Import Updated NBA Game Data

In [ ]:
```
import pandas as pd
NBA_Games=pd.read_csv("../../Data/Week 2/NBA_Games2.csv")
NBA_Games.head()
```

More on Summary Statistics

Central Tendency vs. Variation

We will compare the success rates of two-point field goals and three-point field goals to demonstrate the difference between central tendency and variation.

Calculate summary statistics for the percentages of two-point field goals and three-point field goals

- Two-point field goals

In [ ]:
```
NBA_Games['FG_PCT'].describe()
```

- Three-point field goals

In [ ]:
```
NBA_Games['FG3_PCT'].describe()
```

We can see that the average success rate of 2-point field goals is about 45.27% while the average success rate of 3-point field goals is 35.07%. That means that the overall success rate of 2-point field goals is about 10% higher than the overall success rate of 3-point field goals. The median of 2-point field goal success rate is 45.20%, while the median 3-point field goal success rate is 35.00%. This means half of the teams have 2-point field-goal success rates less than 45% and half of the teams have 3-point field goal success rate of less than 35%.

The standard deviation for 2-point field goal success rate is 0.056, while the standard deviation for 3-point field goal success rate is 0.09956. This means that there is a greater variation in 3-point field goals than 2-point field goals.

Compare the distribution of two-point field goal percentage and three-point field goal percentage using a Histogram

Plot two histograms side by side

The options "sharex" and "sharey" ask if we want to restrict the same range of x and same range of y for the two histograms

In [ ]:
```
NBA_Games.hist(column=['FG_PCT','FG3_PCT'], bins=20, sharex=True, sharey=True)
```

Plot two histograms in the same graph in different colors

We will first introduce a new library "matplotlib" that provides more useful functions to make plots.

- We will use "plot.hist" instead of "hist" to make this plot
- The option "alpha" specifies transparency, so that the two histograms would not block each other entirely (alpha=0: fully transparent; alpha=1: fully opaque)
- We can also add a title and axis labels using "plt.title," "plt.xlabel" and "plt.ylabel" commands
- We can also export the graph as a png file using the "plt.savefig" command

In [ ]:
```
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')
```

Histogram by the result of the game using the "by" option

We can also change the colors of the graphs using the "color" option

In [ ]:
```
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

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 [ ]:
```
#Your Code Here
```

Create time series graphs

Let's first change the data type of "GAME_DATE" from object to datetime.

In [ ]:
```
import datetime
NBA_Games['GAME_DATE']=pd.to_datetime(NBA_Games['GAME_DATE'])
NBA_Games['GAME_DATE'].head()
```

Subsetting a dataset

The dataset we are working with contains games of different NBA teams. Let's focus on one team to produce a time series graph.

Extract Pistons' game data in the 2017-2018 season.

Note that for date variable, we can use the >, =, < operators. When we specify the condition of the date, we need to use ""

In [ ]:
```
Pistons_Games=NBA_Games[(NBA_Games.NICKNAME == 'Pistons')&(NBA_Games.SEASON_ID==22017)& (NBA_Games.GAME_DATE>='2017-10-17')]
display(Pistons_Games)
```

Now we can plot the points earned by the Pistons by time.

In [ ]:
```
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 [ ]:
```
#Your Code Here
```

Correlation Analysis

We can first detect the relationship between two variables in a scatterplot.

Let's use the number of assists and the number of field goals made as an example.

We can create a scatter plot using the "plot.scatter" function with the number of assists in the horizontal axis and the number of field goals made in the vertical axis.

In [ ]:
```
NBA_Games.plot.scatter(x='AST', y='FGM')
```

We can use the functions in the "seaborn" library to graph the relationships between two variables

We will use the function "regplot" to graph the two variables. This function graphs a scatterplot as well as a regression line.

We will learn about regression analysis more systematically in week 4

In [ ]:
```
import seaborn as sns
sns.regplot(x='AST', y='FGM', data=NBA_Games,  marker='.')
plt.xlabel('Assists')
plt.ylabel('Field Goals Made')
plt.title("Relationship between the Numbers of Assists and Field Goals Made", fontsize=15)
```

As we can see from the graph, as the number of assists increase, the number of field goals made also increases. In this case, we say there is a positive relationship between the two variables, or a positive correlation.

Correlation Coefficient

We can quantify the linear correlation by a correlation coefficient. A correlation coefficient measures the joint variability of two random variables. We can calculate correlation coefficient using the "corr" function.

In [ ]:
```
NBA_Games['AST'].corr(NBA_Games['FGM'])
```

The correlation coefficient between the number of assist and field goal made is 0.70 so there is a positive correlation between the two.

Let's investigate the relationship between the number of assists and the number of field goals attempted.

In [ ]:
```
sns.regplot(x='AST', y='FGA', data=NBA_Games,  marker='.')
plt.xlabel('Assists')
plt.ylabel('Field Goals Attempted')
plt.title("Relationship between the Numbers of Assists and Field Goals Attempted", fontsize=15)
```

In [ ]:
```
NBA_Games['AST'].corr(NBA_Games['FGA'])
```

Both the graph and the correlation coefficient suggest that there is only a slight positive relationship between the two.

We can further graph the scatter plot by group using the "hue" option.

Let's separate by the results of the game (win or lose), and produce scatter plots between number of assists and field goals made.

In this case, we can use lmplot() instead of regplot().

- lmplot() combines regplot() and FacetGrid.
- FacetGrid produces multi-plot grid for plotting conditional relationships. Thus, FacetGrid allows us to separate the dataset into multiple panels based on specified conditions to visualize the relationship between multiple variables.

In [ ]:
```
sns.lmplot(x='AST', y='FGA', hue='WL', data=NBA_Games)
plt.xlabel('Assists')
plt.ylabel('Field Goals Made')
plt.title("Relationship between the Numbers of Assists and Field Goals Made", fontsize=15)
```

We can also find correlation coefficients for all the numerical variables.

We will specify the method to be pearson.

In [ ]:
```
NBA_Games.corr(method='pearson')
```

In [ ]:
```

```