In [3]:

```
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_playoffs_00_to_21.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', 'games_played', 'minutes'])  # 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()
```

```
   team_abbreviation  team  games_played  wins  losses  \
0                  0  Milwaukee Bucks             23    16       7
1                  1       Phoenix Suns             22    14       8
2                  2     Brooklyn Nets              12     7       5
3                  3  Philadelphia 76ers             12     7       5
4                  4      Atlanta Hawks              18    10       8

   win_percentage  minutes  points  field_goals_made  field_goals_attempted  \
0           0.696     48.4   116.3              42.1                   91.1
1           0.636     48.0   109.9              40.9                   85.0
2           0.583     48.4   112.5              40.7                   86.2
3           0.583     48.0   116.3              42.5                   85.8
4           0.556     48.0   106.3              38.9                   86.8

   ...  rebounds  assists  turnovers  steals  blocks  blocks_attempted  \
0  ...      49.0     22.8       12.7     7.8     4.2               3.8
1  ...      42.7     23.2       11.9     6.5     4.2               3.6
2  ...      42.6     22.6       11.6     7.1     4.8               6.2
3  ...      44.7     24.9       12.8     8.0     6.2               3.8
4  ...      42.4     20.2       12.4     6.7     4.4               4.3

   personal_fouls  personal_fouls_drawn  plus_minus   season
0            18.0                  20.1         5.1  2020-21
1            19.7                  18.5         4.5  2020-21
2            21.2                  18.6         6.3  2020-21
3            23.0                  23.8         7.5  2020-21
4            21.2                  20.3        -1.5  2020-21

[5 rows x 29 columns]
Index(['teamstatspk', 'team', 'games_played', 'wins', 'losses',
       'win_percentage', 'minutes', '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
minutes                  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', 'minutes', '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
minutes                  float64
points                   float64
field_goals_made         float64
field_goals_attempted    float64
field_goal_percentage    float64
three_pointers_made      float64
three_pointers_attempted float64
three_point_percentage   float64
free_throws_made         float64
free_throws_attempted    float64
free_throw_percentage    float64
offensive_rebounds       float64
defensive_rebounds       float64
rebounds                 float64
assists                  float64
turnovers                float64
steals                   float64
blocks                   float64
blocks_attempted         float64
personal_fouls           float64
personal_fouls_drawn     float64
plus_minus               float64
dtype: object
```



```
                    Feature      Score
4      field_goal_percentage  77.738809
7     three_point_percentage  47.724431
11                    points  31.978178
12     defensive_rebounds     27.939550
17                    blocks  22.895380
2          field_goals_made   19.982412
16                   assists  19.566526
13                 rebounds   17.939962
18     blocks_attempted       13.426696
10          personal_fouls     12.217235
5      three_pointers_made    11.393331
9      free_throws_made       9.368876
16                   steals   9.128651
9      free_throw_attempted   5.290554
15                 turnovers  4.841836
19     free_throw_percentage  4.484435
6   three_pointers_attempted  3.036432
0             teamstatspk     1.193557
3      field_goals_attempted   1.075842
20     personal_fouls_drawn    0.441941
14     offensive_rebounds      0.172470
Linear Regression RMSE: 0.15655787480262988
Linear Regression R^2: 0.3504988891546974
Random Forest RMSE: 0.15493914202793602
Random Forest R^2: 0.3540944811763896615
                    Feature  Importance
4      field_goal_percentage    0.191815
7     three_point_percentage    0.094311
16                   steals    0.080896
17                   blocks    0.077771
13                 rebounds    0.067683
12     defensive_rebounds      0.059681
10          personal_fouls     0.050866
3      field_goals_attempted   0.043242
19     free_throw_percentage   0.040166
15                 turnovers   0.038041
8      free_throws_made        0.032832
14     offensive_rebounds      0.029105
18     blocks_attempted        0.028215
11                   points    0.027338
20     personal_fouls_drawn    0.025479
16                   assists   0.025396
9      free_throws_attempted   0.025227
2          field_goals_made    0.019342
0             teamstatspk      0.017644
6   three_pointers_attempted   0.013633
5      three_pointers_made     0.012119
```


In [ ]: