In [15]:

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

file_path = './survey_list.xlsx'
df = pd.read_excel(file_path)

df.head()
```

Out[15]:

	Index	Terms & conditions	성별	생년월일	전화번호	선호하는영화장르	현재정신건강관련치료나상담을받고있다.	내게좋은일이일어난다면,나는보통그일로인해크게영향을받곤한다.	내가무언가를잘해냈을때,나는그상태를계속유지하고싶어한다.	나는경기에서이기면보통매우흥분한다.	...	이 영화를 보고 난 후 흥분(Excited) 되거나 긴장되었다. (1-차분하다, 9-흥분되었다).8	다음을 가타내는골라
0	4	✓	남자	1998. 1. 17.	1049468170	액션|코미디	아니오	다소동의	매우동의	매우동의	...	3	(Conte
1	5	✓	남자	2001. 12. 23	1051350406	코미디|드라마	아니오	다소동의	다소동의	매우동의	...	4	(Dep
2	6	✓	남자	1224. 8. 1.	1039471109	액션|코미디|판타지	아니오	매우동의	다소동의	다소동의	...	4	(
3	7	✓	여자	2004. 1. 25.	1036149407	액션	아니오	다소	다소	다소동	...	4	(

								동의	동의	의하지 않음				
4	8		✓	여자	2003. 9. 19.	1050337288	액션 | 드라마 | 판타지 | 코미디	아니오	다소 동의	다소 동의	매우 동의	...	7	(Dep

5 rows × 71 columns

In [16]:
```
plt.rc('font', family='Malgun Gothic')
pd.set_option('mode.chained_assignment', None)
```

In [17]:
```
df = df.iloc[:, 2:-4]

df.head()
```

Out[17]:

	성별	생년월일	전화번호	선호하는영화장르	현재정신건강관련치료나상담을받고있다.	내게좋은일이일어난다면,나는보통그일로인해크게영향을받곤한다.	내가무언가를잘해냈을때,나는그상태를계속유지하고싶어한다.	나는경기에서이기면보통매우흥분한다.	내가원하는어떤것을얻게되면,나는흔히흥분하고기운이넘친다.	내가좋아하는어떤것을볼기회를갖게되면나는곧바로흥분한다.	...	이 영화를 보고 난 후 흥분(Excited)되거나 긴장되었다. (1-차분하다, 9-흥분되었다).7	다음 중 감정을 가장 잘 나타내는 단어를 골라 주세요..7	이 영화를 이 적 있습 까?.
0	남자	1998. 1. 17.	1049468170	액션|코미디	아니오	다소동의	매우동의	매우동의	매우동의	매우동의	...	4	즐거움(Pleasure)	아니오
1	남자	2001. 12. 23	1051350406	코미디|드라마	아니오	다소동의	다소동의	매우동의	매우동의	다소동의	...	3	행복(Happiness)	아니오
2	남자	1224. 8. 1.	1039471109	액션|코미디|판타지	아니오	매우동의	다소동의	다소동의	다소동의	다소동의	...	3	행복(Happiness)	아니오
3	여자	2004. 1. 25.	1036149407	액션	아니오	다소동의	다소동의	다소동의	다소동의	다소동의	...	1	평온(Calmness)	아니오

							하지 않음	하지 않음	하지 않음					
4	여자	2003. 9. 19.	1050337288	액션 | 드라마 | 판타지 | 코미디	아니오	다소 동의	다소 동의	매우 동의	다소 동의	다소 동의하지 않음	...	2	평온 (Calmness)	아니오

5 rows × 65 columns

In [18]:
```python
import pandas as pd

def rename_columns(df):
    new_columns = {}
    movie_count = 0
    for col in df.columns:
        if "이 영화를 이전에 본 적이 있습니까" in col:
            new_columns[col] = f"Watched_Before_{movie_count}"
            movie_count += 1
        elif "이 영화를 보고난 후 현재 기분이 즐겁고 행복하다." in col:
            new_columns[col] = f"Valence_After_Movie_{movie_count - 1}"
        elif "이 영화를 보고난 후 흥분(Excited)되거나 긴장되었다." in col:
            new_columns[col] = f"Arousal_After_Movie_{movie_count - 1}"
        elif "다음 중 감정을 가장 잘 나타내는 단어를 골라 주세요." in col:
            new_columns[col] = f"Emotion_Keyword_{movie_count - 1}"
        else:
            new_columns[col] = col
    return df.rename(columns=new_columns)

df = rename_columns(df)

df.head()
```

Out[18]:

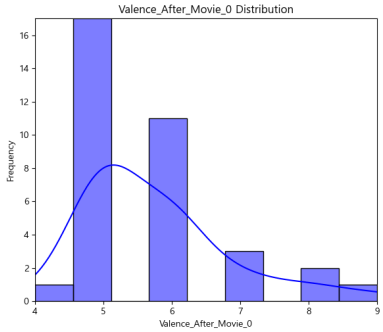
	성별	생년월일	전화번호	선호하는영화장르	현재정신건강관련치료나상담을받고있다.	내게좋은일이일어난다면,나는보통그일로인해크게영향을받곤한다.	내가무언가를잘해냈을때,나는그상태를계속유지하고싶어한다.	나는경기에서이기면보통매우흥분한다.	내가원하는어떤것을얻게되면,나는흔히흥분하고기운이넘친다.	내가좋아하는어떤것을볼기회를갖게되면나는곧바로흥분한다.	...	Arousal_After_Movie_7	En
0	남자	1998. 1. 17.	1049468170	액션|코미디	아니오	다소동의	매우동의	매우동의	매우동의	매우동의	...	4	
1	남자	2001. 12. 23	1051350406	코미디|드라마	아니오	다소동의	다소동의	매우동의	매우동의	다소동의	...	3	
2	남자	1224. 8. 1.	1039471109	액션|코미디|판타지	아니오	매우동의	다소동의	다소동의	다소동의	다소동의	...	3	
3	여자	2004. 1. 25.	1036149407	액션	아니오	다소동의	다소동의	다소동의	다소동의	다소동의	...	1	

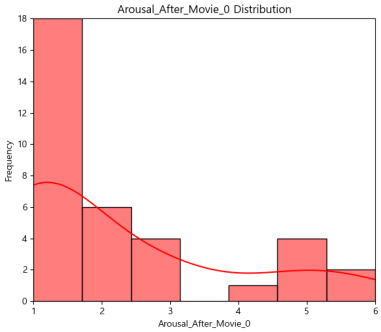
								하지않음	하지않음	하지않음		
4	여자	2003. 9. 19.	1050337288	액션|드라마|판타지|코미디	아니오	다소동의	다소동의	매우동의	다소동의	다소동의하지않음	...	2

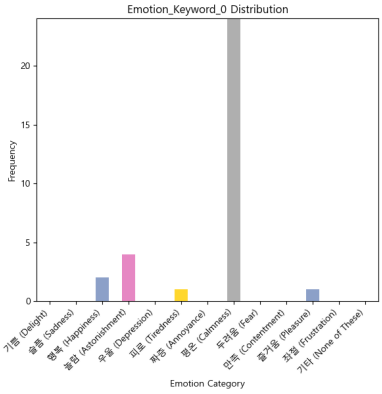
5 rows × 65 columns

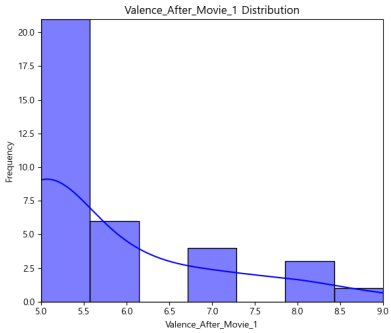
In [19]:
```
# Create a list of movies (0 to 9)
movies = range(10)

# Create a figure to hold multiple subplots
fig, axes = plt.subplots(10, 3, figsize=(18, 60))

# Define the emotion categories order (for consistent labeling if needed)
emotion_order = ['기쁨 (Delight)', '슬픔 (Sadness)', '행복 (Happiness)', '놀람 (

# Loop through each movie
for i, movie in enumerate(movies):
    valence_col = f'Valence_After_Movie_{movie}'
    arousal_col = f'Arousal_After_Movie_{movie}'
    emotion_col = f'Emotion_Keyword_{movie}'

    # Plot for Valence
    sns.histplot(df[valence_col], kde=True, ax=axes[i, 0], color='blue')
    axes[i, 0].set_title(f'Valence_After_Movie_{movie} Distribution')
    axes[i, 0].set_xlabel(f'Valence_After_Movie_{movie}')
    axes[i, 0].set_ylabel('Frequency')
    axes[i, 0].set_xlim(df[valence_col].min(), df[valence_col].max())  # Fix x-a
    axes[i, 0].set_ylim(0, df[valence_col].value_counts().max())  # Fix y-axis r

    # Plot for Arousal
    sns.histplot(df[arousal_col], kde=True, ax=axes[i, 1], color='red')
    axes[i, 1].set_title(f'Arousal_After_Movie_{movie} Distribution')
    axes[i, 1].set_xlabel(f'Arousal_After_Movie_{movie}')
    axes[i, 1].set_ylabel('Frequency')
    axes[i, 1].set_xlim(df[arousal_col].min(), df[arousal_col].max())  # Fix x-a
    axes[i, 1].set_ylim(0, df[arousal_col].value_counts().max())  # Fix y-axis r

    # Plot for Emotion
    emotion_counts = df[emotion_col].value_counts().reindex(emotion_order, fill_
    emotion_counts.plot(kind='bar', stacked=True, color=sns.color_palette("Set2"
    axes[i, 2].set_title(f'Emotion_Keyword_{movie} Distribution')
    axes[i, 2].set_xlabel('Emotion Category')
```

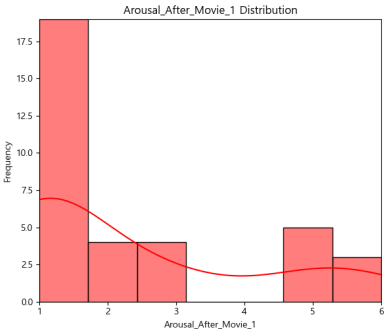
```
    axes[i, 2].set_ylabel('Frequency')
    axes[i, 2].set_ylim(0, emotion_counts.max())  # Fix y-axis range
    axes[i, 2].set_xticklabels(emotion_counts.index, rotation=45, ha="right")  #

# Adjust layout to avoid overlap
plt.tight_layout()
plt.show()
```

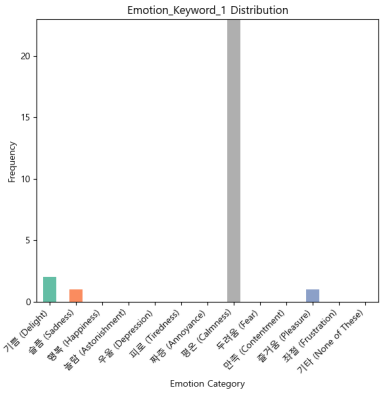

Valence_After_Movie_0 Distribution
Frequency
Valence_After_Movie_0

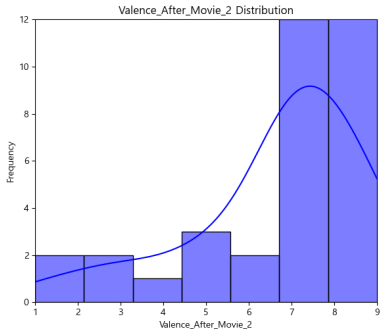

Arousal_After_Movie_0 Distribution
Frequency
Arousal_After_Movie_0

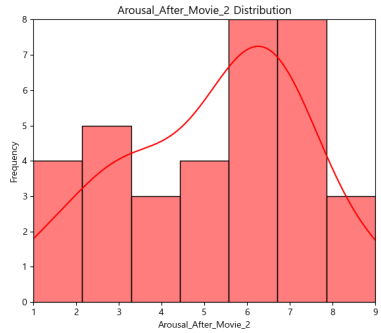

Emotion_Keyword_0 Distribution
Frequency
기쁨 (Delight)
슬픔 (Sadness)
행복 (Happiness)
놀람 (Astonishment)
우울 (Depression)
피로 (Tiredness)
짜증 (Annoyance)
평온 (Calmness)
두려움 (Fear)
만족 (Contentment)
즐거움 (Pleasure)
좌절 (Frustration)
기타 (None of These)
Emotion Category

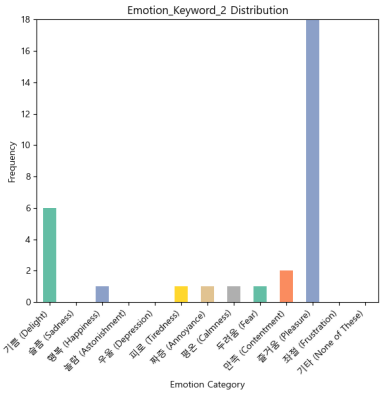

Valence_After_Movie_1 Distribution
Frequency
Valence_After_Movie_1

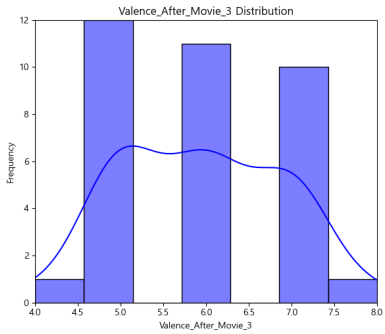

Arousal_After_Movie_1 Distribution
Frequency
Arousal_After_Movie_1

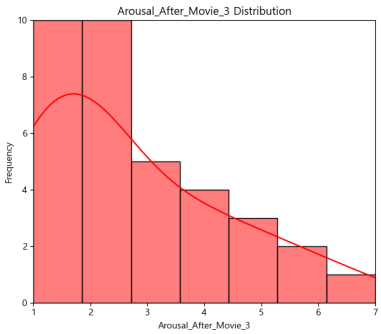

Emotion_Keyword_1 Distribution
Frequency
기쁨 (Delight)
슬픔 (Sadness)
행복 (Happiness)
놀람 (Astonishment)
우울 (Depression)
피로 (Tiredness)
짜증 (Annoyance)
평온 (Calmness)
두려움 (Fear)
만족 (Contentment)
즐거움 (Pleasure)
좌절 (Frustration)
기타 (None of These)
Emotion Category

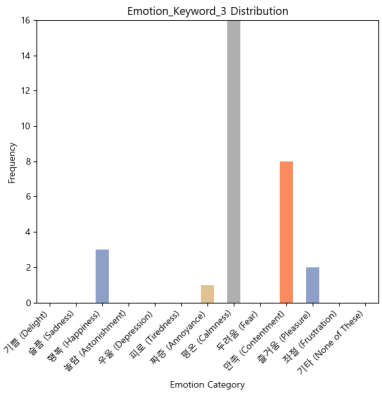

Valence_After_Movie_2 Distribution
Frequency
Valence_After_Movie_2

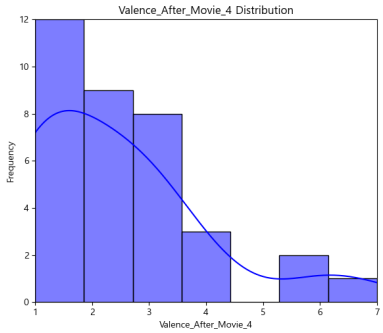

Arousal_After_Movie_2 Distribution
Frequency
Arousal_After_Movie_2

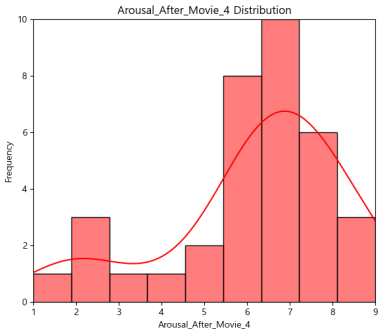

Emotion_Keyword_2 Distribution
Frequency
기쁨 (Delight)
슬픔 (Sadness)
행복 (Happiness)
놀람 (Astonishment)
우울 (Depression)
피로 (Tiredness)
짜증 (Annoyance)
평온 (Calmness)
두려움 (Fear)
만족 (Contentment)
즐거움 (Pleasure)
좌절 (Frustration)
기타 (None of These)
Emotion Category

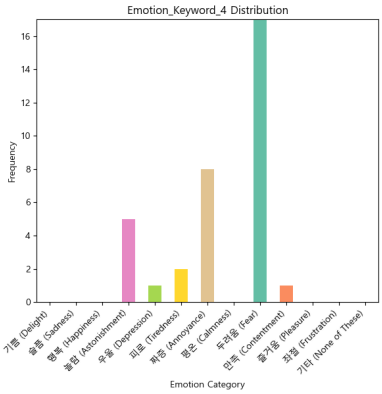

Valence_After_Movie_3 Distribution
Frequency
Valence_After_Movie_3

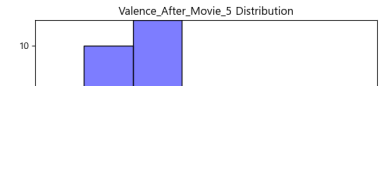

Arousal_After_Movie_3 Distribution
Frequency
Arousal_After_Movie_3

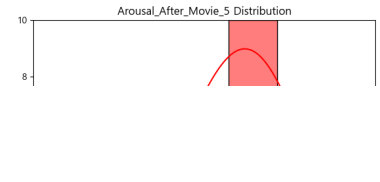

Emotion_Keyword_3 Distribution
Frequency
기쁨 (Delight)
슬픔 (Sadness)
행복 (Happiness)
놀람 (Astonishment)
우울 (Depression)
피로 (Tiredness)
짜증 (Annoyance)
평온 (Calmness)
두려움 (Fear)
만족 (Contentment)
즐거움 (Pleasure)
좌절 (Frustration)
기타 (None of These)
Emotion Category

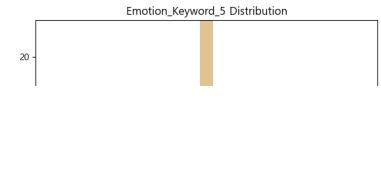

Valence_After_Movie_4 Distribution
Frequency
Valence_After_Movie_4

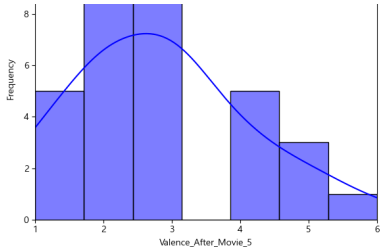

Arousal_After_Movie_4 Distribution
Frequency
Arousal_After_Movie_4

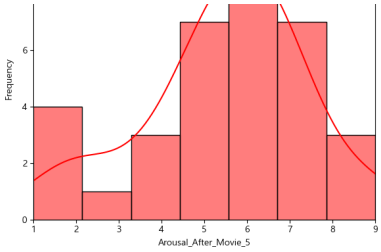

Emotion_Keyword_4 Distribution
Frequency
기쁨 (Delight)
슬픔 (Sadness)
행복 (Happiness)
놀람 (Astonishment)
우울 (Depression)
피로 (Tiredness)
짜증 (Annoyance)
평온 (Calmness)
두려움 (Fear)
만족 (Contentment)
즐거움 (Pleasure)
좌절 (Frustration)
기타 (None of These)
Emotion Category

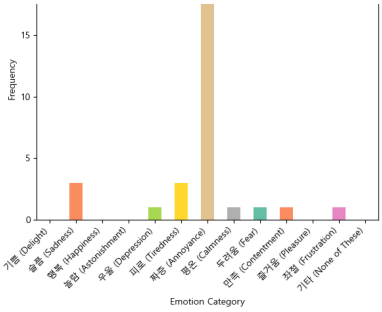

Valence_After_Movie_5 Distribution

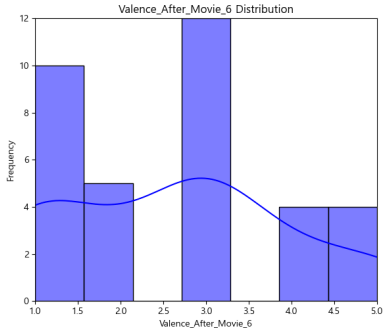

Arousal_After_Movie_5 Distribution

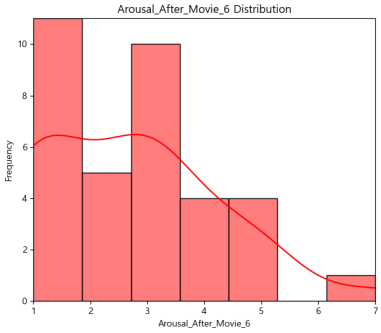

Emotion_Keyword_5 Distribution

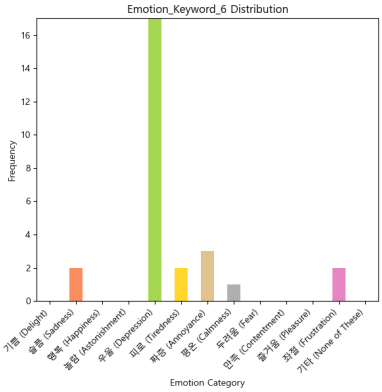


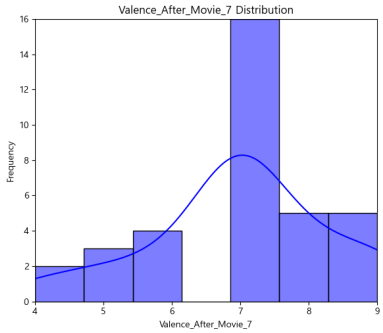


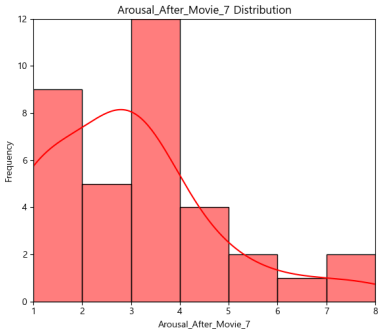


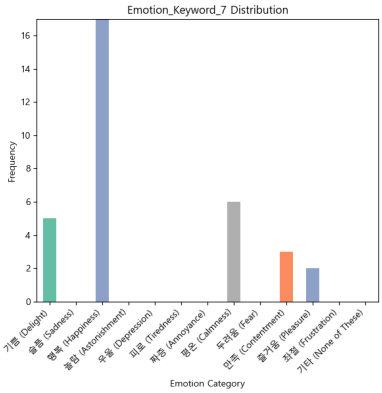


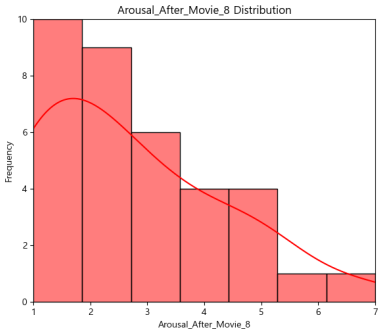


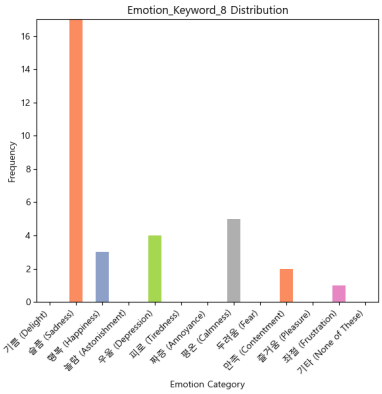


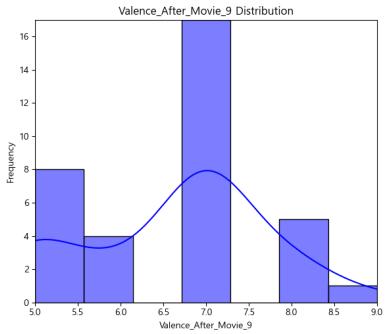


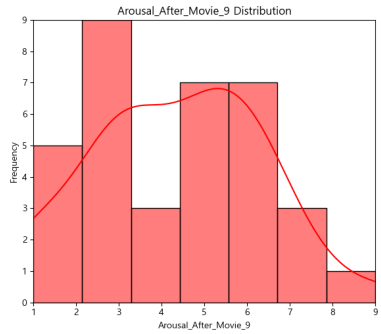


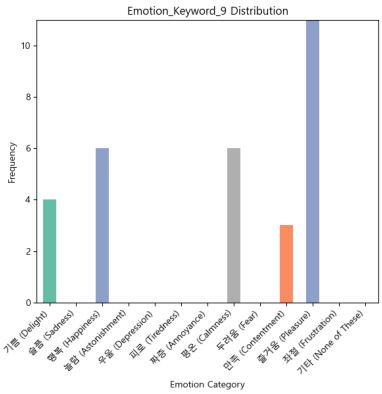








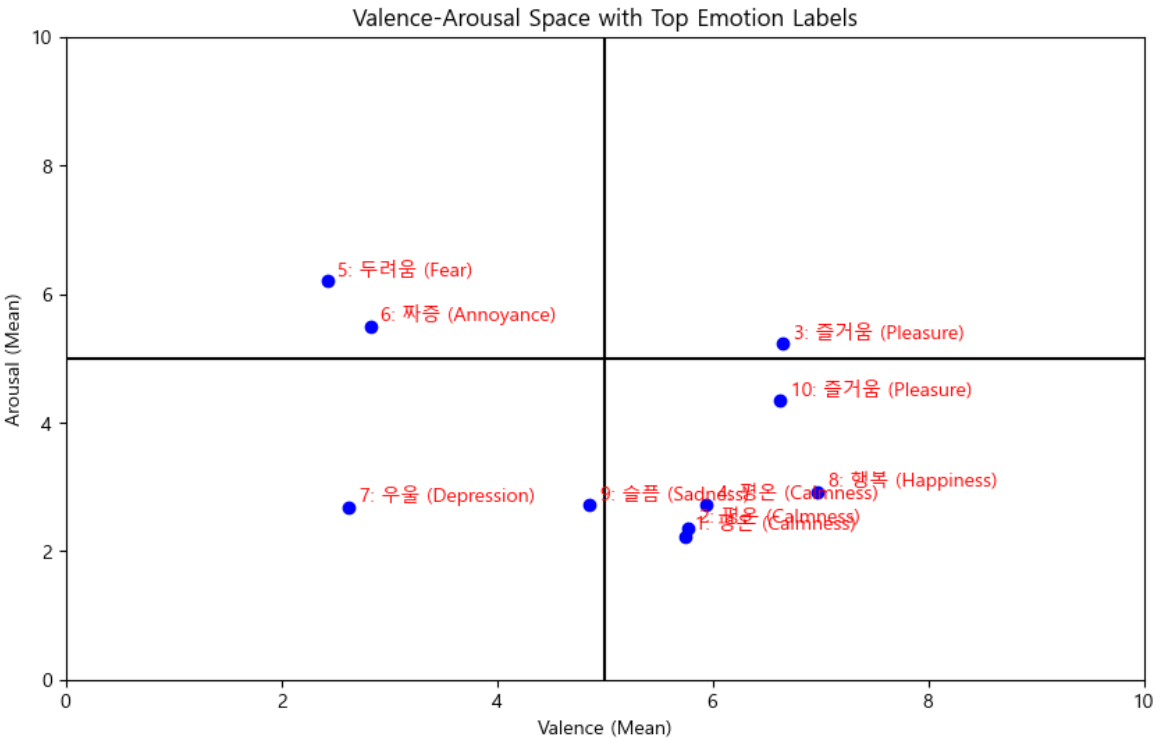




In [20]:
```
# Prepare to store results for table
results = []
```

```python
# Loop through each movie to calculate the statistics
for i, movie in enumerate(range(10)):
    valence_col = f'Valence_After_Movie_{movie}'
    arousal_col = f'Arousal_After_Movie_{movie}'
    emotion_col = f'Emotion_Keyword_{movie}'

    valence_stats = df[valence_col].describe()
    arousal_stats = df[arousal_col].describe()

    # Get the emotion counts for the current movie
    emotion_counts = df[emotion_col].value_counts()
    top_emotions = emotion_counts.head(3)

    # Add the statistics and top emotions to results
    results.append({
        'Movie': movie,
        'Valence Mean': valence_stats['mean'],
        'Valence Var': valence_stats['std']**2,
        'Valence SD': valence_stats['std'],
        'Arousal Mean': arousal_stats['mean'],
        'Arousal Var': arousal_stats['std']**2,
        'Arousal SD': arousal_stats['std'],
        'Top 1 Emotion': top_emotions.index[0] if len(top_emotions) > 0 else "N/
        'Top 2 Emotion': top_emotions.index[1] if len(top_emotions) > 1 else Non
        'Top 3 Emotion': top_emotions.index[2] if len(top_emotions) > 2 else Non
    })

# Convert results to DataFrame for display
results_df = pd.DataFrame(results)

# Plot the Valence-Arousal Space
plt.figure(figsize=(10, 6))

# Iterate over movies and plot points
for i, row in results_df.iterrows():
    valence = row['Valence Mean']
    arousal = row['Arousal Mean']
    emotion = row['Top 1 Emotion']

    # Plot each movie's point
    plt.scatter(valence, arousal, color='blue', label=f'Movie {i+1}' if i == 0 e

    # Add text label with emotion
    plt.text(valence + 0.1, arousal + 0.1, f'{i+1}: {emotion}', fontsize=10, col

# Add labels and title
plt.title('Valence-Arousal Space with Top Emotion Labels')
plt.xlabel('Valence (Mean)')
plt.ylabel('Arousal (Mean)')

# Set limits for valence and arousal
plt.xlim(0, 10)
plt.ylim(0, 10)

# Remove grid
plt.grid(False)

# Draw quadrant axes
plt.axhline(y=5, color='black', linewidth=1.5)  # Horizontal mid-line
plt.axvline(x=5, color='black', linewidth=1.5)  # Vertical mid-line
```

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



In [ ]:
```
# Calculate IQR for Valence and Arousal (to remove outliers for the entire datas
valence_columns = [f'Valence_After_Movie_{i}' for i in range(10)]  # List of Val
arousal_columns = [f'Arousal_After_Movie_{i}' for i in range(10)]  # List of Aro

# Initialize filtered dataframe as df (initially no filtering)
df_filtered = df.copy()

# Loop through each Valence column to calculate IQR and filter outliers
for valence_col in valence_columns:
    Q1 = df_filtered[valence_col].quantile(0.25)
    Q3 = df_filtered[valence_col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Apply the outlier filter to the entire dataset for Valence
    df_filtered = df_filtered[(df_filtered[valence_col] >= lower_bound) & (df_fi

    # Print the number of rows after filtering for Valence
    print(f"After filtering {valence_col}, number of rows: {df_filtered.shape[0]

# Loop through each Arousal column to calculate IQR and filter outliers
for arousal_col in arousal_columns:
    Q1 = df_filtered[arousal_col].quantile(0.25)
    Q3 = df_filtered[arousal_col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Apply the outlier filter to the entire dataset for Arousal
    df_filtered = df_filtered[(df_filtered[arousal_col] >= lower_bound) & (df_fi
```

```
    # Print the number of rows after filtering for Arousal
    print(f"After filtering {arousal_col}, number of rows: {df_filtered.shape[0]
```

```
After filtering Valence_After_Movie_0, number of rows: 32
After filtering Valence_After_Movie_1, number of rows: 31
After filtering Valence_After_Movie_2, number of rows: 26
After filtering Valence_After_Movie_3, number of rows: 26
After filtering Valence_After_Movie_4, number of rows: 25
After filtering Valence_After_Movie_5, number of rows: 25
After filtering Valence_After_Movie_6, number of rows: 25
After filtering Valence_After_Movie_7, number of rows: 21
After filtering Valence_After_Movie_8, number of rows: 21
After filtering Valence_After_Movie_9, number of rows: 21
After filtering Arousal_After_Movie_0, number of rows: 21
After filtering Arousal_After_Movie_1, number of rows: 21
After filtering Arousal_After_Movie_2, number of rows: 21
After filtering Arousal_After_Movie_3, number of rows: 21
After filtering Arousal_After_Movie_4, number of rows: 16
After filtering Arousal_After_Movie_5, number of rows: 15
After filtering Arousal_After_Movie_6, number of rows: 15
After filtering Arousal_After_Movie_7, number of rows: 11
After filtering Arousal_After_Movie_8, number of rows: 11
After filtering Arousal_After_Movie_9, number of rows: 11
```

In [ ]:

```python
# Prepare to store results for table
results = []

# Loop through each movie to calculate the statistics
for i, movie in enumerate(range(10)):
    valence_col = f'Valence_After_Movie_{movie}'
    arousal_col = f'Arousal_After_Movie_{movie}'
    emotion_col = f'Emotion_Keyword_{movie}'

    valence_stats = df_filtered[valence_col].describe()
    arousal_stats = df_filtered[arousal_col].describe()

    # Get the emotion counts for the current movie
    emotion_counts = df_filtered[emotion_col].value_counts()
    top_emotions = emotion_counts.head(3)

    # Add the statistics and top emotions to results
    results.append({
        'Movie': movie,
        'Valence Mean': valence_stats['mean'],
        'Valence Var': valence_stats['std']**2,
        'Valence SD': valence_stats['std'],
        'Arousal Mean': arousal_stats['mean'],
        'Arousal Var': arousal_stats['std']**2,
        'Arousal SD': arousal_stats['std'],
        'Top 1 Emotion': top_emotions.index[0],
        'Top 2 Emotion': top_emotions.index[1] if len(top_emotions) > 1 else Non
        'Top 3 Emotion': top_emotions.index[2] if len(top_emotions) > 2 else Non
    })

# Convert results to DataFrame for display
results_df = pd.DataFrame(results)

# Display the results as a table in the notebook
print(results_df)
```

	Movie	Valence Mean	Valence Var	Valence SD	Arousal Mean	Arousal Var	\
0	0	5.545455	0.472727	0.687552	2.545455	3.072727	
1	1	5.636364	0.654545	0.809040	3.727273	4.018182	
2	2	7.272727	0.818182	0.904534	5.636364	2.854545	
3	3	6.272727	0.818182	0.904534	3.545455	2.272727	
4	4	2.272727	1.218182	1.103713	6.909091	0.490909	
5	5	3.272727	1.018182	1.009050	5.727273	0.618182	
6	6	2.909091	1.690909	1.300350	3.363636	3.054545	
7	7	6.818182	0.763636	0.873863	3.545455	0.672727	
8	8	5.000000	2.400000	1.549193	3.181818	2.363636	
9	9	6.909091	0.290909	0.539360	5.090909	1.690909	

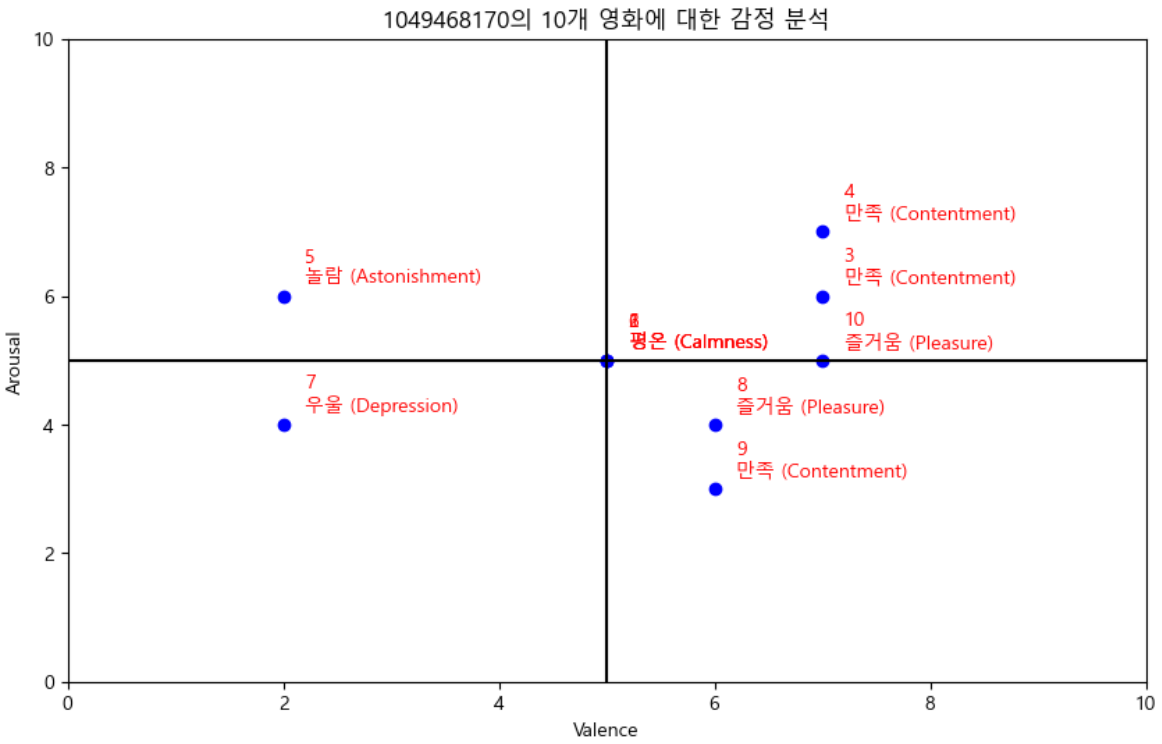
	Arousal SD	Top 1 Emotion	Top 2 Emotion	\
0	1.752920	평온 (Calmness)	행복 (Happiness)	
1	2.004540	평온 (Calmness)	해당하는 감정이 없음(None of These)	
2	1.689540	즐거움 (Pleasure)	기쁨 (Delight)	
3	1.507557	평온 (Calmness)	만족 (Contentment)	
4	0.700649	두려움 (Fear)	짜증 (Annoyance)	
5	0.786245	짜증 (Annoyance)	평온 (Calmness)	
6	1.747726	우울 (Depression)	해당하는 감정이 없음(None of These)	
7	0.820200	행복 (Happiness)	만족 (Contentment)	
8	1.537412	슬픔 (Sadness)	우울 (Depression)	
9	1.300350	행복 (Happiness)	즐거움 (Pleasure)	

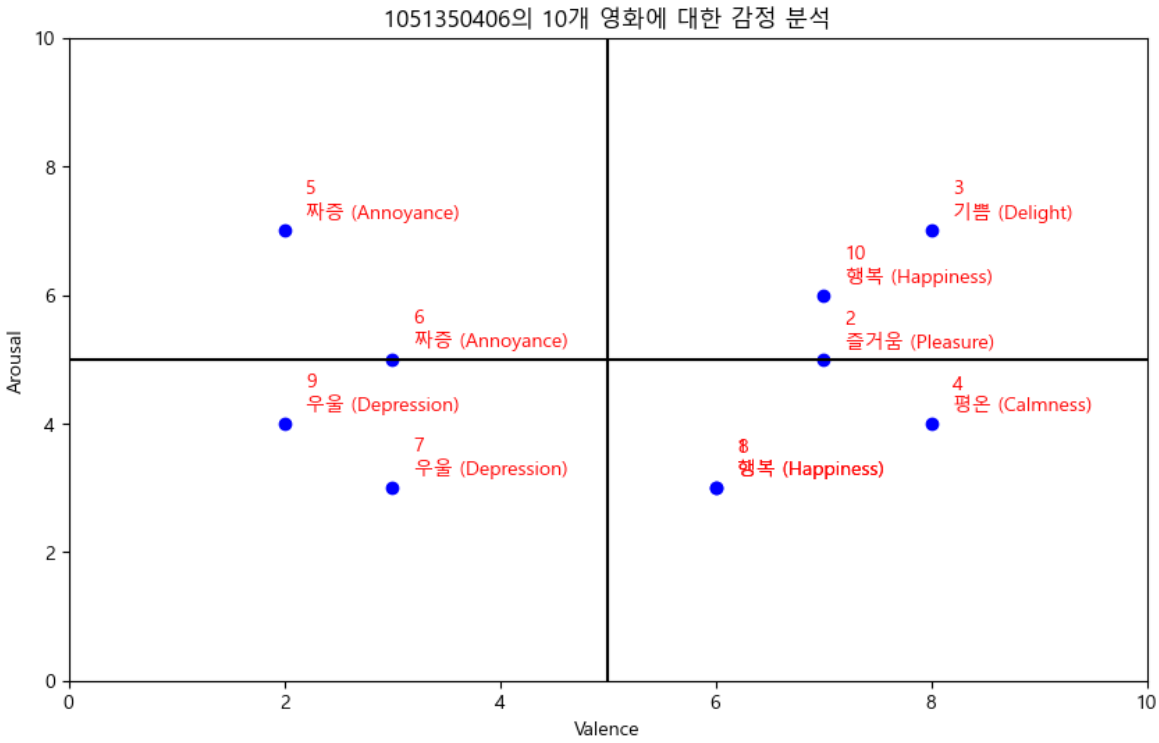
	Top 3 Emotion
0	해당하는 감정이 없음(None of These)
1	즐거움 (Pleasure)
2	만족 (Contentment)
3	행복 (Happiness)
4	놀람 (Astonishment)
5	우울 (Depression)
6	슬픔 (Sadness)
7	즐거움 (Pleasure)
8	행복 (Happiness)
9	기쁨 (Delight)

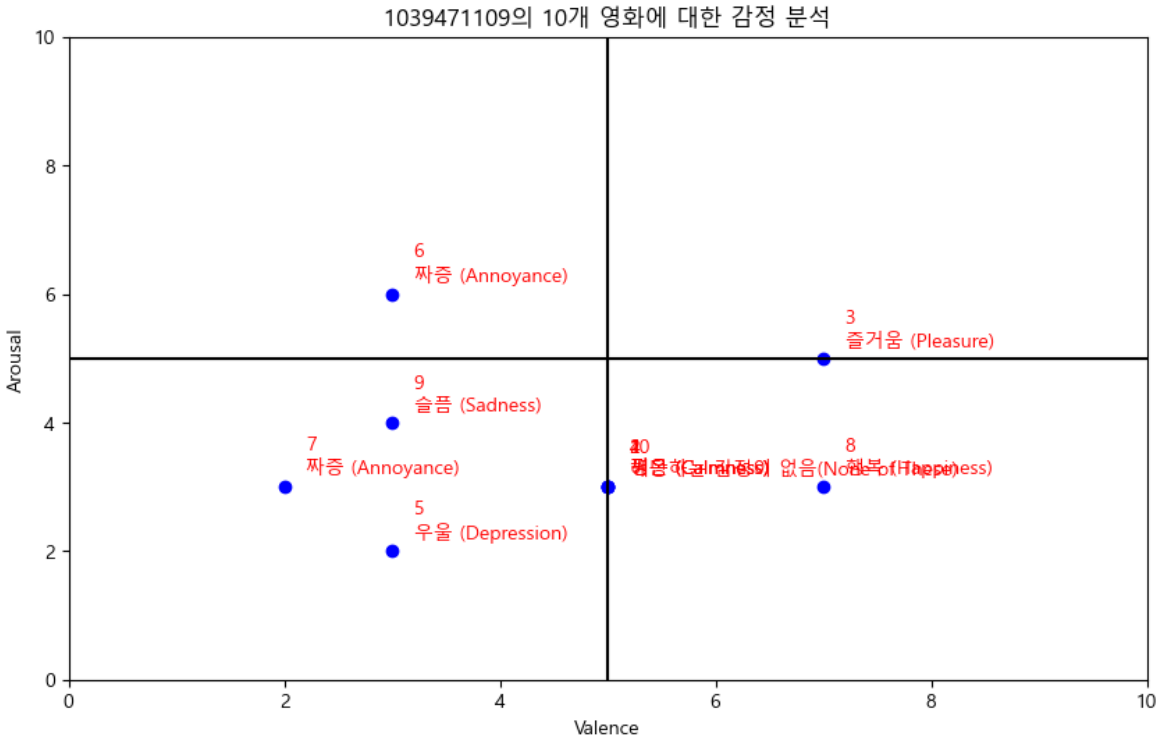
In [ ]:

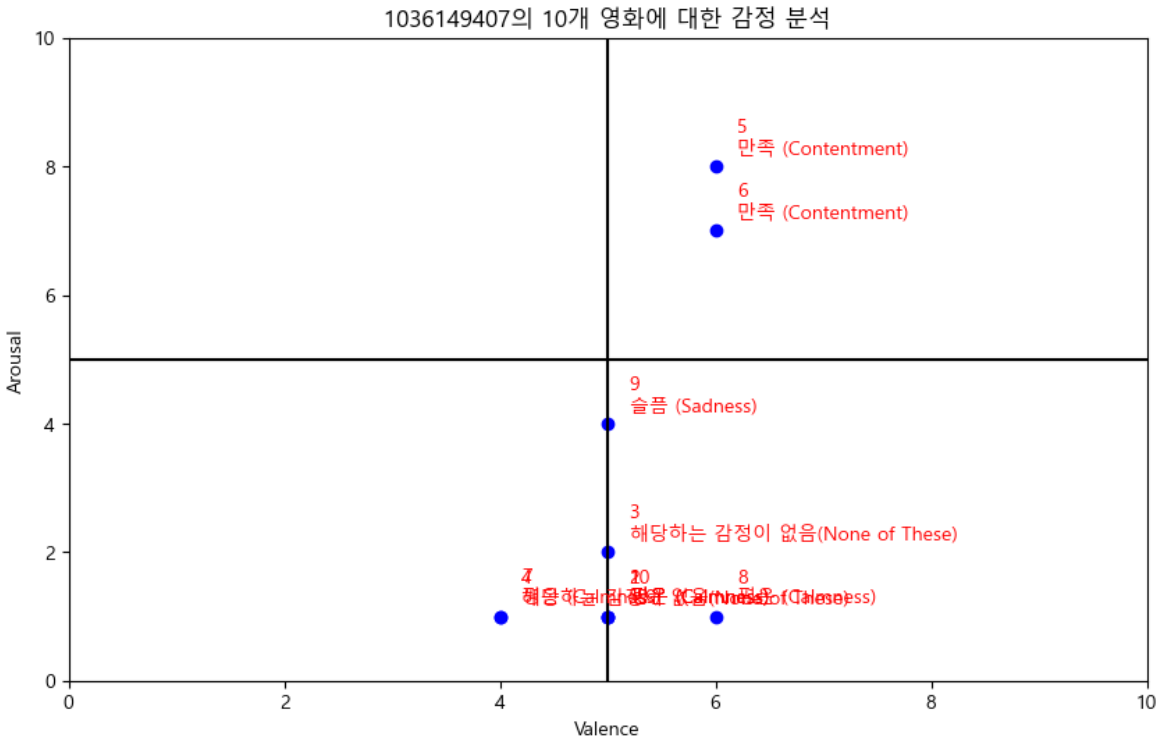
In [22]:
```python
# Selecting Valence, Arousal, and Emotion columns
valence_columns = [col for col in df.columns if 'Valence_After_Movie_' in col]
arousal_columns = [col for col in df.columns if 'Arousal_After_Movie_' in col]
emotion_columns = [col for col in df.columns if 'Emotion_Keyword_' in col]  # Em

# Select necessary columns for the analysis
df_selected = df[['전화번호'] + valence_columns + arousal_columns + emotion_colu

# Add valence, arousal, and emotion for each movie
for i in range(10):
    df_selected[f'valence_{i+1}'] = df_selected[valence_columns[i]].copy()
    df_selected[f'arousal_{i+1}'] = df_selected[arousal_columns[i]].copy()
    df_selected[f'emotion_{i+1}'] = df_selected[emotion_columns[i]].copy()

# Get unique participants
people = df_selected['전화번호'].unique()

# Plot for each person
for person in people:
    try:
        # Extract data for the person
        person_data = df_selected[df_selected['전화번호'] == person]
```

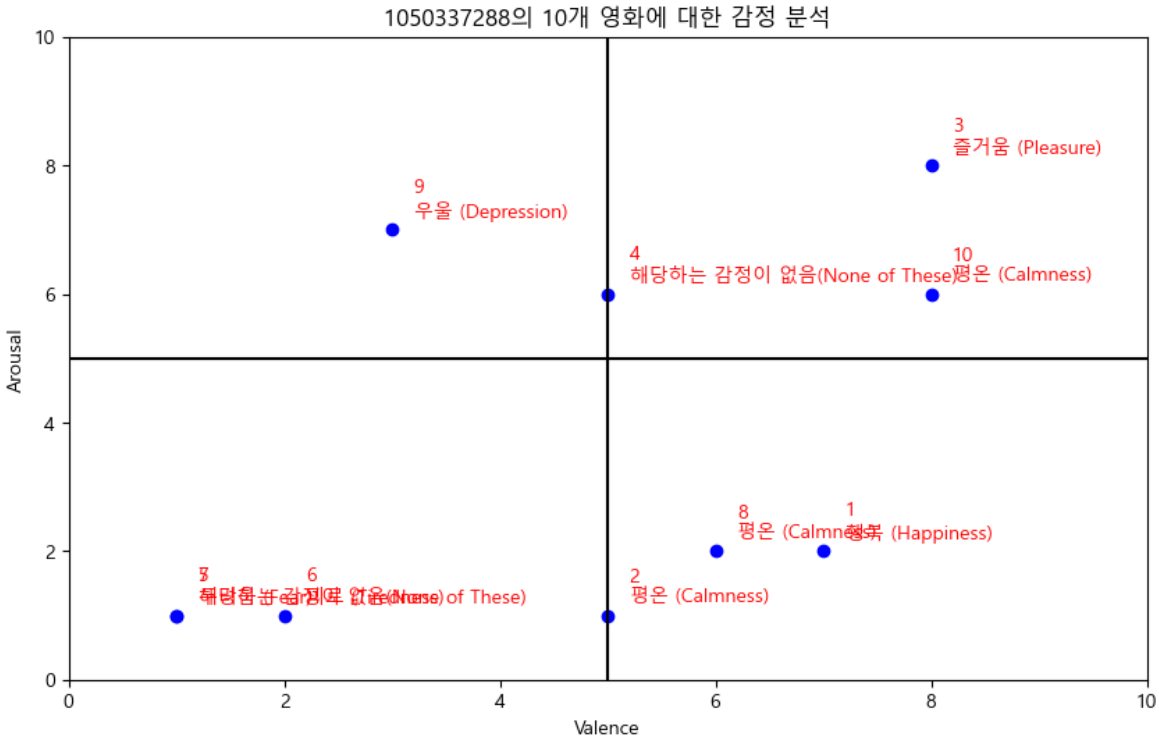
```
        plt.figure(figsize=(10, 6))

        # Plot each movie's valence and arousal
        for i in range(10):
            valence = person_data[f'valence_{i+1}'].values[0]
            arousal = person_data[f'arousal_{i+1}'].values[0]
            emotion = person_data[f'emotion_{i+1}'].values[0]  # 감정 키워드

            # Plot each movie's point with labels
            plt.scatter(valence, arousal, label=f'Movie {i+1}', color='blue')
            plt.text(valence + 0.2, arousal + 0.2, f'{i+1}\n{emotion}', fontsize

        # Add labels and title
        plt.title(f'{person}의 10개 영화에 대한 감정 분석')
        plt.xlabel('Valence')
        plt.ylabel('Arousal')

        # Set limits and remove grid
        plt.xlim(0, 10)
        plt.ylim(0, 10)
        plt.grid(False)

        # Draw quadrant axes
        plt.axhline(y=5, color='black', linewidth=1.5)  # Horizontal mid-line
        plt.axvline(x=5, color='black', linewidth=1.5)  # Vertical mid-line

        plt.show()

    except Exception as e:
        print(f"Error while processing {person}: {e}")
        continue
```

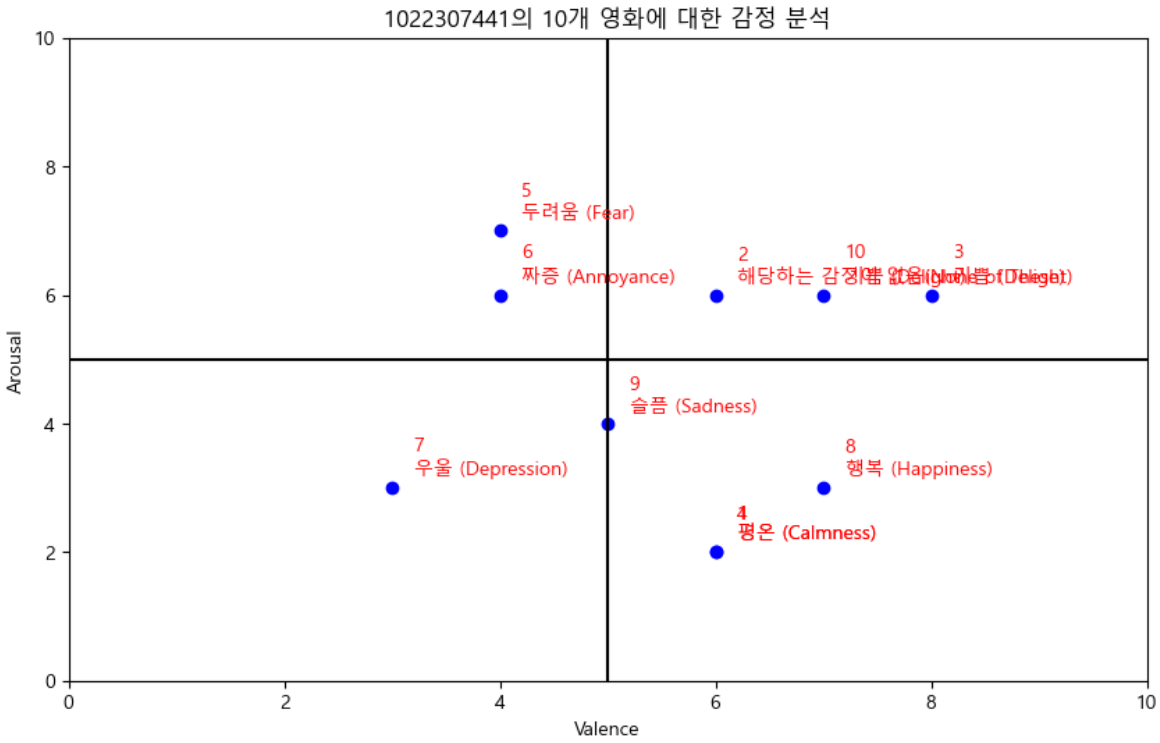


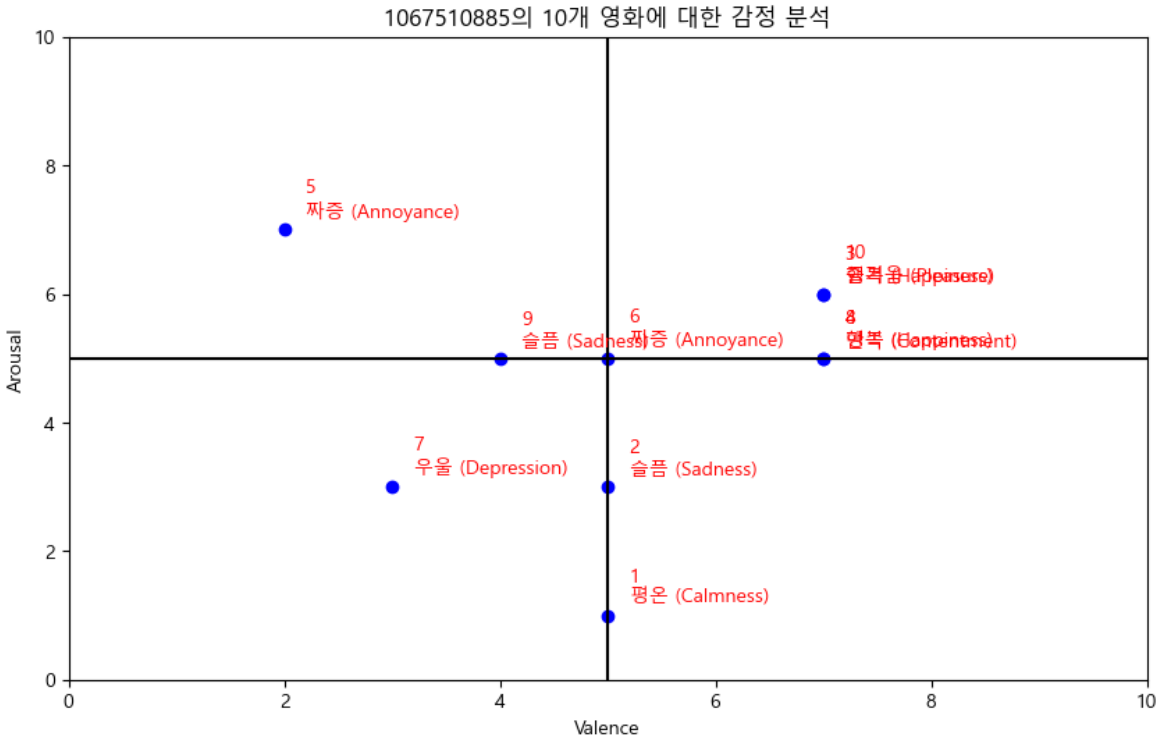

1051350406의 10개 영화에 대한 감정 분석
Arousal
Valence
5
짜증 (Annoyance)
3
기쁨 (Delight)
10
행복 (Happiness)
6
짜증 (Annoyance)
2
즐거움 (Pleasure)
9
우울 (Depression)
4
평온 (Calmness)
7
우울 (Depression)
8
행복 (Happiness)

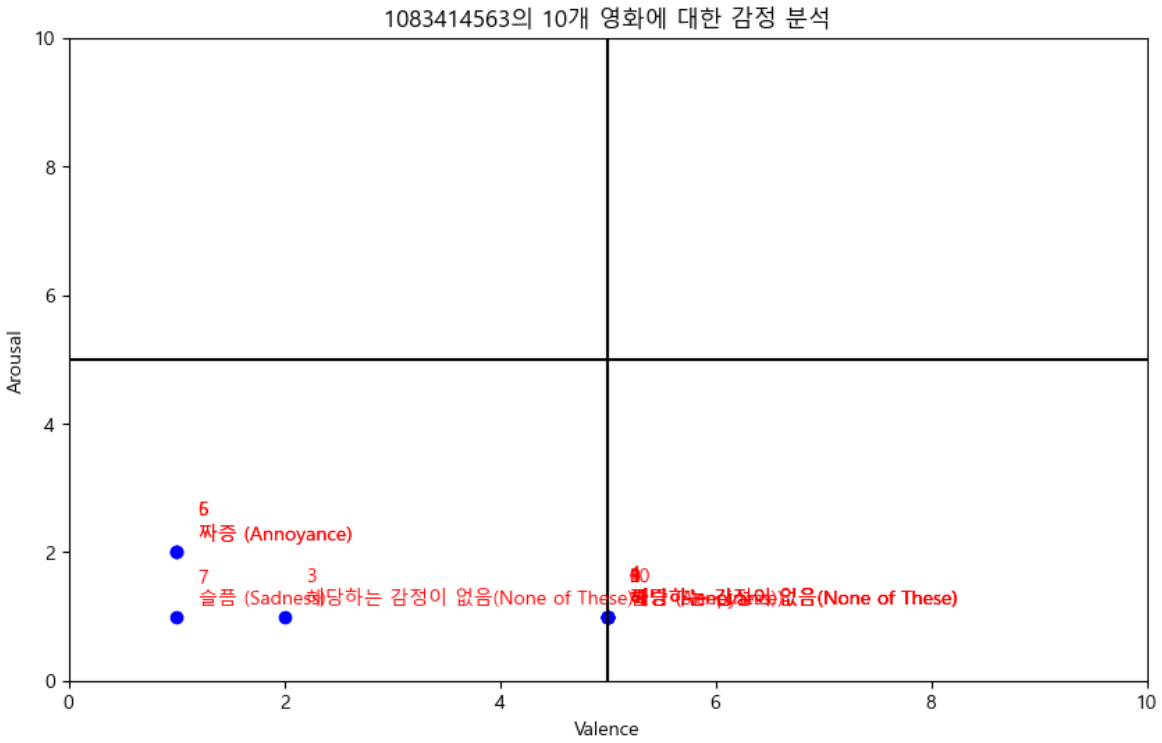

1039471109의 10개 영화에 대한 감정 분석
Arousal
Valence
6
짜증 (Annoyance)
3
즐거움 (Pleasure)
9
슬픔 (Sadness)
7
짜증 (Annoyance)
8
5
우울 (Depression)

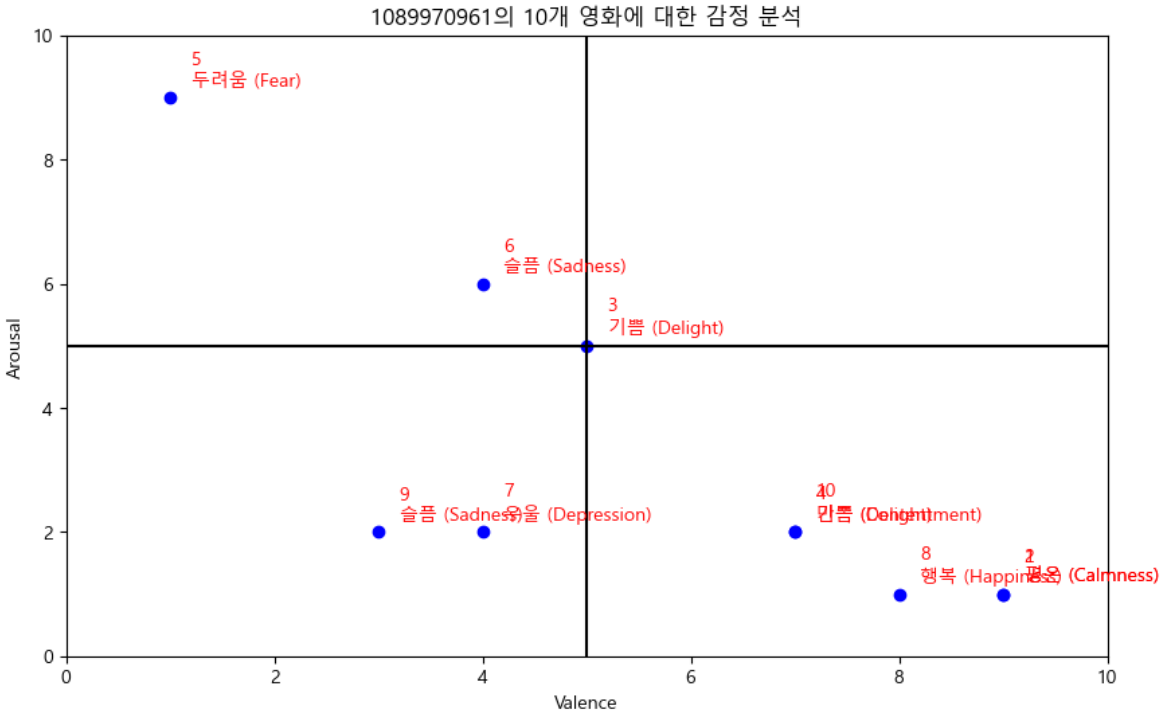

1036149407의 10개 영화에 대한 감정 분석
Arousal
Valence
5
만족 (Contentment)
6
만족 (Contentment)
9
슬픔 (Sadness)
3
해당하는 감정이 없음(None of These)

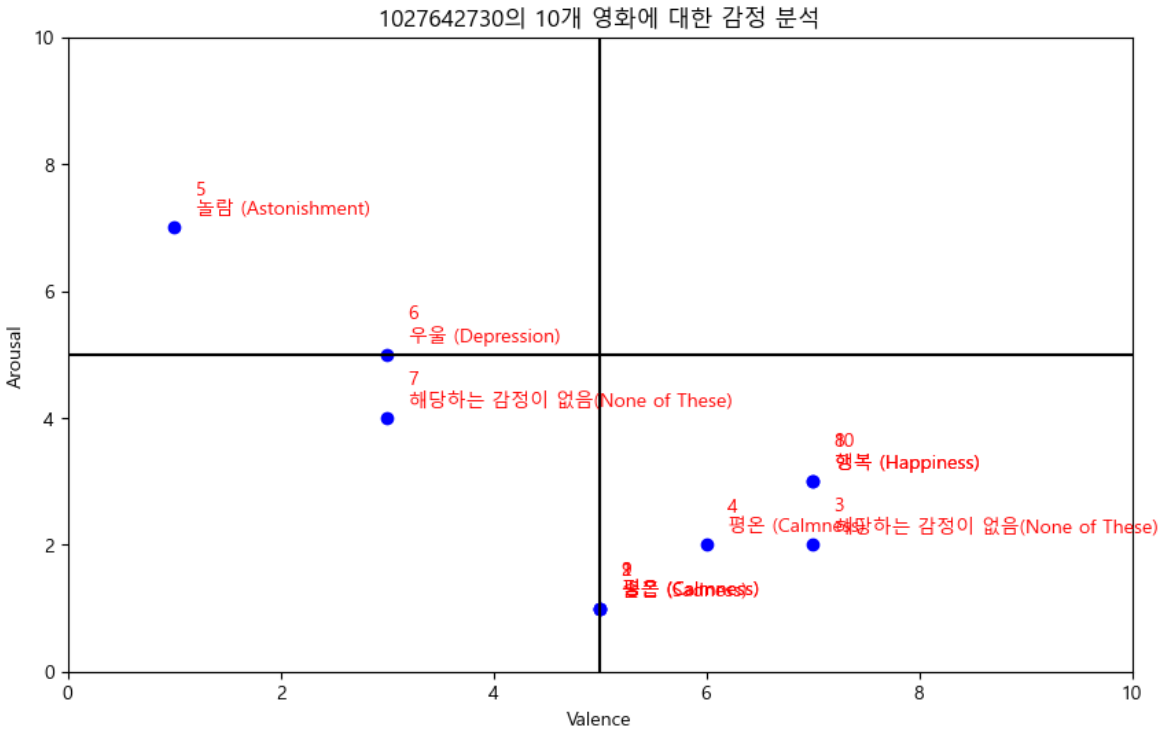

1050337288의 10개 영화에 대한 감정 분석
Arousal
Valence
3
즐거움 (Pleasure)
9
우울 (Depression)
4
해당하는 감정이 없음(None of These)
10
평온 (Calmness)
8
1
2
평온 (Calmness)

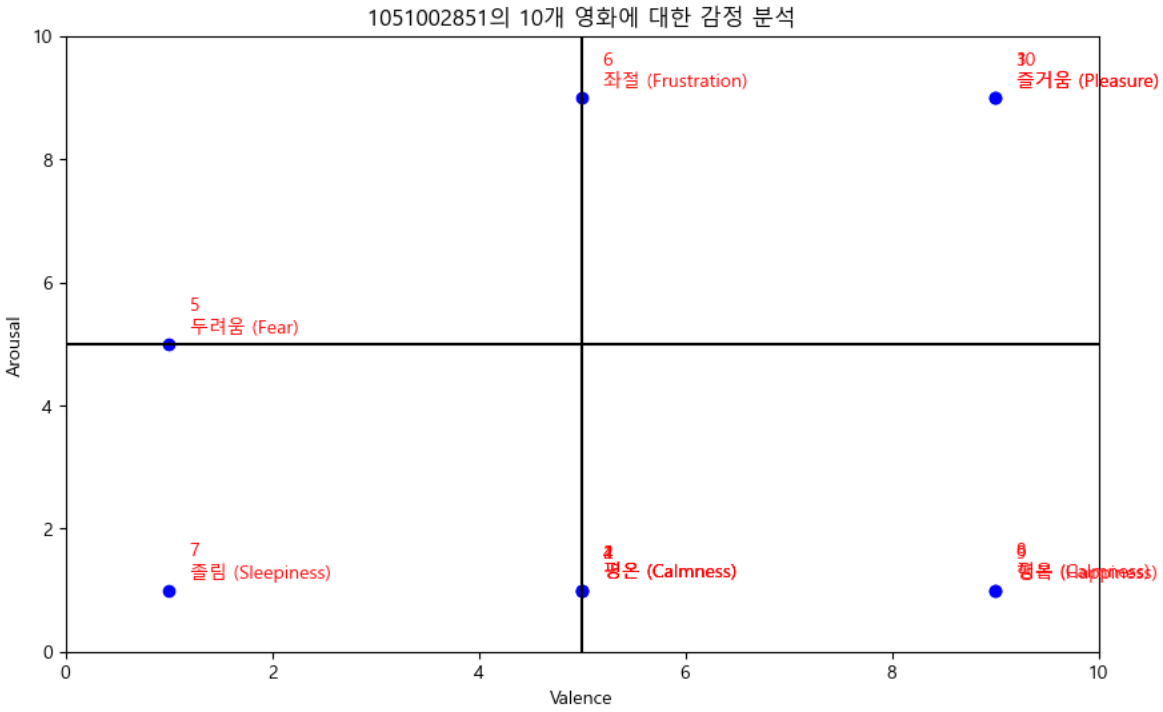

1022307441의 10개 영화에 대한 감정 분석
Arousal
Valence
5
두려움 (Fear)
6
짜증 (Annoyance)
9
슬픔 (Sadness)
7
우울 (Depression)
8
행복 (Happiness)
4
평온 (Calmness)

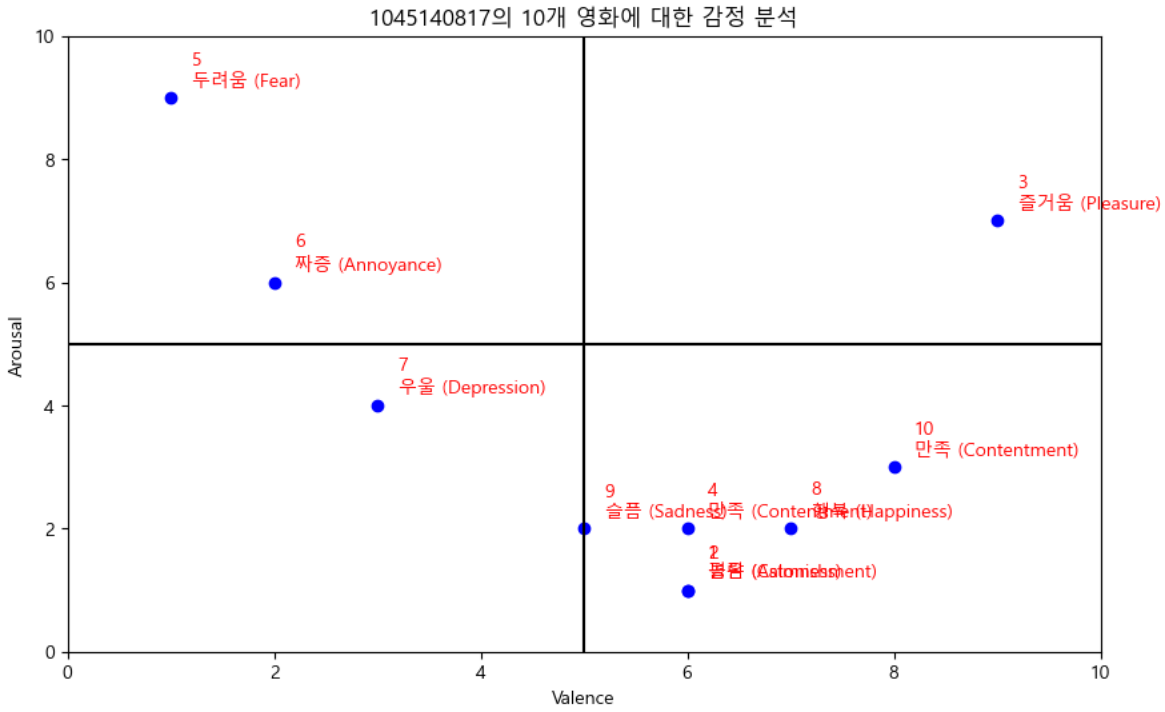

1067510885의 10개 영화에 대한 감정 분석
Arousal
Valence
5
짜증 (Annoyance)
9
슬픔 (Sadness)
6
짜증 (Annoyance)
7
우울 (Depression)
2
슬픔 (Sadness)
1
평온 (Calmness)

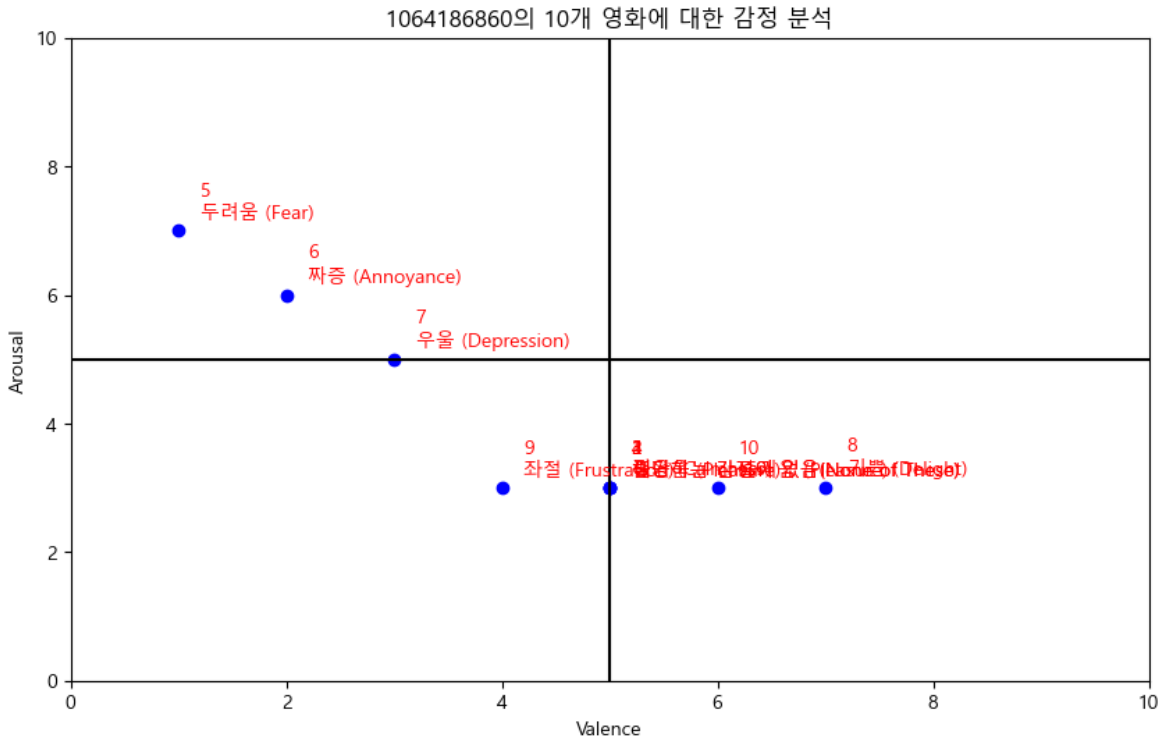

1083414563의 10개 영화에 대한 감정 분석
Arousal
Valence
6
짜증 (Annoyance)
7
슬픔 (Sadness)
3
해당하는 감정이 없음(None of These)
10
해당하는 감정이 없음(None of These)

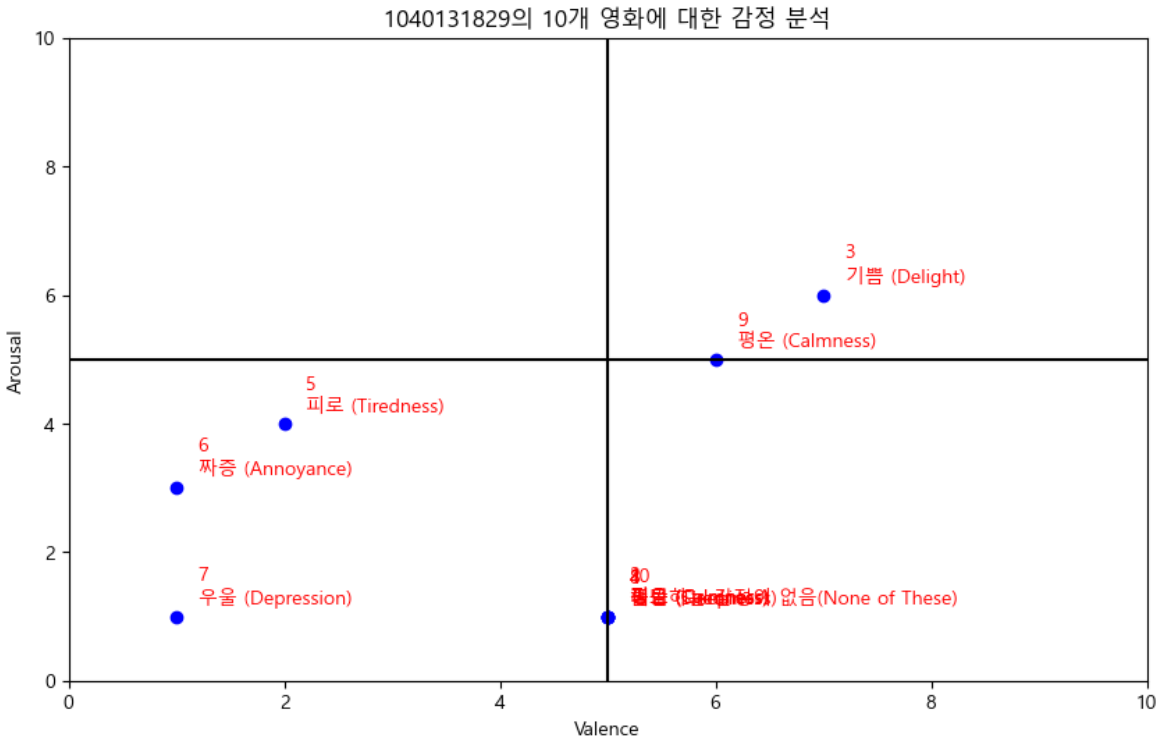

1089970961의 10개 영화에 대한 감정 분석
Arousal
Valence
5
두려움 (Fear)
6
슬픔 (Sadness)
3
기쁨 (Delight)
9
슬픔 (Sadness)
7
우울 (Depression)
10
만족 (Contentment)
8
행복 (Happiness)
2
평온 (Calmness)

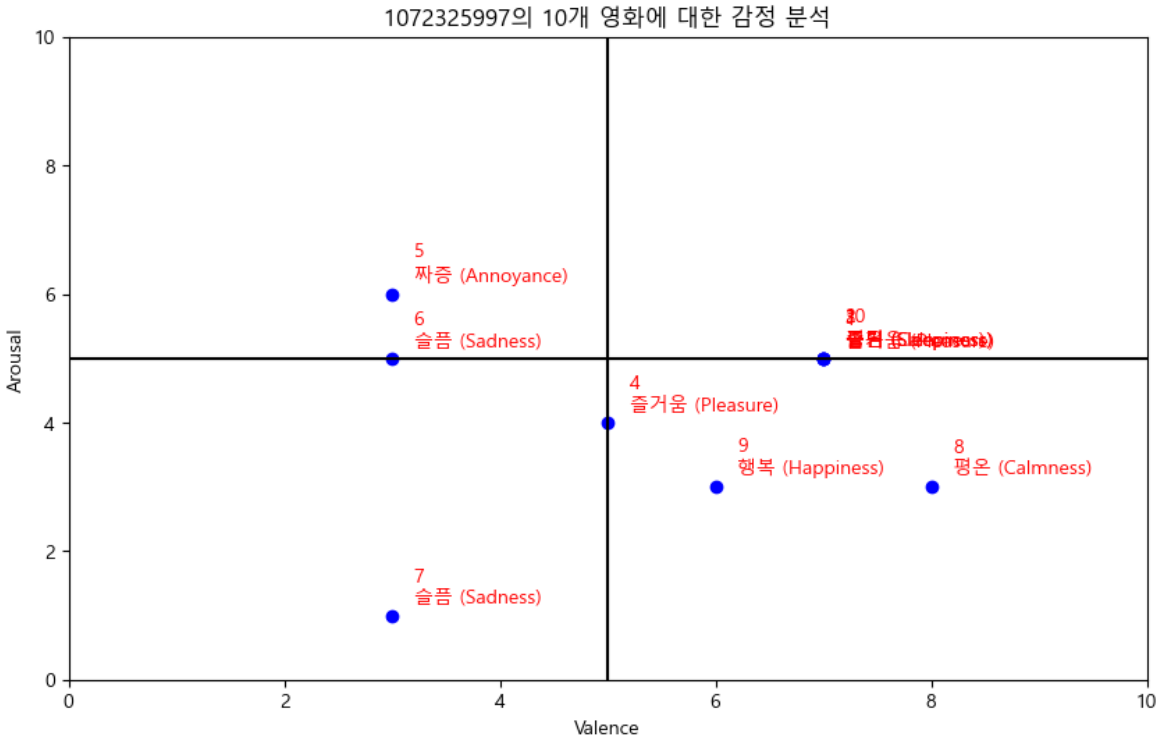

1027642730의 10개 영화에 대한 감정 분석
Arousal
Valence
5
놀람 (Astonishment)
6
우울 (Depression)
7
해당하는 감정이 없음(None of These)
10
행복 (Happiness)
4
평온 (Calmness)
3
해당하는 감정이 없음(None of These)
2
평온 (Calmness)

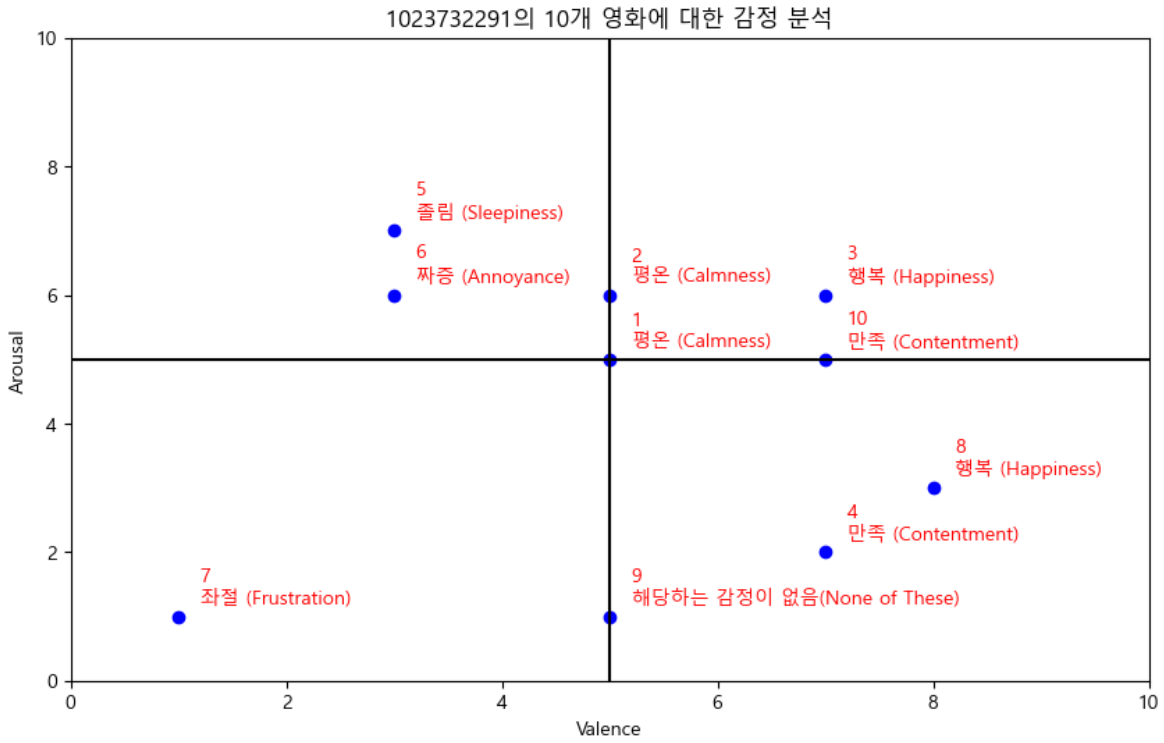

1051002851의 10개 영화에 대한 감정 분석
Arousal
Valence
6
좌절 (Frustration)
10
즐거움 (Pleasure)
5
두려움 (Fear)
7
졸림 (Sleepiness)
4
평온 (Calmness)
9
행복 (Happiness)

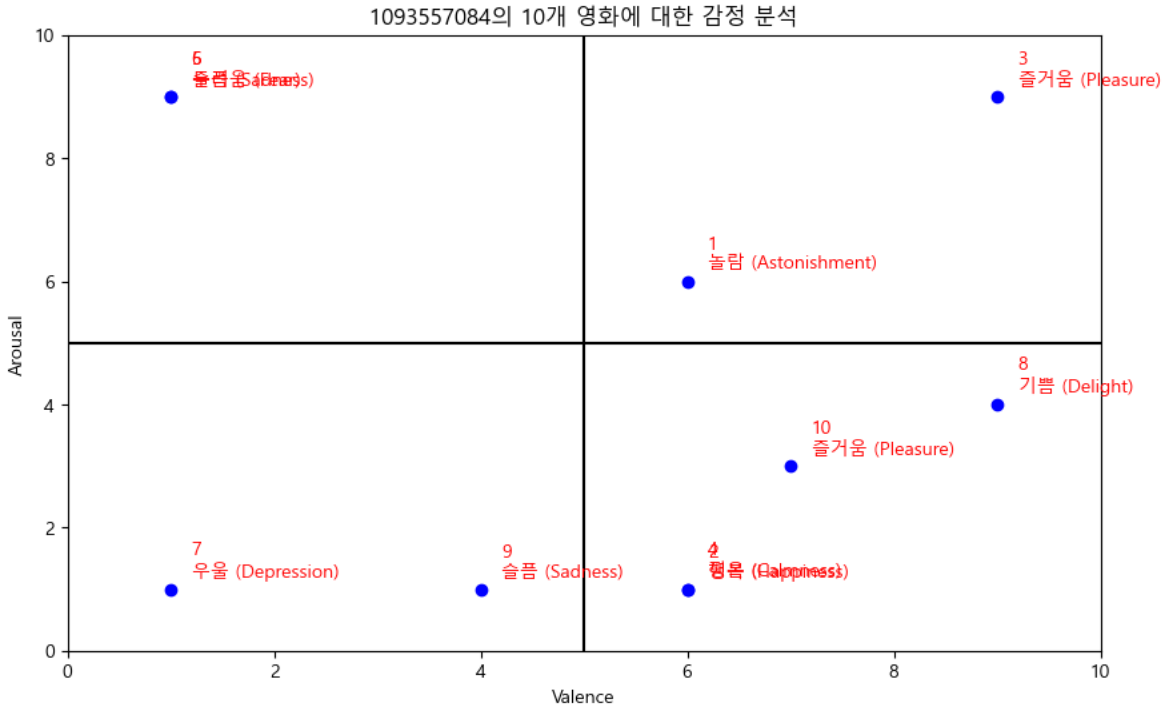

1045140817의 10개 영화에 대한 감정 분석
5
두려움 (Fear)
3
즐거움 (Pleasure)
6
짜증 (Annoyance)
7
우울 (Depression)
10
만족 (Contentment)
9
슬픔 (Sadness)
4
만족 (Contentment)
8
행복 (Happiness)
Arousal
Valence
0
2
4
6
8
10

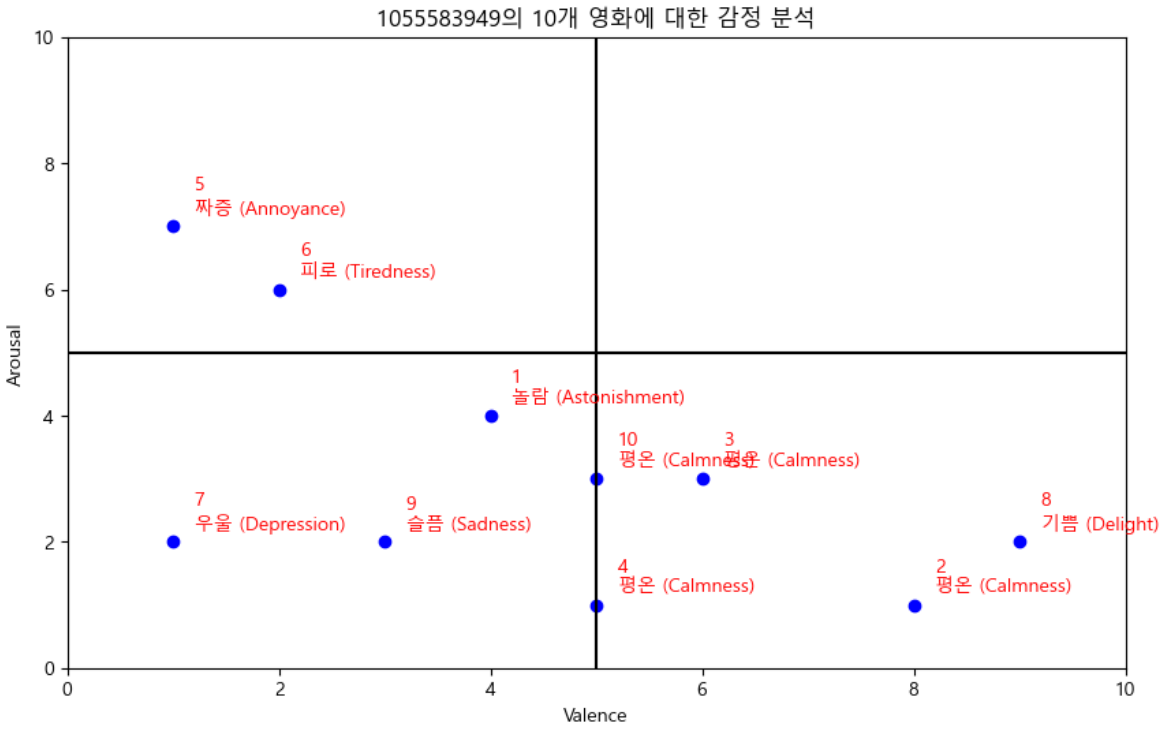

1064186860의 10개 영화에 대한 감정 분석
5
두려움 (Fear)
6
짜증 (Annoyance)
7
우울 (Depression)
9
좌절 (Frustration)
10
8
Arousal
Valence
0
2
4
6
8
10

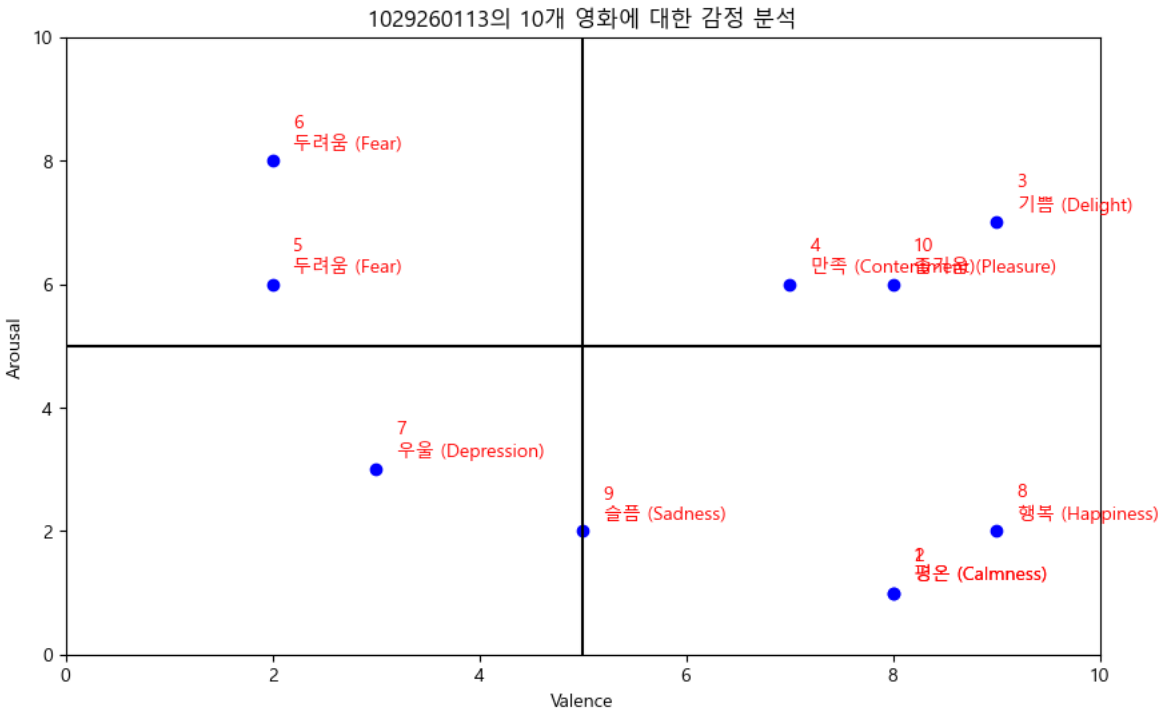

1040131829의 10개 영화에 대한 감정 분석
Arousal
Valence
3
기쁨 (Delight)
9
평온 (Calmness)
5
피로 (Tiredness)
6
짜증 (Annoyance)
7
우울 (Depression)
없음(None of These)

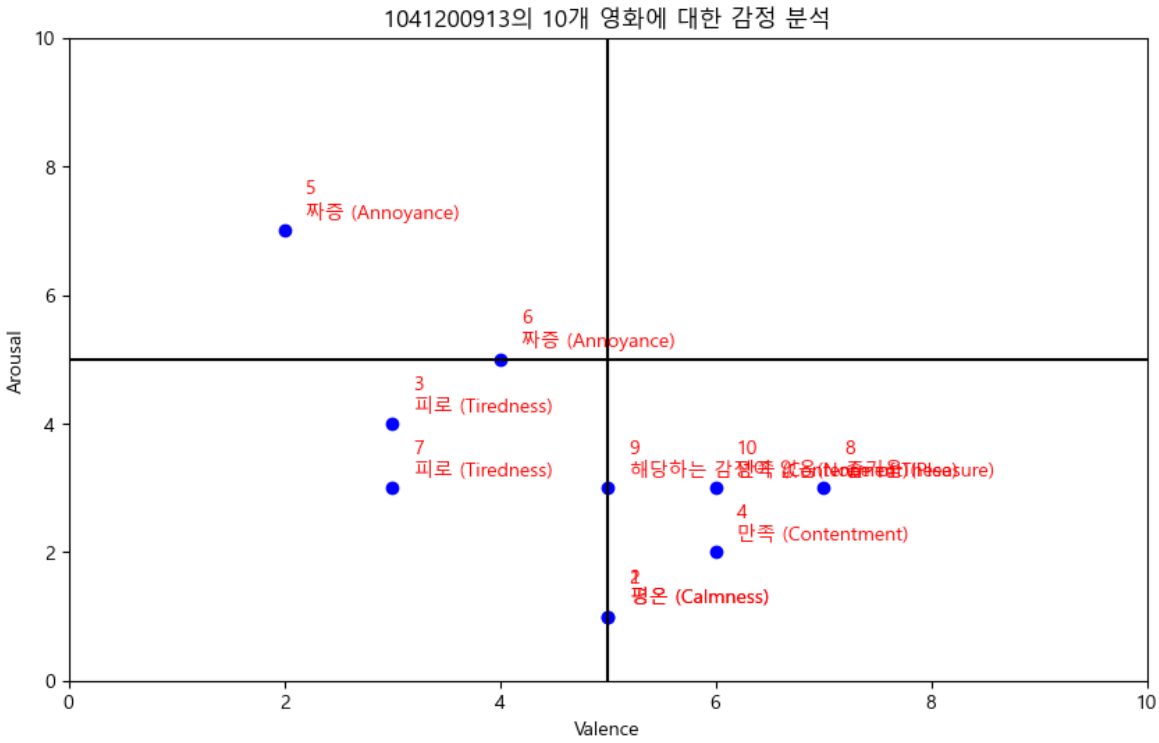

1072325997의 10개 영화에 대한 감정 분석
Arousal
Valence
5
짜증 (Annoyance)
6
슬픔 (Sadness)
4
즐거움 (Pleasure)
9
행복 (Happiness)
8
평온 (Calmness)
7
슬픔 (Sadness)

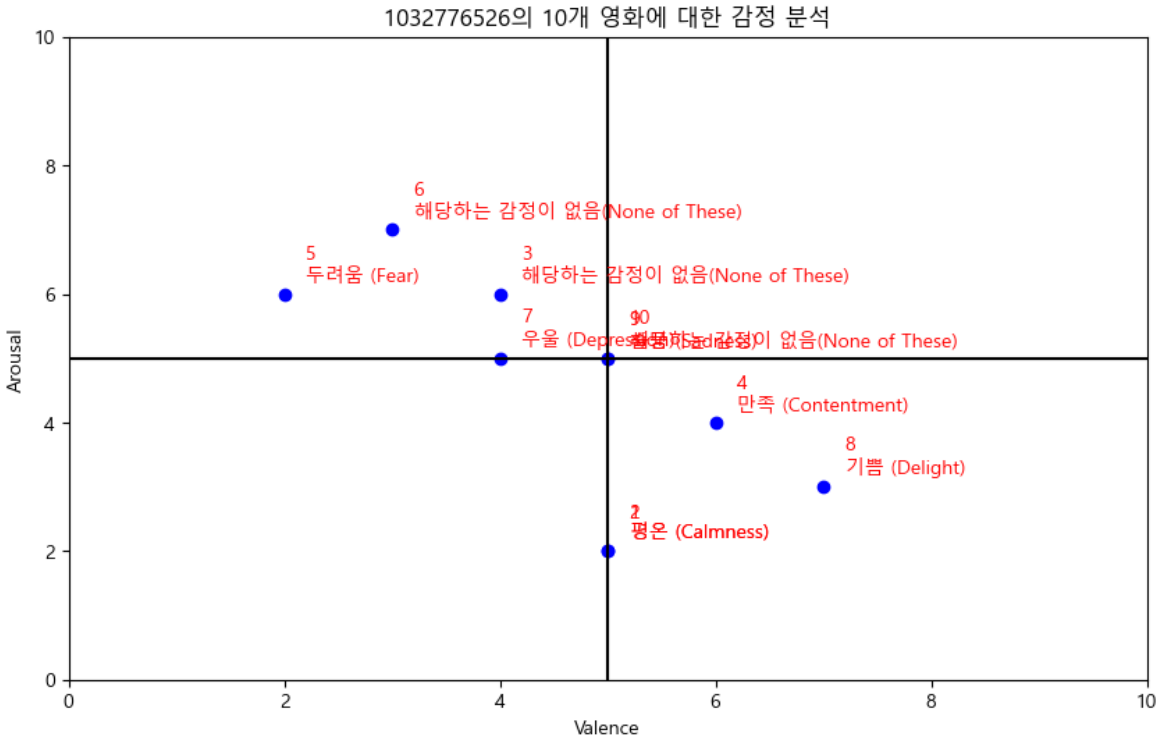

1023732291의 10개 영화에 대한 감정 분석
Arousal
Valence
5
졸림 (Sleepiness)
6
짜증 (Annoyance)
2
평온 (Calmness)
3
행복 (Happiness)
1
평온 (Calmness)
10
만족 (Contentment)
8
행복 (Happiness)
4
만족 (Contentment)
7
좌절 (Frustration)
9
해당하는 감정이 없음(None of These)

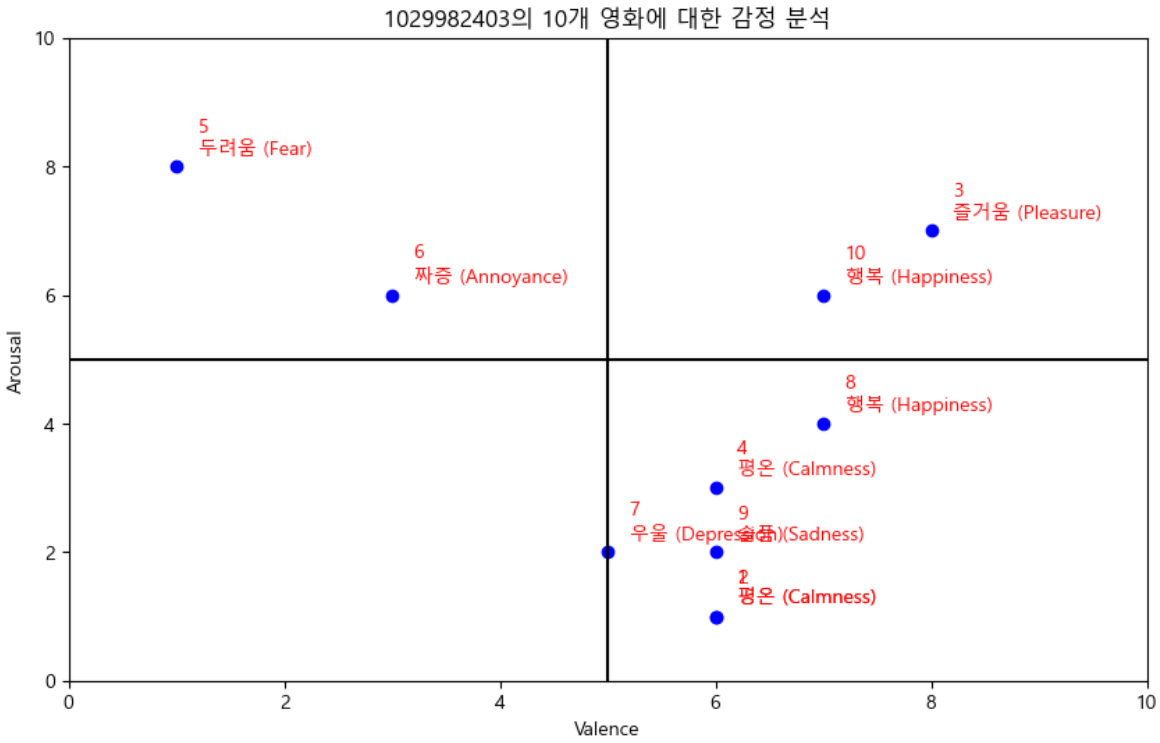

1093557084의 10개 영화에 대한 감정 분석
Arousal
Valence
3
즐거움 (Pleasure)
1
놀람 (Astonishment)
8
기쁨 (Delight)
10
즐거움 (Pleasure)
7
우울 (Depression)
9
슬픔 (Sadness)

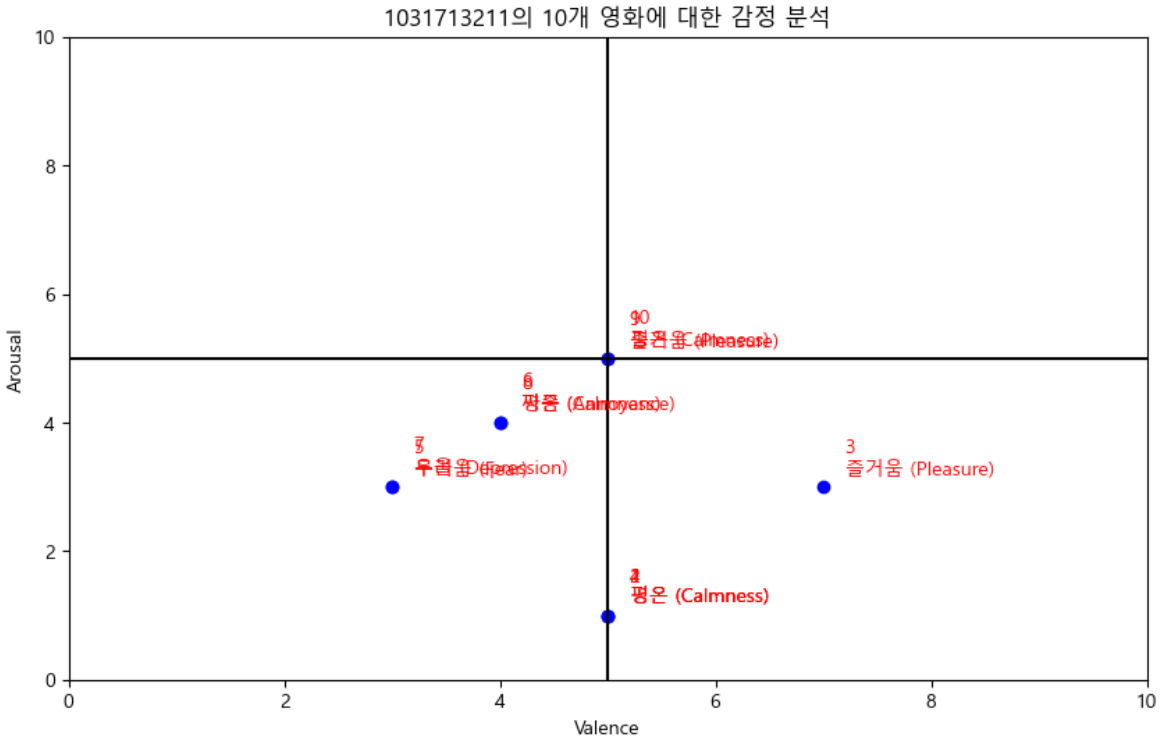

