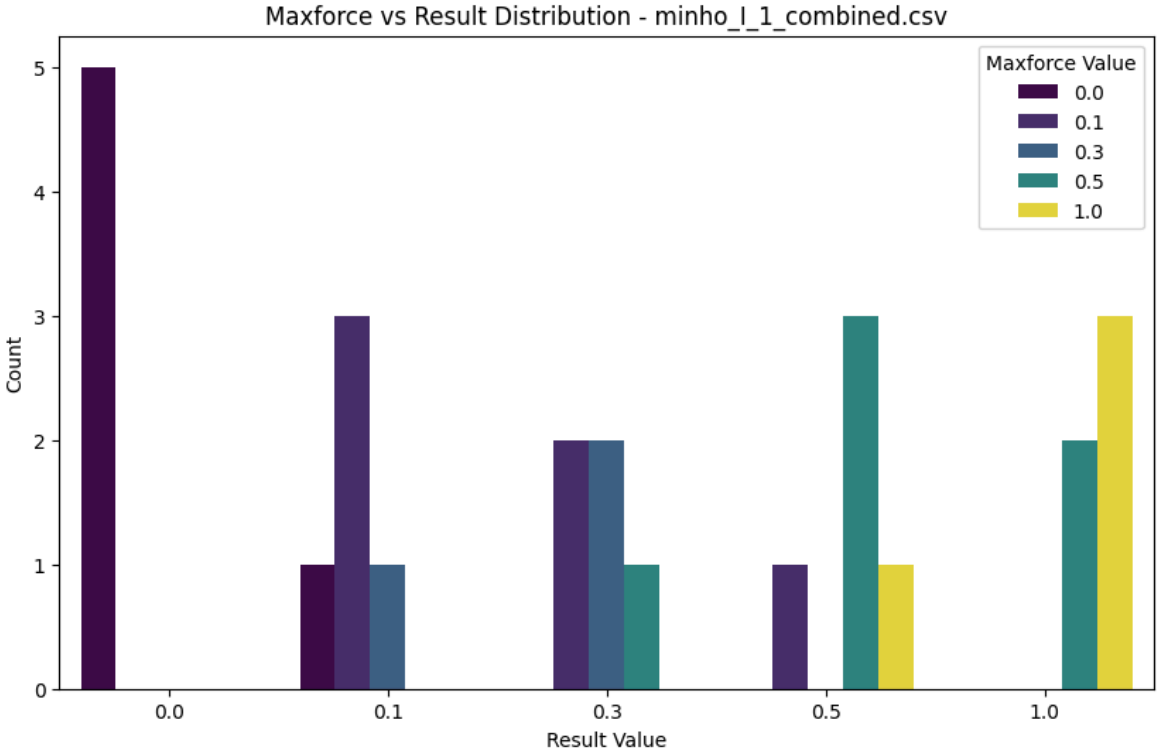
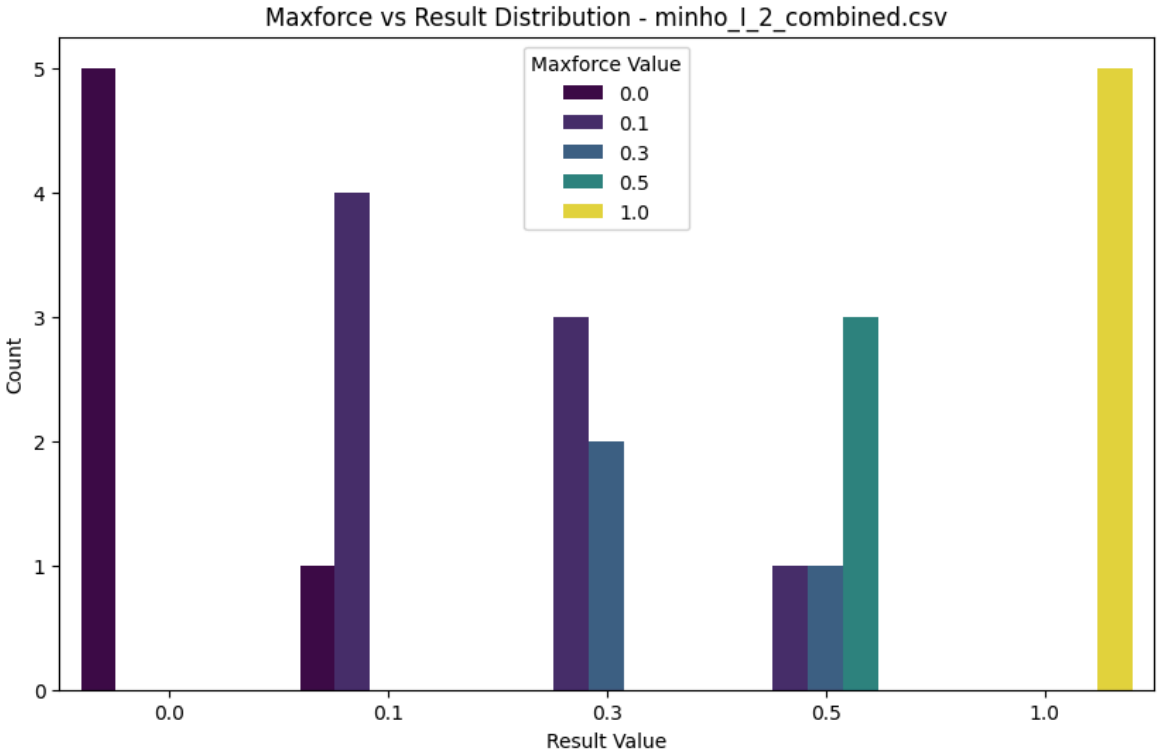
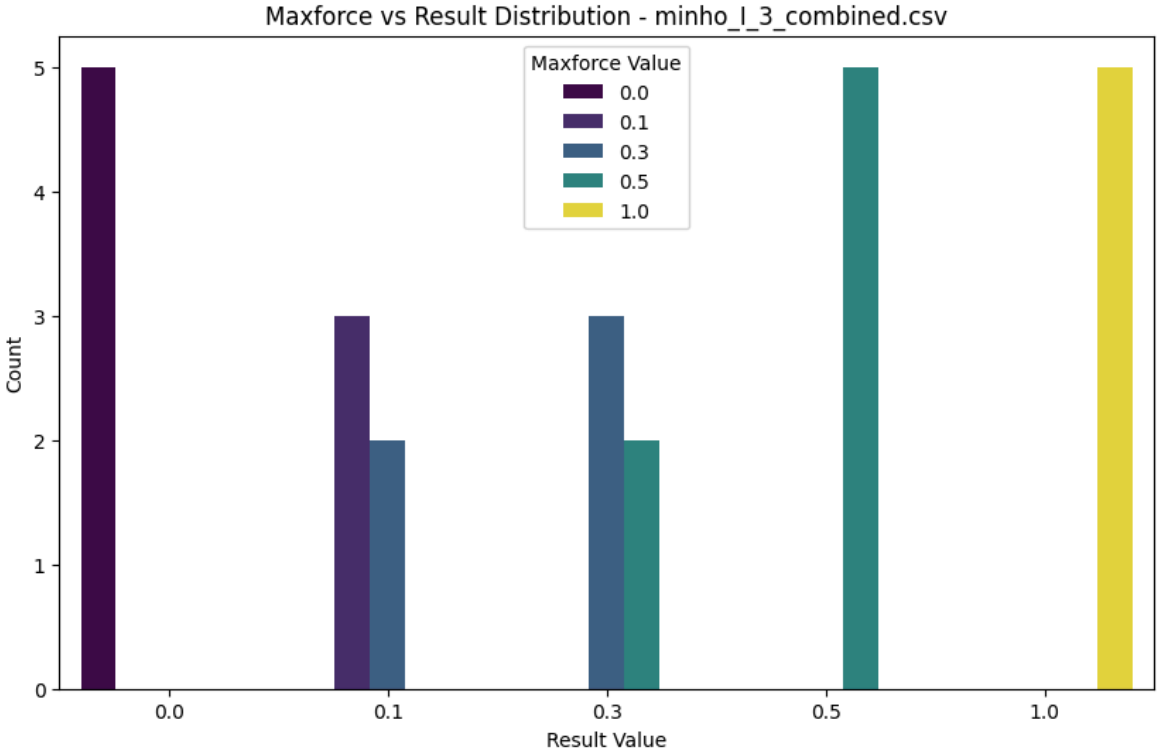
🔗 각 데이터 별 결과 시각화

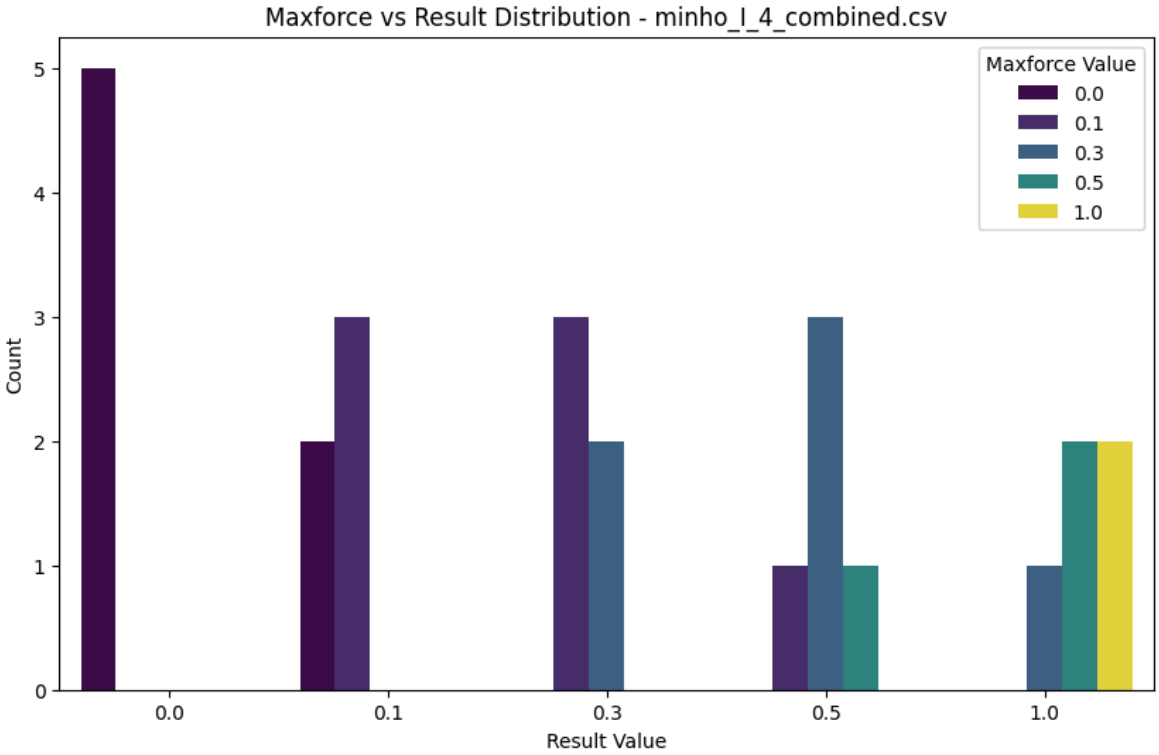
In [3]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns

# 파일이 저장된 디렉토리 경로 (예시)
directory_path = './combined_data'

# 디렉토리에서 모든 CSV 파일 리스트 가져오기
csv_files = [f for f in os.listdir(directory_path) if f.endswith('.csv')]
```

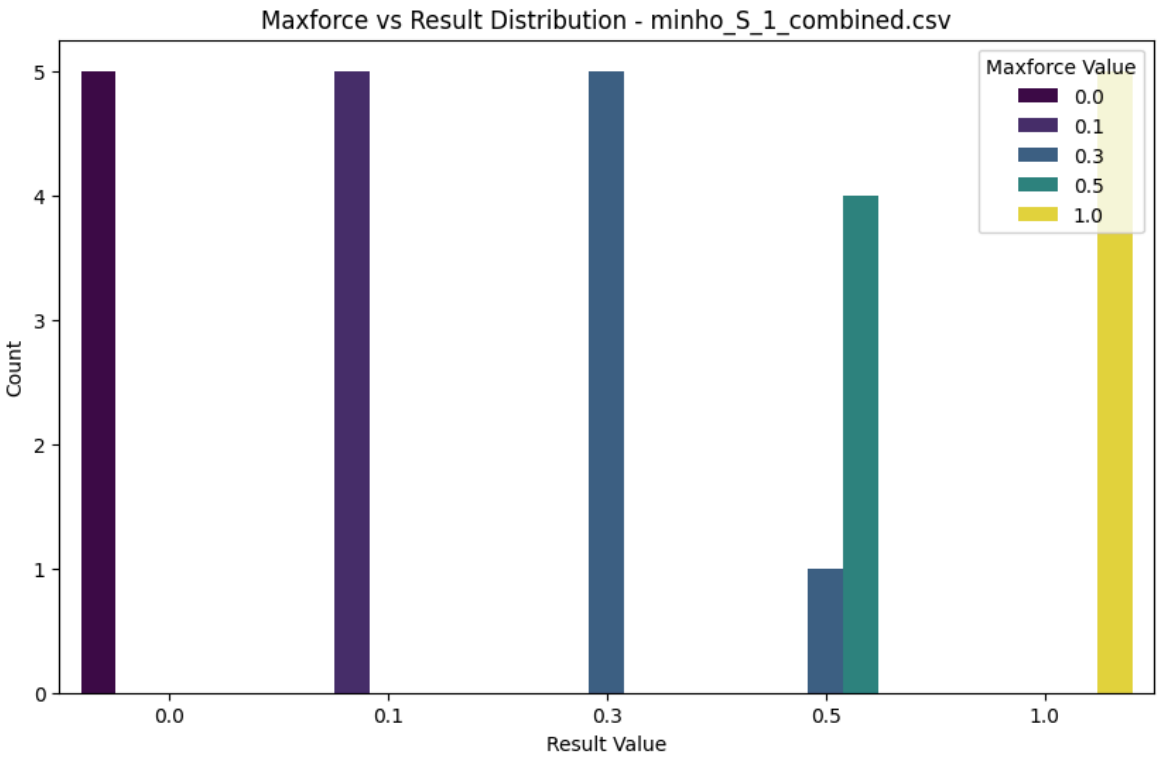
In [4]:
```
# 각 CSV 파일에 대해 작업 수행
for csv_file in csv_files:
    file_path = os.path.join(directory_path, csv_file)

    # CSV 파일 읽기
    df = pd.read_csv(file_path)

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df.columns and 'result' in df.columns:
        maxforce_values = [0.0, 0.1, 0.3, 0.5, 1.0]
        result_values = [0.0, 0.1, 0.3, 0.5, 1.0]

        distribution = []

        for maxforce_value in maxforce_values:
            for result_value in result_values:
                count = len(df[(df['maxforce'] == maxforce_value) & (df['result'
                distribution.append({
                    'maxforce_value': maxforce_value,
                    'result_value': result_value,
                    'count': count
                })

        distribution_df = pd.DataFrame(distribution)

        # 막대그래프 생성
        plt.figure(figsize=(10, 6))
        sns.barplot(x='result_value', y='count', hue='maxforce_value', data=dist

        # 그래프 세부 설정
        plt.title(f'Maxforce vs Result Distribution - {csv_file}')
        plt.xlabel('Result Value')
        plt.ylabel('Count')
        plt.legend(title='Maxforce Value')

        # 그래프 출력
        plt.show()

    else:
        print(f"'maxforce' or 'result' column missing in {csv_file}")
```

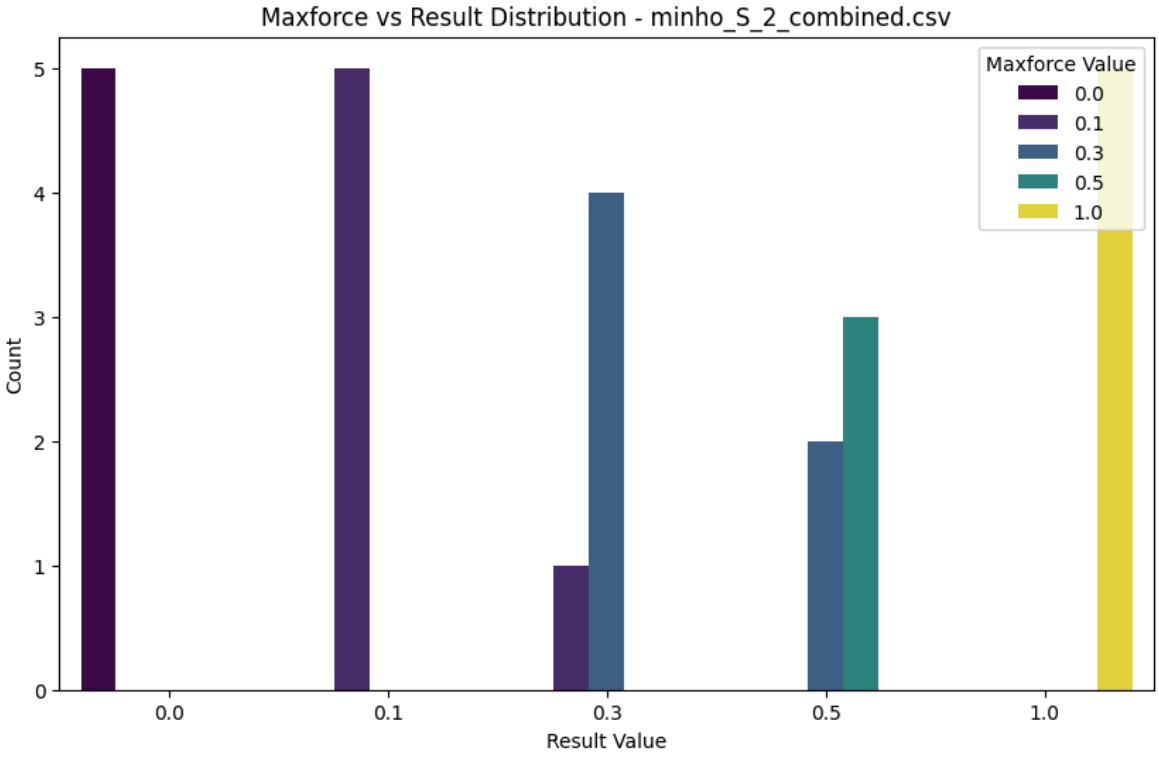

Maxforce vs Result Distribution - minho_I_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

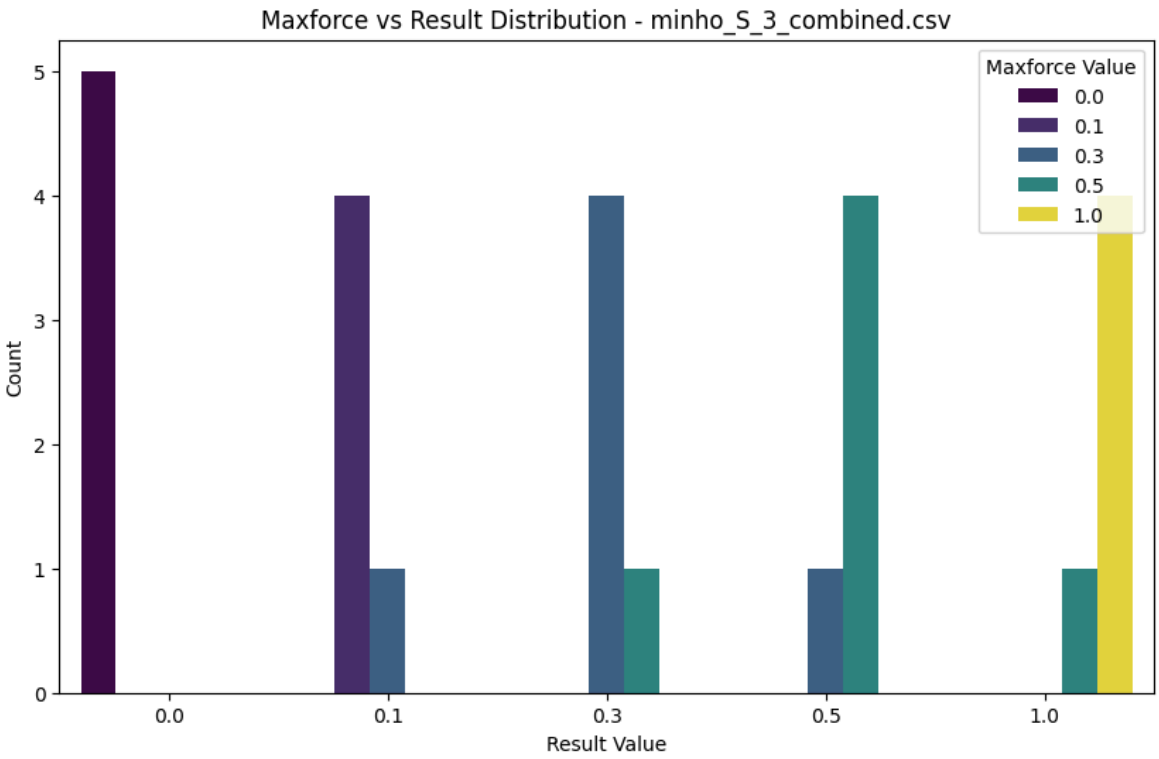

Maxforce vs Result Distribution - minho_I_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

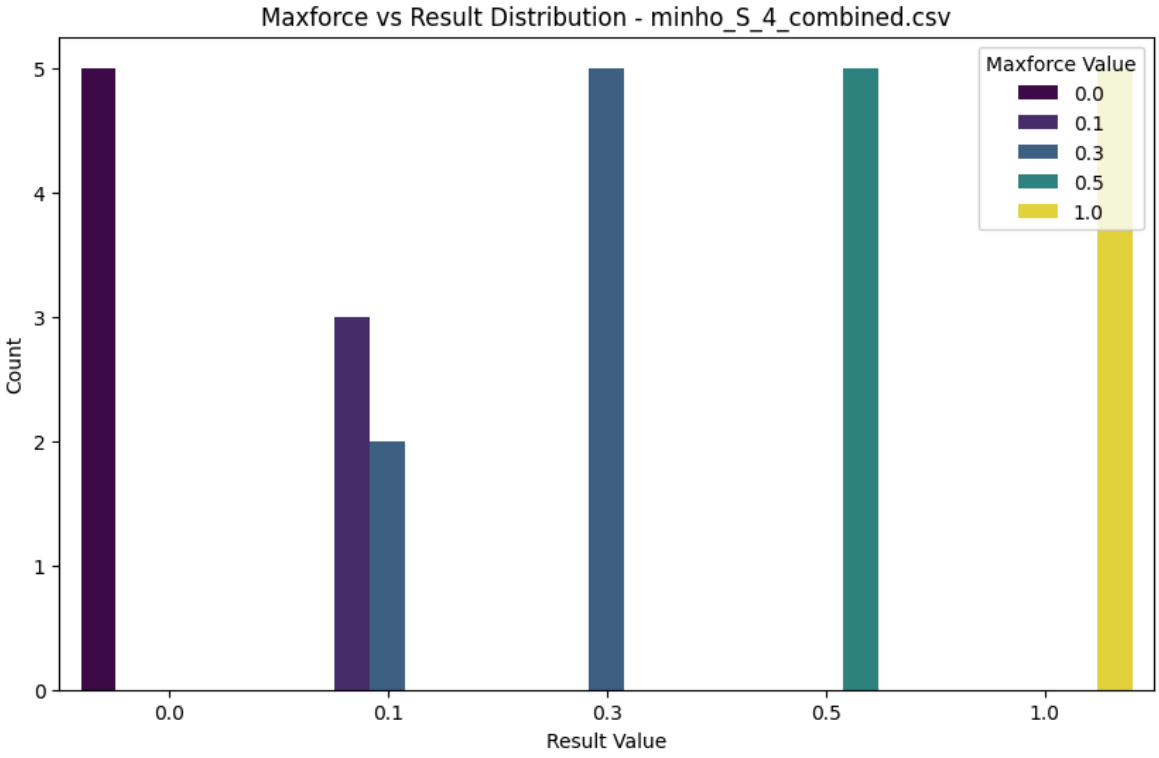

Maxforce vs Result Distribution - minho_I_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
Result Value
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0

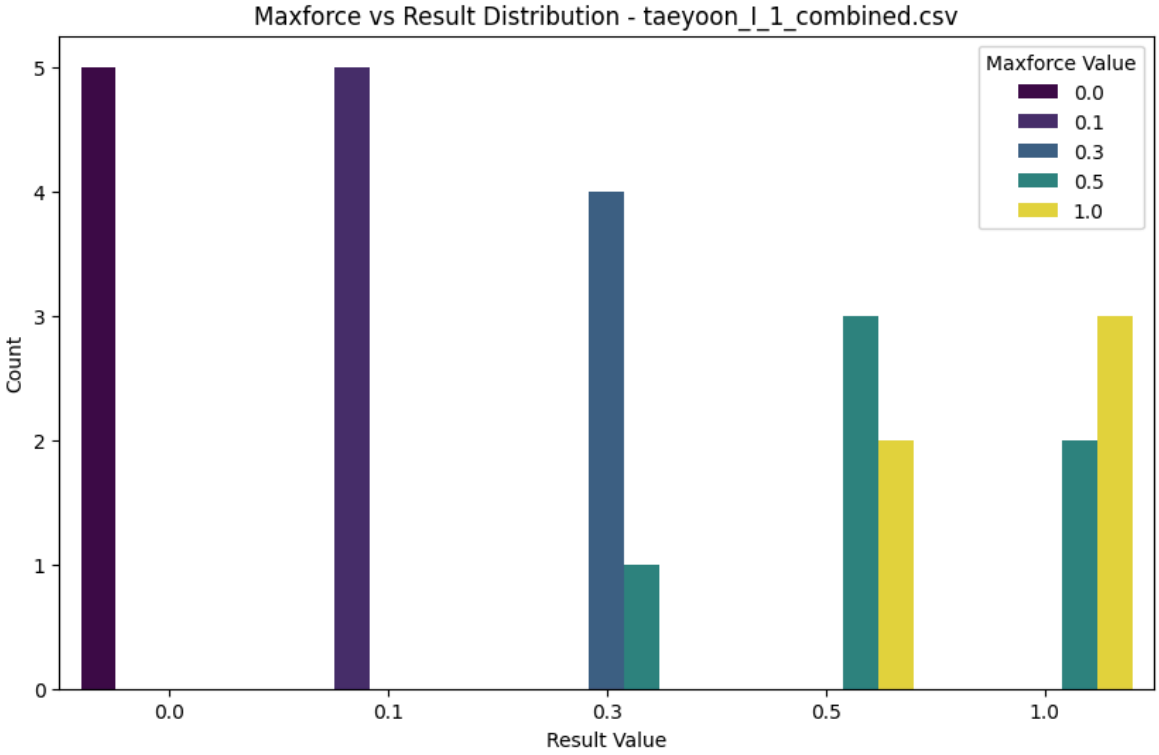

Maxforce vs Result Distribution - minho_I_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
Result Value
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0

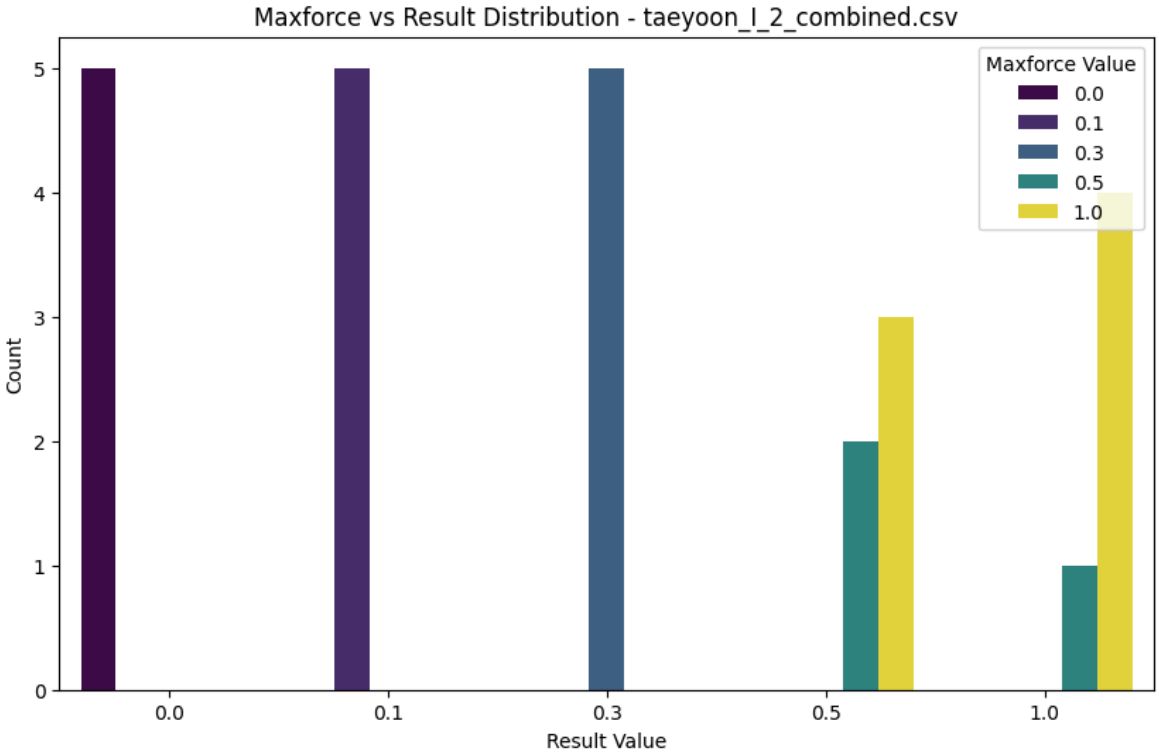

Maxforce vs Result Distribution - minho_S_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

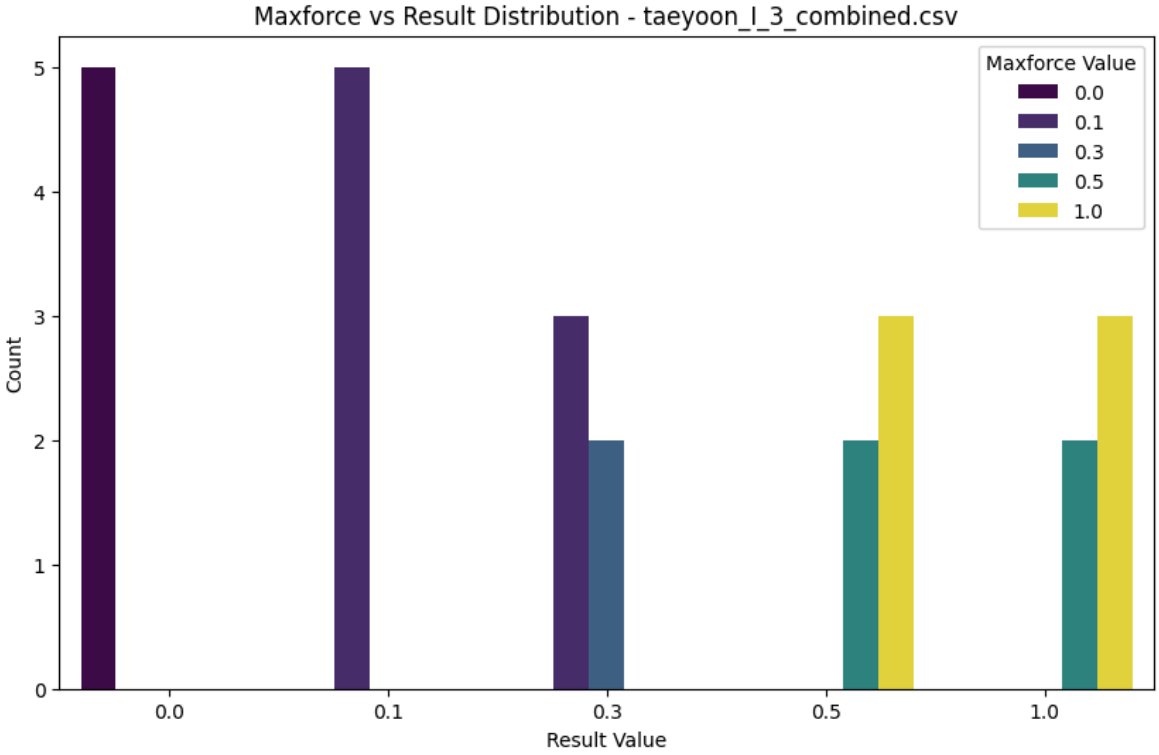

Maxforce vs Result Distribution - minho_S_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

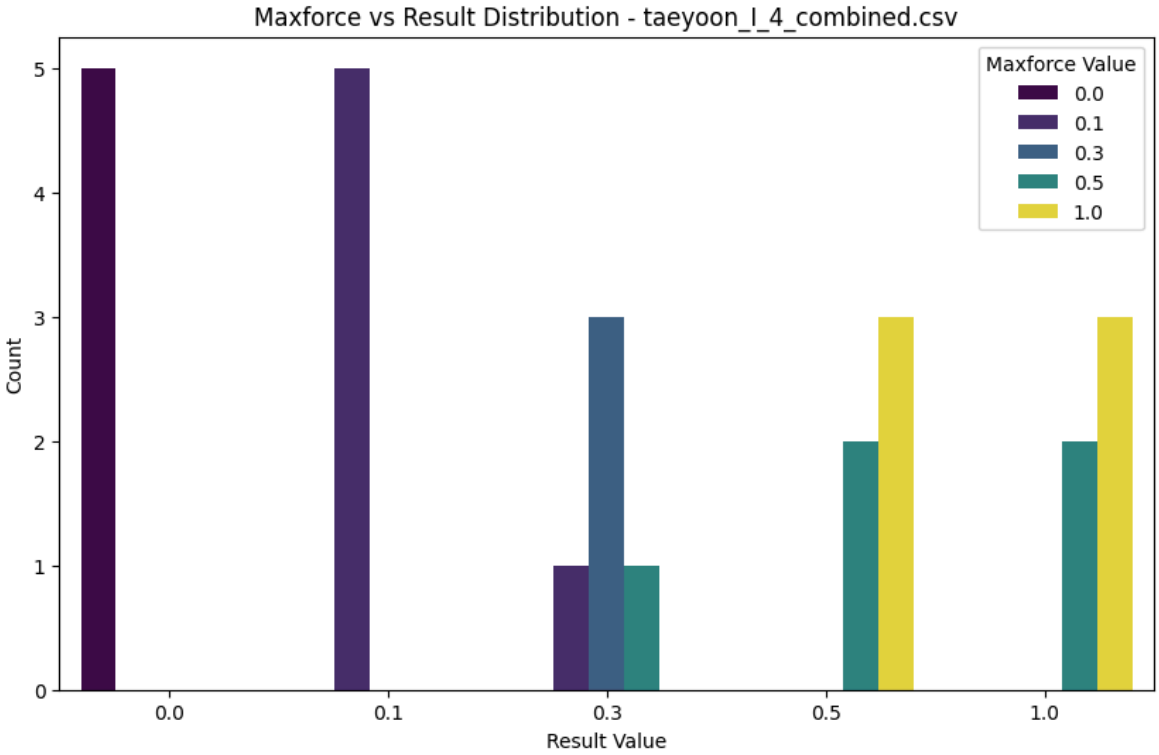

