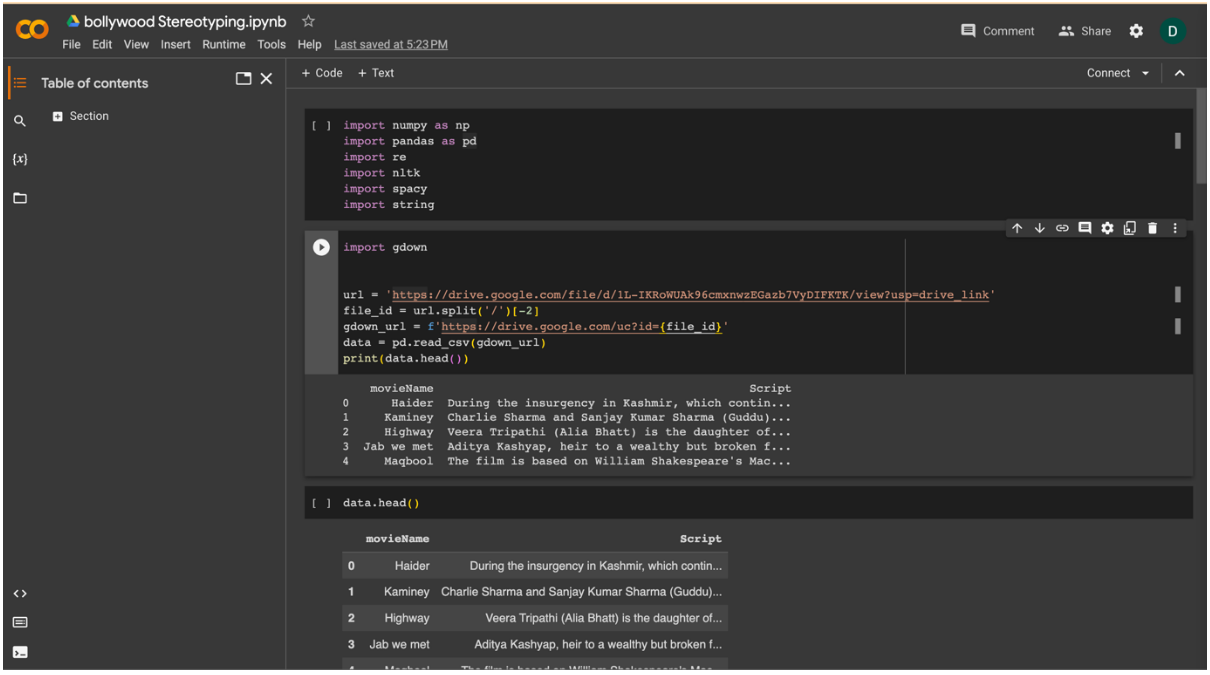
# Code Snippets

## For NLP Preprocessing


bollywood Stereotyping.ipynb
File Edit View Insert Runtime Tools Help Last saved at 5:23 PM
Comment
Share
D
Table of contents
Section
+ Code
+ Text
Connect

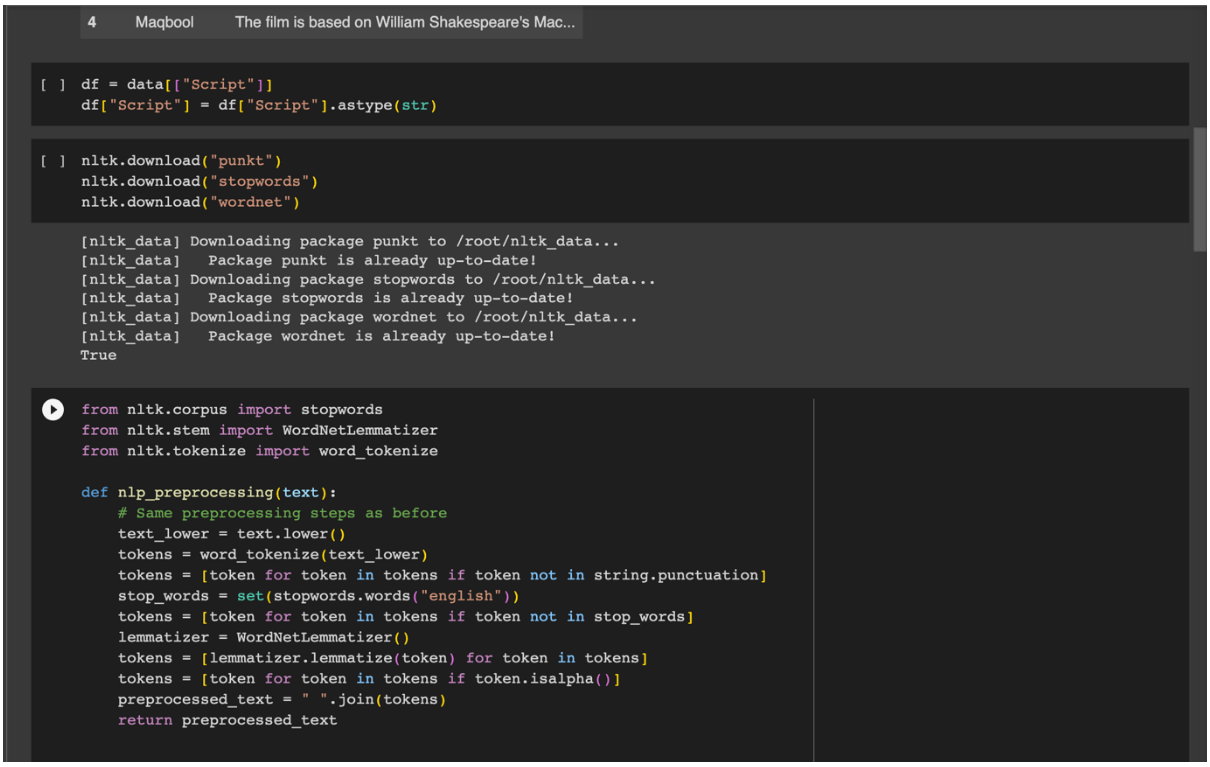
```
[ ] import numpy as np
    import pandas as pd
    import re
    import nltk
    import spacy
    import string
```

```
import gdown


url = 'https://drive.google.com/file/d/1L-IKRoWUAk96cmxnwzEGazb7VyDIFKTK/view?usp=drive_link'
file_id = url.split('/')[-2]
gdown_url = f'https://drive.google.com/uc?id={file_id}'
data = pd.read_csv(gdown_url)
print(data.head())
```

```
    movieName                                             Script
0      Haider  During the insurgency in Kashmir, which contin...
1     Kaminey  Charlie Sharma and Sanjay Kumar Sharma (Guddu)...
2     Highway  Veera Tripathi (Alia Bhatt) is the daughter of...
3  Jab we met  Aditya Kashyap, heir to a wealthy but broken f...
4     Maqbool  The film is based on William Shakespeare's Mac...
```

```
[ ] data.head()
```

|  | movieName | Script |
|---|---|---|
| 0 | Haider | During the insurgency in Kashmir, which contin... |
| 1 | Kaminey | Charlie Sharma and Sanjay Kumar Sharma (Guddu)... |
| 2 | Highway | Veera Tripathi (Alia Bhatt) is the daughter of... |
| 3 | Jab we met | Aditya Kashyap, heir to a wealthy but broken f... |
| 4 | Maqbool | The film is based on William Shakespeare's Mac... |

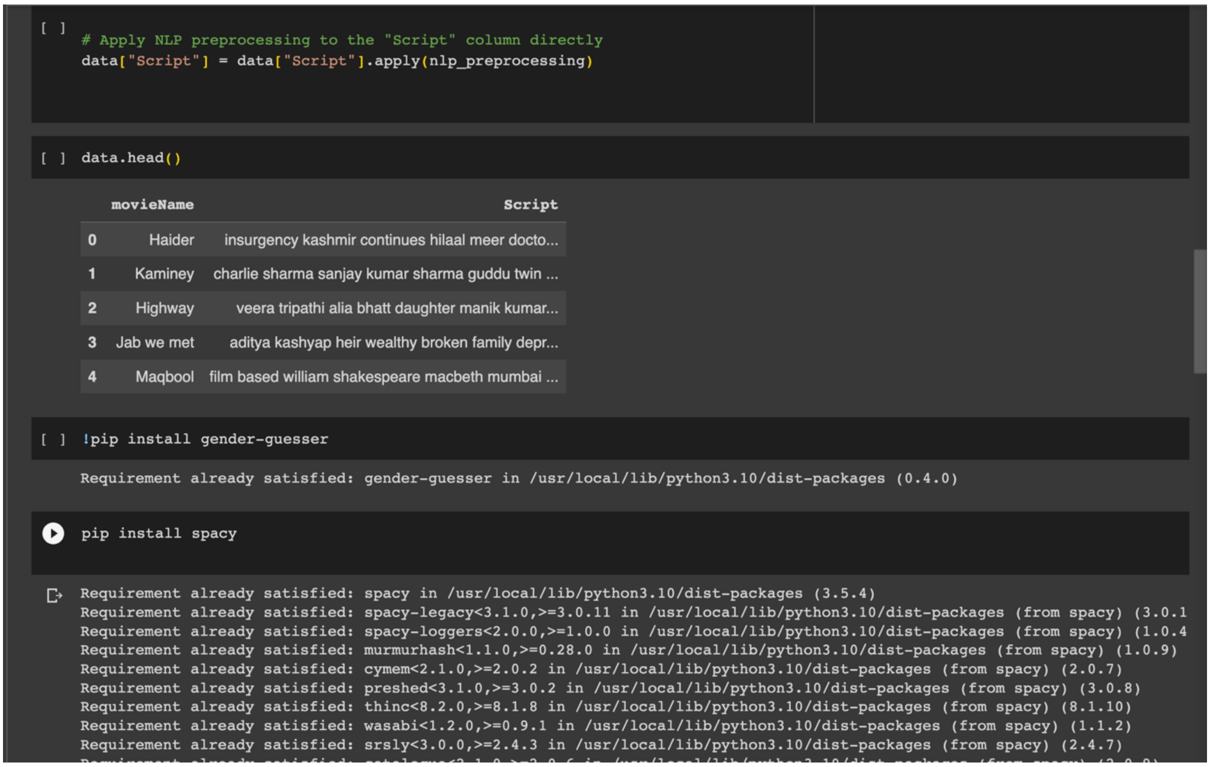


```
[ ] df = data[["Script"]]
    df["Script"] = df["Script"].astype(str)
```

```
[ ] nltk.download("punkt")
    nltk.download("stopwords")
    nltk.download("wordnet")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize

def nlp_preprocessing(text):
    # Same preprocessing steps as before
    text_lower = text.lower()
    tokens = word_tokenize(text_lower)
    tokens = [token for token in tokens if token not in string.punctuation]
    stop_words = set(stopwords.words("english"))
    tokens = [token for token in tokens if token not in stop_words]
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    tokens = [token for token in tokens if token.isalpha()]
    preprocessed_text = " ".join(tokens)
    return preprocessed_text
```



```
[ ] # Apply NLP preprocessing to the "Script" column directly
    data["Script"] = data["Script"].apply(nlp_preprocessing)
```

```
[ ] data.head()
```

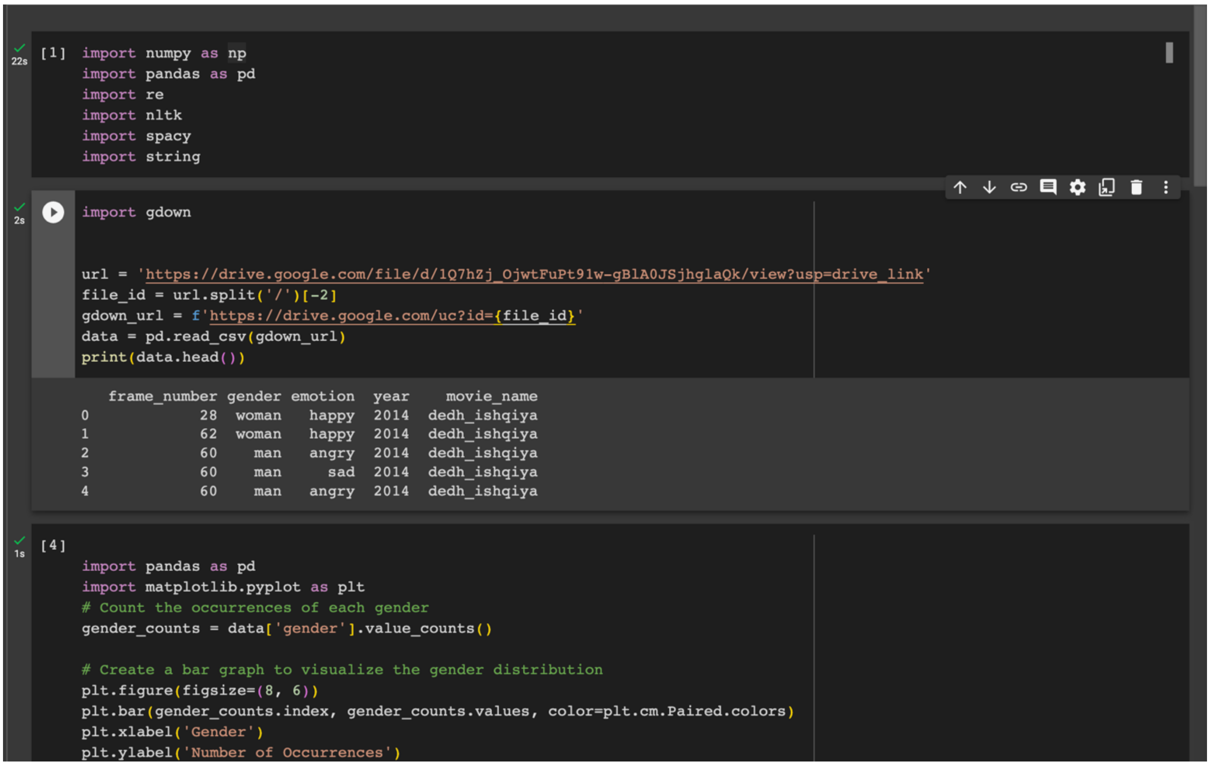
|   | movieName | Script |
|---|---|---|
| 0 | Haider | insurgency kashmir continues hilaal meer docto... |
| 1 | Kaminey | charlie sharma sanjay kumar sharma guddu twin ... |
| 2 | Highway | veera tripathi alia bhatt daughter manik kumar... |
| 3 | Jab we met | aditya kashyap heir wealthy broken family depr... |
| 4 | Maqbool | film based william shakespeare macbeth mumbai ... |

```
[ ] !pip install gender-guesser

    Requirement already satisfied: gender-guesser in /usr/local/lib/python3.10/dist-packages (0.4.0)
```

```
pip install spacy

Requirement already satisfied: spacy in /usr/local/lib/python3.10/dist-packages (3.5.4)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.1
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.4
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.9)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.7)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.8)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy) (8.1.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.4.7)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.9)
```

For Detecting the different types of emotions in Men and Women:

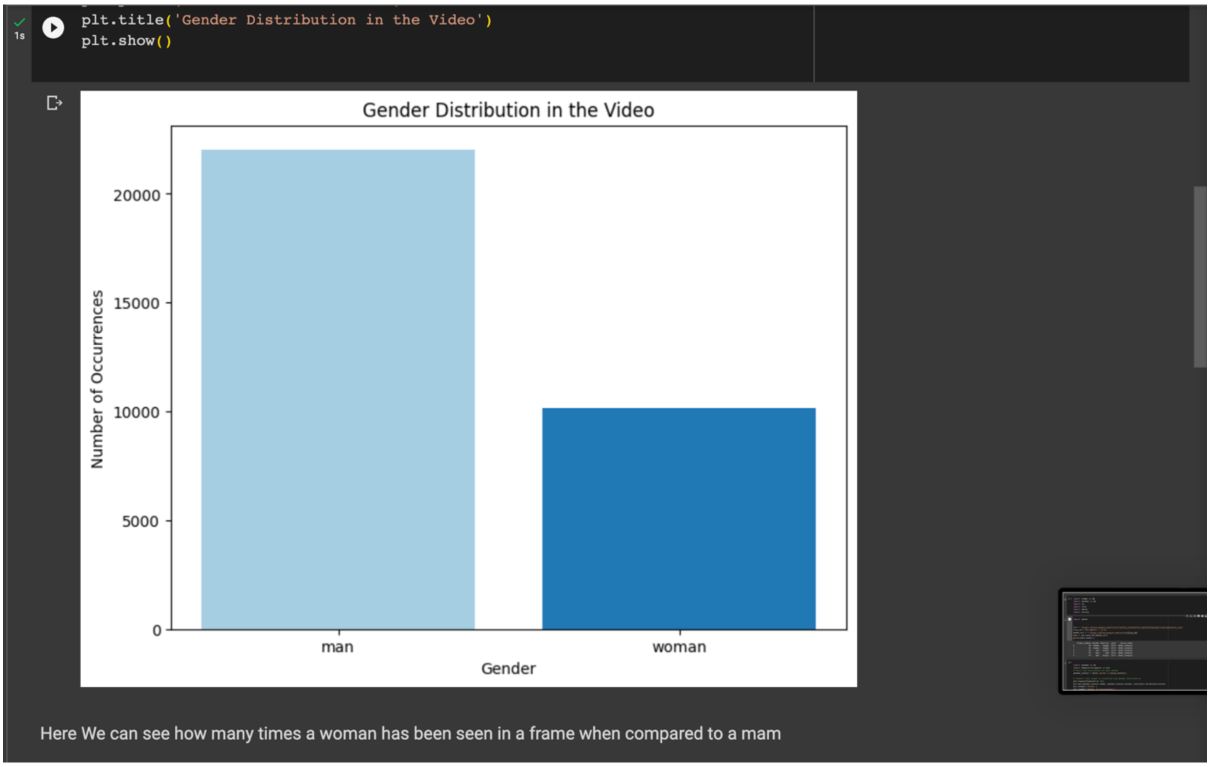
From the complete-data.csv



```
[1] import numpy as np
    import pandas as pd
    import re
    import nltk
    import spacy
    import string
```

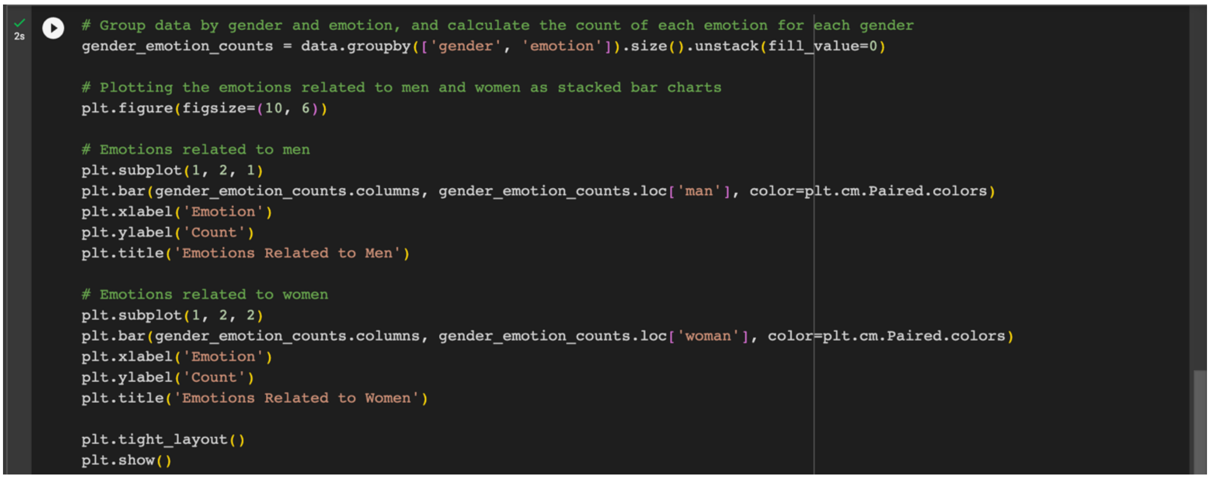
```
import gdown


url = 'https://drive.google.com/file/d/1Q7hZj_OjwtFuPt9lw-gBlA0JSjhglaQk/view?usp=drive_link'
file_id = url.split('/')[-2]
gdown_url = f'https://drive.google.com/uc?id={file_id}'
data = pd.read_csv(gdown_url)
print(data.head())

   frame_number gender emotion  year    movie_name
0            28  woman   happy  2014  dedh_ishqiya
1            62  woman   happy  2014  dedh_ishqiya
2            60    man   angry  2014  dedh_ishqiya
3            60    man     sad  2014  dedh_ishqiya
4            60    man   angry  2014  dedh_ishqiya
```

```
[4]
    import pandas as pd
    import matplotlib.pyplot as plt
    # Count the occurrences of each gender
    gender_counts = data['gender'].value_counts()

    # Create a bar graph to visualize the gender distribution
    plt.figure(figsize=(8, 6))
    plt.bar(gender_counts.index, gender_counts.values, color=plt.cm.Paired.colors)
    plt.xlabel('Gender')
    plt.ylabel('Number of Occurrences')
```

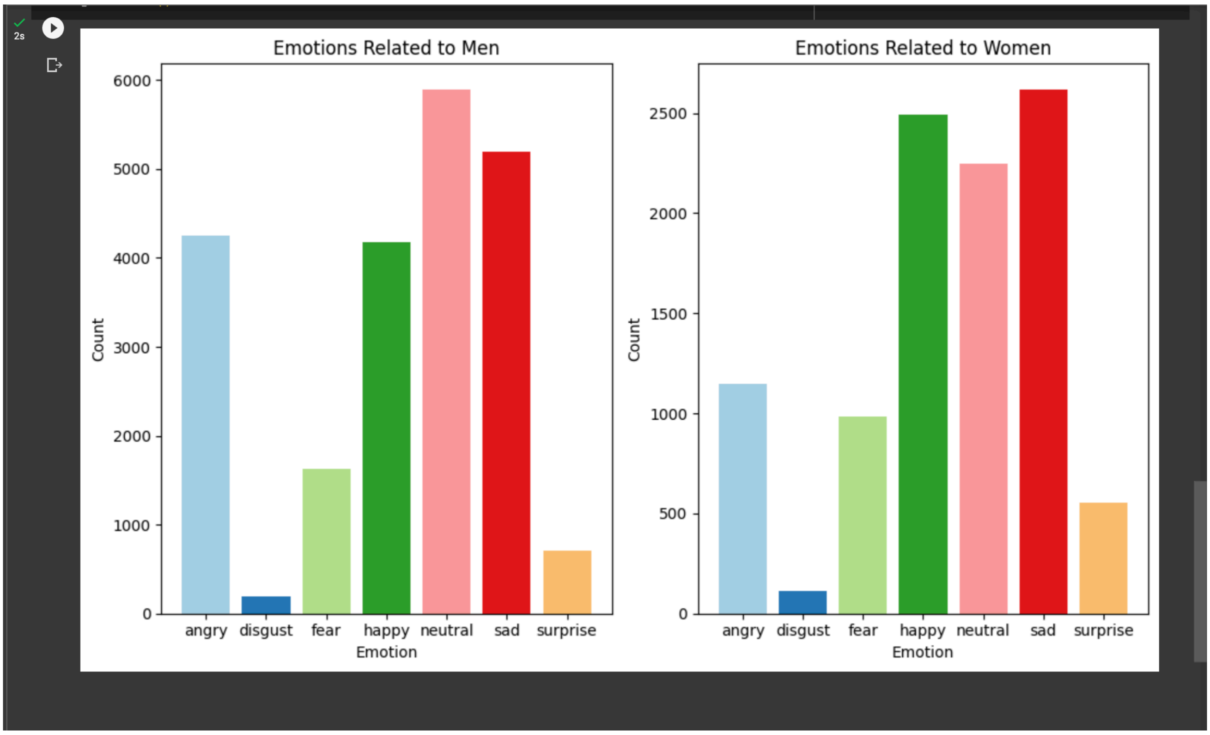


```
plt.title('Gender Distribution in the Video')
plt.show()
```

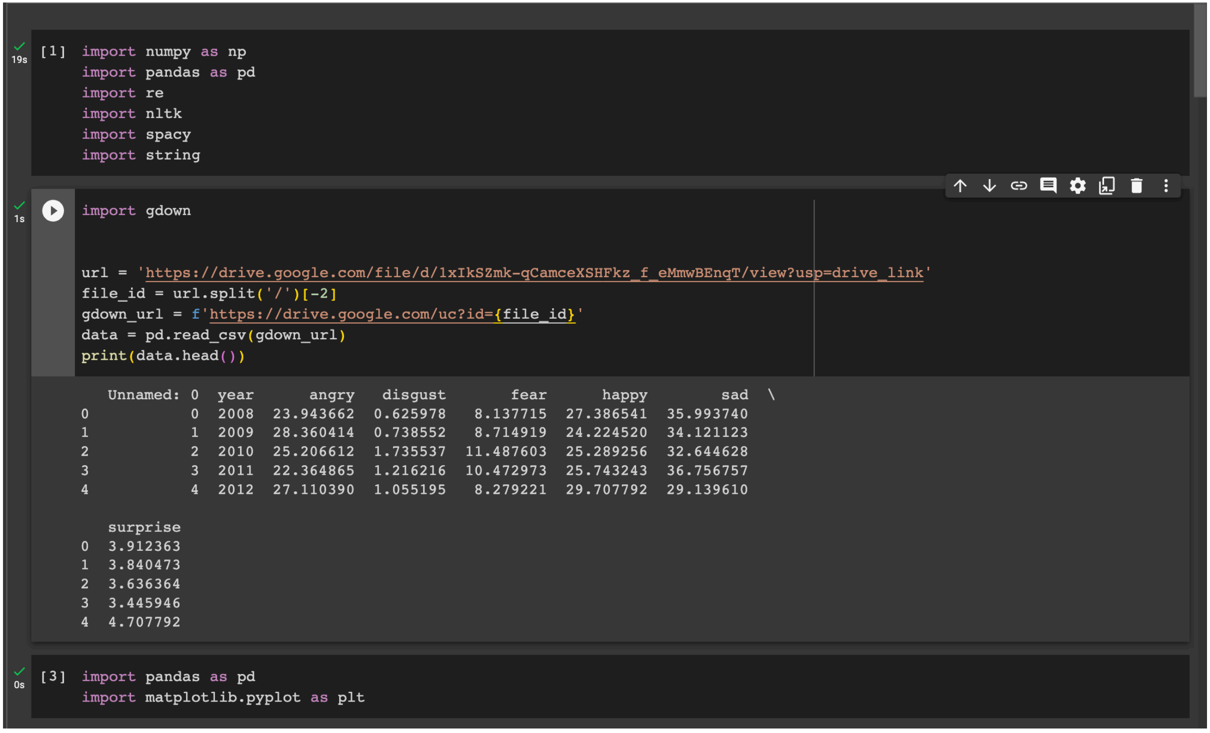
Here We can see how many times a woman has been seen in a frame when compared to a mam



```
# Group data by gender and emotion, and calculate the count of each emotion for each gender
gender_emotion_counts = data.groupby(['gender', 'emotion']).size().unstack(fill_value=0)

# Plotting the emotions related to men and women as stacked bar charts
plt.figure(figsize=(10, 6))

# Emotions related to men
plt.subplot(1, 2, 1)
plt.bar(gender_emotion_counts.columns, gender_emotion_counts.loc['man'], color=plt.cm.Paired.colors)
plt.xlabel('Emotion')
plt.ylabel('Count')
plt.title('Emotions Related to Men')

# Emotions related to women
plt.subplot(1, 2, 2)
plt.bar(gender_emotion_counts.columns, gender_emotion_counts.loc['woman'], color=plt.cm.Paired.colors)
plt.xlabel('Emotion')
plt.ylabel('Count')
plt.title('Emotions Related to Women')

plt.tight_layout()
plt.show()
```



Using the emotions of men, emotions of women data set

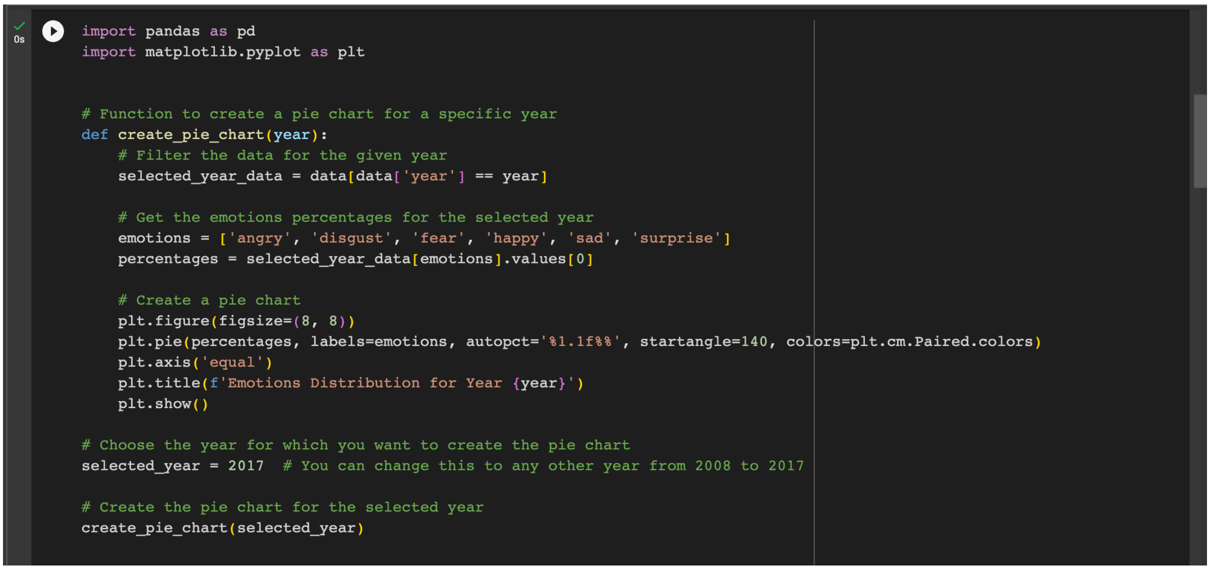


```
import numpy as np
import pandas as pd
import re
import nltk
import spacy
import string
```

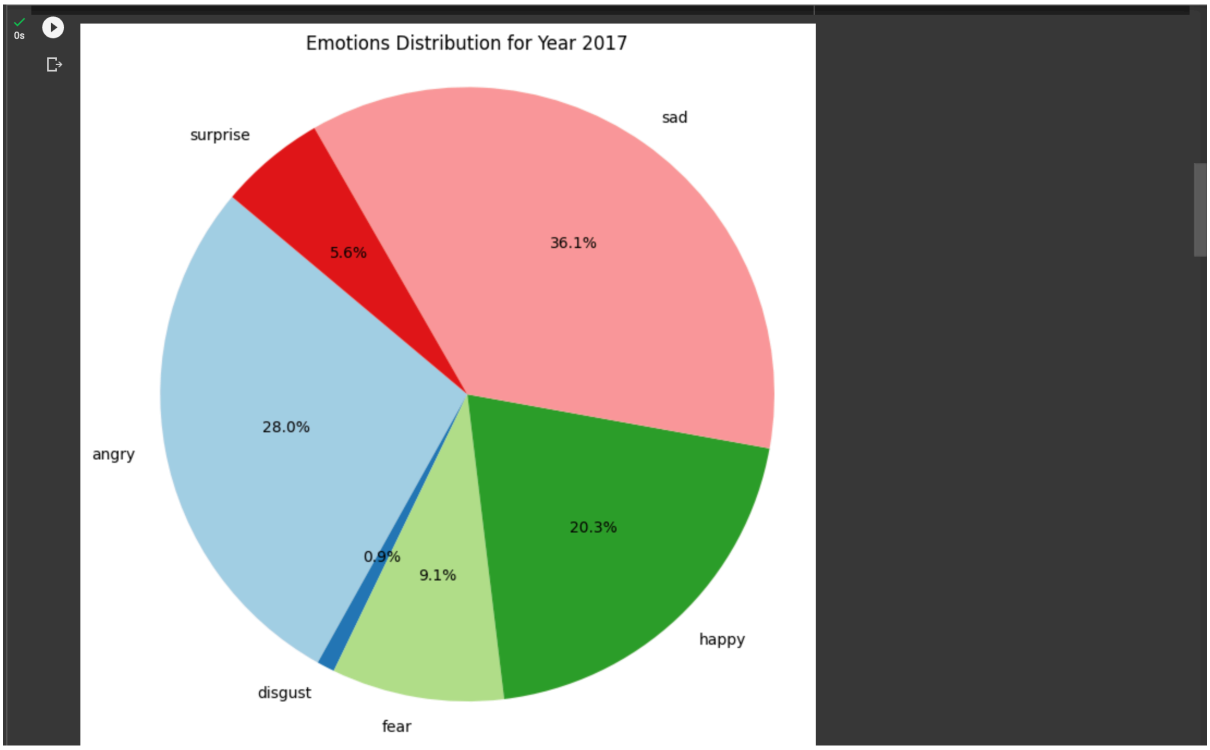
```
import gdown


url = 'https://drive.google.com/file/d/1xIkSZmk-qCamceXSHFkz_f_eMmwBEnqT/view?usp=drive_link'
file_id = url.split('/')[-2]
gdown_url = f'https://drive.google.com/uc?id={file_id}'
data = pd.read_csv(gdown_url)
print(data.head())
```

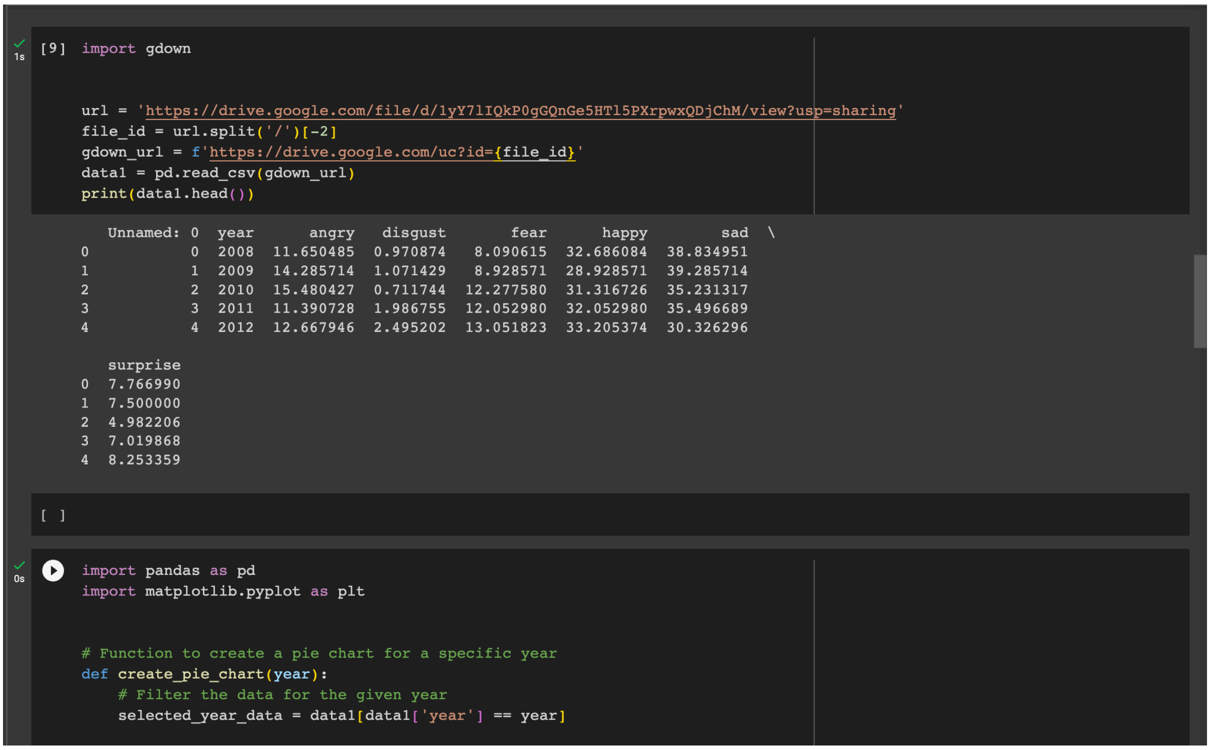
```
   Unnamed: 0  year      angry   disgust       fear      happy        sad  \
0           0  2008  23.943662  0.625978   8.137715  27.386541  35.993740
1           1  2009  28.360414  0.738552   8.714919  24.224520  34.121123
2           2  2010  25.206612  1.735537  11.487603  25.289256  32.644628
3           3  2011  22.364865  1.216216  10.472973  25.743243  36.756757
4           4  2012  27.110390  1.055195   8.279221  29.707792  29.139610

   surprise
0  3.912363
1  3.840473
2  3.636364
3  3.445946
4  4.707792
```

```
import pandas as pd
import matplotlib.pyplot as plt
```



```python
import pandas as pd
import matplotlib.pyplot as plt


# Function to create a pie chart for a specific year
def create_pie_chart(year):
    # Filter the data for the given year
    selected_year_data = data[data['year'] == year]

    # Get the emotions percentages for the selected year
    emotions = ['angry', 'disgust', 'fear', 'happy', 'sad', 'surprise']
    percentages = selected_year_data[emotions].values[0]

    # Create a pie chart
    plt.figure(figsize=(8, 8))
    plt.pie(percentages, labels=emotions, autopct='%1.1f%%', startangle=140, colors=plt.cm.Paired.colors)
    plt.axis('equal')
    plt.title(f'Emotions Distribution for Year {year}')
    plt.show()

# Choose the year for which you want to create the pie chart
selected_year = 2017  # You can change this to any other year from 2008 to 2017

# Create the pie chart for the selected year
create_pie_chart(selected_year)
```



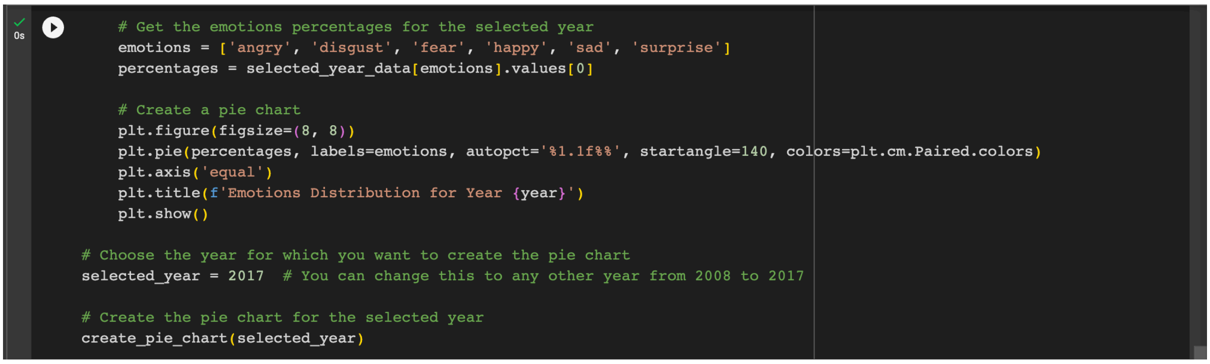


```
[9] import gdown


    url = 'https://drive.google.com/file/d/1yY7lIQkP0gGQnGe5HTl5PXrpwxQDjChM/view?usp=sharing'
    file_id = url.split('/')[-2]
    gdown_url = f'https://drive.google.com/uc?id={file_id}'
    data1 = pd.read_csv(gdown_url)
    print(data1.head())
```

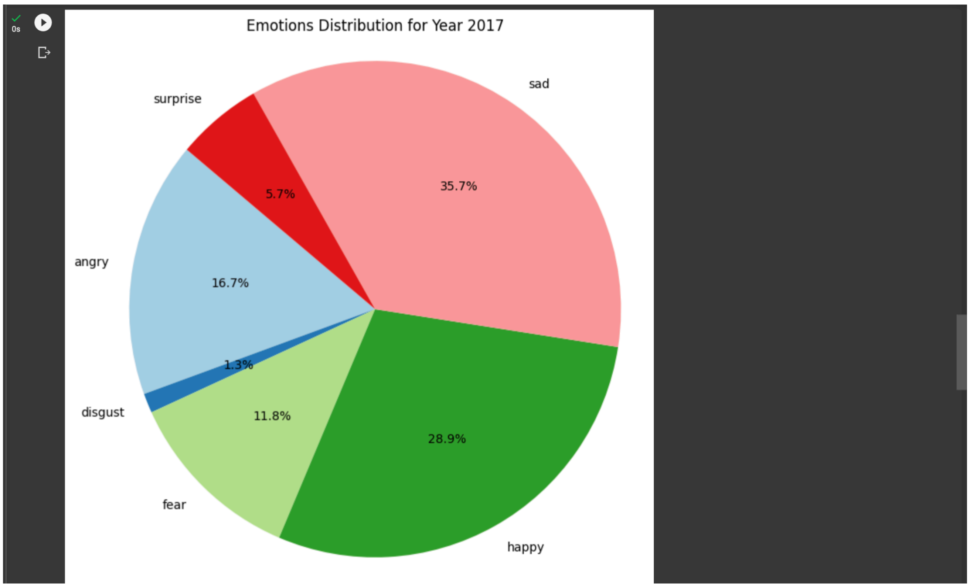
```
   Unnamed: 0  year      angry   disgust       fear      happy        sad  \
0           0  2008  11.650485  0.970874   8.090615  32.686084  38.834951
1           1  2009  14.285714  1.071429   8.928571  28.928571  39.285714
2           2  2010  15.480427  0.711744  12.277580  31.316726  35.231317
3           3  2011  11.390728  1.986755  12.052980  32.052980  35.496689
4           4  2012  12.667946  2.495202  13.051823  33.205374  30.326296

   surprise
0  7.766990
1  7.500000
2  4.982206
3  7.019868
4  8.253359
```

```
[ ]
```

```
import pandas as pd
import matplotlib.pyplot as plt


# Function to create a pie chart for a specific year
def create_pie_chart(year):
    # Filter the data for the given year
    selected_year_data = data1[data1['year'] == year]
```



```
    # Get the emotions percentages for the selected year
    emotions = ['angry', 'disgust', 'fear', 'happy', 'sad', 'surprise']
    percentages = selected_year_data[emotions].values[0]

    # Create a pie chart
    plt.figure(figsize=(8, 8))
    plt.pie(percentages, labels=emotions, autopct='%1.1f%%', startangle=140, colors=plt.cm.Paired.colors)
    plt.axis('equal')
    plt.title(f'Emotions Distribution for Year {year}')
    plt.show()

# Choose the year for which you want to create the pie chart
selected_year = 2017  # You can change this to any other year from 2008 to 2017

# Create the pie chart for the selected year
create_pie_chart(selected_year)
```



0s
Emotions Distribution for Year 2017
sad
surprise
angry
disgust
fear
happy