```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
```

Load the dataset

```
df = pd.read_csv("/content/news_summary.csv", encoding='latin-1')
```

View basic structure

Double-click (or enter) to edit

```
print("Dataset shape:", df.shape)
print(df.head())
```

```
Dataset shape: (4514, 6)
               author                  date  \
0        Chhavi Tyagi  03 Aug 2017,Thursday
1         Daisy Mowke  03 Aug 2017,Thursday
2      Arshiya Chopra  03 Aug 2017,Thursday
3       Sumedha Sehra  03 Aug 2017,Thursday
4  Aarushi Maheshwari  03 Aug 2017,Thursday

                                           headlines  \
0  Daman & Diu revokes mandatory Rakshabandhan in...
1  Malaika slams user who trolled her for 'divorc...
2  'Virgin' now corrected to 'Unmarried' in IGIMS...
3  Aaj aapne pakad liya: LeT man Dujana before be...
4  Hotel staff to get training to spot signs of s...

                                           read_more  \
0  http://www.hindustantimes.com/india-news/raksh...
1  http://www.hindustantimes.com/bollywood/malaik...
2  http://www.hindustantimes.com/patna/bihar-igim...
3  http://indiatoday.intoday.in/story/abu-dujana-...
4  http://indiatoday.intoday.in/story/sex-traffic...

                                                text  \
0  The Administration of Union Territory Daman an...
1  Malaika Arora slammed an Instagram user who tr...
2  The Indira Gandhi Institute of Medical Science...
3  Lashkar-e-Taiba's Kashmir commander Abu Dujana...
4  Hotels in Maharashtra will train their staff t...

                                               ctext
0  The Daman and Diu administration on Wednesday ...
1  From her special numbers to TV?appearances, Bo...
2  The Indira Gandhi Institute of Medical Science...
3  Lashkar-e-Taiba's Kashmir commander Abu Dujana...
4  Hotels in Mumbai and other Indian cities are t...
```

Data types and null values

Duplicate records

Unique headline count

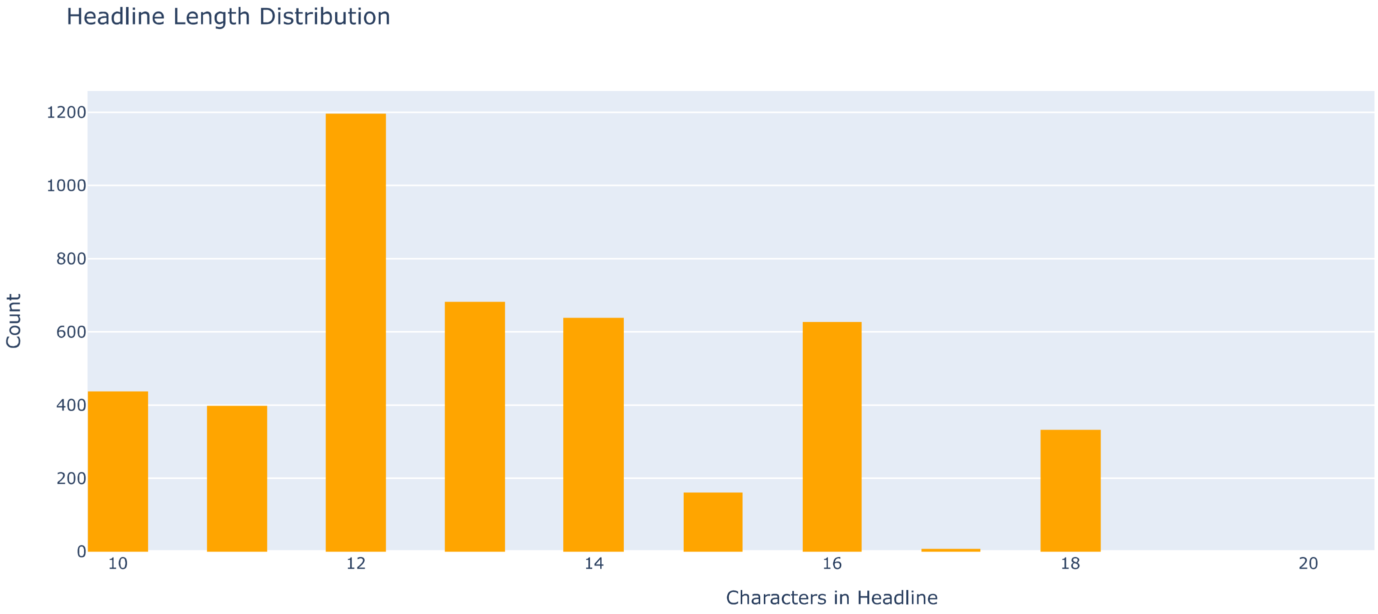
```
print(df.columns)
df.columns = ['headlines','text','column1','column2','column3','column4']

print("Data Types:\n", df.dtypes)
print("Missing Values:\n", df.isnull().sum())

print("Duplicate Rows:", df.duplicated().sum())
print("Unique Headlines:", df['headlines'].nunique())
```

```
Index(['headlines', 'text', 'column1', 'column2', 'column3', 'column4'], dtype='object')
Data Types:
 headlines    object
text         object
column1      object
column2      object
column3      object
column4      object
dtype: object
Missing Values:
 headlines      0
text           0
column1        0
column2        0
column3        0
column4      118
dtype: int64
Duplicate Rows: 0
Unique Headlines: 45
```

Headline Length Histogram

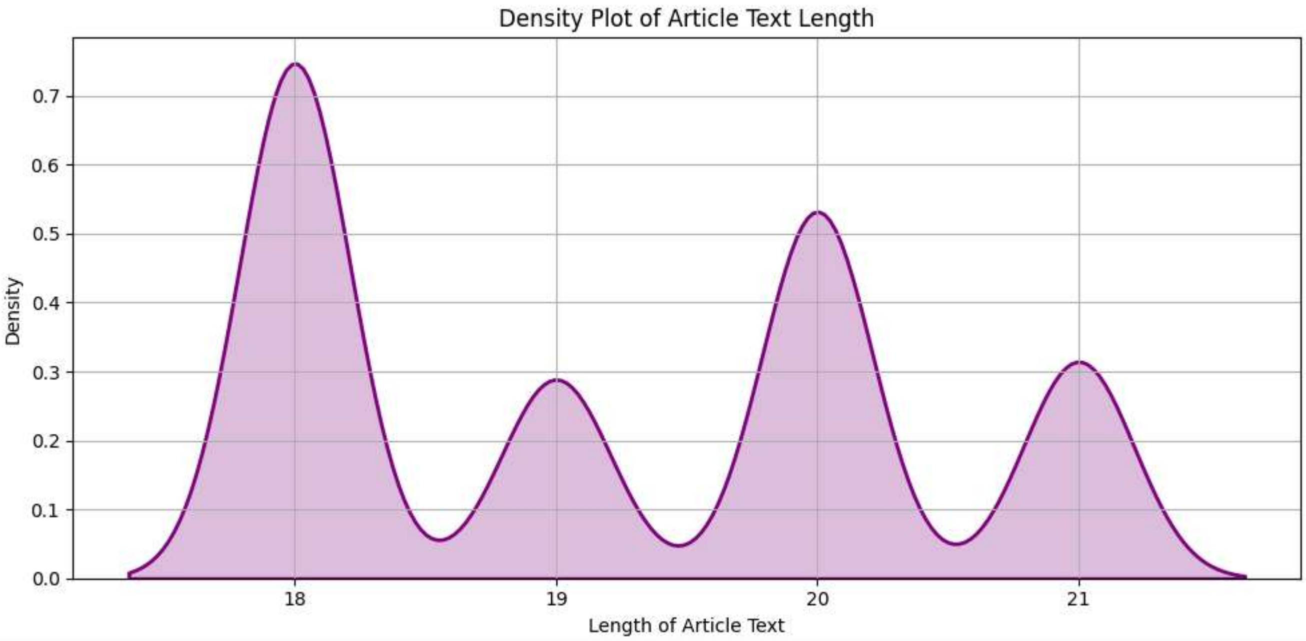
```
df['headline_length'] = df['headlines'].astype(str).apply(len)
df['headline_length'] = df['headlines'].astype(str).apply(len)
df['text_length'] = df['text'].astype(str).apply(len)
df['word_count'] = df['text'].astype(str).apply(lambda x: len(x.split()))
print(df[['headline_length', 'text_length', 'word_count']].describe())
fig1 = px.histogram(df, x='headline_length', nbins=40, title="Headline Length Distribution", color_discrete_sequence=['orange'])
fig1.update_layout(xaxis_title="Characters in Headline", yaxis_title="Count")
fig1.show()
```

```
       headline_length  text_length  word_count
count      4514.000000  4514.000000      4514.0
mean         13.346699    19.219761         3.0
std           2.333411     1.140088         0.0
min          10.000000    18.000000         3.0
25%          12.000000    18.000000         3.0
50%          13.000000    19.000000         3.0
75%          15.000000    20.000000         3.0
max          22.000000    21.000000         3.0
```



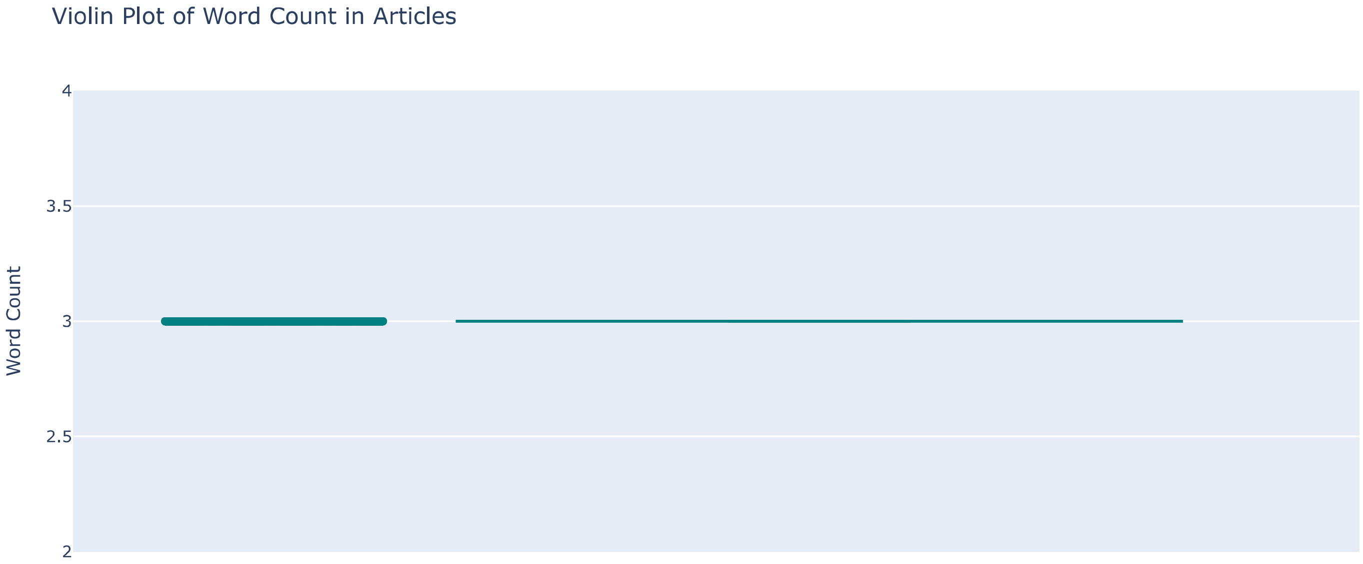
Article Text Length using KDE Curve

```
plt.figure(figsize=(10,5))
sns.kdeplot(df['text_length'], fill=True, color="purple", linewidth=2)
plt.title("Density Plot of Article Text Length")
plt.xlabel("Length of Article Text")
plt.ylabel("Density")
plt.grid(True)
plt.tight_layout()
plt.show()
```



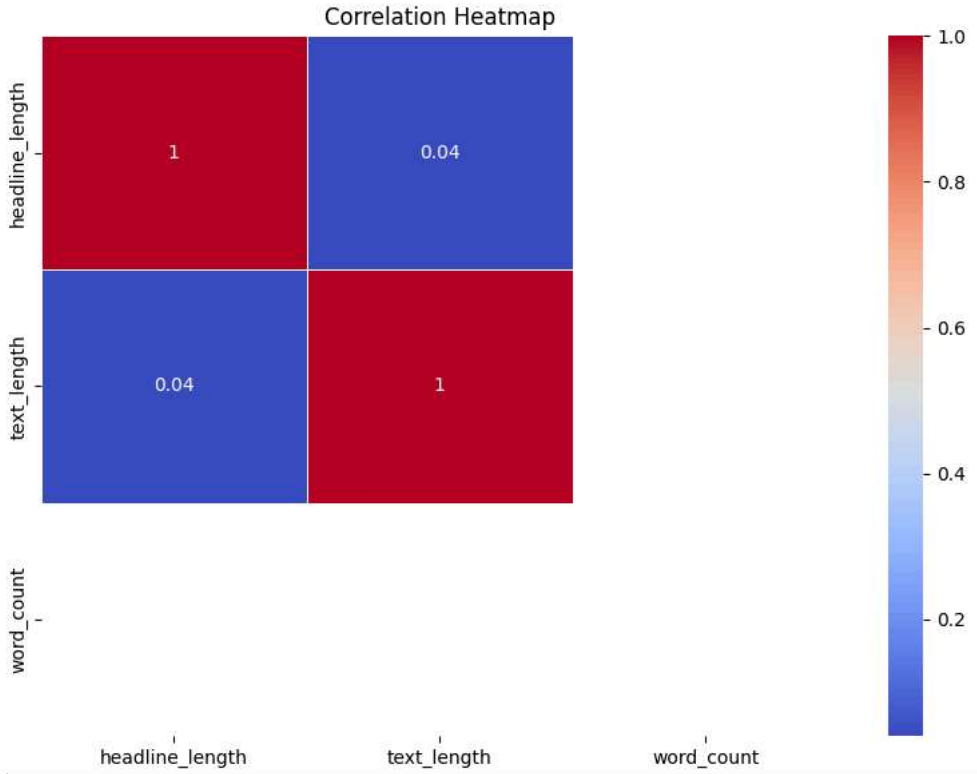
Word Count using Violin Plot

```
fig2 = px.violin(df, y='word_count', box=True, points="all", title="Violin Plot of Word Count in Articles", color_discrete_sequence=['teal']
fig2.update_layout(yaxis_title="Word Count")
fig2.show()
```



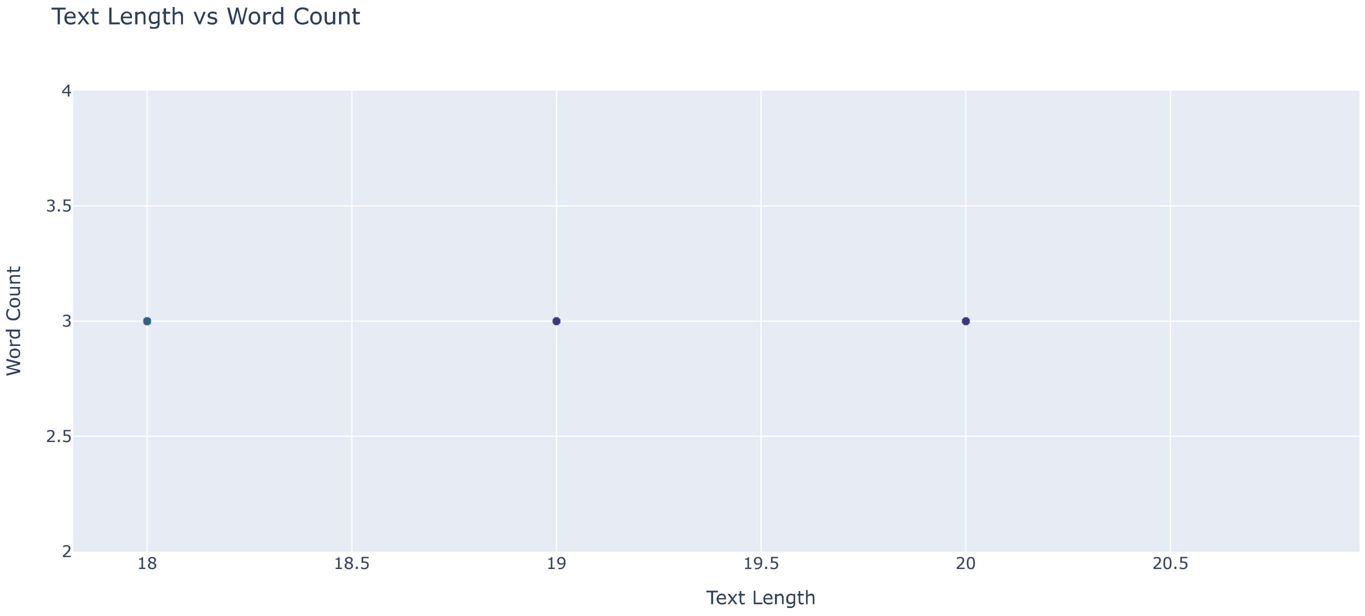
Correlation Matrix with Heatmap

```
correlation_data = df[['headline_length', 'text_length', 'word_count']].corr()
plt.figure(figsize=(8,6))
sns.heatmap(correlation_data, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Heatmap")
plt.tight_layout()
plt.show()
```



Scatter plot of Text Length vs Word Count

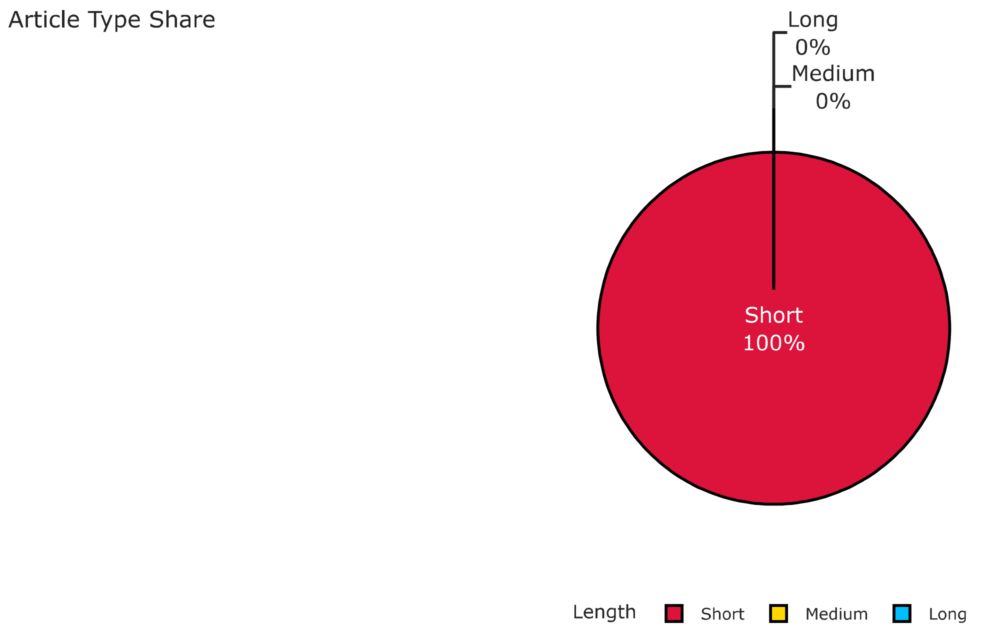
```
fig3 = px.scatter(df, x='text_length', y='word_count', title="Text Length vs Word Count", color='headline_length', color_continuous_scale='V
fig3.update_layout(xaxis_title="Text Length", yaxis_title="Word Count")
fig3.show()
```



Double-click (or enter) to edit

Pie chart of Short vs Medium vs Long Articles

```
df['length_type'] = pd.cut(df['text_length'], bins=[0,500,1500,np.inf], labels=['Short','Medium','Long'])
x = df['length_type'].value_counts().reset_index()
x.columns = ['type','count']
import plotly.express as px
figg = px.pie(x, values='count', names='type', title='Article Type Share',
              color_discrete_sequence=['crimson','gold','deepskyblue'])
figg.update_traces(pull=[0.1,0.12,0.15], hoverinfo='label+percent+value', textinfo='label+percent',
                   textfont_size=16, marker=dict(line=dict(color='#000000', width=2)))
figg.update_layout(template='simple_white',legend_title_text='Length',legend=dict(orientation="h",x=0.3,y=-0.2))
figg.show()
```


Article Type Share
Long
Medium
Short
Length
🟥 Short
🟨 Medium
🟦 Long