Maxforce vs Result Distribution - minho_S_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

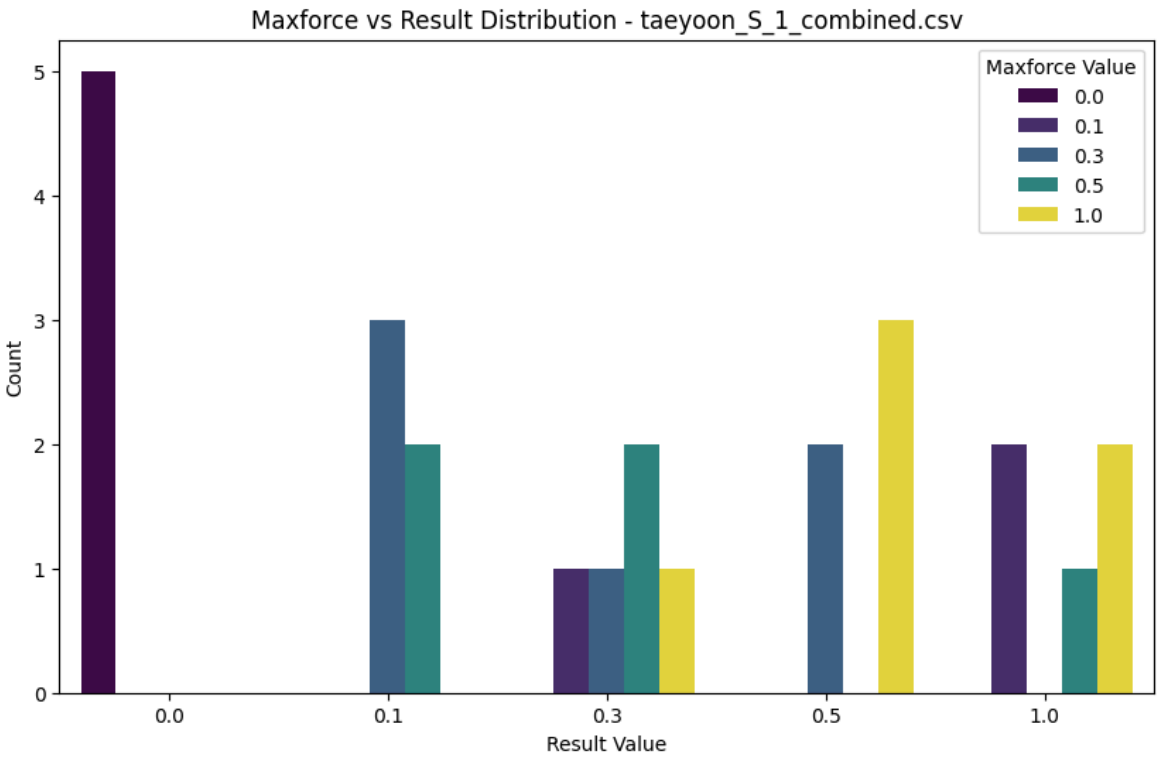

Maxforce vs Result Distribution - minho_S_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

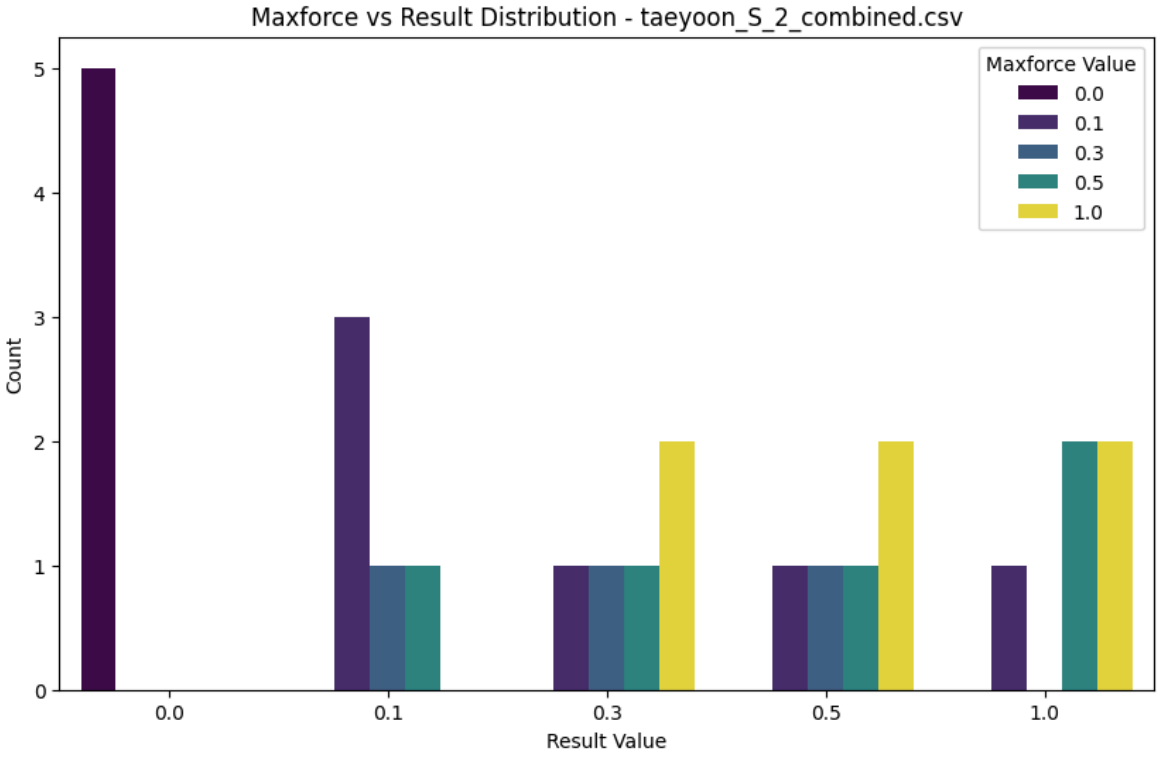

Maxforce vs Result Distribution - taeyoon_l_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

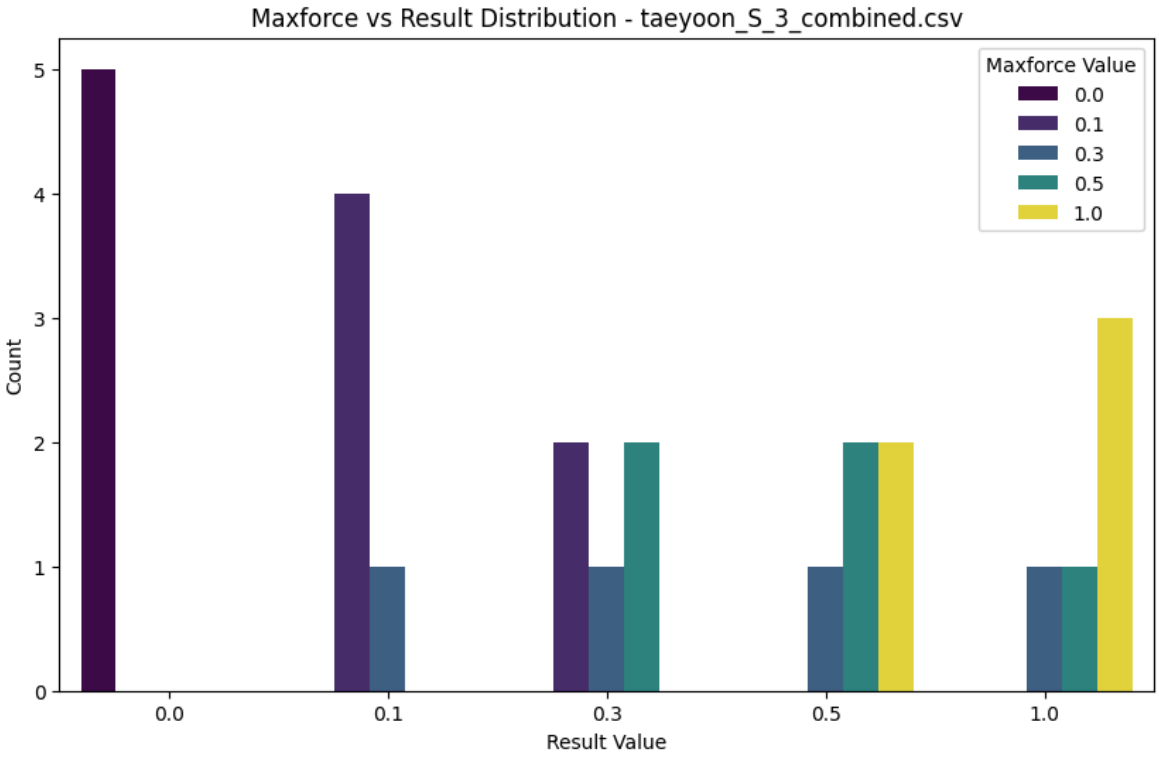

Maxforce vs Result Distribution - taeyoon_l_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

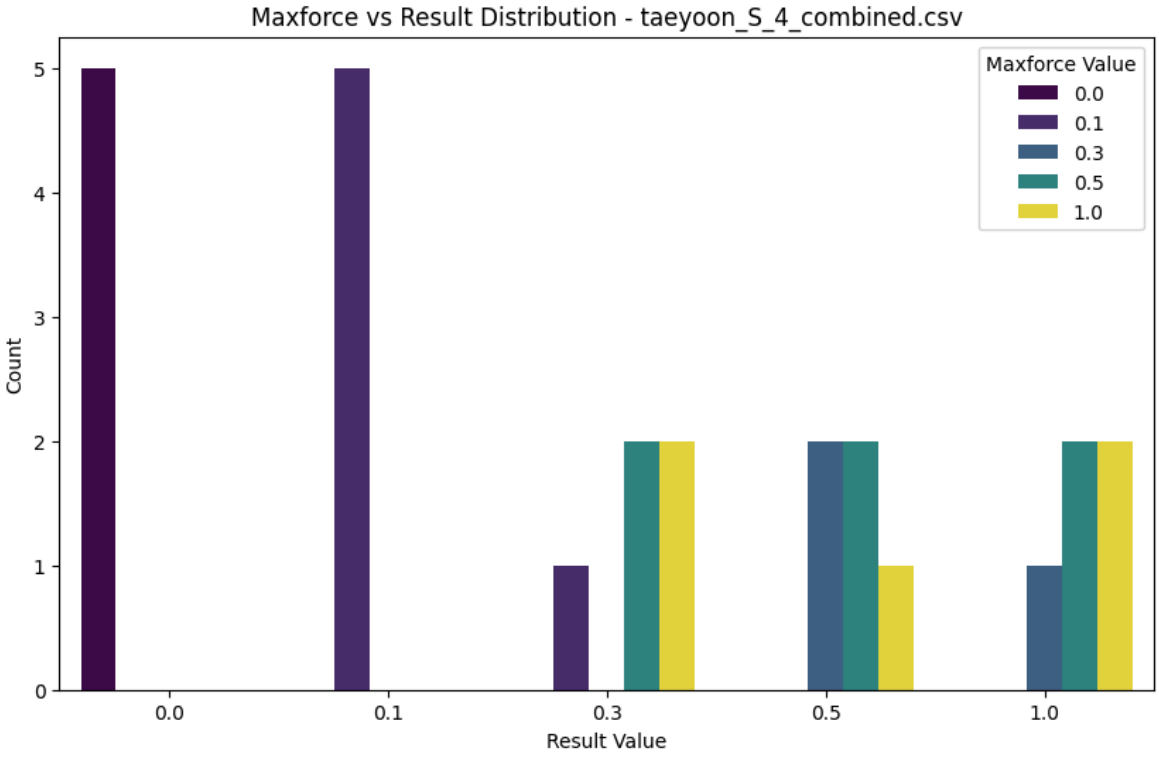

Maxforce vs Result Distribution - taeyoon_I_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value


Maxforce vs Result Distribution - taeyoon_I_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value


Maxforce vs Result Distribution - taeyoon_S_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value


Maxforce vs Result Distribution - taeyoon_S_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

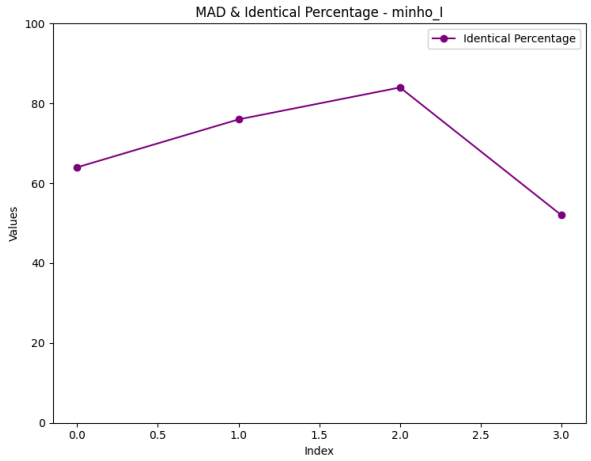


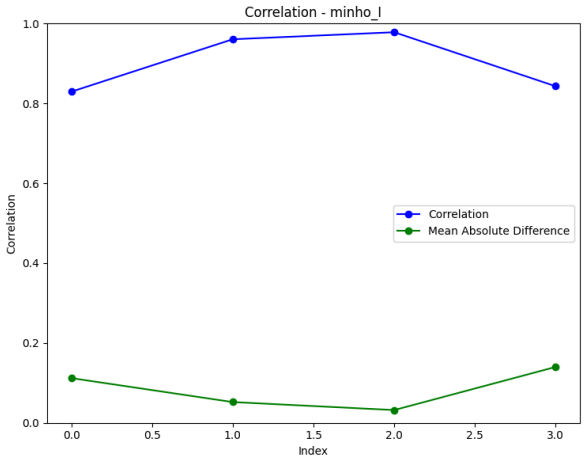


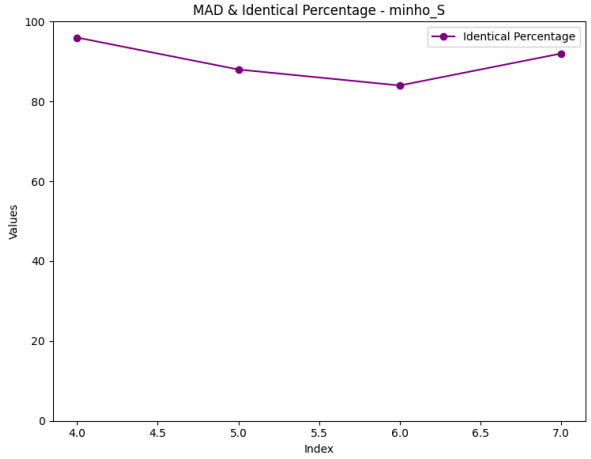
In [7]:
```python
# 결과를 저장할 리스트 초기화
comparison_results = []