1055583949의 10개 영화에 대한 감정 분석
Arousal
Valence
5
짜증 (Annoyance)
6
피로 (Tiredness)
1
놀람 (Astonishment)
10
평온 (Calmness)
3
평온 (Calmness)
7
우울 (Depression)
9
슬픔 (Sadness)
8
기쁨 (Delight)
4
평온 (Calmness)
2
평온 (Calmness)

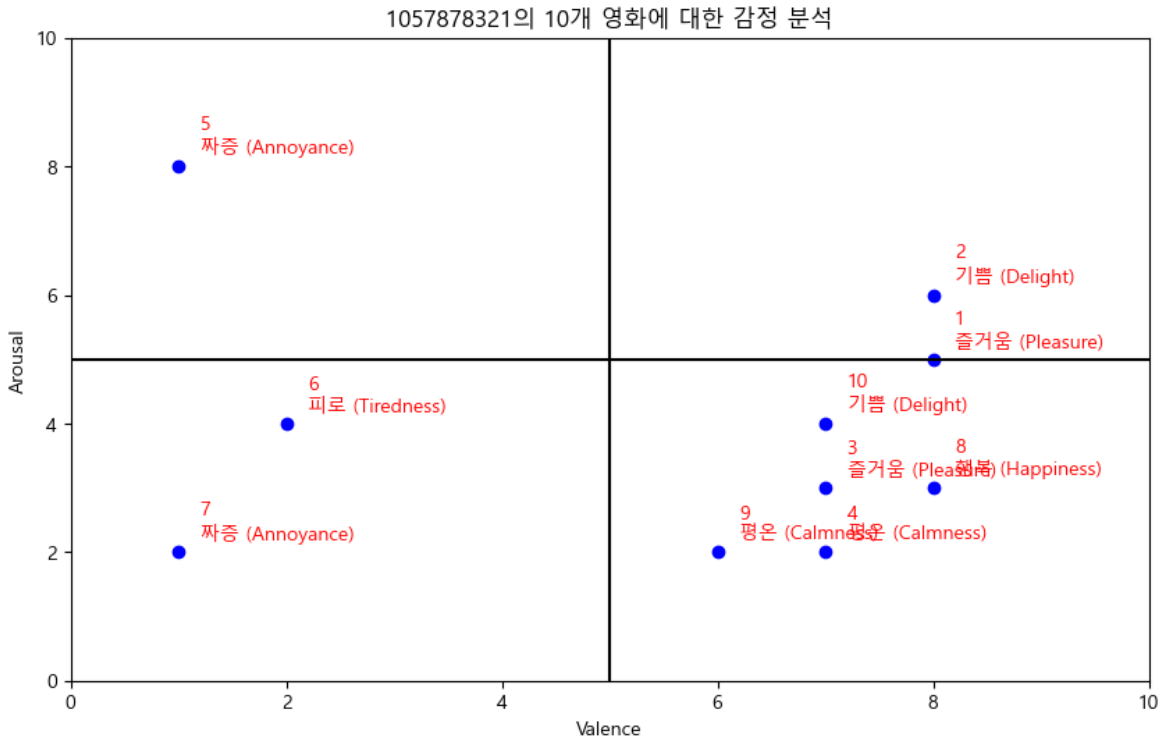

1029260113의 10개 영화에 대한 감정 분석
Arousal
Valence
6
두려움 (Fear)
5
두려움 (Fear)
3
기쁨 (Delight)
4
만족 (Contentment)
10
즐거움 (Pleasure)
7
우울 (Depression)
9
슬픔 (Sadness)
8
행복 (Happiness)
1
평온 (Calmness)

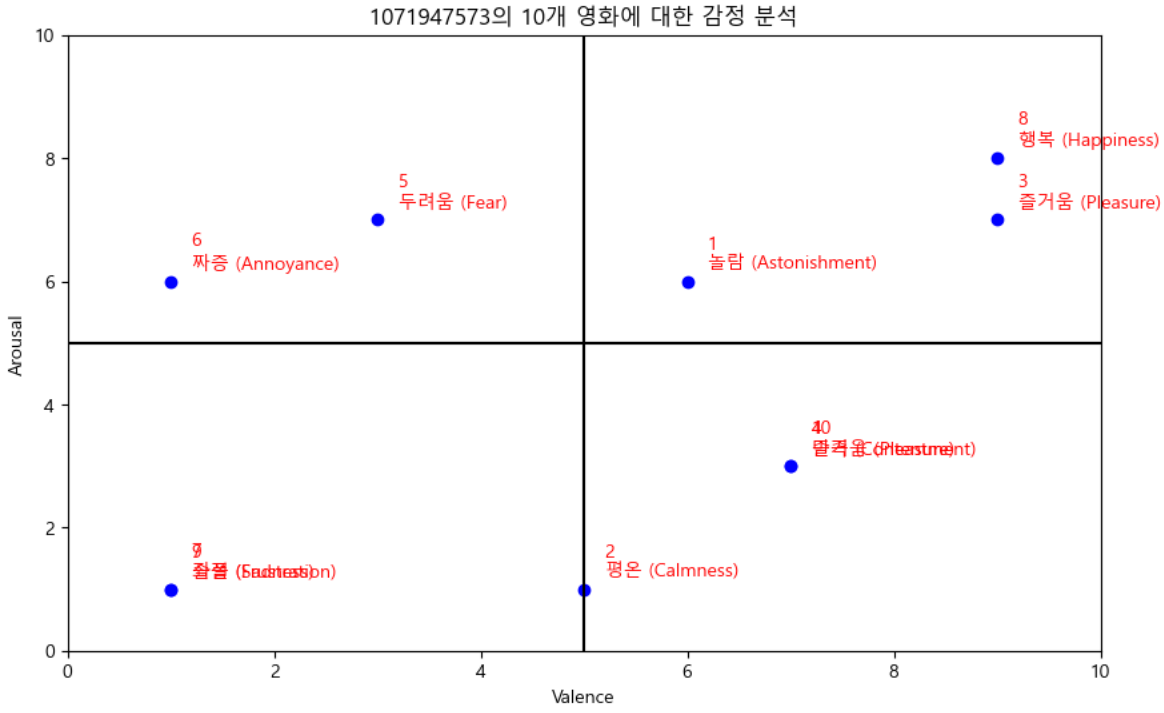

1041200913의 10개 영화에 대한 감정 분석
Arousal
Valence
5
짜증 (Annoyance)
6
짜증 (Annoyance)
3
피로 (Tiredness)
7
피로 (Tiredness)
9
10
8
4
만족 (Contentment)
2
평온 (Calmness)

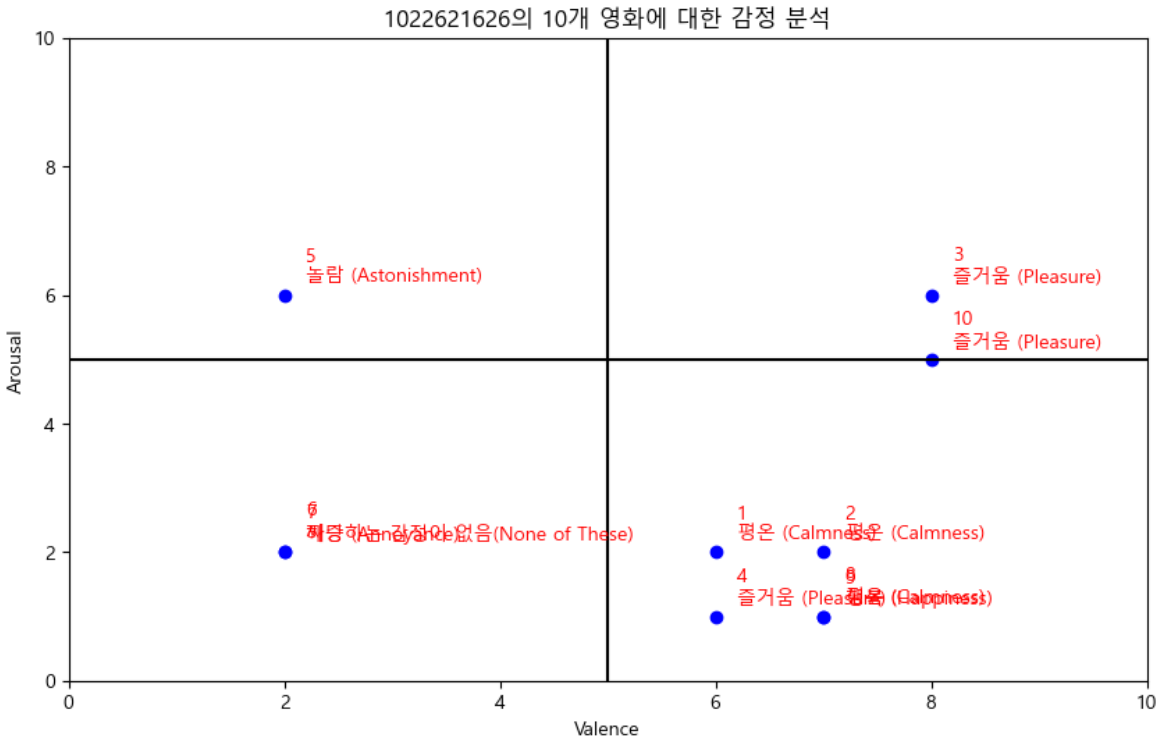

1032776526의 10개 영화에 대한 감정 분석
Arousal
Valence
6
해당하는 감정이 없음(None of These)
5
두려움 (Fear)
3
해당하는 감정이 없음(None of These)
7
우울 (Depressed/Sadness)
9
10
4
만족 (Contentment)
8
기쁨 (Delight)
2
평온 (Calmness)

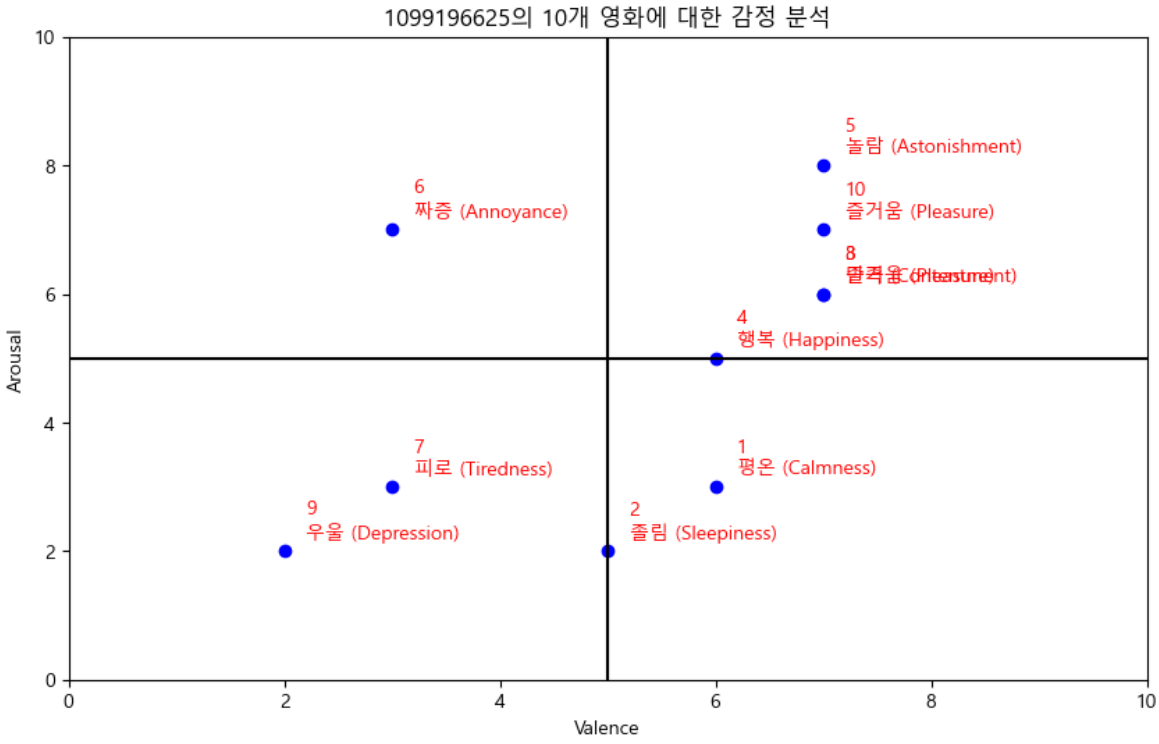

1029982403의 10개 영화에 대한 감정 분석
Arousal
Valence
5
두려움 (Fear)
6
짜증 (Annoyance)
3
즐거움 (Pleasure)
10
행복 (Happiness)
8
행복 (Happiness)
4
평온 (Calmness)
7
우울 (Depression)
9
슬픔 (Sadness)
2
평온 (Calmness)

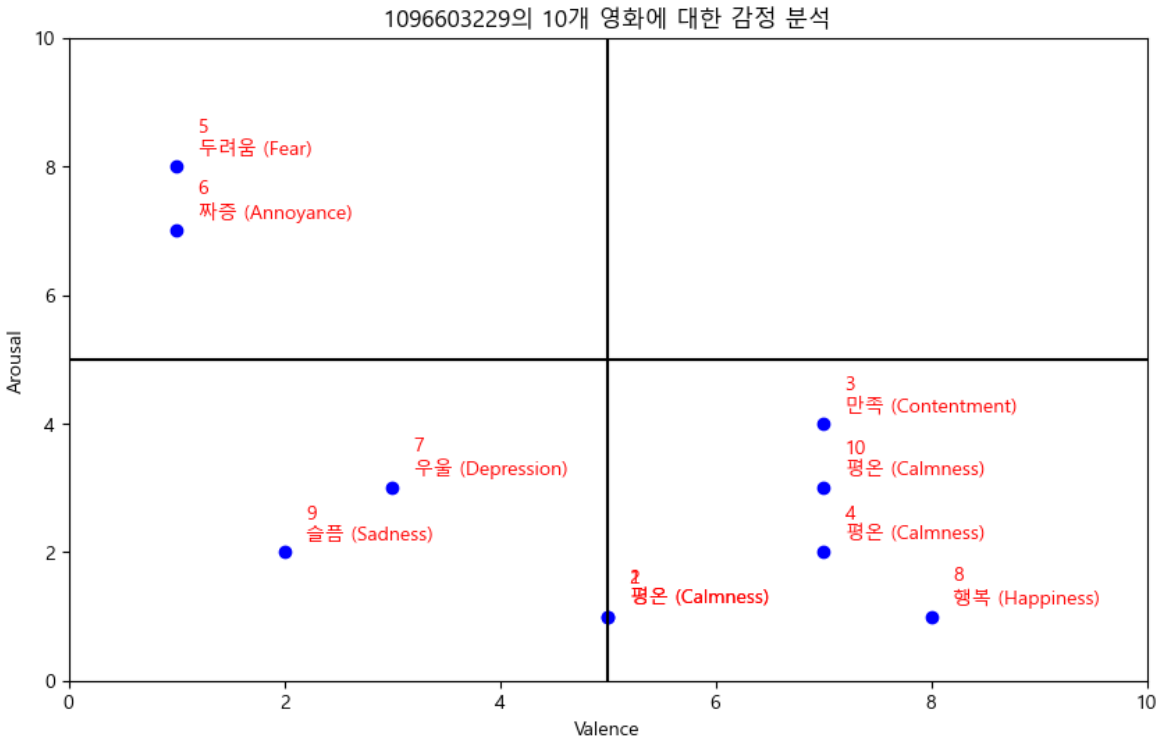

1031713211의 10개 영화에 대한 감정 분석
Arousal
Valence
3
즐거움 (Pleasure)
평온 (Calmness)

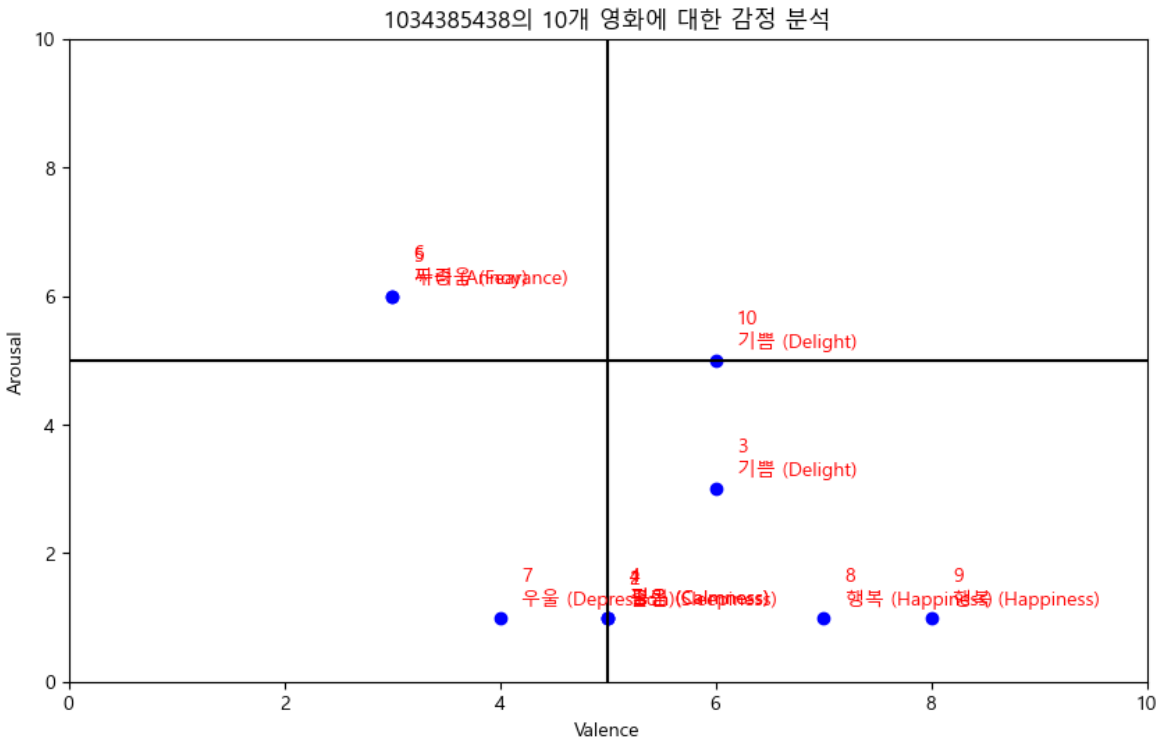

1057878321의 10개 영화에 대한 감정 분석
5
짜증 (Annoyance)
2
기쁨 (Delight)
1
즐거움 (Pleasure)
6
피로 (Tiredness)
10
기쁨 (Delight)
3
즐거움 (Pleasure)
8
행복 (Happiness)
7
짜증 (Annoyance)
9
평온 (Calmness)
4
평온 (Calmness)
Arousal
Valence
0
2
4
6
8
10

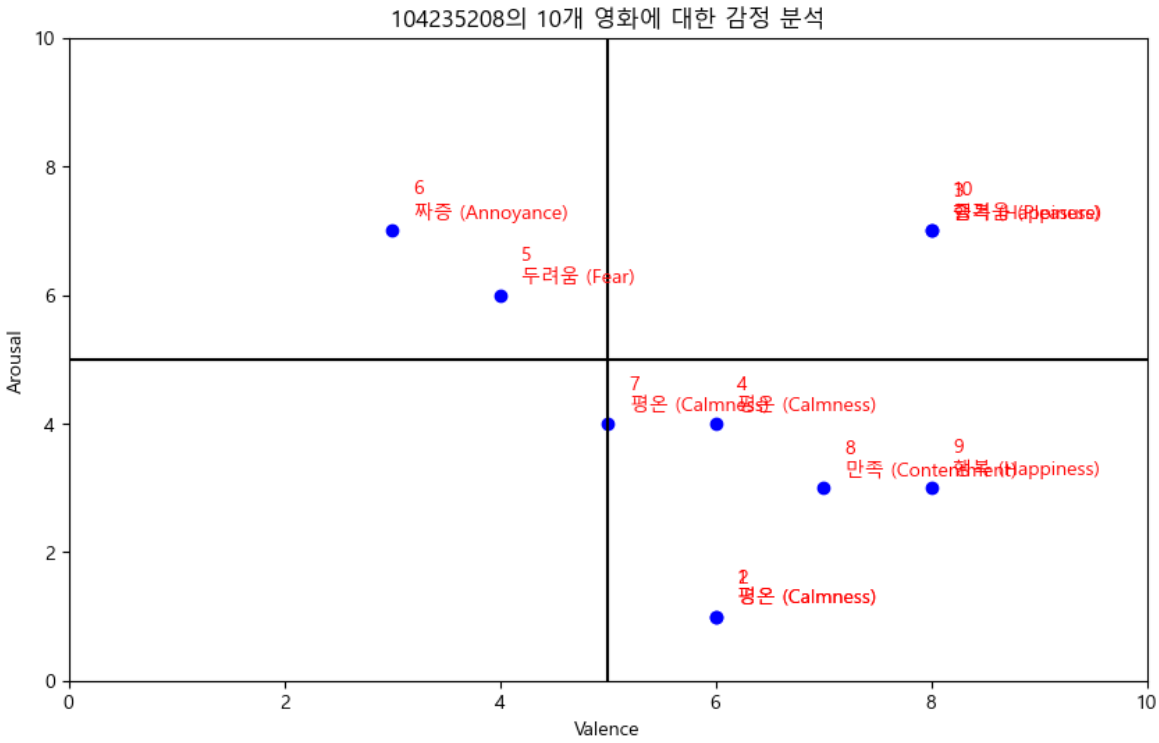

1071947573의 10개 영화에 대한 감정 분석
8
행복 (Happiness)
5
두려움 (Fear)
3
즐거움 (Pleasure)
6
짜증 (Annoyance)
1
놀람 (Astonishment)
2
평온 (Calmness)
Arousal
Valence
0
2
4
6
8
10

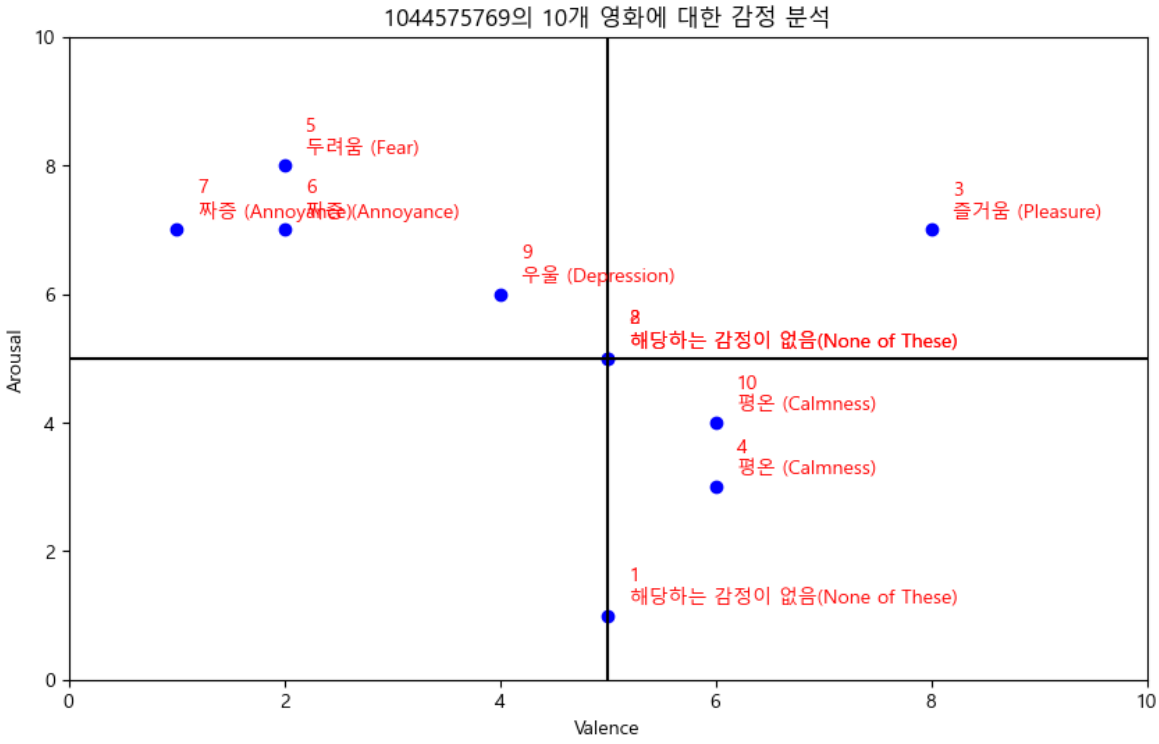

1022621626의 10개 영화에 대한 감정 분석
Arousal
Valence
5
놀람 (Astonishment)
3
즐거움 (Pleasure)
10
즐거움 (Pleasure)
1
평온 (Calmness)
2
평온 (Calmness)
4
즐거움 (Pleasure)

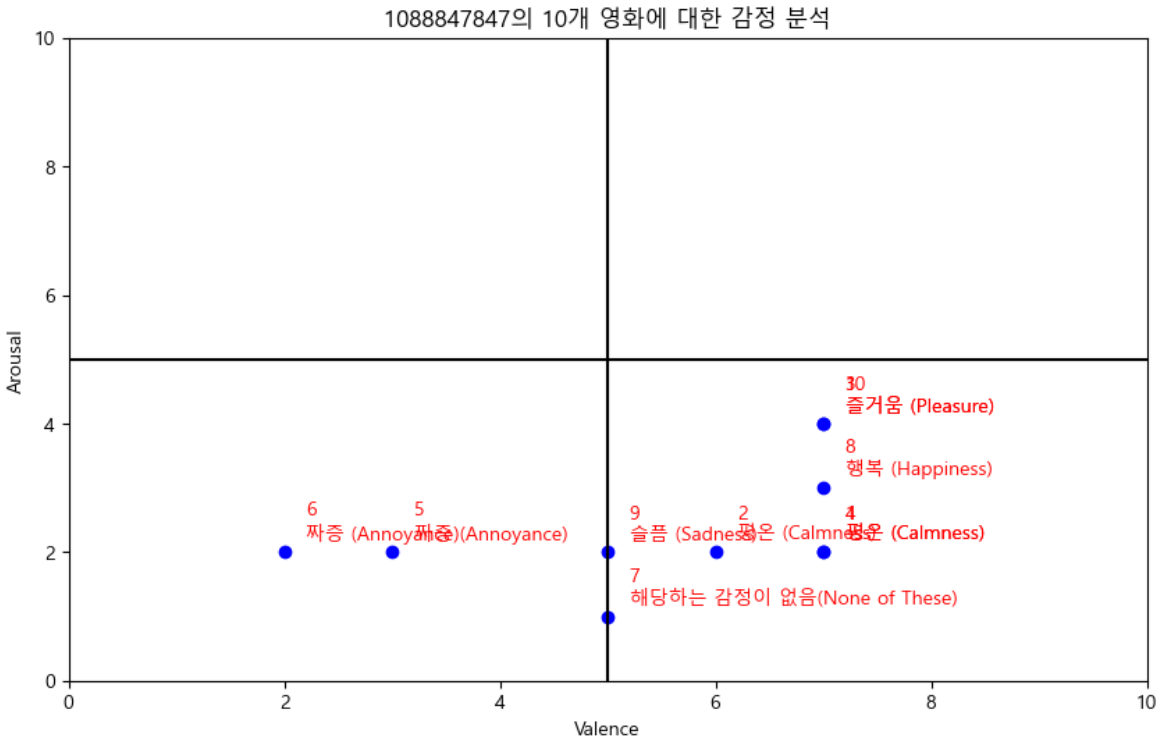

1099196625의 10개 영화에 대한 감정 분석
Arousal
Valence
5
놀람 (Astonishment)
6
짜증 (Annoyance)
10
즐거움 (Pleasure)
4
행복 (Happiness)
7
피로 (Tiredness)
1
평온 (Calmness)
9
우울 (Depression)
2
졸림 (Sleepiness)

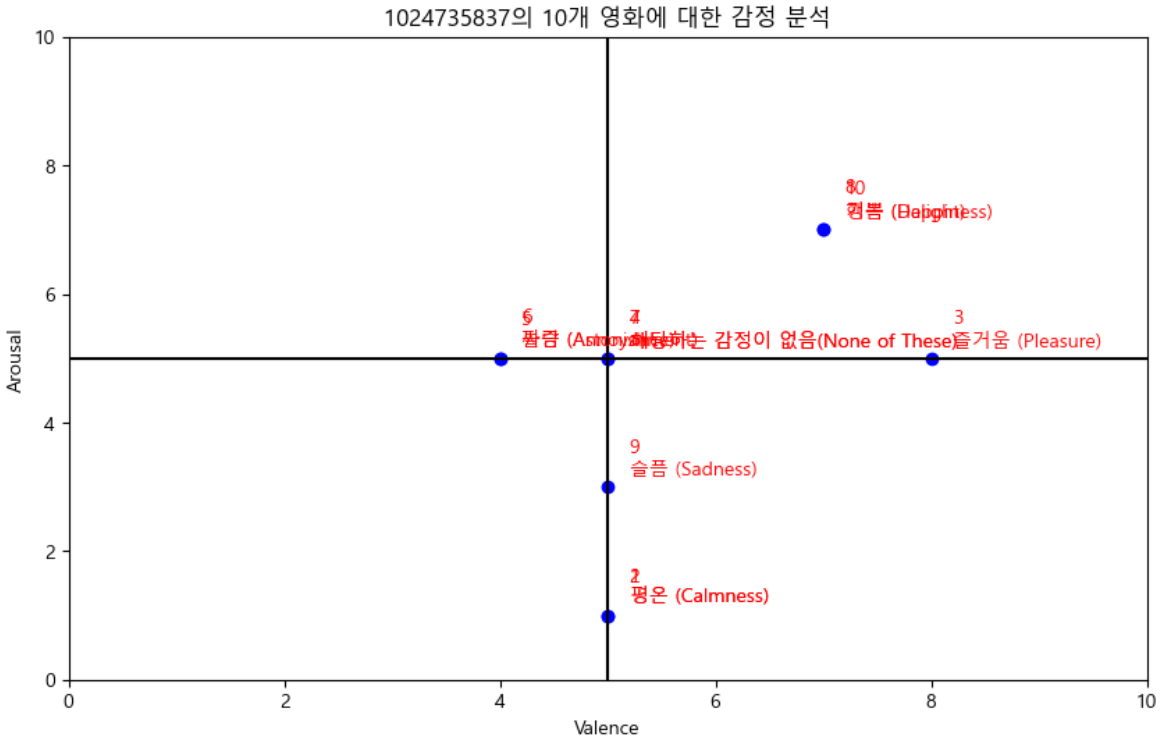

1096603229의 10개 영화에 대한 감정 분석
5
두려움 (Fear)
6
짜증 (Annoyance)
7
우울 (Depression)
9
슬픔 (Sadness)
3
만족 (Contentment)
10
평온 (Calmness)
4
평온 (Calmness)
2
평온 (Calmness)
8
행복 (Happiness)
Arousal
Valence

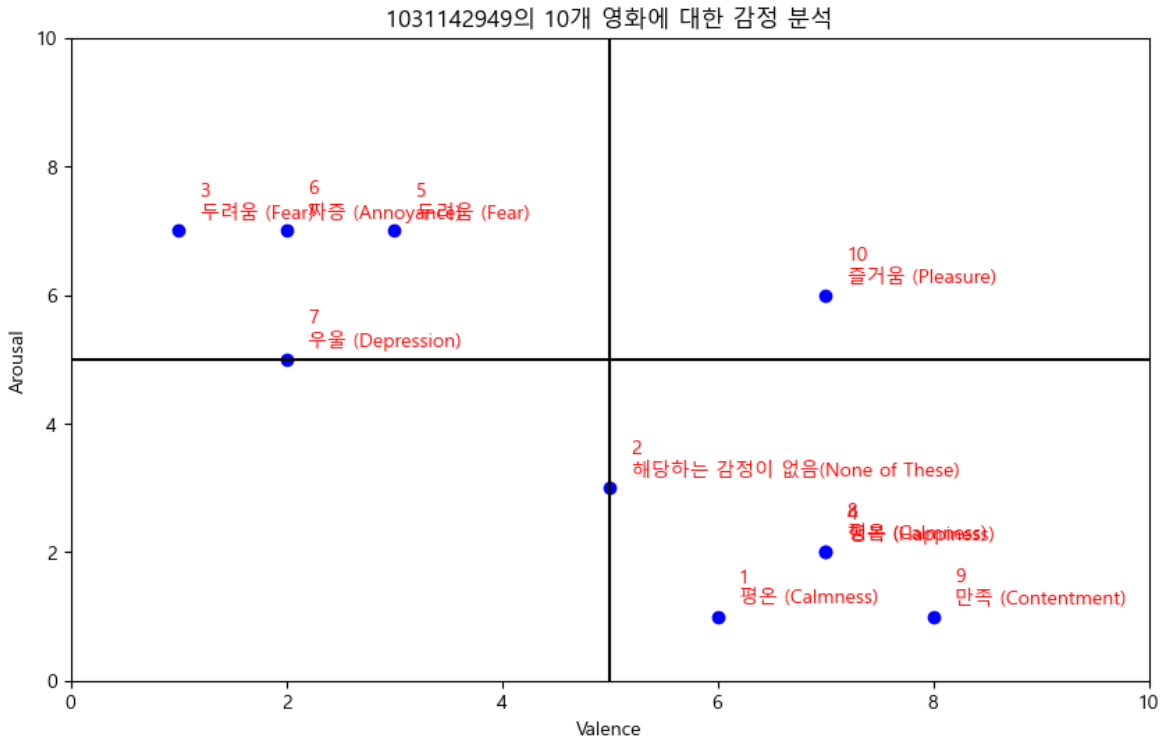

1034385438의 10개 영화에 대한 감정 분석
10
기쁨 (Delight)
3
기쁨 (Delight)
8
행복 (Happiness)
Arousal
Valence

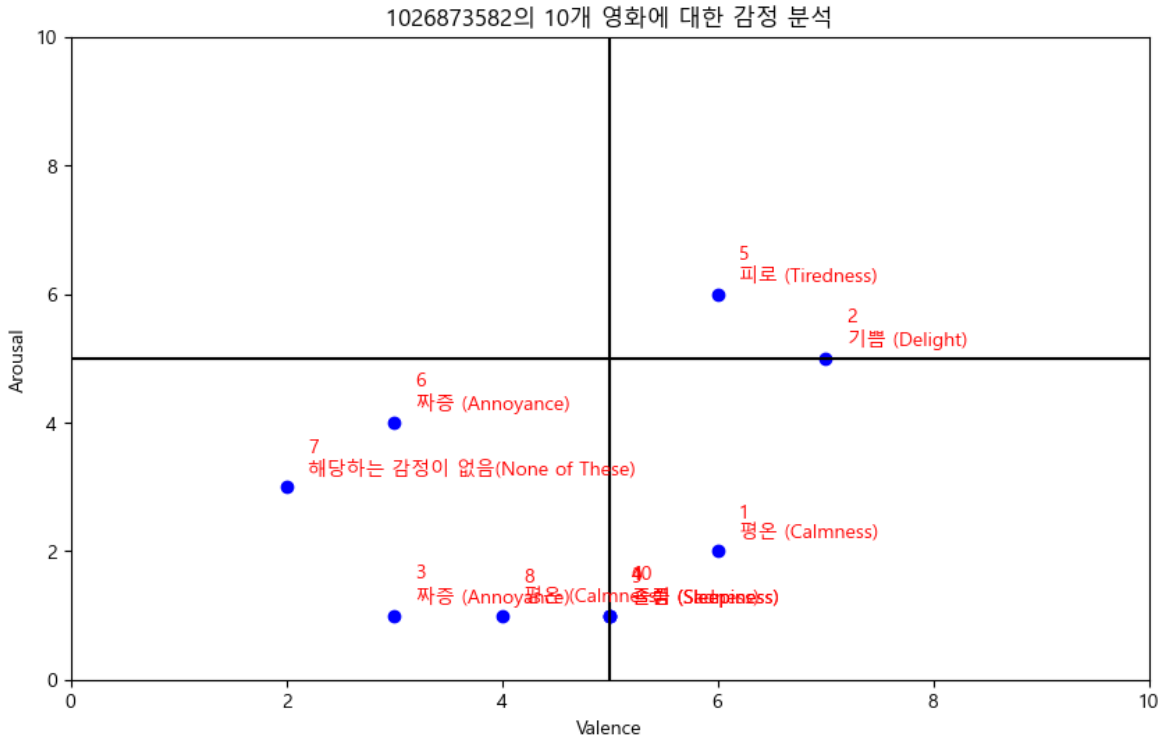

104235208의 10개 영화에 대한 감정 분석
Arousal
Valence
6
짜증 (Annoyance)
5
두려움 (Fear)
7
평온 (Calmness)
4
평온 (Calmness)
8
만족 (Contentment)
9
행복 (Happiness)
2
평온 (Calmness)


1044575769의 10개 영화에 대한 감정 분석
Arousal
Valence
5
두려움 (Fear)
7
짜증 (Annoyance)
6
짜증 (Annoyance)
3
즐거움 (Pleasure)
9
우울 (Depression)
8
해당하는 감정이 없음(None of These)
10
평온 (Calmness)
4
평온 (Calmness)
1
해당하는 감정이 없음(None of These)


1088847847의 10개 영화에 대한 감정 분석
Arousal
Valence
10
8
6
4
2
0
0
2
4
6
8
10
10
즐거움 (Pleasure)
8
행복 (Happiness)
6
짜증 (Annoyance)
5
짜증 (Annoyance)
9
슬픔 (Sadness)
2
평온 (Calmness)
4
평온 (Calmness)
7
해당하는 감정이 없음(None of These)


1024735837의 10개 영화에 대한 감정 분석
Arousal
Valence
10
8
6
4
2
0
0
2
4
6
8
10
10
8
6
5
4
3
즐거움 (Pleasure)
9
슬픔 (Sadness)
2
평온 (Calmness)





In [23]:
```
# Now loop through each participant to calculate the difference from the mean an
final_results = []

# Loop through each participant to calculate the differences
for person in df['전화번호'].unique():
    person_data = df[df['전화번호'] == person]

    person_results = {'전화번호': person}  # Start with phone number

    # Loop through each movie for the person
    for i, movie in enumerate(range(10)):
        valence_col = f'Valence_After_Movie_{movie}'
        arousal_col = f'Arousal_After_Movie_{movie}'
        emotion_col = f'Emotion_Keyword_{movie}'
```

```
        # Get the mean and SD for Valence and Arousal for the current movie
        valence_mean = results_df.loc[results_df['Movie'] == movie, 'Valence Mea
        arousal_mean = results_df.loc[results_df['Movie'] == movie, 'Arousal Mea

        # Get the participant's Valence, Arousal, and Emotion for the movie
        valence = person_data[valence_col].values[0]
        arousal = person_data[arousal_col].values[0]
        emotion = person_data[emotion_col].values[0]

        # Calculate the differences from the mean
        valence_diff = abs(valence - valence_mean)
        arousal_diff = abs(arousal - arousal_mean)

        # Store the differences and the emotion in the result for this person
        person_results[f'Valence_{movie+1}_Diff'] = valence_diff
        person_results[f'Arousal_{movie+1}_Diff'] = arousal_diff
        person_results[f'Emotion_{movie+1}'] = emotion

    final_results.append(person_results)

# Convert the final result list to a DataFrame
final_df = pd.DataFrame(final_results)

# Sort the results based on the smallest difference in Valence and Arousal
# Corrected sorting code, flattening the difference columns
valence_diff_columns = [f'Valence_{i+1}_Diff' for i in range(10)]
arousal_diff_columns = [f'Arousal_{i+1}_Diff' for i in range(10)]

# Sort the DataFrame based on the sum of differences across all movies
final_df_sorted = final_df.loc[:, ['전화번호'] + valence_diff_columns + arousal_
final_df_sorted['Total_Diff'] = final_df_sorted[valence_diff_columns + arousal_d

# Sort by the 'Total_Diff'
final_df_sorted = final_df_sorted.sort_values(by='Total_Diff')

# Display the sorted results
print(final_df_sorted)

final_df_sorted.to_excel('final_df_sorted.xlsx', index=False)
```

	전화번호	Valence_1_Diff	Valence_2_Diff	Valence_3_Diff	\
5	1022307441	0.257143	0.228571	1.342857	
19	1041200913	0.742857	0.771429	3.657143	
9	1027642730	0.742857	0.771429	0.342857	
2	1039471109	0.742857	0.771429	0.342857	
12	1064186860	0.742857	0.771429	1.657143	
20	1032776526	0.742857	0.771429	2.657143	
14	1072325997	1.257143	1.228571	0.342857	
21	1029982403	0.257143	0.228571	1.342857	
6	1067510885	0.742857	0.771429	0.342857	
11	1045140817	0.257143	0.228571	2.342857	
25	1022621626	0.257143	1.228571	1.342857	
15	1023732291	0.742857	0.771429	0.342857	
31	1088847847	1.257143	0.228571	0.342857	
1	1051350406	0.257143	1.228571	1.342857	
29	104235208	0.257143	0.228571	1.342857	
28	1034385438	0.742857	0.771429	0.657143	
27	1096603229	0.742857	0.771429	0.342857	
0	1049468170	0.742857	0.771429	0.342857	
26	1099196625	0.257143	0.771429	0.342857	
22	1031713211	0.742857	0.771429	0.342857	
32	1024735837	0.742857	0.771429	1.342857	
33	1031142949	0.257143	0.771429	5.657143	
23	1057878321	2.257143	2.228571	0.342857	
17	1055583949	1.742857	2.228571	0.657143	
18	1029260113	2.257143	2.228571	2.342857	
30	1044575769	0.742857	0.771429	1.342857	
8	1089970961	3.257143	3.228571	1.657143	
13	1040131829	0.742857	0.771429	0.342857	
34	1026873582	0.257143	1.228571	3.657143	
24	1071947573	0.257143	0.771429	2.342857	
16	1093557084	0.257143	0.228571	2.342857	
3	1036149407	0.742857	0.771429	1.657143	
4	1050337288	1.257143	0.771429	1.342857	
7	1083414563	0.742857	0.771429	4.657143	
10	1051002851	0.742857	0.771429	2.342857	

	Valence_4_Diff	Valence_5_Diff	Valence_6_Diff	Valence_7_Diff	\
5	0.057143	1.571429	1.171429	0.371429	
19	0.057143	0.428571	1.171429	0.371429	
9	0.057143	1.428571	0.171429	0.371429	
2	0.942857	0.571429	0.171429	0.628571	
12	0.942857	1.428571	0.828571	0.371429	
20	0.057143	0.428571	0.171429	1.371429	
14	0.942857	0.571429	0.171429	0.371429	
21	0.057143	1.428571	0.171429	2.371429	
6	1.057143	0.428571	2.171429	0.371429	
11	0.057143	1.428571	0.828571	0.371429	
25	0.057143	0.428571	0.828571	0.628571	
15	1.057143	0.571429	0.171429	1.628571	
31	1.057143	0.571429	0.828571	2.371429	
1	2.057143	0.428571	0.171429	0.371429	
29	0.057143	1.571429	0.171429	2.371429	
28	0.942857	0.571429	0.171429	1.371429	
27	1.057143	1.428571	1.828571	0.371429	
0	1.057143	0.428571	2.171429	0.628571	
26	0.057143	4.571429	0.171429	0.371429	
22	0.942857	0.571429	1.171429	0.371429	
32	0.942857	1.571429	1.171429	2.371429	
33	1.057143	0.571429	0.828571	0.628571	

```
23        1.057143        1.428571        0.828571        1.628571
17        0.942857        1.428571        0.828571        1.628571
18        1.057143        0.428571        0.828571        0.371429
30        0.057143        0.428571        0.828571        1.628571
8         1.057143        1.428571        1.171429        1.371429
13        0.942857        0.428571        1.828571        1.628571
34        0.942857        3.571429        0.171429        0.628571
24        1.057143        0.571429        1.828571        1.628571
16        0.057143        1.428571        1.828571        1.628571
3         1.942857        3.571429        3.171429        1.371429
4         0.942857        1.428571        0.828571        1.628571
7         0.942857        1.428571        1.828571        1.628571
10        0.942857        1.428571        2.171429        1.628571

    Valence_8_Diff  Valence_9_Diff  ...                  Emotion_2  \
5         0.028571        0.142857  ...  해당하는 감정이 없음(None of These)
19        0.028571        0.142857  ...               평온 (Calmness)
9         0.028571        0.142857  ...               평온 (Calmness)
2         0.028571        1.857143  ...               평온 (Calmness)
12        0.028571        0.857143  ...  해당하는 감정이 없음(None of These)
20        0.028571        0.142857  ...               평온 (Calmness)
14        1.028571        1.142857  ...             졸림 (Sleepiness)
21        0.028571        1.142857  ...               평온 (Calmness)
6         0.028571        0.857143  ...                슬픔 (Sadness)
11        0.028571        0.142857  ...               평온 (Calmness)
25        0.028571        2.142857  ...               평온 (Calmness)
15        1.028571        0.142857  ...               평온 (Calmness)
31        0.028571        0.142857  ...               평온 (Calmness)
1         0.971429        2.857143  ...              즐거움 (Pleasure)
29        0.028571        3.142857  ...               평온 (Calmness)
28        0.028571        3.142857  ...               평온 (Calmness)
27        1.028571        2.857143  ...               평온 (Calmness)
0         0.971429        1.142857  ...               평온 (Calmness)
26        0.028571        2.857143  ...             졸림 (Sleepiness)
22        2.971429        0.142857  ...               평온 (Calmness)
32        0.028571        0.142857  ...               평온 (Calmness)
33        0.028571        3.142857  ...  해당하는 감정이 없음(None of These)
23        1.028571        1.142857  ...                기쁨 (Delight)
17        2.028571        1.857143  ...               평온 (Calmness)
18        2.028571        0.142857  ...               평온 (Calmness)
30        1.971429        0.857143  ...  해당하는 감정이 없음(None of These)
8         1.028571        1.857143  ...               평온 (Calmness)
13        1.971429        1.142857  ...             졸림 (Sleepiness)
34        2.971429        0.142857  ...                기쁨 (Delight)
24        2.028571        3.857143  ...               평온 (Calmness)
16        2.028571        0.857143  ...               평온 (Calmness)
3         0.971429        0.142857  ...               평온 (Calmness)
4         0.971429        1.857143  ...               평온 (Calmness)
7         1.971429        0.142857  ...  해당하는 감정이 없음(None of These)
10        2.028571        4.142857  ...               평온 (Calmness)

                     Emotion_3                 Emotion_4          Emotion_5  \
5                 기쁨 (Delight)             평온 (Calmness)         두려움 (Fe
ar)
19             피로 (Tiredness)           만족 (Contentment)     짜증 (Annoyanc
e)
9   해당하는 감정이 없음(None of These)              평온 (Calmness)  놀람 (Asto
nishment)
2              즐거움 (Pleasure)             평온 (Calmness)    우울 (Depressi
on)
```

12 즐거움 (Pleasure) 해당하는 감정이 없음(None of These) 두려움 (Fear)
20 해당하는 감정이 없음(None of These) 만족 (Contentment) 두려움 (Fear)
14 즐거움 (Pleasure) 즐거움 (Pleasure) 짜증 (Annoyance)
21 즐거움 (Pleasure) 평온 (Calmness) 두려움 (Fear)
6 즐거움 (Pleasure) 행복 (Happiness) 짜증 (Annoyance)
11 즐거움 (Pleasure) 만족 (Contentment) 두려움 (Fear)
25 즐거움 (Pleasure) 즐거움 (Pleasure) 놀람 (Astonishment)
15 행복 (Happiness) 만족 (Contentment) 졸림 (Sleepiness)
31 즐거움 (Pleasure) 평온 (Calmness) 짜증 (Annoyance)
1 기쁨 (Delight) 평온 (Calmness) 짜증 (Annoyance)
29 즐거움 (Pleasure) 평온 (Calmness) 두려움 (Fear)
28 기쁨 (Delight) 졸림 (Sleepiness) 두려움 (Fear)
27 만족 (Contentment) 평온 (Calmness) 두려움 (Fear)
0 만족 (Contentment) 만족 (Contentment) 놀람 (Astonishment)
26 즐거움 (Pleasure) 행복 (Happiness) 놀람 (Astonishment)
22 즐거움 (Pleasure) 평온 (Calmness) 두려움 (Fear)
32 즐거움 (Pleasure) 해당하는 감정이 없음(None of These) 놀람 (Astonishment)
33 두려움 (Fear) 평온 (Calmness) 두려움 (Fear)
23 즐거움 (Pleasure) 평온 (Calmness) 짜증 (Annoyance)
17 평온 (Calmness) 평온 (Calmness) 짜증 (Annoyance)
18 기쁨 (Delight) 만족 (Contentment) 두려움 (Fear)
30 즐거움 (Pleasure) 평온 (Calmness) 두려움 (Fear)
8 기쁨 (Delight) 만족 (Contentment) 두려움 (Fear)
13 기쁨 (Delight) 평온 (Calmness) 피로 (Tiredness)
34 짜증 (Annoyance) 졸림 (Sleepiness) 피로 (Tiredness)
24 즐거움 (Pleasure) 만족 (Contentment) 두려움 (Fear)
16 즐거움 (Pleasure) 행복 (Happiness) 두려움 (Fear)
3 해당하는 감정이 없음(None of These) 평온 (Calmness) 만족 (Contentment)
4 즐거움 (Pleasure) 해당하는 감정이 없음(None of These) 두려움 (Fear)
7 해당하는 감정이 없음(None of These) 짜증 (Annoyance) 짜증 (Annoyance)

```
10              즐거움 (Pleasure)              평온 (Calmness)        두려움 (F
ear)

                    Emotion_6                  Emotion_7  \
5              짜증 (Annoyance)            우울 (Depression)
19             짜증 (Annoyance)             피로 (Tiredness)
9              우울 (Depression)  해당하는 감정이 없음(None of These)
2              짜증 (Annoyance)             짜증 (Annoyance)
12             짜증 (Annoyance)            우울 (Depression)
20  해당하는 감정이 없음(None of These)             우울 (Depression)
14               슬픔 (Sadness)              슬픔 (Sadness)
21             짜증 (Annoyance)            우울 (Depression)
6              짜증 (Annoyance)            우울 (Depression)
11             짜증 (Annoyance)            우울 (Depression)
25             짜증 (Annoyance)  해당하는 감정이 없음(None of These)
15             짜증 (Annoyance)           좌절 (Frustration)
31             짜증 (Annoyance)  해당하는 감정이 없음(None of These)
1              짜증 (Annoyance)            우울 (Depression)
29             짜증 (Annoyance)              평온 (Calmness)
28             짜증 (Annoyance)            우울 (Depression)
27             짜증 (Annoyance)            우울 (Depression)
0               평온 (Calmness)            우울 (Depression)
26             짜증 (Annoyance)             피로 (Tiredness)
22             짜증 (Annoyance)            우울 (Depression)
32             짜증 (Annoyance)  해당하는 감정이 없음(None of These)
33             짜증 (Annoyance)            우울 (Depression)
23             피로 (Tiredness)             짜증 (Annoyance)
17             피로 (Tiredness)            우울 (Depression)
18                 두려움 (Fear)             우울 (Depression)
30             짜증 (Annoyance)             짜증 (Annoyance)
8                슬픔 (Sadness)            우울 (Depression)
13             짜증 (Annoyance)            우울 (Depression)
34             짜증 (Annoyance)  해당하는 감정이 없음(None of These)
24             짜증 (Annoyance)           좌절 (Frustration)
16               슬픔 (Sadness)            우울 (Depression)
3             만족 (Contentment)  해당하는 감정이 없음(None of These)
4              피로 (Tiredness)  해당하는 감정이 없음(None of These)
7              짜증 (Annoyance)              슬픔 (Sadness)
10            좌절 (Frustration)            졸림 (Sleepiness)

                    Emotion_8                  Emotion_9  \
5              행복 (Happiness)              슬픔 (Sadness)
19             즐거움 (Pleasure)  해당하는 감정이 없음(None of These)
9              행복 (Happiness)              슬픔 (Sadness)
2              행복 (Happiness)              슬픔 (Sadness)
12               기쁨 (Delight)           좌절 (Frustration)
20               기쁨 (Delight)              슬픔 (Sadness)
14              평온 (Calmness)             행복 (Happiness)
21             행복 (Happiness)              슬픔 (Sadness)
6             만족 (Contentment)              슬픔 (Sadness)
11             행복 (Happiness)              슬픔 (Sadness)
25             행복 (Happiness)              평온 (Calmness)
15             행복 (Happiness)  해당하는 감정이 없음(None of These)
31             행복 (Happiness)              슬픔 (Sadness)
1              행복 (Happiness)            우울 (Depression)
29            만족 (Contentment)             행복 (Happiness)
28             행복 (Happiness)             행복 (Happiness)
27             행복 (Happiness)              슬픔 (Sadness)
0              즐거움 (Pleasure)           만족 (Contentment)
26            만족 (Contentment)            우울 (Depression)
```

22	평온 (Calmness)	평온 (Calmness)
32	기쁨 (Delight)	슬픔 (Sadness)
33	행복 (Happiness)	만족 (Contentment)
23	행복 (Happiness)	평온 (Calmness)
17	기쁨 (Delight)	슬픔 (Sadness)
18	행복 (Happiness)	슬픔 (Sadness)
30	해당하는 감정이 없음(None of These)	우울 (Depression)
8	행복 (Happiness)	슬픔 (Sadness)
13	평온 (Calmness)	평온 (Calmness)
34	평온 (Calmness)	슬픔 (Sadness)
24	행복 (Happiness)	슬픔 (Sadness)
16	기쁨 (Delight)	슬픔 (Sadness)
3	평온 (Calmness)	슬픔 (Sadness)
4	평온 (Calmness)	우울 (Depression)
7	해당하는 감정이 없음(None of These)	해당하는 감정이 없음(None of These)
10	행복 (Happiness)	평온 (Calmness)

	Emotion_10	Total_Diff
5	기쁨 (Delight)	15.571429
19	만족 (Contentment)	15.828571
9	행복 (Happiness)	16.685714
2	평온 (Calmness)	17.371429
12	즐거움 (Pleasure)	17.542857
20	해당하는 감정이 없음(None of These)	17.685714
14	졸림 (Sleepiness)	17.771429
21	행복 (Happiness)	18.485714
6	행복 (Happiness)	19.714286
11	만족 (Contentment)	19.714286
25	즐거움 (Pleasure)	20.028571
15	만족 (Contentment)	20.200000
31	즐거움 (Pleasure)	20.228571
1	행복 (Happiness)	21.171429
29	행복 (Happiness)	22.228571
28	기쁨 (Delight)	22.228571
27	평온 (Calmness)	22.914286
0	즐거움 (Pleasure)	23.142857
26	즐거움 (Pleasure)	24.057143
22	즐거움 (Pleasure)	25.200000
32	행복 (Happiness)	25.571429
33	즐거움 (Pleasure)	26.600000
23	기쁨 (Delight)	26.800000
17	평온 (Calmness)	27.000000
18	즐거움 (Pleasure)	27.000000
30	평온 (Calmness)	28.542857
8	기쁨 (Delight)	28.914286
13	해당하는 감정이 없음(None of These)	30.400000
34	졸림 (Sleepiness)	33.000000
24	즐거움 (Pleasure)	33.028571
16	즐거움 (Pleasure)	33.771429
3	평온 (Calmness)	35.028571
4	평온 (Calmness)	38.257143
7	해당하는 감정이 없음(None of These)	40.600000
10	즐거움 (Pleasure)	41.314286

[35 rows x 32 columns]

```
---------------------------------------------------------------------------
PermissionError                           Traceback (most recent call last)
Cell In[23], line 54
     51 # Display the sorted results
     52 print(final_df_sorted)
---> 54 final_df_sorted.to_excel('final_df_sorted.xlsx', index=False)

File c:\Users\kgty\anaconda3\Lib\site-packages\pandas\core\generic.py:2252, in NDFrame.to_excel(self, excel_writer, sheet_name, na_rep, float_format, columns, header, index, index_label, startrow, startcol, engine, merge_cells, inf_rep, freeze_panes, storage_options)
   2239 from pandas.io.formats.excel import ExcelFormatter
   2241 formatter = ExcelFormatter(
   2242     df,
   2243     na_rep=na_rep,
   (...)
   2250     inf_rep=inf_rep,
   2251 )
-> 2252 formatter.write(
   2253     excel_writer,
   2254     sheet_name=sheet_name,
   2255     startrow=startrow,
   2256     startcol=startcol,
   2257     freeze_panes=freeze_panes,
   2258     engine=engine,
   2259     storage_options=storage_options,
   2260 )

File c:\Users\kgty\anaconda3\Lib\site-packages\pandas\io\formats\excel.py:934, in ExcelFormatter.write(self, writer, sheet_name, startrow, startcol, freeze_panes, engine, storage_options)
    930     need_save = False
    931 else:
    932     # error: Cannot instantiate abstract class 'ExcelWriter' with abstract
    933     # attributes 'engine', 'save', 'supported_extensions' and 'write_cells'
--> 934     writer = ExcelWriter(  # type: ignore[abstract]
    935         writer, engine=engine, storage_options=storage_options
    936     )
    937     need_save = True
    939 try:

File c:\Users\kgty\anaconda3\Lib\site-packages\pandas\io\excel\_openpyxl.py:60, in OpenpyxlWriter.__init__(self, path, engine, date_format, datetime_format, mode, storage_options, if_sheet_exists, engine_kwargs, **kwargs)
     56 from openpyxl.workbook import Workbook
     58 engine_kwargs = combine_kwargs(engine_kwargs, kwargs)
---> 60 super().__init__(
     61     path,
     62     mode=mode,
     63     storage_options=storage_options,
     64     if_sheet_exists=if_sheet_exists,
     65     engine_kwargs=engine_kwargs,
     66 )
     68 # ExcelWriter replaced "a" by "r+" to allow us to first read the excel file from
     69 # the file and later write to it
     70 if "r+" in self._mode:  # Load from existing workbook
```

```
File c:\Users\kgty\anaconda3\Lib\site-packages\pandas\io\excel\_base.py:1219, in ExcelWriter.__init__(self, path, engine, date_format, datetime_format, mode, storage_options, if_sheet_exists, engine_kwargs)
   1215 self._handles = IOHandles(
   1216     cast(IO[bytes], path), compression={"compression": None}
   1217 )
   1218 if not isinstance(path, ExcelWriter):
-> 1219     self._handles = get_handle(
   1220         path, mode, storage_options=storage_options, is_text=False
   1221     )
   1222 self._cur_sheet = None
   1224 if date_format is None:

File c:\Users\kgty\anaconda3\Lib\site-packages\pandas\io\common.py:868, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    859         handle = open(
    860             handle,
    861             ioargs.mode,
   (...)
    864             newline="",
    865         )
    866     else:
    867         # Binary mode
--> 868         handle = open(handle, ioargs.mode)
    869     handles.append(handle)
    871 # Convert BytesIO or file objects passed with an encoding

PermissionError: [Errno 13] Permission denied: 'final_df_sorted.xlsx'
```

In [ ]:

In [ ]:
```
# Selecting Valence, Arousal, and Emotion columns
valence_columns = [col for col in df_filtered.columns if 'Valence_After_Movie_'
arousal_columns = [col for col in df_filtered.columns if 'Arousal_After_Movie_'
emotion_columns = [col for col in df_filtered.columns if 'Emotion_Keyword_' in c

# Select necessary columns for the analysis
df_selected = df_filtered[['전화번호'] + valence_columns + arousal_columns + emo

# Loop through each person to create a table for each
for person in df_selected['전화번호'].unique():
    # Filter the data for the current person
    person_data = df_selected[df_selected['전화번호'] == person]

    # Initialize a list to store the result for this person
    result_list = []

    # Loop through each movie (10 movies)
    for i in range(10):
        valence = person_data[valence_columns[i]].values[0]
        arousal = person_data[arousal_columns[i]].values[0]
        emotion = person_data[emotion_columns[i]].values[0]

        # Add the result to the list along with the phone number
        result_list.append({
            'Movie': i+1,
            'Valence': valence,
            'Arousal': arousal,
            'Emotion': emotion,
```

```
            '전화번호': person  # Add the phone number here
        })

    # Convert the result list to a DataFrame for the current person
    result_df = pd.DataFrame(result_list)

    # Display the DataFrame for the current person
    print(result_df)
```

	Movie	Valence	Arousal	Emotion	전화번호
0	1	5	5	평온 (Calmness)	1049468170
1	2	5	5	평온 (Calmness)	1049468170
2	3	7	6	만족 (Contentment)	1049468170
3	4	7	7	만족 (Contentment)	1049468170
4	5	2	6	놀람 (Astonishment)	1049468170
5	6	5	5	평온 (Calmness)	1049468170
6	7	2	4	우울 (Depression)	1049468170
7	8	6	4	즐거움 (Pleasure)	1049468170
8	9	6	3	만족 (Contentment)	1049468170
9	10	7	5	즐거움 (Pleasure)	1049468170

	Movie	Valence	Arousal	Emotion	전화번호
0	1	6	3	행복 (Happiness)	1051350406
1	2	7	5	즐거움 (Pleasure)	1051350406
2	3	8	7	기쁨 (Delight)	1051350406
3	4	8	4	평온 (Calmness)	1051350406
4	5	2	7	짜증 (Annoyance)	1051350406
5	6	3	5	짜증 (Annoyance)	1051350406
6	7	3	3	우울 (Depression)	1051350406
7	8	6	3	행복 (Happiness)	1051350406
8	9	2	4	우울 (Depression)	1051350406
9	10	7	6	행복 (Happiness)	1051350406

	Movie	Valence	Arousal	Emotion	전화번호
0	1	6	2	평온 (Calmness)	1022307441
1	2	6	6	해당하는 감정이 없음(None of These)	1022307441
2	3	8	6	기쁨 (Delight)	1022307441
3	4	6	2	평온 (Calmness)	1022307441
4	5	4	7	두려움 (Fear)	1022307441
5	6	4	6	짜증 (Annoyance)	1022307441
6	7	3	3	우울 (Depression)	1022307441
7	8	7	3	행복 (Happiness)	1022307441
8	9	5	4	슬픔 (Sadness)	1022307441
9	10	7	6	기쁨 (Delight)	1022307441

	Movie	Valence	Arousal	Emotion	전화번호
0	1	5	1	평온 (Calmness)	1067510885
1	2	5	3	슬픔 (Sadness)	1067510885
2	3	7	6	즐거움 (Pleasure)	1067510885
3	4	7	5	행복 (Happiness)	1067510885
4	5	2	7	짜증 (Annoyance)	1067510885
5	6	5	5	짜증 (Annoyance)	1067510885
6	7	3	3	우울 (Depression)	1067510885
7	8	7	5	만족 (Contentment)	1067510885
8	9	4	5	슬픔 (Sadness)	1067510885
9	10	7	6	행복 (Happiness)	1067510885

	Movie	Valence	Arousal	Emotion	전화번호
0	1	5	1	평온 (Calmness)	1027642730
1	2	5	1	평온 (Calmness)	1027642730
2	3	7	2	해당하는 감정이 없음(None of These)	1027642730
3	4	6	2	평온 (Calmness)	1027642730
4	5	1	7	놀람 (Astonishment)	1027642730
5	6	3	5	우울 (Depression)	1027642730
6	7	3	4	해당하는 감정이 없음(None of These)	1027642730
7	8	7	3	행복 (Happiness)	1027642730
8	9	5	1	슬픔 (Sadness)	1027642730
9	10	7	3	행복 (Happiness)	1027642730

	Movie	Valence	Arousal	Emotion	전화번호
0	1	5	3	평온 (Calmness)	1064186860
1	2	5	3	해당하는 감정이 없음(None of These)	1064186860
2	3	5	3	즐거움 (Pleasure)	1064186860
3	4	5	3	해당하는 감정이 없음(None of These)	1064186860

4	5	1	7	두려움 (Fear)	1064186860
5	6	2	6	짜증 (Annoyance)	1064186860
6	7	3	5	우울 (Depression)	1064186860
7	8	7	3	기쁨 (Delight)	1064186860
8	9	4	3	좌절 (Frustration)	1064186860
9	10	6	3	즐거움 (Pleasure)	1064186860

	Movie	Valence	Arousal	Emotion	전화번호
0	1	7	5	평온 (Calmness)	1072325997
1	2	7	5	졸림 (Sleepiness)	1072325997
2	3	7	5	즐거움 (Pleasure)	1072325997
3	4	5	4	즐거움 (Pleasure)	1072325997
4	5	3	6	짜증 (Annoyance)	1072325997
5	6	3	5	슬픔 (Sadness)	1072325997
6	7	3	1	슬픔 (Sadness)	1072325997
7	8	8	3	평온 (Calmness)	1072325997
8	9	6	3	행복 (Happiness)	1072325997
9	10	7	5	졸림 (Sleepiness)	1072325997

	Movie	Valence	Arousal	Emotion	전화번호
0	1	5	5	평온 (Calmness)	1023732291
1	2	5	6	평온 (Calmness)	1023732291
2	3	7	6	행복 (Happiness)	1023732291
3	4	7	2	만족 (Contentment)	1023732291
4	5	3	7	졸림 (Sleepiness)	1023732291
5	6	3	6	짜증 (Annoyance)	1023732291
6	7	1	1	좌절 (Frustration)	1023732291
7	8	8	3	행복 (Happiness)	1023732291
8	9	5	1	해당하는 감정이 없음(None of These)	1023732291
9	10	7	5	만족 (Contentment)	1023732291

	Movie	Valence	Arousal	Emotion	전화번호
0	1	6	1	평온 (Calmness)	1029982403
1	2	6	1	평온 (Calmness)	1029982403
2	3	8	7	즐거움 (Pleasure)	1029982403
3	4	6	3	평온 (Calmness)	1029982403
4	5	1	8	두려움 (Fear)	1029982403
5	6	3	6	짜증 (Annoyance)	1029982403
6	7	5	2	우울 (Depression)	1029982403
7	8	7	4	행복 (Happiness)	1029982403
8	9	6	2	슬픔 (Sadness)	1029982403
9	10	7	6	행복 (Happiness)	1029982403

	Movie	Valence	Arousal	Emotion	전화번호
0	1	6	1	평온 (Calmness)	104235208
1	2	6	1	평온 (Calmness)	104235208
2	3	8	7	즐거움 (Pleasure)	104235208
3	4	6	4	평온 (Calmness)	104235208
4	5	4	6	두려움 (Fear)	104235208
5	6	3	7	짜증 (Annoyance)	104235208
6	7	5	4	평온 (Calmness)	104235208
7	8	7	3	만족 (Contentment)	104235208
8	9	8	3	행복 (Happiness)	104235208
9	10	8	7	행복 (Happiness)	104235208

	Movie	Valence	Arousal	Emotion	전화번호
0	1	5	1	해당하는 감정이 없음(None of These)	1044575769
1	2	5	5	해당하는 감정이 없음(None of These)	1044575769
2	3	8	7	즐거움 (Pleasure)	1044575769
3	4	6	3	평온 (Calmness)	1044575769
4	5	2	8	두려움 (Fear)	1044575769
5	6	2	7	짜증 (Annoyance)	1044575769
6	7	1	7	짜증 (Annoyance)	1044575769
7	8	5	5	해당하는 감정이 없음(None of These)	1044575769

```
8      9        4        6              우울 (Depression)  1044575769
9     10        6        4                평온 (Calmness)  1044575769
```

In [ ]:

In [ ]:

In [ ]:
```python
# Selecting Valence, Arousal, and Emotion columns
valence_columns = [col for col in df_filtered.columns if 'Valence_After_Movie_'
arousal_columns = [col for col in df_filtered.columns if 'Arousal_After_Movie_'
emotion_columns = [col for col in df_filtered.columns if 'Emotion_Keyword_' in c

# Select necessary columns for the analysis
df_selected = df_filtered[['전화번호'] + valence_columns + arousal_columns + emo

# Add valence, arousal, and emotion for each movie
for i in range(10):
    df_selected[f'valence_{i+1}'] = df_selected[valence_columns[i]].copy()
    df_selected[f'arousal_{i+1}'] = df_selected[arousal_columns[i]].copy()
    df_selected[f'emotion_{i+1}'] = df_selected[emotion_columns[i]].copy()

# Get unique participants
people = df_selected['전화번호'].unique()

# Plot for each person
for person in people:
    try:
        # Extract data for the person
        person_data = df_selected[df_selected['전화번호'] == person]

        plt.figure(figsize=(10, 6))

        # Plot each movie's valence and arousal
        for i in range(10):
            valence = person_data[f'valence_{i+1}'].values[0]
            arousal = person_data[f'arousal_{i+1}'].values[0]
            emotion = person_data[f'emotion_{i+1}'].values[0]  # 감정 키워드

            # Plot each movie's point with labels
            plt.scatter(valence, arousal, label=f'Movie {i+1}', color='blue')
            plt.text(valence + 0.2, arousal + 0.2, f'{i+1}\n{emotion}', fontsize

        # Add labels and title
        plt.title(f'{person}의 10개 영화에 대한 감정 분석')
        plt.xlabel('Valence')
        plt.ylabel('Arousal')

        # Set limits and remove grid
        plt.xlim(0, 10)
        plt.ylim(0, 10)
        plt.grid(False)

        # Draw quadrant axes
        plt.axhline(y=5, color='black', linewidth=1.5)  # Horizontal mid-line
        plt.axvline(x=5, color='black', linewidth=1.5)  # Vertical mid-line

        plt.show()

    except Exception as e:
```

```
        print(f"Error while processing {person}: {e}")
        continue
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_7264\1955633905.py in <module>
      2 arousal_columns = [col for col in df.columns if '이 영화를 보고난 후 흥
분(Excited)되거나 긴장되었다' in col]  # Excitement columns
      3
----> 4 df_selected = df[['Submission ID'] + valence_columns + arousal_columns]
      5
      6 for i in range(10):

c:\Users\kgty\anaconda3\envs\kocca\lib\site-packages\pandas\core\frame.py in __ge
titem__(self, key)
   3462             if is_iterator(key):
   3463                 key = list(key)
-> 3464             indexer = self.loc._get_listlike_indexer(key, axis=1)[1]
   3465
   3466         # take() does not accept boolean indexers

c:\Users\kgty\anaconda3\envs\kocca\lib\site-packages\pandas\core\indexing.py in _
get_listlike_indexer(self, key, axis)
   1312             keyarr, indexer, new_indexer = ax._reindex_non_unique(keyarr)
   1313
-> 1314         self._validate_read_indexer(keyarr, indexer, axis)
   1315
   1316         if needs_i8_conversion(ax.dtype) or isinstance(

c:\Users\kgty\anaconda3\envs\kocca\lib\site-packages\pandas\core\indexing.py in _
validate_read_indexer(self, key, indexer, axis)
   1372                 if use_interval_msg:
   1373                     key = list(key)
-> 1374                 raise KeyError(f"None of [{key}] are in the [{axis_nam
e}]")
   1375
   1376             not_found = list(ensure_index(key)[missing_mask.nonzero()
[0]].unique())

KeyError: "None of [Index(['Submission ID'], dtype='object')] are in the [column
s]"
```

In [ ]:
```
def classify_quadrant(valence, arousal):
    if valence >= 5 and arousal >= 5:
        return 'Top-right'
    elif valence < 5 and arousal >= 5:
        return 'Top-left'
    elif valence >= 5 and arousal < 5:
        return 'Bottom-right'
    else:
        return 'Bottom-left'

movie_quadrants_count = {f'영화 {i+1}': {'Top-right': 0, 'Top-left': 0, 'Bottom-
                         for i in range(10)}

for _, row in df_selected.iterrows():
    for i in range(10):
        valence = row[valence_columns[i]]
        arousal = row[arousal_columns[i]]
```

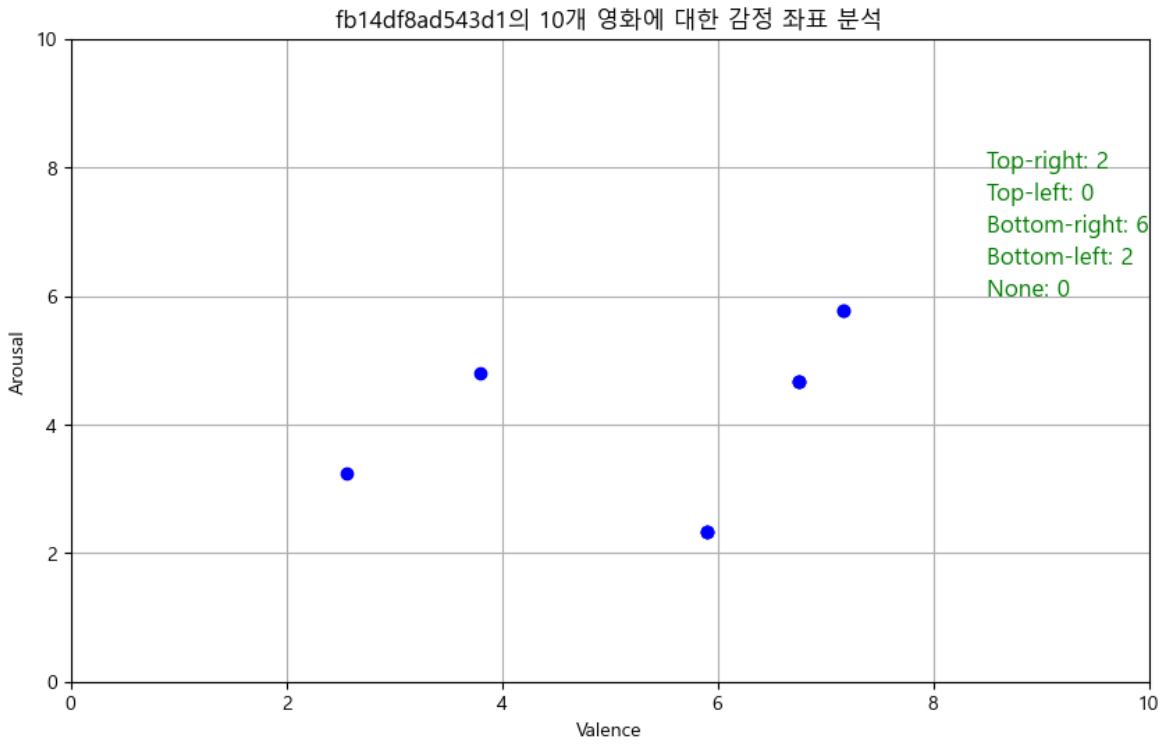
```
        quadrant = classify_quadrant(valence, arousal)

        movie_quadrants_count[f'영화 {i+1}'][quadrant] += 1

sorted_movies = sorted(movie_quadrants_count.items(), key=lambda x: sum(x[1].val

print("영화별 사분면에 포함된 횟수 순위:")
for movie, quadrants in sorted_movies:
    print(f"{movie}: {quadrants}")
```

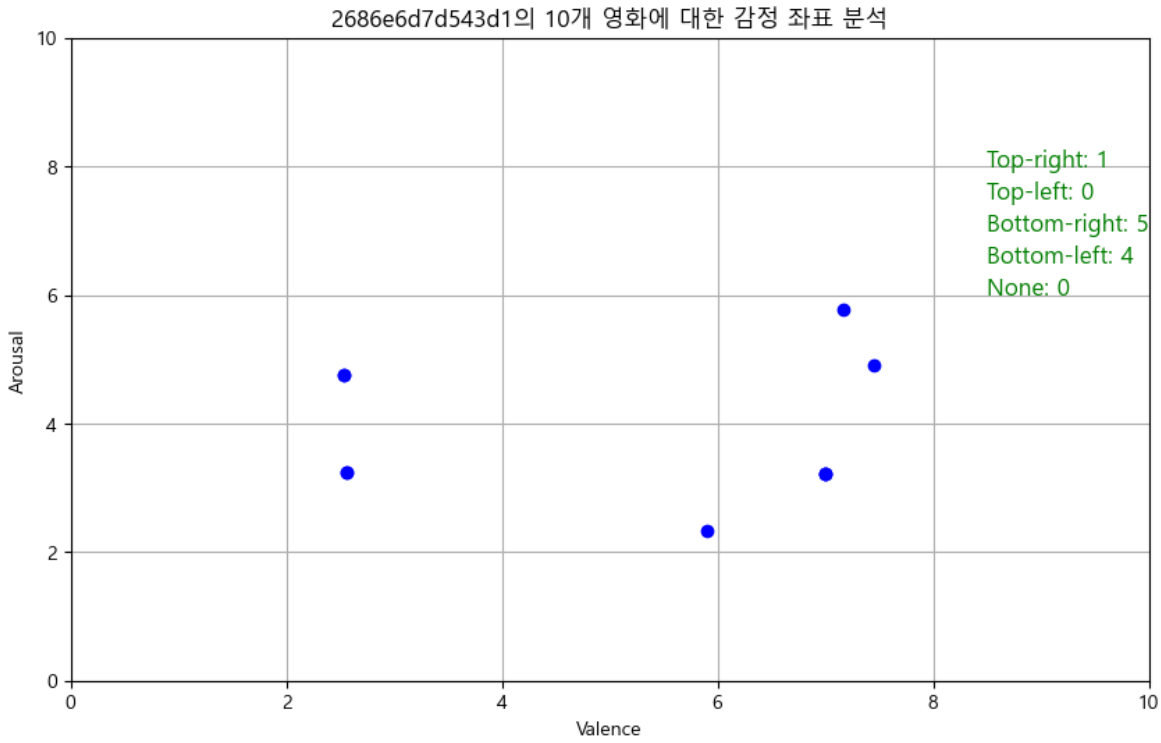
```
영화별 사분면에 포함된 횟수 순위:
영화 1: {'Top-right': 6, 'Top-left': 0, 'Bottom-right': 19, 'Bottom-left': 1}
영화 2: {'Top-right': 6, 'Top-left': 0, 'Bottom-right': 20, 'Bottom-left': 0}
영화 3: {'Top-right': 17, 'Top-left': 1, 'Bottom-right': 6, 'Bottom-left': 2}
영화 4: {'Top-right': 4, 'Top-left': 0, 'Bottom-right': 21, 'Bottom-left': 1}
영화 5: {'Top-right': 1, 'Top-left': 20, 'Bottom-right': 0, 'Bottom-left': 5}
영화 6: {'Top-right': 4, 'Top-left': 16, 'Bottom-right': 0, 'Bottom-left': 6}
영화 7: {'Top-right': 0, 'Top-left': 2, 'Bottom-right': 1, 'Bottom-left': 23}
영화 8: {'Top-right': 2, 'Top-left': 0, 'Bottom-right': 23, 'Bottom-left': 1}
영화 9: {'Top-right': 3, 'Top-left': 2, 'Bottom-right': 14, 'Bottom-left': 7}
영화 10: {'Top-right': 13, 'Top-left': 0, 'Bottom-right': 13, 'Bottom-left': 0}
```

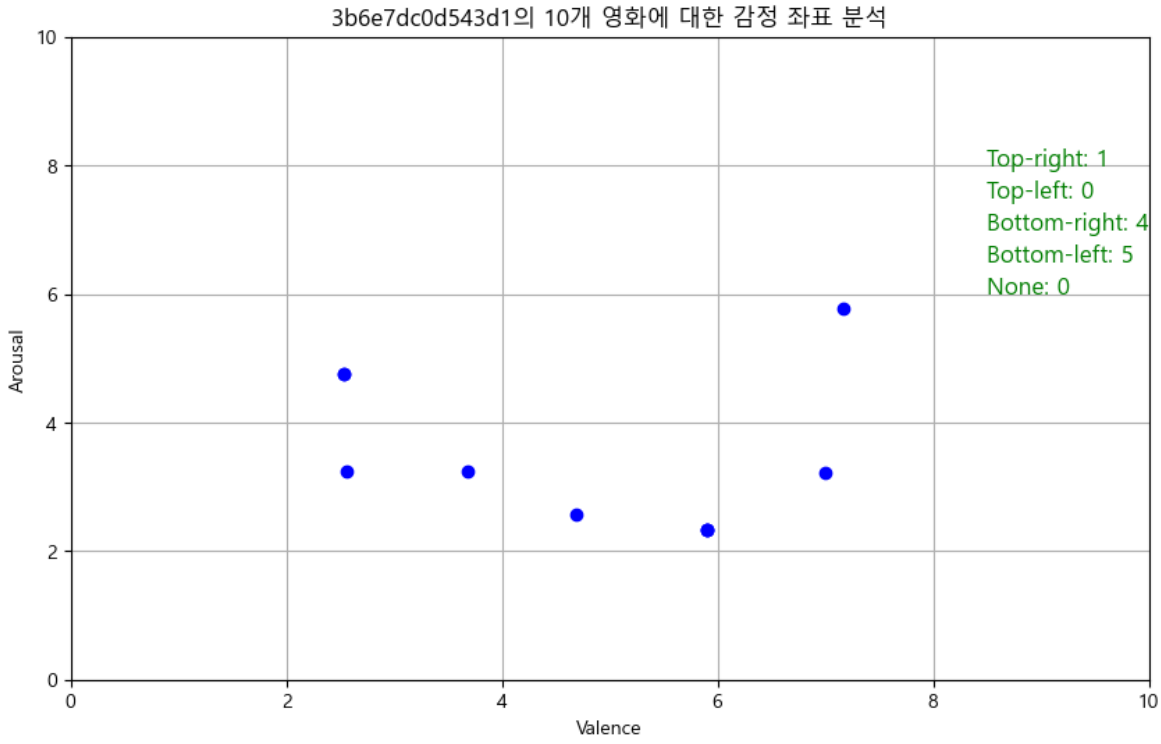
In [ ]:

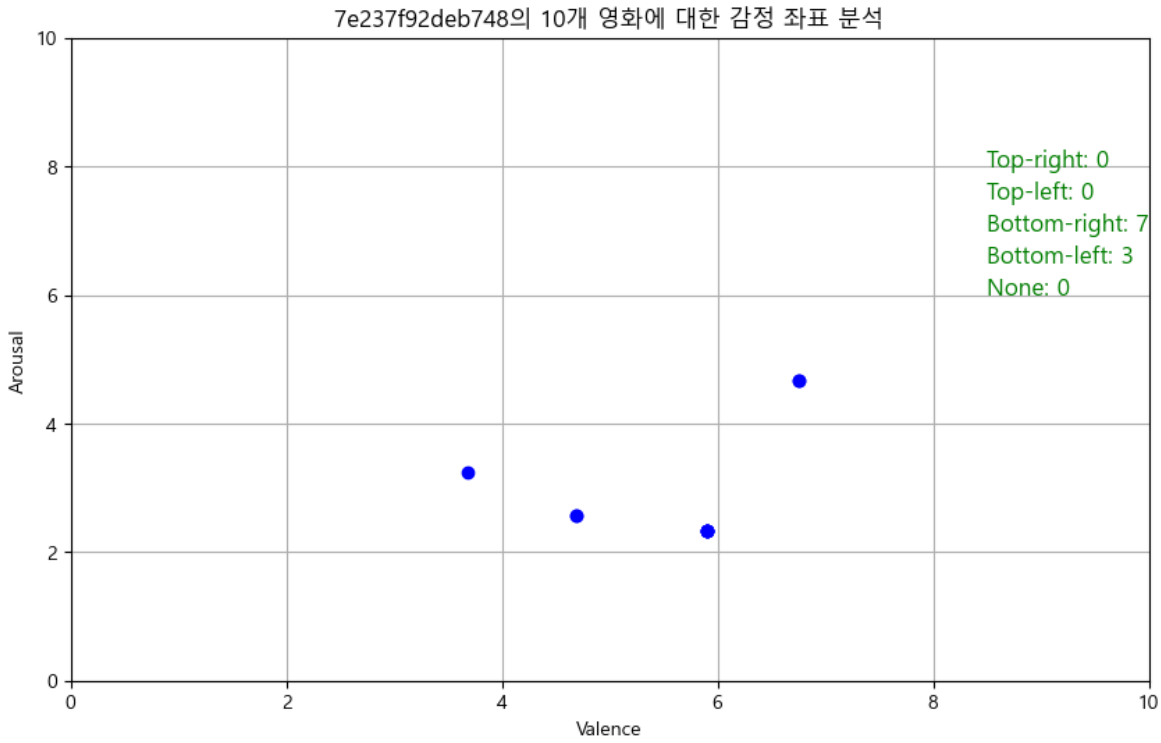
```
emotion_to_coordinates = {
    '기쁨 (Delight)': (7.454545, 4.909091),
    '행복 (Arousal)': (7.000000, 3.210526),
    '즐거움 (Pleasure)': (7.166667, 5.777778),
    '두려움 (Fear)': (1.800000, 6.500000),
    '짜증 (Annoyance)': (2.529412, 4.764706),
    '좌절 (Frustration)': (3.333333, 4.333333),
    '만족 (Contentment)': (6.750000, 4.666667),
    '평온 (Calmness)': (5.909091, 2.340909),
    '놀람 (Astonishment)': (3.800000, 4.800000),
    '슬픔 (Sadness)': (3.687500, 3.250000),
    '우울 (Depression)': (2.562500, 3.250000),
    '피로 (Tiredness)': (2.750000, 3.000000),
    '해당하는 감정이 없음(None of These)': (4.684211, 2.578947)
}

emotion_columns = [col for col in df.columns if '다음 중 감정을 가장 잘 나타내는

df_selected = df[['Submission ID'] + emotion_columns]

for i in range(10):
    df_selected[f'emotion_{i+1}'] = df_selected[emotion_columns[i]].copy()

people = df_selected['Submission ID'].unique()

for person in people:
    try:
        person_data = df_selected[df_selected['Submission ID'] == person]

        plt.figure(figsize=(10, 6))

        quadrants_count = {'Top-right': 0, 'Top-left': 0, 'Bottom-right': 0, 'Bo

        for i in range(10):
            emotion = person_data[f'emotion_{i+1}'].values[0]
            emotion_coordinate = emotion_to_coordinates.get(emotion, (5, 5))
            emotion_valence, emotion_arousal = emotion_coordinate
```

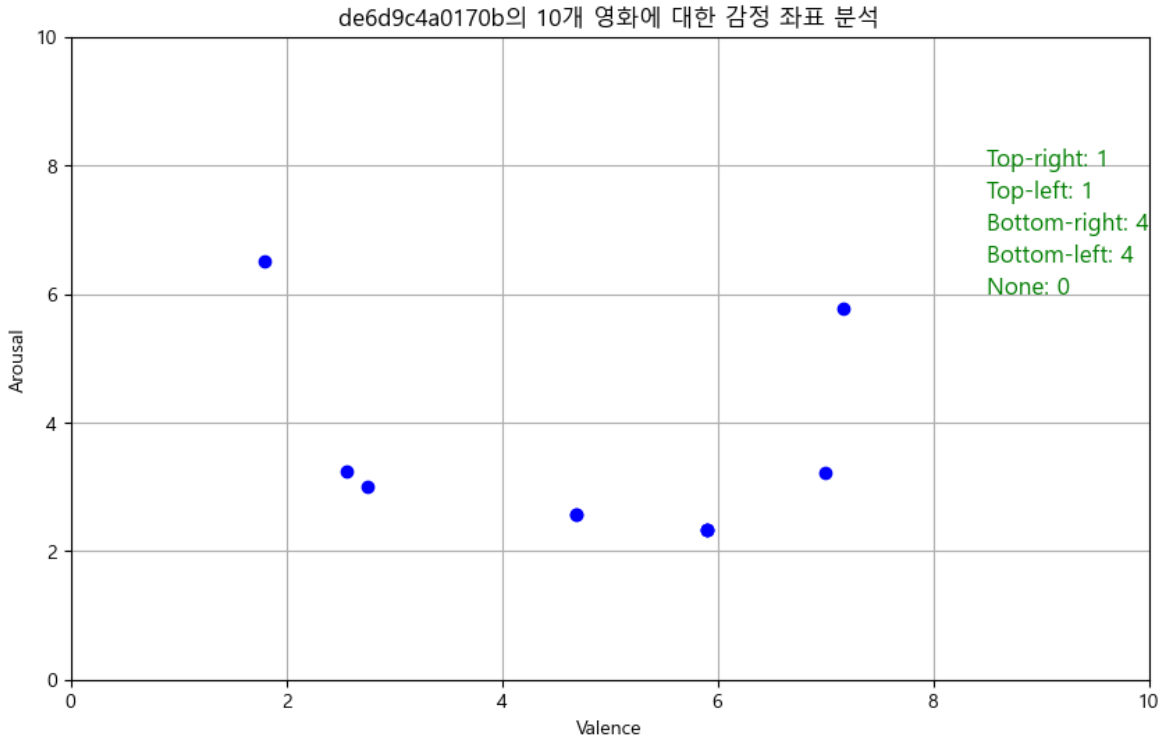
```
            quadrant = 'None'
            if emotion_valence >= 5 and emotion_arousal >= 5:
                quadrant = 'Top-right'
            elif emotion_valence < 5 and emotion_arousal >= 5:
                quadrant = 'Top-left'
            elif emotion_valence >= 5 and emotion_arousal < 5:
                quadrant = 'Bottom-right'
            elif emotion_valence < 5 and emotion_arousal < 5:
                quadrant = 'Bottom-left'

            quadrants_count[quadrant] += 1

            plt.scatter(emotion_valence, emotion_arousal, label=f'영화 {i+1} 감정

        plt.text(8.5, 8, f'Top-right: {quadrants_count["Top-right"]}', fontsize=
        plt.text(8.5, 7.5, f'Top-left: {quadrants_count["Top-left"]}', fontsize=
        plt.text(8.5, 7, f'Bottom-right: {quadrants_count["Bottom-right"]}', fon
        plt.text(8.5, 6.5, f'Bottom-left: {quadrants_count["Bottom-left"]}', fon
        plt.text(8.5, 6, f'None: {quadrants_count["None"]}', fontsize=12, color=

        plt.title(f'{person}의 10개 영화에 대한 감정 좌표 분석')
        plt.xlabel('Valence')
        plt.ylabel('Arousal')
        plt.xlim(0, 10)
        plt.ylim(0, 10)
        plt.grid(True)
        plt.show()

    except Exception as e:
        print(f"Error while processing {person}: {e}")
        continue
```

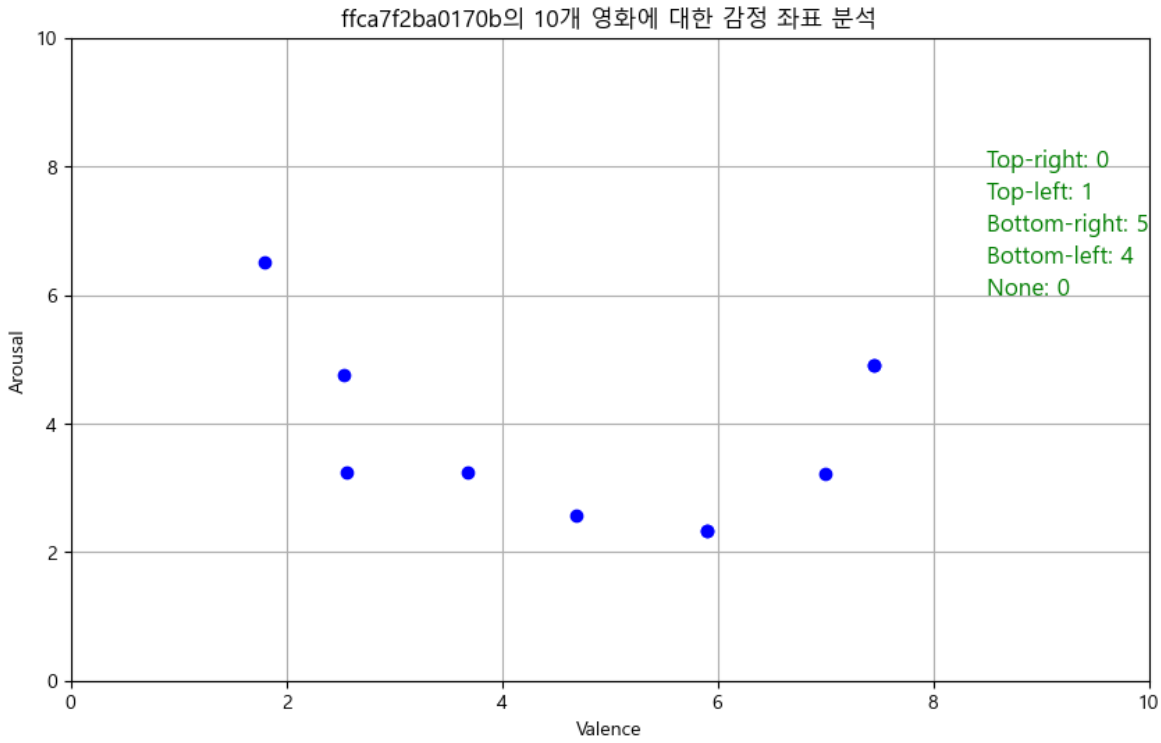


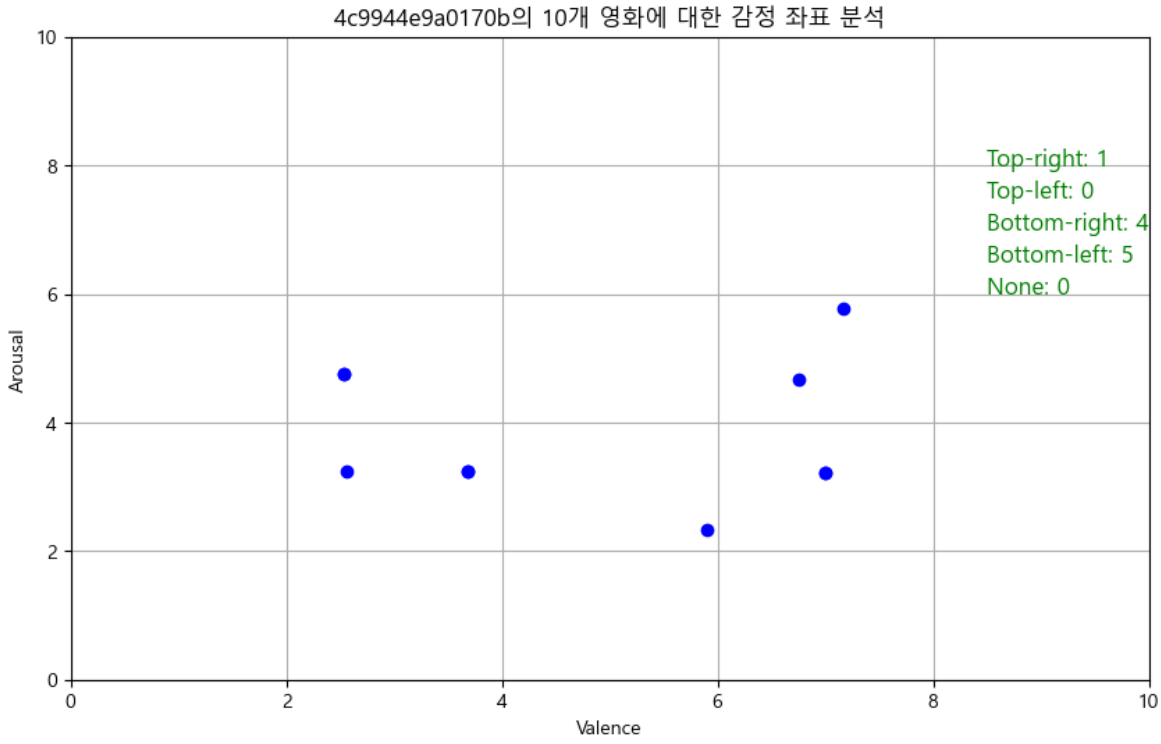

2686e6d7d543d1의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 0
Bottom-right: 5
Bottom-left: 4
None: 0
Arousal
Valence
0
2
4
6
8
10

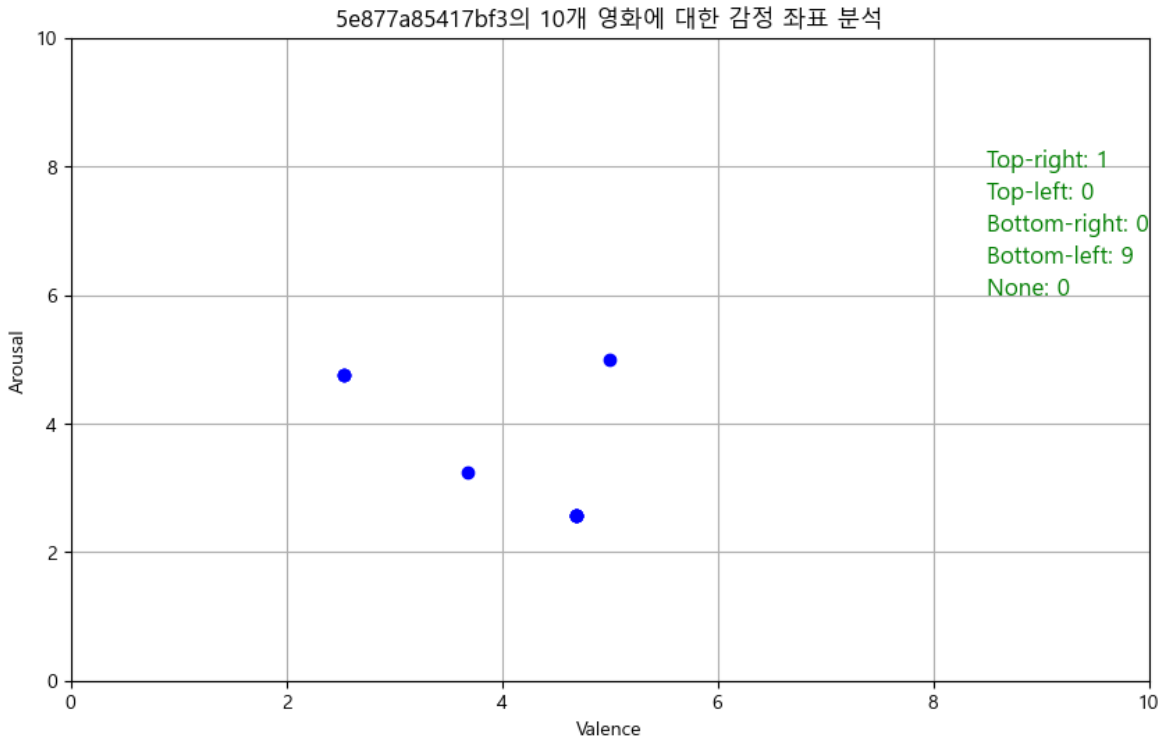

3b6e7dc0d543d1의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 0
Bottom-right: 4
Bottom-left: 5
None: 0
Arousal
Valence
0
2
4
6
8
10

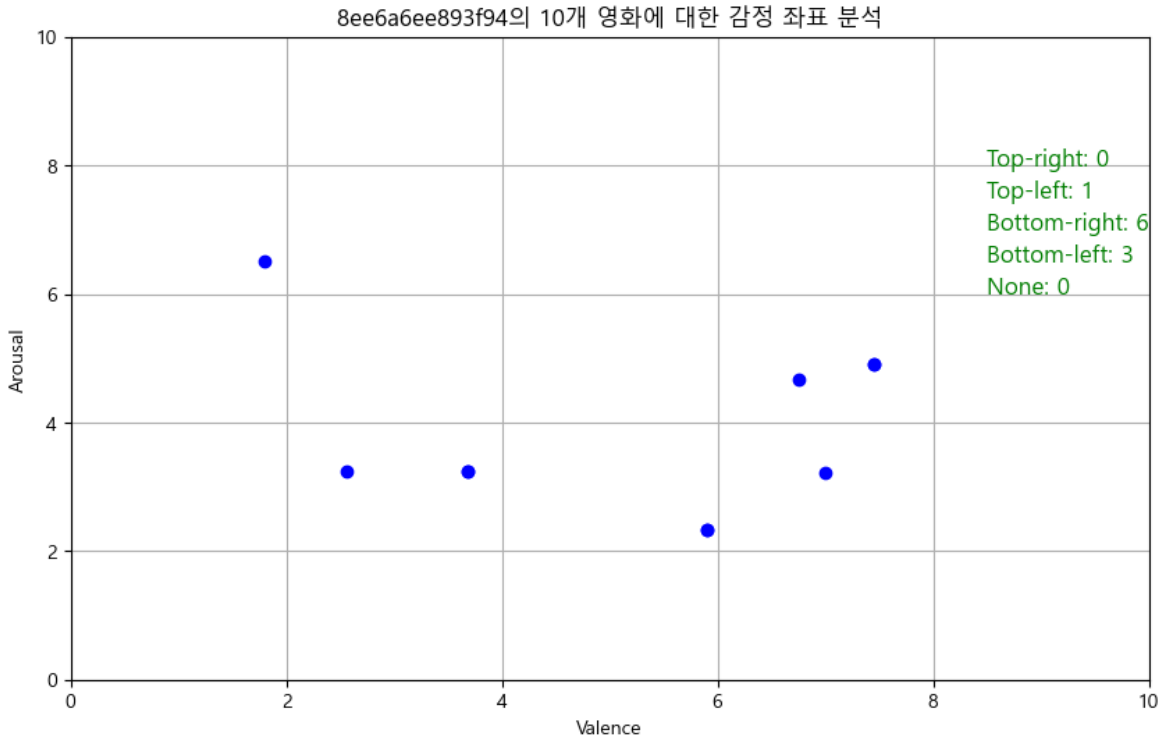

7e237f92deb748의 10개 영화에 대한 감정 좌표 분석
Top-right: 0
Top-left: 0
Bottom-right: 7
Bottom-left: 3
None: 0
Arousal
Valence
0
2
4
6
8
10

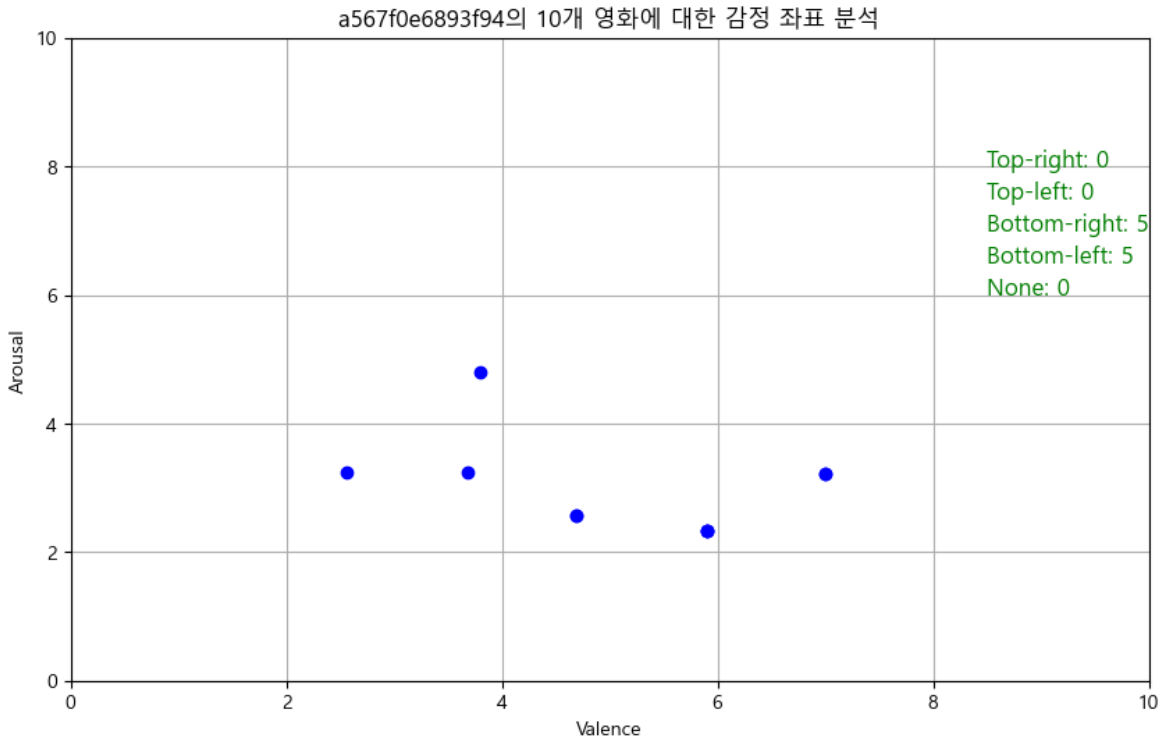

de6d9c4a0170b의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 1
Bottom-right: 4
Bottom-left: 4
None: 0
Arousal
Valence
0
2
4
6
8
10

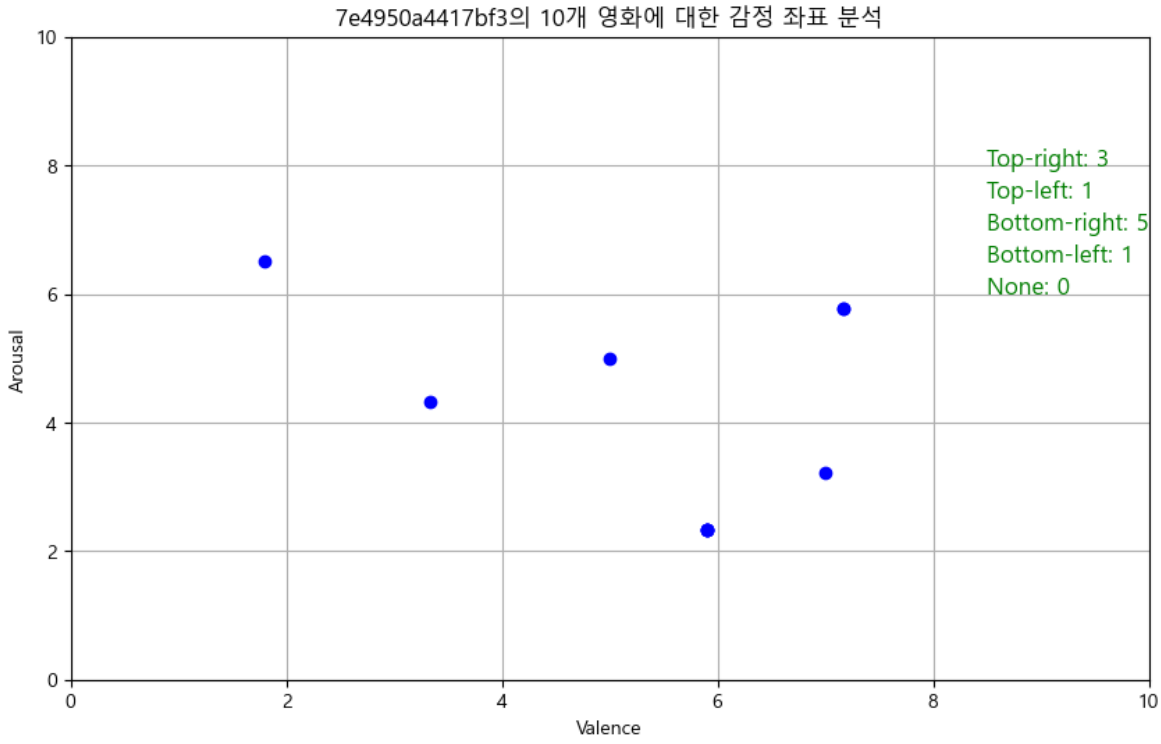

ffca7f2ba0170b의 10개 영화에 대한 감정 좌표 분석
Top-right: 0
Top-left: 1
Bottom-right: 5
Bottom-left: 4
None: 0
Arousal
Valence
0
2
4
6
8
10

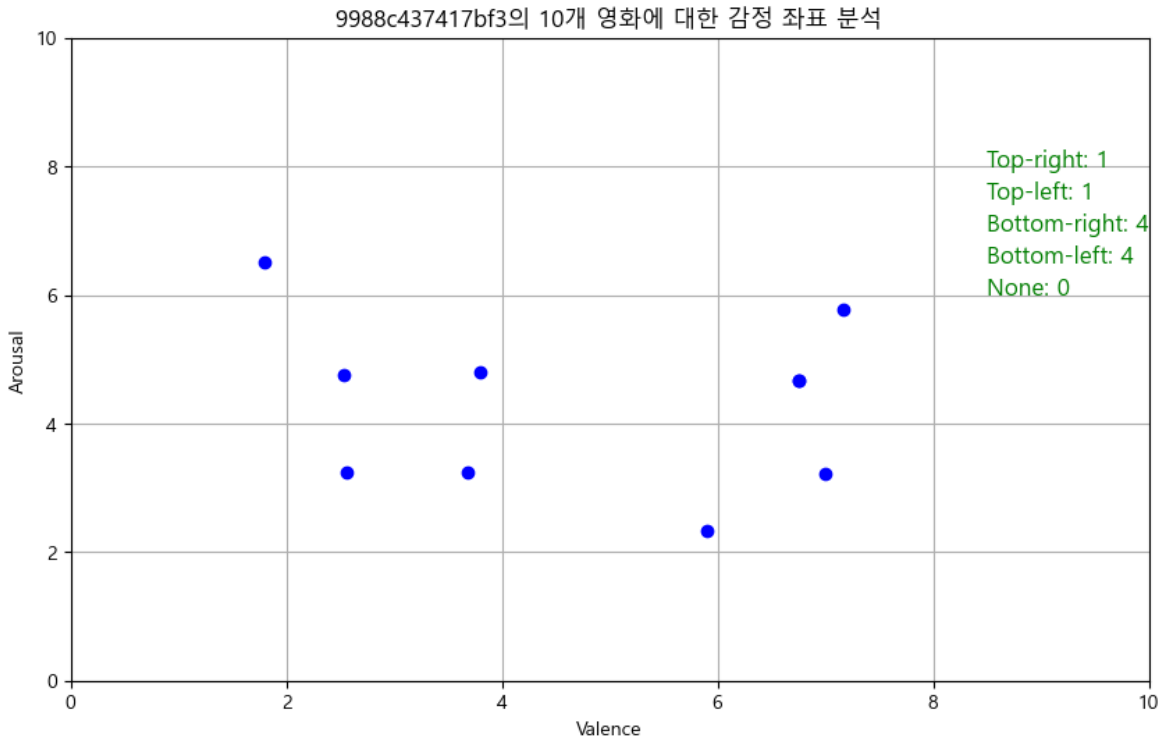

4c9944e9a0170b의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 0
Bottom-right: 4
Bottom-left: 5
None: 0
Arousal
Valence
0
2
4
6
8
10

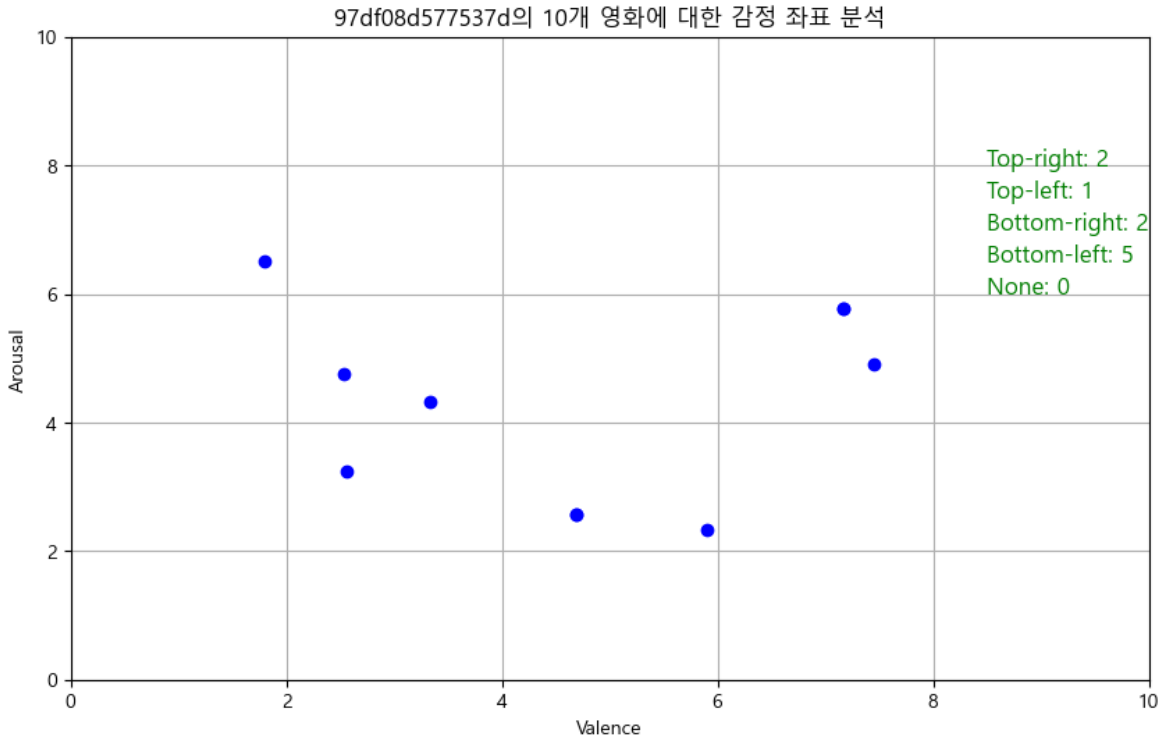

5e877a85417bf3의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 0
Bottom-right: 0
Bottom-left: 9
None: 0
Arousal
Valence
0
2
4
6
8
10

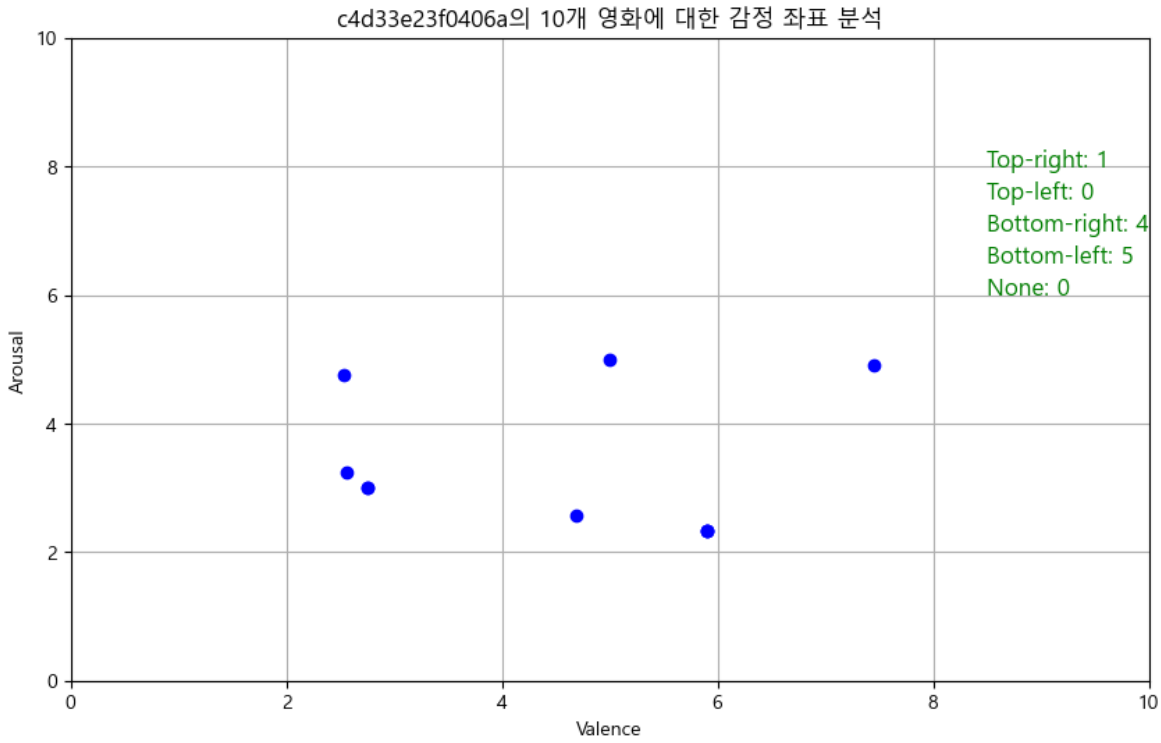

8ee6a6ee893f94의 10개 영화에 대한 감정 좌표 분석
Top-right: 0
Top-left: 1
Bottom-right: 6
Bottom-left: 3
None: 0
Arousal
Valence
0
2
4
6
8
10

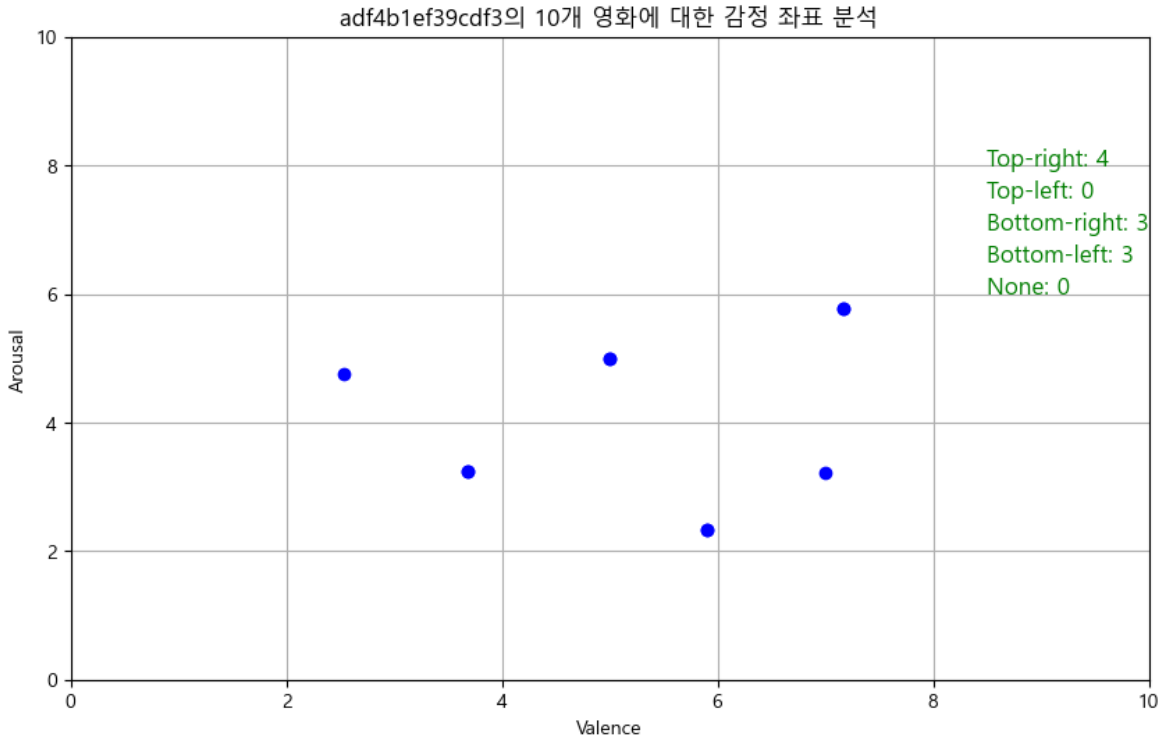

a567f0e6893f94의 10개 영화에 대한 감정 좌표 분석
Top-right: 0
Top-left: 0
Bottom-right: 5
Bottom-left: 5
None: 0
Arousal
Valence
0
2
4
6
8
10

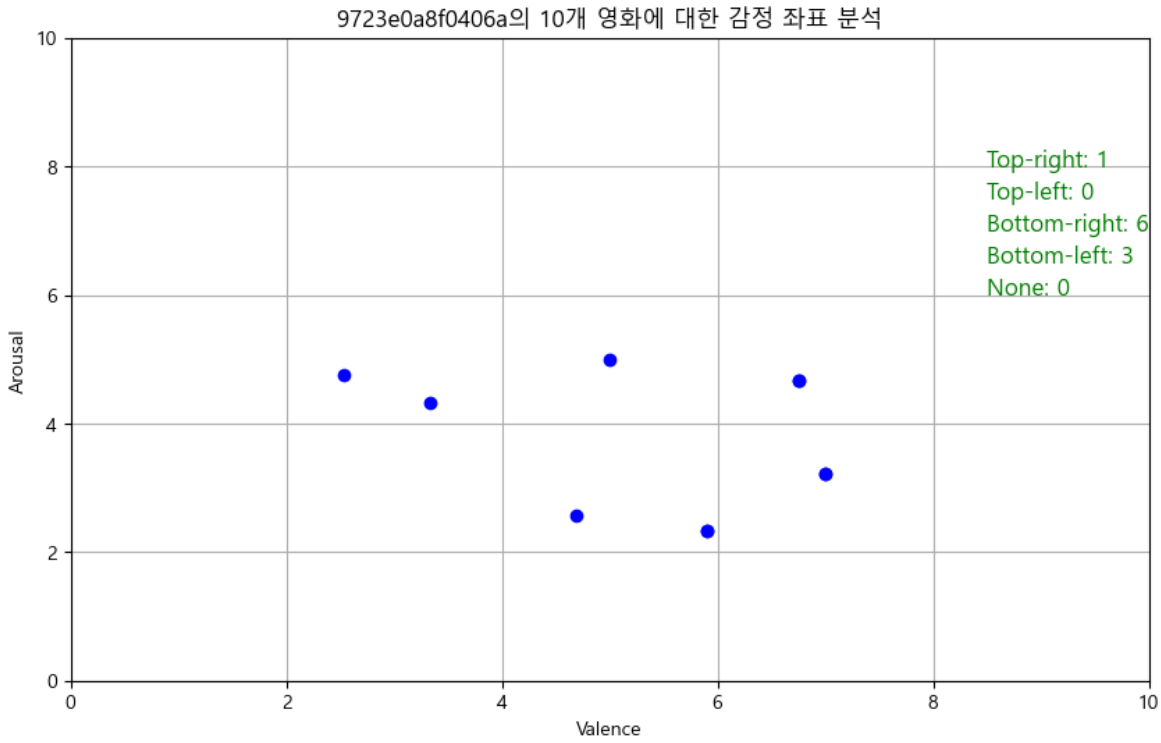

7e4950a4417bf3의 10개 영화에 대한 감정 좌표 분석
Top-right: 3
Top-left: 1
Bottom-right: 5
Bottom-left: 1
None: 0
Arousal
Valence
0
2
4
6
8
10

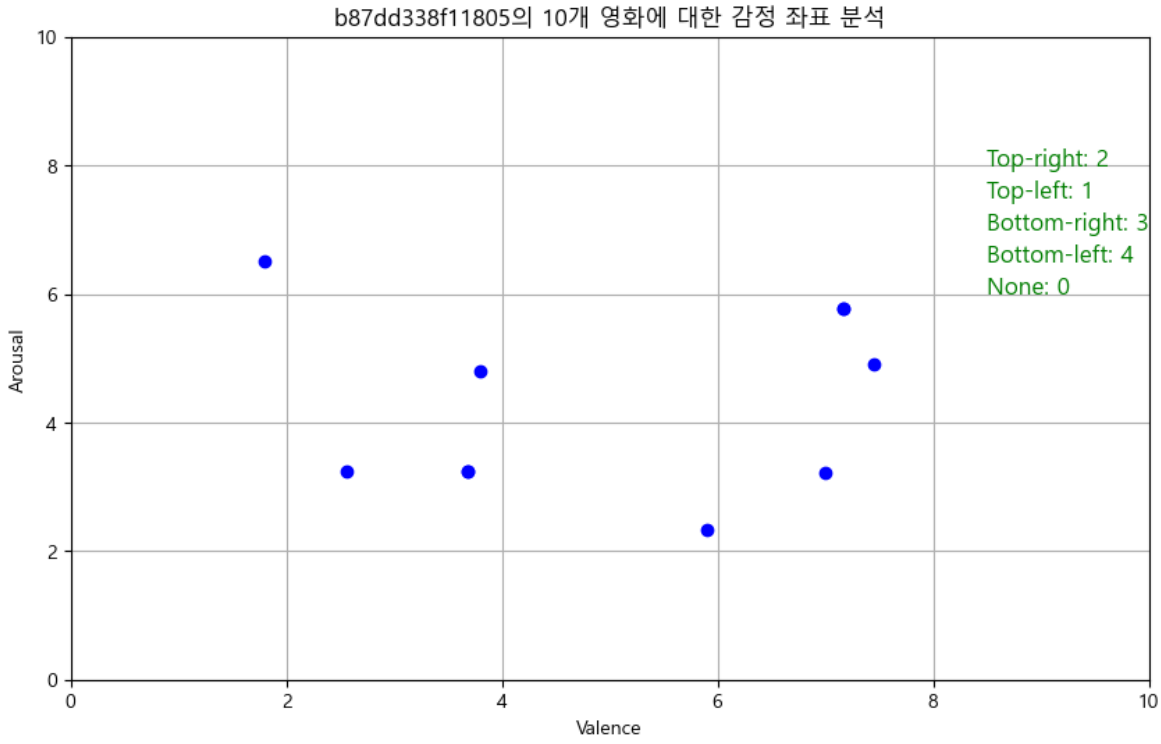

9988c437417bf3의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 1
Bottom-right: 4
Bottom-left: 4
None: 0
Arousal
Valence
0
2
4
6
8
10

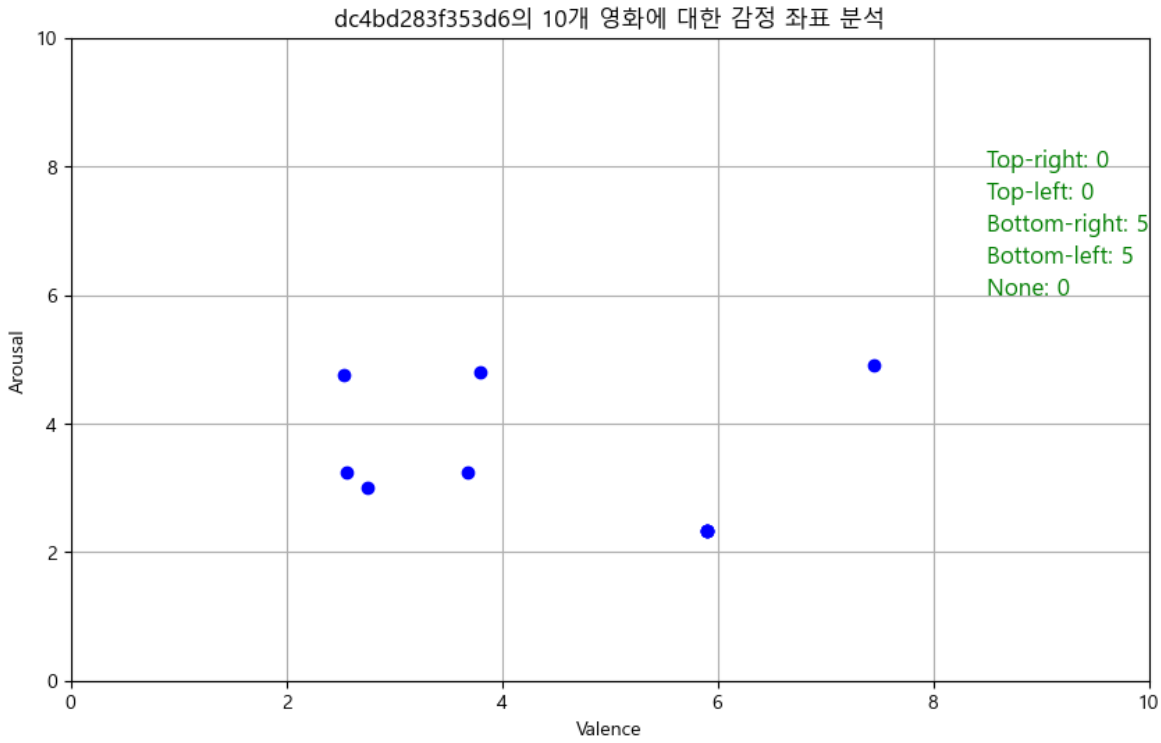

97df08d577537d의 10개 영화에 대한 감정 좌표 분석
Top-right: 2
Top-left: 1
Bottom-right: 2
Bottom-left: 5
None: 0
Arousal
Valence
0
2
4
6
8
10

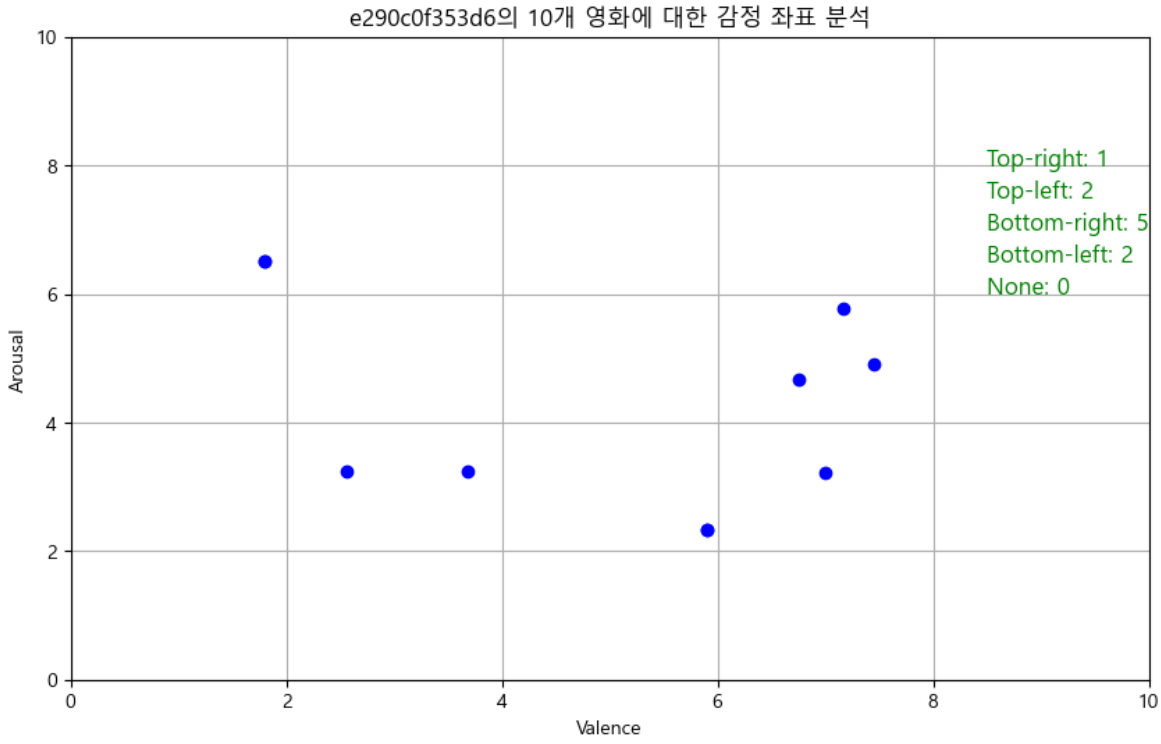

c4d33e23f0406a의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 0
Bottom-right: 4
Bottom-left: 5
None: 0
Arousal
Valence
0
2
4
6
8
10

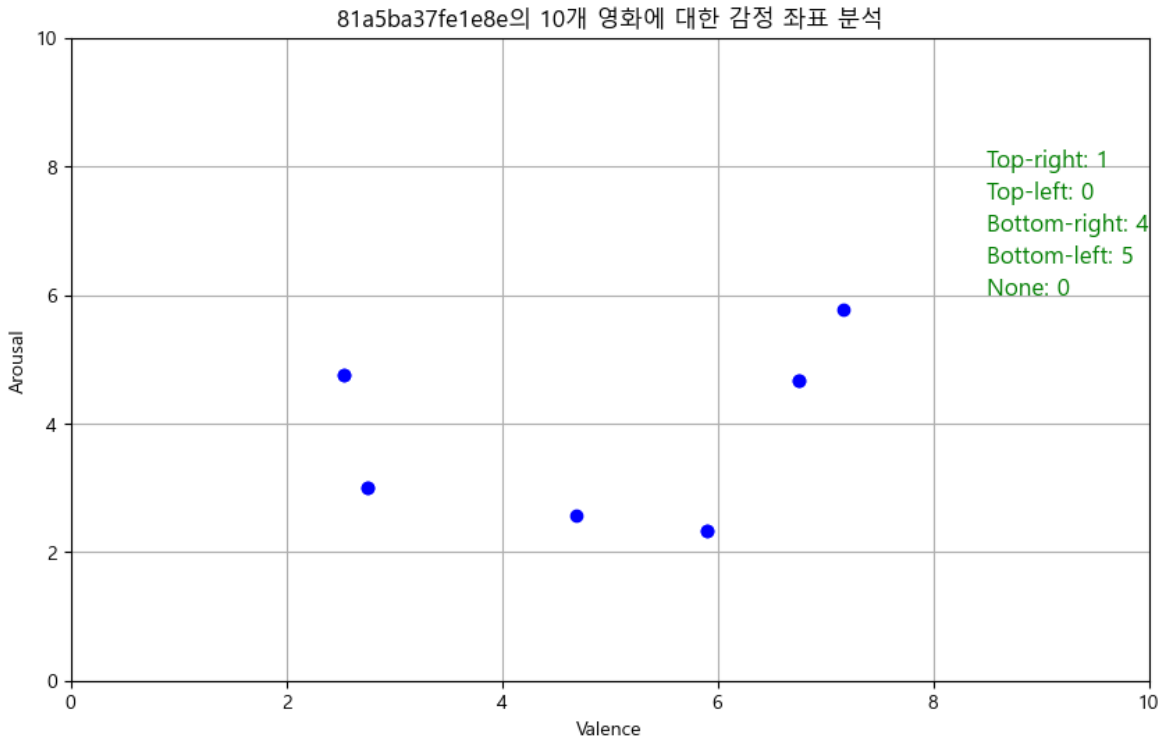

adf4b1ef39cdf3의 10개 영화에 대한 감정 좌표 분석
Top-right: 4
Top-left: 0
Bottom-right: 3
Bottom-left: 3
None: 0
Arousal
Valence
0
2
4
6
8
10

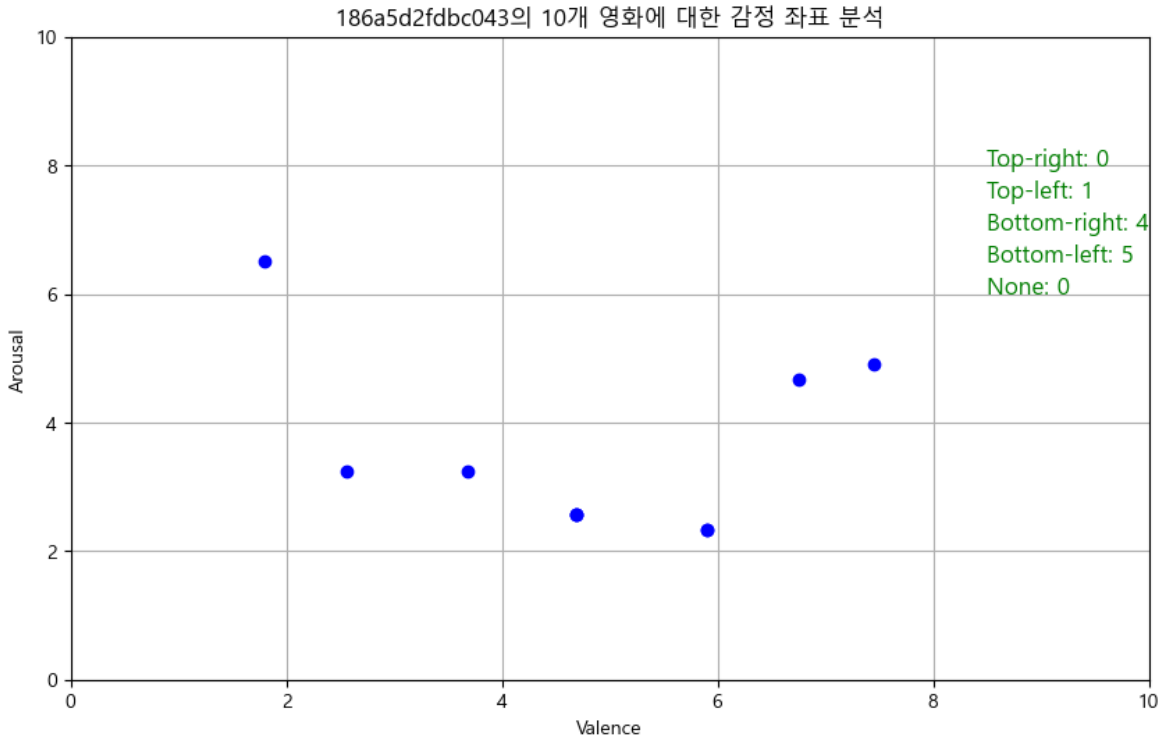

9723e0a8f0406a의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 0
Bottom-right: 6
Bottom-left: 3
None: 0
Arousal
Valence
0
2
4
6
8
10

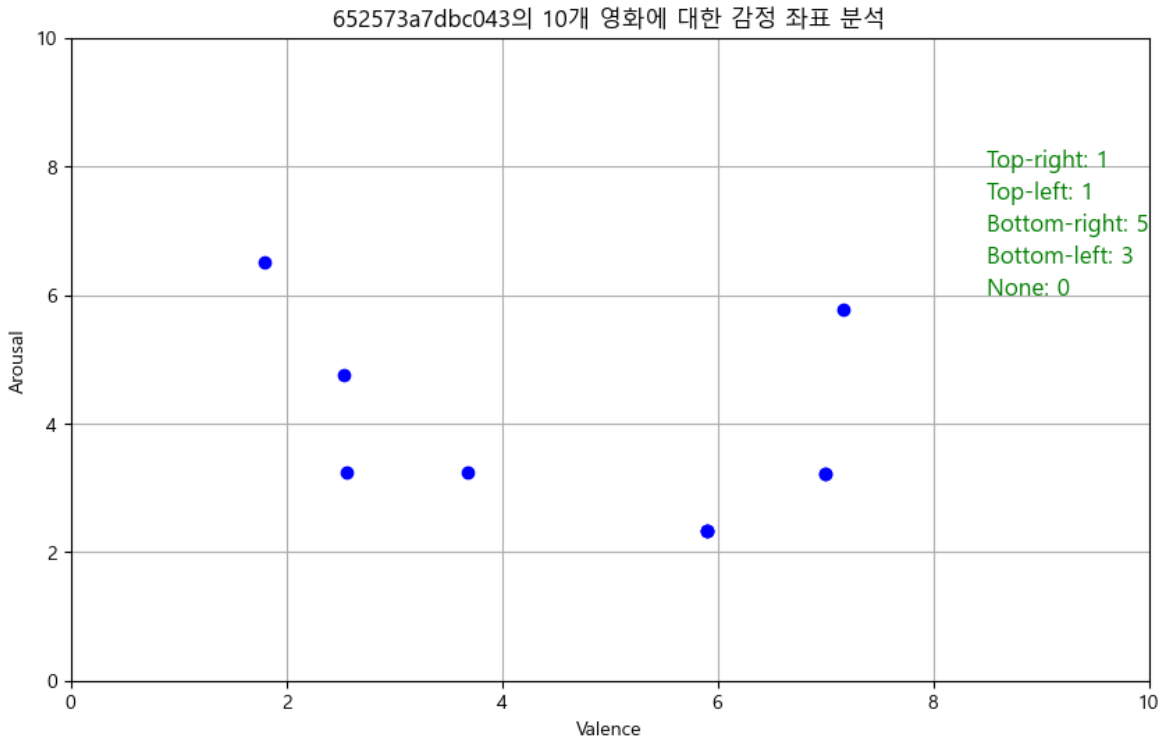

b87dd338f11805의 10개 영화에 대한 감정 좌표 분석
Top-right: 2
Top-left: 1
Bottom-right: 3
Bottom-left: 4
None: 0
Arousal
Valence
0
2
4
6
8
10

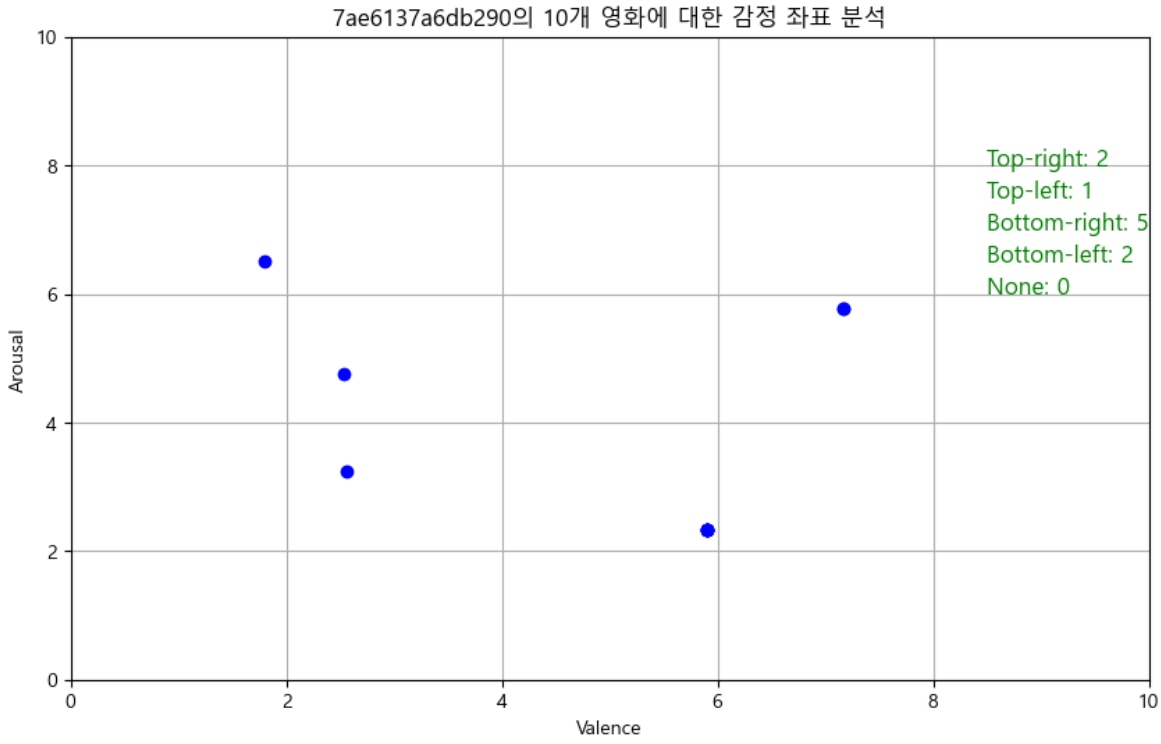

dc4bd283f353d6의 10개 영화에 대한 감정 좌표 분석
Top-right: 0
Top-left: 0
Bottom-right: 5
Bottom-left: 5
None: 0
Arousal
Valence
0
2
4
6
8
10

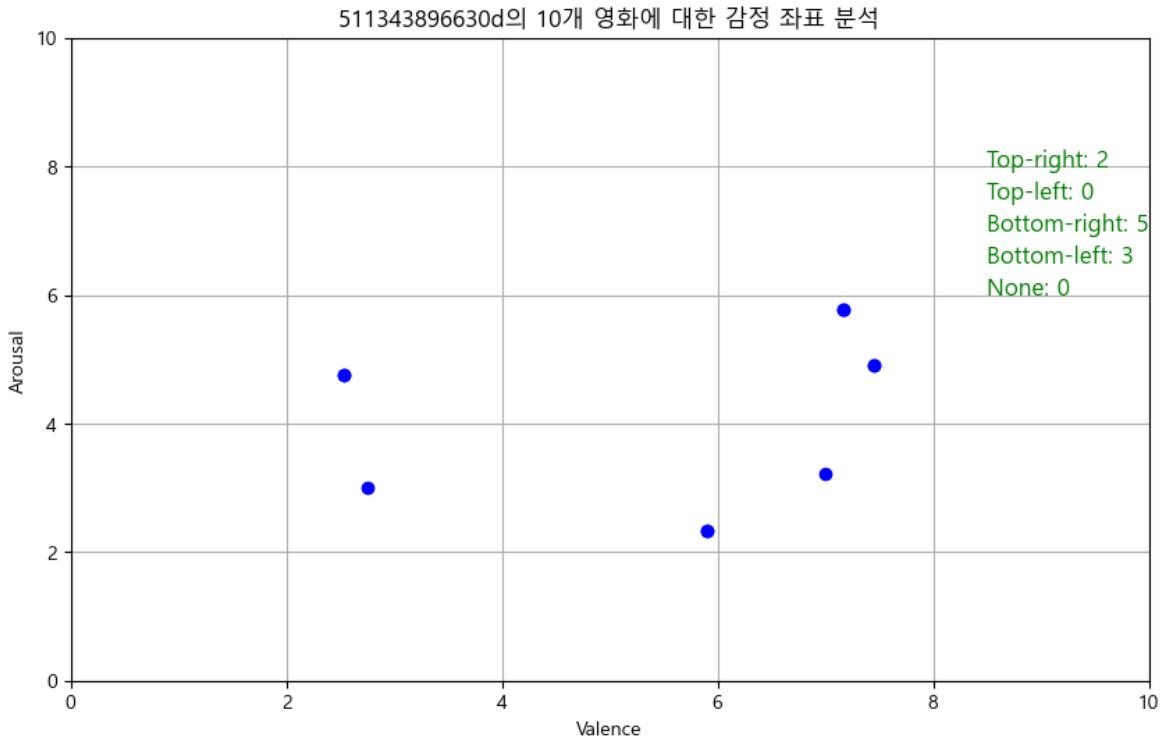

e290c0f353d6의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 2
Bottom-right: 5
Bottom-left: 2
None: 0
Arousal
Valence
0
2
4
6
8
10

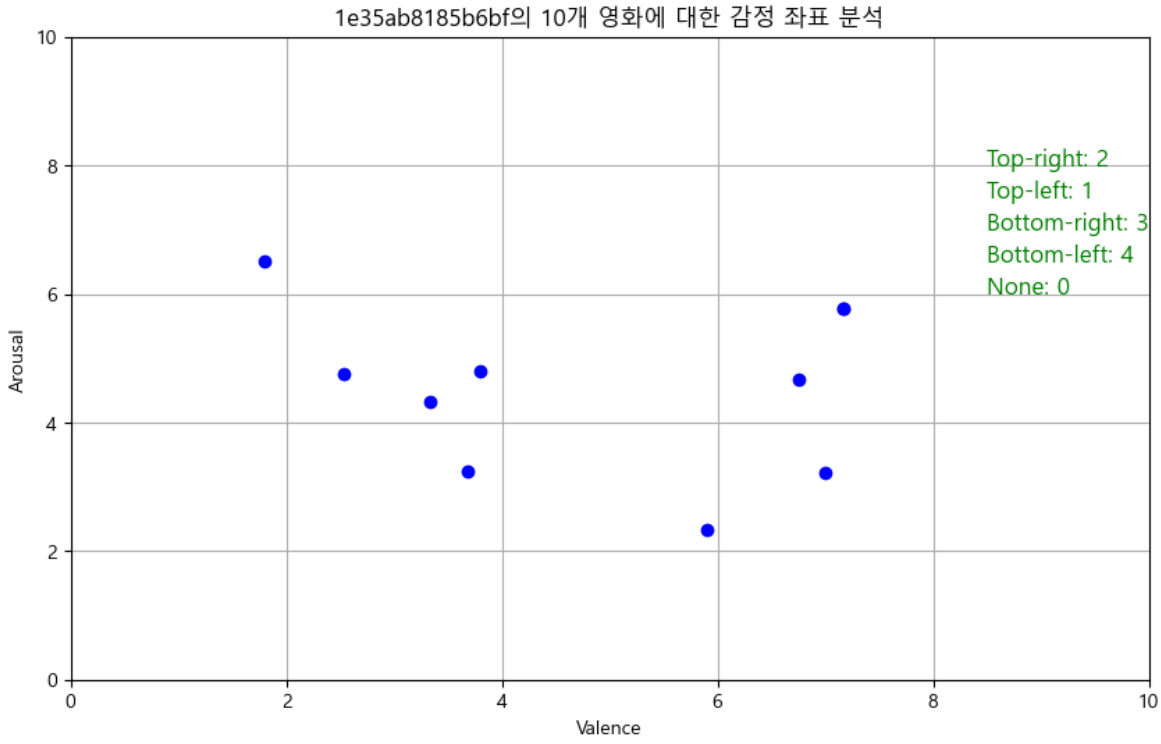

81a5ba37fe1e8e의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 0
Bottom-right: 4
Bottom-left: 5
None: 0
Arousal
Valence
0
2
4
6
8
10

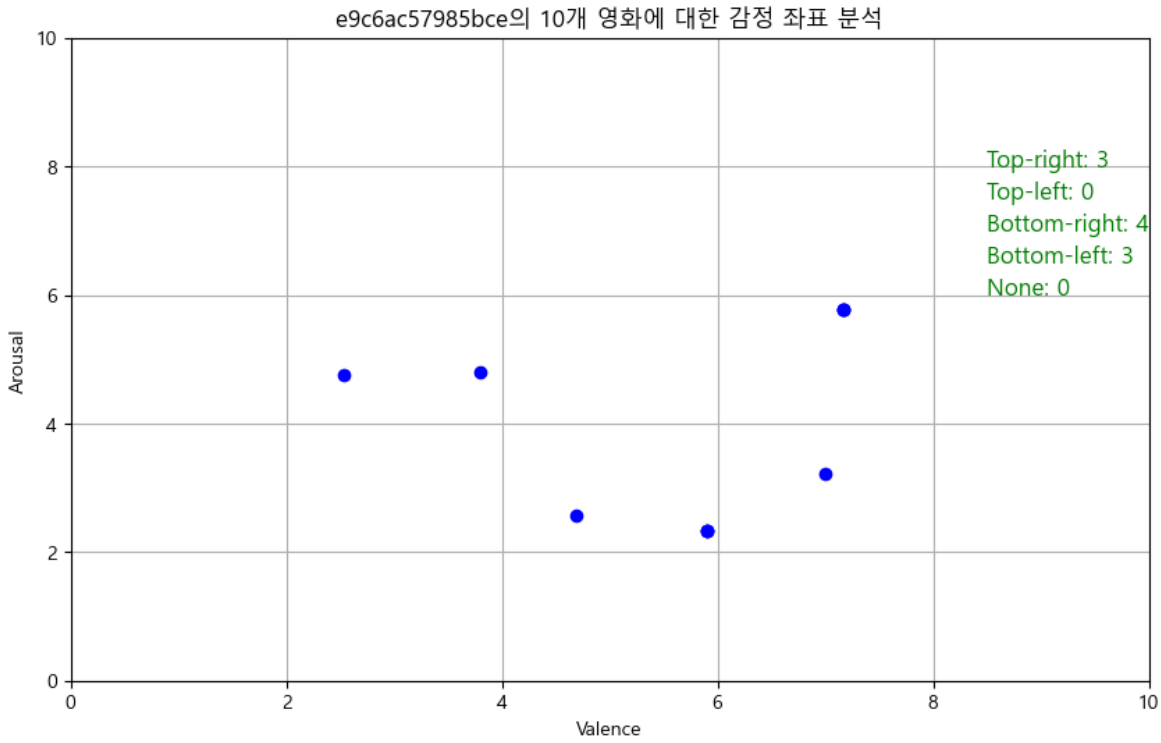

186a5d2fdbc043의 10개 영화에 대한 감정 좌표 분석
Top-right: 0
Top-left: 1
Bottom-right: 4
Bottom-left: 5
None: 0
Arousal
Valence
0
2
4
6
8
10


652573a7dbc043의 10개 영화에 대한 감정 좌표 분석
Top-right: 1
Top-left: 1
Bottom-right: 5
Bottom-left: 3
None: 0
Arousal
Valence
0
2
4
6
8
10


7ae6137a6db290의 10개 영화에 대한 감정 좌표 분석
Top-right: 2
Top-left: 1
Bottom-right: 5
Bottom-left: 2
None: 0
Arousal
Valence
0
2
4
6
8
10


511343896630d의 10개 영화에 대한 감정 좌표 분석
Top-right: 2
Top-left: 0
Bottom-right: 5
Bottom-left: 3
None: 0
Arousal
Valence
0
2
4
6
8
10


1e35ab8185b6bf의 10개 영화에 대한 감정 좌표 분석
Top-right: 2
Top-left: 1
Bottom-right: 3
Bottom-left: 4
None: 0
Arousal
Valence
0
2
4
6
8
10



In [ ]:
```
df_selected.head()
```

Out[ ]:

	Submission ID	다음 중 감정을 가장 잘 나타내는 단어를 골라 주세요.	다음 중 감정을 가장 잘 나타내는 단어를 골라 주세요..1	다음 중 감정을 가장 잘 나타내는 단어를 골라 주세요..2	다음 중 감정을 가장 잘 나타내는 단어를 골라 주세요..3	다음 중 감정을 가장 잘 나타내는 단어를 골라 주세요..
0	fb14df8ad543d1	평온 (Calmness)	평온 (Calmness)	만족 (Contentment)	만족 (Contentment)	놀 (Astonishmen
1	2686e6d7d543d1	행복 (Happiness)	즐거움 (Pleasure)	기쁨 (Delight)	평온 (Calmness)	짜 (Annoyanc
2	3b6e7dc0d543d1	해당하는 감정이 없음(None of These)	평온 (Calmness)	즐거움 (Pleasure)	평온 (Calmness)	우 (Depressio
3	7e237f92deb748	평온 (Calmness)	평온 (Calmness)	해당하는 감정이 없음(None of These)	평온 (Calmness)	만 (Contentmen
4	de6d9c4a0170b	행복 (Happiness)	평온 (Calmness)	즐거움 (Pleasure)	해당하는 감정이 없음(None of These)	두려움 (Fea

5 rows × 21 columns

In [8]:
```
for i in range(10):
    df_selected[f'emotion_{i+1}'] = df_selected[emotion_columns[i]].copy()
```

```
emotion_rank_per_movie = {}

for i in range(10):
    emotion_counts = {emotion: 0 for emotion in emotion_to_coordinates.keys()}

    for person in df_selected['Submission ID']:
        emotion = df_selected.loc[df_selected['Submission ID'] == person, f'emot
        if emotion in emotion_counts:
            emotion_counts[emotion] += 1

    sorted_emotions = sorted(emotion_counts.items(), key=lambda x: x[1], reverse
    emotion_rank_per_movie[f'영화 {i+1}'] = sorted_emotions

for movie, rankings in emotion_rank_per_movie.items():
    print(f"\n{movie} 감정 순위:")
    for emotion, count in rankings:
        print(f"{emotion}: {count}회")
```

```
C:\Users\kgty\AppData\Local\Temp\ipykernel_22044\4289841536.py:2: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_selected[f'emotion_{i+1}'] = df_selected[emotion_columns[i]].copy()
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[8], line 7
      4 emotion_rank_per_movie = {}
      6 for i in range(10):
----> 7     emotion_counts = {emotion: 0 for emotion in emotion_to_coordinates.ke
ys()}
      9     for person in df_selected['Submission ID']:
     10         emotion = df_selected.loc[df_selected['Submission ID'] == person,
f'emotion_{i+1}'].values[0]

NameError: name 'emotion_to_coordinates' is not defined
```

In [9]:
```
df_selected = df[['Submission ID'] + emotion_columns + valence_columns + arousal

emotion_va_mean = []

for person in df_selected['Submission ID'].unique():
    person_data = df_selected[df_selected['Submission ID'] == person]

    for i in range(len(emotion_columns)):
        emotion = person_data[emotion_columns[i]].values[0]

        valence = person_data[valence_columns[i]].values[0]
        arousal = person_data[arousal_columns[i]].values[0]

        emotion_va_mean.append({
            'Submission ID': person,
            'Emotion': emotion,
            'Valence': valence,
            'Arousal': arousal,
            'Mean VA': (valence + arousal) / 2
        })
```

```
df_va_mean = pd.DataFrame(emotion_va_mean)

emotion_avg_va = df_va_mean.groupby('Emotion')[['Valence', 'Arousal', 'Mean VA']]
emotion_response_count = df_va_mean.groupby('Emotion').size()

emotion_avg_va['Response Count'] = emotion_response_count

print(emotion_avg_va)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[9], line 1
----> 1 df_selected = df[['Submission ID'] + emotion_columns + valence_columns +
arousal_columns]
      3 emotion_va_mean = []
      5 for person in df_selected['Submission ID'].unique():

File c:\Users\kgty\anaconda3\Lib\site-packages\pandas\core\frame.py:3767, in Data
Frame.__getitem__(self, key)
   3765     if is_iterator(key):
   3766         key = list(key)
-> 3767     indexer = self.columns._get_indexer_strict(key, "columns")[1]
   3769 # take() does not accept boolean indexers
   3770 if getattr(indexer, "dtype", None) == bool:

File c:\Users\kgty\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:5877,
in Index._get_indexer_strict(self, key, axis_name)
   5874 else:
   5875     keyarr, indexer, new_indexer = self._reindex_non_unique(keyarr)
-> 5877 self._raise_if_missing(keyarr, indexer, axis_name)
   5879 keyarr = self.take(indexer)
   5880 if isinstance(key, Index):
   5881     # GH 42790 - Preserve name from an Index

File c:\Users\kgty\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:5941,
in Index._raise_if_missing(self, key, indexer, axis_name)
   5938     raise KeyError(f"None of [{key}] are in the [{axis_name}]")
   5940 not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
-> 5941 raise KeyError(f"{not_found} not in index")

KeyError: "['Submission ID'] not in index"
```

여기서 부터 추가된 코드 ----------

In [194…

```
# 중복 응답률(Redundancy Rate) 계산 함수
def calculate_diversity(series):
    unique_values = len(set(series))  # 고유한 값 개수
    total_values = len(series)  # 전체 응답 개수
    diversity_ratio = unique_values / total_values  # 다양성 비율
    redundancy_rate = (1 - diversity_ratio) * 100  # 중복 응답률 (%)

    return pd.Series([unique_values, diversity_ratio, redundancy_rate],
                     index=['Unique_Count', 'Diversity_Ratio', 'Redundancy_Rate'

# 개인별 응답 신뢰도 분석
df['Valence_Diversity'] = df_selected[valence_columns].apply(calculate_diversity
df['Arousal_Diversity'] = df_selected[arousal_columns].apply(calculate_diversity
```

```
# 중복 응답률이 너무 높은 사용자 제거 (예: Diversity Ratio < 0.5인 경우)
df_filtered = df[(df['Valence_Diversity'] >= 0.4) & (df['Arousal_Diversity'] >=
```

In [195…

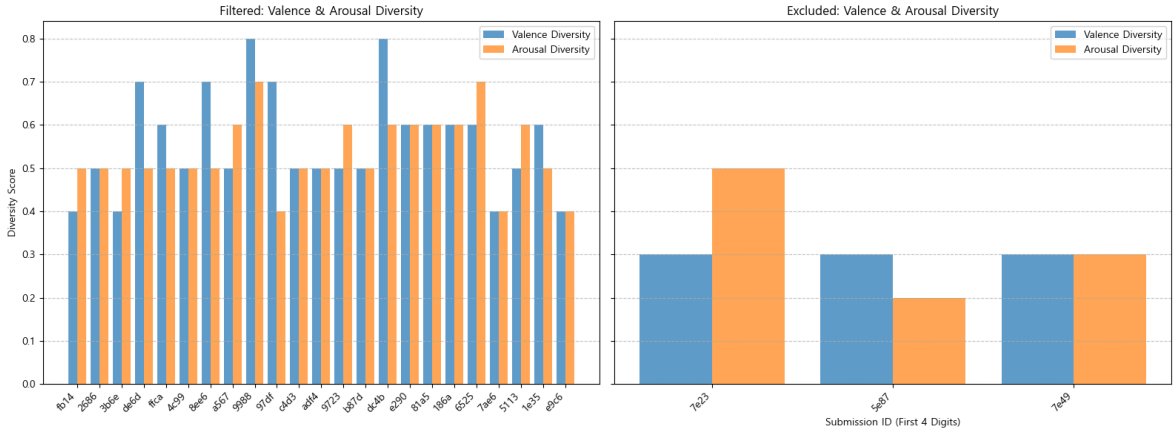
```
df_excluded = df[~((df['Valence_Diversity'] >= 0.4) & (df['Arousal_Diversity'] >
df_excluded[['Submission ID']]
```

Out[195…

	Submission ID
3	7e237f92deb748
7	5e877a85417bf3
10	7e4950a4417bf3

In [196…

```
import matplotlib.pyplot as plt
import numpy as np

# Submission ID의 앞 4자리 추출
df_filtered['Submission_ID_Short'] = df_filtered['Submission ID'].astype(str).st
df_excluded['Submission_ID_Short'] = df_excluded['Submission ID'].astype(str).st

# 그래프 설정
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(16, 6), sharey=True)

# 막대 너비 설정
bar_width = 0.4
x_filtered = np.arange(len(df_filtered))
x_excluded = np.arange(len(df_excluded))

# df_filtered 막대 그래프 생성
axes[0].bar(x_filtered - bar_width/2, df_filtered['Valence_Diversity'], bar_widt
axes[0].bar(x_filtered + bar_width/2, df_filtered['Arousal_Diversity'], bar_widt
axes[0].set_xticks(x_filtered)
axes[0].set_xticklabels(df_filtered['Submission_ID_Short'].values, rotation=45,
axes[0].set_ylabel('Diversity Score')
axes[0].set_title('Filtered: Valence & Arousal Diversity')
axes[0].legend()
axes[0].grid(axis='y', linestyle='--', alpha=0.7)

# df_excluded 막대 그래프 생성
axes[1].bar(x_excluded - bar_width/2, df_excluded['Valence_Diversity'], bar_widt
axes[1].bar(x_excluded + bar_width/2, df_excluded['Arousal_Diversity'], bar_widt
axes[1].set_xticks(x_excluded)
axes[1].set_xticklabels(df_excluded['Submission_ID_Short'].values, rotation=45,
axes[1].set_xlabel('Submission ID (First 4 Digits)')
axes[1].set_title('Excluded: Valence & Arousal Diversity')
axes[1].legend()
axes[1].grid(axis='y', linestyle='--', alpha=0.7)

# 그래프 출력
plt.tight_layout()
plt.show()
```



In [197…
```
pd.set_option("display.max_rows", None)  # 모든 행 출력

len(df_filtered)
```

Out[197…
```
23
```

In [201…
```
emotion_columns = [col for col in df.columns if '다음 중 감정을 가장 잘 나타내는


df_filtered[emotion_columns]

# 중복 응답률 계산 함수
def calculate_text_diversity(series):
    unique_values = len(set(series))  # 고유한 응답 개수
    total_values = len(series)  # 전체 응답 개수
    if total_values == 0:
        print('0 is exist')
    diversity_ratio = unique_values / total_values  # 다양성 비율
    redundancy_rate = (1 - diversity_ratio) * 100  # 중복 응답률 (%)
    return pd.Series([unique_values, diversity_ratio, redundancy_rate],
                     index=['Unique_Count', 'Diversity_Ratio', 'Redundancy_Rate'


df_filtered['Emotion_Diversity'] = df_filtered[emotion_columns].apply(calculate_

df_filtered

df_excluded = df_filtered[df_filtered['Emotion_Diversity'] < 0.6].reset_index(dr


df_filtered = df_filtered[df_filtered['Emotion_Diversity'] >= 0.6].reset_index(d
```

In [205…
```
df_excluded[['Submission ID']]
```

Out[205…

	Submission ID
0	fb14df8ad543d1
1	7ae6137a6db290

In [ ]:
```

```