```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
df = pd.read_csv('husband_wife_arguments.csv')
```

```
df.head()
```

	Argument ID	Date	Duration (minutes)	Topic	Emotional Tone	Resolution Type	Mood Swing	Notes
0	1	2024-01-05	43	Parenting Styles	Disappointed	No Resolution	Mild	Discussed parenting styles for 43 minutes. Emo...
1	2	2024-02-05	24	Parenting Styles	Annoyed	Compromise	Severe	Discussed parenting styles for 24 minutes. Emo...
2	3	2024-02-27	25	Money Management	Annoyed	No Resolution	Moderate	Discussed money management for 25 minutes. Emo...
3	4	2024-02-07	20	Time Management	Angry	Resolution Discussed	Moderate	Discussed time management for 20 minutes. Emot...
4	5	2024-01-26	7	Future Goals	Frustrated	Resolution Discussed	Mild	Discussed future goals for 7 minutes. Emotiona...

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.tail()
```

	Argument ID	Date	Duration (minutes)	Topic	Emotional Tone	Resolution Type	Mood Swing	Notes
1495	1496	2024-01-02	43	Parenting Styles	Calm	Compromise	Mild	Discussed parenting styles for 43 minutes. Emo...
1496	1497	2024-01-15	22	Money Management	Supportive	Apology	NaN	Discussed money management for 22 minutes. Emo...
1497	1498	2024-01-21	16	Social Activities	Happy	Resolution Discussed	Mild	Discussed social activities for 16 minutes. Em...
1498	1499	2024-01-26	12	Social Activities	Annoyed	Compromise	Mild	Discussed social activities for 12 minutes. Em...
1499	1500	2024-02-22	51	Time Management	Calm	No Resolution	NaN	Discussed time management for 51 minutes. Emot...

```
df.shape
```

```
(1500, 8)
```

```
df.columns
```

```
Index(['Argument ID', 'Date', 'Duration (minutes)', 'Topic', 'Emotional Tone',
       'Resolution Type', 'Mood Swing', 'Notes'],
      dtype='object')
```

```
df = df.drop('Argument ID', axis = 1)
```

```
df.duplicated().sum()
```

```
0
```

```
df.isnull().sum()
```

	0
Date	0
Duration (minutes)	0
Topic	0
Emotional Tone	0
Resolution Type	0
Mood Swing	336
Notes	0

dtype: int64

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Date                1500 non-null   object
 1   Duration (minutes)  1500 non-null   int64 
 2   Topic               1500 non-null   object
 3   Emotional Tone      1500 non-null   object
 4   Resolution Type     1500 non-null   object
 5   Mood Swing          1164 non-null   object
 6   Notes               1500 non-null   object
dtypes: int64(1), object(6)
memory usage: 82.2+ KB
```

```
df.describe()
```

	Duration (minutes)
count	1500.000000
mean	32.763333
std	16.379607
min	5.000000
25%	18.000000
50%	33.000000
75%	47.000000
max	60.000000

```
df['Mood Swing'].fillna('Cant Say', inplace=True)
```

```
df.nunique()
```

	0
Date	61
Duration (minutes)	56
Topic	8
Emotional Tone	8
Resolution Type	4
Mood Swing	4
Notes	1484

dtype: int64

```
object_columns = df.select_dtypes(include=['object']).columns
print("Object type columns:")
print(object_columns)

numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns
print("\nNumerical type columns:")
print(numerical_columns)
```

```
Object type columns:
Index(['Date', 'Topic', 'Emotional Tone', 'Resolution Type', 'Mood Swing',
       'Notes'],
      dtype='object')

Numerical type columns:
Index(['Duration (minutes)'], dtype='object')
```

```
def classify_features(df):
    categorical_features = []
    non_categorical_features = []
    discrete_features = []
    continuous_features = []

    for column in df.columns:
        if df[column].dtype == 'object':
            if df[column].nunique() < 10:
                categorical_features.append(column)
            else:
                non_categorical_features.append(column)
        elif df[column].dtype in ['int64', 'float64']:
            if df[column].nunique() < 10:
                discrete_features.append(column)
            else:
                continuous_features.append(column)
    return categorical_features, non_categorical_features, discrete_features, continuous_features
```

```
categorical, non_categorical, discrete, continuous = classify_features(df)
```

```
print("Categorical Features:", categorical)
print("Non-Categorical Features:", non_categorical)
print("Discrete Features:", discrete)
print("Continuous Features:", continuous)
```

```
Categorical Features: ['Topic', 'Emotional Tone', 'Resolution Type', 'Mood Swing']
Non-Categorical Features: ['Date', 'Notes']
Discrete Features: []
Continuous Features: ['Duration (minutes)']
```

```
for i in categorical:
    print(i)
    print(df[i].unique())
    print()
```

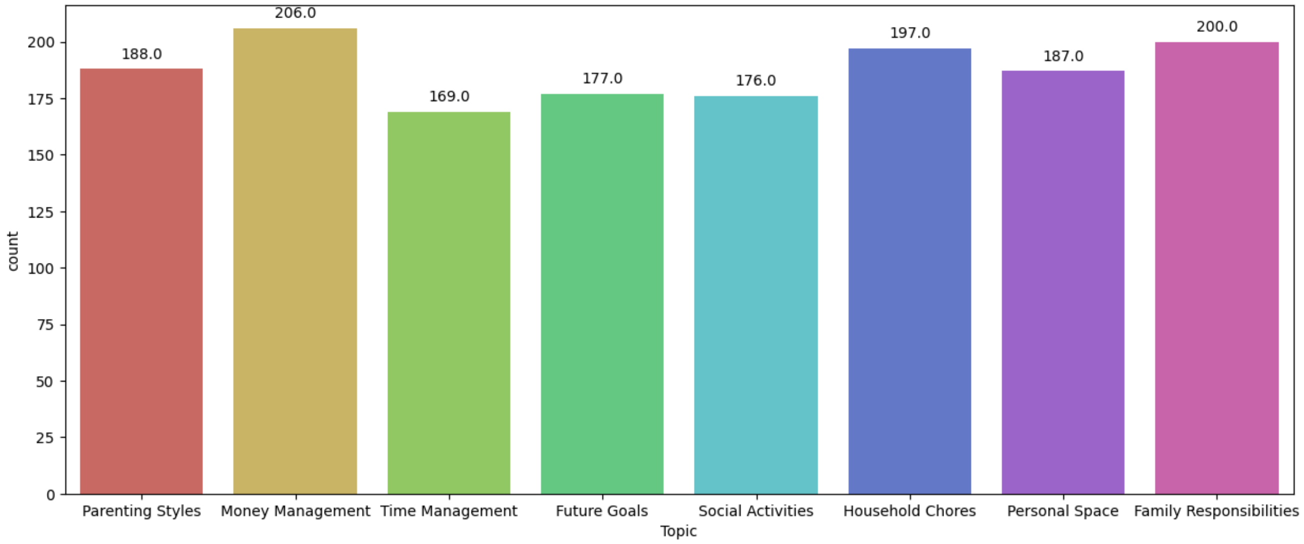
```
Topic
['Parenting Styles' 'Money Management' 'Time Management' 'Future Goals'
 'Social Activities' 'Household Chores' 'Personal Space'
 'Family Responsibilities']

Emotional Tone
['Disappointed' 'Annoyed' 'Angry' 'Frustrated' 'Happy' 'Supportive' 'Calm'
 'Hurt']

Resolution Type
['No Resolution' 'Compromise' 'Resolution Discussed' 'Apology']

Mood Swing
['Mild' 'Severe' 'Moderate' 'Cant Say']
```

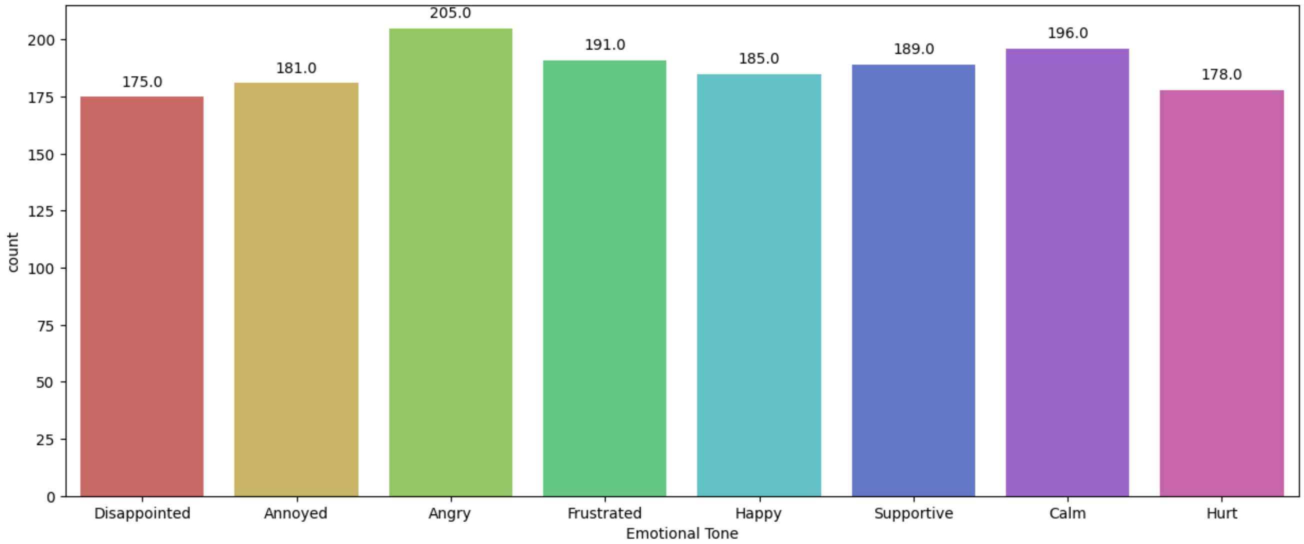
```
for i in categorical:
    print(i)
    print(df[i].value_counts())
    print()
```

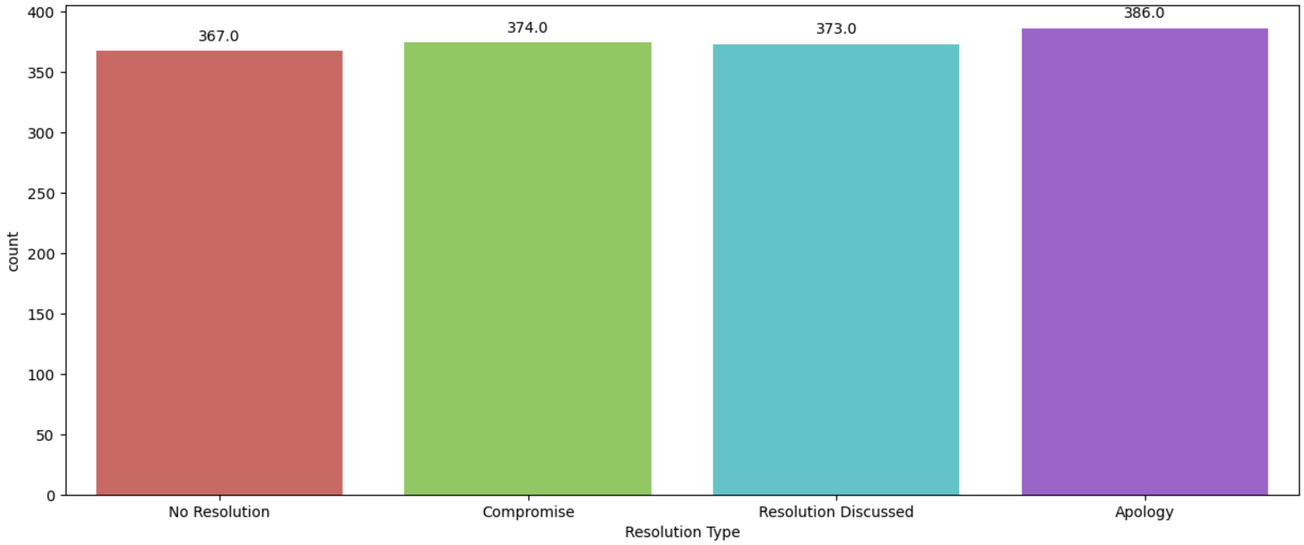
```
Topic
Topic
Money Management           206
Family Responsibilities    200
Household Chores           197
Parenting Styles           188
Personal Space             187
Future Goals               177
Social Activities          176
Time Management            169
Name: count, dtype: int64

Emotional Tone
Emotional Tone
Angry           205
Calm            196
Frustrated      191
Supportive      189
Happy           185
Annoyed         181
Hurt            178
Disappointed    175
Name: count, dtype: int64
```

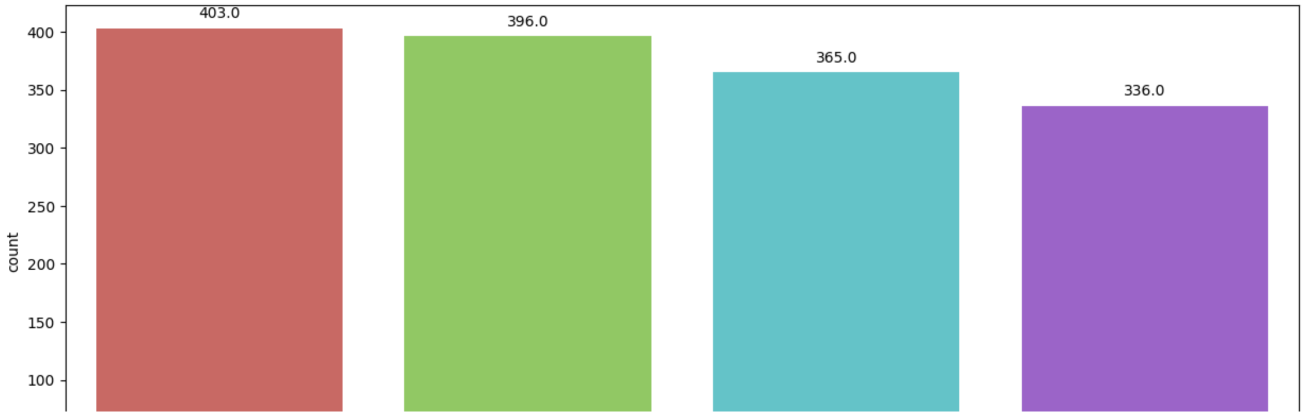
```
Resolution Type
Resolution Type
Apology                386
Compromise             374
Resolution Discussed   373
No Resolution          367
Name: count, dtype: int64

Mood Swing
Mood Swing
Mild        403
Severe      396
Moderate    365
Cant Say    336
Name: count, dtype: int64
```

```
for i in categorical:
    plt.figure(figsize=(15, 6))
    ax = sns.countplot(x=i, data=df, palette='hls')

    for p in ax.patches:
        height = p.get_height()
        ax.annotate(f'{height}',
                    xy=(p.get_x() + p.get_width() / 2., height),
                    xytext=(0, 10),
                    textcoords='offset points',
                    ha='center', va='center')

    plt.show()
```

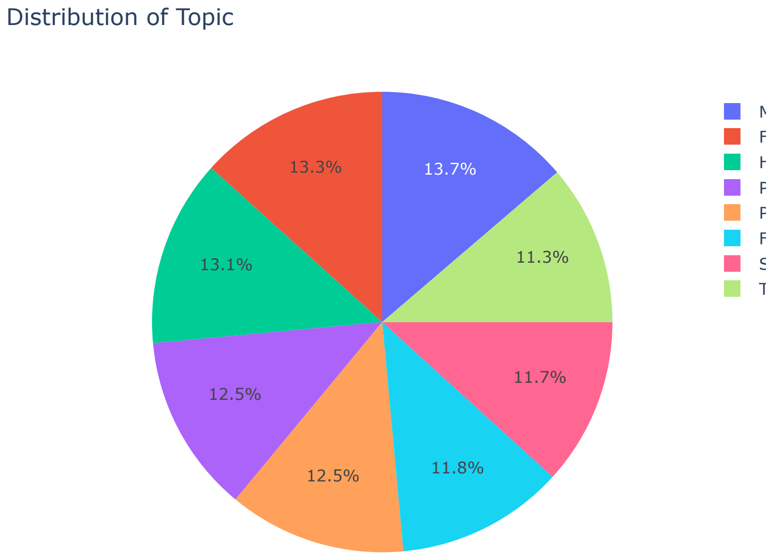

188.0
206.0
169.0
177.0
176.0
197.0
187.0
200.0
count
0
25
50
75
100
125
150
175
200
Parenting Styles
Money Management
Time Management
Future Goals
Social Activities
Household Chores
Personal Space
Family Responsibilities
Topic

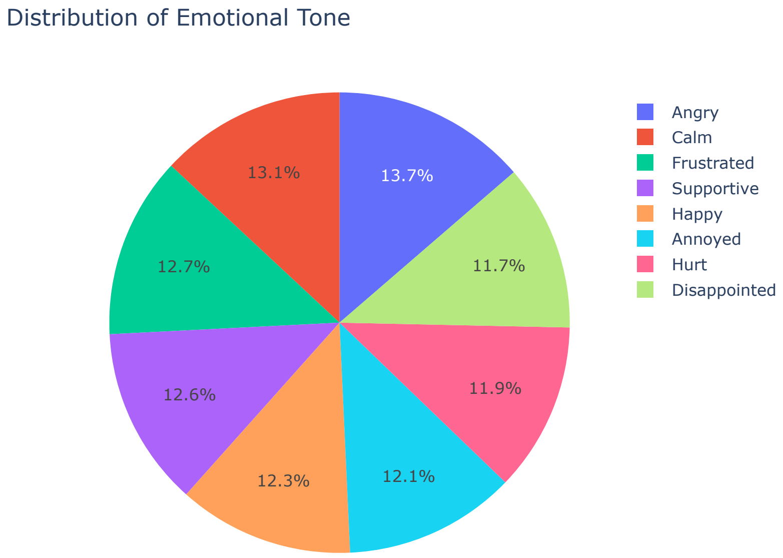

175.0
181.0
205.0
191.0
185.0
189.0
196.0
178.0
count
0
25
50
75
100
125
150
175
200
Disappointed
Annoyed
Angry
Frustrated
Happy
Supportive
Calm
Hurt
Emotional Tone

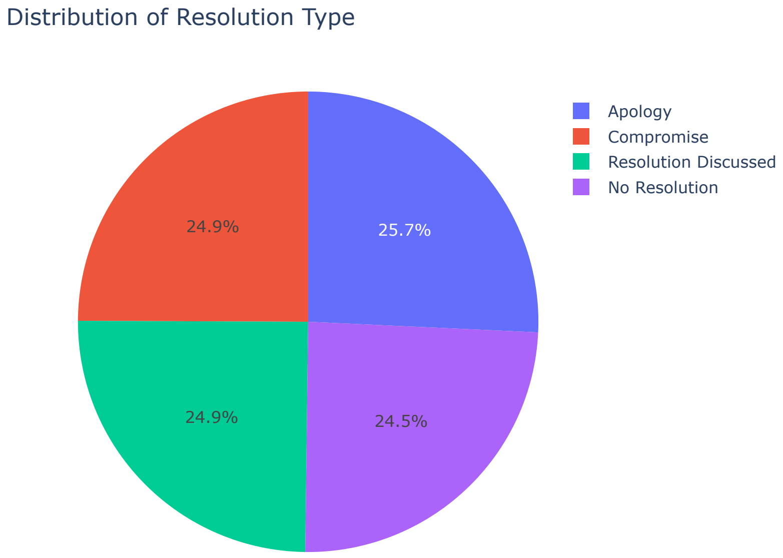

367.0
374.0
373.0
386.0
count
0
50
100
150
200
250
300
350
400
No Resolution
Compromise
Resolution Discussed
Apology
Resolution Type

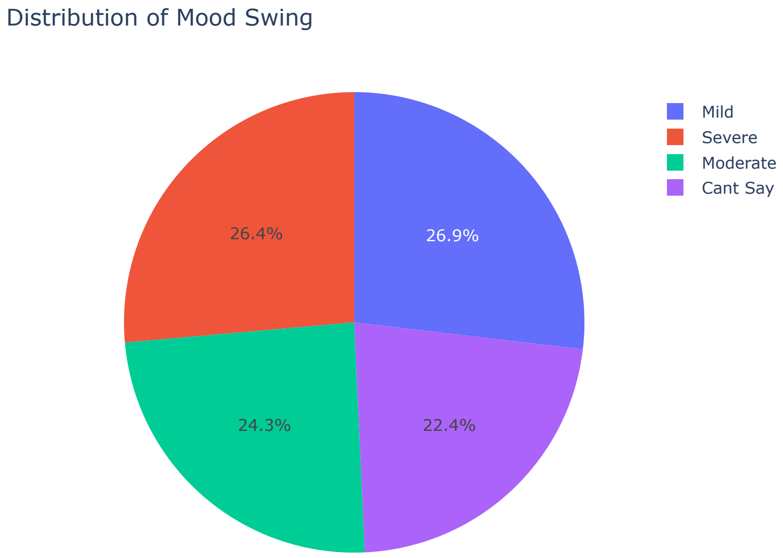

403.0
396.0
365.0
336.0
count
100
150
200
250
300
350
400

```
import plotly.express as px

for i in categorical:
    counts = df[i].value_counts()
    fig = px.pie(counts, values=counts.values, names=counts.index, title=f'Distribution of {i}')
    fig.show()
```

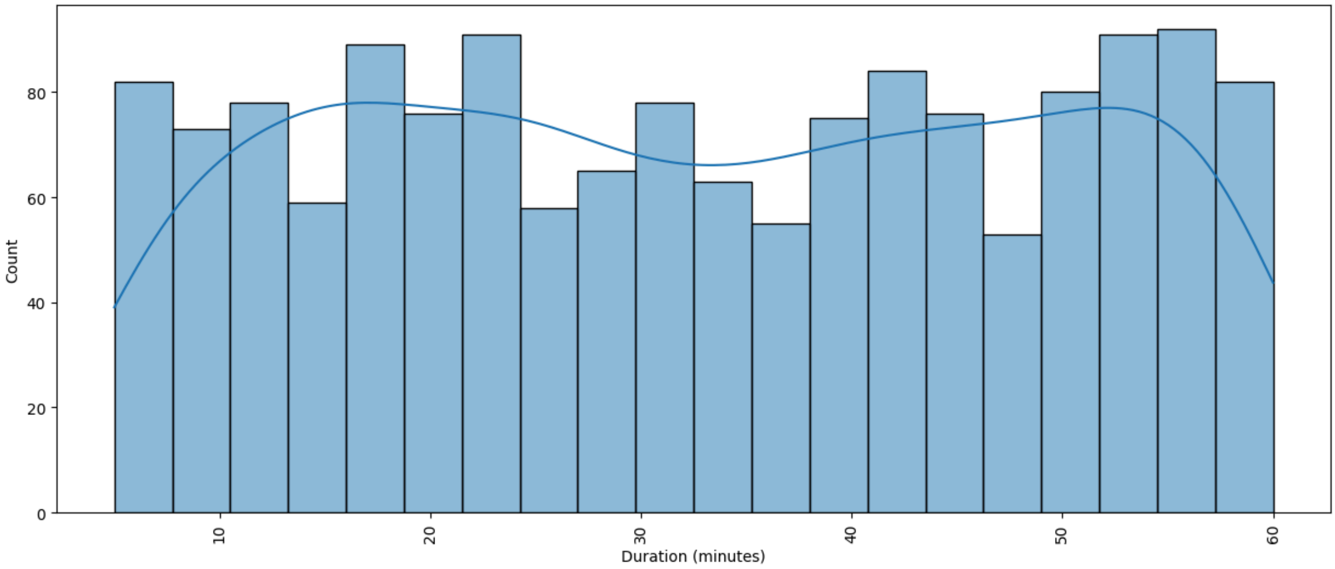

Distribution of Topic
13.7%
11.3%
11.7%
11.8%
12.5%
12.5%
13.1%
13.3%
N
F
H
F
F
F
S
T


Distribution of Emotional Tone
13.7%
11.7%
11.9%
12.1%
12.3%
12.6%
12.7%
13.1%
Angry
Calm
Frustrated
Supportive
Happy
Annoyed
Hurt
Disappointed

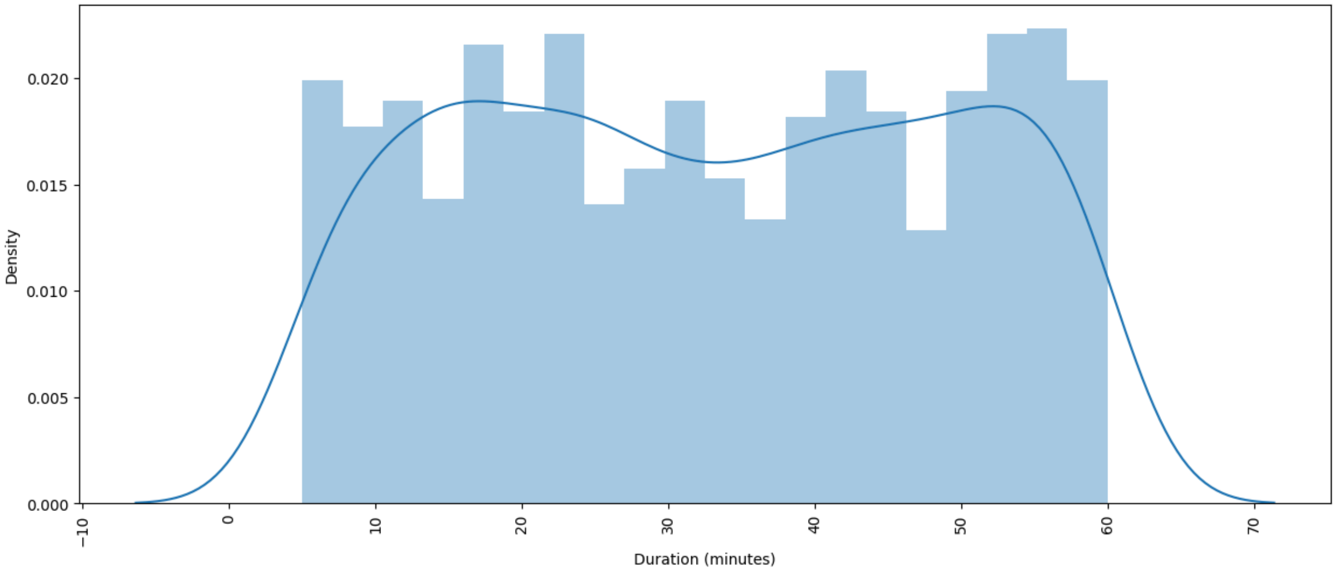

Distribution of Resolution Type
25.7%
24.5%
24.9%
24.9%
Apology
Compromise
Resolution Discussed
No Resolution



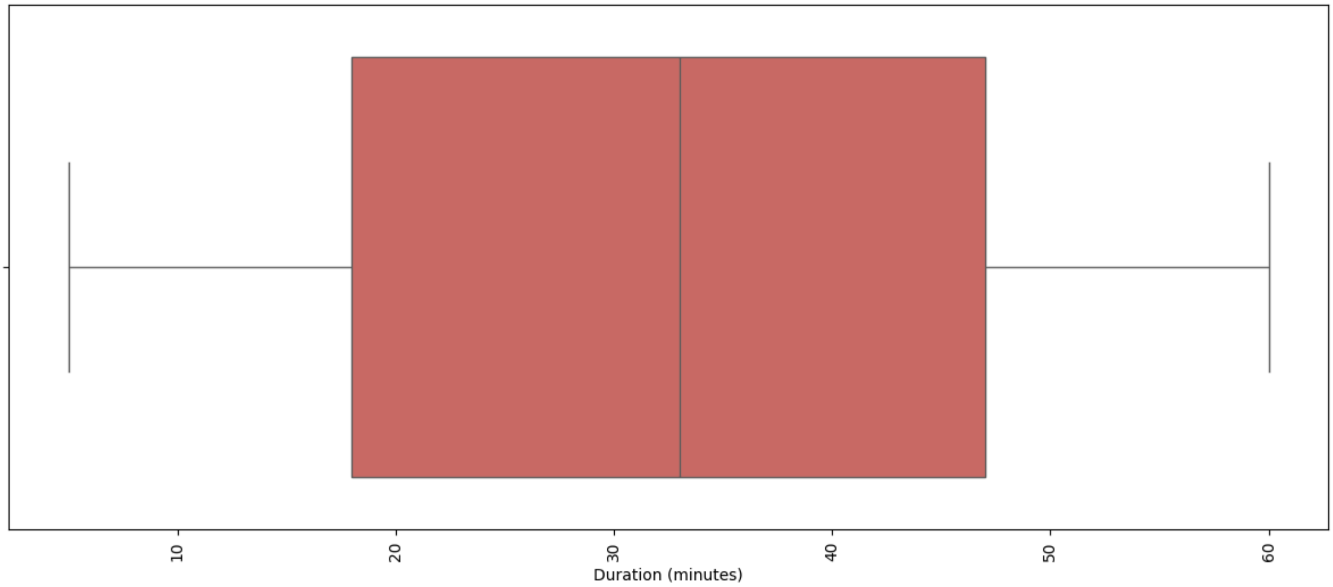
```
for i in continuous:
    plt.figure(figsize=(15,6))
    sns.histplot(df[i], bins = 20, kde = True, palette='hls')
    plt.xticks(rotation = 90)
    plt.show()
```



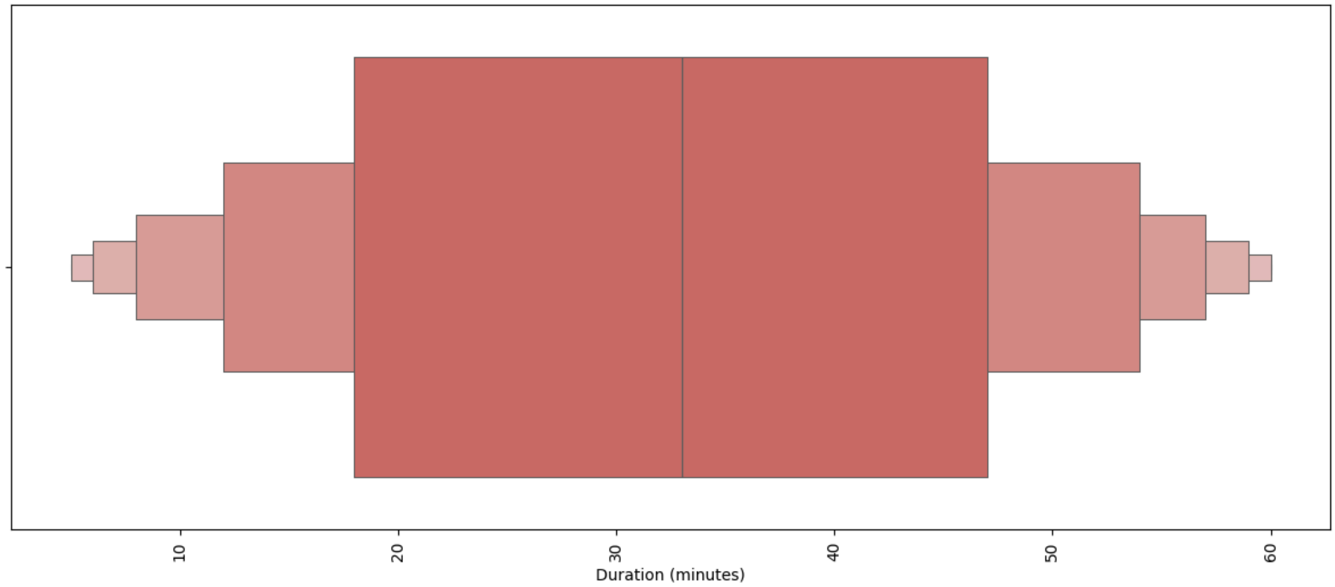
```
for i in continuous:
    plt.figure(figsize=(15,6))
    sns.distplot(df[i], bins = 20, kde = True)
    plt.xticks(rotation = 90)
    plt.show()
```



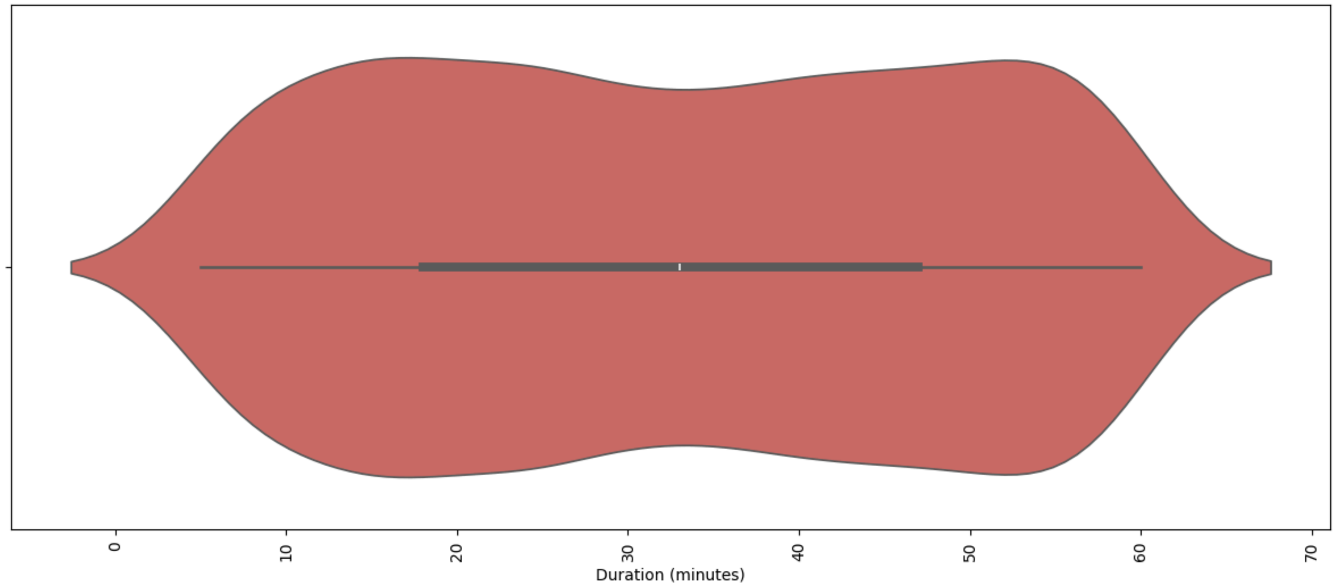
```
for i in continuous:
    plt.figure(figsize=(15, 6))
    sns.boxplot(x=i, data=df, palette='hls')
    plt.xticks(rotation=90)
    plt.show()
```



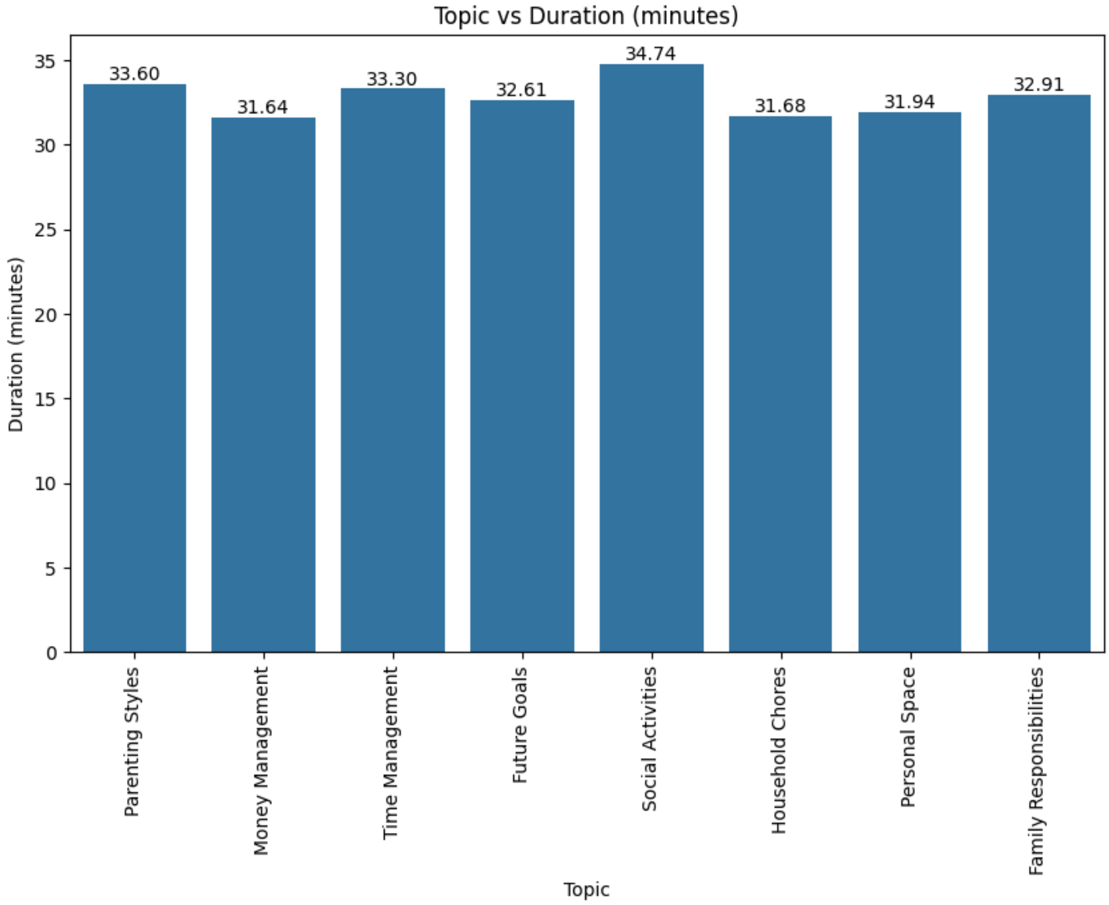
```
for i in continuous:
    plt.figure(figsize=(15, 6))
    sns.boxenplot(x=i, data=df, palette='hls')
    plt.xticks(rotation=90)
    plt.show()
```



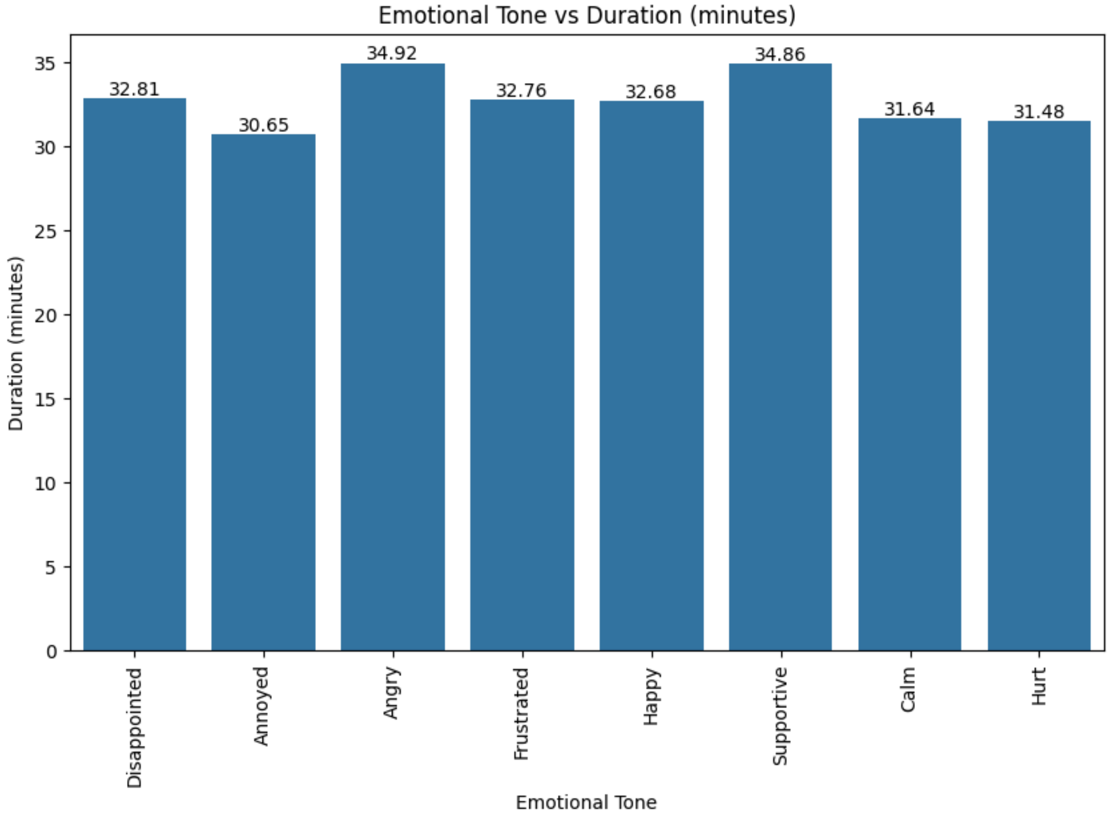
```
for i in continuous:
    plt.figure(figsize=(15, 6))
    sns.violinplot(x=i, data=df, palette='hls')
    plt.xticks(rotation=90)
    plt.show()
```



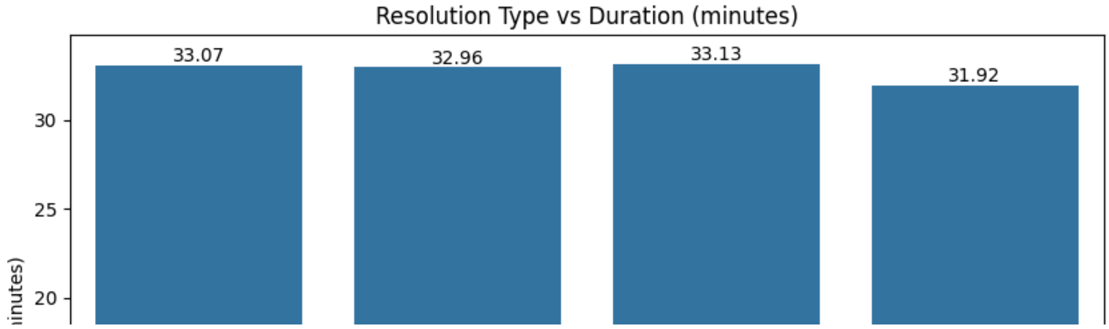
```
for dis in categorical:
    for cont in continuous:
        plt.figure(figsize=(10, 6))
        ax = sns.barplot(data=df, x=dis, y=cont, ci=None)
        plt.title(f'{dis} vs {cont}')

        for p in ax.patches:
            height = p.get_height()
            ax.annotate(f'{height:.2f}', (p.get_x() + p.get_width() / 2., height),
                        ha='center', va='bottom', fontsize=10, color='black', rotation=0)
        plt.xticks(rotation = 90)
        plt.show()
```



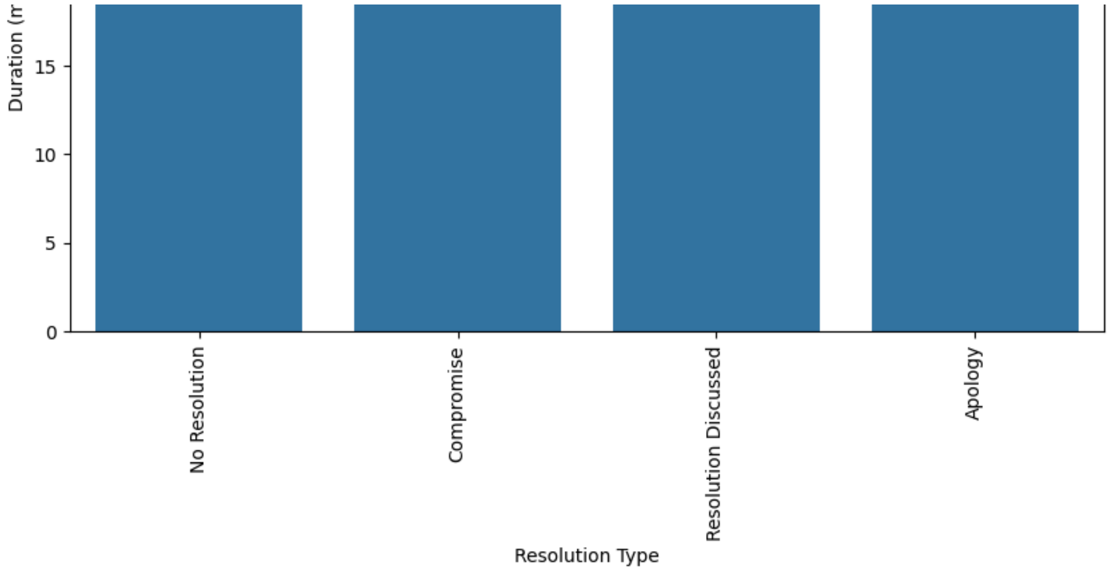
Topic vs Duration (minutes)

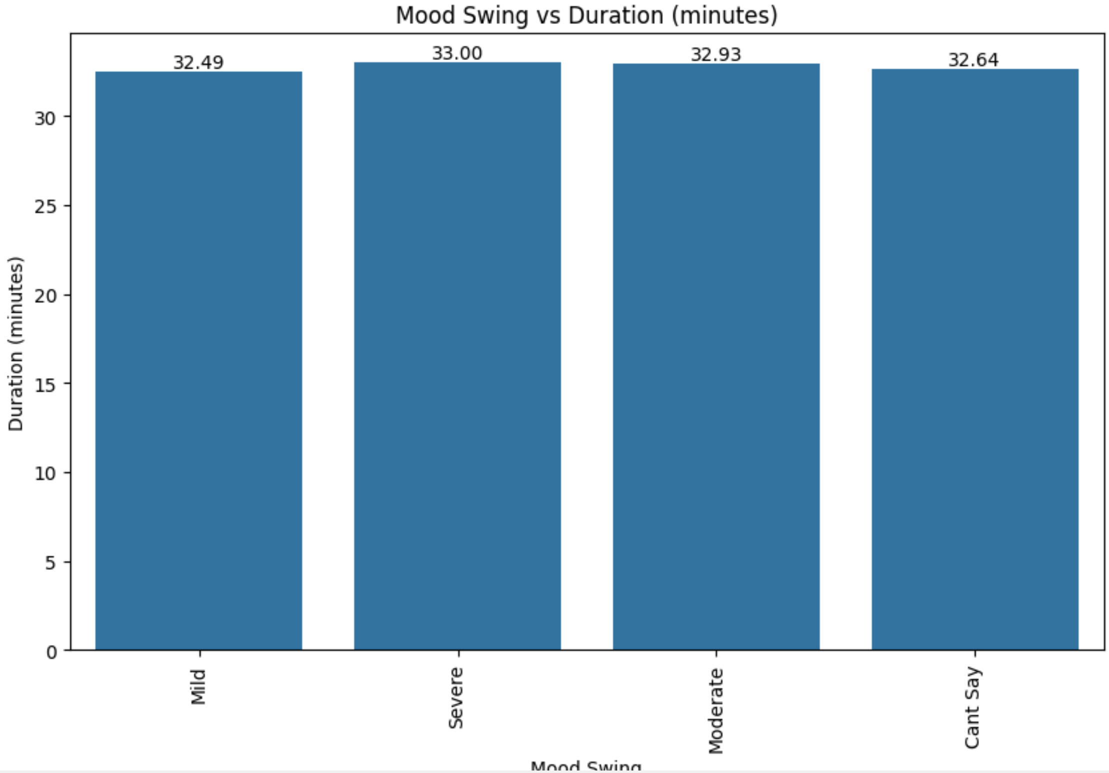


Emotional Tone vs Duration (minutes)

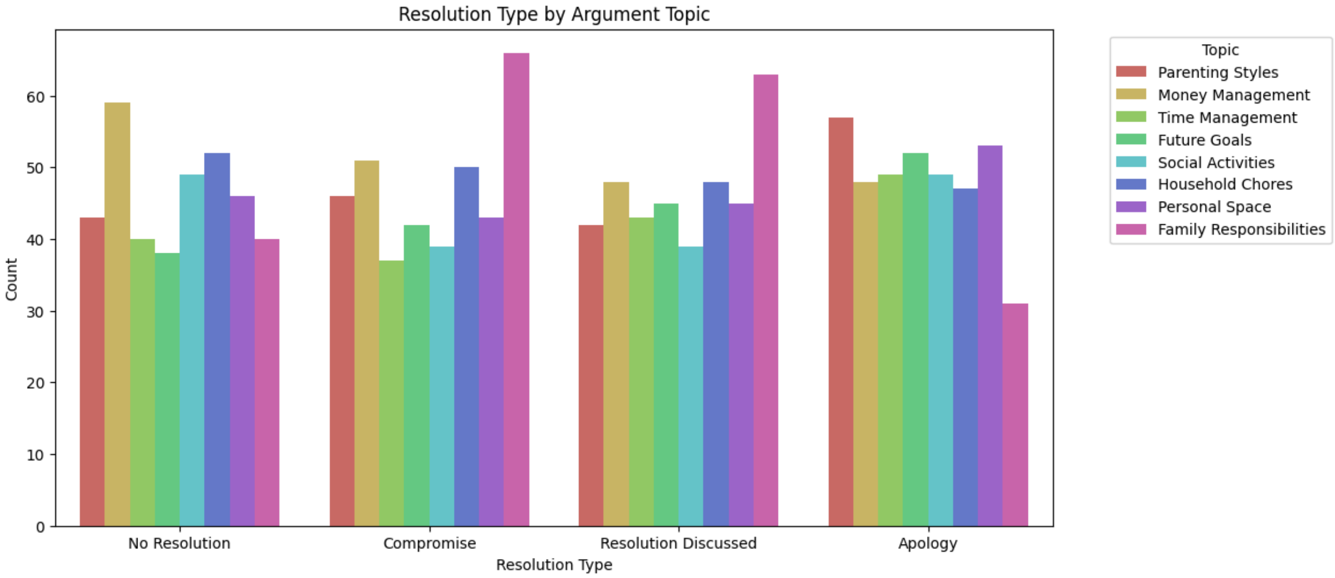


Resolution Type vs Duration (minutes)

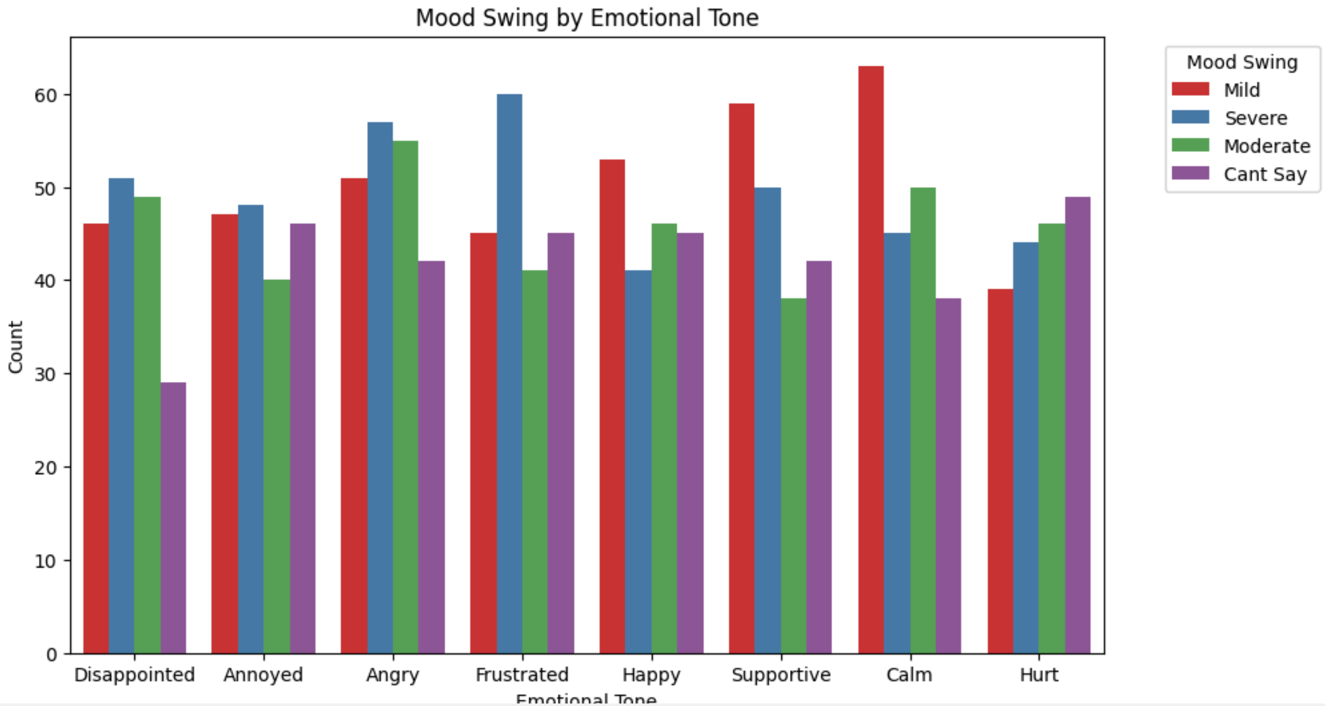

Duration (m
15
10
5
0
No Resolution
Compromise
Resolution Discussed
Apology
Resolution Type


Mood Swing vs Duration (minutes)
32.49
33.00
32.93
32.64
Duration (minutes)
30
25
20
15
10
5
0
Mild
Severe
Moderate
Cant Say
Mood Swing

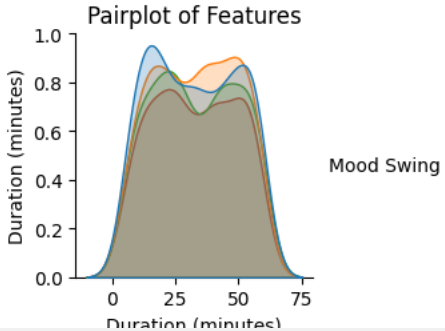
```
plt.figure(figsize=(12, 6))
sns.countplot(x='Resolution Type', hue='Topic', data=df, palette="hls")
plt.title('Resolution Type by Argument Topic')
plt.xlabel('Resolution Type')
plt.ylabel('Count')
plt.legend(title='Topic', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



```
plt.figure(figsize=(10, 6))
sns.countplot(x='Emotional Tone', hue='Mood Swing', data=df, palette="Set1")
plt.title('Mood Swing by Emotional Tone')
plt.xlabel('Emotional Tone')
plt.ylabel('Count')
plt.legend(title='Mood Swing', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



```
sns.pairplot(df, hue='Mood Swing', vars=['Duration (minutes)'])
plt.title('Pairplot of Features')
plt.show()
```



```
from mpl_toolkits.mplot3d import Axes3D
```

```
df['Topic_code'] = df['Topic'].astype('category').cat.codes
df['Resolution_code'] = df['Resolution Type'].astype('category').cat.codes
df['Mood_code'] = df['Mood Swing'].astype('category').cat.codes
```

```
agg_df = df.groupby(['Topic', 'Resolution Type', 'Mood Swing']).size().reset_index(name='Count')
```

```
agg_df
```

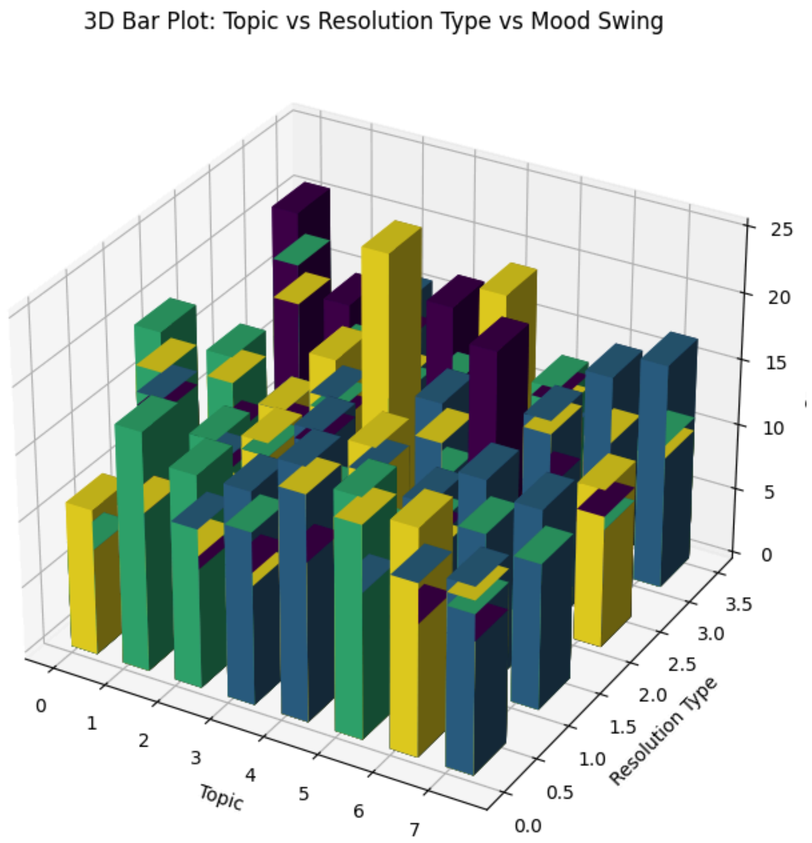
	Topic	Resolution Type	Mood Swing	Count
0	Family Responsibilities	Apology	Cant Say	4
1	Family Responsibilities	Apology	Mild	8
2	Family Responsibilities	Apology	Moderate	8
3	Family Responsibilities	Apology	Severe	11
4	Family Responsibilities	Compromise	Cant Say	14
...	...	...	...	...
123	Time Management	No Resolution	Severe	12
124	Time Management	Resolution Discussed	Cant Say	5
125	Time Management	Resolution Discussed	Mild	17
126	Time Management	Resolution Discussed	Moderate	11
127	Time Management	Resolution Discussed	Severe	10

128 rows × 4 columns

Next steps: Generate code with agg_df View recommended plots New interactive sheet

```
agg_df['Topic_code'] = agg_df['Topic'].astype('category').cat.codes
agg_df['Resolution_code'] = agg_df['Resolution Type'].astype('category').cat.codes
agg_df['Mood_code'] = agg_df['Mood Swing'].astype('category').cat.codes
```

```
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

x = agg_df['Topic_code']
y = agg_df['Resolution_code']
z = np.zeros(len(agg_df))

dx = dy = 0.5
dz = agg_df['Count']

ax.bar3d(x, y, z, dx, dy, dz, shade=True, color=plt.cm.viridis(agg_df['Mood_code'] / max(agg_df['Mood_code'])))

ax.set_xlabel('Topic')
ax.set_ylabel('Resolution Type')
ax.set_zlabel('Count')
ax.set_title('3D Bar Plot: Topic vs Resolution Type vs Mood Swing')
plt.show()
```



```
df
```

	Date	Duration (minutes)	Topic	Emotional Tone	Resolution Type	Mood Swing	Notes	Topic_code	Resolution_code	Mood_code
0	2024-01-05	43	Parenting Styles	Disappointed	No Resolution	Mild	Discussed parenting styles for 43 minutes. Emo...	4	2	1
1	2024-02-05	24	Parenting Styles	Annoyed	Compromise	Severe	Discussed parenting styles for 24 minutes. Emo...	4	1	3
2	2024-02-27	25	Money Management	Annoyed	No Resolution	Moderate	Discussed money management for 25 minutes. Emo...	3	2	2
3	2024-02-07	20	Time Management	Angry	Resolution Discussed	Moderate	Discussed time management for 20 minutes. Emot...	7	3	2
4	2024-01-26	7	Future Goals	Frustrated	Resolution Discussed	Mild	Discussed future goals for 7 minutes. Emotiona...	1	3	1
...	...	...	...	...	...	...	...	...	...	...
1495	2024-01-02	43	Parenting Styles	Calm	Compromise	Mild	Discussed parenting styles for 43 minutes. Emo...	4	1	1

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df['Date'] = pd.to_datetime(df['Date'])
```