# 각 CSV 파일에 대해 작업 수행
for csv_file in csv_files:
    file_path = os.path.join(directory_path, csv_file)

    # CSV 파일 읽기
    df = pd.read_csv(file_path)

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df.columns and 'result' in df.columns:
        # 상관계수 계산
```

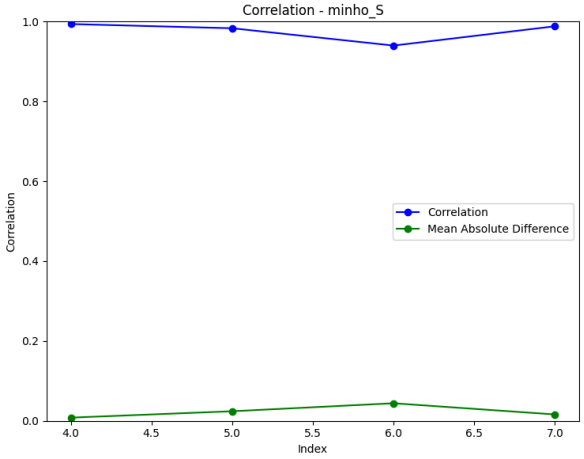
```
        correlation = df['maxforce'].corr(df['result'])

        # 차이의 절대값 평균 계산
        mean_absolute_difference = (df['maxforce'] - df['result']).abs().mean()

        # maxforce와 result가 동일한 값의 비율 계산
        identical_percentage = (df['maxforce'] == df['result']).mean() * 100

        # 결과 저장
        comparison_results.append({
            'filename': csv_file,
            'correlation': correlation,
            'mean_absolute_difference': mean_absolute_difference,
            'identical_percentage': identical_percentage
        })

        # # 그래프 생성
        # plt.figure(figsize=(10, 6))

        # # 산점도 (scatter plot)
        # plt.scatter(df.index, df['maxforce'], label='Maxforce', color='blue',
        # plt.scatter(df.index, df['result'], label='Result', color='orange', al

        # # 선 그래프 (line plot)
        # plt.plot(df.index, df['maxforce'], color='blue', alpha=0.3)
        # plt.plot(df.index, df['result'], color='orange', alpha=0.3)

        # # 제목 및 라벨 추가
        # plt.title(f'Result vs Maxforce - {csv_file}')
        # plt.xlabel('Index')
        # plt.ylabel('Values')
        # plt.legend()

    else:
        print(f"'maxforce' or 'result' column missing in {csv_file}")

# Convert the results into a DataFrame
results_df = pd.DataFrame(comparison_results)

# Extract the common part of the filename (e.g., 'minho_I' or 'minho_S')
results_df['group'] = results_df['filename'].apply(lambda x: '_'.join(x.split('_

# Remove the 'filename' column
results_df = results_df.drop(columns=['filename'])

# Group the DataFrame by the new 'group' column
grouped = results_df.groupby('group')

# Plot graphs for each group
for name, group in grouped:
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))  # Create two subplots

    # Plot mean absolute difference and identical percentage on the left (ax1)
    ax1.plot(group.index, group['identical_percentage'], marker='o', label='Iden
    ax1.set_ylim(0, 100)  # Set y-axis limits from 0 to 100
    ax1.set_title(f'MAD & Identical Percentage - {name}')
    ax1.set_xlabel('Index')
    ax1.set_ylabel('Values')
    ax1.legend()
```

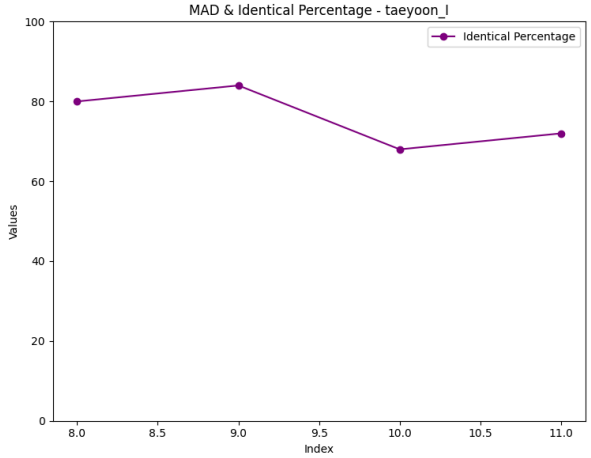
```
    # Plot correlation on the right (ax2)
    ax2.plot(group.index, group['correlation'], marker='o', label='Correlation',
    ax2.plot(group.index, group['mean_absolute_difference'], marker='o', label='

    ax2.set_title(f'Correlation - {name}')
    ax2.set_xlabel('Index')
    ax2.set_ylabel('Correlation')
    ax2.set_ylim(0, 1)
    ax2.legend()

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

# Output the grouped DataFrames
for group_df in grouped:
    print(group_df)
```

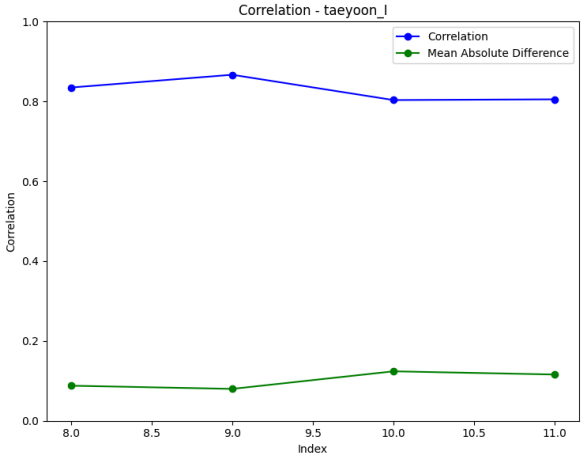


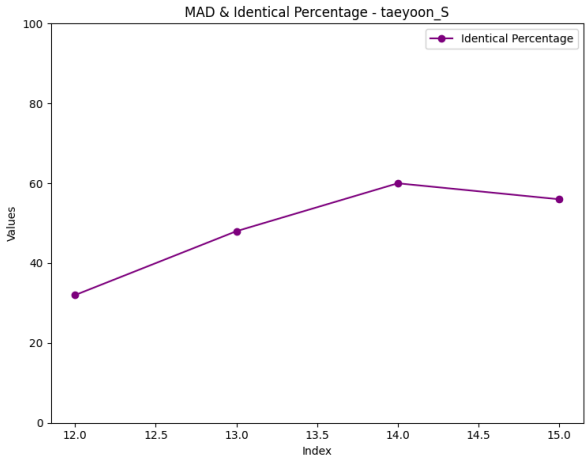


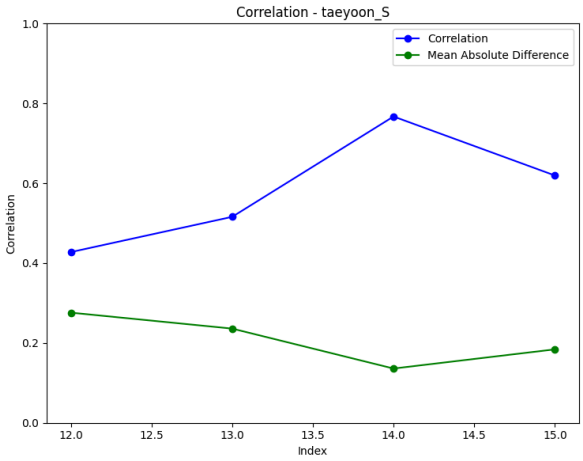












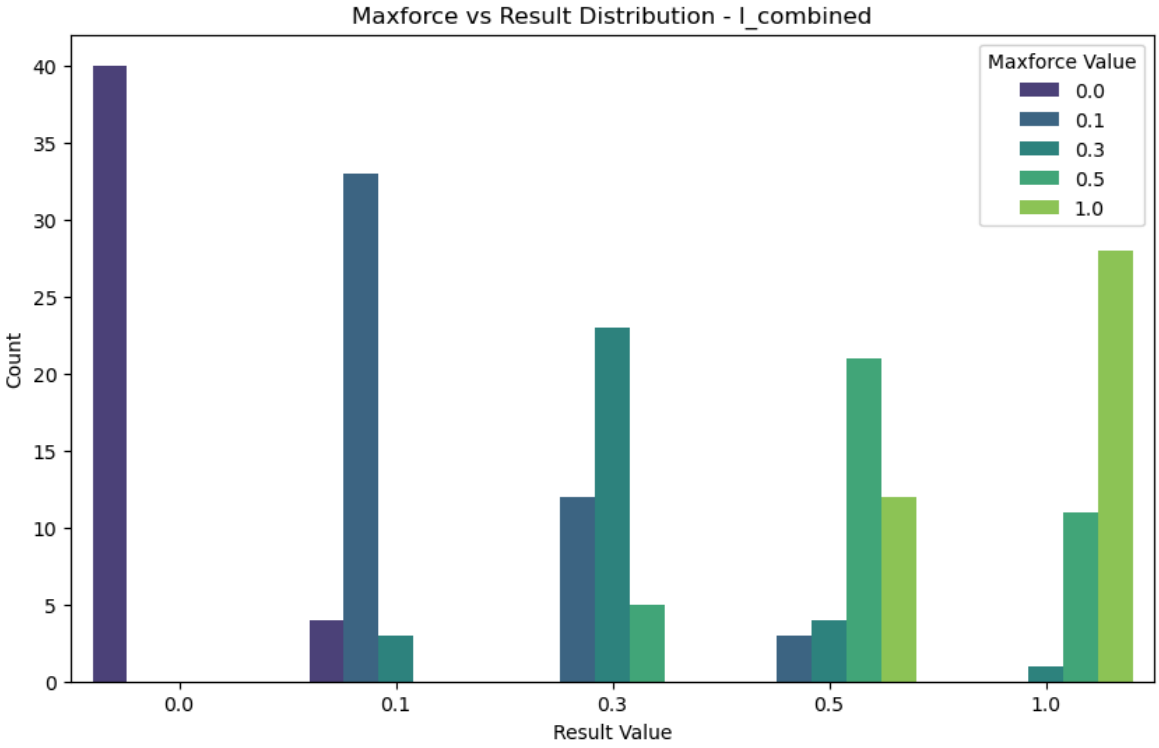

```
('minho_I',    correlation  mean_absolute_difference  identical_percentage    gro
up
0     0.829396                    0.112                 64.0  minho_I
1     0.960760                    0.052                 76.0  minho_I
2     0.978370                    0.032                 84.0  minho_I
3     0.843053                    0.140                 52.0  minho_I)
('minho_S',    correlation  mean_absolute_difference  identical_percentage    gro
up
4     0.993877                    0.008                 96.0  minho_S
5     0.983317                    0.024                 88.0  minho_S
6     0.939945                    0.044                 84.0  minho_S
7     0.988290                    0.016                 92.0  minho_S)
('taeyoon_I',     correlation  mean_absolute_difference  identical_percentage
group
8      0.834818                    0.088                  80.0  taeyoon_I
9      0.866834                    0.080                  84.0  taeyoon_I
10     0.803441                    0.124                  68.0  taeyoon_I
11     0.805299                    0.116                  72.0  taeyoon_I)
('taeyoon_S',     correlation  mean_absolute_difference  identical_percentage
group
12     0.427669                    0.276                  32.0  taeyoon_S
13     0.515988                    0.236                  48.0  taeyoon_S
14     0.767243                    0.136                  60.0  taeyoon_S
15     0.619546                    0.184                  56.0  taeyoon_S)
```

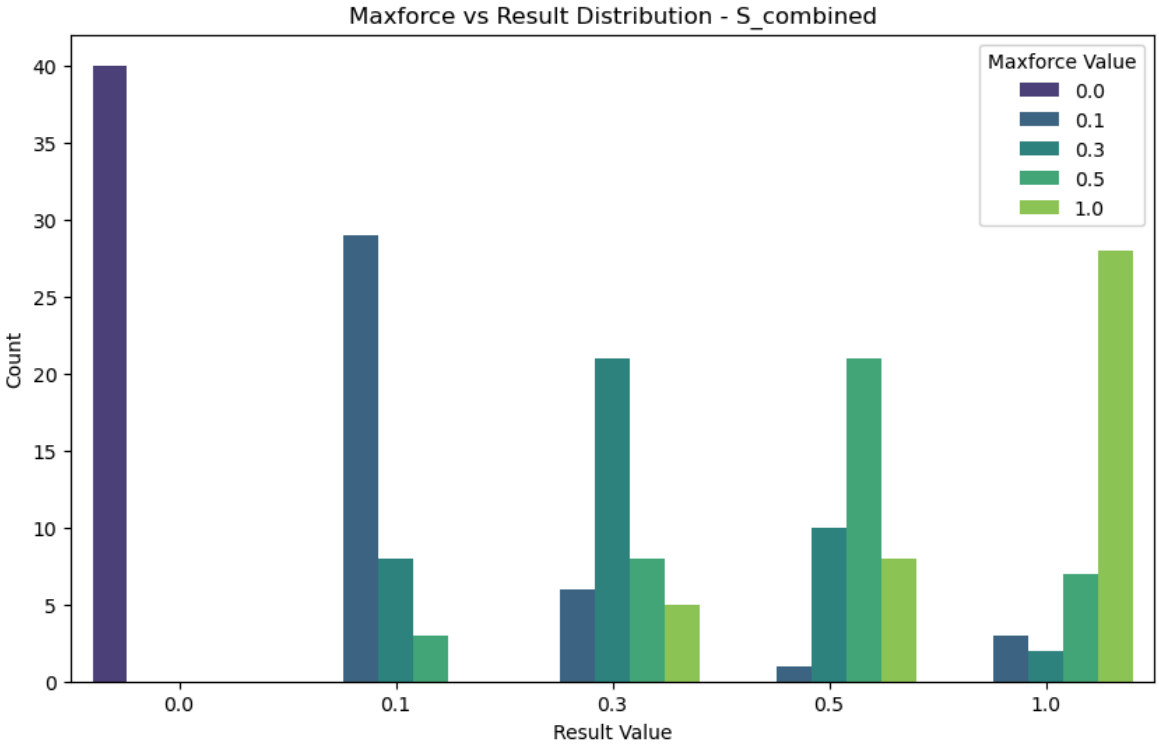
In [ ]:

🔗 렌더링별 각 데이터 합산 결과

In [15]:
```python
import re

# E_combined와 S_combined 파일 리스트 필터링
i_combined_files = [f for f in os.listdir(directory_path) if re.search(r'I_\d+_c
s_combined_files = [f for f in os.listdir(directory_path) if re.search(r'S_\d+_c

# E_combined와 S_combined 데이터프레임 각각 합치기
df_i_combined = pd.concat([pd.read_csv(os.path.join(directory_path, file)) for f
df_s_combined = pd.concat([pd.read_csv(os.path.join(directory_path, file)) for f

# 데이터프레임을 합친 후 각각 동일한 작업 수행
for df_combined, combined_name in [(df_i_combined, 'I_combined'), (df_s_combined

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df_combined.columns and 'result' in df_combined.columns:
        maxforce_values = [0.0, 0.1, 0.3, 0.5, 1.0]
```

```
        result_values = [0.0, 0.1, 0.3, 0.5, 1.0]

        distribution = []

        for maxforce_value in maxforce_values:
            for result_value in result_values:
                count = len(df_combined[(df_combined['maxforce'] == maxforce_val
                distribution.append({
                    'maxforce_value': maxforce_value,
                    'result_value': result_value,
                    'count': count
                })

        distribution_df = pd.DataFrame(distribution)

        # 막대그래프 생성
        plt.figure(figsize=(10, 6))
        sns.barplot(x='result_value', y='count', hue='maxforce_value', data=dist

        # 그래프 세부 설정
        plt.title(f'Maxforce vs Result Distribution - {combined_name}')
        plt.xlabel('Result Value')
        plt.ylabel('Count')
        plt.legend(title='Maxforce Value')

        # 그래프 출력
        plt.show()

    else:
        print(f"'maxforce' or 'result' column missing in {combined_name}")
```





In [17]:
```python
# 결과를 저장할 리스트 초기화
comparison_results = []

# 각 데이터프레임에 대해 작업 수행
for df_combined, combined_name in [(df_i_combined, 'I_combined'), (df_s_combined

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df_combined.columns and 'result' in df_combined.columns:
        # 상관계수 계산
        correlation = df_combined['maxforce'].corr(df_combined['result'])

        # 차이의 절대값 평균 계산
        mean_absolute_difference = (df_combined['maxforce'] - df_combined['resul

        # maxforce와 result가 동일한 값의 비율 계산
        identical_percentage = (df_combined['maxforce'] == df_combined['result']

        # 결과 저장
        comparison_results.append({
            'filename': combined_name,
            'correlation': correlation,
            'mean_absolute_difference': mean_absolute_difference,
            'identical_percentage': identical_percentage
        })

    else:
        print(f"'maxforce' or 'result' column missing in {combined_name}")

# 결과를 데이터프레임으로 변환
results_df = pd.DataFrame(comparison_results)

# 결과 출력
print(results_df)
```

	filename	correlation	mean_absolute_difference	identical_percentage
0	I_combined	0.850776	0.0930	72.5
1	S_combined	0.771905	0.1155	69.5

In [ ]: