I'm Beside You
Exploratory Data Analysis Report

Rohit Shaha

Data Scientist Intern

Contents

We will perform our EDA on python using libraries such as pandas, matplotlib, seaborn etc. We will first visualise the data of individual candidates and then we will compare it together.

1 Importing Libraries

```python
import numpy as np
import pandas as pd #For Data Processing
import matplotlib.pyplot as plt #For Plotting Graphs
import seaborn as sns #For Plotting Graphs
```

2 EDA on each Candidate

2.1 Candidate 1

Importing emotion file

```python
emotion=pd.read_csv(r".\emotion_data\1\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	0	4.31735	0.000594	2.879790	1.65035	2.779980	0.600814	87.77110	neutral
1	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	1	53.22530	2.981640	12.736800	1.52347	1.051320	27.216800	1.26462	angry
2	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	2	8.79651	0.029468	2.968160	16.83150	39.884600	0.279335	31.21050	sad
3	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	3	9.45303	0.106778	1.553080	20.93010	3.503870	0.909426	63.54370	neutral
4	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	4	56.00020	0.000004	0.162231	5.58358	0.197026	12.807600	25.24940	angry

Finding the data type, missing and unique values of each column using the Autoviz library.

```python
from autoviz.AutoViz_Class import AutoViz_Class
AV = AutoViz_Class()

filename = "emotion.csv"

dfte = AV.AutoViz(filename, sep=',', depVar='', dfte=None, header=0, verbose=1,
↪ lowess=False,chart_format='svg',max_rows_analyzed=150000,max_cols_analyzed=30,
↪ save_plot_dir=None)
```

	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
movie_id	object	0.000000	1	nan	nan	Zero-variance colum: drop before modeling process.
image_seq	int64	0.000000	100	0.000000	94.000000	Possible ID colum: drop before modeling process.
angry	float64	0.000000	NA	0.164384	71.172500	has 9 outliers greater than upper bound (44.11) or lower than lower bound(-23.48). Cap them or remove them.
disgust	float64	0.000000	NA	0.000000	21.508900	has 15 outliers greater than upper bound (0.16) or lower than lower bound(-0.10). Cap them or remove them.
fear	float64	0.000000	NA	0.079219	94.981800	has 15 outliers greater than upper bound (49.73) or lower than lower bound(-26.86). Cap them or remove them.
happy	float64	0.000000	NA	0.000005	66.222300	has 10 outliers greater than upper bound (13.71) or lower than lower bound(-8.00). Cap them or remove them.
sad	float64	0.000000	NA	0.000073	91.563600	has 12 outliers greater than upper bound (32.37) or lower than lower bound(-16.47). Cap them or remove them.
surprise	float64	0.000000	NA	0.000008	97.834400	has 13 outliers greater than upper bound (15.99) or lower than lower bound(-9.26). Cap them or remove them.
neutral	float64	0.000000	NA	0.000117	97.823000	No issue
dominant_emotion	object	0.000000	6	nan	nan	No issue

There are no missing values in the dataset. Only 2 columns, movie_id and dominant_emotion is of object data type each having 1 and 6 unique values As movie_id is same for a particular candidate, we can remove that column.

```python
emotion.pop('movie_id')
```

Calculating basic statistics for each emotion, such as mean, median, standard deviation, and range, to get an overview of the candidates' emotional expressions.

```python
print(emotion.shape)
emotion.describe()
```

shape = (87, 10)

	image_seq	angry	disgust	fear	happy	sad	surprise	neutral
count	87.000000	87.000000	8.700000e+01	87.000000	87.000000	87.000000	87.000000	87.000000
mean	45.390805	14.451059	6.168965e-01	18.382797	5.865318	13.575324	8.744969	38.363648
std	27.587643	18.544205	2.679399e+00	25.073562	11.237819	19.787221	19.621163	33.468718
min	0.000000	0.164384	2.400910e-10	0.079219	0.000005	0.000073	0.000008	0.000117
25%	22.500000	1.867450	8.769180e-05	1.862945	0.143215	1.845405	0.210637	8.035385
50%	44.000000	6.412790	3.443590e-03	6.366870	1.476330	5.578010	0.970922	28.221400
75%	68.500000	18.765500	6.486260e-02	21.010100	5.569355	14.056700	6.524355	71.227450
max	94.000000	71.172500	2.150890e+01	94.981800	66.222300	91.563600	97.834400	97.823000

Plotting the Correlation matrix heatmap using seaborn library.

```python
corelation = emotion.corr()
sns.heatmap(corelation, xticklabels=corelation.columns, yticklabels=corelation.columns,annot=True)
```

Upon analyzing the correlation matrix for Candidate 1's emotional expressions, it is evident that there isn't a substantial correlation between any two emotions. This suggests that Candidate 1's emotional expressions are quite diverse and do not consistently occur together.

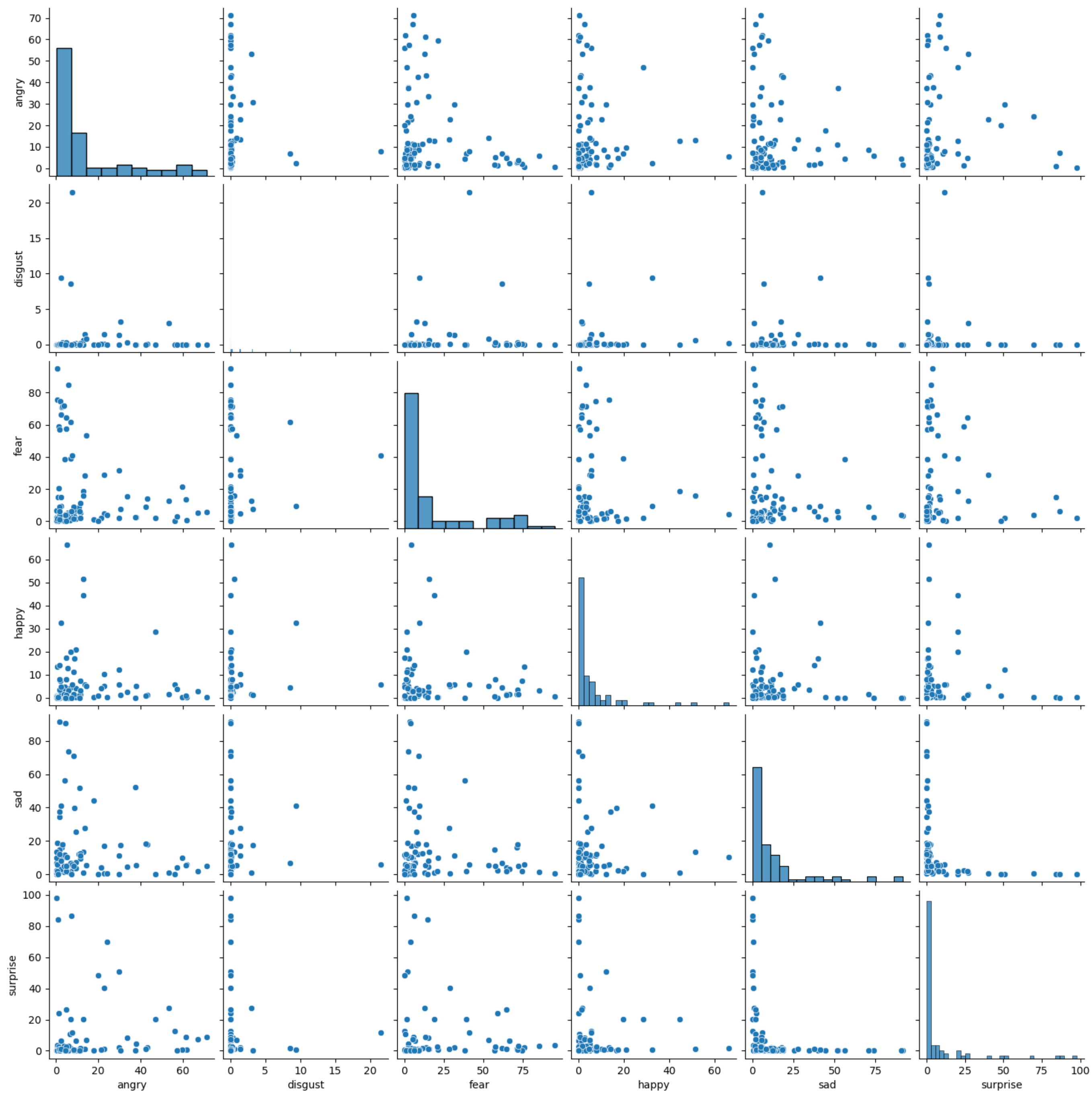
The highest observed correlation is between 'Sad' and 'Surprise'. Specifically, when Candidate 1 exhibits lower levels of surprise, there tends to be an increase in their expression of sadness. This finding aligns with common human emotional experiences, as a decrease in surprise might be associated with moments of realization or recognition, which can lead to a shift towards a sadder emotional state.

However, it's worth noting that there is no significant correlation between 'Angry' and 'Happy' values for Candidate 1. This implies that their expressions of anger and happiness do not seem to be related, suggesting a degree of emotional complexity.

This information provides valuable insights into the emotional dynamics of Candidate 1, highlighting their ability to express a wide range of emotions and the nuanced interplay between sadness and surprise in their emotional responses.

Plotting the scatterplot of each emotion with other emotion using pairplot.

```
sns.pairplot(emotion)
```



Upon examining the pairplot for Candidate 1's emotional expressions, several noteworthy observations come to light:

- Disgust Values: Throughout the video, Candidate 1 consistently maintains very low values for 'Disgust'. This indicates that the candidate seldom displays signs of disgust in their emotional expressions, which can be seen as a positive trait, especially in a professional context where excessive disgust may not be well-received.

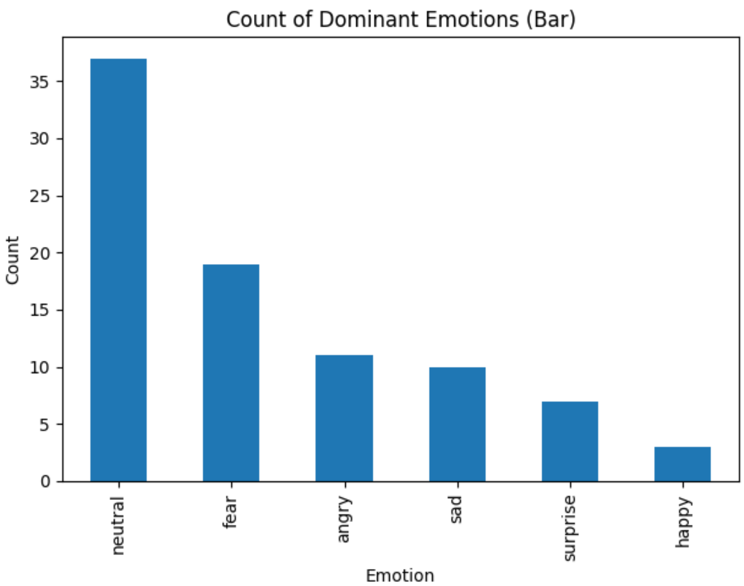
- Anger Dynamics: Candidate 1's emotional state in terms of 'Anger' appears to be quite variable. There are instances where the candidate exhibits high values of anger, and these spikes in anger could be perceived as a potential area of concern. Consistent high levels of anger might be considered undesirable in a candidate, as it may impact their ability to work collaboratively and maintain composure.
- Fear Expression: Candidate 1's 'Fear' values tend to be relatively high compared to other emotions. This indicates that the candidate frequently displays signs of fear or anxiety during the video. While a certain level of caution and concern can be healthy, excessively high fear levels may be indicative of nervousness or discomfort, which could affect their performance and confidence in a professional setting.

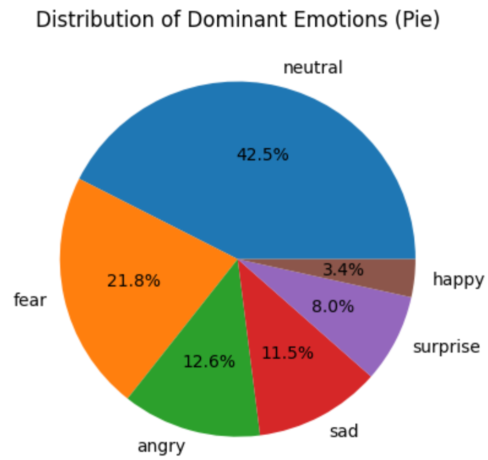
In summary, Candidate 1's emotional expressions reveal a consistent absence of disgust, but they exhibit variability in anger levels and tend to express a relatively high degree of fear. These findings can be considered when evaluating the candidate's suitability for a role that requires emotional stability and composure, especially in potentially stressful situations.

Plotting a bar chart and pie chart showing the occurences of dominant emotions.

```python
# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

# Plot the bar graph on the first subplot
emotion['dominant_emotion'].value_counts().plot(kind='bar', ax=ax1)
ax1.set_title('Count of Dominant Emotions (Bar)')
ax1.set_xlabel('Emotion')
ax1.set_ylabel('Count')

# Plot the pie chart on the second subplot
emotion['dominant_emotion'].value_counts().plot(kind='pie', autopct='%0.1f%%',
↪  startangle=0, ax=ax2)
ax2.set_title('Distribution of Dominant Emotions (Pie)')
ax2.set_ylabel(' ')

# Adjust subplot layout
plt.tight_layout()
```

```python

# Show the plots
plt.show()
```





1. Pie Chart

   - The pie chart reveals that a substantial portion of Candidate 1's emotional expressions were categorized as 'Neutral'. This dominant neutral emotion suggests that the candidate often maintained a composed and unemotional demeanor during the introduction video.
   - However, it's noteworthy that the candidate exhibited negative emotions, specifically 'Fear,' 'Anger,' and 'Sadness,' collectively accounting for more than 40% of their emotional expressions. This significant presence of negative emotions implies that the candidate may not have appeared very confident and was, in fact, quite fearful and potentially agitated during the video.

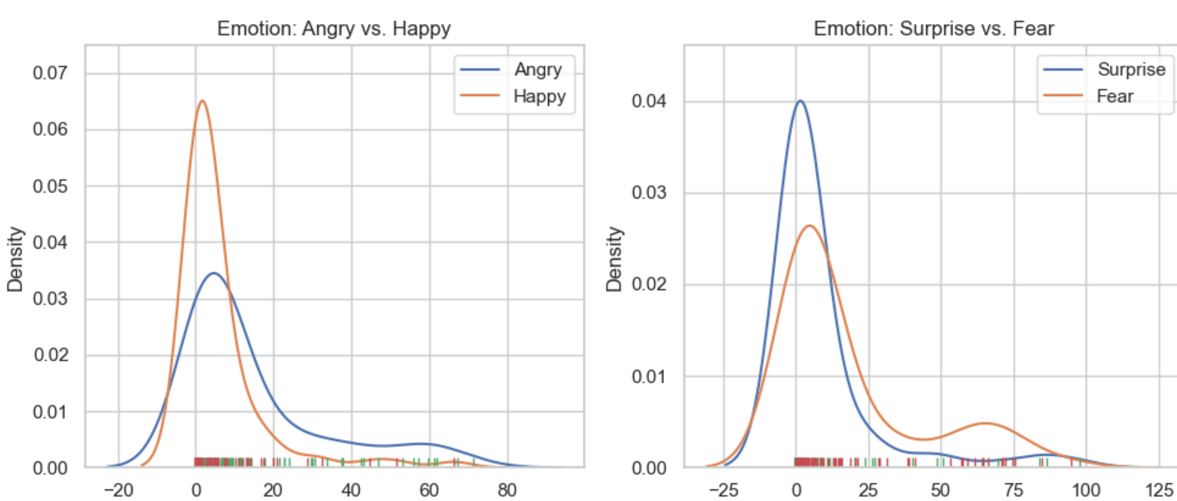
2. Bar Chart

   - The bar chart provides a more detailed breakdown of the emotions expressed by Candidate 1. It reiterates the prevalence of 'Neutral' emotional expressions as the dominant category.
   - Notably, the candidate's expression of 'Fear,' 'Anger,' and 'Sadness' is visibly higher than other emotions, reinforcing the observation that these

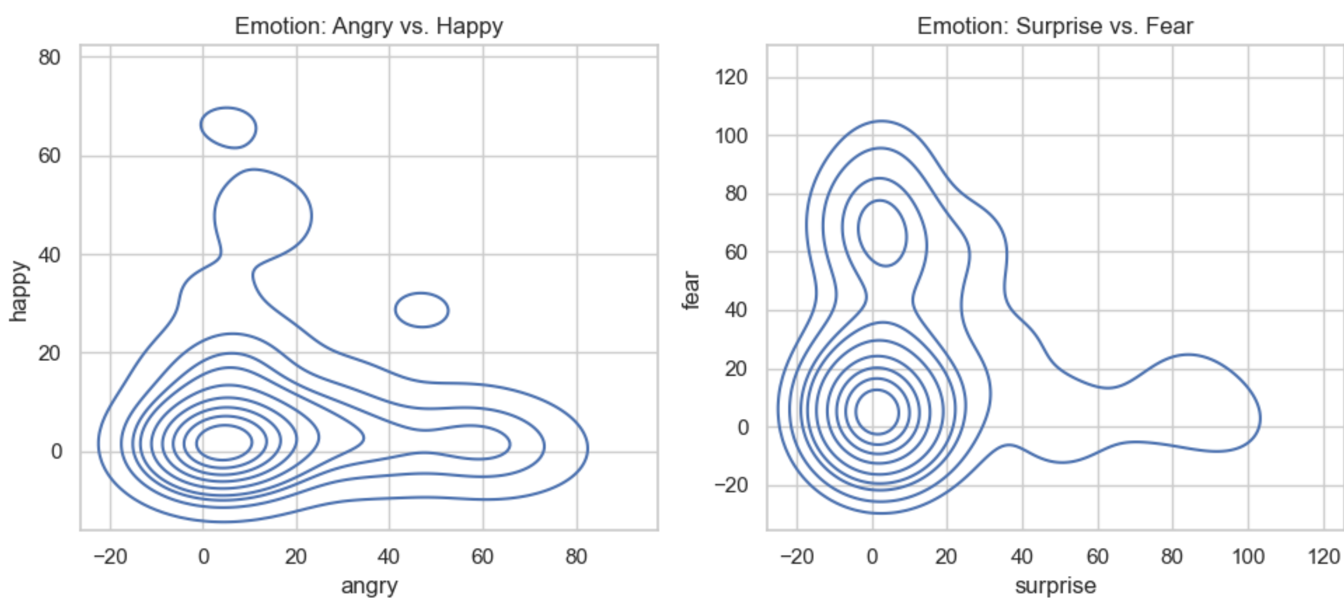
negative emotions played a substantial role in the candidate's emotional spectrum during the video.

Now plotting the KDE plot between the 2 negative and 2 positive emotions to analyze the density.

```
# Create a figure with two subplots in one row and two columns
fig, axes = plt.subplots(1, 2, figsize=(10, 4.5))

# Create KDE plots for 'angry' and 'happy' in the first subplot
sns.kdeplot(x='angry', data=emotion, label='Angry', ax=axes[0])
sns.kdeplot(x='happy', data=emotion, label='Happy', ax=axes[0])
sns.rugplot(x='angry', data=emotion, ax=axes[0])
sns.rugplot(x='happy', data=emotion, ax=axes[0])
axes[0].set_xlabel(' ')
axes[0].set_title('Emotion: Angry vs. Happy')
axes[0].legend()

# Create KDE plots for 'surprise' and 'fear' in the second subplot
sns.kdeplot(x='surprise', data=emotion, label='Surprise', ax=axes[1])
sns.kdeplot(x='fear', data=emotion, label='Fear', ax=axes[1])
sns.rugplot(x='surprise', data=emotion, ax=axes[1])
sns.rugplot(x='fear', data=emotion, ax=axes[1])
axes[1].set_xlabel(' ')
axes[1].set_title('Emotion: Surprise vs. Fear')
axes[1].legend()

# Adjust subplot layout
plt.tight_layout()

# Show the plots
plt.show()
```



```python
# Create a figure with two subplots in one row and two columns
fig, axes = plt.subplots(1, 2, figsize=(10, 4.5))

# Create KDE plots for 'angry' and 'happy' in the first subplot
sns.kdeplot(x='angry', y='happy', data=emotion, ax=axes[0])
axes[0].set_title('Emotion: Angry vs. Happy')

# Create KDE plots for 'surprise' and 'fear' in the second subplot
sns.kdeplot(x='surprise', y='fear', data=emotion, ax=axes[1])
axes[1].set_title('Emotion: Surprise vs. Fear')

# Adjust subplot layout
plt.tight_layout()

# Show the plots
plt.show()
```

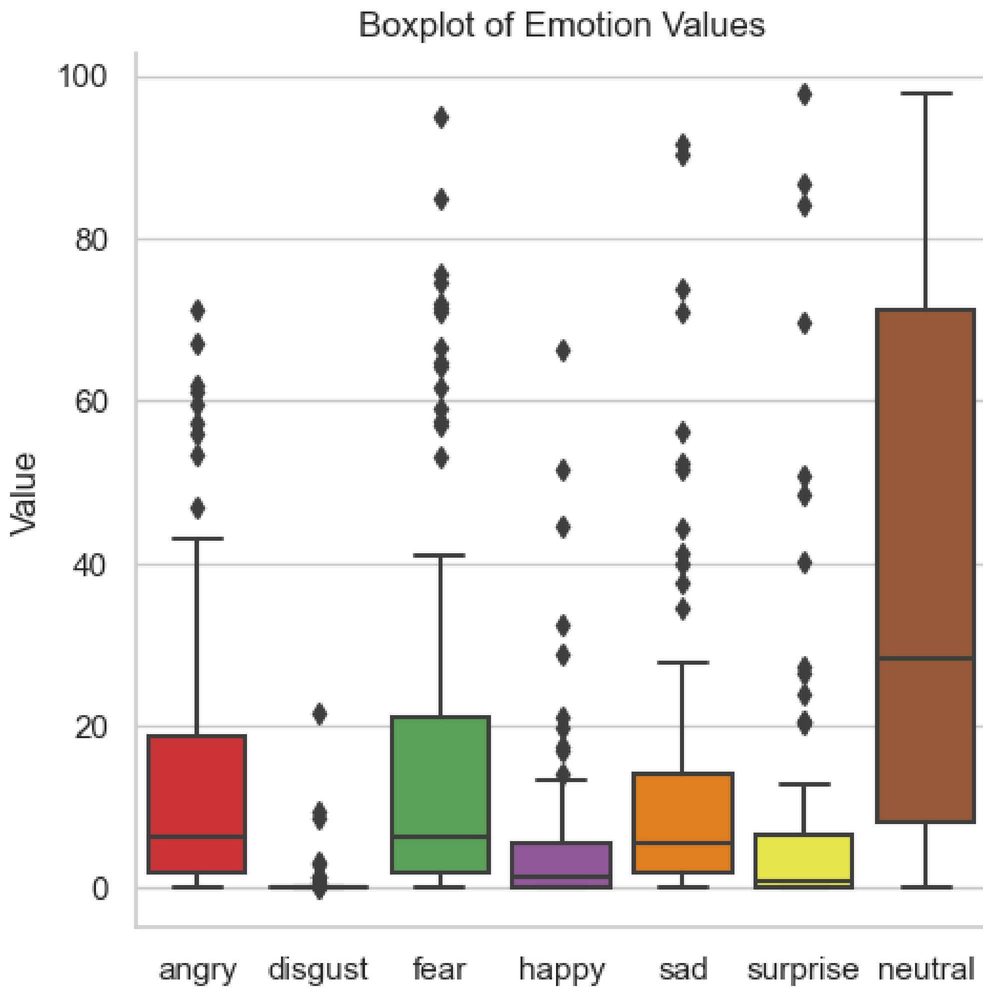


1. Happy and Angry Emotions:
    - The KDE plot illustrates that the density of 'Happy' emotion scores is notably high in the range of 0 to 10. This indicates that Candidate 1 frequently exhibits low 'Happy' scores, suggesting a generally less cheerful or content disposition.
    - In contrast, the density of 'Angry' emotion scores appears to be greater for larger values. This implies that the candidate's emotional expressions often include higher levels of anger. The combination of low 'Happy' scores and higher 'Angry' scores suggests a potential disposition towards negative emotions.

2. Surprise and Fear Emotions:
   - For 'Surprise' emotion, the KDE plot reveals that the density of scores is generally quite low and concentrated in the range of 0 to 15. This indicates that Candidate 1 rarely expresses high levels of surprise during the video, suggesting a lack of overt astonishment or amazement.
   - Conversely, 'Fear' emotion scores display a wider range, with values ranging from 0 to 75. This implies that Candidate 1's emotional expressions encompass a broader spectrum of fear, with instances of both low and high fear levels. The varying fear levels might be indicative of moments of uncertainty or apprehension during the video.

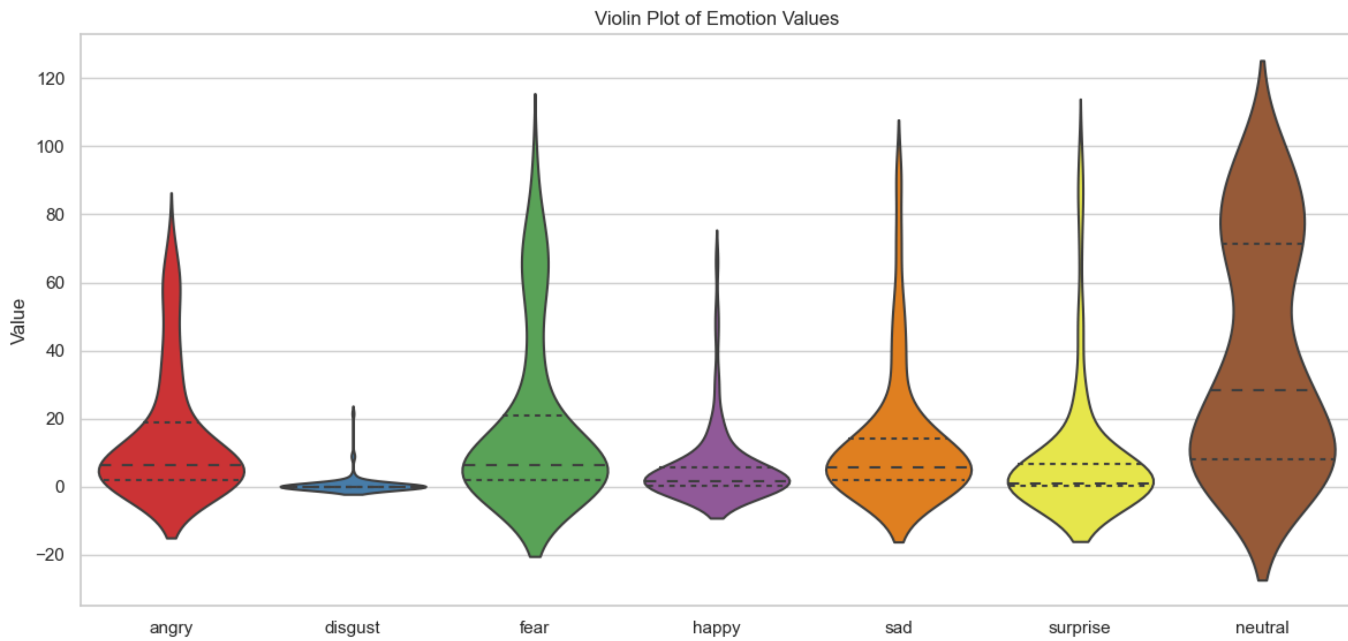
Plotting the box plot and violin plot to find the outliers in the data.

```python
# List of emotions
emotions = ['angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral']

# Create a boxplot for each emotion in one graph
plt.figure(figsize=(12, 6))
sns.set(style="whitegrid")

# Melt the DataFrame to make it suitable for plotting
melted_emotion = pd.melt(emotion, value_vars=emotions, var_name='Emotion',
    value_name='Value')

# Create a boxplot using catplot with hue
sns.catplot(x='Emotion', y='Value', kind='box', data=melted_emotion, palette='Set1')

# Set labels and title
plt.xlabel(' ')
plt.ylabel('Value')
plt.title('Boxplot of Emotion Values')

# Show the plot
plt.show()
```




```python
# Create a violin plot for each emotion in one graph
plt.figure(figsize=(12, 6))
sns.set(style="whitegrid")

# Melt the DataFrame to make it suitable for plotting
melted_emotion = pd.melt(emotion, value_vars=emotions, var_name='Emotion',
↪   value_name='Value')

# Create a violin plot with adjusted scale
sns.violinplot(x='Emotion', y='Value', data=melted_emotion, palette='Set1',
↪   inner='quartile', scale='width')

# Set labels and title
plt.xlabel(' ')
plt.ylabel('Value')
plt.title('Violin Plot of Emotion Values')

# Show the plot
plt.xticks()
```

```
plt.tight_layout()  # Adjust subplot layout
plt.show()
```



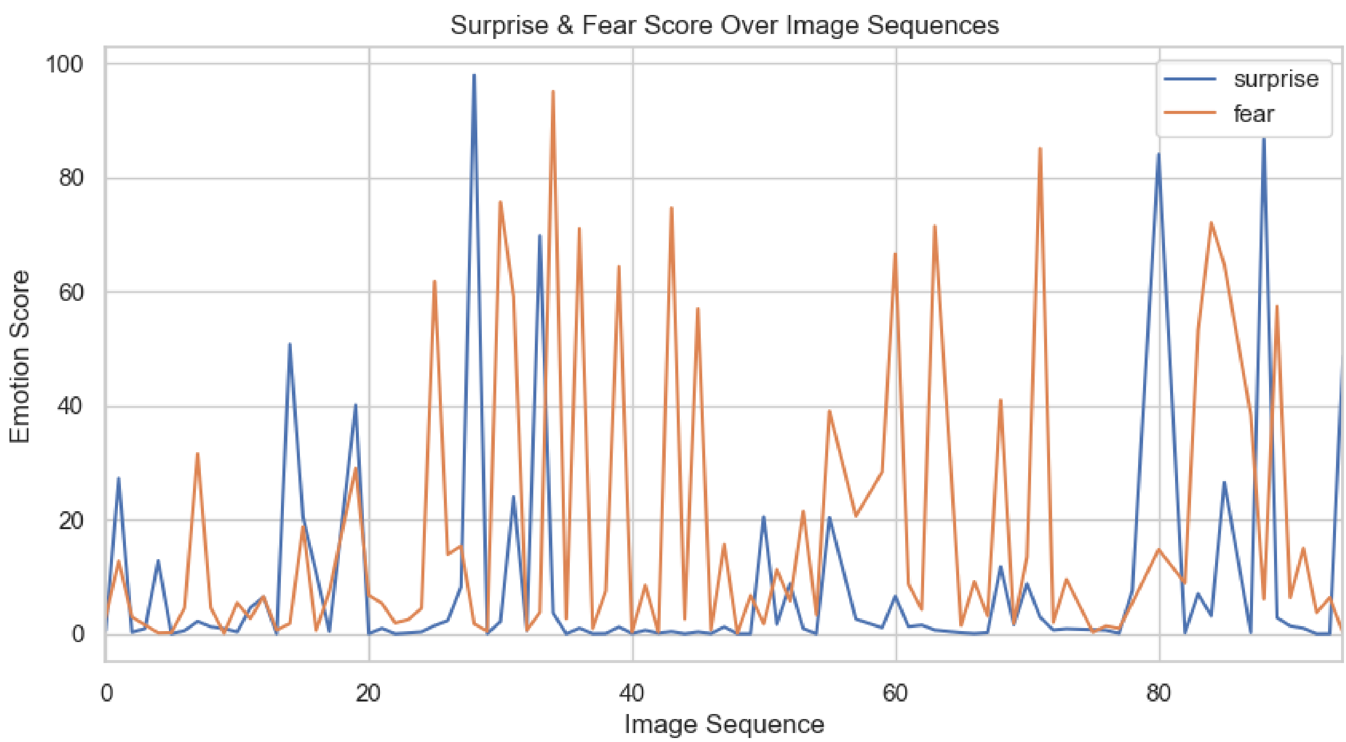
Also, there are many outliers in our dataset. It tells us that:

- Emotional Variability: The presence of outliers in various emotions could indicate that the candidate has a wide range of emotional expressions. The candidate may be highly expressive and responsive to different situations, which can be an asset in roles that require adaptability and quick emotional responses.
- Stress or Nervousness: Outliers in emotions may also be a result of stress or nervousness during the video recording. Candidates may express extreme emotions when they are anxious or under pressure. It's essential to consider whether these outliers are situational or reflective of their usual behavior.

Plotting line chart of emotion scores over image sequences.

```
# Set the figure size to make the x-axis wider
plt.figure(figsize=(10, 5))

# Plot the lineplot for the 'surprise' and 'fear' emotions
sns.lineplot(x='image_seq', y='surprise', label='surprise', data=emotion)
sns.lineplot(x='image_seq', y='fear', label='fear', data=emotion)

# Set the x-axis limits (adjust the values as needed)
```

```
plt.xlim(0, emotion['image_seq'].max())  # Use the maximum image_seq value as the
↪  upper limit

# Set labels, title, and legend (if needed)
plt.legend()
plt.xlabel('Image Sequence')
plt.ylabel('Emotion Score')
plt.title('Surprise & Fear Score Over Image Sequences')

# Show the plot
plt.show()
```



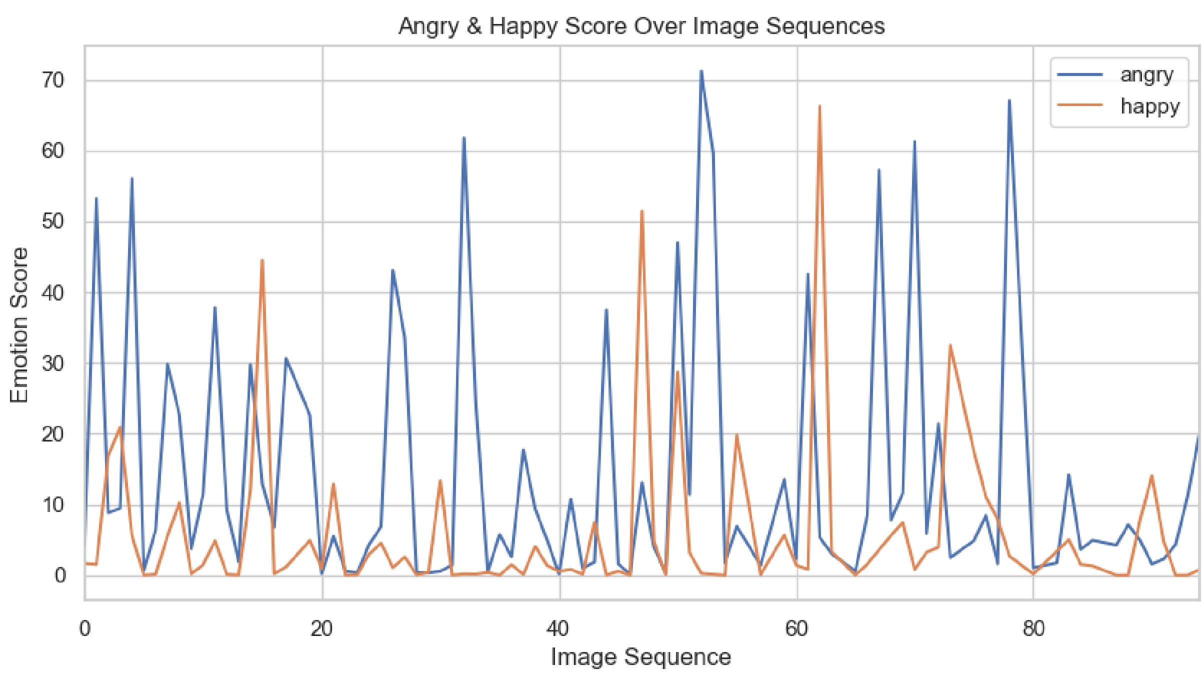
1. Surprise Scores:

   - In the beginning of the video, Candidate 1's surprise scores show noticeable fluctuations, indicating varying levels of surprise. This suggests that the candidate experienced a range of surprising moments early on.
   - However, as the video progresses into the middle portion, the surprise scores stabilize and remain somewhat constant. This suggests that the candidate's initial reactions of surprise subside, and they maintain a more consistent level of surprise throughout this phase.

2. Fear Scores:

- In contrast to surprise, Candidate 1's fear scores exhibit a different pattern. They continuously fluctuate and drift from low to high over the course of the video.
- This indicates that Candidate 1's emotional state in terms of fear is highly dynamic, with moments of both low and high fear occurring intermittently. The fact that fear scores fluctuate suggests that the candidate experiences a range of emotions related to fear throughout the video.

```python
# Set the figure size to make the x-axis wider
plt.figure(figsize=(10, 5))

# Plot the lineplot for the 'angry' and 'happy'  emotions
sns.lineplot(x='image_seq', y='angry', label='angry', data=emotion)
sns.lineplot(x='image_seq', y='happy', label='happy', data=emotion)

# Set the x-axis limits (adjust the values as needed)
plt.xlim(0, emotion['image_seq'].max())  # Use the maximum image_seq value as the upper limit

# Set labels, title, and legend (if needed)
plt.legend()
plt.xlabel('Image Sequence')
plt.ylabel('Emotion Score')
plt.title('Angry & Happy Score Over Image Sequences')

plt.show()
```



The graph illustrates a significant drift in 'Angry' scores compared to 'Happy' scores for Candidate 1. This suggests that the candidate's emotional expressions are characterized by a substantial fluctuation in anger levels.

Furthermore, the observation that 'Angry' scores exhibit a more pronounced drift than 'Happy' scores indicates a higher variability in the candidate's expressions of anger.

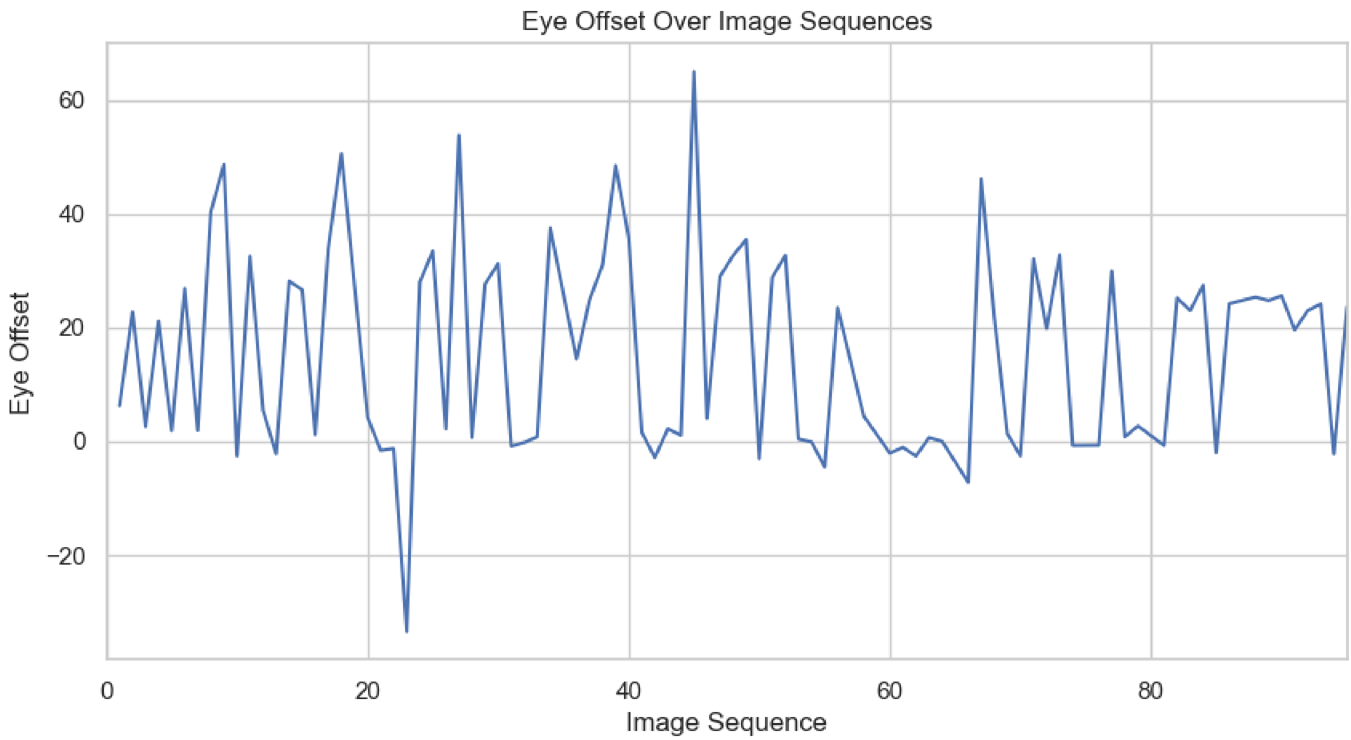
Importing the data.

```
gaze = pd.read_csv(r".\emotion_data\1\gaze.csv")
gaze.head()
```

	movie_id	image_seq	gaze	blink	eye_offset
0	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	1	1	0	6.2253
1	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	2	1	0	22.7274
2	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	3	1	0	2.5704
3	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	4	1	0	21.1097
4	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	5	1	0	1.8453

Plotting line chart of eye_offset over image sequences.

```
# Set the figure size to make the x-axis wider
plt.figure(figsize=(10, 5))

# Plot the lineplot for the 'eye_offset' emotion
sns.lineplot(x='image_seq', y='eye_offset', data=gaze)

# Set the x-axis limits (adjust the values as needed)
plt.xlim(0, gaze['image_seq'].max())  # Use the maximum image_seq value as the upper
↪  limit

# Set labels, title, and legend (if needed)
plt.xlabel('Image Sequence')
plt.ylabel('Eye Offset')
plt.title('Eye Offset Over Image Sequences')

# Show the plot
plt.show()
```



Throughout the video, Candidate 1's eye offset predominantly falls within the angle range of 0-40° degrees. This suggests that the candidate maintains their gaze primarily within this narrow angle for a significant portion of the video

2.2 Candidate 2

Importing emotion file

```
emotion=pd.read_csv(r".\emotion_data\1\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	baa26895-85b2-465b-a972-649b41d9870e	0	4.903760	0.000024	1.847580	2.55923	48.79130	0.033327	41.864800	sad
1	baa26895-85b2-465b-a972-649b41d9870e	1	0.179621	0.000185	0.055258	93.56640	6.18999	0.001184	0.007402	happy
2	baa26895-85b2-465b-a972-649b41d9870e	2	10.126300	0.087004	6.057070	42.70380	19.81920	15.360900	5.845700	happy
3	baa26895-85b2-465b-a972-649b41d9870e	3	37.344900	0.427457	2.784040	16.53680	35.73190	0.534506	6.640390	angry
4	baa26895-85b2-465b-a972-649b41d9870e	4	0.003088	0.000003	0.002681	98.51810	1.47585	0.000055	0.000212	happy

After doing some minor changes in the code of Candidate 1, I plot various graphs for performing EDA on Candidate 2.

Finding the data type, missing and unique values of each column using the Autoviz library.

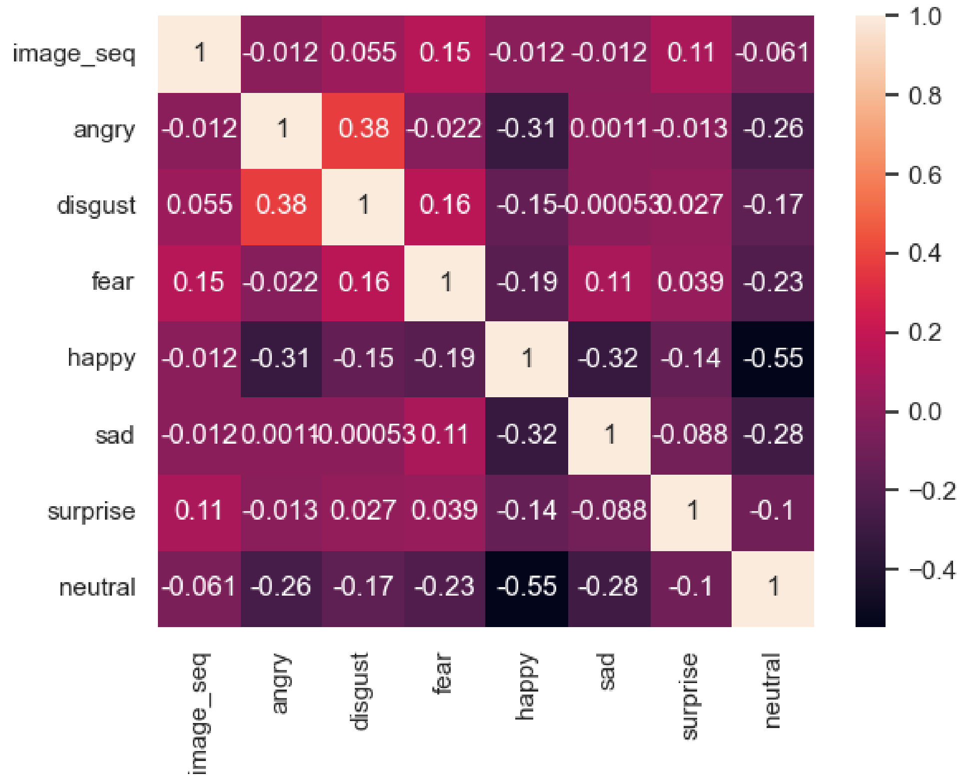
	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
movie_id	object	0.000000	1	nan	nan	Zero-variance colum: drop before modeling process.
image_seq	int64	0.000000	100	0.000000	86.000000	Possible ID colum: drop before modeling process.
angry	float64	0.000000	NA	0.000015	95.056300	has 14 outliers greater than upper bound (15.82) or lower than lower bound(-9.34). Cap them or remove them.
disgust	float64	0.000000	NA	0.000000	12.700100	has 13 outliers greater than upper bound (0.22) or lower than lower bound(-0.13). Cap them or remove them.
fear	float64	0.000000	NA	0.000002	82.107700	has 13 outliers greater than upper bound (12.16) or lower than lower bound(-7.04). Cap them or remove them.
happy	float64	0.000000	NA	0.011498	99.981800	No issue
sad	float64	0.000000	NA	0.005248	97.396900	has 2 outliers greater than upper bound (70.31) or lower than lower bound(-39.74). Cap them or remove them.
surprise	float64	0.000000	NA	0.000001	99.769100	has 14 outliers greater than upper bound (0.69) or lower than lower bound(-0.41). Cap them or remove them.
neutral	float64	0.000000	NA	0.000104	99.854700	No issue
dominant_emotion	object	0.000000	6	nan	nan	No issue

Calculating basic statistics for each emotion, such as mean, median, standard deviation, and range, to get an overview of the candidates' emotional expressions.

shape = (87, 10)

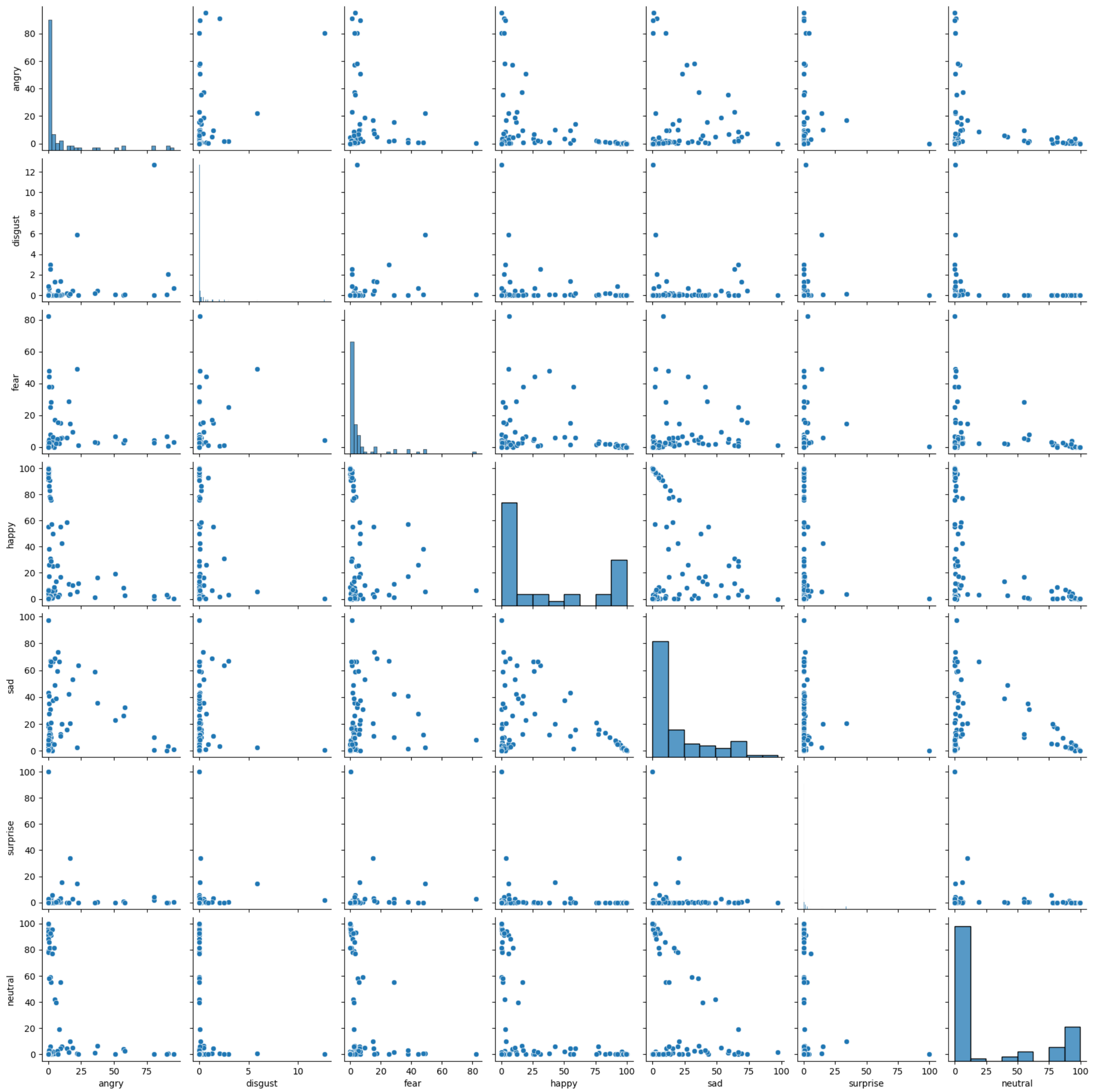
	image_seq	angry	disgust	fear	happy	sad	surprise	neutral
count	87.000000	87.000000	8.700000e+01	87.000000	87.000000	87.000000	8.700000e+01	87.000000
mean	43.000000	10.395041	3.918153e-01	6.747536	35.063288	18.558391	2.267330e+00	26.576600
std	25.258662	22.483989	1.555114e+00	13.716758	40.139694	23.306587	1.140703e+01	38.906958
min	0.000000	0.000015	2.198970e-10	0.000002	0.011498	0.005248	8.514920e-07	0.000104
25%	21.500000	0.098704	2.648140e-06	0.159929	1.595905	1.525660	5.336645e-03	0.122892
50%	43.000000	0.965159	5.163640e-04	1.824250	11.561100	7.884060	5.422400e-02	1.470510
75%	64.500000	6.389125	8.824245e-02	4.961165	80.356200	29.037500	2.809355e-01	58.685050
max	86.000000	95.056300	1.270010e+01	82.107700	99.981800	97.396900	9.976910e+01	99.854700

Plotting the Correlation matrix heatmap using seaborn library.



- The correlation coefficient of -0.31 suggests a moderate negative correlation between 'Angry' and 'Happy' emotions for Candidate 2. In practical terms, this means that when 'Angry' scores increase, 'Happy' scores tend to decrease, and vice versa.
- This negative correlation indicates that Candidate 2 is less likely to express both anger and happiness simultaneously. When they exhibit anger, they are less likely to display happiness, and when they exhibit happiness, they are less likely to express anger.

Plotting the scatterplot of each emotion with other emotion using pairplot.

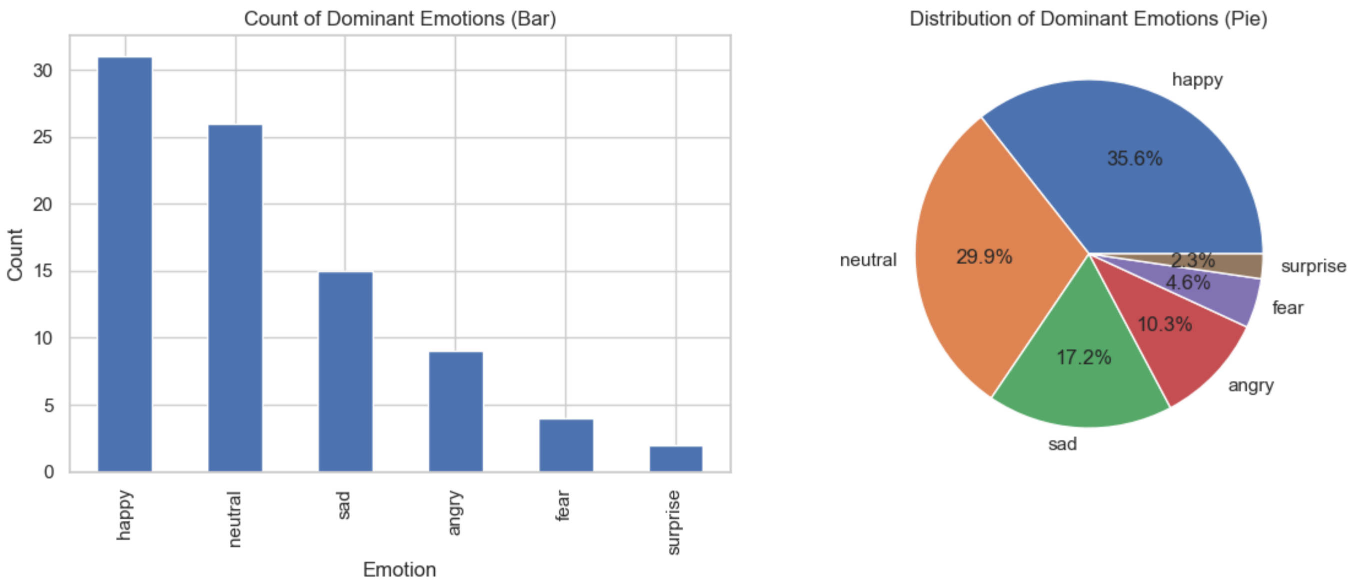


1. Happy Values: The scatterplot shows that Candidate 2's 'Happy' values tend to be generally high across the entire duration of the video. This indicates that the candidate consistently exhibits positive and happy emotions throughout the presentation.

2. Disgust Values: In contrast to 'Happy' values, 'Disgust' values for Candidate 2 are nearly zero for most of the video. This suggests that the candidate rarely expresses disgust and maintains a relatively neutral or non-disgusted demeanor.

3. Sad and Angry Values: Candidate 2's 'Sad' and 'Angry' values are notably high, indicating the frequent expression of these negative emotions. This suggests that the candidate experiences and displays sadness and anger during the video presentation.

Plotting a bar chart and pie chart showing the occurences of dominant emotions.

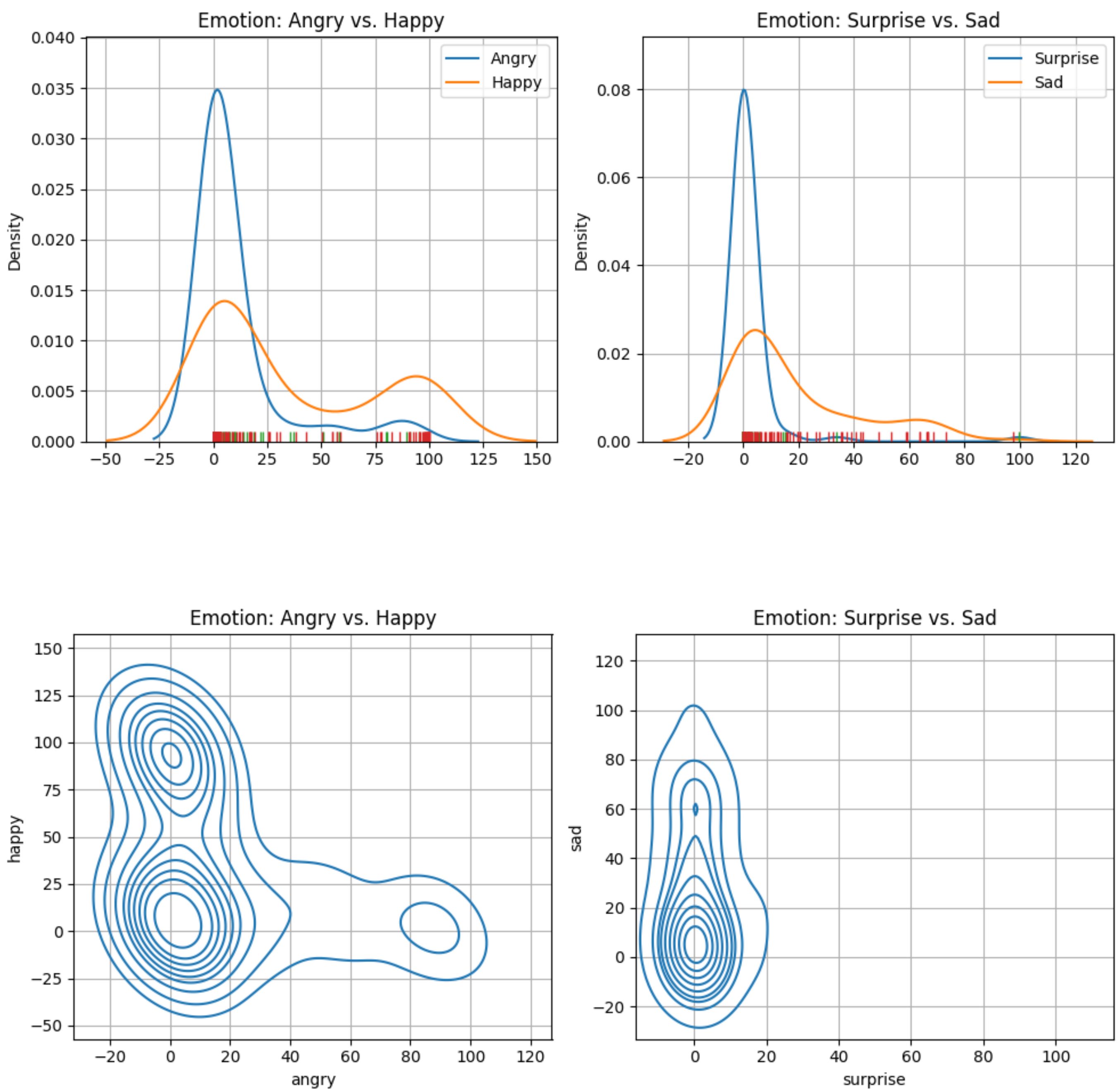


Based on the bar and pie charts, here are the key observations for Candidate 2:

- Happy Emotion: Candidate 2's emotional expressions are primarily characterized by 'Happy' emotions, constituting a significant portion of their emotional state (approximately 35.6%). This suggests that the candidate frequently exhibits positive and cheerful emotions during the introduction video.

- Negative Emotions: In addition to 'Happy' emotions, Candidate 2 also displays a substantial presence of negative emotions, collectively accounting for more than 30% of their emotional expressions. Specifically, 'Fear,' 'Anger,' and 'Sadness' are prevalent, indicating that the candidate experiences and expresses these negative emotions during the video presentation.

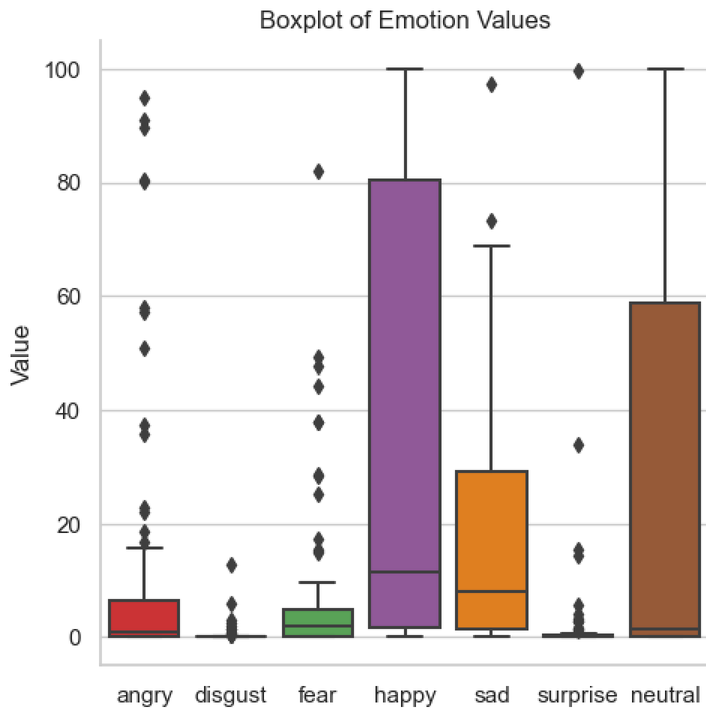
Now plotting the KDE plot between the 2 negative and 2 positive emotions to analyze the density.

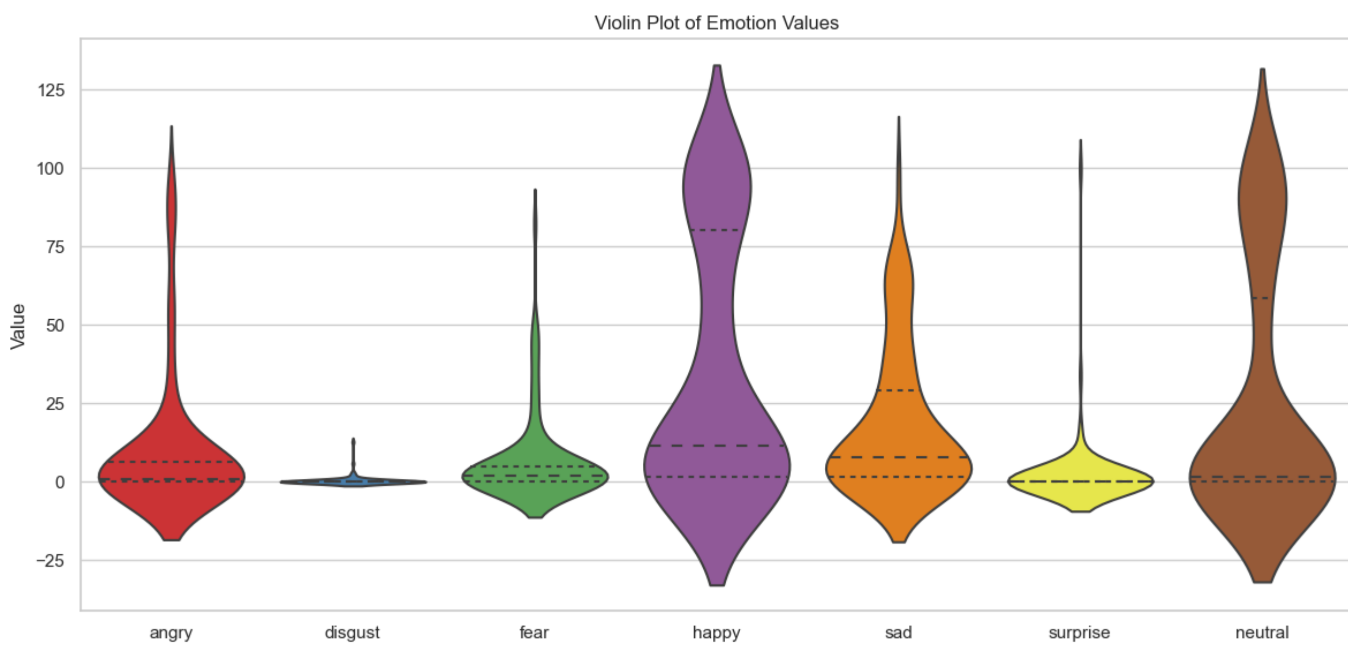


1. Happy vs. Angry: Candidate 2's emotional expressions show that 'Happy' values are generally higher than 'Angry' values. This suggests that the candidate tends to exhibit positive and happy emotions more frequently than expressions of anger.

2. Surprise vs. Sad: Additionally, the KDE plots reveal that the density of 'Surprise' values is consistently low, indicating that Candidate 2 rarely expresses high levels of surprise during the video presentation. This suggests that the candidate's emotional reactions involving surprise are infrequent or subdued.

Plotting the box plot and violin plot to find the outliers in the data.





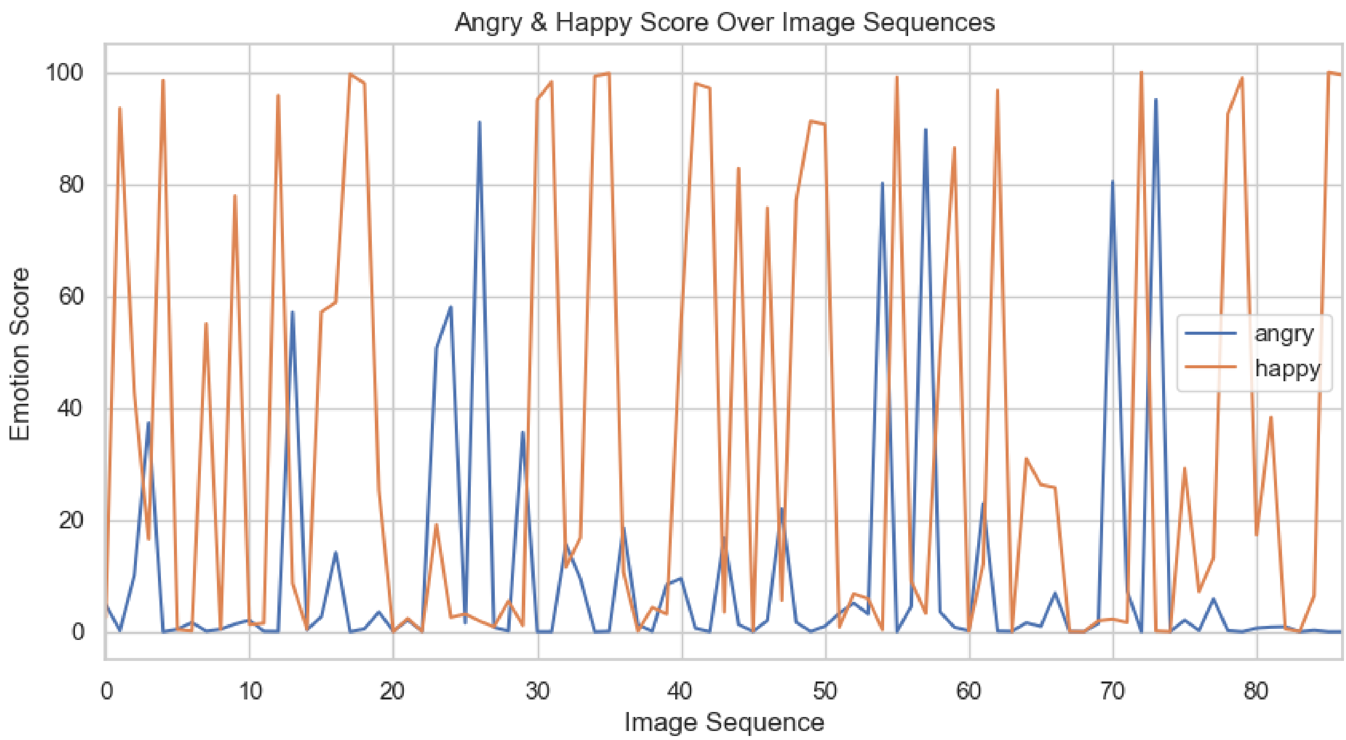
The box and violin plots for Candidate 2's emotional expressions reveal the presence of outliers in emotions other than 'Happy.' Here are the key observations:

- Happy: The 'Happy' emotion appears to have a relatively narrow and symmetric distribution, with no significant outliers. This suggests that Candidate

2's expressions of happiness are relatively consistent and do not deviate significantly from the median.

- Outliers in Other Emotions: In contrast, all other emotions, such as 'Fear,' 'Anger,' and 'Sadness,' exhibit a notable presence of outliers. These outliers indicate instances where Candidate 2's emotional expressions deviate significantly from the median or central tendency.

Plotting line chart of emotion scores over image sequences.



Happy Values: The line plot shows that 'Happy' values tend to exhibit more noticeable drift or variation over the course of the image sequences. This suggests that Candidate 2's expressions of happiness fluctuate and may vary in intensity or frequency throughout the video presentation.

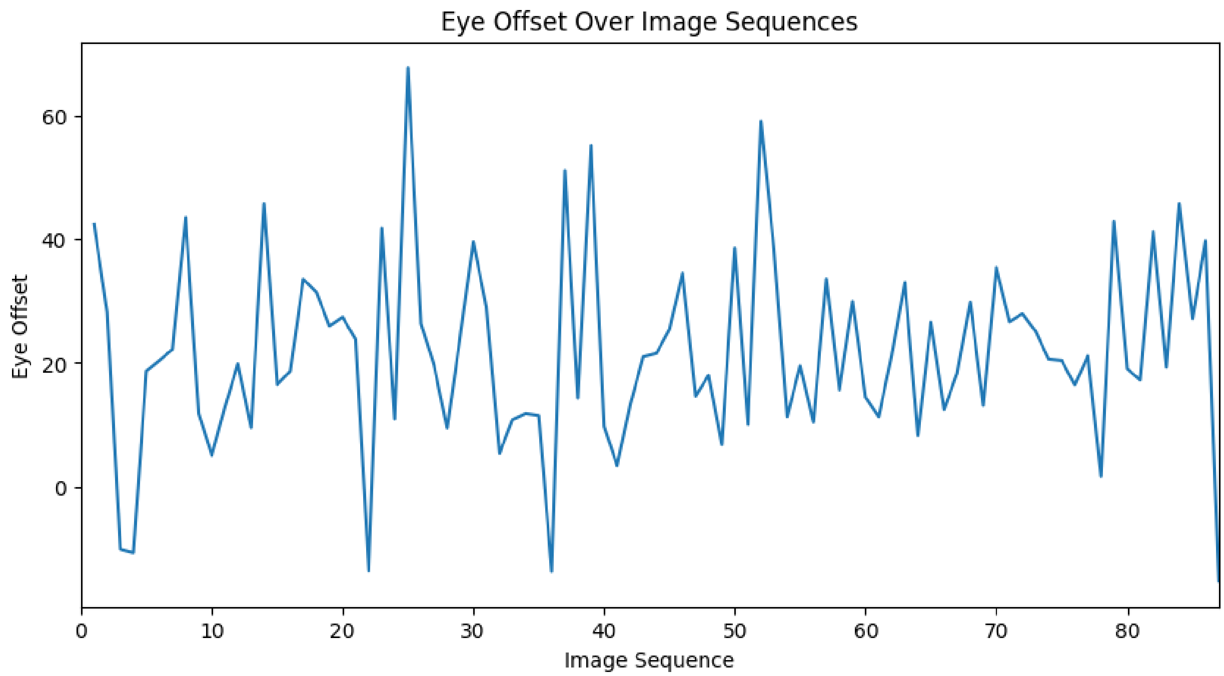
Angry Values: In contrast, 'Angry' values appear to exhibit less drift or variation over the image sequences. This implies that the candidate's expressions of anger remain relatively stable or consistent throughout the video.

Importing the data.

```
gaze = pd.read_csv(r".\emotion_data\2\gaze.csv")
gaze.head()
```

	movie_id	image_seq	gaze	blink	eye_offset
0	baa26895-85b2-465b-a972-649b41d9870e	1	0	1	42.3816
1	baa26895-85b2-465b-a972-649b41d9870e	2	0	0	28.1727
2	baa26895-85b2-465b-a972-649b41d9870e	3	1	0	-10.0732
3	baa26895-85b2-465b-a972-649b41d9870e	4	1	0	-10.6335
4	baa26895-85b2-465b-a972-649b41d9870e	5	1	0	18.6988

Plotting line chart of eye_offset over image sequences.



Most of the time, the eye of the candidate is in angle between 0-40°.

2.3 Candidate 3

Importing emotion file

```
emotion=pd.read_csv(r".\emotion_data\3\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	0	0.258826	2.532100e-05	41.1414	15.87730	33.46870	0.095324	9.15839	fear
1	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	1	0.016925	1.391500e-07	63.5732	28.61190	1.63691	0.569473	5.59155	fear
2	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	2	0.095944	2.349850e-06	35.4845	9.79849	13.52990	8.988050	32.10300	fear
3	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	3	1.411020	3.130660e-04	43.9037	6.15238	17.37470	30.132300	1.02557	fear
4	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	4	1.127490	1.218520e-04	18.1006	56.51250	2.52799	2.319830	19.41150	happy

Analysing the data of Candidate 3:

Finding the data type, missing and unique values of each column using the Autoviz library.

	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
movie_id	object	0.000000	1	nan	nan	Zero-variance colum: drop before modeling process.
image_seq	int64	0.000000	100	0.000000	99.000000	Possible ID colum: drop before modeling process.
angry	float64	0.000000	NA	0.000017	42.171200	has 13 outliers greater than upper bound (2.68) or lower than lower bound(-1.53). Cap them or remove them.
disgust	float64	0.000000	NA	0.000000	0.054649	has 18 outliers greater than upper bound (0.00) or lower than lower bound(-0.00). Cap them or remove them.
fear	float64	0.000000	NA	0.003125	95.139500	has 2 outliers greater than upper bound (87.14) or lower than lower bound(-48.12). Cap them or remove them.
happy	float64	0.000000	NA	0.002380	95.781900	has 5 outliers greater than upper bound (71.50) or lower than lower bound(-37.42). Cap them or remove them.
sad	float64	0.000000	NA	0.004126	98.309400	has 16 outliers greater than upper bound (22.65) or lower than lower bound(-12.79). Cap them or remove them.
surprise	float64	0.000000	NA	0.000029	91.265300	has 17 outliers greater than upper bound (8.73) or lower than lower bound(-4.99). Cap them or remove them.
neutral	float64	0.000000	NA	0.021739	99.918800	No issue
dominant_emotion	object	0.000000	6	nan	nan	No issue

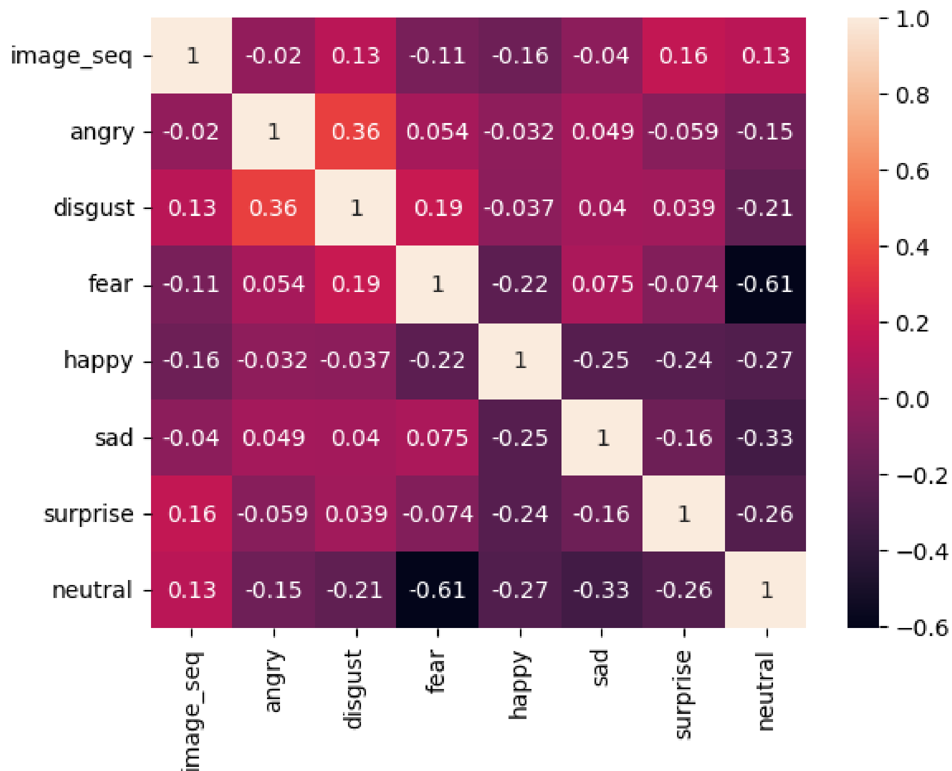
There are no missing values in the dataset. Only 2 columns, movie_id and dominant_emotion is of object data type each having 1 and 6 unique values As movie_id is same for a particular candidate, we can remove that column.

Calculating basic statistics for each emotion, such as mean, median, standard deviation, and range, to get an overview of the candidates' emotional expressions.

shape = (100, 10)

	image_seq	angry	disgust	fear	happy	sad	surprise	neutral
count	100.000000	100.000000	1.000000e+02	100.000000	100.000000	100.000000	100.000000	100.000000
mean	49.500000	1.531082	1.846983e-03	21.633298	21.428420	10.295898	7.268728	37.840726
std	29.011492	4.683116	7.318047e-03	24.906563	23.409376	17.599308	19.010296	32.812847
min	0.000000	0.000017	1.924780e-12	0.003125	0.002380	0.004126	0.000029	0.021739
25%	24.750000	0.048191	2.133243e-07	2.604662	3.423355	0.499891	0.153879	7.150482
50%	49.500000	0.384763	2.660785e-05	10.101240	13.237950	2.485550	0.931967	30.483550
75%	74.250000	1.101012	3.186135e-04	36.418800	30.653125	9.359340	3.586158	65.437775
max	99.000000	42.171200	5.464910e-02	95.139500	95.781900	98.309400	91.265300	99.918800

Plotting the Correlation matrix heatmap using seaborn library.



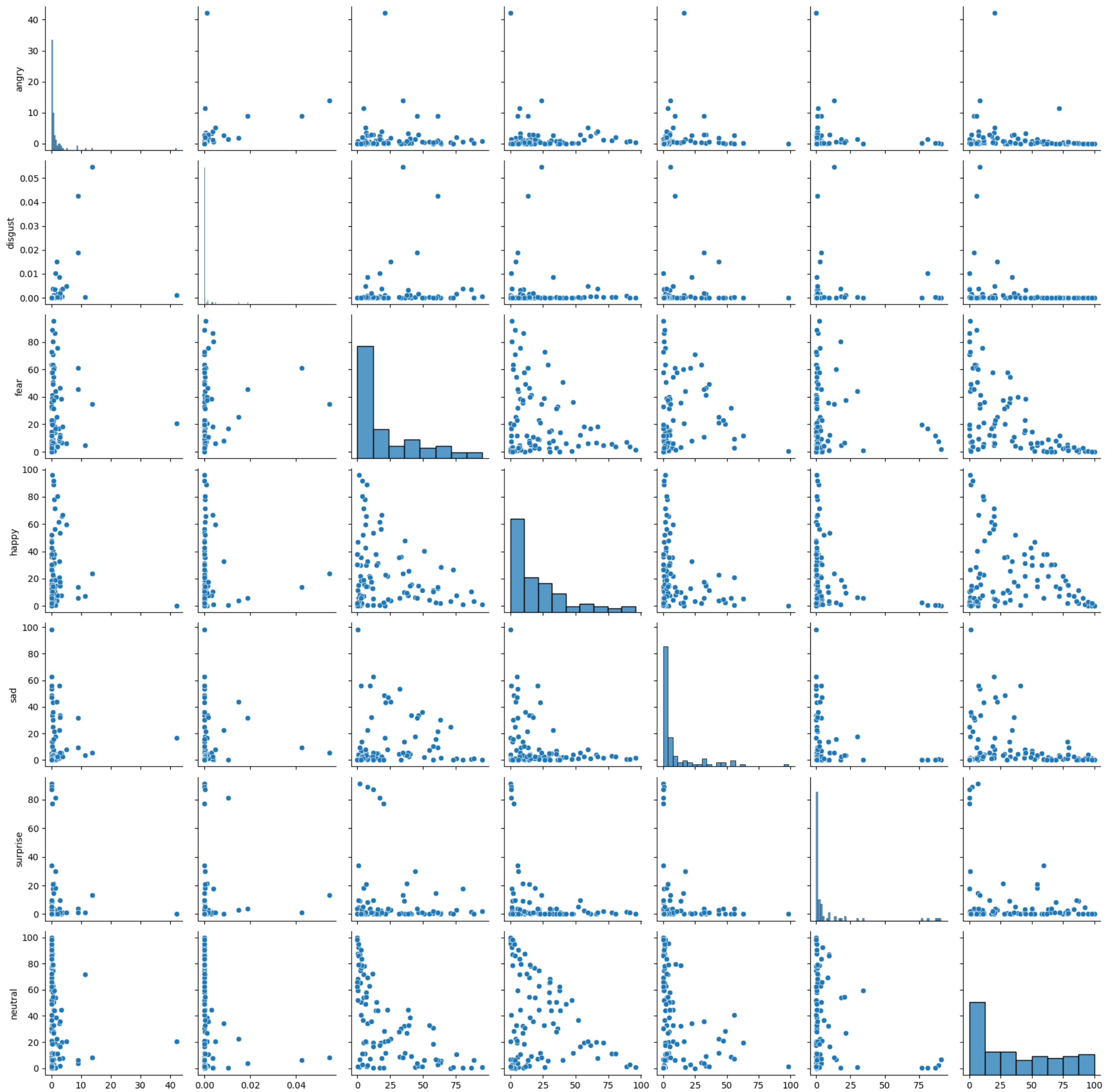
The correlation matrix for Candidate 3's emotional expressions reveals the following notable correlations:

1. Happy and Surprise: There is a correlation of -0.24 between 'Happy' and 'Surprise' emotions. This negative correlation suggests an inverse relationship between these emotions for Candidate 3. In practical terms, when 'Happy' scores increase, 'Surprise' scores tend to decrease, and vice versa. This contrast might

indicate that the candidate's happiness and surprise expressions are somewhat contradictory.

2. Happy and Sad: Another negative correlation is observed between 'Happy' and 'Sad' emotions, with a correlation coefficient of -0.25. This implies that when 'Happy' scores increase, 'Sad' scores tend to decrease, and vice versa. This suggests that the candidate's expressions of happiness and sadness are inversely related.

Plotting the scatterplot of each emotion with other emotion using pairplot.

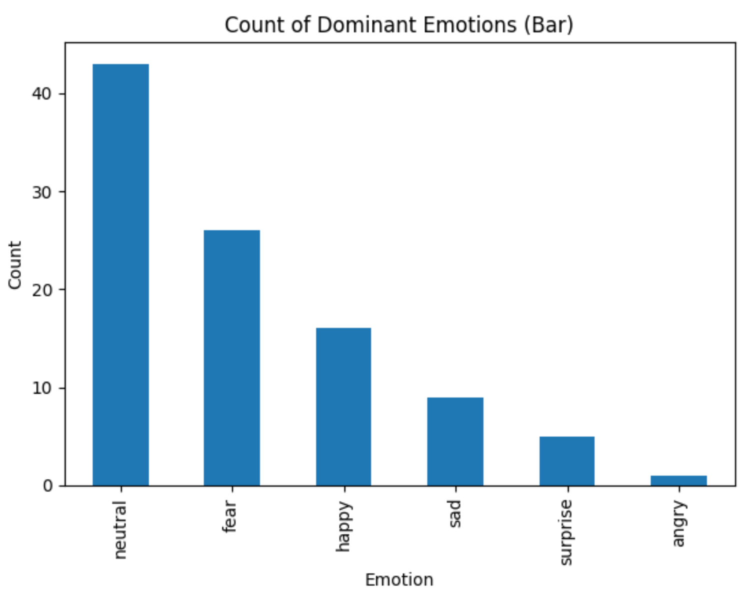


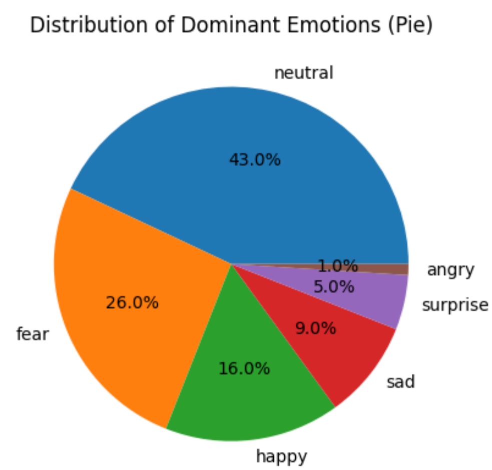
1. Fear Values: The scatterplot indicates that Candidate 3's 'Fear' values exhibit very large values, suggesting significant and potentially intense expressions of fear during the video presentation.

2. Happy Values: In comparison to other emotions, 'Happy' values also appear to be somewhat large, suggesting a considerable presence of positive and cheerful emotions during the video.

3. Surprise Values: 'Surprise' values show the presence of outliers, indicating moments of potentially heightened surprise that cannot be easily justified within the context of the video. These outliers suggest that Candidate 3 experienced particular instances of strong surprise reactions.

Plotting a bar chart and pie chart showing the occurences of dominant emotions.



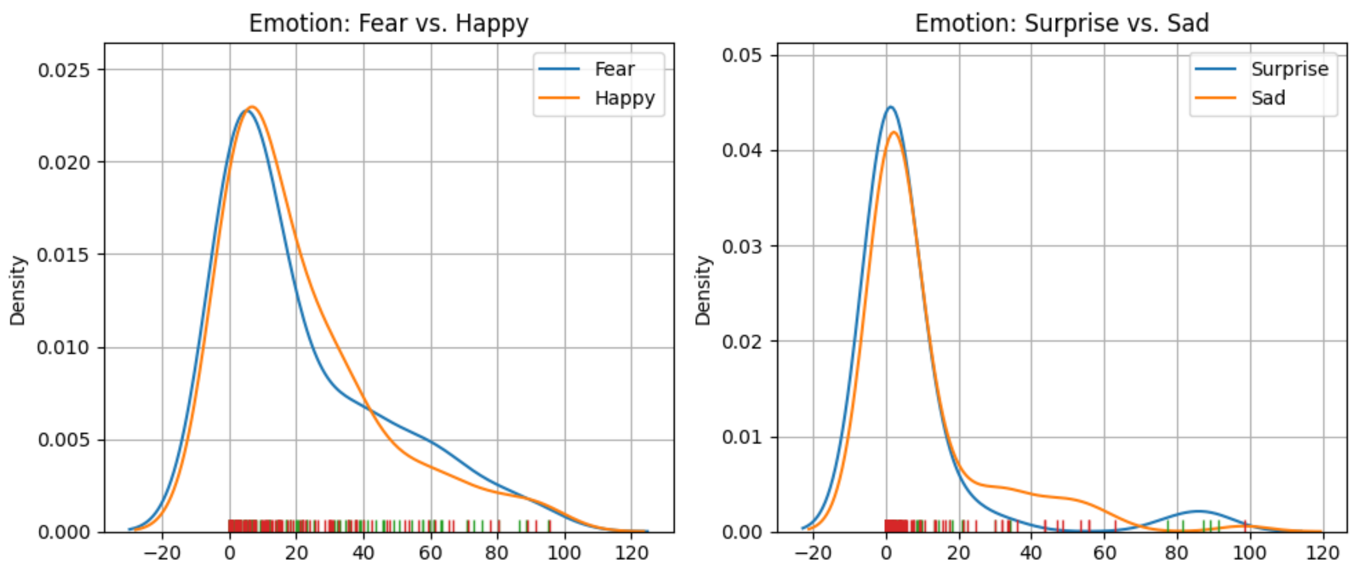


- Neutral Emotion: The most prevalent emotional state for Candidate 3 is 'Neutral,' accounting for a substantial 43% of their overall emotional expressions. This indicates that the candidate maintains a composed and unemotional demeanor for a significant portion of the video presentation.

- Fear Emotion: Following 'Neutral,' Candidate 3 exhibits a notable presence of 'Fear' emotions, constituting approximately 26% of their emotional expressions. This suggests that the candidate experiences and expresses fear or anxiety during various segments of the video.

- Happy Emotion: 'Happy' emotions are the third most prominent, representing around 16% of Candidate 3's emotional expressions. This indicates that the candidate frequently exhibits positive and cheerful emotions during certain parts of the video.

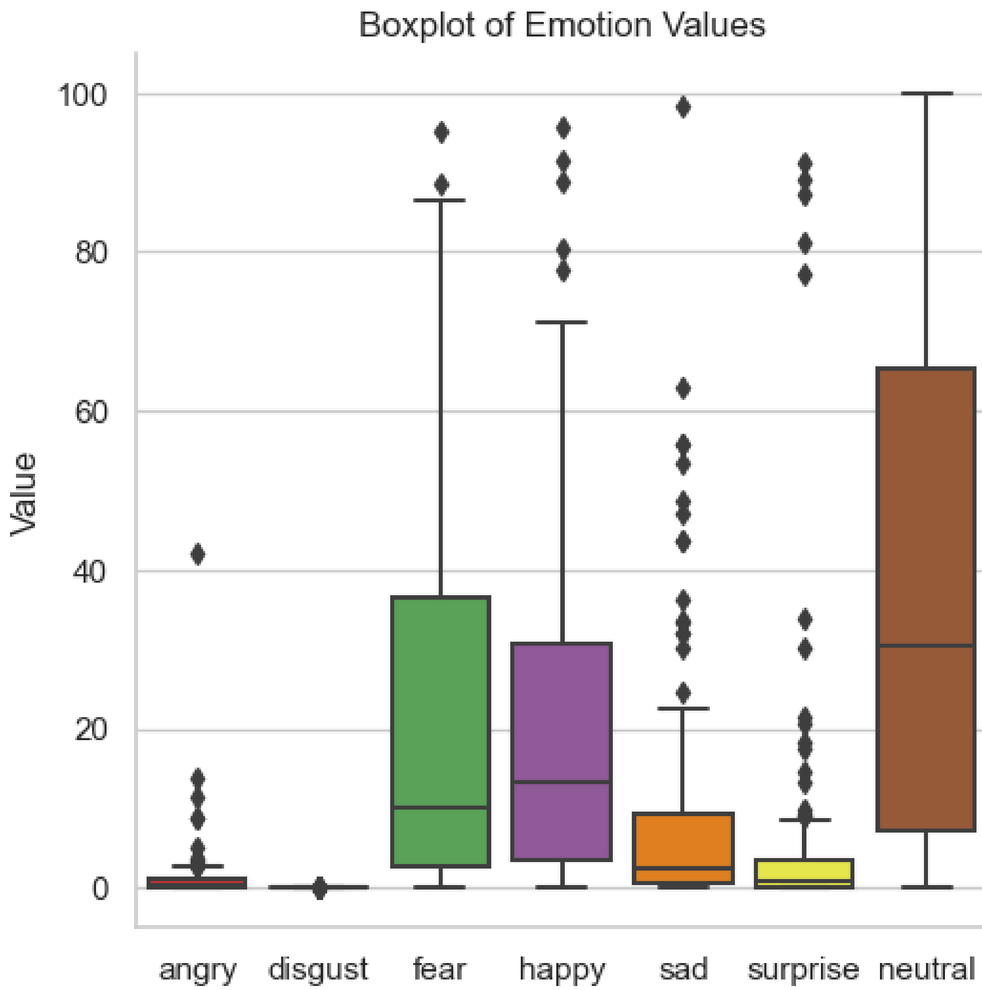
Now plotting the KDE plot between the 2 negative and 2 positive emotions to analyze the density.

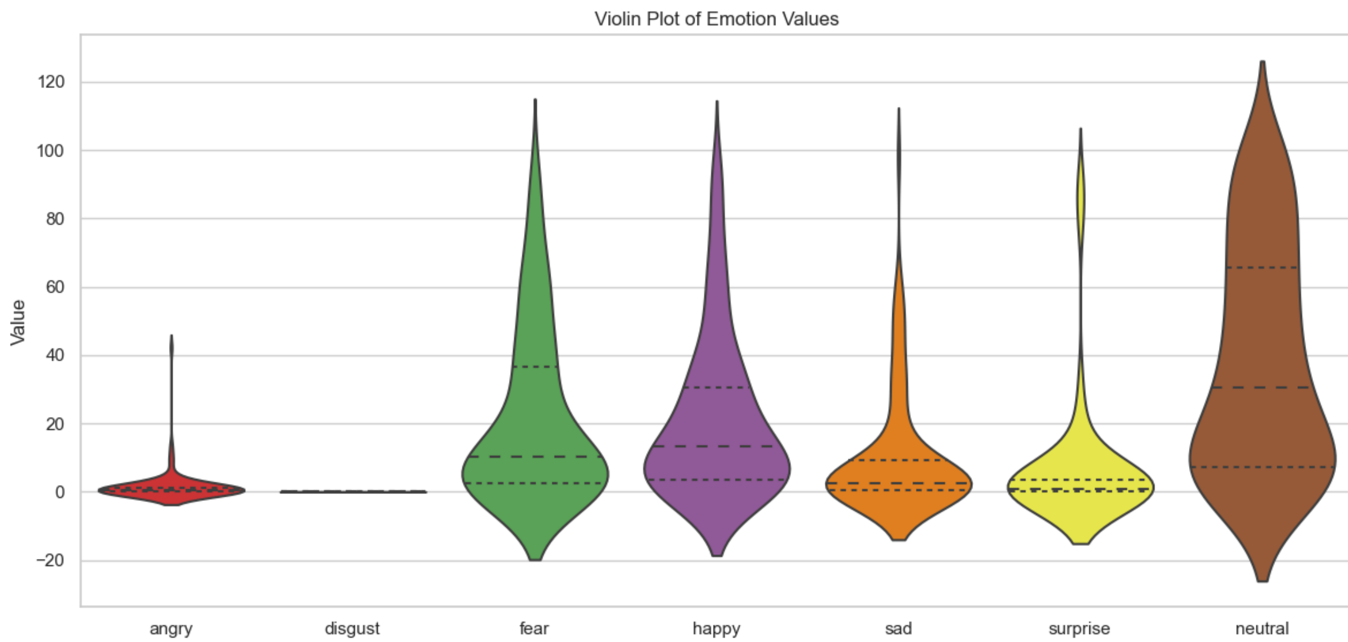


- Fear vs. Happy: The KDE plot indicates that the density of 'Fear' values is notably higher than 'Happy' values for values greater than 40. This suggests that Candidate 3 experiences and expresses fear more frequently and intensely when compared to happiness, especially for higher values.

- Sad vs. Surprise: Similarly, the KDE plot shows that 'Sad' values dominate 'Surprise' values for values greater than 20. This implies that Candidate 3's emotional expressions tend to be characterized by sadness over surprise when the emotional values are higher.

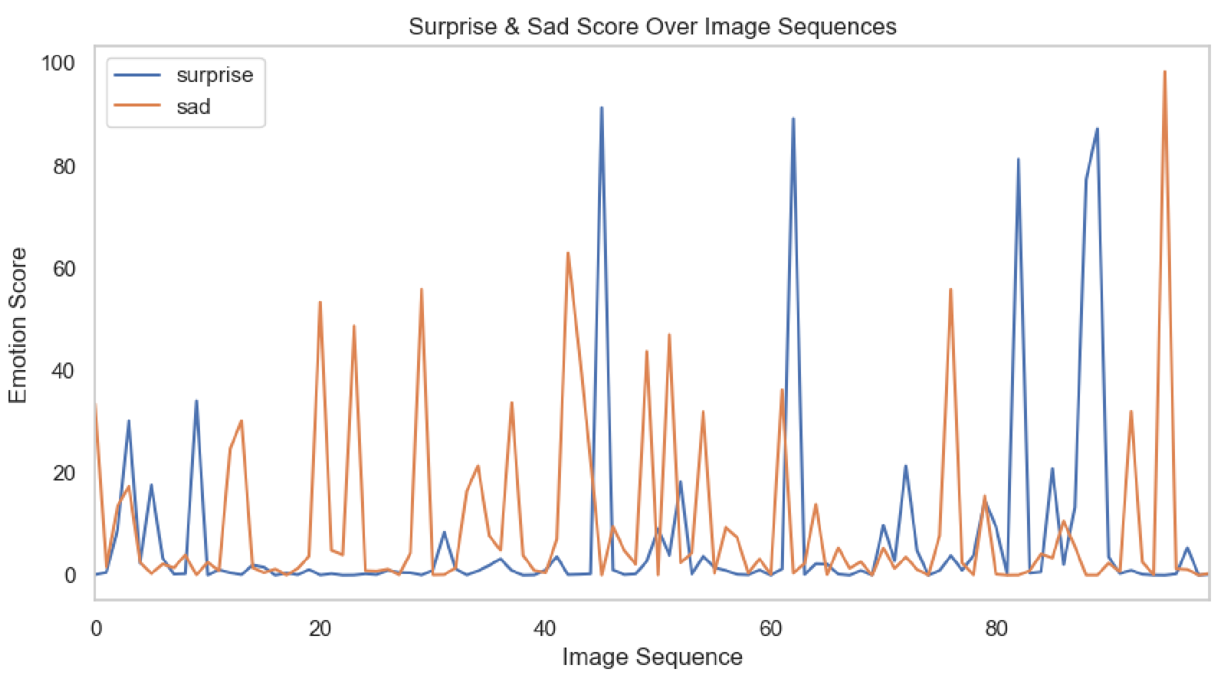
Plotting the box plot and violin plot to find the outliers in the data.

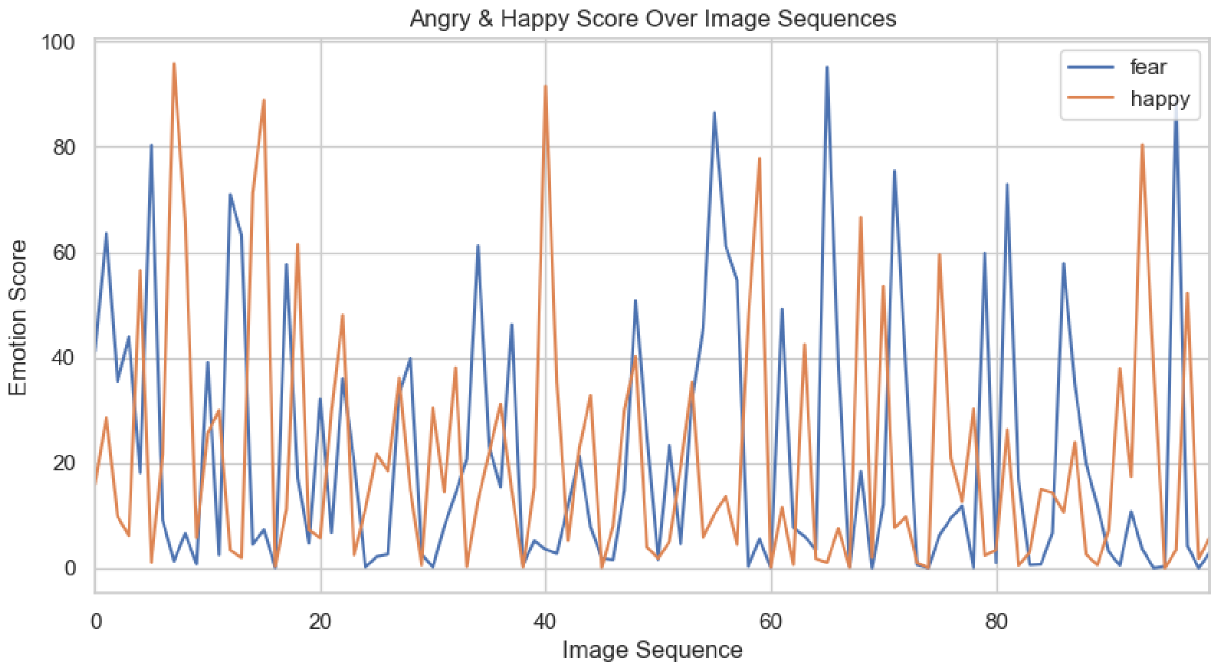




In all emotional categories, including fear, surprise, happy, and sad, there are noticeable outliers. These outliers indicate instances where Candidate 3's emotional expressions deviate significantly from the central tendency or median for each emotion.

Plotting line chart of emotion scores over image sequences.



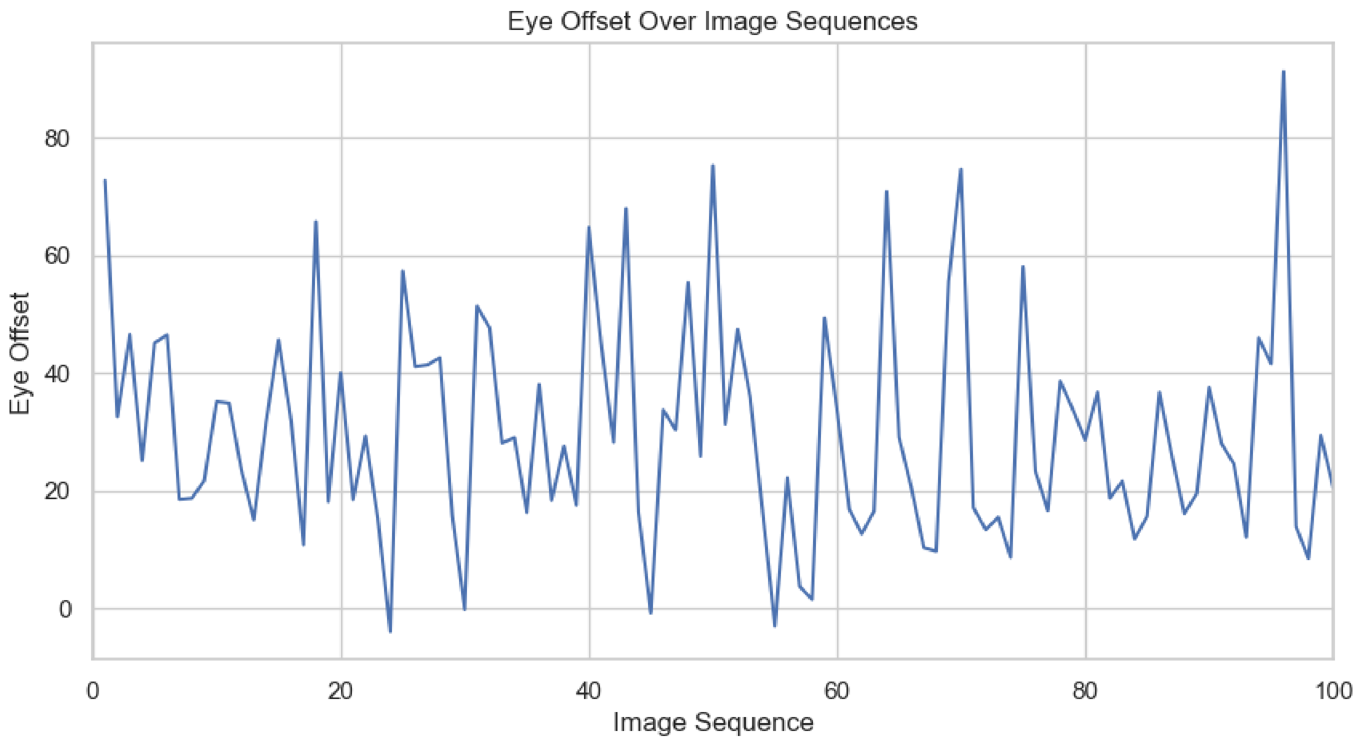


Importing the data.

```
gaze = pd.read_csv(r".\emotion_data\3\gaze.csv")
gaze.head()
```

	movie_id	image_seq	gaze	blink	eye_offset
0	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	1	0	1	72.7051
1	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	2	0	0	32.5648
2	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	3	0	0	46.5201
3	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	4	0	0	25.0913
4	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	5	0	0	45.0658

Plotting line chart of eye_offset over image sequences.



Most of the time, the eye of the candidate is in angle between 20-40°.

2.4 Candidate 4

Importing emotion file

```
emotion=pd.read_csv(r".\emotion_data\4\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	0	1.914030	2.009500e-07	2.105260	0.001374	0.179918	0.475532	95.3239	neutral
1	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	1	0.273406	2.415080e-06	0.046480	0.113288	0.156266	0.022630	99.3879	neutral
2	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	2	0.087041	2.285780e-10	0.025851	0.000082	0.186978	0.002026	99.6980	neutral
3	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	3	2.452160	9.030170e-07	0.333398	0.006685	1.721240	0.019011	95.4675	neutral
4	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	4	0.181937	2.037530e-07	0.018704	0.207840	0.002536	7.805310	91.7837	neutral

Finding the data type, missing and unique values of each column using the Autoviz library.

	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
movie_id	object	0.000000	1	nan	nan	Zero-variance colum: drop before modeling process.
image_seq	int64	0.000000	100	0.000000	99.000000	Possible ID colum: drop before modeling process.
angry	float64	0.000000	NA	0.000017	42.171200	has 13 outliers greater than upper bound (2.68) or lower than lower bound(-1.53). Cap them or remove them.
disgust	float64	0.000000	NA	0.000000	0.054649	has 18 outliers greater than upper bound (0.00) or lower than lower bound(-0.00). Cap them or remove them.
fear	float64	0.000000	NA	0.003125	95.139500	has 2 outliers greater than upper bound (87.14) or lower than lower bound(-48.12). Cap them or remove them.
happy	float64	0.000000	NA	0.002380	95.781900	has 5 outliers greater than upper bound (71.50) or lower than lower bound(-37.42). Cap them or remove them.
sad	float64	0.000000	NA	0.004126	98.309400	has 16 outliers greater than upper bound (22.65) or lower than lower bound(-12.79). Cap them or remove them.
surprise	float64	0.000000	NA	0.000029	91.265300	has 17 outliers greater than upper bound (8.73) or lower than lower bound(-4.99). Cap them or remove them.
neutral	float64	0.000000	NA	0.021739	99.918800	No issue
dominant_emotion	object	0.000000	6	nan	nan	No issue

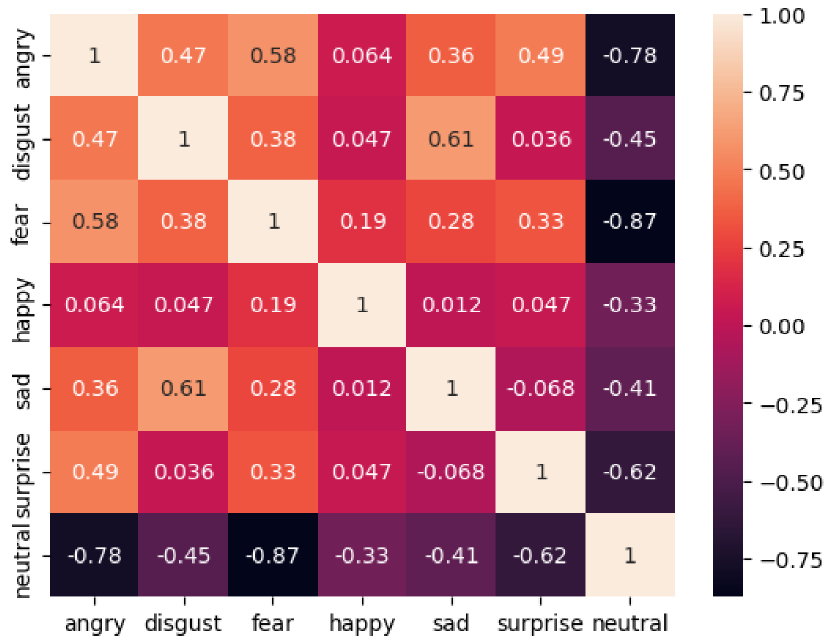
There are no missing values in the dataset. Only 2 columns, movie_id and dominant_emotion is of object data type each having 1 and 6 unique values As movie_id is same for a particular candidate, we can remove that column.

Calculating basic statistics for each emotion, such as mean, median, standard deviation, and range, to get an overview of the candidates' emotional expressions.

shape = (99, 10)

	image_seq	angry	disgust	fear	happy	sad	surprise	neutral
count	99.000000	99.000000	9.900000e+01	99.000000	99.000000	99.000000	99.000000	99.00000
mean	49.000000	1.734140	3.365219e-04	2.602291	0.572650	1.057942	1.403702	92.62894
std	28.722813	2.627746	1.533258e-03	6.784930	2.402168	2.492823	4.309405	12.46486
min	0.000000	0.000722	9.403450e-13	0.000546	0.000009	0.000251	0.000153	31.88270
25%	24.500000	0.096085	1.153220e-08	0.044912	0.002360	0.023519	0.004437	91.88845
50%	49.000000	0.603546	1.152230e-06	0.251244	0.021476	0.245196	0.050619	98.10390
75%	73.500000	2.105545	1.982120e-05	1.427960	0.228083	0.828137	0.546825	99.65325
max	98.000000	12.977300	1.074940e-02	54.999700	21.394600	17.941200	31.586400	99.99140

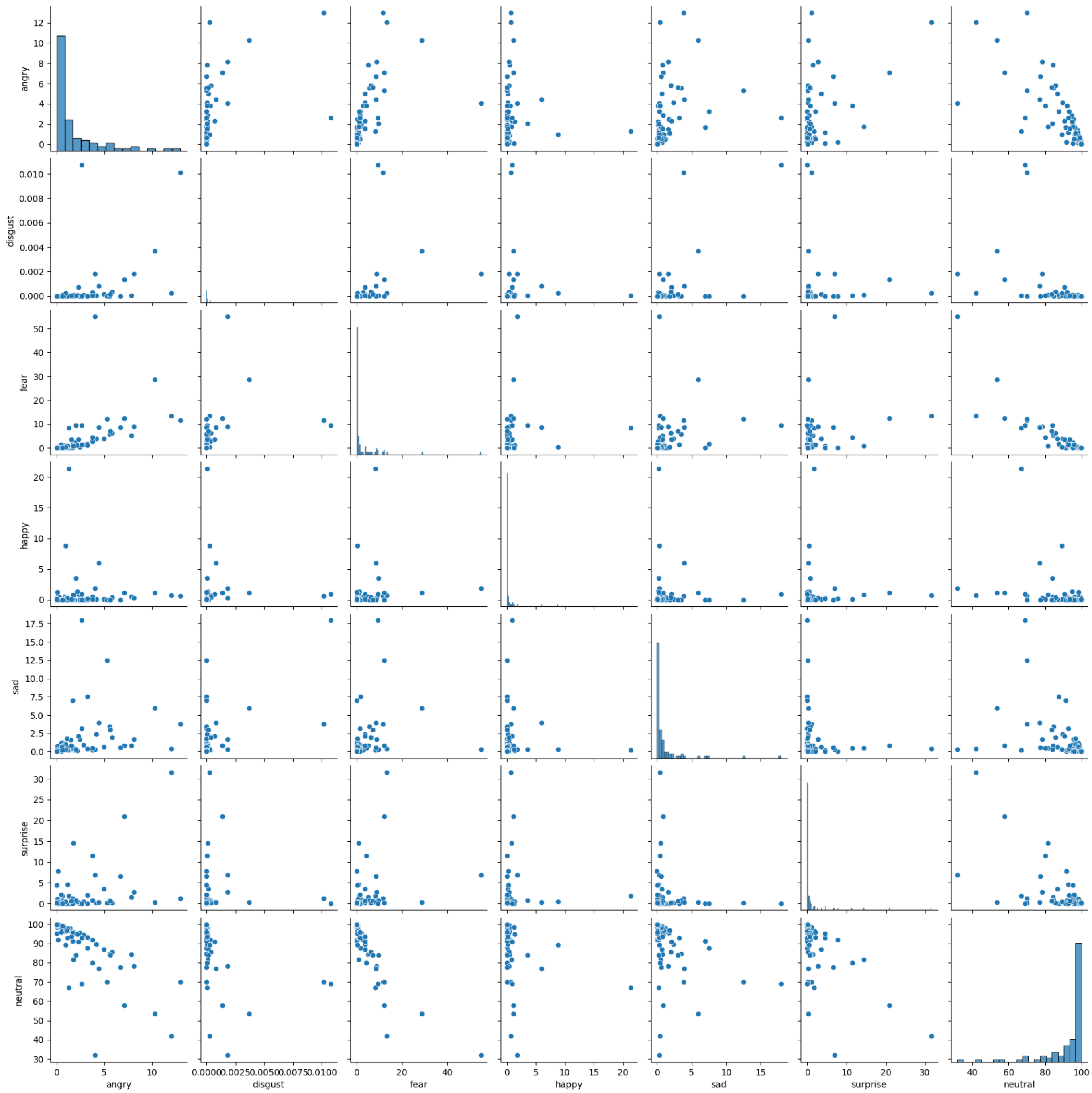
Plotting the Correlation matrix heatmap using seaborn library.



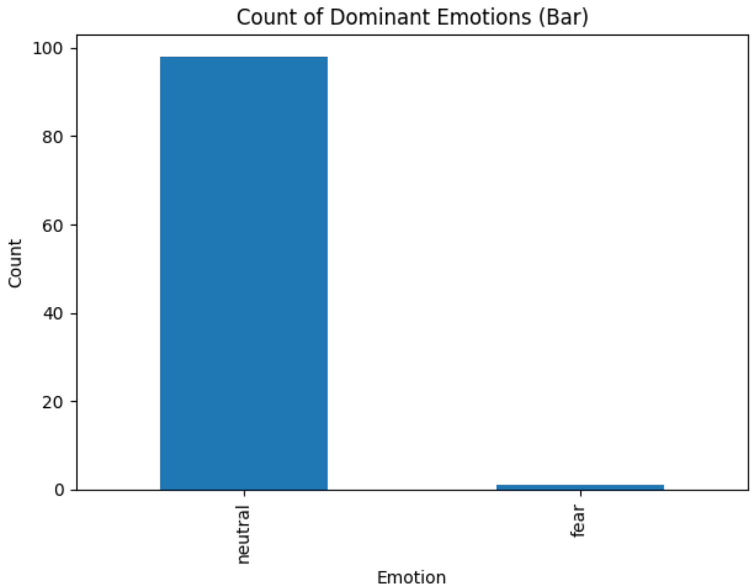
- Fear and Neutral Correlation: The strong negative correlation between 'Fear' and 'Neutral' emotions suggests an inverse relationship between these emotions for Candidate 4. In practical terms, when 'Fear' scores increase, 'Neutral' scores tend to decrease, and vice versa. This indicates that when the candidate experiences fear, they are less likely to exhibit a neutral emotional state.

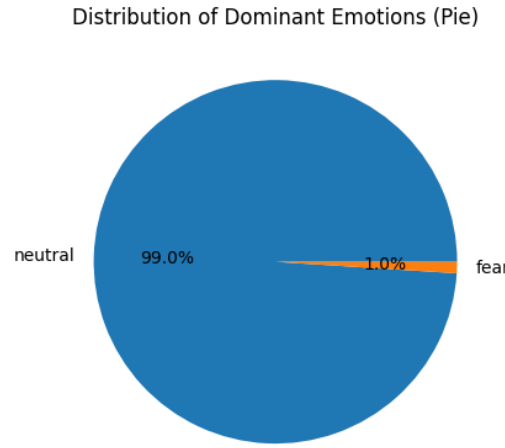
- Angry and Neutral Correlation: Similarly, the strong negative correlation between 'Angry' and 'Neutral' emotions implies that when 'Angry' scores increase, 'Neutral' scores tend to decrease, and vice versa. This indicates that the candidate's expressions of anger are inversely related to their expressions of a neutral emotional state.

Plotting the scatterplot of each emotion with other emotion using pairplot.



Plotting a bar chart and pie chart showing the occurences of dominant emotions.

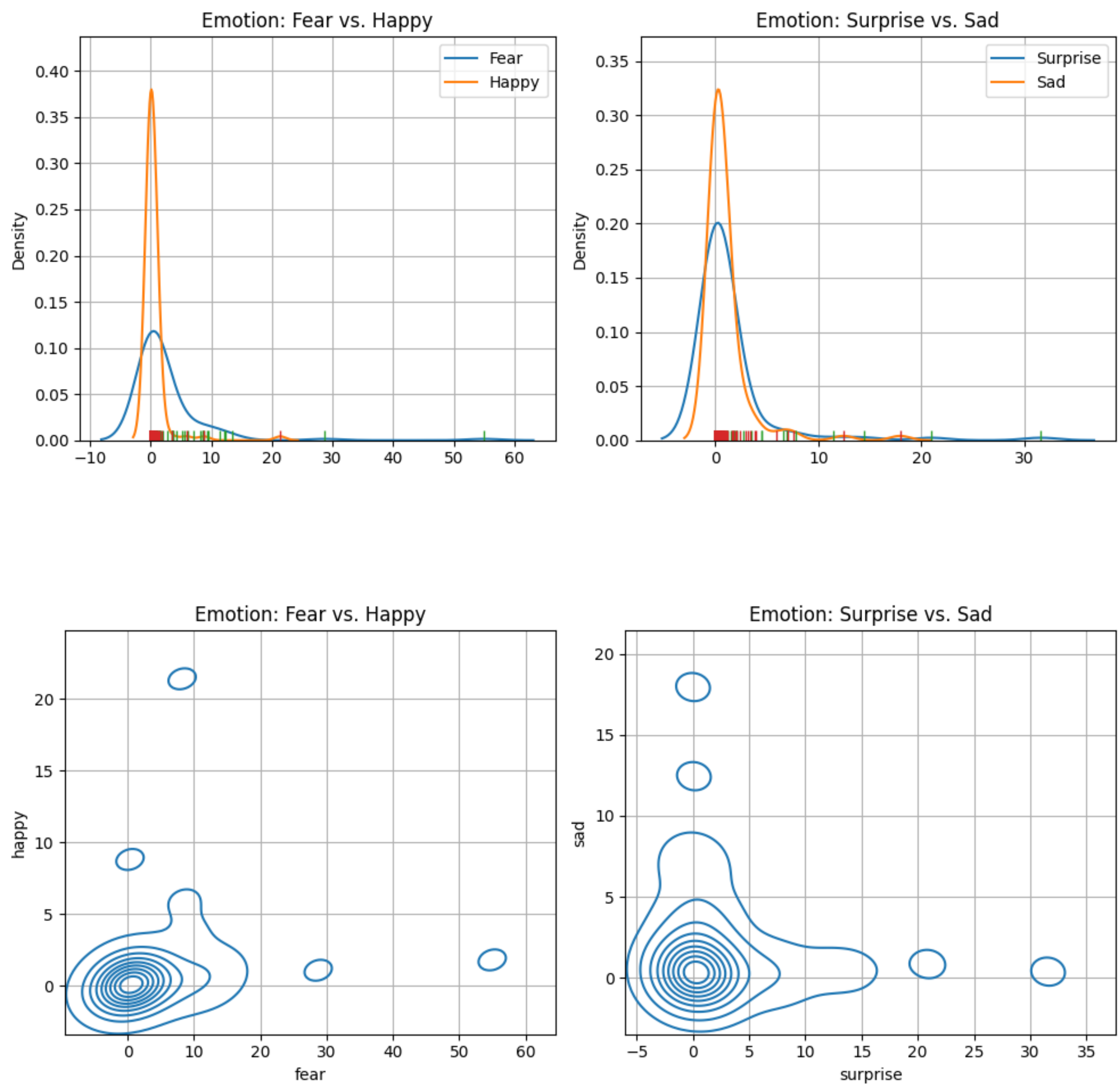




Neutral Emotion: The overwhelming majority of Candidate 4's emotional expressions, approximately 99%, are categorized as 'Neutral.' This indicates that the candidate maintains a highly composed and unemotional demeanor throughout the entire video presentation.

Fear Emotion: In contrast, 'Fear' emotions are only present to a minimal extent, constituting approximately 1% of Candidate 4's emotional expressions. This suggests that the candidate experiences only occasional moments of fear or anxiety during the video.

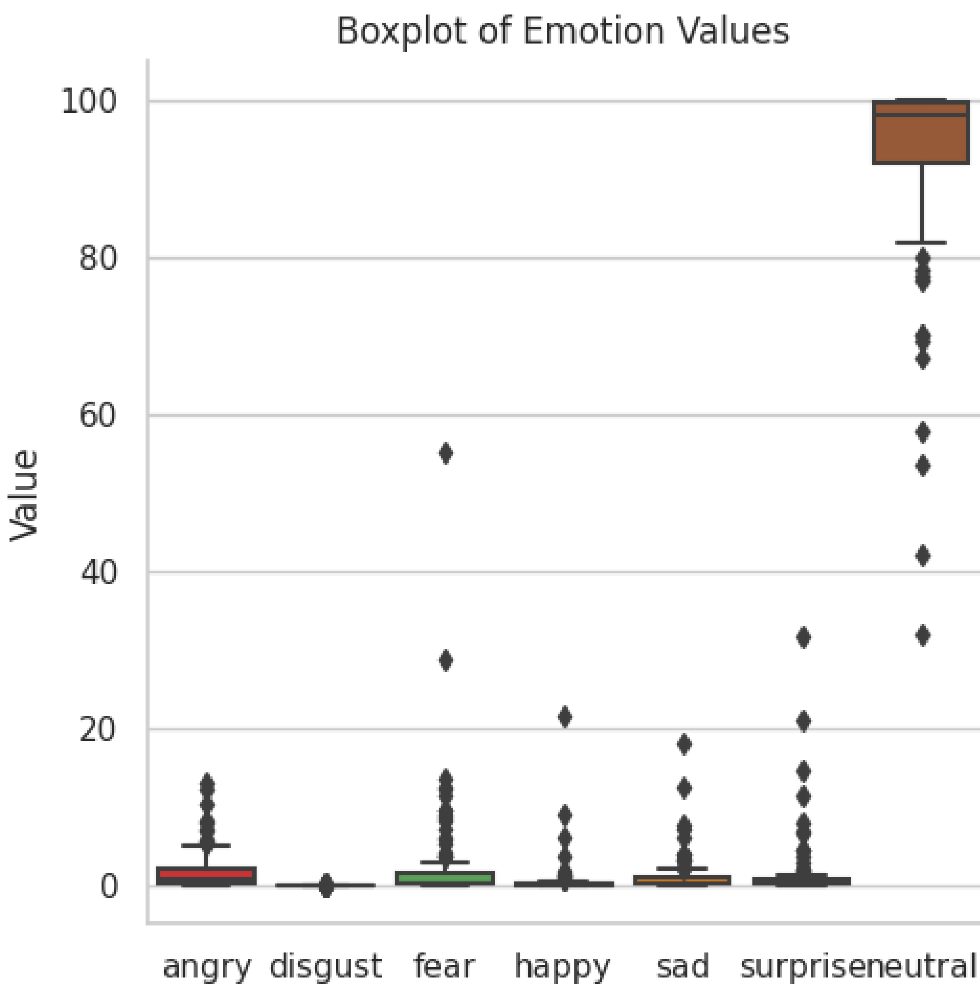
Now plotting the KDE plot between the 2 negative and 2 positive emotions to analyze the density.

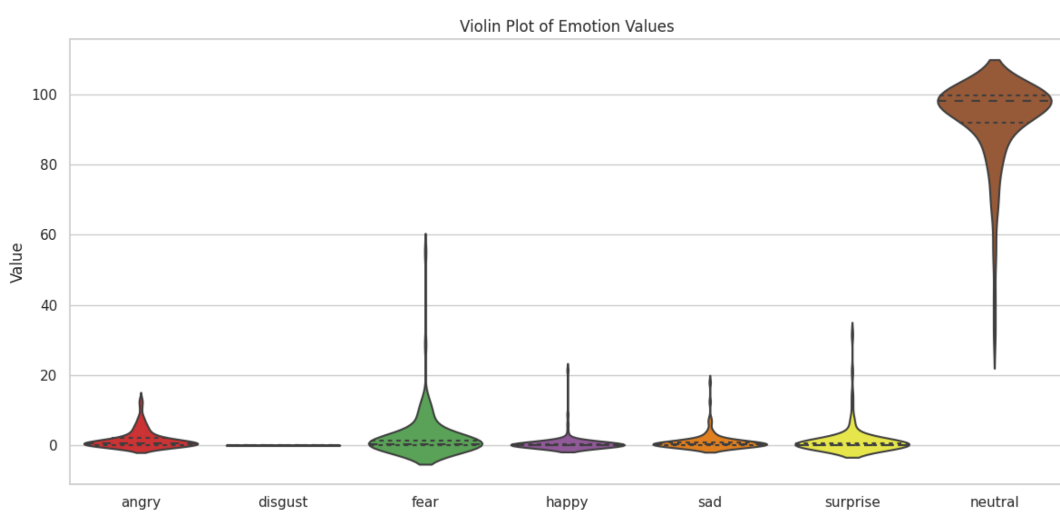


Fear and Happy: The KDE plot illustrates that the probability of occurrence of 'Fear' values and 'Happy' values is generally concentrated in the range of 0-10. This indicates that Candidate 4's emotional expressions of fear and happiness tend to be within a relatively low range, with very few instances of extremely high values.

Surprise and Sad: Similarly, the KDE plot shows that the probability of occurrence of 'Surprise' values and 'Sad' values is also predominantly in the range of 0-10. This suggests that the candidate's emotional expressions related to surprise and sadness are typically of low to moderate intensity, with limited occurrences of very high values.

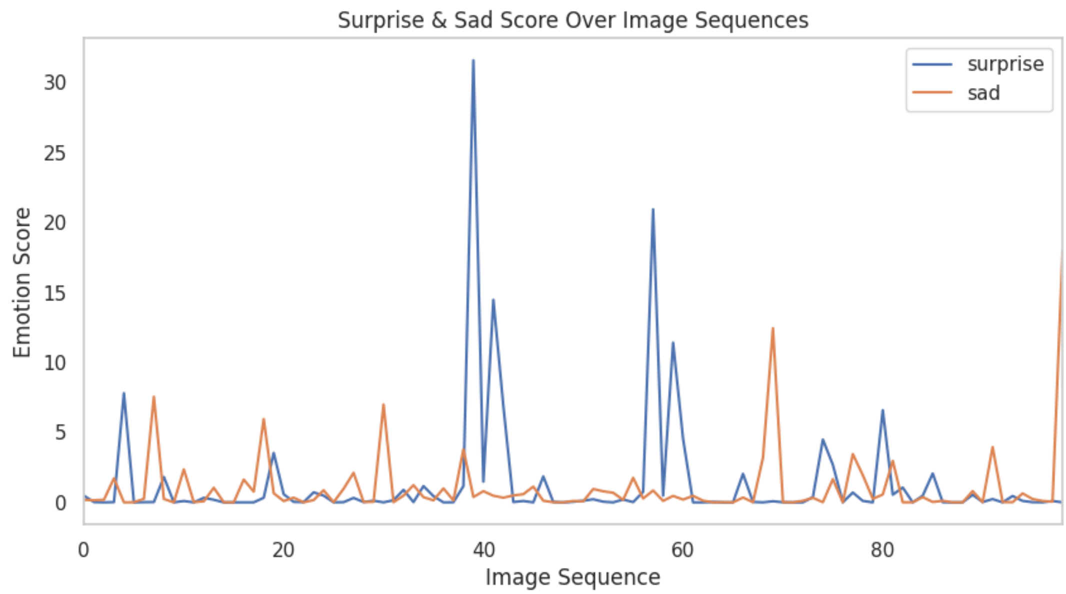
Plotting the box plot and violin plot to find the outliers in the data.





In all emotional categories, including fear, surprise, happy, and sad, there are noticeable outliers. These outliers indicate instances where Candidate 4's emotional expressions deviate significantly from the central tendency or median for each emotion.

Plotting line chart of emotion scores over image sequences.



The variation in 'Surprise' scores during the middle of the video implies that Candidate 4 goes through a significant shift in their emotional response, potentially reacting with heightened surprise to specific content or events during that portion of the presentation.

- Happiness: Candidate 4 exhibits very low levels of happiness throughout the video interval. This suggests that the candidate maintains a consistently low degree of positive or cheerful emotions during the presentation.
- Fear: While fear is generally not a dominant emotion for Candidate 4, there is a noticeable rise in fear scores during the middle of the video. However, this increase is temporary and occurs for a short period of time.

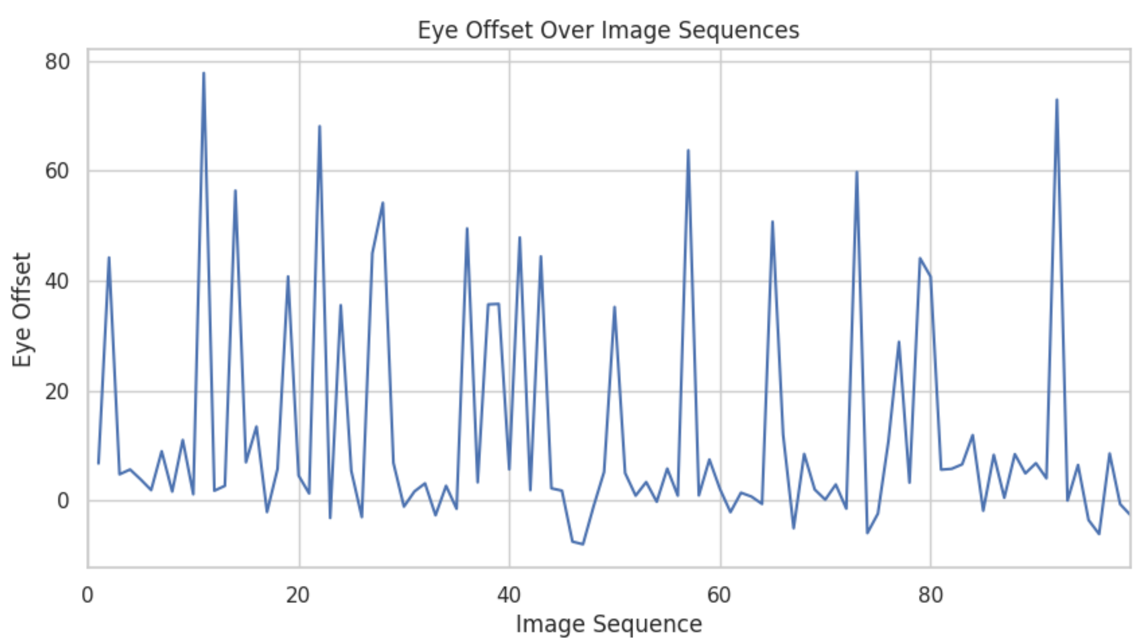
In summary, Candidate 4's emotional profile during the video is characterized by a sustained low level of happiness and a brief but noticeable spike in fear scores during a specific segment of the presentation

Importing the data.

```
gaze = pd.read_csv(r".\emotion_data\4\gaze.csv")
gaze.head()
```

	movie_id	image_seq	gaze	blink	eye_offset
0	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	1	1	0	6.7544
1	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	2	0	1	44.2002
2	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	3	1	0	4.7784
3	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	4	1	0	5.6502
4	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	5	1	0	3.8508

Plotting line chart of eye_offset over image sequences.



The eye offset deviations spanning a range of 0 to 60 degrees indicate that Candidate 4's gaze direction is more dynamic and tends to shift significantly during the video presentation.

2.5 Candidate 5

Importing emotion file

```python
emotion=pd.read_csv(r".\emotion_data\5\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	9c350343-e895-49df-af90-d50b91d19d3e	0	24.902800	6.655820e-05	0.810725	0.001305	2.511150	0.029861	71.7441	neutral
1	9c350343-e895-49df-af90-d50b91d19d3e	1	0.105126	2.913510e-11	0.000546	0.004444	0.020332	0.000050	99.8695	neutral
2	9c350343-e895-49df-af90-d50b91d19d3e	2	0.032465	4.927800e-11	0.000134	0.326047	0.076447	0.000012	99.5649	neutral
3	9c350343-e895-49df-af90-d50b91d19d3e	3	0.006253	3.979720e-10	0.000096	0.073862	0.008517	0.002353	99.9089	neutral

```python
emotion.shape
```

shape = (4, 10)

As very less values are available for Candidate 5, we cannot analyse the Candidate 5.

2.6 Candidate 6

Importing emotion file

```python
emotion=pd.read_csv(r".\emotion_data\6\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	92016995-e455-4651-9f6e-fbca0d423f21	0	0.001311	1.414360e-07	0.013233	37.1427	0.178181	0.008260	62.6563	neutral
1	92016995-e455-4651-9f6e-fbca0d423f21	1	0.001332	1.421510e-07	0.013304	36.9452	0.179674	0.008271	62.8523	neutral
2	92016995-e455-4651-9f6e-fbca0d423f21	2	0.001311	1.414390e-07	0.013234	37.1393	0.178144	0.008264	62.6598	neutral
3	92016995-e455-4651-9f6e-fbca0d423f21	3	0.001810	1.431440e-07	0.023875	11.0442	0.208299	0.002817	88.7190	neutral
4	92016995-e455-4651-9f6e-fbca0d423f21	4	0.001834	1.460720e-07	0.024232	11.0388	0.209914	0.002865	88.7224	neutral

```python
emotion.shape
```

shape = (14, 10)

As very less values are available for Candidate 6, we cannot analyse the Candidate 6.

2.7 Candidate 7

Importing emotion file

```
emotion=pd.read_csv(r".\emotion_data\7\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	6539370c-256e-4ed2-9d00-1be1f051163f	0	0.011716	2.442630e-03	0.350391	93.426900	4.564420	0.016044	1.628070e+00	happy
1	6539370c-256e-4ed2-9d00-1be1f051163f	1	6.587830	2.033840e-02	35.106600	0.103848	38.967200	0.001294	1.921290e+01	sad
2	6539370c-256e-4ed2-9d00-1be1f051163f	2	5.066870	7.610640e-05	1.008420	0.285020	29.728900	0.000373	6.391040e+01	neutral
3	6539370c-256e-4ed2-9d00-1be1f051163f	3	0.000539	1.678270e-05	0.041162	99.408400	0.239465	0.067905	2.425000e-01	happy
4	6539370c-256e-4ed2-9d00-1be1f051163f	4	0.000672	1.765070e-09	4.977040	0.001237	0.000022	95.021000	7.548270e-07	surprise

Finding the data type, missing and unique values of each column using the Autoviz library.

	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
movie_id	object	0.000000	1	nan	nan	Zero-variance colum: drop before modeling process.
image_seq	int64	0.000000	100	0.000000	99.000000	Possible ID colum: drop before modeling process.
angry	float64	0.000000	NA	0.000017	42.171200	has 13 outliers greater than upper bound (2.68) or lower than lower bound(-1.53). Cap them or remove them.
disgust	float64	0.000000	NA	0.000000	0.054649	has 18 outliers greater than upper bound (0.00) or lower than lower bound(-0.00). Cap them or remove them.
fear	float64	0.000000	NA	0.003125	95.139500	has 2 outliers greater than upper bound (87.14) or lower than lower bound(-48.12). Cap them or remove them.
happy	float64	0.000000	NA	0.002380	95.781900	has 5 outliers greater than upper bound (71.50) or lower than lower bound(-37.42). Cap them or remove them.
sad	float64	0.000000	NA	0.004126	98.309400	has 16 outliers greater than upper bound (22.65) or lower than lower bound(-12.79). Cap them or remove them.
surprise	float64	0.000000	NA	0.000029	91.265300	has 17 outliers greater than upper bound (8.73) or lower than lower bound(-4.99). Cap them or remove them.
neutral	float64	0.000000	NA	0.021739	99.918800	No issue
dominant_emotion	object	0.000000	6	nan	nan	No issue

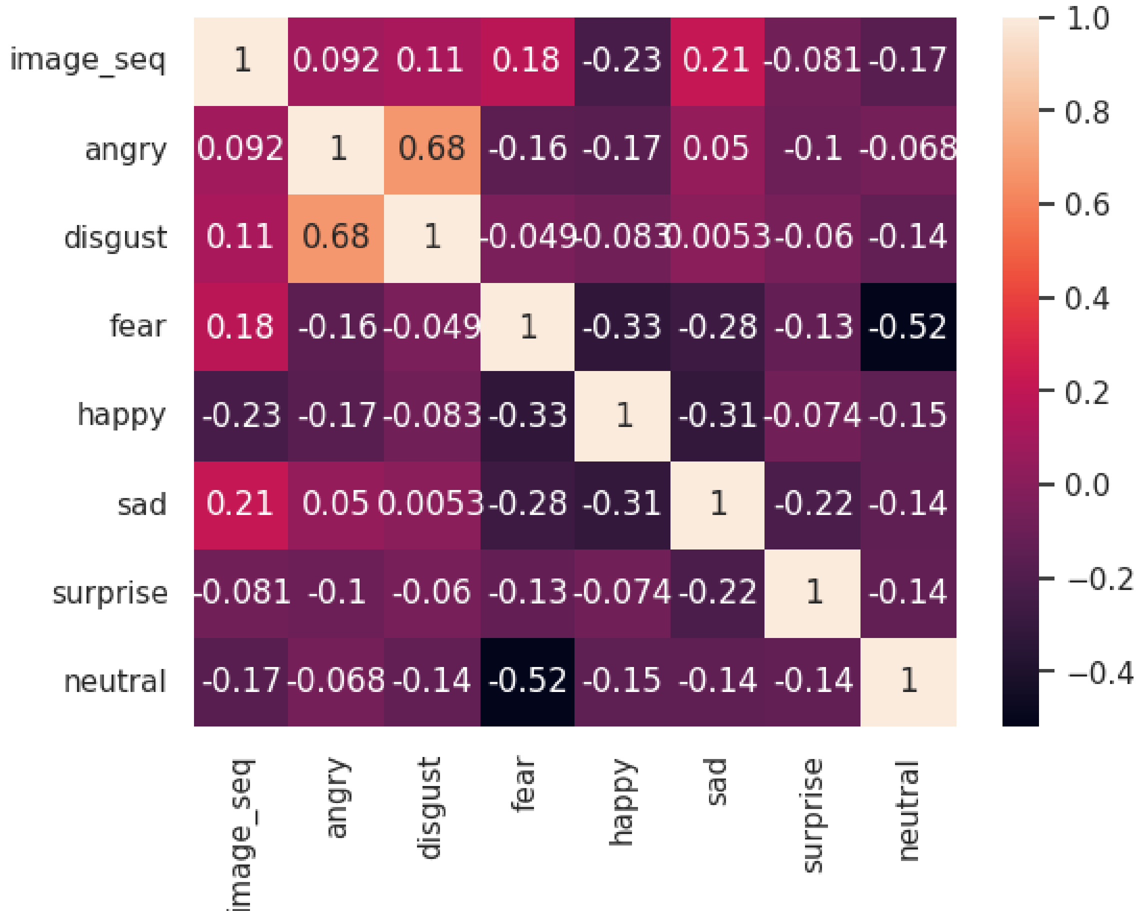
There are no missing values in the dataset. Only 2 columns, movie_id and dominant_emotion is of object data type each having 1 and 6 unique values As movie_id is same for a particular candidate, we can remove that column.

Calculating basic statistics for each emotion, such as mean, median, standard deviation, and range, to get an overview of the candidates' emotional expressions.

shape = (87, 10)

	image_seq	angry	disgust	fear	happy	sad	surprise	neutral
count	87.000000	87.000000	8.700000e+01	87.000000	87.000000	87.000000	87.000000	8.700000e+01
mean	43.000000	5.641183	1.507393e-01	41.652400	8.994113	23.106425	4.081041	1.637410e+01
std	25.258662	10.851536	5.674268e-01	31.527371	24.288680	22.893766	15.613091	2.606668e+01
min	0.000000	0.000026	1.632180e-09	0.002298	0.000115	0.000022	0.000004	7.548270e-07
25%	21.500000	0.268764	4.342120e-04	12.554650	0.092540	5.145540	0.002647	2.398415e-01
50%	43.000000	1.708550	8.027740e-03	34.581800	0.293205	16.238400	0.023693	3.981520e+00
75%	64.500000	5.755515	4.025450e-02	64.482900	1.591935	35.911650	0.295826	2.081150e+01
max	86.000000	59.135600	4.551240e+00	99.750200	99.956300	91.471200	95.021000	9.118620e+01

Plotting the Correlation matrix heatmap using seaborn library.

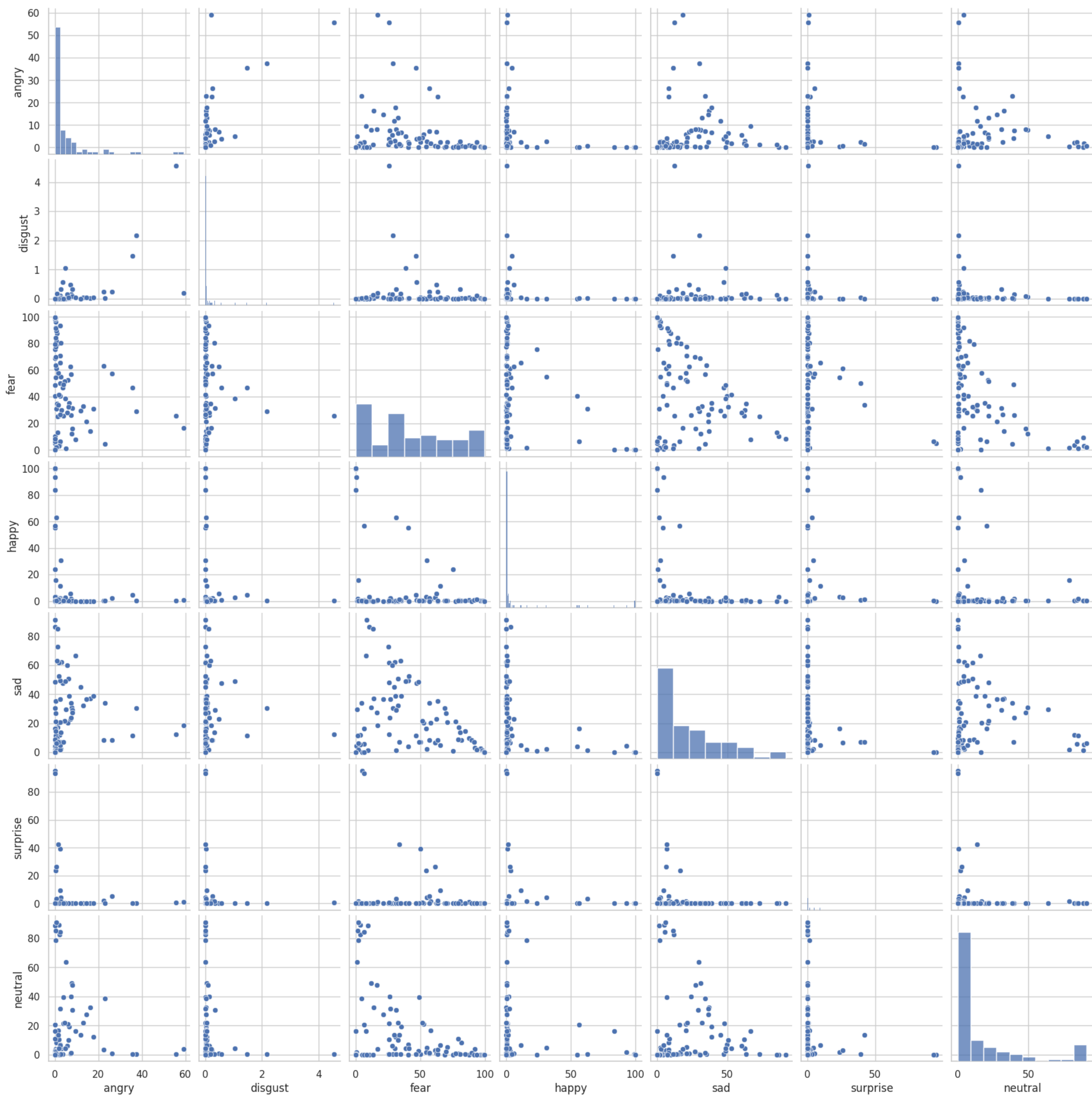


1. Happy and Fear Correlation: The negative correlation between 'Happy' and 'Fear' emotions suggests an inverse relationship between these emotions for

Candidate 7. In practical terms, when 'Happy' scores increase, 'Fear' scores tend to decrease, and vice versa. This indicates that the candidate's expressions of happiness are inversely related to their expressions of fear.

2. Happy and Sad Correlation: Similarly, the negative correlation between 'Happy' and 'Sad' emotions implies that when 'Happy' scores increase, 'Sad' scores tend to decrease, and vice versa. This indicates that the candidate's expressions of happiness are inversely related to their expressions of sadness.

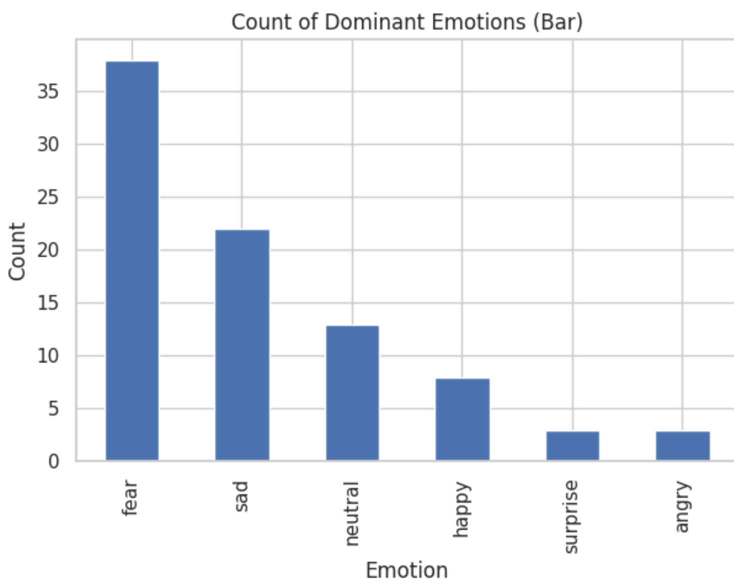
Plotting the scatterplot of each emotion with other emotion using pairplot.

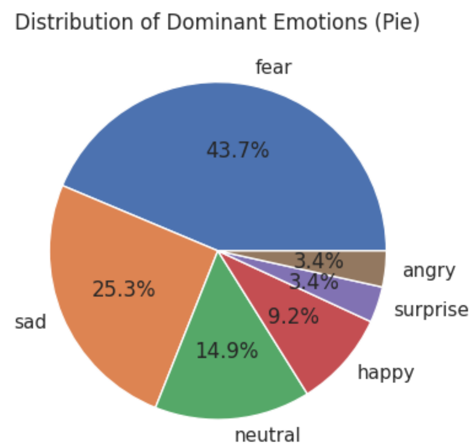


1. Fear vs. Sad: The scatterplot indicates that 'Fear' and 'Sad' values are more scattered, suggesting a broader range of emotional responses and a lack of a strong linear relationship between these two emotions. This means that Candidate 7's expressions of fear and sadness may vary independently of each other, leading to scattered data points.
2. Angry, Surprise, and Disgust: In contrast, 'Angry,' 'Surprise,' and 'Disgust' values appear to be concentrated at relatively low levels. This suggests that Candidate 7's expressions of anger, surprise, and disgust are generally less frequent or intense compared to other emotions.

These observations provide insights into the distribution and relationships among Candidate 7's emotional expressions. The scattered distribution of 'Fear' vs. 'Sad' values indicates a degree of emotional variability, while the concentration of 'Angry,' 'Surprise,' and 'Disgust' values at low levels suggests a lower frequency or intensity of these emotions during the video presentation.

Plotting a bar chart and pie chart showing the occurences of dominant emotions.



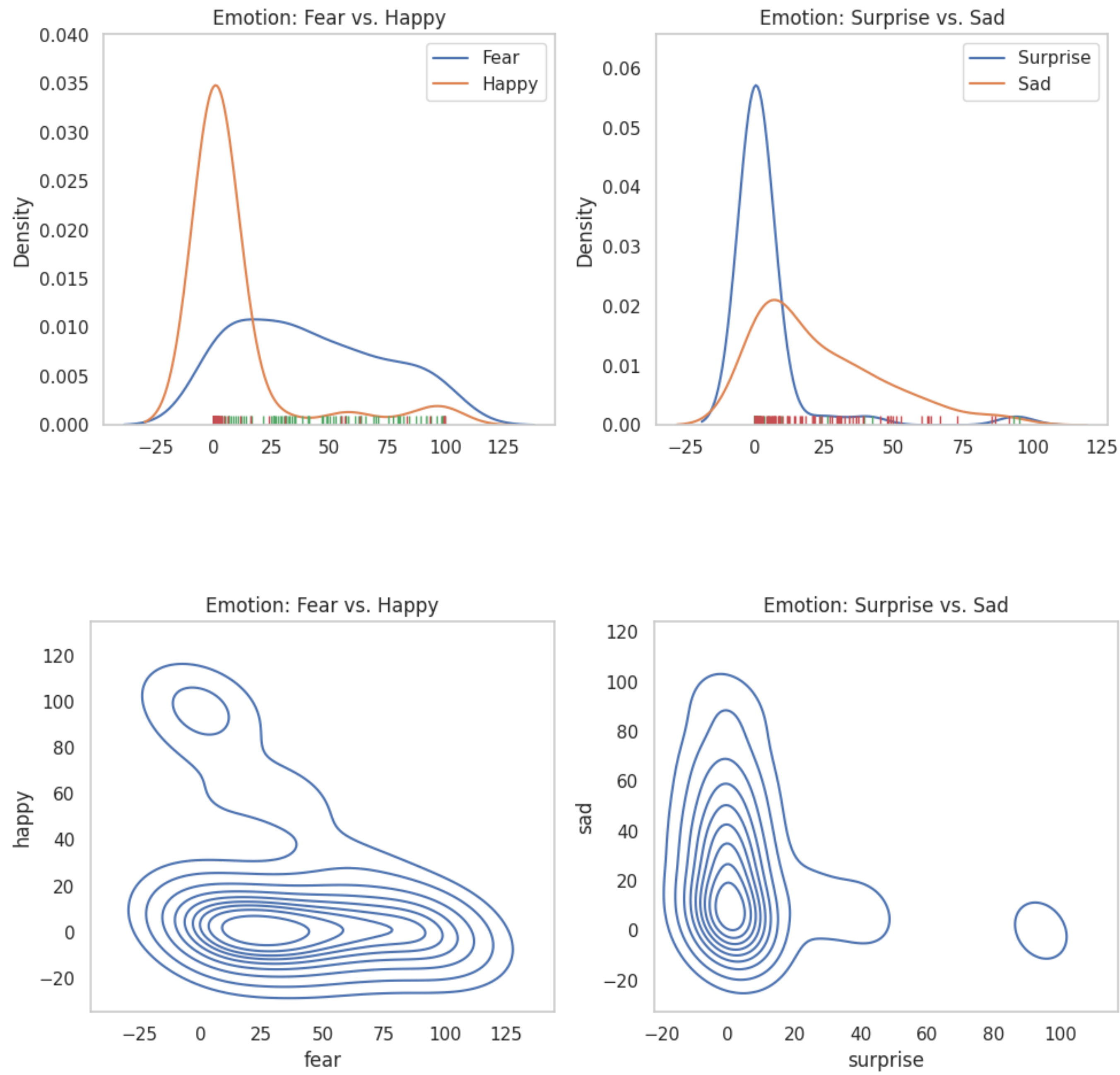


- Bar Chart:
    1. The bar chart reveals that Candidate 7 predominantly expresses 'Fear' emotions. This indicates that fear is the most prominent emotion exhibited by the candidate during the video presentation.
    2. Additionally, 'Sad' emotions are also notable in the candidate's emotional expressions, although they are secondary to 'Fear.'

- Pie Chart:

    1. The pie chart shows that negative emotions, including 'Fear,' 'Sad,' and 'Angry,' collectively make up nearly 70% of Candidate 7's emotional expressions.

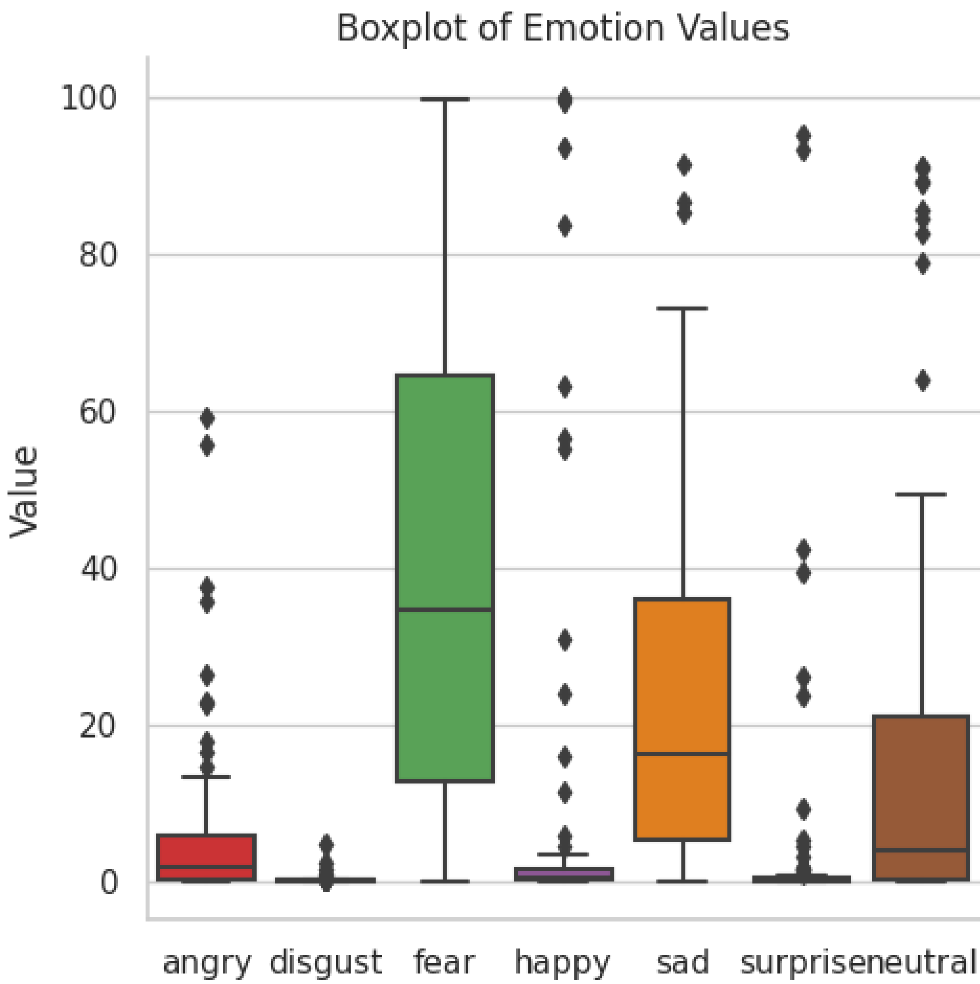
Now plotting the KDE plot between the 2 negative and 2 positive emotions to analyze the density.

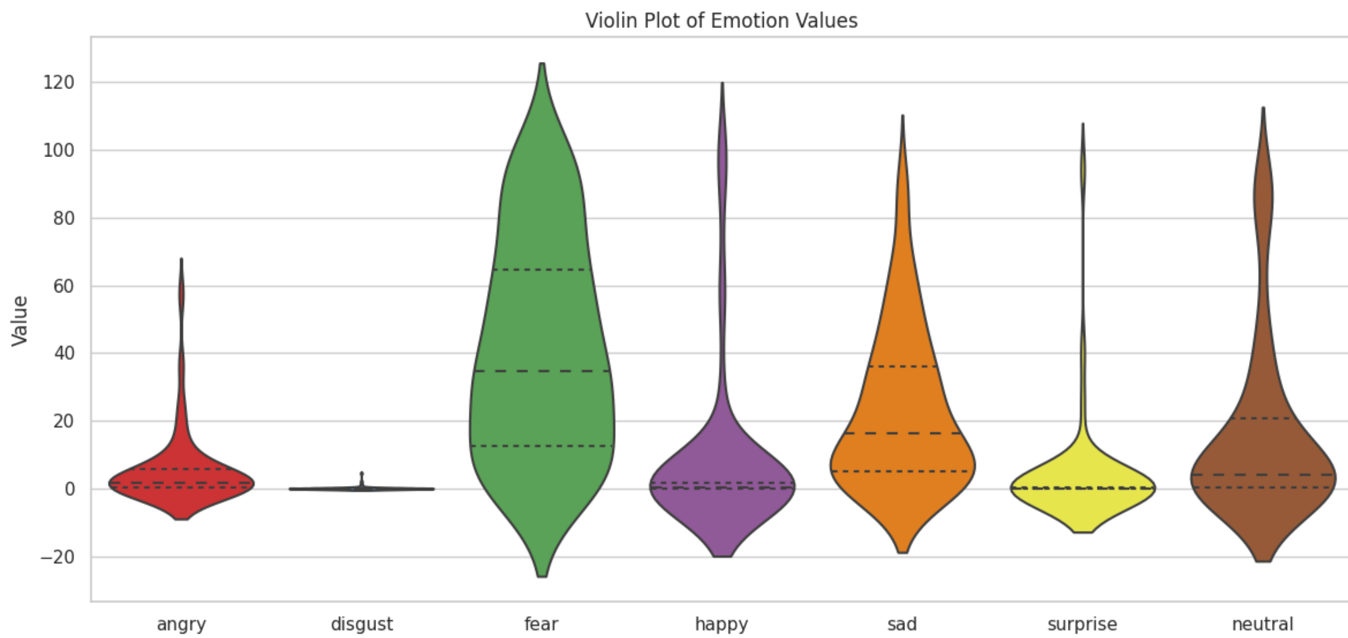


Fear Values: The KDE plot indicates that 'Fear' values have a relatively uniform distribution across the entire range of values, suggesting that Candidate 7's expressions of fear are spread out and exhibit no pronounced peaks or troughs.

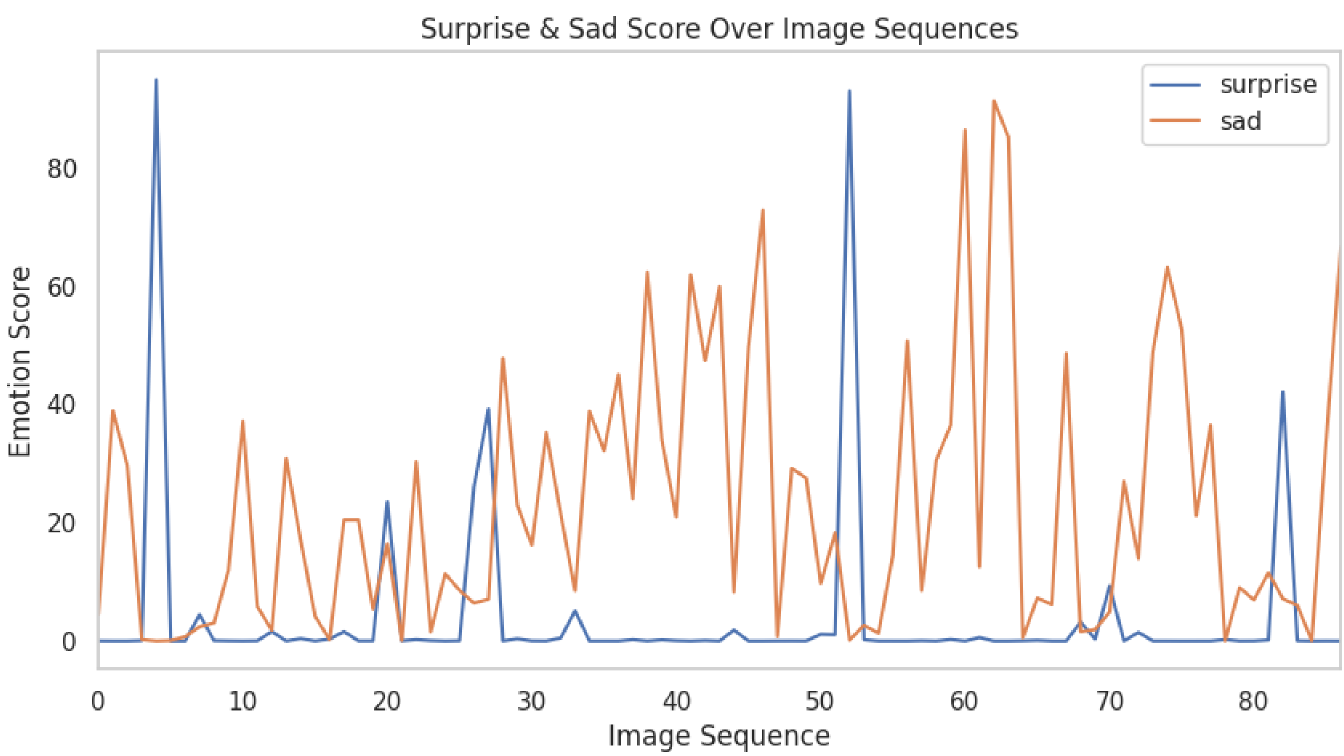
Happy Values: In contrast, 'Happy' values are generally concentrated at smaller values, predominantly below 20. This suggests that the candidate's expressions of happiness tend to be of lower intensity and less frequent compared to other emotions.

Plotting the box plot and violin plot to find the outliers in the data.





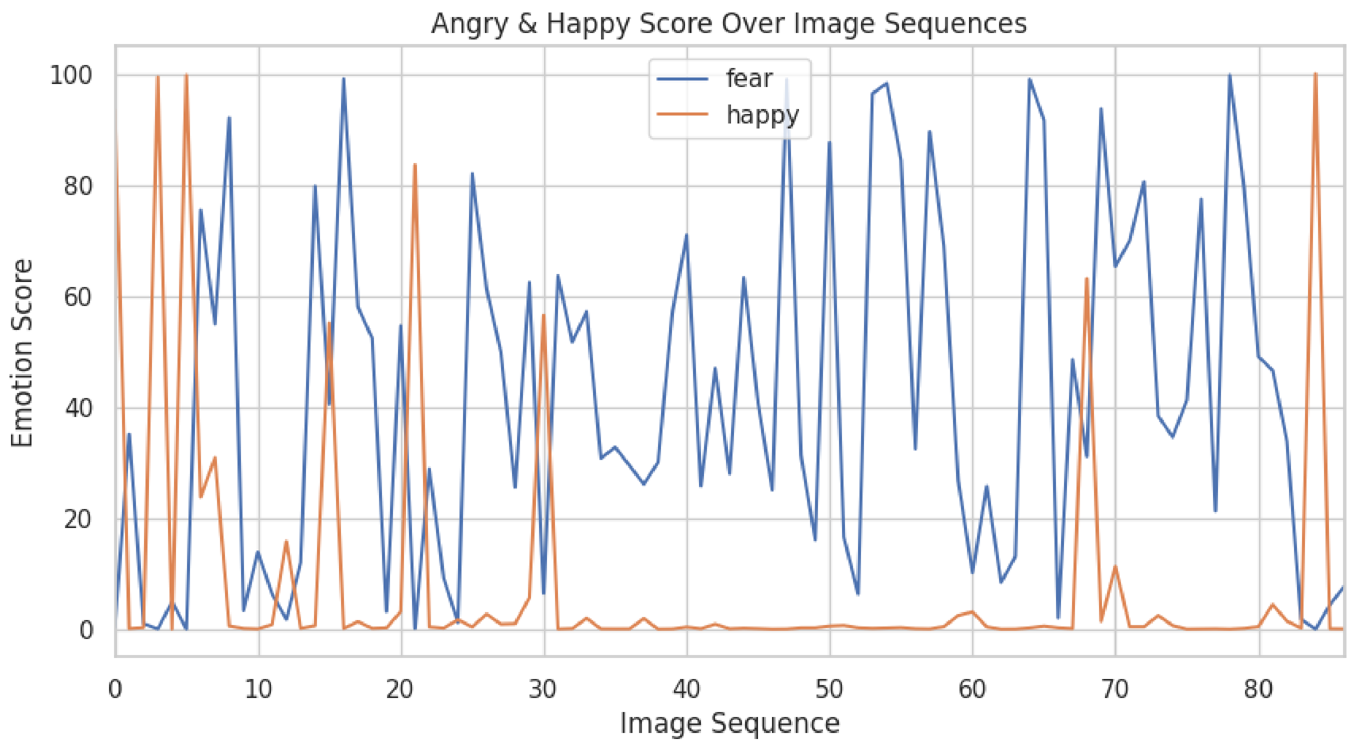
Plotting line chart of emotion scores over image sequences.



- Sad Values: Throughout the video, the 'Sad' values for Candidate 7 are generally higher than 20, indicating that the candidate frequently expresses a relatively high degree of sadness during the presentation.

- Surprise Scores: The line chart shows that 'Surprise' scores exhibit a pattern of peaking up after every quartile or segment of the video. This suggests that Candidate 7 experiences repeated and noticeable spikes in surprise emotions at specific intervals during the presentation.

These observations provide insights into the dynamics of Candidate 7's emotional expressions. The consistently elevated 'Sad' values indicate a frequent experience of sadness, while the recurring spikes in 'Surprise' scores suggest moments of heightened surprise reactions at distinct points in the video.



- Happiness: The candidate initially exhibits happiness at the beginning of the video, but these 'Happy' scores tend to decrease over time, eventually approaching zero. This suggests that the candidate starts the presentation on a positive note but gradually transitions to a less happy emotional state.

- Anger: Throughout the video, 'Angry' scores for Candidate 7 exhibit significant variability, indicating that the candidate's expressions of anger fluctuate considerably during the presentation. This suggests a dynamic range of emotional responses related to anger throughout the video.

Candidate 7's emotional profile during the video is characterized by an initial display of happiness that diminishes over time, coupled with a noteworthy variation in expressions of anger.

Importing the data.

```
gaze = pd.read_csv(r".\emotion_data\7\gaze.csv")
gaze.head()
```

	movie_id	image_seq	gaze	blink	eye_offset
0	6539370c-256e-4ed2-9d00-1be1f051163f	1	1	1	11.9221
1	6539370c-256e-4ed2-9d00-1be1f051163f	2	1	0	13.8024
2	6539370c-256e-4ed2-9d00-1be1f051163f	3	1	0	3.0297
3	6539370c-256e-4ed2-9d00-1be1f051163f	4	1	0	-1.7059
4	6539370c-256e-4ed2-9d00-1be1f051163f	5	0	0	-35.0095

Plotting line chart of eye_offset over image sequences.

- For the majority of the video, the candidate maintains a consistent gaze direction with an angle of around 0 degrees. This suggests that the candidate's attention and focus are generally aligned with the center or a specific point of interest in the video content.
- However, there are notable instances in the middle of the video where the eye offset deviates significantly, exceeding 60 degrees. These deviations indicate moments when the candidate's gaze direction abruptly shifts away from the center or primary focus point.

These observations suggest that Candidate 7's attention and visual engagement are predominantly stable, but there are specific segments in the middle of the video where their gaze direction shifts noticeably.

2.8 Candidate 8

Importing emotion file

```python
emotion=pd.read_csv(r".\emotion_data\8\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	813af424-a584-4417-b7ee-0d4c705e83c9	0	0.334470	1.199280e-06	13.786700	0.302347	0.539346	10.447900	74.5893	neutral
1	813af424-a584-4417-b7ee-0d4c705e83c9	1	0.840043	9.234260e-05	2.242650	0.560995	0.417769	0.022157	95.9163	neutral
2	813af424-a584-4417-b7ee-0d4c705e83c9	2	2.290200	1.809870e-04	5.731700	0.132673	3.990380	0.011224	87.8436	neutral
3	813af424-a584-4417-b7ee-0d4c705e83c9	3	0.417658	7.776570e-07	0.063667	0.017022	0.107120	0.000130	99.3944	neutral
4	813af424-a584-4417-b7ee-0d4c705e83c9	4	0.219031	5.969380e-06	2.211050	0.103567	0.352093	0.035190	97.0791	neutral

Finding the data type, missing and unique values of each column using the Autoviz library.

	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
movie_id	object	0.000000	1	nan	nan	Zero-variance colum: drop before modeling process.
image_seq	int64	0.000000	100	0.000000	99.000000	Possible ID colum: drop before modeling process.
angry	float64	0.000000	NA	0.000017	42.171200	has 13 outliers greater than upper bound (2.68) or lower than lower bound(-1.53). Cap them or remove them.
disgust	float64	0.000000	NA	0.000000	0.054649	has 18 outliers greater than upper bound (0.00) or lower than lower bound(-0.00). Cap them or remove them.
fear	float64	0.000000	NA	0.003125	95.139500	has 2 outliers greater than upper bound (87.14) or lower than lower bound(-48.12). Cap them or remove them.
happy	float64	0.000000	NA	0.002380	95.781900	has 5 outliers greater than upper bound (71.50) or lower than lower bound(-37.42). Cap them or remove them.
sad	float64	0.000000	NA	0.004126	98.309400	has 16 outliers greater than upper bound (22.65) or lower than lower bound(-12.79). Cap them or remove them.
surprise	float64	0.000000	NA	0.000029	91.265300	has 17 outliers greater than upper bound (8.73) or lower than lower bound(-4.99). Cap them or remove them.
neutral	float64	0.000000	NA	0.021739	99.918800	No issue
dominant_emotion	object	0.000000	6	nan	nan	No issue

There are no missing values in the dataset. Only 2 columns, movie_id and dominant_emotion is of object data type each having 1 and 6 unique values As movie_id is same for a particular candidate, we can remove that column.

Calculating basic statistics for each emotion, such as mean, median, standard deviation, and range, to get an overview of the candidates' emotional expressions.

shape = (93, 10)

	image_seq	angry	disgust	fear	happy	sad	surprise	neutral
count	93.000000	93.000000	9.300000e+01	93.000000	93.000000	93.000000	93.000000	93.000000
mean	46.000000	8.115339	1.119884e-02	11.939264	2.120367	1.955231	1.901795	73.956810
std	26.990739	13.794058	4.575753e-02	18.479163	8.388213	3.985409	6.809604	32.030199
min	0.000000	0.025846	8.551100e-11	0.004501	0.002123	0.007992	0.000016	0.791256
25%	23.000000	0.334470	2.932250e-07	0.296672	0.052007	0.205621	0.003340	43.812200
50%	46.000000	1.440180	3.979970e-05	1.958190	0.302347	0.511124	0.040144	92.759500
75%	69.000000	8.874540	1.594090e-03	16.662700	0.776652	1.603200	0.550705	98.513200
max	92.000000	59.193800	3.162430e-01	87.632500	71.745700	28.907800	58.758100	99.888500

Plotting the Correlation matrix heatmap using seaborn library.

Angry and Fear Correlation: There is a positive correlation of 0.41 between 'Angry' and 'Fear' emotions. This positive correlation suggests that when 'Angry' scores increase, 'Fear' scores tend to increase as well. This indicates a degree of co-occurrence or similarity in the expression of anger and fear for Candidate 7.

Disgust and Sad Correlation: Similarly, there is a positive correlation of 0.44 between 'Disgust' and 'Sad' emotions. This positive correlation implies that when

'Disgust' scores increase, 'Sad' scores tend to increase as well. This suggests that the candidate's expressions of disgust and sadness are positively associated.

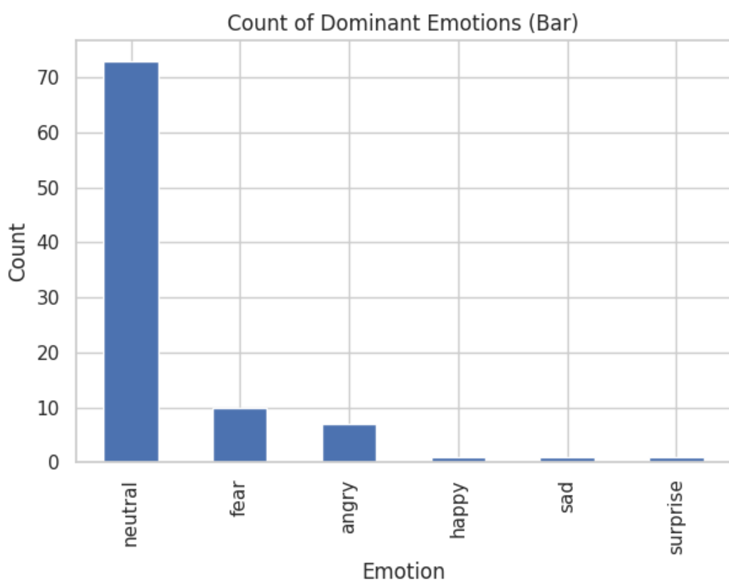
Plotting the scatterplot of each emotion with other emotion using pairplot.

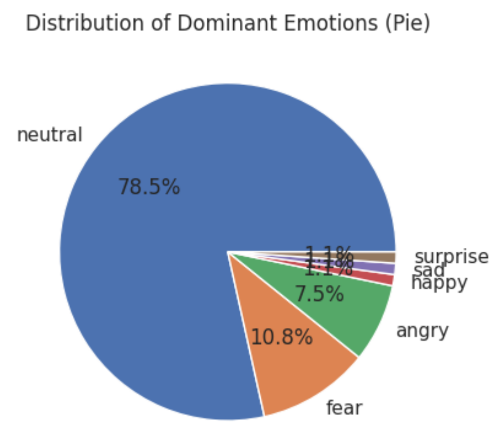
- Anger Values: The scatterplot indicates that 'Anger' values are high and exhibit a broad range of variation. This suggests that Candidate 8 expresses a wide spectrum of anger during the video presentation, and these expressions are not limited to a narrow range.

- Low Scatter for Happy, Sad, and Disgust: In contrast, 'Happy,' 'Sad,' and 'Disgust' values do not scatter much, indicating that these emotions are relatively stable and consistent. This suggests that Candidate 8's expressions of happiness, sadness, and disgust remain within a limited range and do not vary significantly throughout the video.

These observations highlight that Candidate 8 displays a prominent and variable range of anger expressions, while emotions such as happiness, sadness, and disgust are comparatively more stable and less variable during the video presentation.

Plotting a bar chart and pie chart showing the occurences of dominant emotions.



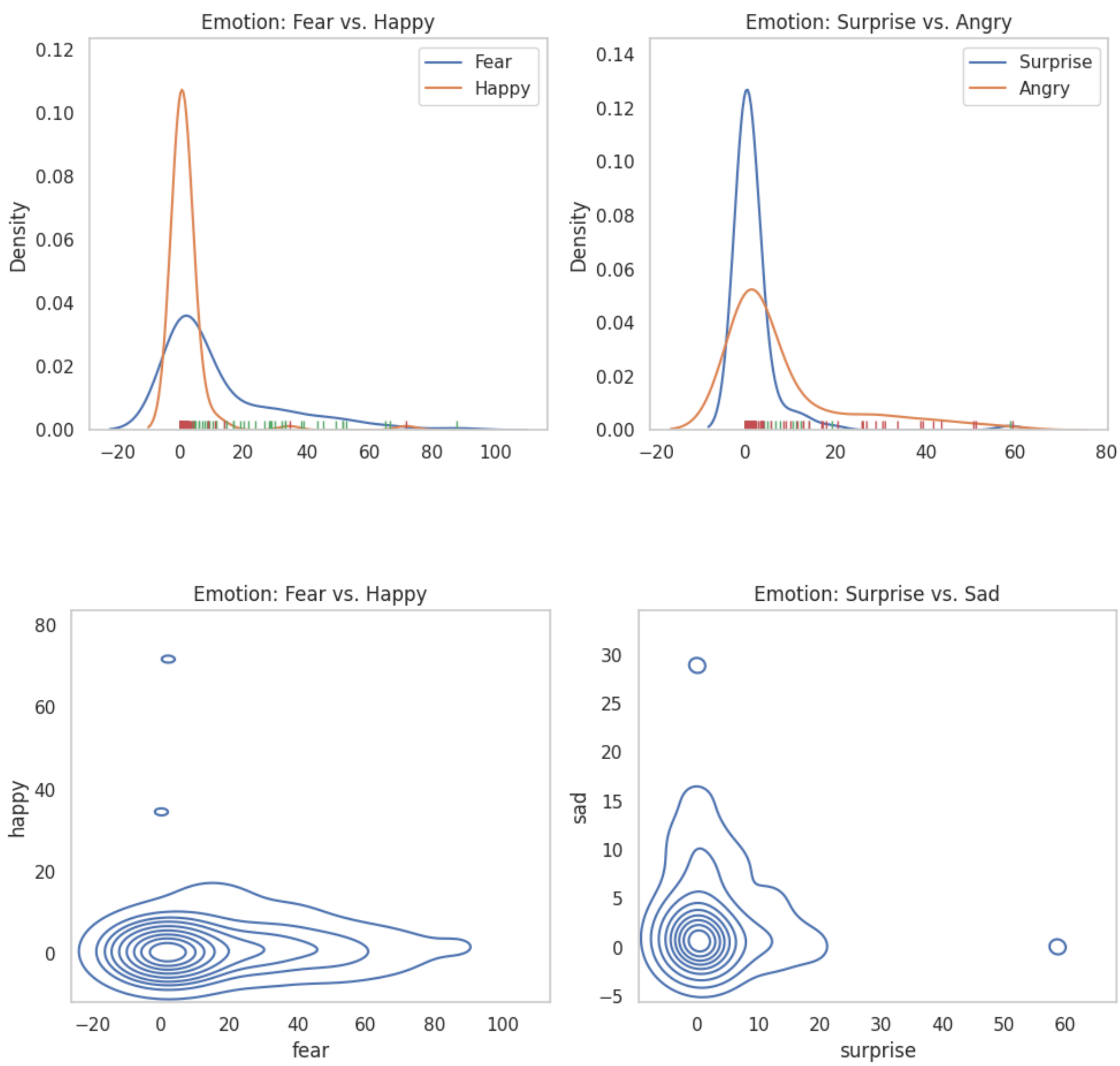


1. Bar Chart:
   - The bar chart reveals that Candidate 1 predominantly maintains a 'Neutral' emotional state, constituting approximately 80% of their overall emotional expressions. This indicates that the candidate maintains a composed and unemotional demeanor for the majority of the video presentation.
   - Additionally, there is a notable presence of negative emotions, including 'Fear' and 'Angry,' which together make up around 18% of Candidate 1's emotional expressions.
2. Pie Chart:

- The pie chart provides a visual representation of the distribution of emotions and underscores that the candidate primarily exhibits 'Neutral' emotions, followed by negative emotions (fear and anger), with 'Happy' and 'Surprise' being relatively less frequent.

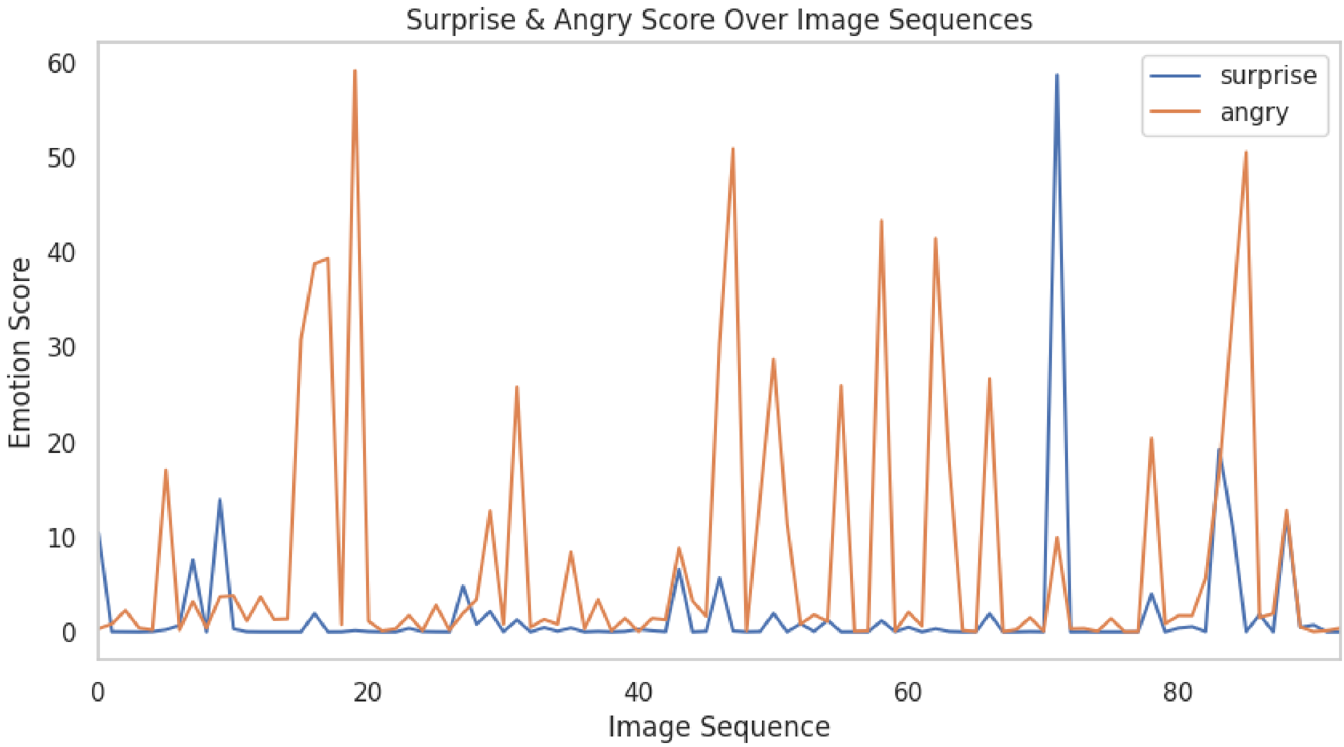
In summary, Candidate 8's emotional profile is characterized by a dominant display of neutrality, with occasional expressions of fear and anger. The presence of 'Happy' and 'Surprise' emotions is notably low in comparison.

Now plotting the KDE plot between the 2 negative and 2 positive emotions to analyze the density.



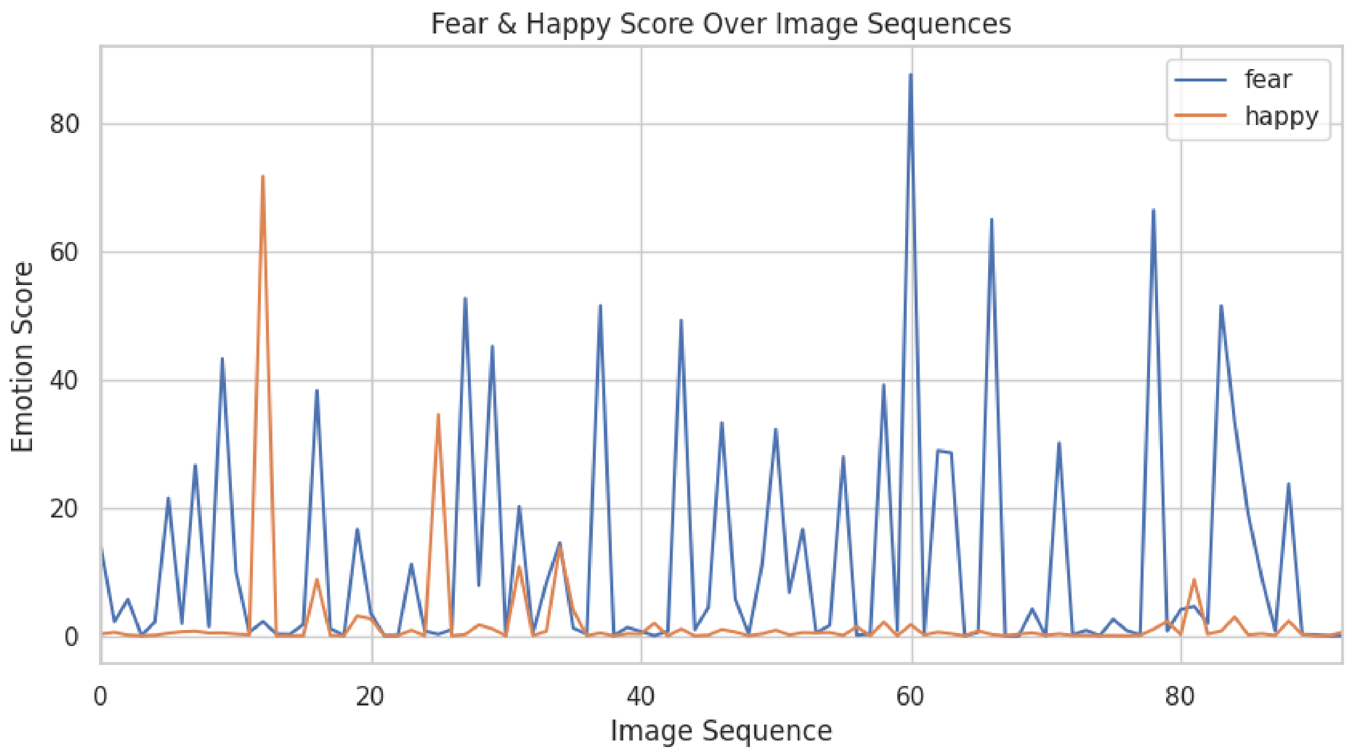
As the happy and surprise values were generally very low throughout the video, we can not infer much from KDE plot.

Plotting line chart of emotion scores over image sequences.



- Angry Scores: Throughout the video presentation, 'Angry' scores consistently maintain a significant presence, indicating that Candidate 8 frequently expresses anger during various segments of the video.
- Surprise Scores: In contrast, 'Surprise' scores remain relatively lower and do not exhibit significant peaks or variations. This suggests that the candidate's expressions of surprise are less frequent and intense compared to anger.

The dominance of 'Angry' scores over 'Surprise' scores in the line chart suggests that Candidate 8's emotional state is characterized by a notable presence of anger, while expressions of surprise are relatively subdued



- Fear Scores: Throughout the video, 'Fear' scores display a noticeable presence and fluctuate between 0 and 40. This indicates that Candidate 8 frequently expresses fear during various segments of the video, and these expressions vary in intensity.
- Happy Scores: In contrast, 'Happy' scores generally remain close to 0 and do not exhibit significant variations. This suggests that the candidate's expressions of happiness are infrequent and typically occur at very low intensity levels.

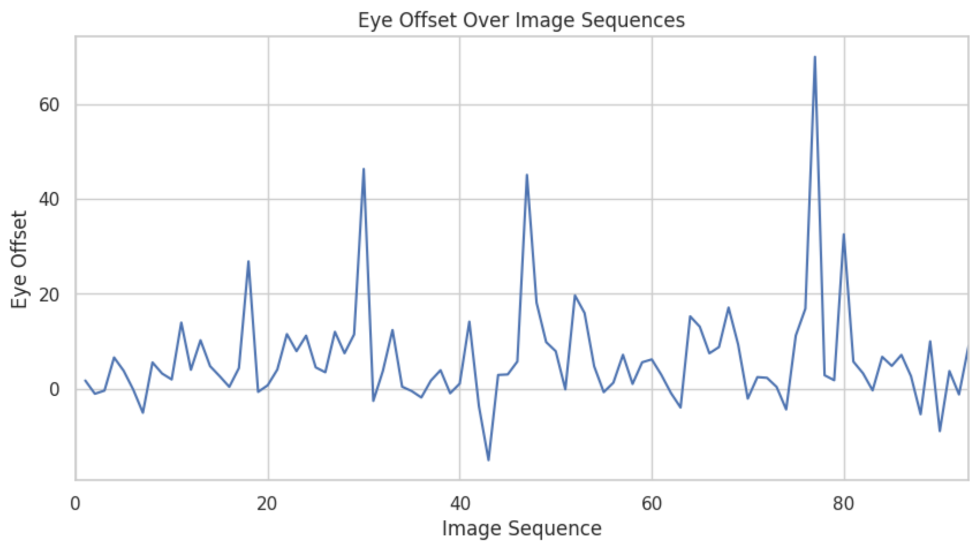
The dominance of 'Fear' scores over 'Happy' scores in the line chart underscores that Candidate 8's emotional state is characterized by a consistent presence of fear, while expressions of happiness are relatively rare and mild.

Importing the data.

```
gaze = pd.read_csv(r".\emotion_data\8\gaze.csv")
gaze.head()
```

	movie_id	image_seq	gaze	blink	eye_offset
0	813af424-a584-4417-b7ee-0d4c705e83c9	1	1	0	1.7394
1	813af424-a584-4417-b7ee-0d4c705e83c9	2	1	0	-1.0581
2	813af424-a584-4417-b7ee-0d4c705e83c9	3	1	0	-0.3826
3	813af424-a584-4417-b7ee-0d4c705e83c9	4	1	0	6.6005
4	813af424-a584-4417-b7ee-0d4c705e83c9	5	1	0	3.7838

Plotting line chart of eye_offset over image sequences.



The eye offset seems to be stable and in the beginning of the video but in the end, we can see some sharp peaks

2.9 Candidate 9

Importing emotion file

```
emotion=pd.read_csv(r".\emotion_data\9\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	dfb0d746-609f-4dac-8e1d-c0325fb64394	0	0.986249	1.054770e-04	3.882840	0.139500	5.511320	0.108728	89.371300	neutral
1	dfb0d746-609f-4dac-8e1d-c0325fb64394	1	2.081140	2.935960e-01	7.371850	76.190600	1.524340	7.602480	4.936020	happy
2	dfb0d746-609f-4dac-8e1d-c0325fb64394	2	9.438360	4.605680e-01	9.067420	71.409600	3.412750	5.922060	0.289199	happy
3	dfb0d746-609f-4dac-8e1d-c0325fb64394	3	4.378860	2.232070e-02	24.838800	14.193200	1.357880	31.498200	23.710700	surprise
4	dfb0d746-609f-4dac-8e1d-c0325fb64394	4	0.001452	1.465590e-07	0.936351	0.365306	0.004872	95.902500	2.789520	surprise

Finding the data type, missing and unique values of each column using the Autoviz library.

	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
movie_id	object	0.000000	1	nan	nan	Zero-variance colum: drop before modeling process.
image_seq	int64	0.000000	100	0.000000	99.000000	Possible ID colum: drop before modeling process.
angry	float64	0.000000	NA	0.000017	42.171200	has 13 outliers greater than upper bound (2.68) or lower than lower bound(-1.53). Cap them or remove them.
disgust	float64	0.000000	NA	0.000000	0.054649	has 18 outliers greater than upper bound (0.00) or lower than lower bound(-0.00). Cap them or remove them.
fear	float64	0.000000	NA	0.003125	95.139500	has 2 outliers greater than upper bound (87.14) or lower than lower bound(-48.12). Cap them or remove them.
happy	float64	0.000000	NA	0.002380	95.781900	has 5 outliers greater than upper bound (71.50) or lower than lower bound(-37.42). Cap them or remove them.
sad	float64	0.000000	NA	0.004126	98.309400	has 16 outliers greater than upper bound (22.65) or lower than lower bound(-12.79). Cap them or remove them.
surprise	float64	0.000000	NA	0.000029	91.265300	has 17 outliers greater than upper bound (8.73) or lower than lower bound(-4.99). Cap them or remove them.
neutral	float64	0.000000	NA	0.021739	99.918800	No issue
dominant_emotion	object	0.000000	6	nan	nan	No issue

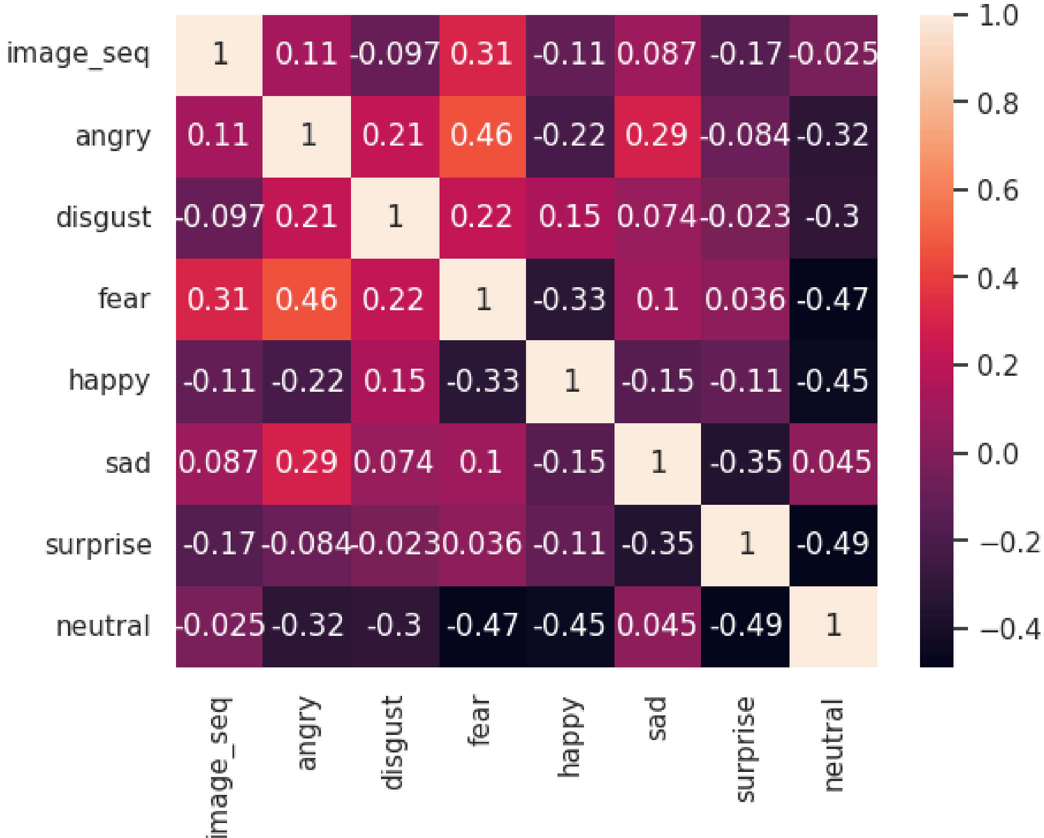
There are no missing values in the dataset. Only 2 columns, movie_id and dominant_emotion is of object data type each having 1 and 6 unique values As movie_id is same for a particular candidate, we can remove that column.

Calculating basic statistics for each emotion, such as mean, median, standard deviation, and range, to get an overview of the candidates' emotional expressions.

shape = (86, 10)

	image_seq	angry	disgust	fear	happy	sad	surprise	neutral
count	86.000000	86.000000	8.600000e+01	86.000000	86.000000	86.000000	86.000000	86.000000
mean	42.500000	6.337654	1.396405e-01	18.602680	16.734190	4.293063	15.761101	38.131673
std	24.969982	7.712496	3.905607e-01	20.513298	27.389401	5.937296	22.198794	35.152782
min	0.000000	0.001452	1.465590e-07	0.013609	0.007067	0.004872	0.001558	0.186039
25%	21.250000	0.850265	3.740085e-04	2.960657	0.325851	0.601723	0.590199	7.115168
50%	42.500000	3.243240	8.396090e-03	9.066630	3.332290	1.893290	4.469680	28.956600
75%	63.750000	9.500697	9.576575e-02	27.268000	14.564825	5.768842	22.935025	75.421375
max	85.000000	46.511700	2.944570e+00	76.316400	98.331100	28.771000	95.902500	99.552500

Plotting the Correlation matrix heatmap using seaborn library.

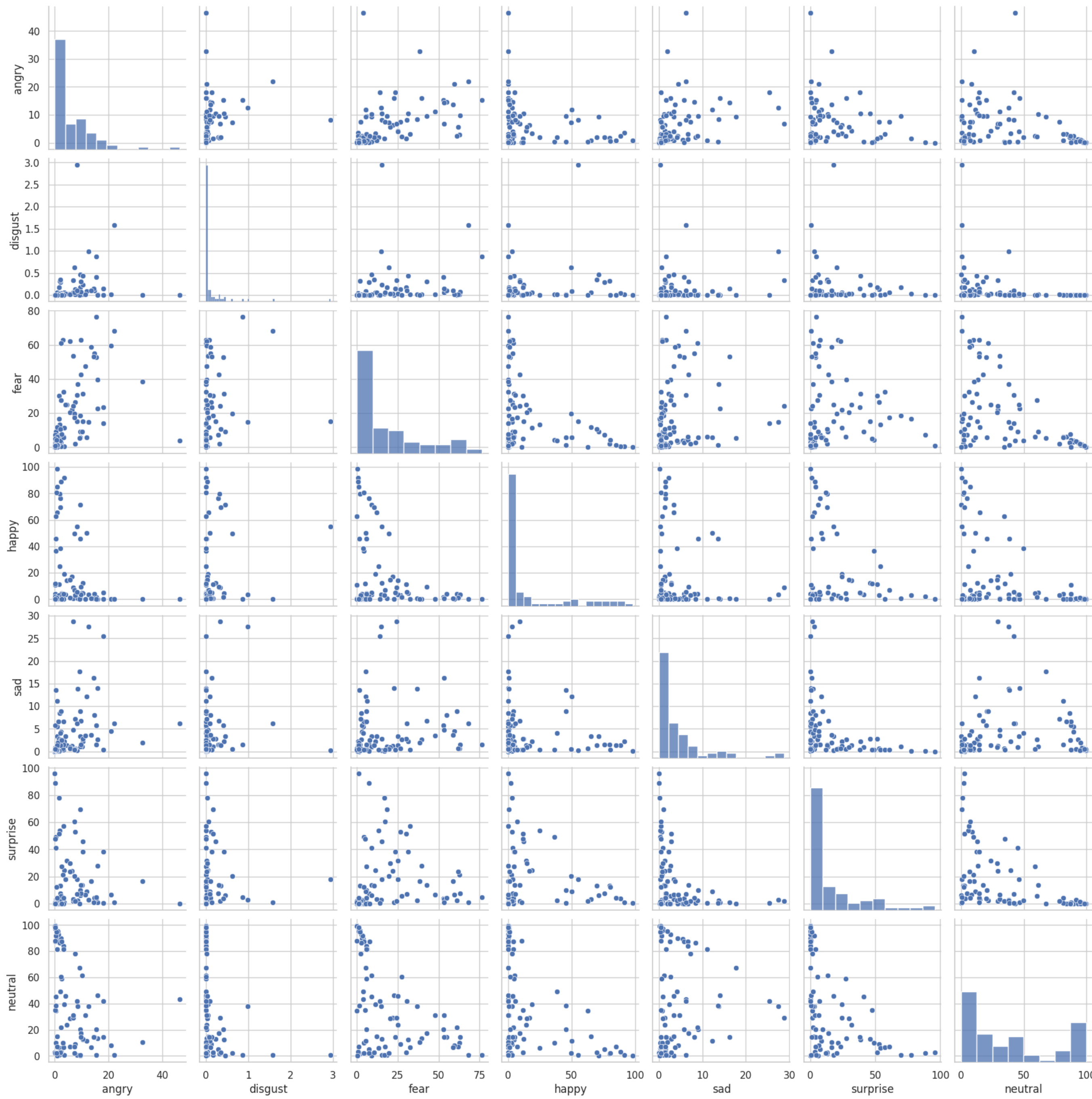


- Fear and Matrix Correlation: There is a positive correlation of 0.46 between 'Fear' and 'Matrix' emotions. This positive correlation suggests that when 'Fear' scores increase, 'Matrix' scores tend to increase as well. This indicates a degree of co-occurrence or similarity in the expression of fear and the 'Matrix' emotion for Candidate 9.

- Happy and Fear Correlation: There is a negative correlation of -0.36 between 'Happy' and 'Fear' emotions. This negative correlation implies that when 'Happy' scores increase, 'Fear' scores tend to decrease, and vice versa. This indicates an inverse relationship between the expressions of happiness and fear for the candidate.

These correlations provide insights into the relationships between specific pairs of emotions and the 'Matrix' emotion for Candidate 9. The positive correlation between 'Fear' and 'Matrix' suggests a positive association between these emotions, while the negative correlation between 'Happy' and 'Fear' indicates an inverse relationship between happiness and fear.

Plotting the scatterplot of each emotion with other emotion using pairplot.



The scatterplot analysis for Candidate 9's emotional expressions provides the following observations:

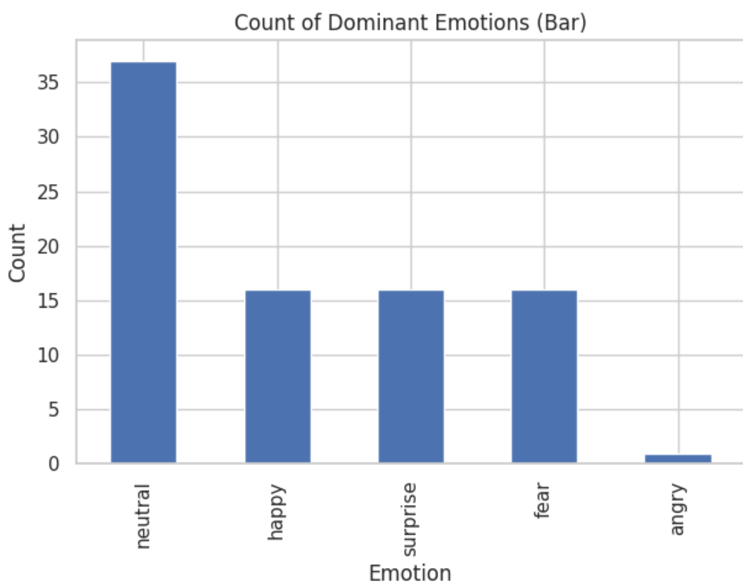
1. Happy and Surprise: There is a positive relationship between 'Happy' and 'Surprise' values, with positive values of happiness being associated with high scores of surprise. This suggests that when Candidate 9 expresses happiness, there is a tendency for simultaneous expressions of surprise at higher intensities. The positive correlation between these two emotions indicates their co-occurrence.

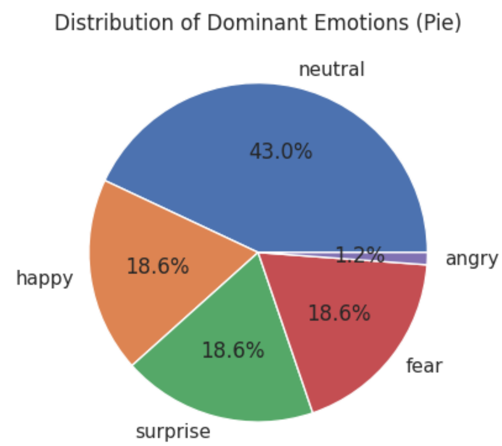
2. Disgust and Angry: 'Disgust' and 'Angry' values tend to possess lower scores and are clustered together in the scatterplot. This indicates that when Candidate 9 expresses either disgust or anger, the scores for both emotions remain low or relatively consistent. There is a lack of strong variability or co-occurrence between these emotions.

3. Surprise and Fear: The scatterplot between 'Surprise' and 'Fear' values is more scattered, suggesting a broader range of emotional responses. This indicates that Candidate 9's expressions of surprise and fear may vary independently of each other, leading to a scattered distribution of data points.

These observations provide insights into the relationships and variability among different emotional expressions for Candidate 9. While happiness and surprise tend to co-occur, disgust and anger exhibit lower scores and less variability, and surprise and fear show a more scattered relationship.

Plotting a bar chart and pie chart showing the occurences of dominant emotions.





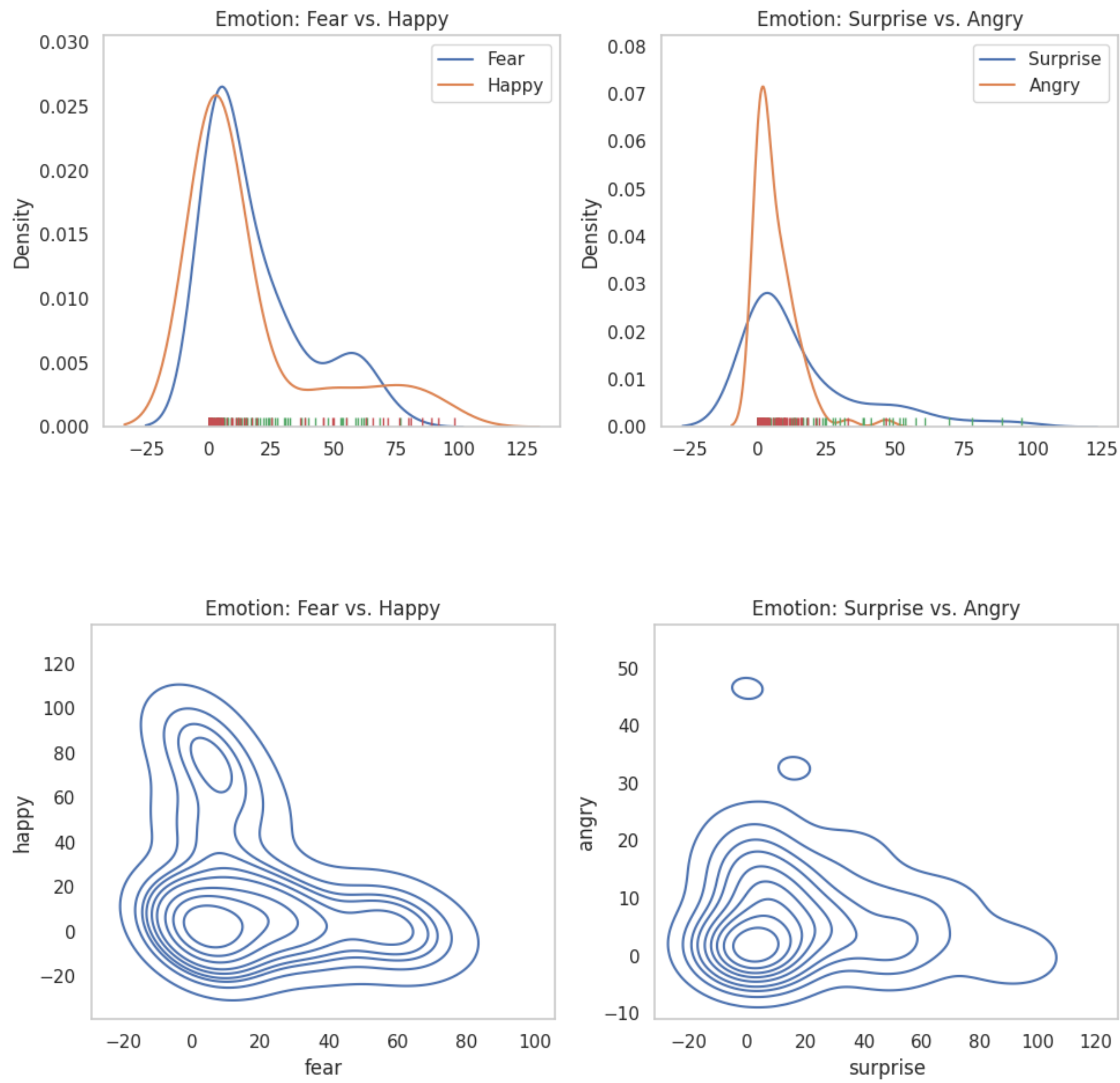
- Bar Chart:
    1. The bar chart reveals that Candidate 9 predominantly maintains a 'Neutral' emotional state, constituting approximately 43% of their overall emotional expressions. This indicates that the candidate maintains a composed and unemotional demeanor for a significant portion of the video presentation.
    2. Additionally, the candidate exhibits positive emotions, such as 'Happy' and 'Surprise,' which collectively make up around 40% of their emotional

expressions. This suggests that the candidate experiences moments of happiness and surprise during the video.

- Pie Chart:

  1. The pie chart provides a visual representation of the distribution of emotions and underscores that 'Neutral' emotions dominate, followed by positive emotions (happy and surprise).

Now plotting the KDE plot between the 2 negative and 2 positive emotions to analyze the density.



1. The KDE plot indicates that 'Happy' scores tend to have a higher density over 'Fear' scores for almost all values. This suggests that Candidate 9's expressions of happiness are more frequent and concentrated across a broader range of values compared to their expressions of fear.

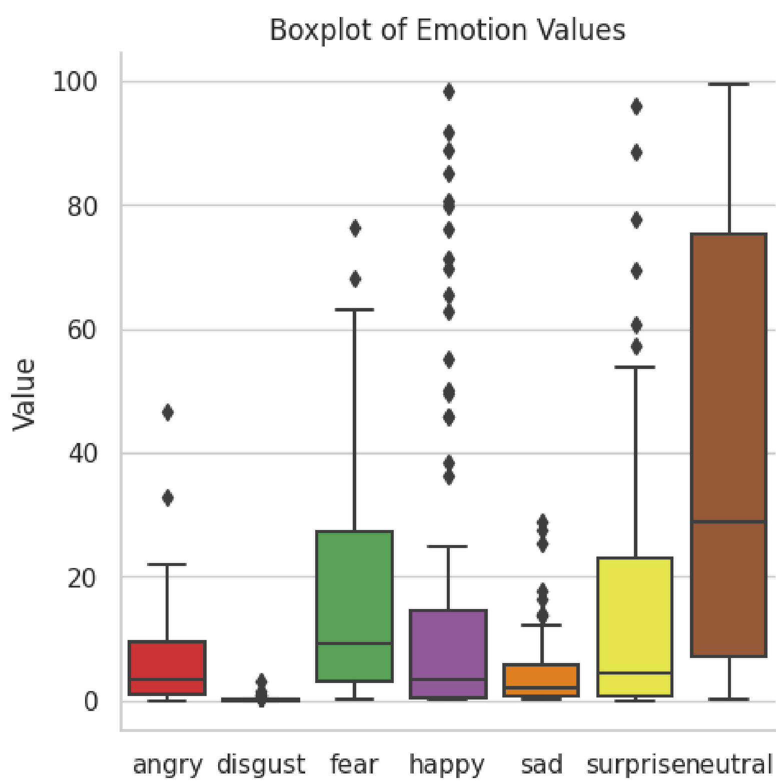
2. The higher density of 'Happy' scores over 'Fear' in the KDE plot implies that happiness is a more prevalent and frequently expressed emotion for Candidate 9 throughout the video presentation.

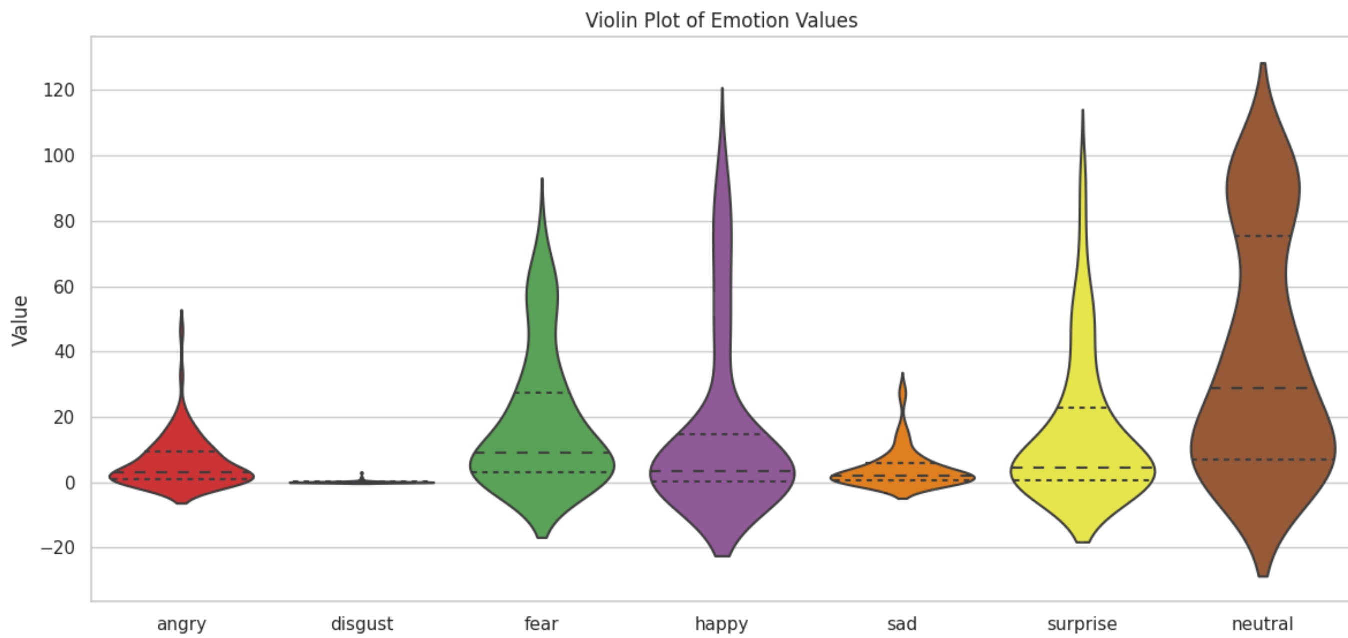
3. Angry Scores: The KDE plot indicates that 'Angry' scores are generally less than 25, suggesting that Candidate 9's expressions of anger tend to be limited in intensity and rarely exceed this threshold. This implies a relatively subdued expression of anger throughout the video.

4. Surprise Scores: In contrast, 'Surprise' scores exhibit a broader range and greater variability, spanning from 0 to 90. This indicates that Candidate 9's expressions of surprise vary significantly in intensity, with moments of both low and high surprise scattered throughout the video.

5. These observations highlight that Candidate 9's emotional expressions are characterized by limited intensity in anger and a wider range of variability in surprise

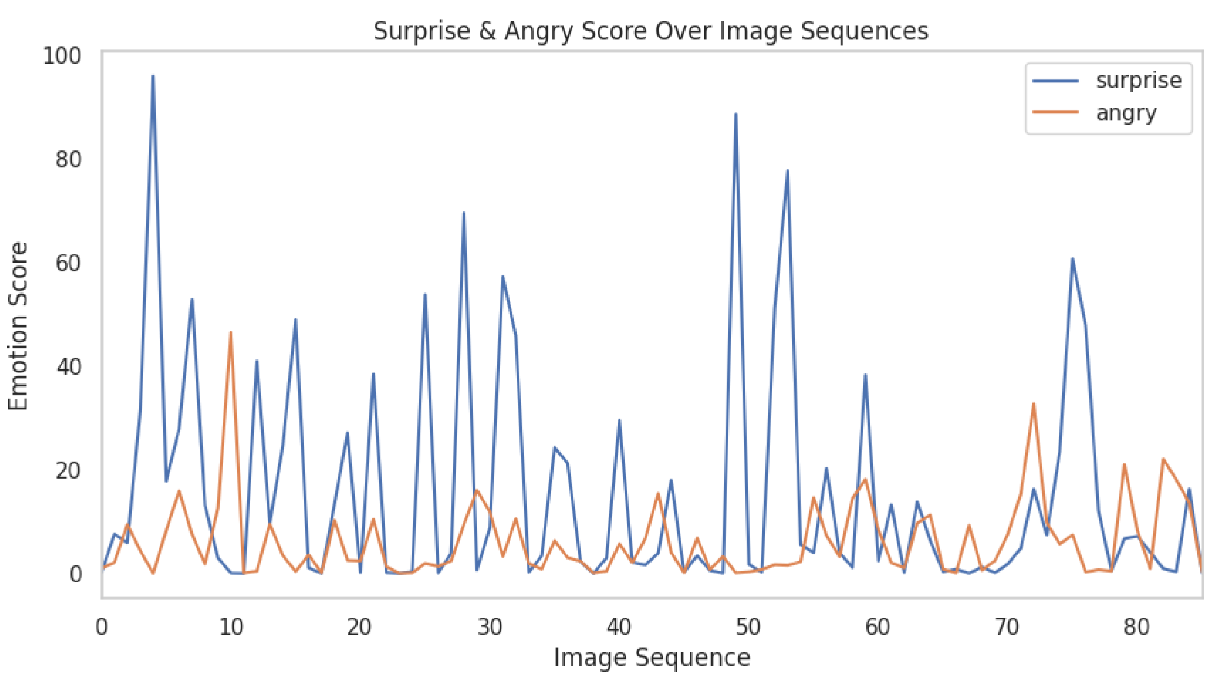
Plotting the box plot and violin plot to find the outliers in the data.





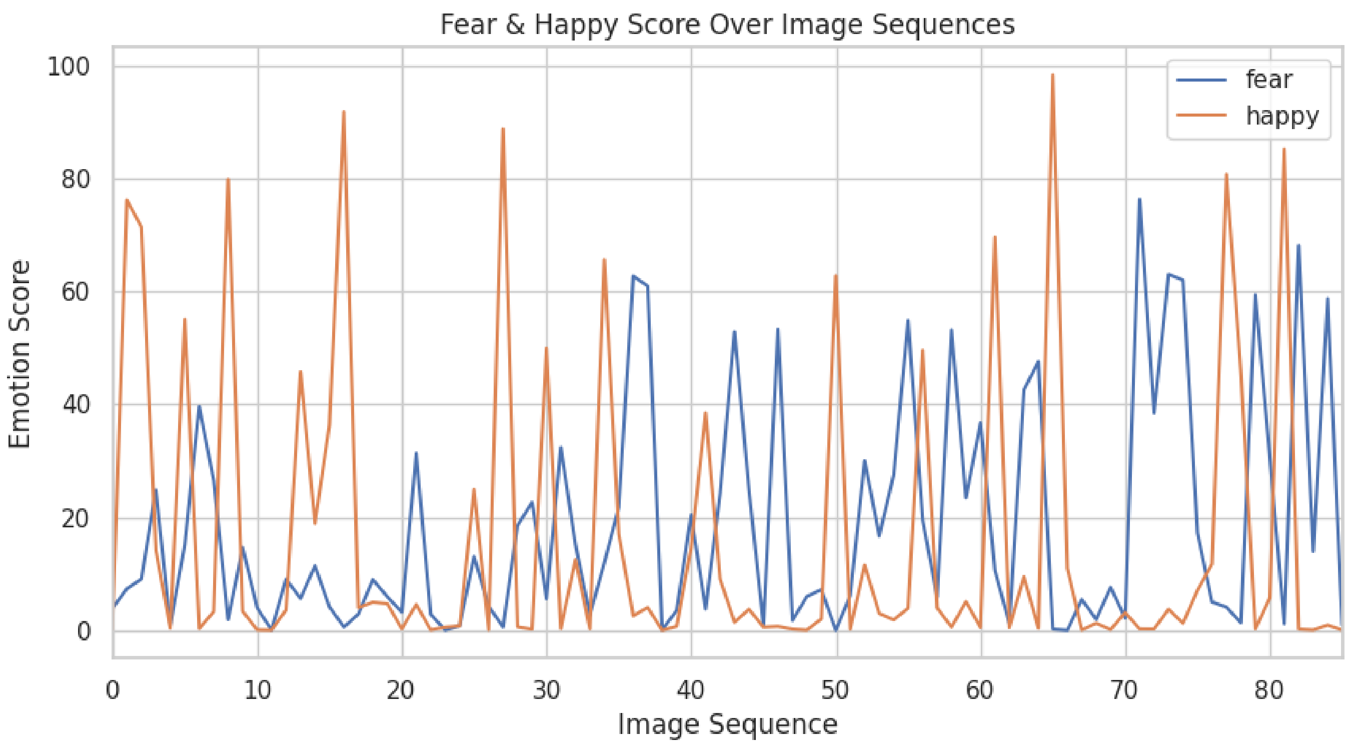
Happy Outliers: The box plot reveals that there are numerous data points in the 'Happy' emotion category that fall significantly above or below the whiskers of the box plot. These data points are considered outliers because they deviate substantially from the central tendency or median of the 'Happy' scores.

Plotting line chart of emotion scores over image sequences.



- Surprise Scores: Throughout the video presentation, 'Surprise' scores consistently maintain a noticeable presence, suggesting that Candidate 9 frequently expresses surprise during various segments of the video. The line chart may show fluctuating levels of surprise.

- Angry Scores: In contrast, 'Angry' scores appear to be less prominent and do not exhibit the same level of consistency or intensity as 'Surprise' scores. This suggests that Candidate 9's expressions of anger are comparatively subdued and occur less frequently during the video.



Initially, at the beginning of the video, 'Happy' scores dominate over 'Fear' scores. This suggests that Candidate 9 starts the presentation in a relatively happy emotional state, with expressions of happiness being more prominent than fear.

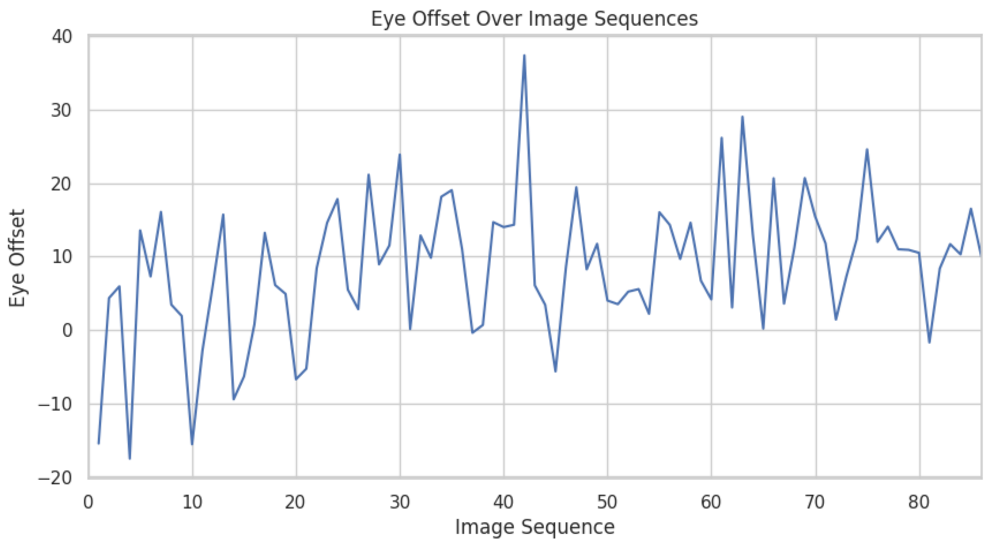
However, as the video progresses, 'Happy' and 'Fear' scores seem to go hand in hand, indicating a shift in emotional dynamics. This suggests that over time, the candidate's expressions of happiness become more balanced with expressions of fear. The two emotions co-occur more frequently as the video continues.

Importing the data.

```
gaze = pd.read_csv(r".\emotion_data\9\gaze.csv")
gaze.head()
```

	movie_id	image_seq	gaze	blink	eye_offset
0	dfb0d746-609f-4dac-8e1d-c0325fb64394	1	1	0	-15.4688
1	dfb0d746-609f-4dac-8e1d-c0325fb64394	2	1	0	4.3447
2	dfb0d746-609f-4dac-8e1d-c0325fb64394	3	1	0	5.9378
3	dfb0d746-609f-4dac-8e1d-c0325fb64394	4	1	0	-17.5304
4	dfb0d746-609f-4dac-8e1d-c0325fb64394	5	1	0	13.5553

Plotting line chart of eye_offset over image sequences.



The eye offset tends to be stable and is generally between 0 - 20°

2.10 Candidate 10

Importing emotion file

```
emotion=pd.read_csv(r".\emotion_data\7\emotion.csv")
emotion.head()
```

	movie_id	image_seq	angry	disgust	fear	happy	sad	surprise	neutral	dominant_emotion
0	83c20b83-7881-499d-a40d-cc06b65869f8	0	0.393006	2.191800e-07	0.622241	0.000231	98.333400	4.278190e-07	0.651150	sad
1	83c20b83-7881-499d-a40d-cc06b65869f8	1	0.099217	2.922750e-02	1.279490	97.453700	0.837835	2.791860e-03	0.297701	happy
2	83c20b83-7881-499d-a40d-cc06b65869f8	2	0.430551	3.103540e-04	43.123700	0.283289	0.202208	5.554470e+01	0.415176	surprise
3	83c20b83-7881-499d-a40d-cc06b65869f8	3	11.622700	1.721370e-04	8.821420	0.007699	0.813382	6.848630e+00	71.886000	neutral
4	83c20b83-7881-499d-a40d-cc06b65869f8	4	1.428140	2.151490e-03	82.596400	3.834700	0.931493	1.101040e+01	0.196761	fear

Finding the data type, missing and unique values of each column using the Autoviz library.

	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
movie_id	object	0.000000	1	nan	nan	Zero-variance colum: drop before modeling process.
image_seq	int64	0.000000	100	0.000000	99.000000	Possible ID colum: drop before modeling process.
angry	float64	0.000000	NA	0.000017	42.171200	has 13 outliers greater than upper bound (2.68) or lower than lower bound(-1.53). Cap them or remove them.
disgust	float64	0.000000	NA	0.000000	0.054649	has 18 outliers greater than upper bound (0.00) or lower than lower bound(-0.00). Cap them or remove them.
fear	float64	0.000000	NA	0.003125	95.139500	has 2 outliers greater than upper bound (87.14) or lower than lower bound(-48.12). Cap them or remove them.
happy	float64	0.000000	NA	0.002380	95.781900	has 5 outliers greater than upper bound (71.50) or lower than lower bound(-37.42). Cap them or remove them.
sad	float64	0.000000	NA	0.004126	98.309400	has 16 outliers greater than upper bound (22.65) or lower than lower bound(-12.79). Cap them or remove them.
surprise	float64	0.000000	NA	0.000029	91.265300	has 17 outliers greater than upper bound (8.73) or lower than lower bound(-4.99). Cap them or remove them.
neutral	float64	0.000000	NA	0.021739	99.918800	No issue
dominant_emotion	object	0.000000	6	nan	nan	No issue

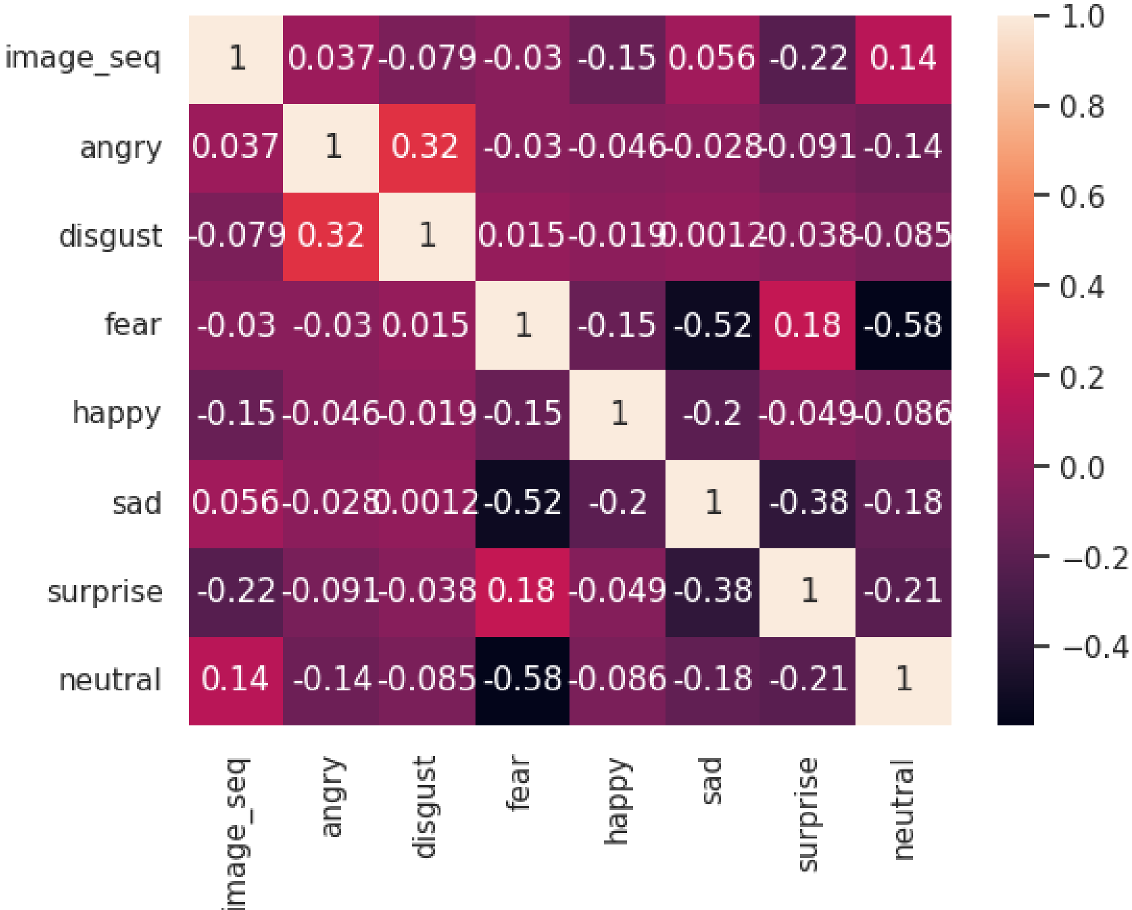
There are no missing values in the dataset. Only 2 columns, movie_id and dominant_emotion is of object data type each having 1 and 6 unique values As movie_id is same for a particular candidate, we can remove that column.

Calculating basic statistics for each emotion, such as mean, median, standard deviation, and range, to get an overview of the candidates' emotional expressions.

shape = (90, 10)

	image_seq	angry	disgust	fear	happy	sad	surprise	neutral
count	90.000000	90.000000	9.000000e+01	90.000000	9.000000e+01	90.000000	9.000000e+01	90.000000
mean	45.311111	3.856539	4.845111e-02	36.143804	4.215283e+00	32.656818	4.330881e+00	18.748220
std	26.393157	6.837119	2.822450e-01	29.611261	1.326247e+01	27.452548	1.180172e+01	26.503800
min	0.000000	0.002080	1.680020e-14	0.005674	1.959540e-09	0.150580	4.278190e-07	0.008691
25%	23.250000	0.246497	1.011588e-05	12.486100	8.162750e-03	7.890707	1.008948e-02	1.391075
50%	45.500000	1.197890	3.969240e-04	30.292750	3.268370e-01	26.850100	1.304275e-01	7.152030
75%	67.750000	4.197088	1.341535e-02	65.440375	1.797740e+00	56.829700	1.926030e+00	22.284400
max	90.000000	45.034900	2.656160e+00	99.031300	9.745370e+01	98.333400	6.127830e+01	99.061200

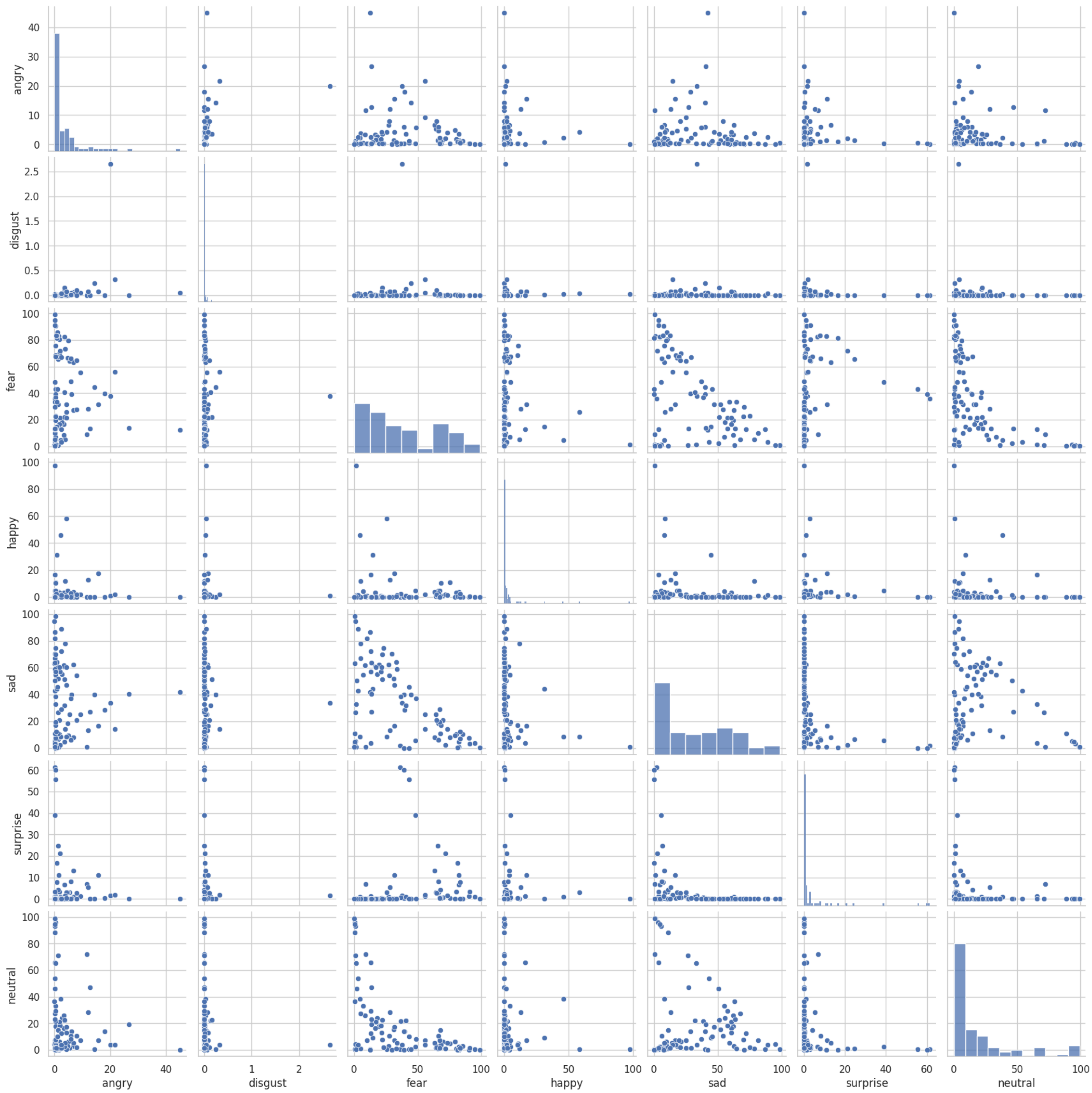
Plotting the Correlation matrix heatmap using seaborn library.



Sad and Fear Correlation: There is a strong negative correlation of -0.52 between 'Sad' and 'Fear' emotions. This negative correlation suggests that when 'Sad' scores increase, 'Fear' scores tend to decrease, and vice versa. This indicates an inverse relationship between the expressions of sadness and fear for Candidate 10.

Sad and Surprise Correlation: There is also a negative correlation of -0.38 between 'Sad' and 'Surprise' emotions. This negative correlation implies that when 'Sad' scores increase, 'Surprise' scores tend to decrease, and vice versa. This indicates an inverse relationship between the expressions of sadness and surprise for the candidate.

Plotting the scatterplot of each emotion with other emotion using pairplot.



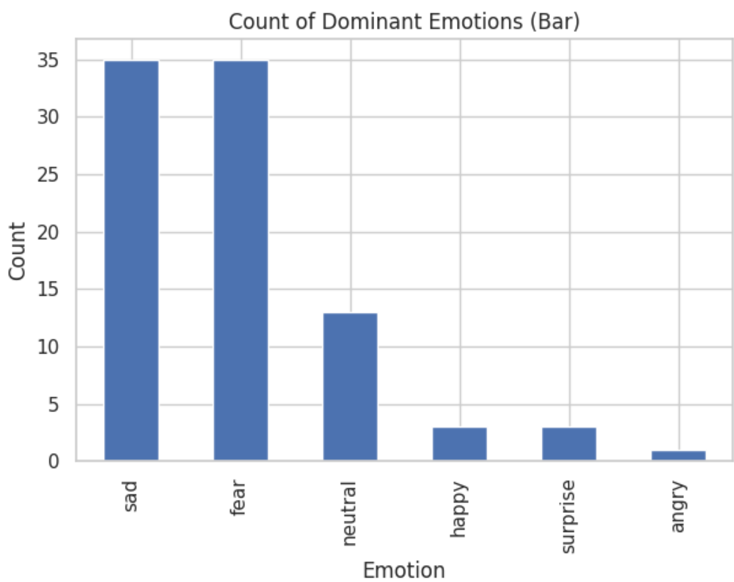
Happy Scores: 'Happy' scores are generally very low and exhibit a limited range of values, indicating that Candidate 10's expressions of happiness are infrequent and typically occur at low intensities.

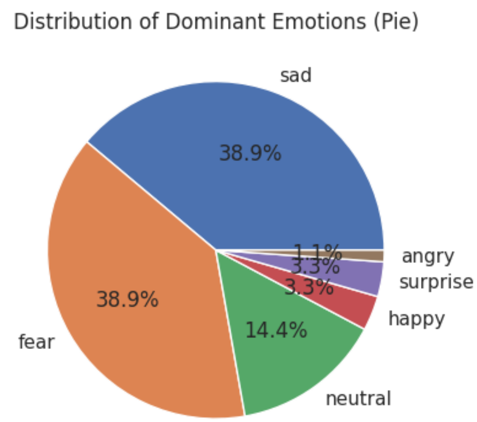
Sad Scores: 'Sad' scores are high and show a broader range of values, suggesting that Candidate 10 frequently expresses sadness during the video presentation. These expressions of sadness vary in intensity.

Fear Scores: Similarly, 'Fear' scores are also high and exhibit a broad range of values, indicating that Candidate 10 frequently expresses fear during the video. These expressions of fear vary in intensity.

Sad vs. Fear Plot: The scatterplot between 'Sad' and 'Fear' scores appears to be more scattered, indicating a wider range of emotional responses and less predictability in the relationship between sadness and fear.

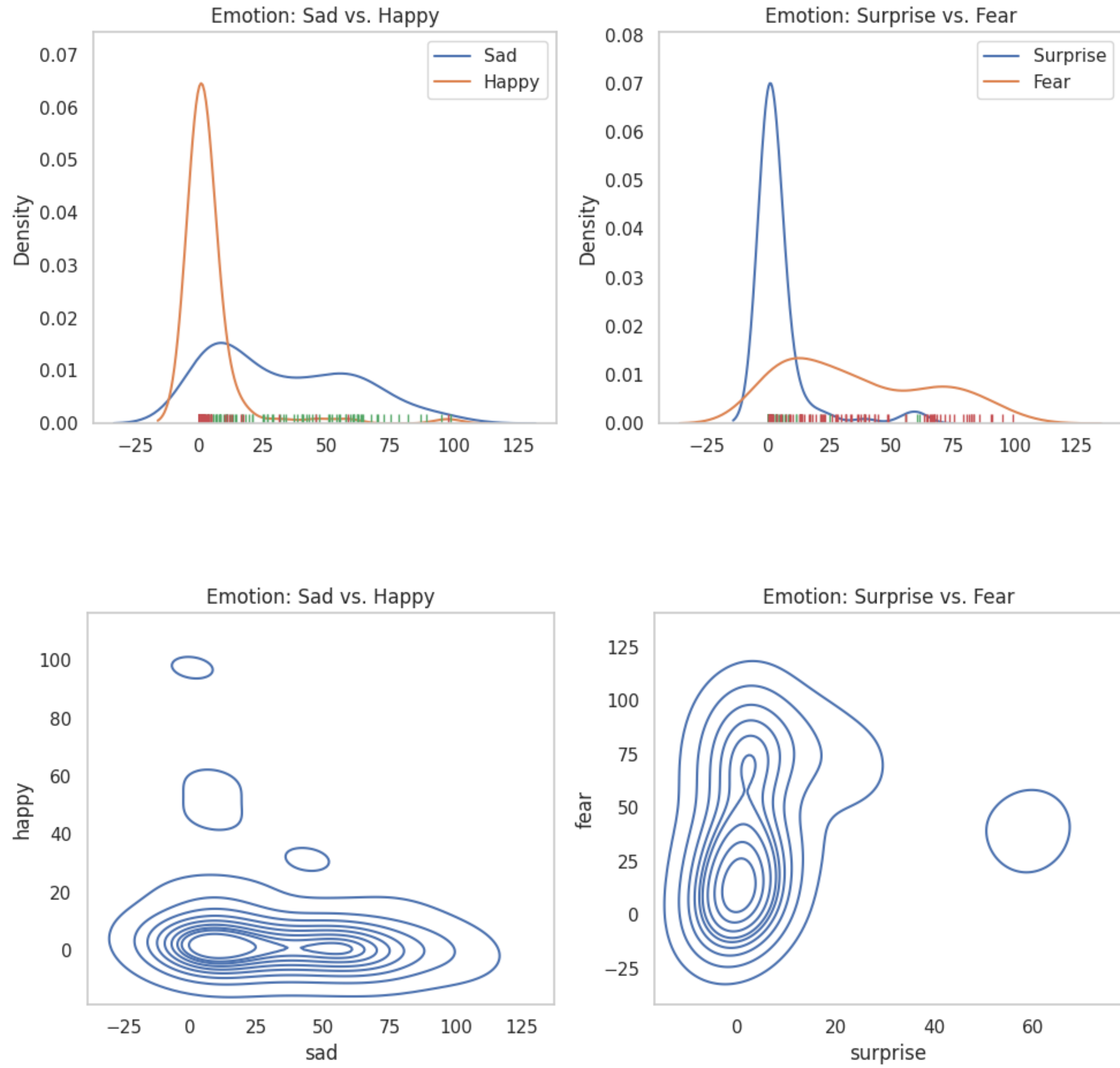
Plotting a bar chart and pie chart showing the occurences of dominant emotions.





1. Bar Chart:
   - The bar chart reveals that Candidate 10 predominantly possesses negative emotions, with 'Fear' and 'Sad' emotions each constituting approximately 39
   - There is limited presence of positive emotions, such as 'Happy' and 'Surprise,' which suggests that the candidate's expressions of happiness and surprise are relatively rare.
2. Pie Chart:
   - The pie chart provides a visual representation of the distribution of emotions, emphasizing that negative emotions (fear and sadness) dominate Candidate 10's emotional expressions, while positive emotions are less prevalent.

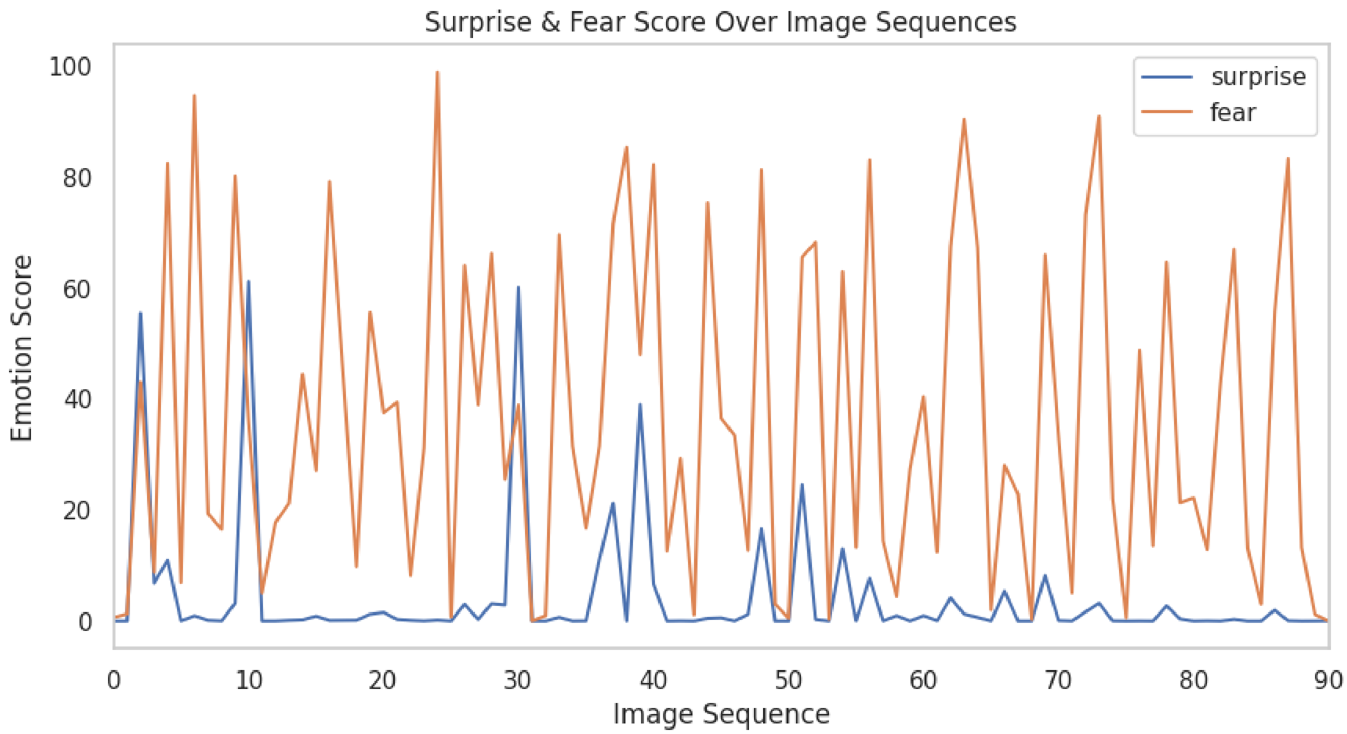
Now plotting the KDE plot between the 2 negative and 2 positive emotions to analyze the density.



- Happy and Surprise Scores: The KDE plot suggests that high 'Happy' and 'Surprise' scores have an approximate density of 0, indicating that Candidate 10's expressions of happiness and surprise at high intensities are infrequent during the video presentation. These emotions do not exhibit pronounced peaks or patterns at high values.
- Sad and Fear Scores: In contrast, 'Sad' and 'Fear' scores are more widespread and occupy a range from 0 to 100. This suggests that Candidate 10 expresses

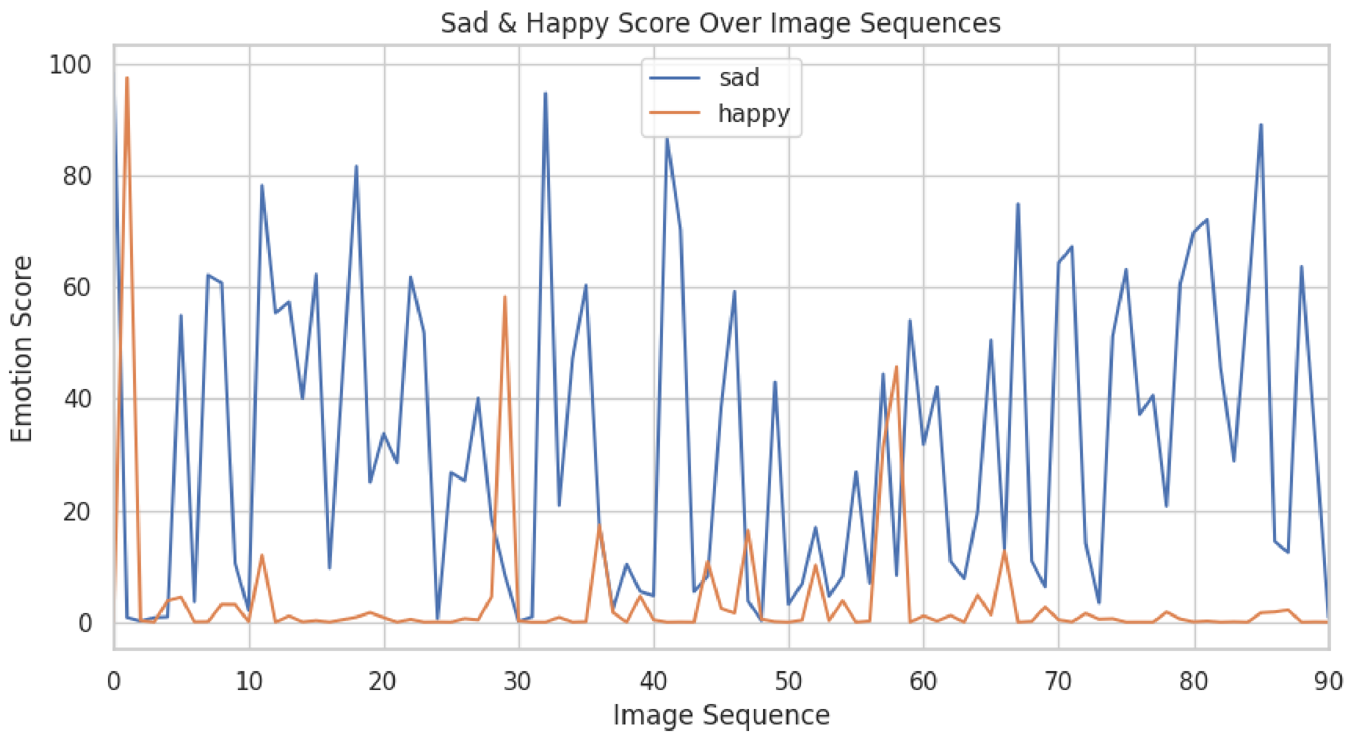
sadness and fear across a broader range of intensities, with moments of both mild and more intense sadness and fear scattered throughout the video.

Plotting line chart of emotion scores over image sequences.



Fear Scores: Throughout the video presentation, 'Fear' scores consistently maintain a noticeable presence, indicating that Candidate 10 frequently expresses fear during various segments of the video. The line plot may show fluctuating levels of fear.

Surprise Scores: In contrast, 'Surprise' scores appear to be less prominent and do not exhibit the same level of consistency or intensity as 'Fear' scores. This suggests that Candidate 10's expressions of surprise are comparatively subdued and occur less frequently during the video.



Sad Scores: Throughout the video presentation, 'Sad' scores consistently maintain a noticeable presence, indicating that Candidate 10 frequently expresses sadness during various segments of the video. The line plot may show fluctuating levels of sadness.

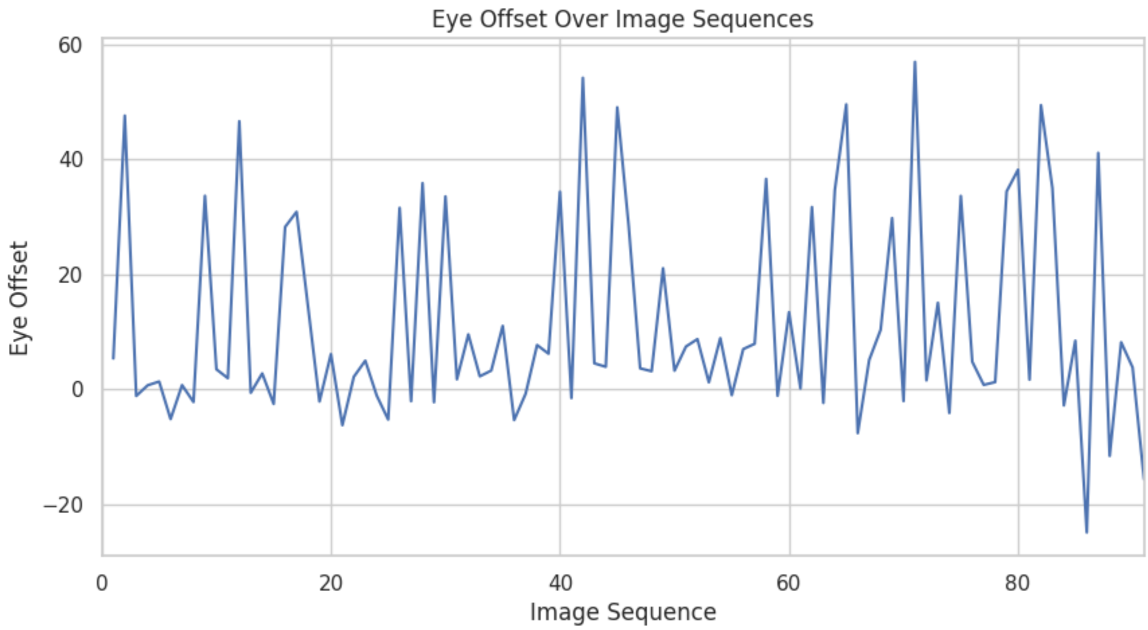
Happy Scores: In contrast, 'Happy' scores appear to be less prominent and do not exhibit the same level of consistency or intensity as 'Sad' scores. This suggests that Candidate 10's expressions of happiness are comparatively subdued and occur less frequently during the video.

Importing the data.

```
gaze = pd.read_csv(r".\emotion_data\10\gaze.csv")
gaze.head()
```

	movie_id	image_seq	gaze	blink	eye_offset
0	83c20b83-7881-499d-a40d-cc06b65869f8	1	1	1	5.3099
1	83c20b83-7881-499d-a40d-cc06b65869f8	2	0	1	47.5657
2	83c20b83-7881-499d-a40d-cc06b65869f8	3	1	0	-1.2162
3	83c20b83-7881-499d-a40d-cc06b65869f8	4	1	0	0.6371
4	83c20b83-7881-499d-a40d-cc06b65869f8	5	1	0	1.2903

Plotting line chart of eye_offset over image sequences.



The candidate has a eye offset lying between 0 - 40°

3 EDA After combining dataset

3.1 Emotion Scores

Inserting and combining all the emotion dataframes

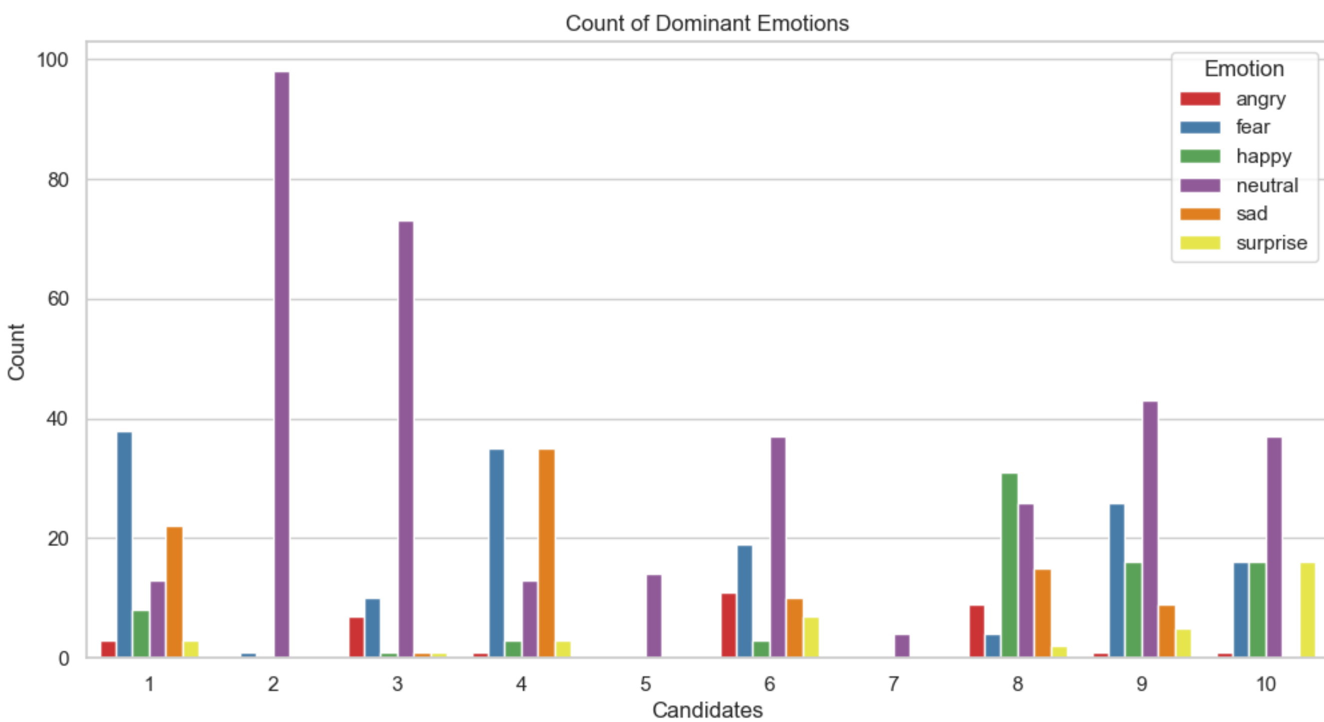
```
df1 = pd.read_csv(r"./emotion_data/1/emotion.csv")
df2 = pd.read_csv(r"./emotion_data/2/emotion.csv")
df3 = pd.read_csv(r"./emotion_data/3/emotion.csv")
df4 = pd.read_csv(r"./emotion_data/4/emotion.csv")
df5 = pd.read_csv(r"./emotion_data/5/emotion.csv")
df6 = pd.read_csv(r"./emotion_data/6/emotion.csv")
df7 = pd.read_csv(r"./emotion_data/7/emotion.csv")
df8 = pd.read_csv(r"./emotion_data/8/emotion.csv")
df9 = pd.read_csv(r"./emotion_data/9/emotion.csv")
df10 = pd.read_csv(r"./emotion_data/10/emotion.csv")

alldf = [df1,df2,df3,df4,df5,df6,df7,df8,df9,df10]
all_candidates = pd.concat([df1,df2,df3,df4,df5,df6,df7,df8,df9,df10])
```

Defining the list of emotions

```
emotions = ['angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral']
```

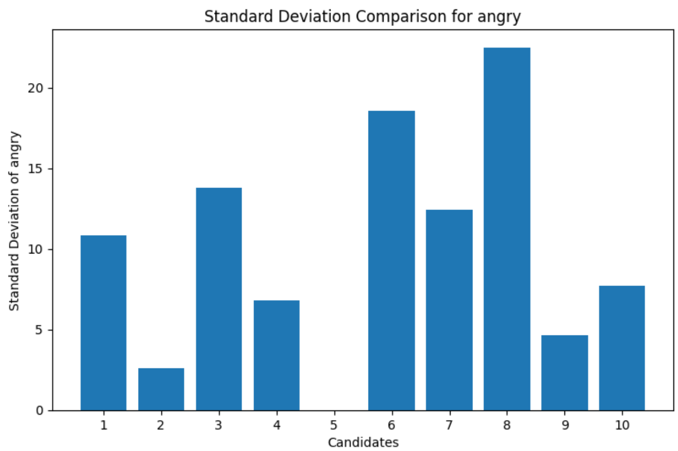
Count of Dominant Emotions

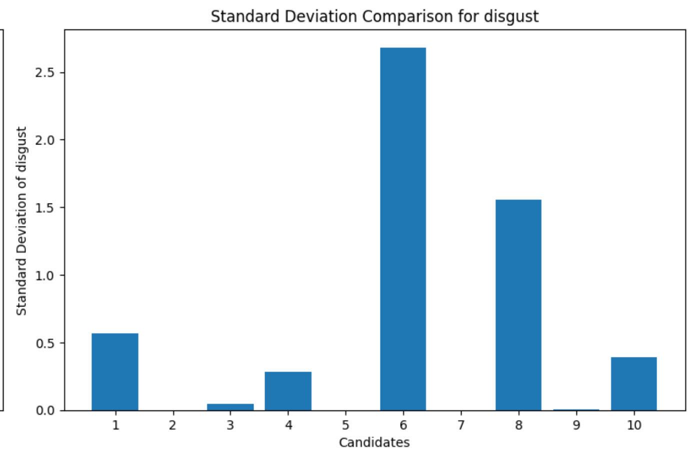
```
data_grouped = all_candidates.groupby(['movie_id',
↪  'dominant_emotion']).size().reset_index(name='count')

# Create a countplot
plt.figure(figsize=(12, 6))
sns.set(style="whitegrid")

# Create the countplot
ax = sns.barplot(data=data_grouped, x='movie_id', y='count', hue='dominant_emotion',
↪  palette='Set1')

# Set x-axis labels as numbers from 1 to 10
ax.set_xticklabels(range(1, 11))

# Set labels and title
plt.xlabel('Candidates')
plt.ylabel('Count')
plt.title('Count of Dominant Emotions')

# Show the plot
plt.legend(title='Emotion', loc='upper right')
plt.show()
```

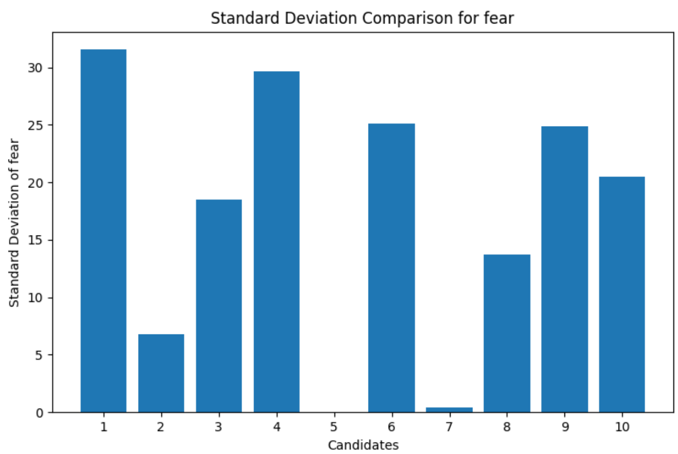


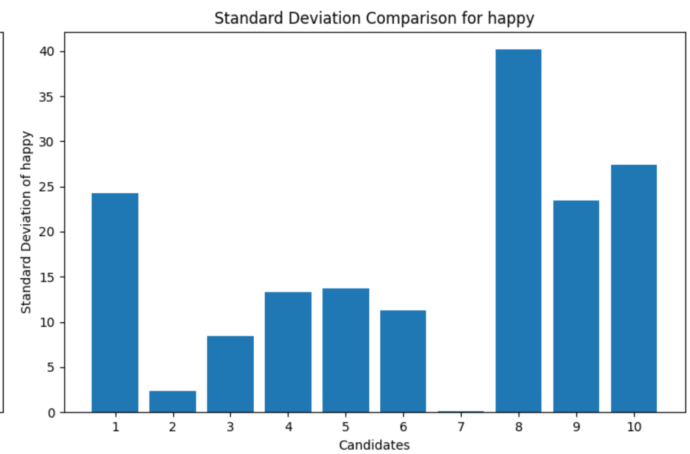
Plotting the bar graphs showing std of which emotion was dominant for which candidate

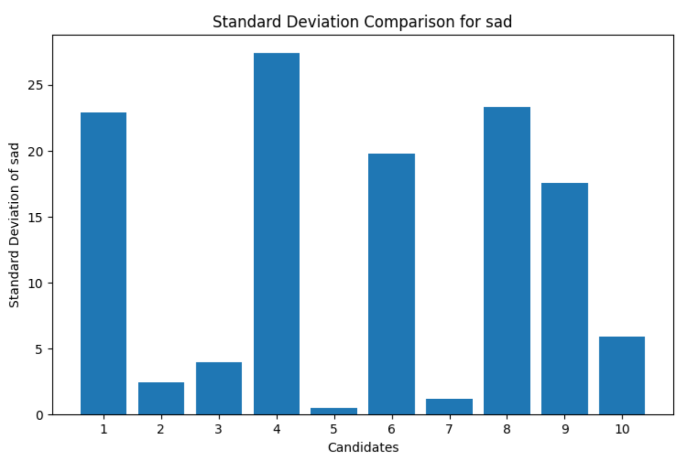
```python
# Determine the number of emotions and calculate the number of rows and columns
num_emotions = len(emotions)
num_cols = 2  # You want 2 columns
num_rows = (num_emotions + num_cols - 1) // num_cols  # Calculate the number of rows

# Create subplots for each emotion in the specified grid
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 5 * num_rows))

# Calculate and plot standard deviation for each emotion
for i, emotion in enumerate(emotions):
    row, col = divmod(i, num_cols)  # Calculate the row and column index for the subplot
    std_df = all_candidates.groupby('movie_id')[emotion].std().reset_index()

    sequential_numbers = list(range(1, len(std_df) + 1))

    # Plot the bar chart in the current subplot
    axes[row, col].bar(sequential_numbers, std_df[emotion])
    axes[row, col].set_xticks(sequential_numbers)  # Set x-axis ticks to custom list
    axes[row, col].set_xlabel('Candidates')
    axes[row, col].set_ylabel('Standard Deviation of ' + emotion)
    axes[row, col].set_title('Standard Deviation Comparison for ' + emotion)

if num_emotions < num_rows * num_cols:
    for i in range(num_emotions, num_rows * num_cols):
        fig.delaxes(axes.flatten()[i])

# Adjust layout and display subplots
plt.tight_layout()
plt.show()
```

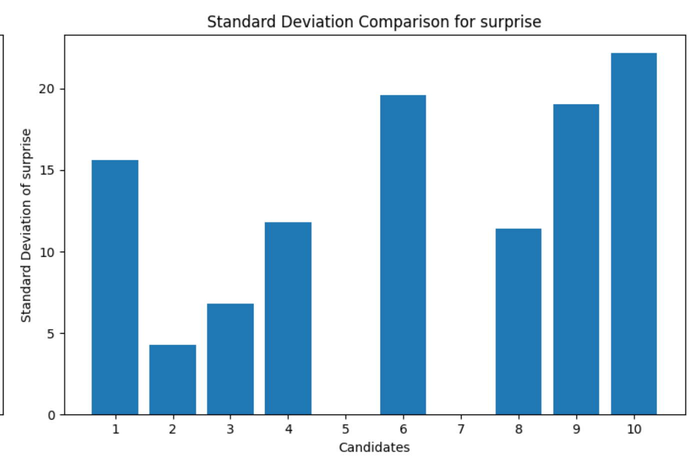

Standard Deviation Comparison for angry
Standard Deviation of angry
0 5 10 15 20
1 2 3 4 5 6 7 8 9 10
Candidates

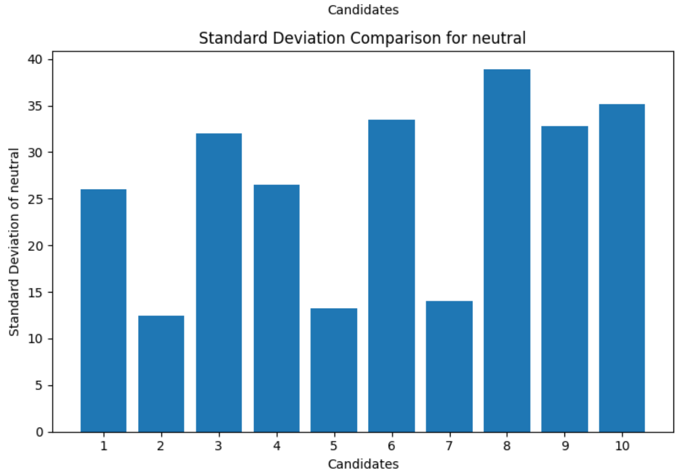

Standard Deviation Comparison for disgust
Standard Deviation of disgust
0.0 0.5 1.0 1.5 2.0 2.5
1 2 3 4 5 6 7 8 9 10
Candidates


Standard Deviation Comparison for fear
Standard Deviation of fear
0 5 10 15 20 25 30
1 2 3 4 5 6 7 8 9 10
Candidates


Standard Deviation Comparison for happy
Standard Deviation of happy
0 5 10 15 20 25 30 35 40
1 2 3 4 5 6 7 8 9 10
Candidates


Standard Deviation Comparison for sad
Standard Deviation of sad
0 5 10 15 20 25
1 2 3 4 5 6 7 8 9 10
Candidates


Standard Deviation Comparison for surprise
Standard Deviation of surprise
0 5 10 15 20
1 2 3 4 5 6 7 8 9 10
Candidates


Standard Deviation Comparison for neutral
Standard Deviation of neutral
0 5 10 15 20 25 30 35 40
1 2 3 4 5 6 7 8 9 10
Candidates

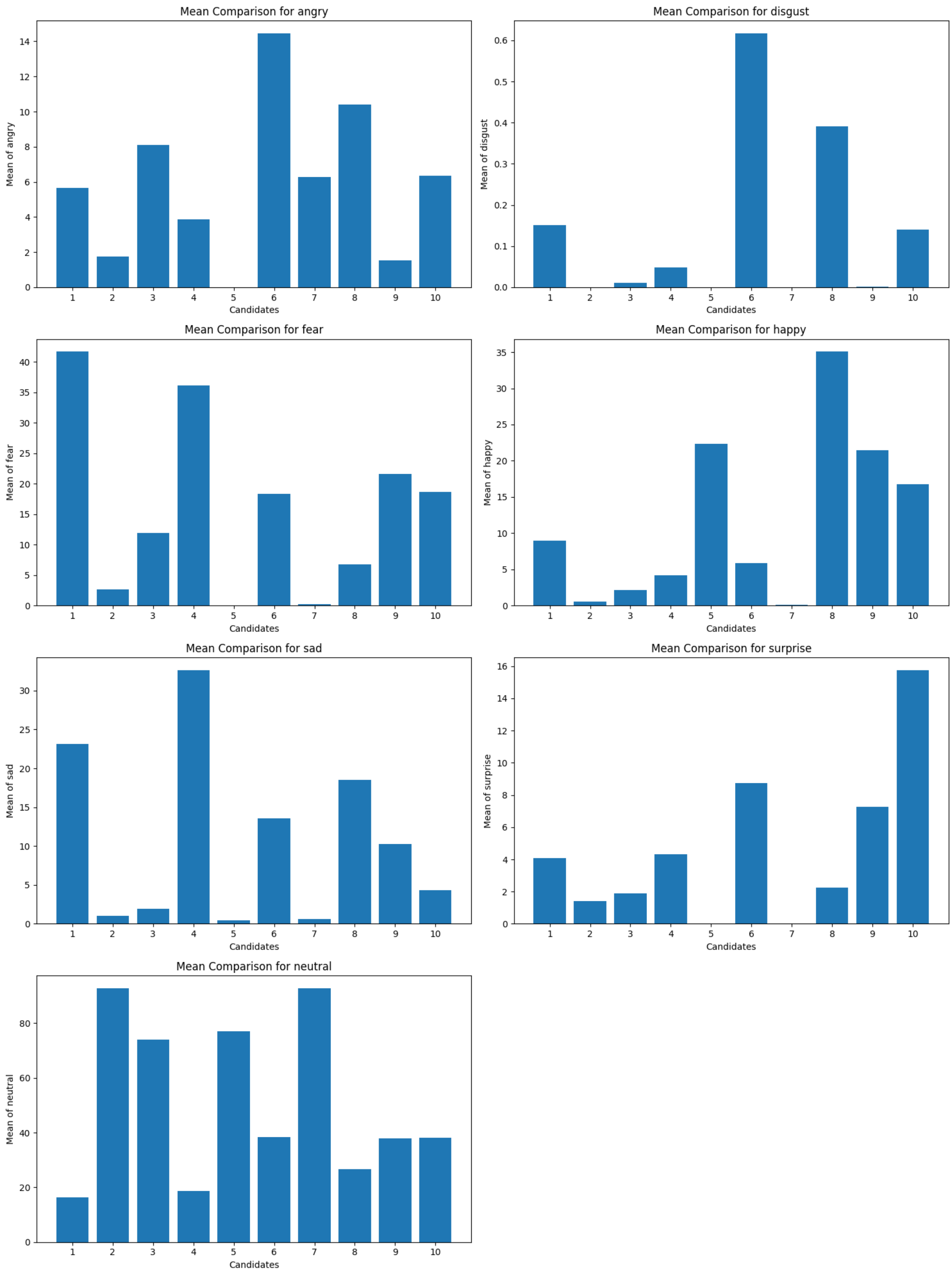
Plotting the bar graphs showing mean of which emotion was dominant for which candidate

- Candidate 1: High fear and sad score, low surprise and happy score.
- Candidate 2: High neutral score, very low happy and surprise score.
- Candidate 3: High neutral and angry score, low happy score.
- Candidate 4: High fear and sad score, low happy and neutral score.
- Candidate 5: Very less emotion data
- Candidate 6: Very less emotion data
- Candidate 7: High neutral score, zero happy and surprise score.
- Candidate 8: High happy and angry score
- Candidate 9: High happy and surprise scores, low angry and sad scores.
- Candidate 10: High surprise score, moderate angry, happy and fear score.

Candidate 1 and Candidate 4 has more fearful and sad emotions rather than happy and surprise. So they are also not a good fit to hire.

Candidate 10 is very surprised as compared to others. After reading his transcript, I realised he is applying for a role in Finance(Accounting Associate,Tax Associate etc.). In the context of a role like an "Accounting Associate," excessive "surprise" emotion may not be seen as a particularly positive trait. Accounting positions typically require a high degree of attention to detail, precision, and adherence to established financial procedures and regulations. So Candidate 10 is also not good to hire.

Candidate 9 may be a potential candidate to consider. They exhibit high scores in 'Happy' and 'Surprise,' which could suggest a generally positive and adaptable emotional disposition. Additionally, their 'Angry' and 'Sad' scores are comparatively lower, indicating a relatively lower frequency of negative emotions.


Mean Comparison for angry
Mean of angry
Candidates
Mean Comparison for disgust
Mean of disgust
Candidates
Mean Comparison for fear
Mean of fear
Candidates
Mean Comparison for happy
Mean of happy
Candidates
Mean Comparison for sad
Mean of sad
Candidates
Mean Comparison for surprise
Mean of surprise
Candidates
Mean Comparison for neutral
Mean of neutral
Candidates

3.2 Gaze

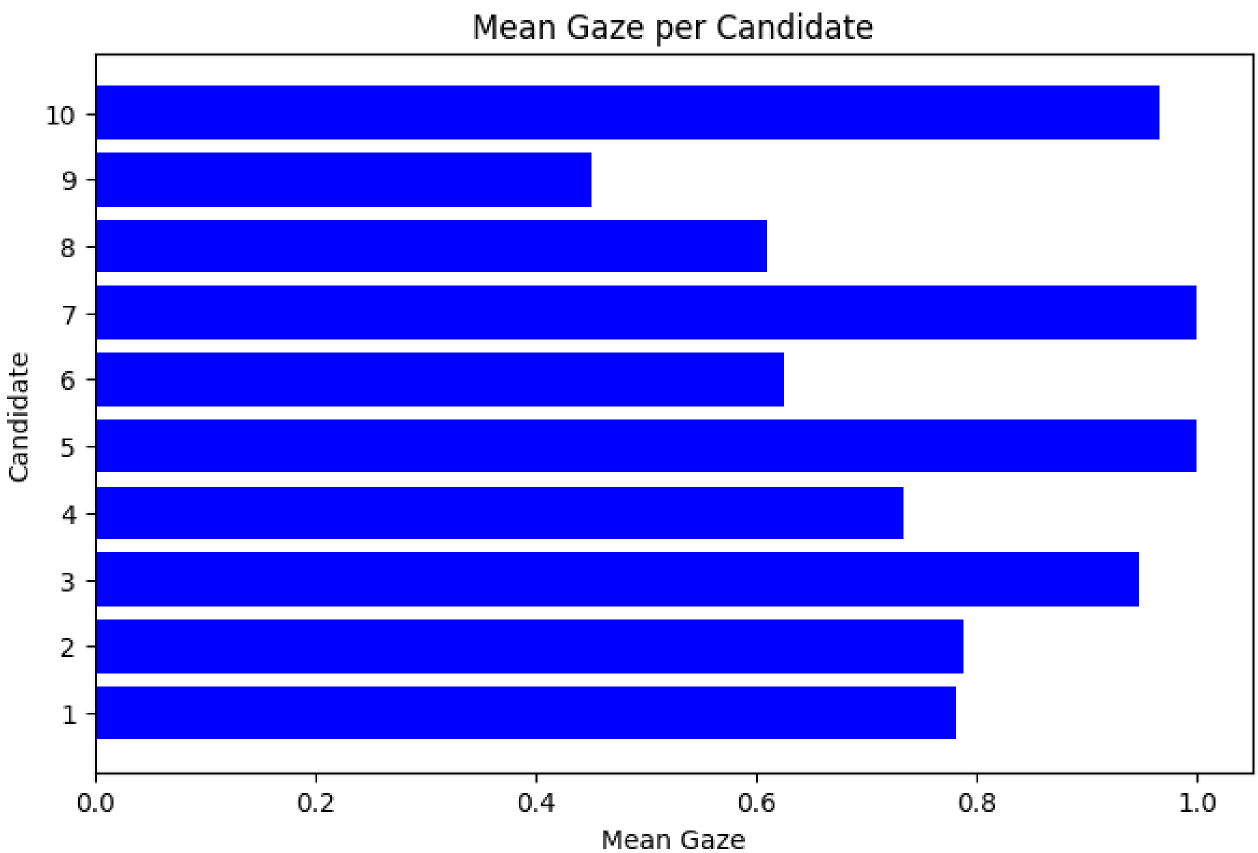
Importing the gaze data of all the candidates

```
eye1 = pd.read_csv(r"./emotion_data/1/gaze.csv")
eye2 = pd.read_csv(r"./emotion_data/2/gaze.csv")
eye3 = pd.read_csv(r"./emotion_data/3/gaze.csv")
eye4 = pd.read_csv(r"./emotion_data/4/gaze.csv")
eye5 = pd.read_csv(r"./emotion_data/5/gaze.csv")
eye6 = pd.read_csv(r"./emotion_data/6/gaze.csv")
eye7 = pd.read_csv(r"./emotion_data/7/gaze.csv")
eye8 = pd.read_csv(r"./emotion_data/8/gaze.csv")
eye9 = pd.read_csv(r"./emotion_data/9/gaze.csv")
eye10 = pd.read_csv(r"./emotion_data/10/gaze.csv")

alleye = pd.concat([eye1,eye2,eye3,eye4,eye5,eye6,eye7,eye8,eye9,eye10])
```

Calculating and plotting the mean gaze of each candidate.

