In [19]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.feature_selection import SelectKBest, f_regression

# Load the dataset
df = pd.read_csv('nba_team_stats_00_to_23.csv')

# Display the first few rows of the dataset and column names
print(df.head())
print(df.columns)

# Check for missing values
print(df.isnull().sum())

# Inspect the column names to identify non-numeric columns
print(df.columns)

# Ensure column names are correctly identified
columns_to_drop = ['Team']
for col in df.columns:
    if 'season' in col.lower():
        columns_to_drop.append(col)

print(f"Columns to drop: {columns_to_drop}")

# Drop non-numeric columns if they exist (adjust as necessary based on actual column names)
df = df.drop(columns=columns_to_drop)

# If there are missing values, handle them (e.g., fill with mean, median, or drop)
df = df.fillna(df.mean())

# Verify all columns are numeric now
print(df.dtypes)

# Define the features (X) and the target variable (y)
X = df.drop(columns=['win_percentage', 'wins', 'losses', 'plus_minus', 'Min', 'games_played']) # Adjust based on your dataset's column names
y = df['win_percentage']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Normalize/scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Feature selection using correlation
correlation_matrix = df.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Feature selection using SelectKBest
selector = SelectKBest(score_func=f_regression, k='all')
selector.fit(X_train_scaled, y_train)
feature_scores = pd.DataFrame({'Feature': X.columns, 'Score': selector.scores_})
feature_scores = feature_scores.sort_values(by='Score', ascending=False)
print(feature_scores)

# Fit a linear regression model
lr = LinearRegression()
lr.fit(X_train_scaled, y_train)
y_pred_lr = lr.predict(X_test_scaled)

# Evaluate the linear regression model
print('Linear Regression RMSE:', np.sqrt(mean_squared_error(y_test, y_pred_lr)))
print('Linear Regression R^2:', r2_score(y_test, y_pred_lr))

# Fit a random forest regressor model
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train_scaled, y_train)
y_pred_rf = rf.predict(X_test_scaled)

# Evaluate the random forest model
print('Random Forest RMSE:', np.sqrt(mean_squared_error(y_test, y_pred_rf)))
print('Random Forest R^2:', r2_score(y_test, y_pred_rf))

# Feature importance from random forest
feature_importances = pd.DataFrame({'Feature': X.columns, 'Importance': rf.feature_importances_})
feature_importances = feature_importances.sort_values(by='Importance', ascending=False)
print(feature_importances)

# Plot feature importance
plt.figure(figsize=(12, 8))
sns.barplot(x='Importance', y='Feature', data=feature_importances)
plt.title('Feature Importance from Random Forest')
plt.show()
```

```
   teamstatspk            Team  games_played  wins  losses  \
0            0  Boston Celtics            82    64      18
1            1   Denver Nuggets            82    57      25
2            2  Oklahoma City Thunder      82    57      25
3            3  Minnesota Timberwolves     82    56      26
4            4       LA Clippers           82    51      31

   win_percentage    Min  points  field_goals_made  field_goals_attempted  ...  \
0           0.780  3966    9887              3601                   7396  ...
1           0.695  3941    9418              3610                   7279  ...
2           0.695  3961    9847              3653                   7324  ...
3           0.683  3961    9264              3383                   6974  ...
4           0.622  3941    9481              3473                   7108  ...

   rebounds  assists  turnovers  steals  blocks  blocks_attempted  \
0      3799     2207        979     557     538               384
1      3643     2415       1036     585     456               394
2      3447     2223       1039     694     538               419
3      3577     2184       1162     647     497               371
4      3523     2097       1078     640     413               384

   personal_fouls  personal_fouls_drawn  plus_minus  season
0            1326                  1416         930  2023-24
1            1489                  1467         431  2023-24
2            1545                  1548         608  2023-24
3            1544                  1630         529  2023-24
4            1519                  1537         269  2023-24

[5 rows x 29 columns]
Index(['teamstatspk', 'Team', 'games_played', 'wins', 'losses',
       'win_percentage', 'Min', 'points', 'field_goals_made',
       'field_goals_attempted', 'field_goal_percentage', 'three_pointers_made',
       'three_pointers_attempted', 'three_point_percentage',
       'free_throws_made', 'free_throws_attempted', 'free_throw_percentage',
       'offensive_rebounds', 'defensive_rebounds', 'rebounds', 'assists',
       'turnovers', 'steals', 'blocks', 'blocks_attempted', 'personal_fouls',
       'personal_fouls_drawn', 'plus_minus', 'season'],
      dtype='object')
teamstatspk               0
Team                      0
games_played              0
wins                      0
losses                    0
win_percentage            0
Min                       0
points                    0
field_goals_made          0
field_goals_attempted     0
field_goal_percentage     0
three_pointers_made       0
three_pointers_attempted  0
three_point_percentage    0
free_throws_made          0
free_throws_attempted     0
free_throw_percentage     0
offensive_rebounds        0
defensive_rebounds        0
rebounds                  0
assists                   0
turnovers                 0
steals                    0
blocks                    0
blocks_attempted          0
personal_fouls            0
personal_fouls_drawn      0
plus_minus                0
season                    0
dtype: int64
Index(['teamstatspk', 'Team', 'games_played', 'wins', 'losses',
       'win_percentage', 'Min', 'points', 'field_goals_made',
       'field_goals_attempted', 'field_goal_percentage', 'three_pointers_made',
       'three_pointers_attempted', 'three_point_percentage',
       'free_throws_made', 'free_throws_attempted', 'free_throw_percentage',
       'offensive_rebounds', 'defensive_rebounds', 'rebounds', 'assists',
       'turnovers', 'steals', 'blocks', 'blocks_attempted', 'personal_fouls',
       'personal_fouls_drawn', 'plus_minus', 'season'],
      dtype='object')
Columns to drop: ['Team', 'season']
teamstatspk               int64
games_played              int64
wins                      int64
losses                    int64
win_percentage          float64
Min                       int64
points                    int64
field_goals_made          int64
field_goals_attempted     int64
field_goal_percentage   float64
three_pointers_made       int64
three_pointers_attempted  int64
three_point_percentage  float64
free_throws_made          int64
free_throws_attempted     int64
free_throw_percentage   float64
offensive_rebounds        int64
defensive_rebounds        int64
rebounds                  int64
assists                   int64
turnovers                 int64
steals                    int64
blocks                    int64
blocks_attempted          int64
personal_fouls            int64
personal_fouls_drawn      int64
plus_minus                int64
dtype: object
```



```
                    Feature       Score
4      field_goal_percentage  229.871594
7    three_point_percentage  197.396318
18       blocks_attempted  121.122450
15              turnovers   54.557410
12      defensive_rebounds   45.935441
1                   points   36.177063
17                  blocks   31.782392
14                 assists   31.027409
2       field_goals_made   25.383699
13               rebounds   19.315771
16                 steals   15.958365
5     three_pointers_made   15.795790
9      free_throws_made   15.729232
19       personal_fouls   12.699336
10  free_throw_percentage   10.539699
8    free_throws_attempted    8.502649
11    offensive_rebounds    7.485912
6   three_pointers_attempted    5.102940
3     field_goals_attempted    1.201633
0               teamstatspk    0.041154
20   personal_fouls_drawn    0.000336
Linear Regression RMSE: 0.05931544413440501
Linear Regression R^2: 0.8266906413378285
Random Forest RMSE: 0.03747374501891919
Random Forest R^2: 0.9230993911730262
                    Feature  Importance
4      field_goal_percentage    0.255391
7    three_point_percentage    0.135597
15              turnovers    0.072308
18       blocks_attempted    0.064347
13               rebounds    0.051049
16                 steals    0.048786
17                 blocks    0.046012
2       field_goals_attempted    0.041481
0               teamstatspk    0.038122
12      defensive_rebounds    0.034206
8    free_throws_made    0.031029
9    free_throw_percentage    0.027258
20   personal_fouls_drawn    0.026108
14                 assists    0.021721
11    offensive_rebounds    0.019896
10  free_throw_percentage    0.019088
19       personal_fouls    0.016410
6   three_pointers_attempted    0.014089
1                   points    0.013989
5     three_pointers_made    0.012205
3       field_goals_made    0.011257
```


In [ ]: