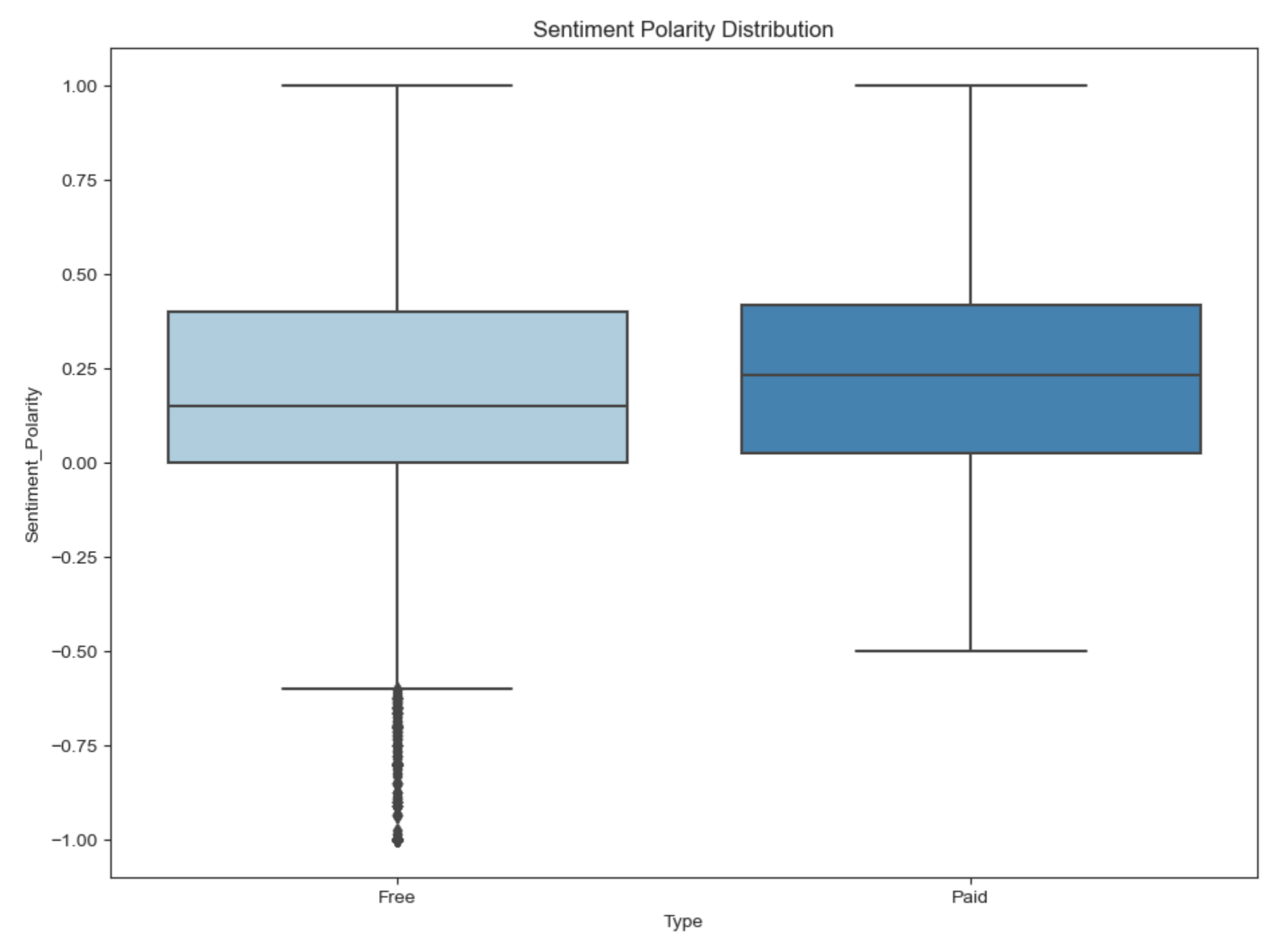
Sentiment analysis of user reviews

In [16]:
```
# Load user_reviews.csv
reviews_df = pd.read_csv(r"C:\Users\ho\Desktop\user_reviews.csv")
# Join and merge the two dataframe
merged_df = pd.merge(apps, reviews_df, on='App', how = "inner")
# Drop NA values from Sentiment and Translated_Review columns
merged_df = merged_df.dropna(subset=['Sentiment', 'Translated_Review'])

sns.set_style('ticks')
fig, ax = plt.subplots()
fig.set_size_inches(11, 8)

# User review sentiment polarity for paid vs. free apps
ax = sns.boxplot(x = 'Type', y = 'Sentiment_Polarity', palette="Blues", data = merged_df)
ax.set_title('Sentiment Polarity Distribution')
```

Out[16]:
```
Text(0.5, 1.0, 'Sentiment Polarity Distribution')
```



In [ ]: