Week 2.1

In [ ]:
```
#Pre-requisite Code to compute up to Self Test
import pandas as pd
import numpy as np
NBA_Teams=pd.read_csv("../../Data/Week 2/nba_teams.csv")
NBA_Teams.rename(columns={'Unnamed: 0':'TEAM_NUMBER'}, inplace=True)
NBA_Teams.rename(columns={'ID':'TEAM_ID'}, inplace=True)
display(NBA_Teams)
```

Self Test - 1 Solution

- Rename "FULL_NAME" to "TEAM_NAME"

In [ ]:
```
NBA_Teams.rename(columns={'FULL_NAME':'TEAM_NAME'}, inplace=True)
display(NBA_Teams)
```

Dropping Columns and Rows

In [ ]:
```
#Pre-requisite Code to compute up to Self Test
NBA_Teams.drop(['TEAM_NUMBER'], axis=1, inplace=True)
display(NBA_Teams)
Games=pd.read_csv("../../Data/Week 2/basketball_games.csv")
Games.drop([0], axis=0, inplace=True)
Games=Games[Games.TEAM_NAME !="Las Vegas Aces"]
Games.head()
```

Self Test - 2 Solution

- Drop all the Phoenix Mercury games

In [ ]:
```
Games=Games[Games.TEAM_NAME !="Phoenix Mercury"]
Games.head()
```

Merging Dataframes

In [ ]:
```
#Pre-requisite Code to compute up to Self Test
NBA_Games=pd.merge(NBA_Teams, Games, on=['TEAM_ID', 'TEAM_NAME'])
NBA_Games.head()
NBA_Games.columns
NBA_Games.drop(['ABBREVIATION'], axis=1, inplace=True, errors='ignore')
```

Self Test - 3 Solution

- Find the number of observations and the number of variables in the dataset

In [ ]:
```
NBA_Games.shape
```

In [ ]:
```
NBA_Games.sort_values(by=['GAME_ID'], ascending=[False]).head(20)
```

Missing Values

In [ ]:
```
#Pre-requisite Code to compute up to Self Test
NBA_Games.info()
NBA_Games.notnull()
NBA_Games=NBA_Games[pd.notnull(NBA_Games["FG_PCT"])]
NBA_Games.shape
NBA_Games=NBA_Games.fillna(NBA_Games.mean())
NBA_Games.info()
NBA_Games['GM']=NBA_Games['FGM']+NBA_Games['FG3M']+NBA_Games['FTM']
```

Self Test - 4 Solution

- Create a variable called "GA" equals to the total number of goals attempted.

In [ ]:
```
NBA_Games['GA']=NBA_Games['FGA']+NBA_Games['FG3A']+NBA_Games['FTA']
```

Week 2.2

In [ ]:
```
#Pre-requisite Code to compute up to Self Test
import pandas as pd
NBA_Games=pd.read_csv("../../Data/Week 2/NBA_Games.csv")
NBA_Games.head()
NBA_Games.dtypes
dummy=pd.get_dummies(NBA_Games, columns=['WL'])
dummy.columns
NBA_Games=pd.concat([NBA_Games, dummy['WL_W']], axis=1)
NBA_Games.head()
NBA_Games.rename(columns={'WL_W':'WIN'}, inplace=True)
NBA_Games.head()
NBA_Games['GAME_DATE'].dtype
import datetime
NBA_Games['GAME_DATE']=pd.to_datetime(NBA_Games['GAME_DATE'])
NBA_Games['GAME_DATE'].head()
NBA_Games.describe()
NBA_Games.describe(include='all')
NBA_Games['PTS'].describe()
NBA_Games['FGM'].mean()
```

```
NBA_Games['FGM'].median()
NBA_Games['FGM'].std()
```

Self Test

1. Find the mean of field goals attempted;
2. Find the median of 3-point field goals made;
3. Find the standard deviation of the number of rebounds

In [ ]:

```
NBA_Games['FGA'].mean()
NBA_Games['FG3M'].median()
NBA_Games['REB'].std()
```

Week 2.3

In [ ]:

```
#Pre-requisite Code to compute up to Self Test
import pandas as pd
NBA_Games=pd.read_csv("../../Data/Week 2/NBA_Games2.csv")
NBA_Games.head()
NBA_Games['FG_PCT'].describe()
NBA_Games['FG3_PCT'].describe()
NBA_Games.hist(column=['FG_PCT','FG3_PCT'], bins=20, sharex=True, sharey=True)
import matplotlib.pyplot as plt
NBA_Games[['FG_PCT','FG3_PCT']].plot.hist(alpha=0.3, bins=20)
plt.xlabel('Field Goal Percentage')
plt.ylabel('Frequency')
plt.title("Distributions of Field Goal Percentages", fontsize=15)
plt.savefig('FG_PCT_Distributions.png')
NBA_Games.hist(by='WL', column='FG_PCT', color='red', bins=15, sharex=True, sharey=True)
plt.savefig('FG_PCT_WL.png')
```

Self Test - 1 Solution

1. Calculate summary statistics for the three-point field goal percentage by the result of the game
2. Graph a histogram of the three-point field goal percentage by the result of the game and provide interpretation
3. Number of bins=10, the two subgraphs should have the same x and y ranges, color is green
4. Export the graph as "FG3_PCT_Distribution" in png format

In [ ]:

```
NBA_Games.groupby(['WL'])['FG3_PCT'].describe()
NBA_Games.hist(by='WL', column='FG3_PCT', bins=10, color='green', sharex=True, sharey=True)
plt.savefig('FG3_PCT_Distribution.png')
```

In [ ]:

```
#Pre-requisite Code to compute up to Self Test
import datetime
NBA_Games['GAME_DATE']=pd.to_datetime(NBA_Games['GAME_DATE'])
NBA_Games['GAME_DATE'].head()
Pistons_Games=NBA_Games[(NBA_Games.NICKNAME == 'Pistons')&(NBA_Games.SEASON_ID==22017)& (NBA_Games.GAME_DATE>='2017-10-17')]
display(Pistons_Games)
Pistons_Games.plot(x='GAME_DATE', y='PTS')
plt.savefig('PISTONS_PTS_TIME.png')
```

Self Test - 2

1. Graph Toronto Raptors' points in each game throughout the 2018-2019 seaon. (SEASON ID is 22018, and the regular season started on October 16, 2018.)
2. Export the graph as "RAPTORS_PTS_TIME" in png format

In [ ]:

```
Raptors_Games=NBA_Games[(NBA_Games.NICKNAME == 'Raptors')&(NBA_Games.SEASON_ID==22018)& (NBA_Games.GAME_DATE>='2018-10-16')]
Raptors_Games.plot(x='GAME_DATE', y='PTS')
plt.savefig('RAPTORS_PTS_TIME.png')
```