```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!pip install requests beautifulsoup4 pandas matplotlib wordcloud
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (2.32.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.12/dist-packages (4.13.5)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: wordcloud in /usr/local/lib/python3.12/dist-packages (1.9.5)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests) (2026.1.4)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.12/dist-packages (from beautifulsoup4) (2.8.1)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.12/dist-packages (from beautifulsoup4) (4.
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.3)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.61.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.3.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.
```

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
url = "https://www.azquotes.com/author/6560-Alfred_Hitchcock"
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")

quotes = [q.text.strip() for q in soup.find_all("a", class_="title")]
quotes[:10]
```

```
['When the power of love overcomes the love of power the world will know peace.',
 'Knowledge talks, wisdom listens.',
 'May the dreams of your past be the reality of your future.',
 'Before you start pointing fingers, make sure your hands are clean.',
 "I'm the one that's got to die when it's time for me to die, so let me live my life the way I want to.",
 'Happiness is within you... so unlock the chains from your heart and let yourself grow- like the sweet flower you are. I
know the answer- just spread your wings and set yourself free.',
 'Music is Magic. Magic is Life',
 "Technically, I'm not a guitar player, all I play is truth and emotion.",
 'Knowledge is speaking, wisdom is listening',
 "Music doesn't lie. If there is something to be changed in this world, then it can only happen through music."]
```

```
df = pd.DataFrame(quotes, columns=["Quote"])
df.head()
```

| | Quote |
|---|---|
| 0 | When the power of love overcomes the love of p... |
| 1 | Knowledge talks, wisdom listens. |
| 2 | May the dreams of your past be the reality of ... |
| 3 | Before you start pointing fingers, make sure y... |
| 4 | I'm the one that's got to die when it's time f... |

Next steps: Generate code with df | New interactive sheet

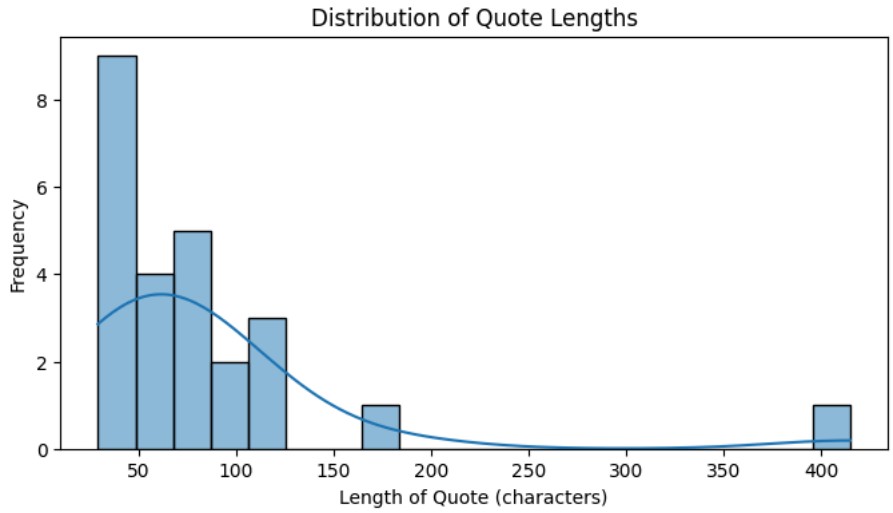
```
df['length'] = df['Quote'].str.len()
df['word_count'] = df['Quote'].str.split().str.len()
```

```
df.head()
```

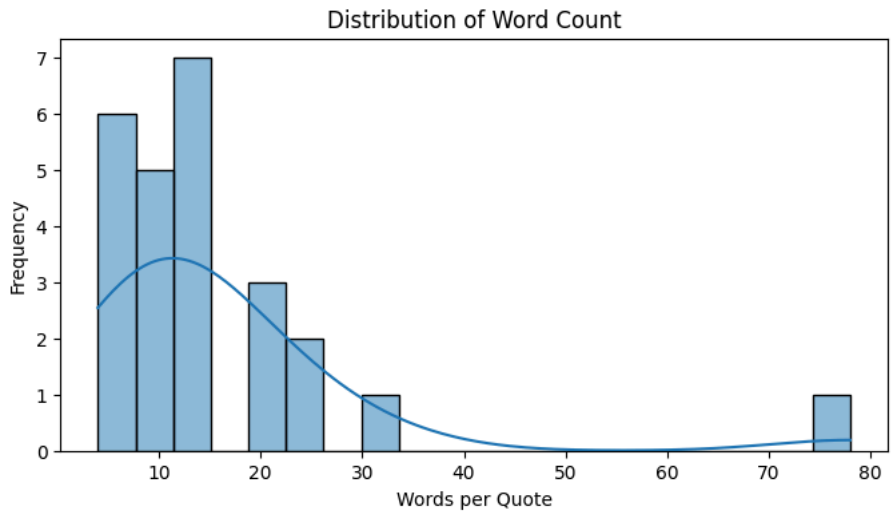
| | Quote | length | word_count |
|---|---|---|---|
| 0 | When the power of love overcomes the love of p... | 77 | 15 |
| 1 | Knowledge talks, wisdom listens. | 32 | 4 |
| 2 | May the dreams of your past be the reality of ... | 58 | 12 |
| 3 | Before you start pointing fingers, make sure y... | 66 | 11 |
| 4 | I'm the one that's got to die when it's time f... | 101 | 25 |

Next steps: Generate code with df | New interactive sheet

```
plt.figure(figsize=(8,4))
sns.histplot(df['length'], bins=20, kde=True)
plt.title("Distribution of Quote Lengths")
plt.xlabel("Length of Quote (characters)")
plt.ylabel("Frequency")
plt.show()
```




```
plt.figure(figsize=(8,4))
sns.histplot(df['word_count'], bins=20, kde=True)
plt.title("Distribution of Word Count")
plt.xlabel("Words per Quote")
plt.ylabel("Frequency")
plt.show()
```

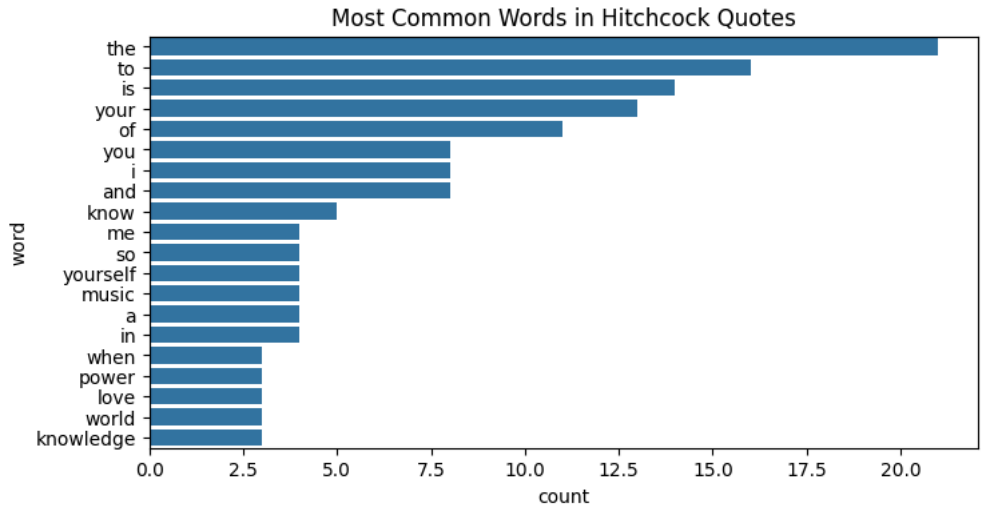


```
import re
from collections import Counter

words = " ".join(df['Quote']).lower()
words = re.sub(r'[^a-z\s]', '', words)
word_list = words.split()
common = Counter(word_list).most_common(20)
common
```

```
[('the', 21),
 ('to', 16),
 ('is', 14),
 ('your', 13),
 ('of', 11),
 ('you', 8),
 ('i', 8),
 ('and', 8),
 ('know', 5),
 ('me', 4),
 ('so', 4),
 ('yourself', 4),
 ('music', 4),
 ('a', 4),
 ('in', 4),
 ('when', 3),
 ('power', 3),
 ('love', 3),
 ('world', 3),
 ('knowledge', 3)]
```

```
words_df = pd.DataFrame(common, columns=['word','count'])
plt.figure(figsize=(8,4))
sns.barplot(x='count', y='word', data=words_df)
plt.title("Most Common Words in Hitchcock Quotes")
plt.show()
```




```
import os
os.makedirs("reports", exist_ok=True)
```

```
with open("reports/insights.txt","w") as f:
    f.write("QUOTES ANALYSIS INSIGHTS\n")
    f.write(f"Total quotes: {len(df)}\n")
    f.write(f"Average length: {df['length'].mean():.2f}\n")
    f.write(f"Average word count: {df['word_count'].mean():.2f}\n")
```

```
import os
os.makedirs("data", exist_ok=True)
os.makedirs("reports", exist_ok=True)
os.makedirs("images", exist_ok=True)
```

```
df.to_csv("data/hitchcock_quotes.csv", index=False)
```

```
df.columns
```

```
Index(['Quote', 'length', 'word_count', 'theme'], dtype='object')
```

```
col = df.columns[0]   # first column quote hoga

themes = {
    "fear": ["fear", "scared", "terror"],
    "love": ["love", "heart"],
    "life": ["life", "living"],
    "wisdom": ["knowledge", "wise", "truth"],
    "art": ["art", "music", "painting"]
}

theme_counts = {t:0 for t in themes}

df['theme'] = ""

for i, q in enumerate(df[col]):
    q_low = q.lower()
    assigned = False
    for theme, keys in themes.items():
        for k in keys:
            if k in q_low:
                df.at[i, 'theme'] = theme
                theme_counts[theme] += 1
                assigned = True
                break
        if assigned:
            break
    if not assigned:
        df.at[i, 'theme'] = "other"

theme_counts
```

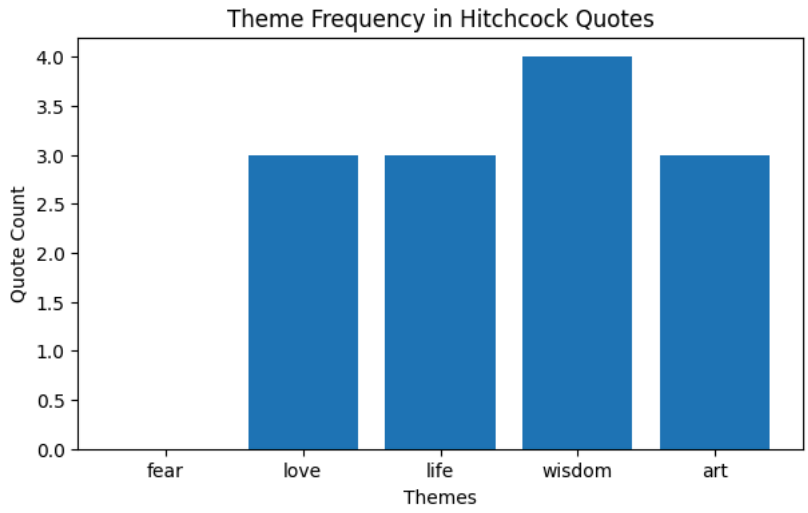
```
{'fear': 0, 'love': 3, 'life': 3, 'wisdom': 4, 'art': 3}
```

```
df.to_csv('/content/data/hitchcock_quotes_themes.csv', index=False)
```

```
import matplotlib.pyplot as plt

labels = list(theme_counts.keys())
values = list(theme_counts.values())

plt.figure(figsize=(7,4))
plt.bar(labels, values)
plt.title("Theme Frequency in Hitchcock Quotes")
plt.xlabel("Themes")
plt.ylabel("Quote Count")
plt.show()
```




```
with open('/content/reports/insights.txt', 'w') as f:
    f.write("Hitchcock Quotes Analysis - Insights\n\n")
    f.write(f"Total Quotes: {len(df)}\n")
    f.write("\nTheme counts:\n")
    for k,v in theme_counts.items():
        f.write(f"- {k}: {v}\n")
```

```
blog = """
# Hitchcock Quotes Analysis (Minor-II)

This project analyzes quotes from Alfred Hitchcock and categorizes them into themes such as Love, Wisdom, Life, Art, etc.

### Key Findings
- Wisdom is the most dominant theme.
- Life and Art themes also appear frequently.
- Love theme is present in emotional/philosophical context.
- Other quotes fall under miscellaneous category.

### Tools Used
- Python (Requests, BeautifulSoup, Pandas, Matplotlib)
- Google Colab
- Jupyter Notebook

Submitted by:
Alok Kumar
B.Tech CSE (AI)
Rungta College of Engineering & Technology
"""

with open('/content/reports/blog.md', 'w') as f:
    f.write(blog)
```

```
import os

os.makedirs("/content/images", exist_ok=True)
os.makedirs("/content/data", exist_ok=True)
os.makedirs("/content/reports", exist_ok=True)

print("Folders created successfully!")
```

```
Folders created successfully!
```

```
df.to_csv('/content/data/hitchcock_quotes.csv', index=False)
print("CSV saved!")
```

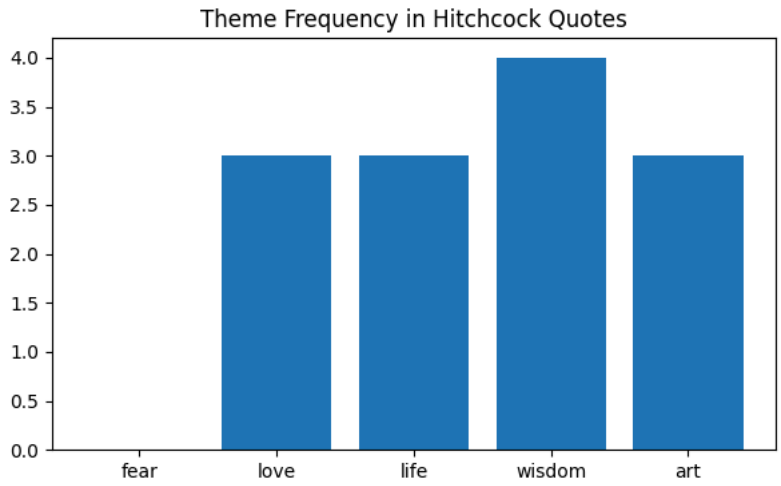
```
CSV saved!
```

```
import matplotlib.pyplot as plt

labels = list(theme_counts.keys())
values = list(theme_counts.values())

plt.figure(figsize=(7,4))
plt.bar(labels, values)
plt.title("Theme Frequency in Hitchcock Quotes")
plt.savefig('/content/images/theme_graph.png')
plt.show()

print("Graph saved!")
```




```
Graph saved!
```

```
with open('/content/reports/insights.txt', 'w') as f:
    f.write(f"Total Quotes: {len(df)}\n")
    f.write("Theme Counts:\n")
    for k,v in theme_counts.items():
```

```
            f.write(f"- {k}: {v}\n")

    print("Report saved!")
```

```
Report saved!
```

```
!ls /content/data
!ls /content/images
!ls /content/reports
```

```
hitchcock_quotes.csv  hitchcock_quotes_themes.csv
theme_graph.png
blog.md  insights.txt
```