

# NFL 팀의 패싱수비 실점 예측하기

# 데이터 수집 출처

- <https://www.kaggle.com/datasets/jedidahwavinya/nfl-team-passing-stats-2025-regular-season>
- Nfl 2025시즌기준

# 전처리 과정

1. 팀 이름 변경 및 불필요한 내용 제거 및 결측치 채우기
2. 공격력 순위 분석하기
3. 팀별 공격력 시각화하기
4. 선형 회귀로 학습 및 평가

# 시각화 과정

- 팀별로 잭슨빌 상대 예상 패스공격 터치다운 수 구하기
- 히스토그램으로 시각화하기
- 예상 실점을 상자그림으로 시각화하기

# 폰트 설치

그래프에 쓰이는 한글이 깨져서 폰트를 설치

```
!sudo apt-get install -y fonts-nanum  
!sudo fc-cache -fv  
!rm ~/.cache/matplotlib -rf
```

```
import matplotlib.pyplot as plt  
import matplotlib.font_manager as fm  
  
font_path = '/usr/share/fonts/truetype/nanum/NanumBarunGothic.ttf'  
plt.rc('font', family=fm.FontProperties(fname=font_path).get_name())
```

# 데이터 불러오기

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns

df_stats = pd.read_csv('nfl_team_stats.csv')
```

# 팀 이름 변경 및 불필요한 내용 제거 및 결측치 채우기

```
df_stats['Team'] = df_stats['Team'].str.replace(r'#[s+([a-zA-Z]+)#$', '', regex=True)
df_stats = df_stats.set_index('Team')

key_cols = ['Att', 'Pass Yds', 'TD', 'INT']
df_analysis = df_stats[key_cols].copy()

for col in df_analysis.columns:
    df_analysis[col] = (df_analysis[col]
                        .astype(str)
                        .str.replace('T', ''))
                        .replace('--', np.nan)
                        .astype(float))

df_analysis = df_analysis.fillna(df_analysis.mean())
print(df_analysis.head())
```

# 공격력 순위 분석하기

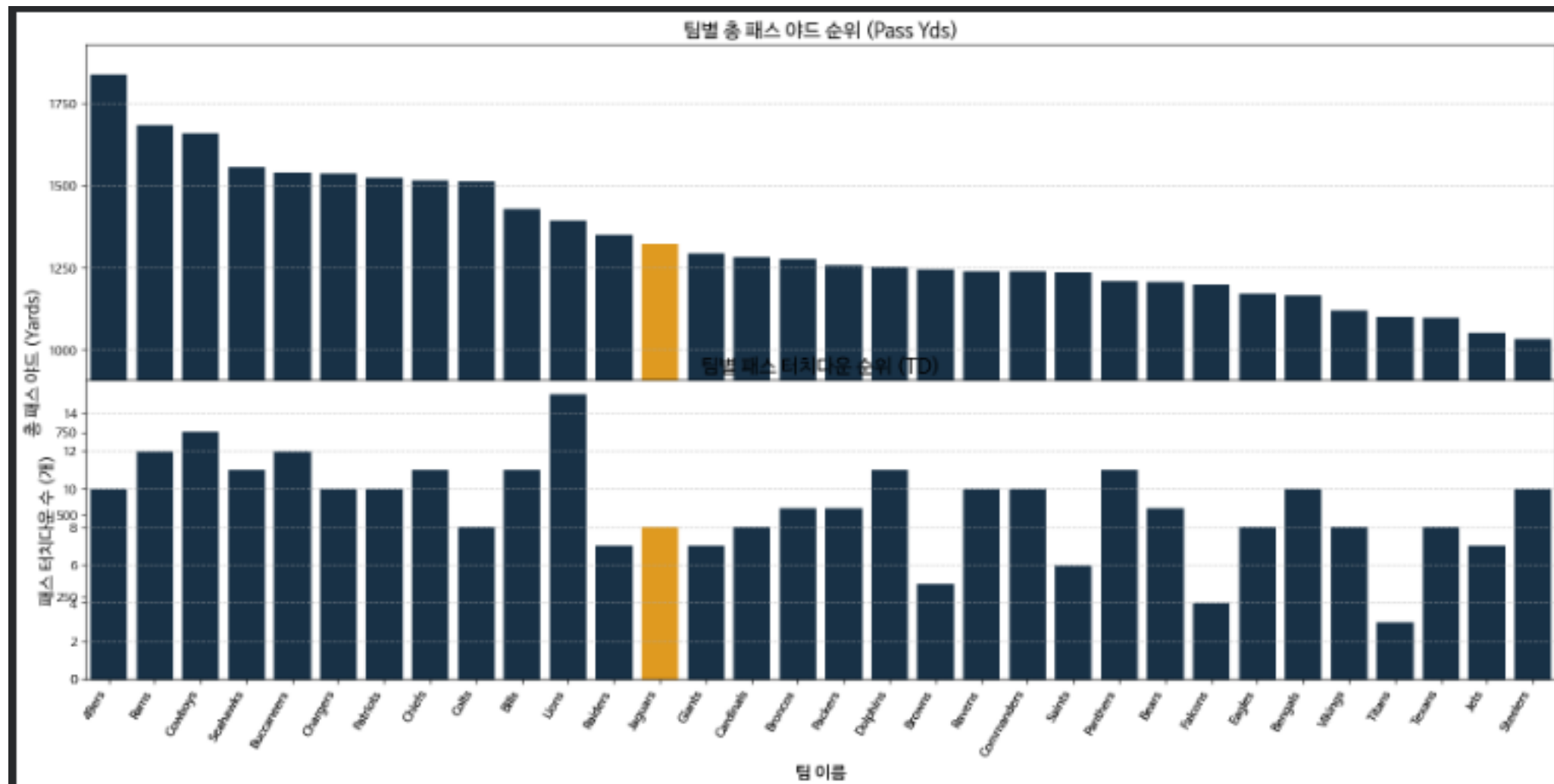
```
df_rank = df_analysis.sort_values(by='Pass Yds', ascending=False)
df_rank_1 = df_rank[['Pass Yds', 'TD']].rename(columns={
    'Pass Yds': '총 패스 야드',
    'TD': '패스 터치다운 수'
})

print(df_rank_1.head(10))

if 'Jaguars' in df_rank.index:
    jax_rank = df_rank.index.get_loc('Jaguars') + 1
    jax_yds = df_rank.loc['Jaguars']['Pass Yds']
    print(f"뎀잭슨빌(Jaguars)은 총 {len(df_rank)}개 팀 중 {jax_rank}위 입니다.")
    print(f"총 패스 야드: {jax_yds:.0f} 야드")
```



# 팀별 공격력 시각화하기



# 선형 회귀로 학습 및 평가

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import numpy as np
import pandas as pd

X = df_analysis[['Pass Yds']]
y = df_analysis['TD']

GAMES_PER_SEASON = 17
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred_test = model.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, y_pred_test))

print(f"모델의 RMSE (오차 지표): {rmse:.2f} TD (시즌 총합 기준)")
```

# 패스 예상 실점 예측하기

```
predicted_total_tds = model.predict(df_analysis[['Pass Yds']])
predicted_td_per_game = predicted_total_tds / GAMES_PER_SEASON
predicted_score_per_game = predicted_td_per_game * 7

df_prediction1 = pd.DataFrame({
    '총 패스 야드': df_analysis['Pass Yds'],
    '예측 경기당 패스 TD': predicted_td_per_game.round(2),
    '예상 실점 (경기당)': predicted_score_per_game.round(1)
}, index=df_analysis.index)

df_pred_rank_corrected = df_prediction1.sort_values(by='예상 실점 (경기당)', ascending=False)

print("\n2026 시즌 잭슨빌 상대 예상 경기당 실점 순위 (상위 10개 팀)")
print(df_pred_rank_corrected.head(10))

print("\n잭슨빌 상대 예상 경기당 실점 평균: ", df_pred_rank_corrected['예상 실점 (경기당)'].mean().round(1),
```

# 2026시즌 잭슨빌 상대 예상 실점 시각화1

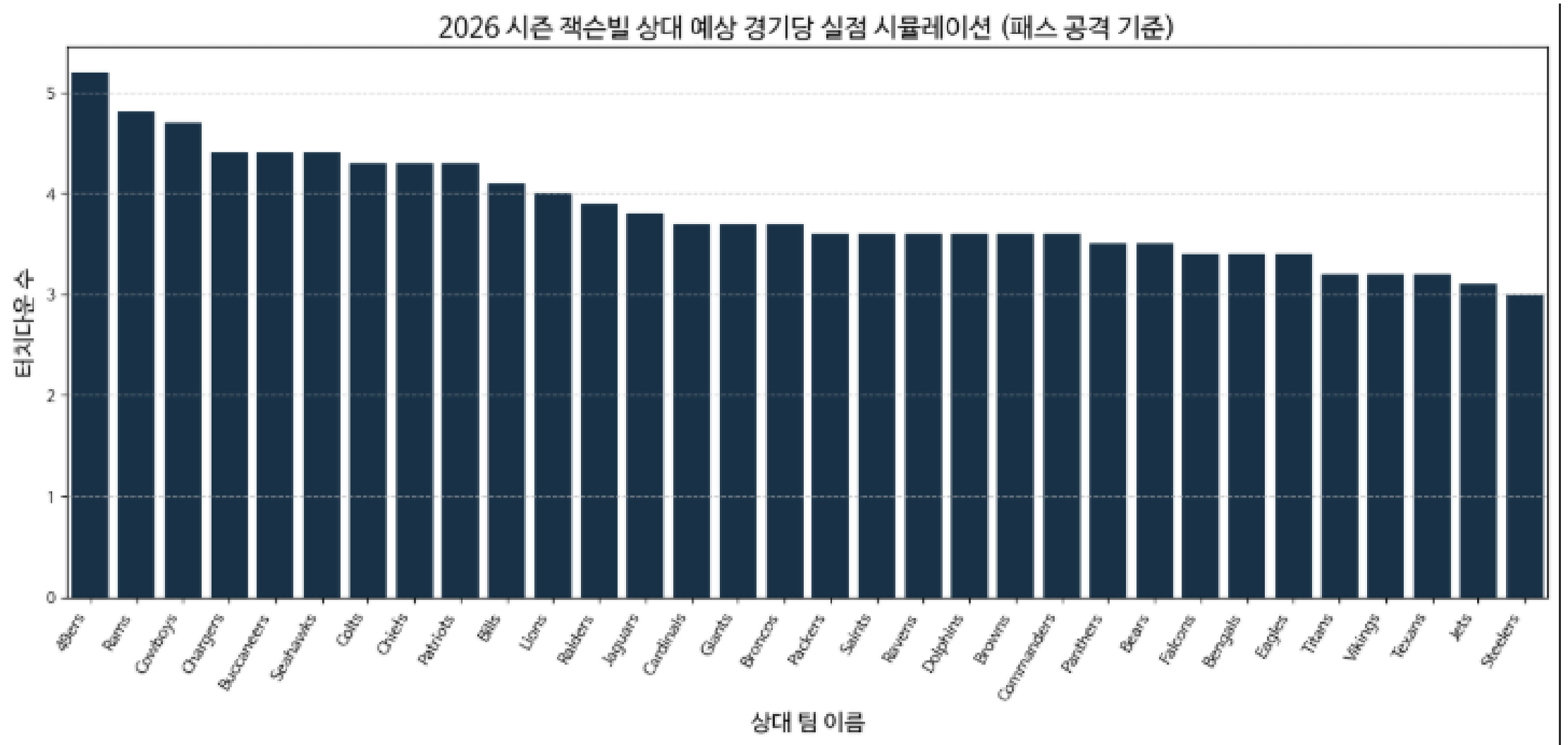
```
plot_data = df_pred_rank_corrected.sort_values(by='예상 실점 (경기당)', ascending=False)

plt.figure(figsize=(16, 6))
sns.barplot(x=plot_data.index, y='예상 실점 (경기당)', data=plot_data, color='#11324D')

plt.title('2026 시즌 잭슨빌 상대 예상 경기당 실점 시뮬레이션 (패스 공격 기준)', fontsize=16)
plt.xlabel('상대 팀 이름', fontsize=14)
plt.ylabel('터치다운 수', fontsize=14)
plt.xticks(rotation=60, ha='right', fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

# 2026시즌 잭슨빌 상대 예상 실점 시각화1

## - 그래프 -



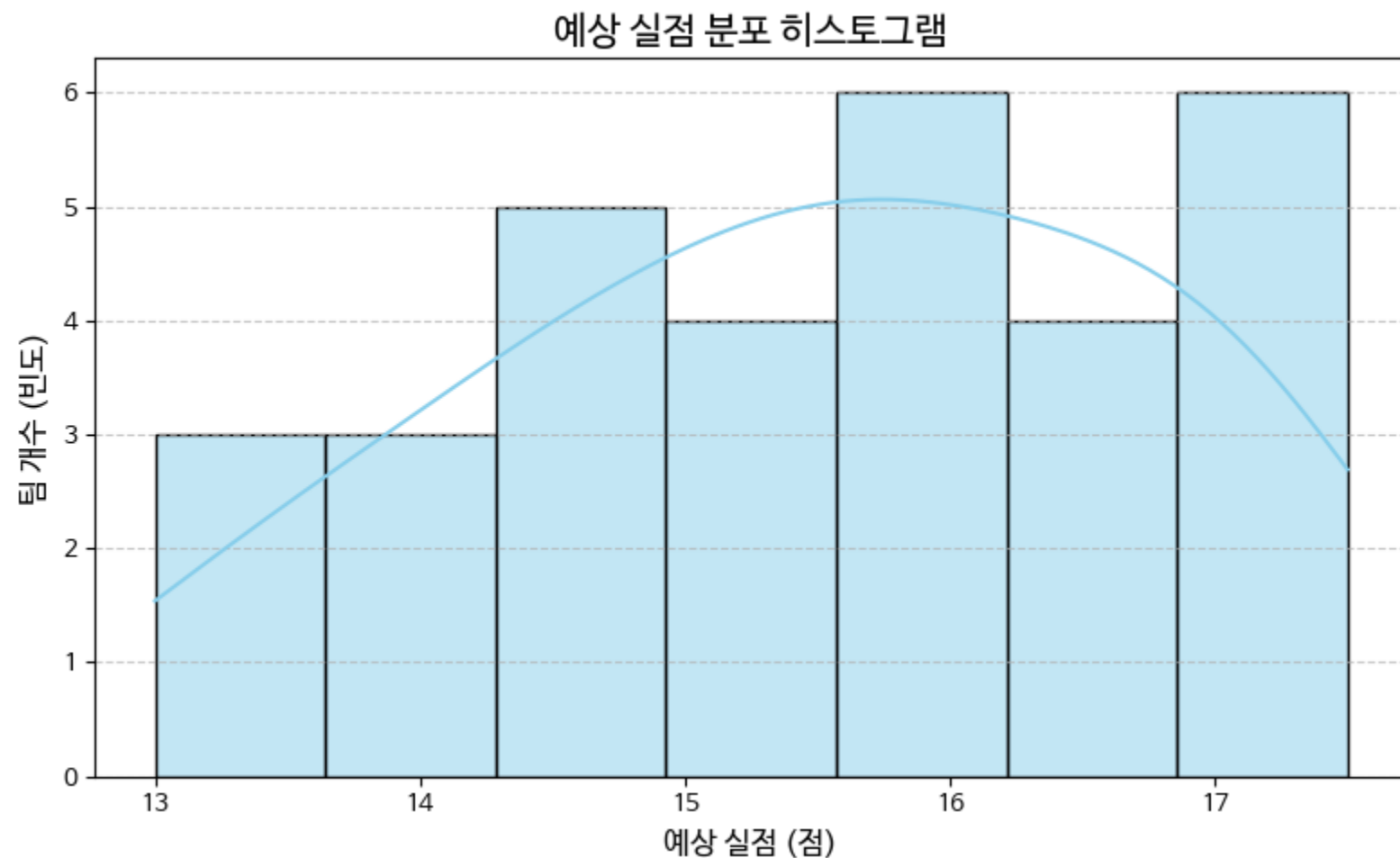
# 2026시즌 잭슨빌 상대 예상 실점 시각화2

```
plt.figure(figsize=(8, 5))
sns.histplot(
    df_prediction['Predicted Score'],
    bins=7,
    kde=True,
    color='skyblue',
    edgecolor='black'
)

plt.title('예상 실점 분포 히스토그램', fontsize=15)
plt.xlabel('예상 실점 (점)', fontsize=12)
plt.ylabel('팀 개수 (빈도)', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

# 2026시즌 잭슨빌 상대 예상 실점 시각화2

## -히스토그램-



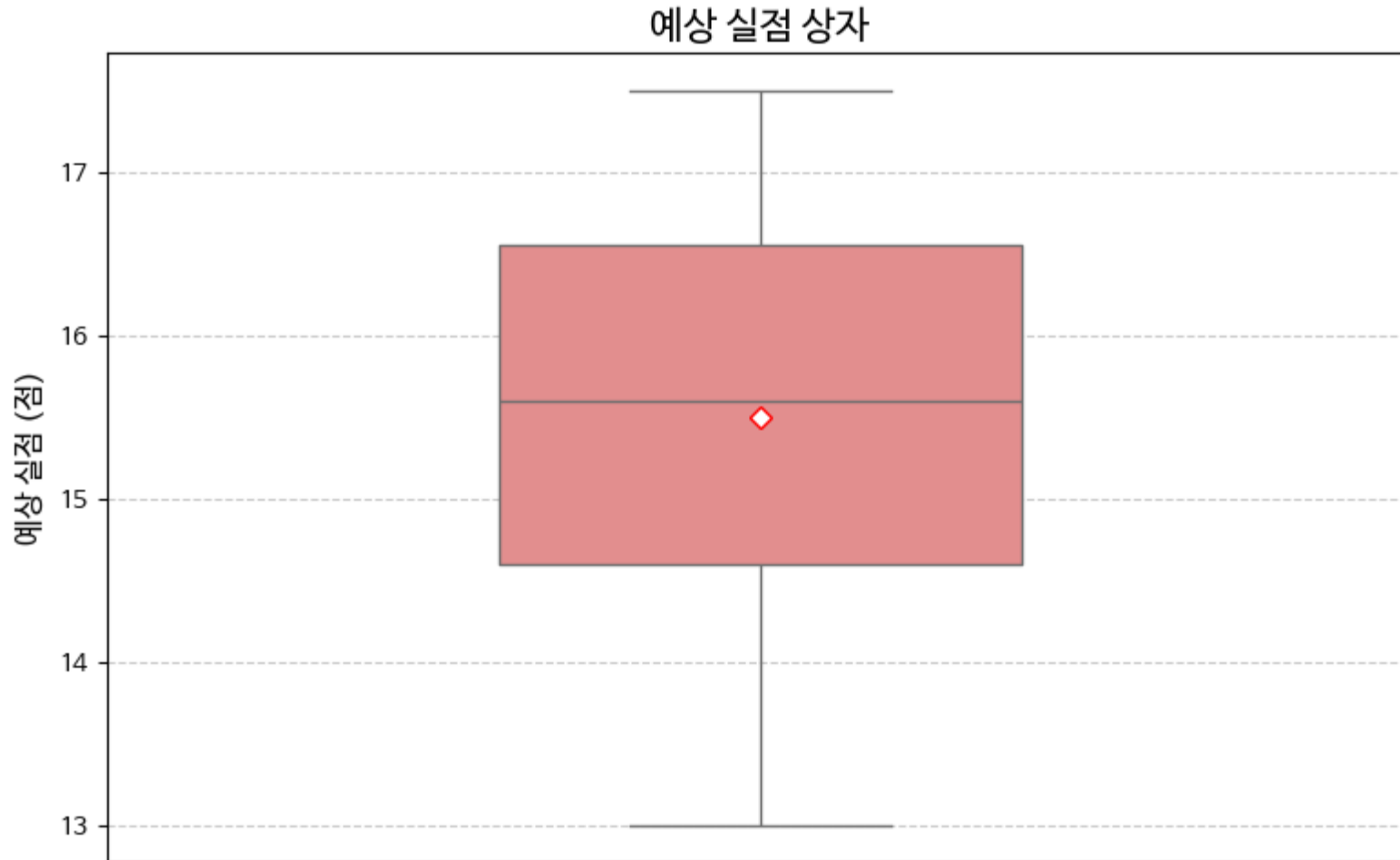
# 2026시즌 잭슨빌 상대 예상 실점 시각화3

```
plt.figure(figsize=(8, 5))
sns.boxplot(
    y=df_prediction['Predicted Score'],
    color='lightcoral',
    width=0.4,
    showmeans=True, meanprops={"marker": "D", "markerfacecolor": "white", "markeredgecolor": "red"}
)

plt.title('예상 실점 상자 그림 (Box Plot)', fontsize=15)
plt.ylabel('예상 실점 (점)', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.xticks([])
plt.tight_layout()
plt.show()
```



# 2026시즌 잭슨빌 상대 예상 실점 시각화3 -상자-



# 결론

- 예상 패스 실점은 작년 실점을 기준으로는 생각보다 비슷하다. 물론 수비팀 보강이라든지, 상대 팀도 공격팀을 보강하거나 탱킹을 할 경우에는 예상 실점은 많이 바뀔 수 있다.
- 예상한 결과로 얻을수 있는점은 그 팀을 패스 수비를 할때 그 팀이 왜 우리팀을 상대로 패스 터치다운 비중이 어떠한지 보면, 그 팀을 상대할때 어떠한 공격에 더 비중을 줘야하는지 수비 코디네이터들이 더 보완할수 있을거 같다.

# 결론

- 시각화 할때 애니메이션을 못써서 많이 아쉬웠다. 수업을 들으면서 이 기능을 무조건을 넣어야겠다고 생각을 했지만, 막상 해보니 이 데이터들로 어떻게 애니메이션을 써야하는지 떠오르지 않아서 사용하지 못했다.