kenpom_analysis

October 6, 2025

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

# Data Loading and Cleaning
df = pd.read_csv("KenpomRatings.csv", header=1)

df.columns = (
    df.columns.astype(str)
    .str.replace('\xa0', ' ', regex=False)
    .str.replace(r'\s+', ' ', regex=True)
    .str.strip()
)

df = df.loc[:, ~df.columns.str.contains('^Unnamed')]

print("Columns:", df.columns.tolist())

# Calculating the Win Percentage of Each Team
df[['Wins', 'Losses']] = df['W-L'].str.split('-', expand=True).astype(int)
df['WinPct'] = df['Wins'] / (df['Wins'] + df['Losses'])

num_cols = ['NetRtg','ORtg','DRtg','AdjT','Luck','Strength of Schedule','NCSOS']
for col in num_cols:
    if col in df.columns:
        df[col] = pd.to_numeric(df[col], errors='coerce')

print(df.head())


# Question 1: Average Tempo by Conference

tempo_by_conf = (
    df.groupby("Conf")["AdjT"]
      .mean()
      .sort_values(ascending=False)
)
```

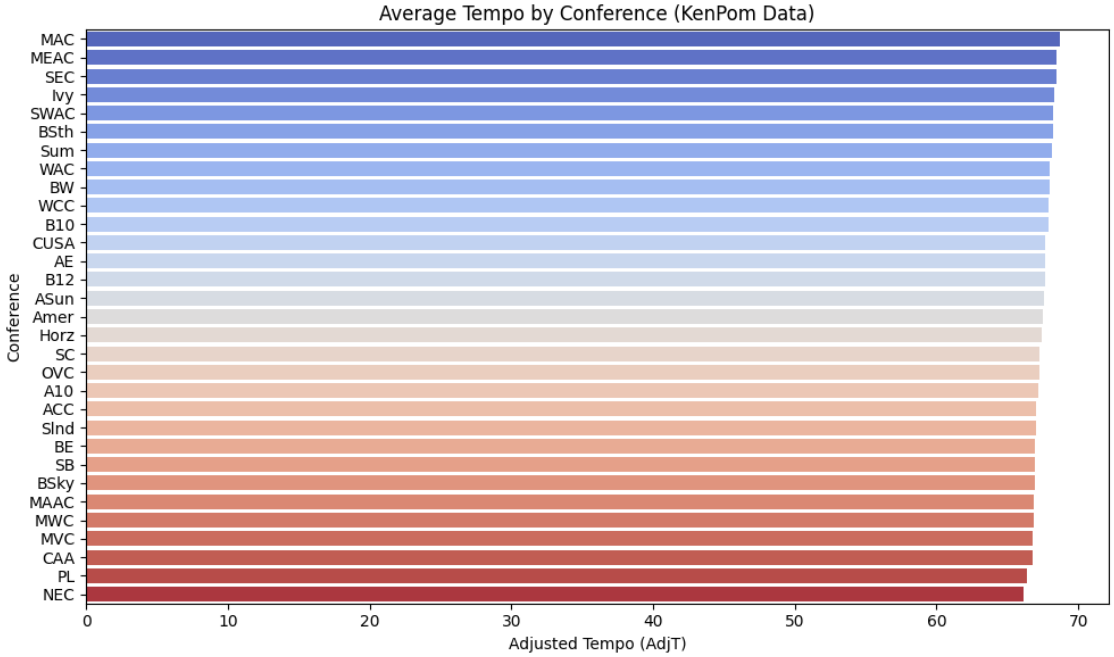
```
print("Average Adjusted Tempo (by Conference):")
print(tempo_by_conf)


plt.figure(figsize=(10,6))
sns.barplot(x=tempo_by_conf.values, y=tempo_by_conf.index, palette="coolwarm")
plt.title("Average Tempo by Conference (KenPom Data)")
plt.xlabel("Adjusted Tempo (AdjT)")
plt.ylabel("Conference")
plt.tight_layout()
plt.show()

# Question 2: Strength of Schedule vs. Offensive Efficiency

sos_col = 'NetRtg.1'
ortg_col = 'ORtg'

df[sos_col]  = pd.to_numeric(df[sos_col], errors='coerce')
df[ortg_col] = pd.to_numeric(df[ortg_col], errors='coerce')


plt.figure(figsize=(8,6))
sns.regplot(x=sos_col, y=ortg_col, data=df, scatter_kws={'alpha':0.6})
plt.title("Strength of Schedule (NetRtg.1) vs Offensive Rating (ORtg)")
plt.xlabel("Strength of Schedule (NetRtg.1)")
plt.ylabel("Offensive Rating (ORtg)")
plt.tight_layout()
plt.show()


corr_val = df[[sos_col, ortg_col]].corr().iloc[0,1]
print(f"Correlation between {sos_col} and {ortg_col}: {corr_val:.3f}")

df['CombinedScore'] = (
    (df[ortg_col] - df[ortg_col].mean()) / df[ortg_col].std() +
    (df[sos_col] - df[sos_col].mean()) / df[sos_col].std()
)

top_combined = df[['Team', 'Conf', ortg_col, sos_col, 'CombinedScore']] \
                  .sort_values('CombinedScore', ascending=False) \
                  .head(10)

print("\nTop 10 Teams Combining High Offensive Efficiency and Strength of␣
 ↪Schedule:")
print(top_combined.to_string(index=False))
```

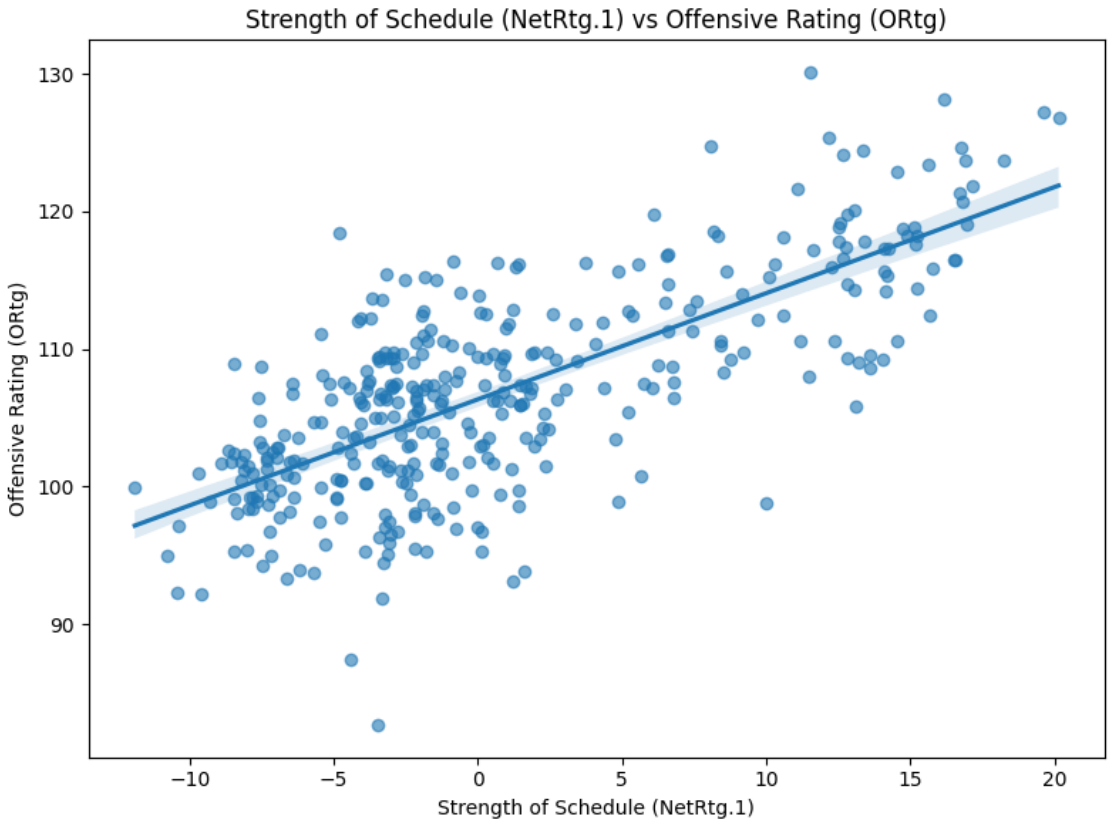
```
# Question 3: Underrated Teams

ortg_col = 'ORtg'
drtg_col = 'DRtg'
win_col  = 'WinPct'

for col in [ortg_col, drtg_col, win_col]:
    df[col] = pd.to_numeric(df[col], errors='coerce')


df['z_ORtg'] = (df[ortg_col] - df[ortg_col].mean()) / df[ortg_col].std()
df['z_DRtg'] = (df[drtg_col] - df[drtg_col].mean()) / df[drtg_col].std()
df['z_WinPct'] = (df[win_col] - df[win_col].mean()) / df[win_col].std()


df['UnderratedIndex'] = df['z_ORtg'] - df['z_DRtg'] - df['z_WinPct']


underrated = df[['Team', 'Conf', ortg_col, drtg_col, win_col,␣
 ↪'UnderratedIndex']] \
                .sort_values('UnderratedIndex', ascending=False) \
                .head(10)

print("\n Top 10 'Underrated' Teams (strong both ends, underperformed record):")
print(underrated.to_string(index=False))

plt.figure(figsize=(8,6))
sns.scatterplot(x='WinPct', y='UnderratedIndex', data=df, alpha=0.6)
plt.title("Underrated Index vs Win Percentage")
plt.xlabel("Win Percentage")
plt.ylabel("Underrated Index (High = Underrated)")
plt.tight_layout()
plt.show()

# Scaled version that auto-penalizes high-win teams
# Magnifies gaps for low-win teams even without a hard cutoff.
df['UnderratedIndex_scaled'] = (df['z_ORtg'] - df['z_DRtg']) * (1 - df[win_col])

underrated_scaled = (
    df[['Team','Conf', ortg_col, drtg_col, win_col, 'UnderratedIndex_scaled']]
    .sort_values('UnderratedIndex_scaled', ascending=False)
    .head(10)
)

print("\n Top 10 by Scaled Underrated Index ( (z_ORtg - z_DRtg) * (1 - WinPct)␣
 ↪):")
print(underrated_scaled.to_string(index=False))
```

```
Columns: ['Rk', 'Team', 'Conf', 'W-L', 'NetRtg', 'ORtg', 'DRtg', 'AdjT', 'Luck',
'NetRtg.1', 'ORtg.1', 'DRtg.1', 'NetRtg.2']
   Rk          Team Conf   W-L  NetRtg   ORtg  DRtg  AdjT   Luck  NetRtg.1  \
0   1        Duke 1  ACC  35-4   39.29  130.1  90.8  66.0 -0.026     11.51
1   2     Houston 1  B12  35-5   36.59  123.4  86.8  61.9  0.020     15.63
2   3     Florida 1  SEC  36-4   36.46  128.2  91.8  70.1  0.032     16.15
3   4      Auburn 1  SEC  32-6   35.05  127.2  92.1  68.2  0.026     19.63
4   5  Tennessee 2  SEC  30-8   30.93  120.7  89.7  63.7  0.029     16.80

   ORtg.1  DRtg.1  NetRtg.2  Wins  Losses    WinPct
0   113.4   101.9      9.46    35       4  0.897436
1   115.0    99.4      2.62    35       5  0.875000
2   115.6    99.5     -2.20    36       4  0.900000
3   117.4    97.8     10.51    32       6  0.842105
4   117.0   100.2     -1.04    30       8  0.789474
Average Adjusted Tempo (by Conference):
Conf
MAC     68.716667
MEAC    68.500000
SEC     68.481250
Ivy     68.312500
SWAC    68.258333
BSth    68.233333
Sum     68.122222
WAC     67.966667
BW      67.945455
WCC     67.936364
B10     67.866667
CUSA    67.680000
AE      67.644444
B12     67.643750
ASun    67.550000
Amer    67.515385
Horz    67.400000
SC      67.270000
OVC     67.245455
A10     67.180000
ACC     66.983333
Slnd    66.983333
BE      66.972727
SB      66.950000
BSky    66.940000
MAAC    66.892308
MWC     66.845455
MVC     66.800000
CAA     66.785714
PL      66.400000
NEC     66.155556
```

```
Name: AdjT, dtype: float64
```

```
/var/folders/vy/mvkd6mt13wx7vcvmbwpshs8m0000gn/T/ipykernel_24160/1398921258.py:4
4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=tempo_by_conf.values, y=tempo_by_conf.index, palette="coolwarm")
```





Correlation between NetRtg.1 and ORtg: 0.728

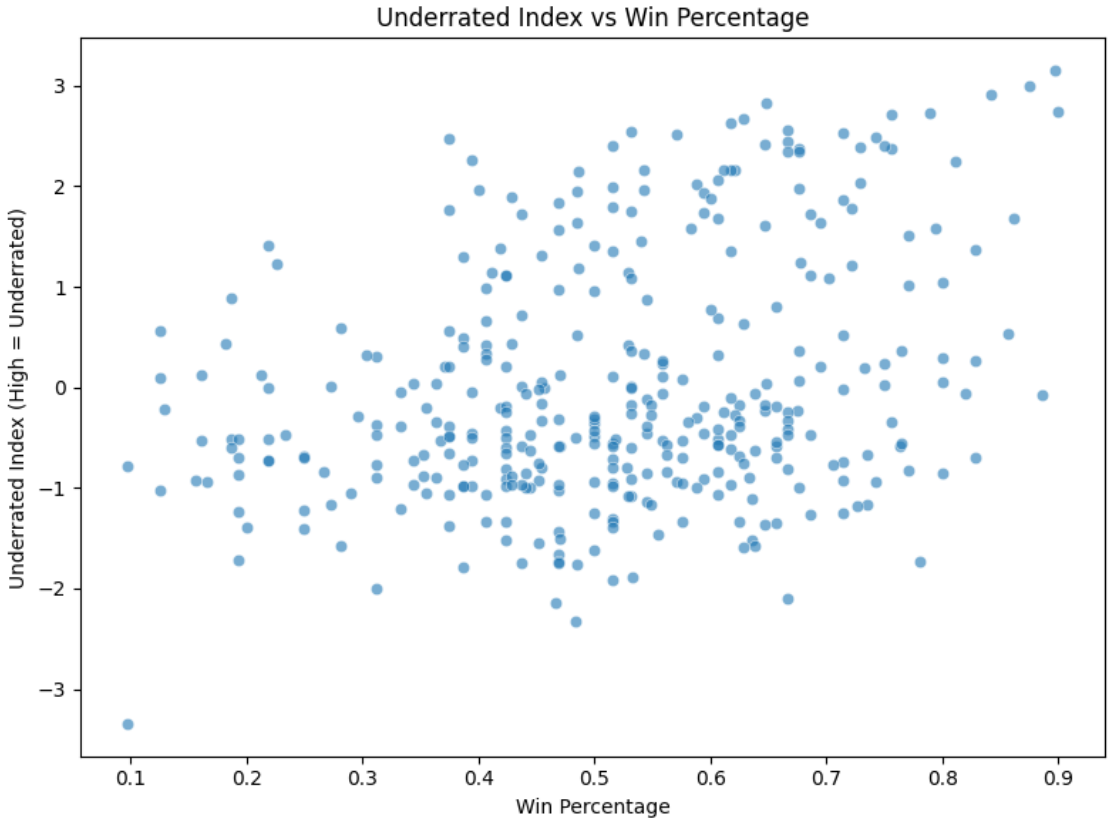
Top 10 Teams Combining High Offensive Efficiency and Strength of Schedule:

Team	Conf	ORtg	NetRtg.1	CombinedScore
Alabama 2	SEC	126.8	20.14	5.301965
Auburn 1	SEC	127.2	19.63	5.283587
Florida 1	SEC	128.2	16.15	4.931893
Kentucky 3	SEC	123.7	18.23	4.631498
Purdue 4	B10	124.6	16.76	4.545428
Duke 1	ACC	130.1	11.51	4.537116
Arizona 4	B12	123.7	16.93	4.451238
Illinois 6	B10	121.9	17.14	4.244833
Houston 1	B12	123.4	15.63	4.231725
Baylor 9	B12	121.3	16.72	4.108087

Top 10 'Underrated' Teams (strong both ends, underperformed record):

Team	Conf	ORtg	DRtg	WinPct	UnderratedIndex
Duke 1	ACC	130.1	90.8	0.897436	3.147509
Houston 1	B12	123.4	86.8	0.875000	2.998022
Auburn 1	SEC	127.2	92.1	0.842105	2.904974
Arizona 4	B12	123.7	97.3	0.648649	2.828600

Florida	1	SEC	128.2	91.8	0.900000	2.735252
Tennessee	2	SEC	120.7	89.7	0.789474	2.724252
Alabama	2	SEC	126.8	96.5	0.756757	2.708545
Illinois	6	B10	121.9	97.6	0.628571	2.668235
Kansas	7	B12	115.9	93.0	0.617647	2.630746
Purdue	4	B10	124.6	99.2	0.666667	2.557074



Top 10 by Scaled Underrated Index ((z_ORtg - z_DRtg) * (1 - WinPct)):

Team	Conf	ORtg	DRtg	WinPct	UnderratedIndex_scaled
Arizona 4	B12	123.7	97.3	0.648649	1.272891
Illinois 6	B10	121.9	97.6	0.628571	1.241615
Kansas 7	B12	115.9	93.0	0.617647	1.238902
Ohio St.	B10	118.2	99.1	0.531250	1.235685
Baylor 9	B12	121.3	100.4	0.571429	1.220949
Northwestern	B10	114.3	97.3	0.515152	1.160186
Purdue 4	B10	124.6	99.2	0.666667	1.152904
Missouri 6	SEC	124.4	100.8	0.647059	1.127703
Kentucky 3	SEC	123.7	99.2	0.666667	1.113650
Mississippi 6	SEC	119.1	95.8	0.666667	1.081165