Google Play Store reviews scraping and Text Analytics

Reviews scraping from Google Play Store.

In [1]:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google_play_scraper import app, Sort, reviews_all
```

In [2]:

```
# Define and configure Google Play Scraper library
hk_users_reviews = reviews_all(
    'com.aiahk.idirect',
    sleep_milliseconds=0, # defaults to 0
    lang='en', # defaults to 'en'
    country='us', # defaults to 'us'
    sort=Sort.MOST_RELEVANT, # defaults to Sort.MOST_RELEVANT
    count=3
)
```

In [3]:

```
# Convert collected reviews data into dataframe
df_reviews = pd.DataFrame(np.array(hk_users_reviews),columns=['review'])
df_reviews = df_reviews.join(pd.DataFrame(df_reviews.pop('review').tolist()))
# Display dataframe header
df_reviews.head()
```

Out[3]:

	reviewId	userName	userImage	content	score	thumbsU
0	25297eac-39f9-4195-9730-cb5bd645887d	Jacky Lei	https://play-lh.googleusercontent.com/a-/ACB-R...	The app seems lack of cache memory. while swit...	3	
1	1b2b1a8e-1bdb-4b8e-a364-c5f04403b65c	Tim Kwan	https://play-lh.googleusercontent.com/a/AGNmyx...	Bad experience that i cannot go through the fl...	1	
2	bd0766f5-7e72-4ff9-a834-3cad367396a8	Chau Selena	https://play-lh.googleusercontent.com/a-/ACB-R...	Horrible app, slow and doesn't work for linkin...	1	
3	2bde5b83-7c9f-4f44-941c-de7d759664d6	Patrick Kwan	https://play-lh.googleusercontent.com/a/AGNmyx...	I used this app almost daily. Start from this ...	3	
4	8fa6bf3f-8a09-44c8-86ce-7ff84dafcf88	Thomas Godzilla	https://play-lh.googleusercontent.com/a-/ACB-R...	Too hard to use. Have to flip between many scr...	1	

In [4]:

```
# Check dataframe information
df_reviews.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 383 entries, 0 to 382
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype         
---  ------                --------------  -----         
 0   reviewId              383 non-null    object        
 1   userName              383 non-null    object        
 2   userImage             383 non-null    object        
 3   content               383 non-null    object        
 4   score                 383 non-null    int64         
 5   thumbsUpCount         383 non-null    int64         
 6   reviewCreatedVersion  337 non-null    object        
 7   at                    383 non-null    datetime64[ns]
 8   replyContent          334 non-null    object        
 9   repliedAt             334 non-null    datetime64[ns]
dtypes: datetime64[ns](2), int64(2), object(6)
memory usage: 30.0+ KB
```

In [5]:

```
# Count number of review scores
df_reviews['score'].value_counts()
```

Out[5]:

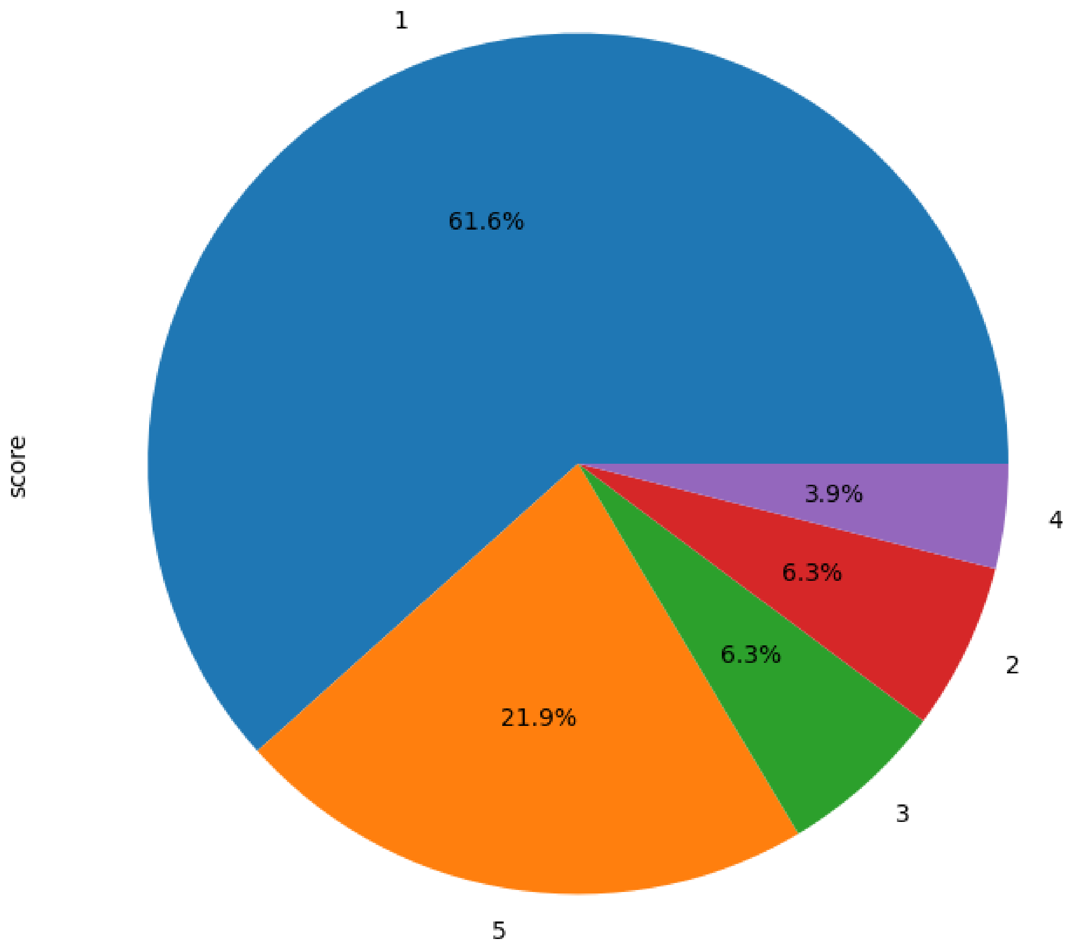
```
1    236
5     84
3     24
2     24
4     15
Name: score, dtype: int64
```

In [6]:

```
# Visualize review scores as pie chart
df_reviews['score'].value_counts().plot(kind='pie',figsize=(8,8), autopct='%1.1f%%')
```

Out[6]:

```
<AxesSubplot:ylabel='score'>
```



In [7]:

```
# Create new dataframe with review content and score for further analysis
df_reviews_content = pd.DataFrame(df_reviews, columns=['content','score'])
```

In [8]:

```
# Display new dataframe header
df_reviews_content.head()
```

Out[8]:

	content	score
0	The app seems lack of cache memory. while swit...	3
1	Bad experience that i cannot go through the fl...	1
2	Horrible app, slow and doesn't work for linkin...	1
3	I used this app almost daily. Start from this ...	3
4	Too hard to use. Have to flip between many scr...	1

Text Analytics (Sentiment Analysis) of Reviews Content dataframe.

In [9]:

```
# Import necessary libraries
from azure.ai.textanalytics import TextAnalyticsClient
from azure.core.credentials import AzureKeyCredential
```

In [10]:

```
# Define the service key and endpoint of Azure Text Analytics
key = ""
endpoint = "https://sma-exp10.cognitiveservices.azure.com/"
```

In [11]:

```
# Configure Azure Text Analytics client library
ta_credential = AzureKeyCredential(key)
text_analytics_client = TextAnalyticsClient(
        endpoint=endpoint,
        credential=ta_credential)
client = text_analytics_client

reviews_content_sentiment = []

# Pass review content to Azure Text Analytics and collect sentiment result
for index, headers in df_reviews_content.iterrows():
    reviews_content = str(headers['content'])
    print("Review Content: {}".format(reviews_content))
    documents = [reviews_content]
    response = client.analyze_sentiment(documents=documents, language="zh-hant")[0]
    sentiment = response.sentiment
    print("Review Content Sentiment: {}".format(sentiment))
    reviews_score = str(headers['score'])
    print("Review Content Score: {}".format(reviews_score))
    reviews_content_sentiment.append([reviews_content, sentiment, reviews_score])

# Convert collected news headers with sentiment to Pandas dataframes.
reviews_content_sentiment = pd.DataFrame(reviews_content_sentiment, columns=['content','
```

```
Review Content: The app seems lack of cache memory. while switching bet
ween different apps, all typed information in AIA app will be swiped ou
t and jumped back to the front page. Need to retype everyone again. Thi
s is unacceptable! Last but not least, the loading time of this app is
quite slow, I thought I am connecting to a small-scale home network ser
ver when I am loading my page each time. I hope these problems could be
fixed in an enterprise network standard to gain a better user experienc
e for the user.
Review Content Sentiment: mixed
Review Content Score: 3
Review Content: Bad experience that i cannot go through the flow of for
get password. So cannot login. Register flow found i have an account be
fore. So cannot register. Thus, i cannot use the app. Sometimes it show
s blank page or frame. Clicked the live chat icon but it showed a blank
page. I have never seen such a bad app before. Will try to login later.
Bad user experience. :(
Review Content Sentiment: negative
Review Content Score: 1
Review Content: Horrible app, slow and doesn't work for linking mpf acc
```

In [12]:

```
# Count number of review content sentiment
reviews_content_sentiment['sentiment'].value_counts()
```

Out[12]:

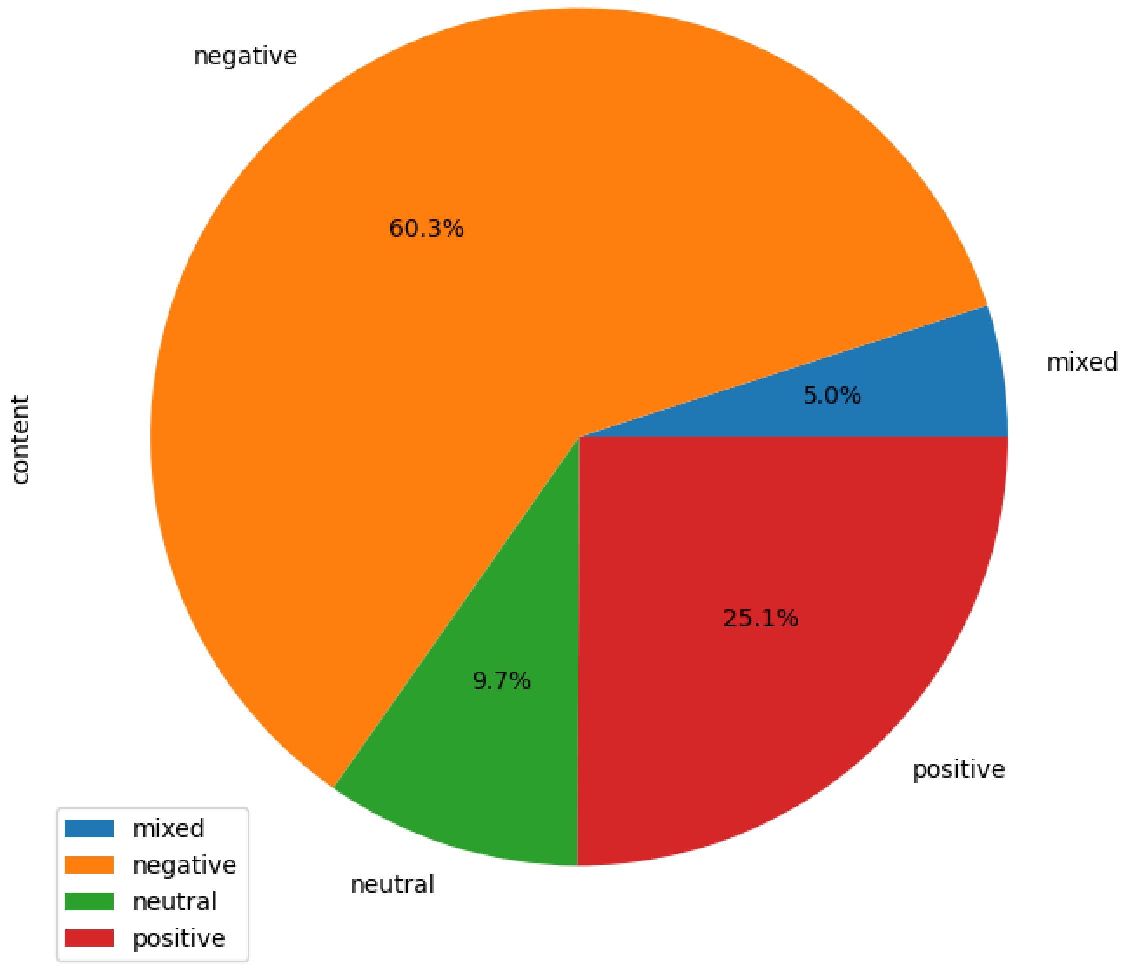
```
negative    231
positive     96
neutral      37
mixed        19
Name: sentiment, dtype: int64
```

In [13]:

```
# Visualize review content sentiment as pie chart.
reviews_content_sentiment.groupby(['sentiment']).count().plot(kind='pie', y='content', f
```

Out[13]:

```
<AxesSubplot:ylabel='content'>
```



In [14]:

```
# Group by sentiment & reviews_score
reviews_content_sentiment.groupby(["sentiment", "reviews_score"])["content"].count()
```

Out[14]:

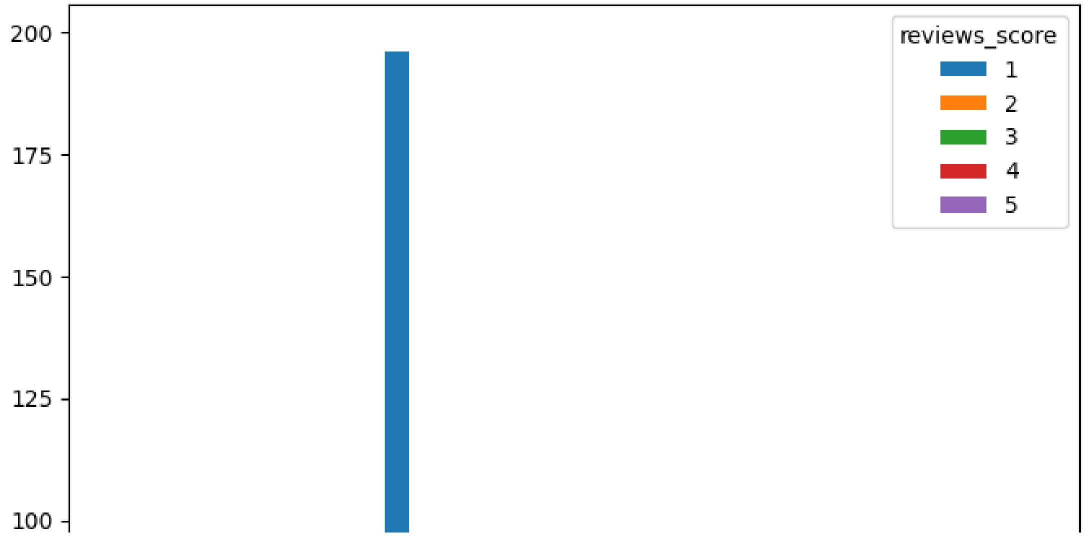
```
sentiment  reviews_score
mixed      1                 10
           2                  1
           3                  5
           4                  2
           5                  1
negative   1                196
           2                 20
           3                 14
           5                  1
neutral    1                 24
           2                  2
           3                  3
           4                  1
           5                  7
positive   1                  6
           2                  1
           3                  2
           4                 12
           5                 75
Name: content, dtype: int64
```

In [15]:

```
# Visual sentiment & reviews_score as bar chart
pd.crosstab(reviews_content_sentiment['sentiment'],reviews_content_sentiment['reviews_sc
```

Out[15]:

```
<AxesSubplot:xlabel='sentiment'>
```



From observation, neutral sentiment in review content would most likely be giving lowest review score. Let's doing some more statistical analysis below.

In [16]:

```
# Check dataframe information
reviews_content_sentiment.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 383 entries, 0 to 382
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   content        383 non-null    object
 1   sentiment      383 non-null    object
 2   reviews_score  383 non-null    object
dtypes: object(3)
memory usage: 9.1+ KB
```

In [17]:

```
# Create new dataframe to perform factorization
reviews_content_sentiment_factorized = reviews_content_sentiment.copy()
```

In [18]:

```
# Perform factorization for sentiment column
reviews_content_sentiment_factorized.sentiment = pd.factorize(reviews_content_sentiment_
```

In [19]:

```
# Convert reviews_score column data type to intager
reviews_content_sentiment_factorized['reviews_score'] = reviews_content_sentiment_factor
```

In [20]:

```
# Check dataframe information
reviews_content_sentiment_factorized.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 383 entries, 0 to 382
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   content        383 non-null    object
 1   sentiment      383 non-null    int64 
 2   reviews_score  383 non-null    int32 
dtypes: int32(1), int64(1), object(1)
memory usage: 7.6+ KB
```

In [21]:

```
# Group by sentiment (factorized) & reviews_score
reviews_content_sentiment_factorized.groupby(["sentiment", "reviews_score"])["content"].
```

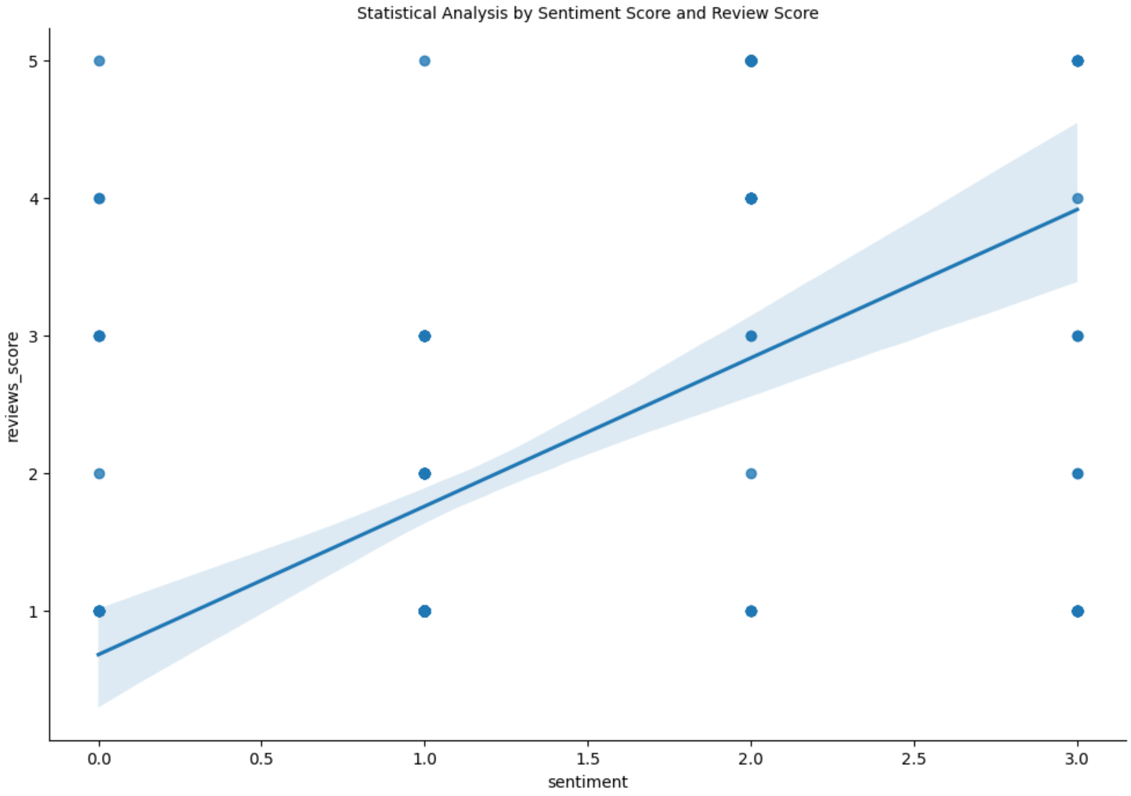
```
sentiment  reviews_score
0          1                 10
           2                  1
           3                  5
           4                  2
           5                  1
1          1                196
           2                 20
           3                 14
           5                  1
2          1                  6
           2                  1
           3                  2
           4                 12
           5                 75
3          1                 24
           2                  2
           3                  3
           4                  1
```

After factorization of sentiment column, below is numberic values the mapping.

- 0 = Neutral
- 1 = Negative
- 2 = Positive
- 3 = Mixed

In [22]:

```
# Plotting sentiment & reviews_score columns relationship by Seaborn.
fig, ax = plt.subplots()
fig.set_size_inches(12, 8)
plt.title('Statistical Analysis by Sentiment Score and Review Score', fontsize=10)
sns.regplot(x='sentiment', y= 'reviews_score', data=reviews_content_sentiment_factorized
sns.despine()
```



Data Analysis from Visualization

- Positive sentiment (2) from review content is trending to higher reivew score.
- Negative sentiment (1) from review content is trending to lower review score.
- Neutral sentiment (0) from review content is trending to lower review score.
- In other word, lower review score is trending to Neutral sentiment (0).
- From this observation, if sentiment is negative to neutral, user would give lower review score.