```
gaze_count = alleye.groupby('movie_id')['gaze'].sum().reset_index()
movie_id_count = alleye['movie_id'].value_counts().reset_index()

merged_data = pd.merge(gaze_count,movie_id_count,on='movie_id')
merged_data['mean_gaze'] = merged_data['gaze'] / merged_data['count']

# Display the resulting DataFrame
print(merged_data)

# Create a figure and axis
fig, ax = plt.subplots(figsize=(8, 5))

# Calculate the count values from 1 to 10
count_values = np.arange(1, 11)

# Extract the mean gaze values
mean_gaze_values = merged_data['mean_gaze']

# Plot the horizontal bar chart
ax.barh(count_values, mean_gaze_values, color='b')
ax.set_yticks(count_values)
ax.set_yticklabels(count_values)
ax.set_xlabel('Mean Gaze')
ax.set_ylabel('Candidate')
ax.set_title('Mean Gaze per Candidate')

# Show the plot
plt.show()
```

	movie_id	gaze	count	mean_gaze
0	6539370c-256e-4ed2-9d00-1be1f051163f	68	87	0.781609
1	6b0386fc-41de-4196-b0d6-3d0b815c2dbc	78	99	0.787879
2	813af424-a584-4417-b7ee-0d4c705e83c9	88	93	0.946237
3	83c20b83-7881-499d-a40d-cc06b65869f8	66	90	0.733333
4	92016995-e455-4651-9f6e-fbca0d423f21	14	14	1.000000
5	93663f94-bf0a-4ce8-a29a-a5236cc7fe6a	55	88	0.625000
6	9c350343-e895-49df-af90-d50b91d19d3e	4	4	1.000000
7	baa26895-85b2-465b-a972-649b41d9870e	53	87	0.609195
8	d0b9170b-98b9-48e1-a1b2-1d661bb0d853	45	100	0.450000
9	dfb0d746-609f-4dac-8e1d-c0325fb64394	83	86	0.965116

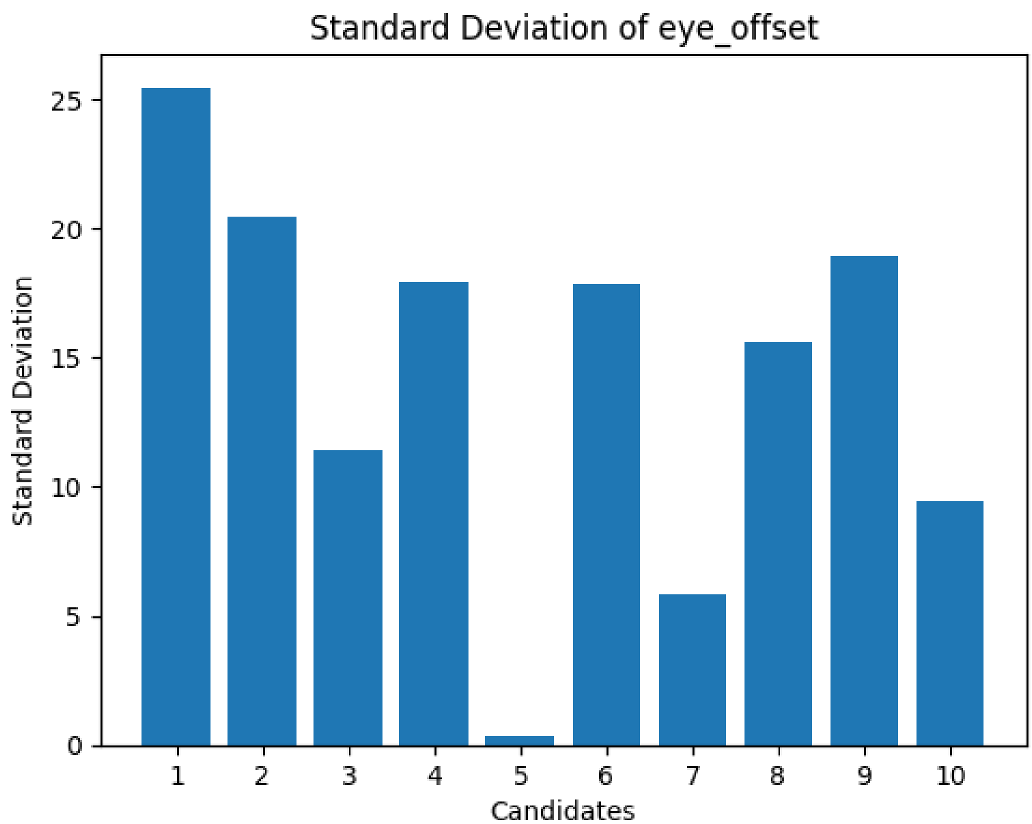


Candidate 3,5,7, and 10 have a very high mean gaze. High mean gaze, indicating that these candidates maintain consistent eye contact with the camera, can be seen as a positive non-verbal behavior during a video presentation or interview.

Maintaining eye contact with the camera suggests confidence, engagement, and a strong connection with the audience, which can be valuable in a professional context. It's generally considered a positive trait in job interviews and presentations, as it conveys attentiveness and sincerity.

Plotting the standard deviation of eye offset of each candidate.

```python
# Group the DataFrame by the 'group_column' and calculate the standard deviation of
↪   'data_column'
grouped_data = alleye.groupby('movie_id')['eye_offset'].std()

# Calculate the count values from 1 to 10
x_values = np.arange(1, 11)

# Plot the bar chart with custom x-axis
plt.bar(x_values, grouped_data)
plt.xlabel('Candidates')
plt.ylabel('Standard Deviation')
plt.title(f'Standard Deviation of eye_offset')
plt.xticks(x_values, x_values)  # Set x-axis labels to the custom values
plt.show()
```

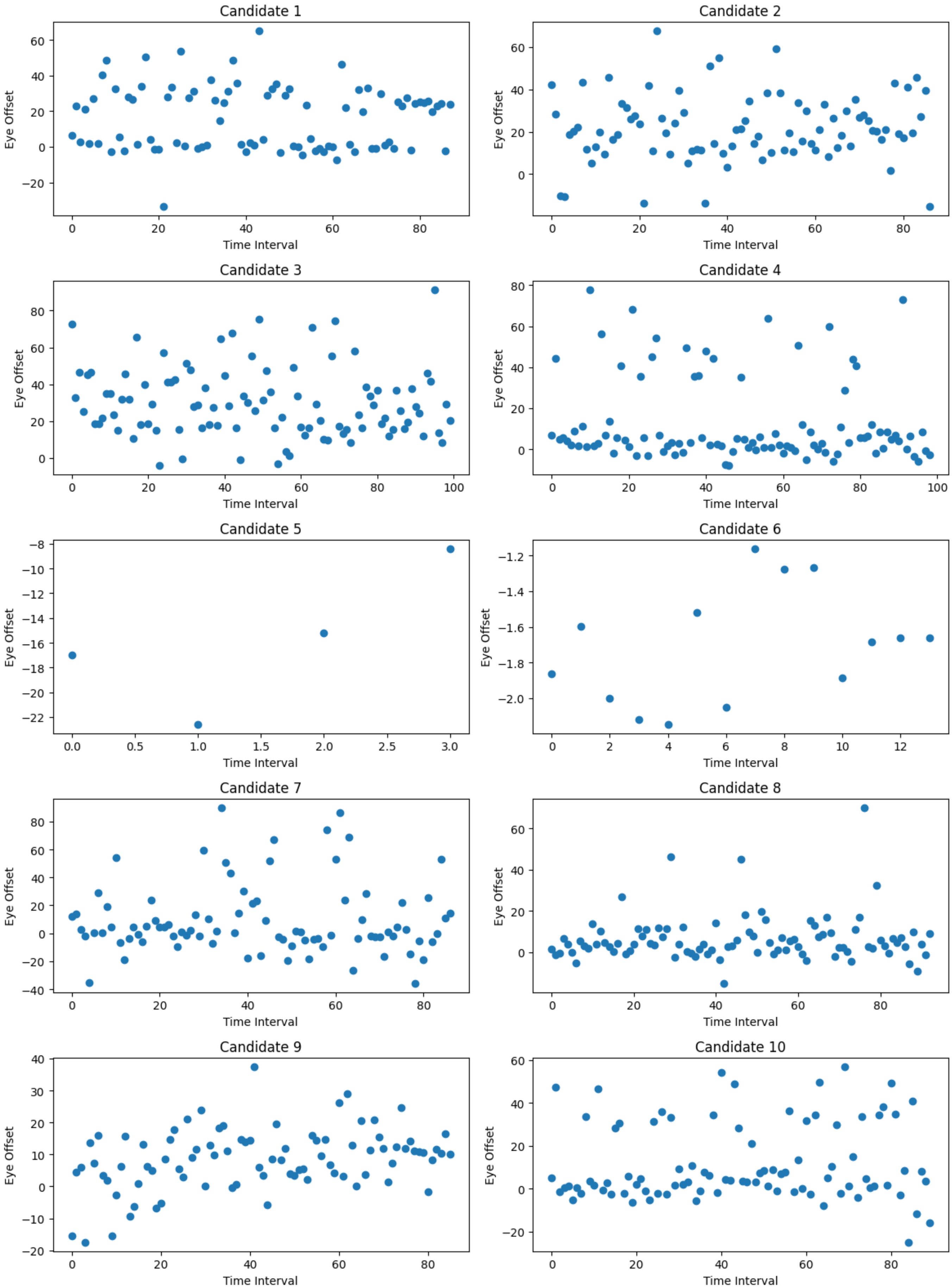


Candidate 1 and Candidate 2 has a very high eye offset. A high eye offset, indicating that these candidates keep rotating their eyes, can indeed be an important non-verbal cue during a video presentation.

Frequent eye movement or a lack of steady eye contact with the camera may suggest that the candidates are reading from a script or transcript rather than engaging directly with the camera or audience. This behavior can be interpreted in several ways, including a potential lack of confidence, discomfort, or a need for external assistance while speaking.

Plotting scatterplots of eye_offset with time interval for each candidate.

```python
# Assuming 'alleye' is your DataFrame
# Get unique movie_id values
unique_movie_ids = alleye['movie_id'].unique()

# Define the number of rows and columns for the subplots
num_rows = 5
num_cols = 2

# Create subplots
fig, axs = plt.subplots(num_rows, num_cols, figsize=(12, 16))  # Adjust figsize as
↪  needed

# Create scatterplots for each unique movie_id
for i, movie_id in enumerate(unique_movie_ids):
    subset_df = alleye[alleye['movie_id'] == movie_id]

    # Calculate row and column indices
    row = i // num_cols
    col = i % num_cols

    # Use a 2D array for subplots
    ax = axs[row, col]

    ax.scatter(subset_df.index, subset_df['eye_offset'])
    ax.set_xlabel('Time Interval')
    ax.set_ylabel('Eye Offset')
    ax.set_title(f'Candidate {i + 1}')  # Use numbers from 1 to 10 as titles

# Fill in any empty subplots
for i in range(len(unique_movie_ids), num_rows * num_cols):
    row = i // num_cols
    col = i % num_cols
    fig.delaxes(axs[row, col])

# Adjust subplot layout
plt.tight_layout()

# Show the plots
plt.show()
```


Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10
Eye Offset
Time Interval

Scatterplot of eye_offset shows that candidate 2,3 and 7 keeps on changing there eye angles from the camera. Rest all the candidates have eyes near 0 degree which means they are looking in the camera most of the time. Candidate 2,3 and 7 might be reading the transcript from a screen or paper or they might be indulge in some unfair practices.

3.3 Transcript Scores

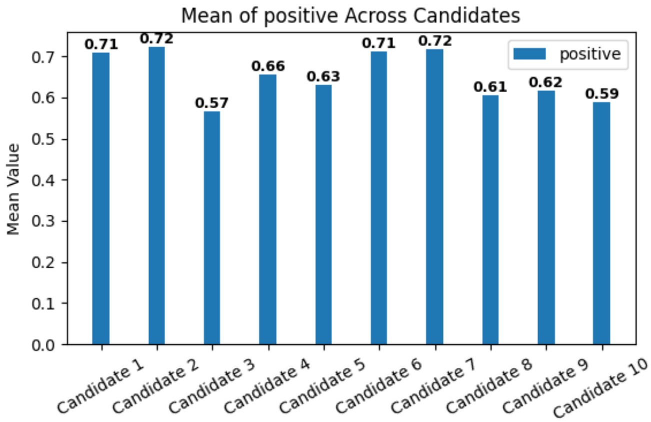
Importing Transcript scores of each candidate.

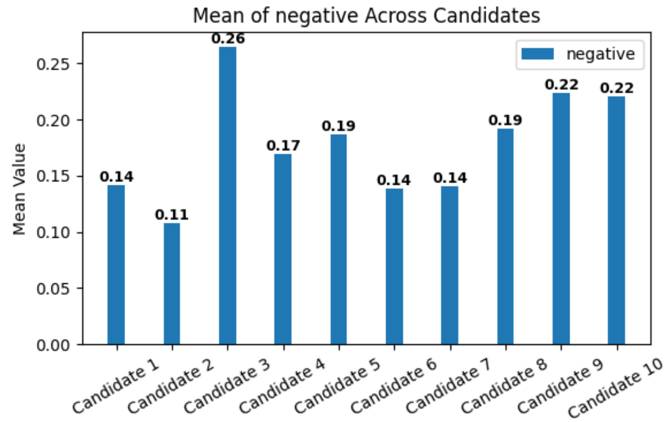
```
tf1 = pd.read_csv('./transcript_data/1.csv')
tf2 = pd.read_csv('./transcript_data/2.csv')
tf3 = pd.read_csv('./transcript_data/3.csv')
tf4 = pd.read_csv('./transcript_data/4.csv')
tf5 = pd.read_csv('./transcript_data/5.csv')
tf6 = pd.read_csv('./transcript_data/6.csv')
tf7 = pd.read_csv('./transcript_data/7.csv')
tf8 = pd.read_csv('./transcript_data/8.csv')
tf9 = pd.read_csv('./transcript_data/9.csv')
tf10 = pd.read_csv('./transcript_data/10.csv')

dataframes = [tf1, tf2, tf3, tf4, tf5, tf6, tf7, tf8, tf9, tf10]
```

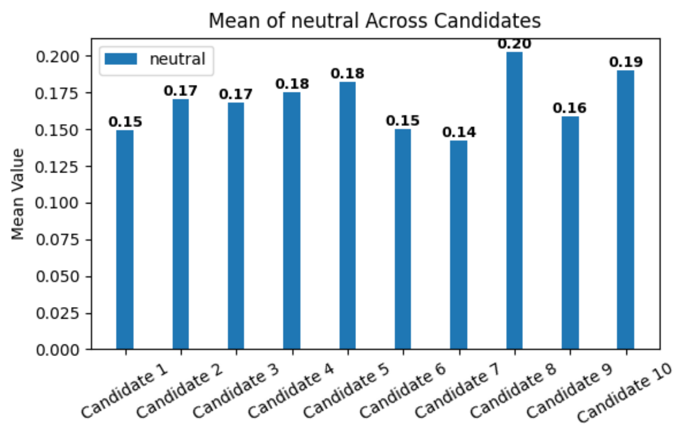
Plotting mean of different scores across different candidates

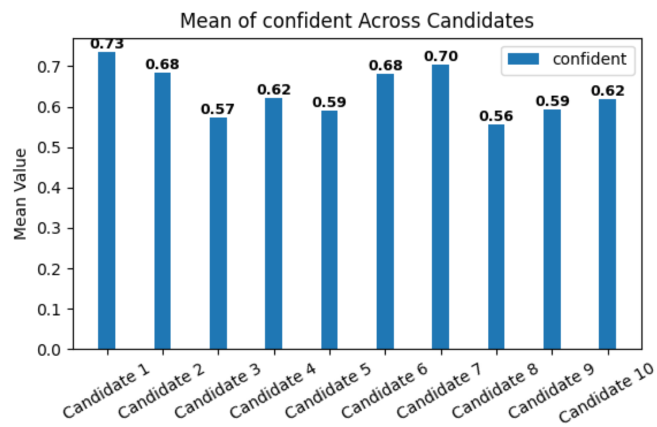
```
# Define the list of columns to compare (replace with your column names)
columns_to_compare = ['positive', 'negative', 'neutral', 'confident', 'hesitant',
↪  'concise', 'enthusiastic', 'speech_speed']

# Initialize a dictionary to store means for each column
column_means = {col: [] for col in columns_to_compare}

# Calculate means for each column across dataframes
for col in columns_to_compare:
    for df in dataframes:
        mean = df[col].mean()
        column_means[col].append(mean)

# Define labels for the dataframes (optional, replace with actual labels)
labels = ['Candidate 1', 'Candidate 2', 'Candidate 3', 'Candidate 4', 'Candidate 5',
          'Candidate 6', 'Candidate 7', 'Candidate 8', 'Candidate 9', 'Candidate
          ↪  10']

# Create an array of x positions for the bars
x = np.arange(len(labels))

# Define the width of each bar group
```

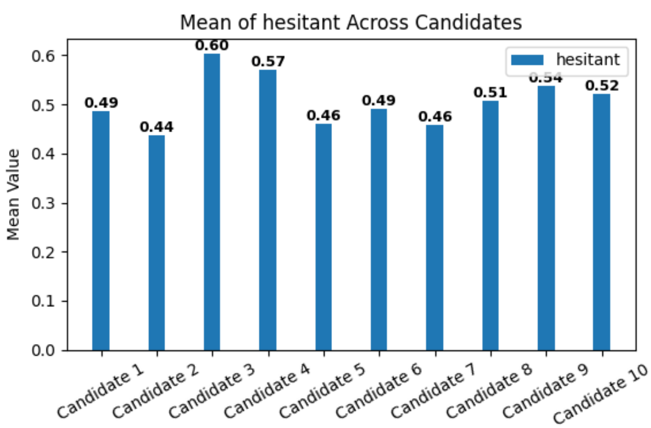
```python
bar_width = 0.30

# Create subplots for each column
fig, axs = plt.subplots(4, 2, figsize=(12, 16))
axs = axs.ravel()

# Iterate through each column and create a subplot
for i, col in enumerate(columns_to_compare):
    # Create bars for the current column
    bars = axs[i].bar(x, column_means[col], bar_width, label=col)

    # Set the x-axis labels
    axs[i].set_xticks(x)
    axs[i].set_xticklabels(labels,rotation=30)

    # Add labels, title, and legend
    axs[i].set_xlabel(' ')
    axs[i].set_ylabel('Mean Value')
    axs[i].set_title(f'Mean of {col} Across Candidates')
    axs[i].legend()

    # Add values above the bars
    for bar in bars:
        height = bar.get_height()
        axs[i].annotate(f'{height:.2f}', (bar.get_x() + bar.get_width() / 2,
        ↪  height),
                        ha='center', va='bottom', fontsize=9, color='black',
                        ↪  weight='bold')

# Adjust subplot layout
plt.tight_layout()

# Show the subplots
plt.show()
```

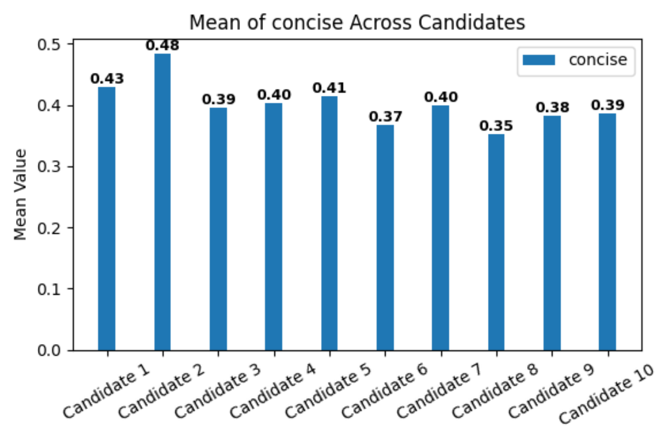

Mean of positive Across Candidates
positive
Mean Value
0.71
0.72
0.57
0.66
0.63
0.71
0.72
0.61
0.62
0.59
Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10

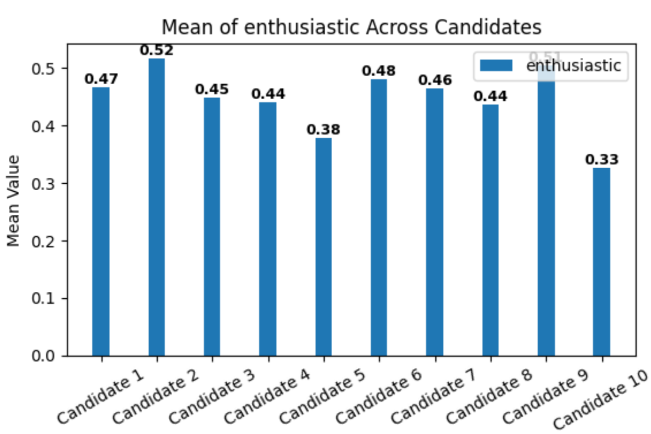

Mean of negative Across Candidates
negative
Mean Value
0.14
0.11
0.26
0.17
0.19
0.14
0.14
0.19
0.22
0.22
Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10

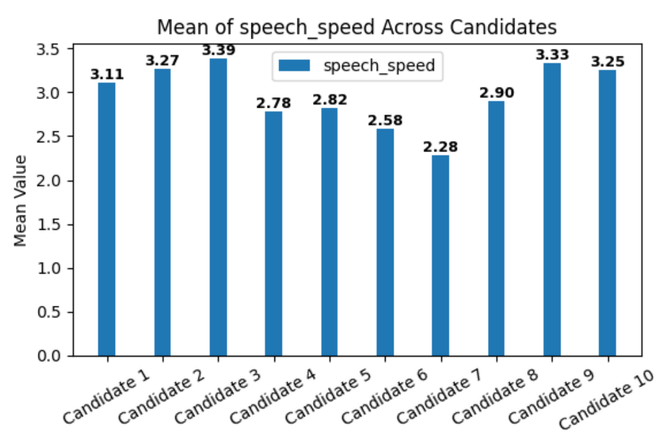

Mean of neutral Across Candidates
neutral
Mean Value
0.15
0.17
0.17
0.18
0.18
0.15
0.14
0.20
0.16
0.19
Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10


Mean of confident Across Candidates
confident
Mean Value
0.73
0.68
0.57
0.62
0.59
0.68
0.70
0.56
0.59
0.62
Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10


Mean of hesitant Across Candidates
hesitant
Mean Value
0.49
0.44
0.60
0.57
0.46
0.49
0.46
0.51
0.52
Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10


Mean of concise Across Candidates
concise
Mean Value
0.43
0.48
0.39
0.40
0.41
0.37
0.40
0.35
0.38
0.39
Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10


Mean of enthusiastic Across Candidates
enthusiastic
Mean Value
0.47
0.52
0.45
0.44
0.38
0.48
0.46
0.44
0.33
Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10

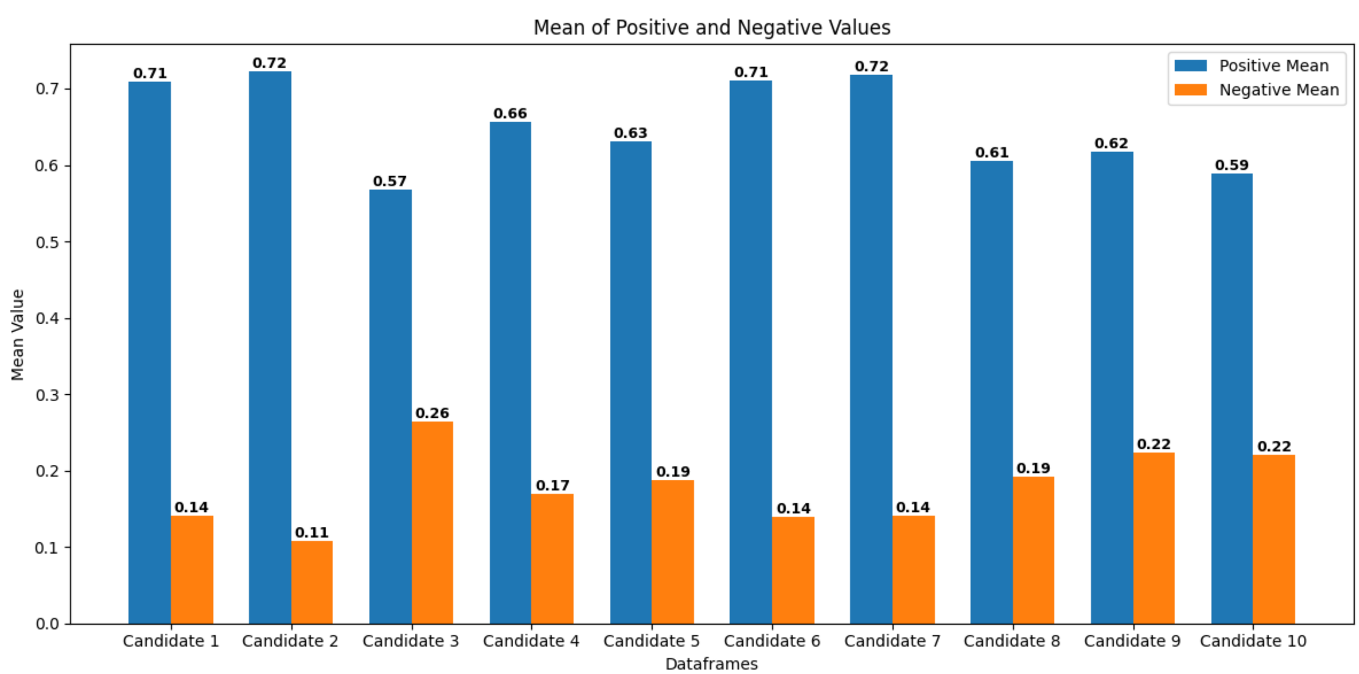

Mean of speech_speed Across Candidates
speech_speed
Mean Value
3.11
3.27
3.39
2.78
2.82
2.58
2.28
2.90
3.33
3.25
Candidate 1
Candidate 2
Candidate 3
Candidate 4
Candidate 5
Candidate 6
Candidate 7
Candidate 8
Candidate 9
Candidate 10

- The negative mean of Candidate 3 is quite high than its peers.
- The speech speed mean of Candidate 3 is very high.
- The positive mean of candidate 2 and 6 is high.
- The concise score of candidate 2 is high.
- The hesitant score of candidate 2 is low.
- The enthusiastic score of candidate 2 and 9 is high.

Plotting mean of positive and negative values.

```
# Initialize lists to store means of positive and negative values for each dataframe
positive_means = []
negative_means = []

# Calculate means for each dataframe
for df in dataframes:
    # Assuming your dataframes have columns 'positive' and 'negative'
    positive_mean = df['positive'].mean()
    negative_mean = df['negative'].mean()
    positive_means.append(positive_mean)
    negative_means.append(negative_mean)

# Define labels for the dataframes (optional, replace with actual labels)
labels = ['Candidate 1', 'Candidate 2', 'Candidate 3', 'Candidate 4', 'Candidate 5',
          'Candidate 6', 'Candidate 7', 'Candidate 8', 'Candidate 9', 'Candidate
          ↪  10']

# Create an array of x positions for the bars
x = np.arange(len(labels))

# Define the width of each bar group
bar_width = 0.35

# Create the figure and axis objects
fig, ax = plt.subplots(figsize=(12, 6))

# Create bars for positive means
positive_bars = ax.bar(x - bar_width/2, positive_means, bar_width, label='Positive
↪  Mean')

# Create bars for negative means
negative_bars = ax.bar(x + bar_width/2, negative_means, bar_width, label='Negative
↪  Mean')

# Set the x-axis labels
ax.set_xticks(x)
ax.set_xticklabels(labels)
```

```

# Add labels, title, and legend
ax.set_xlabel('Dataframes')
ax.set_ylabel('Mean Value')
ax.set_title('Mean of Positive and Negative Values')
ax.legend()

# Add values above the bars
for bar1, bar2 in zip(positive_bars, negative_bars):
    height1 = bar1.get_height()
    height2 = bar2.get_height()
    ax.annotate(f'{height1:.2f}', (bar1.get_x() + bar1.get_width() / 2, height1),
                ha='center', va='bottom', fontsize=9, color='black', weight='bold')
    ax.annotate(f'{height2:.2f}', (bar2.get_x() + bar2.get_width() / 2, height2),
                ha='center', va='bottom', fontsize=9, color='black', weight='bold')

# Show the plot
plt.tight_layout()
plt.show()
```



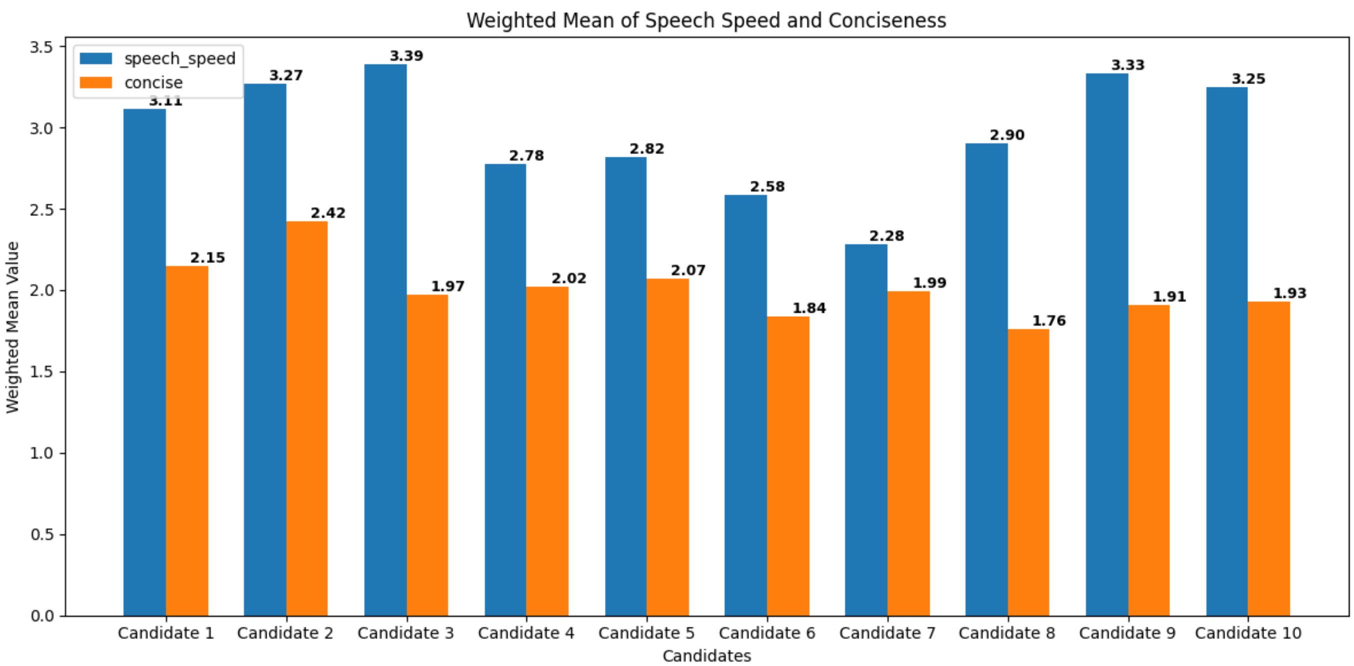
Candidate 2 has a large difference between the positive and negative mean.

Plotting weighted mean of speech speed and conciseness

```
# Define the columns to compare (replace with your column names)
columns_to_compare = ['speech_speed', 'concise']

```

```python
# Define weights for each column
column_weights = {'speech_speed': 1, 'concise': 5}

# Initialize lists to store weighted means for each column
weighted_means = {col: [] for col in columns_to_compare}

# Calculate weighted means for each dataframe
for col in columns_to_compare:
    for df in dataframes:
        weighted_mean = df[col].mean() * column_weights[col]
        weighted_means[col].append(weighted_mean)

# Define labels for the dataframes (optional, replace with actual labels)
labels = ['Candidate 1', 'Candidate 2', 'Candidate 3', 'Candidate 4', 'Candidate 5',
          'Candidate 6', 'Candidate 7', 'Candidate 8', 'Candidate 9', 'Candidate
          ↪  10']

# Create an array of x positions for the bars
x = np.arange(len(labels))

# Define the width of each bar group
bar_width = 0.35

# Create the figure and axis objects
fig, ax = plt.subplots(figsize=(12, 6))

# Create bars for weighted means
for i, col in enumerate(columns_to_compare):
    bars = ax.bar(x + i * bar_width, weighted_means[col], bar_width, label=col)

# Set the x-axis labels
ax.set_xticks(x + bar_width / 2)
ax.set_xticklabels(labels)

# Add labels, title, and legend
ax.set_xlabel('Candidates')
ax.set_ylabel('Weighted Mean Value')
ax.set_title('Weighted Mean of Speech Speed and Conciseness')
ax.legend()

# Add values above the bars
for i in range(len(labels)):
    for col in columns_to_compare:
        height = weighted_means[col][i]
        ax.annotate(f'{height:.2f}', (x[i] + bar_width / 2 +
        ↪  columns_to_compare.index(col) * bar_width, height),
                    ha='center', va='bottom', fontsize=9, color='black',
                    ↪  weight='bold')

# Adjust subplot layout
```

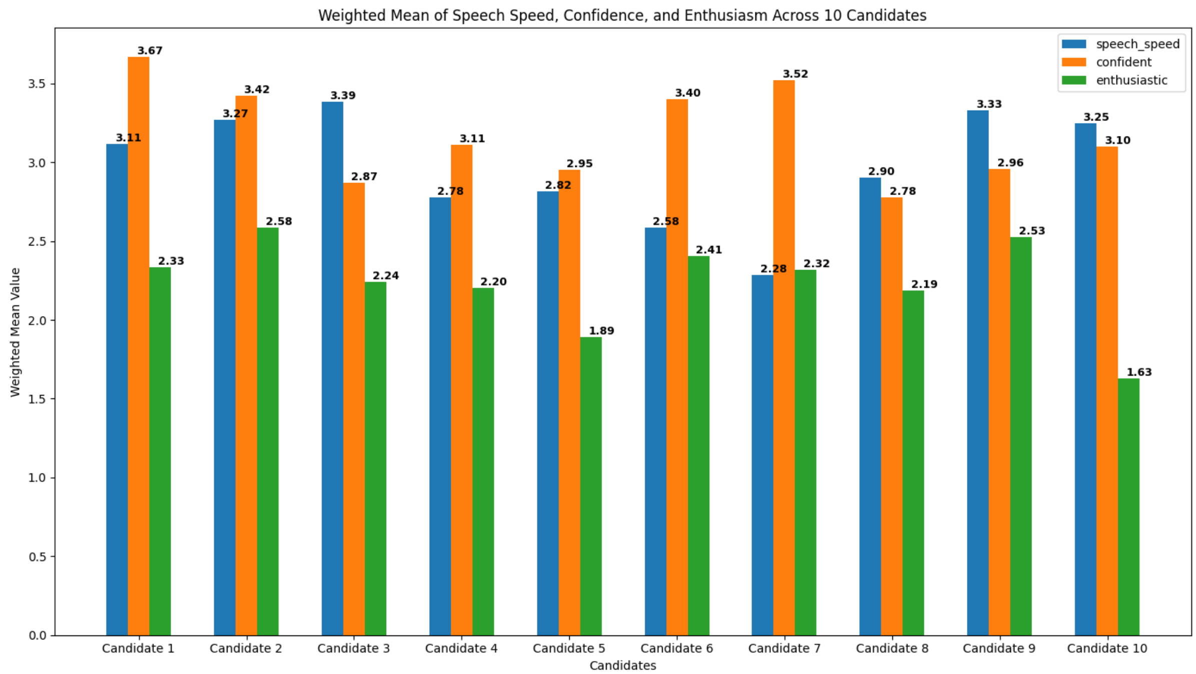
```python
plt.tight_layout()

# Show the plot
plt.show()
```



Plotting weighted Mean of Speech Speed, Confidence, and Enthusiasm Across 10 Candidates

```python
# Define the columns to compare (replace with your column names)
columns_to_compare = ['speech_speed', 'confident', 'enthusiastic']

# Define weights for each column
column_weights = {'speech_speed': 1, 'confident': 5, 'enthusiastic': 5}

# Initialize a dictionary to store weighted means for each column
weighted_column_means = {col: [] for col in columns_to_compare}

# Calculate weighted means for each column across dataframes
for col in columns_to_compare:
    for df in dataframes:
        weighted_mean = df[col].mean() * column_weights[col]
        weighted_column_means[col].append(weighted_mean)

# Define labels for the dataframes (optional, replace with actual labels)
labels = ['Candidate 1', 'Candidate 2', 'Candidate 3', 'Candidate 4', 'Candidate 5',
          'Candidate 6', 'Candidate 7', 'Candidate 8', 'Candidate 9', 'Candidate
          ↪ 10']
```

```python

# Create an array of x positions for the bars
x = np.arange(len(labels))

# Define the width of each bar group
bar_width = 0.2

# Create subplots for each column
fig, axs = plt.subplots(figsize=(14, 8))

# Create bars for each column
for i, col in enumerate(columns_to_compare):
    # Create bars for the current column
    bars = axs.bar(x + i * bar_width, weighted_column_means[col], bar_width, label=col)

# Set the x-axis labels
axs.set_xticks(x + 0.2)
axs.set_xticklabels(labels)

# Add labels, title, and legend
axs.set_xlabel('Candidates')
axs.set_ylabel('Weighted Mean Value')
axs.set_title('Weighted Mean of Speech Speed, Confidence, and Enthusiasm Across 10 Candidates')
axs.legend()

# Add values above the bars
for i in range(len(labels)):
    for col in columns_to_compare:
        height = weighted_column_means[col][i]
        axs.annotate(f'{height:.2f}', (x[i] + bar_width / 2 + columns_to_compare.index(col) * bar_width, height),
                     ha='center', va='bottom', fontsize=9, color='black', weight='bold')

# Adjust subplot layout
plt.tight_layout()

# Show the plot
plt.show()
```



Enthusiastic and confident score of Candidate 2 is quite high than it's peers.

4 Result

Although Candidate 2 and Candidate 9 has a tough competition, Candidate 2 does not perform good on the gaze dataset.

Candidate 9, Alexander Smith is a first-year MBA student at IIM Lucknow with a background in Agricultural Engineering. He has a B.Tech in Agriculture Engineering and an M.Tech in Food Process Engineering. During his M.Tech, he co-founded an Agritech startup, which he ran for 10 months, generating approximately 1,000 units of surplus revenue. He also worked in an Agritech farm, leading a project that applied remote sensing IoT and artificial intelligence to agriculture and allied sectors. Alexander is passionate about entrepreneurship, with a particular interest in business development and strategy. He is dedicated to developing a deep learning algorithm for analyzing facial expressions during online communication, aiming for over 95% accuracy to benefit human resource development and mental health. He believes in the importance of artificial intelligence in shaping the future and is motivated to contribute to such noble causes where AI plays a vital role in societal development.

So, the best Candidate to hire is **Candidate 9**.