# CS240 – Exploratory Data Analysis

# by Mehmet BAYSAN

# Final Project Report


## Abdülkadir Kaan IŞILAY

## 214050639

## Introduction:

I wonder which conference's all-star teams have had more points between 1950 and 2009. I will analyse scores of each player who played in an all-star team. Thinkstats2, Thinkplot, Numpy and Pandas will be used.

## Part 1:

- Is there any relationship between player heights and having a place in all-star team?
- Is there any relationship between team winning times and coach awards in one season?
- Is there any relationship between years and average points of east conference and west conference all-star teams?

I will analyse 3rd question.

## Part 2:

The dataset file is basketball_player_allstar.csv.

Only "conference" column is used. Rows for "West" conference and "East" conference are used as datasets. dropna() function cleans the empty cells.

```
In [2]: #creating data frame
        df = pd.read_csv('basketball_player_allstar.csv')
```

```
In [3]: #selecting 'conference' column and defining rows which will be used
        west = df.points[df.conference == "West"].dropna()
        east = df.points[df.conference == "East"].dropna()
```

## Part 3:

- Showing maximum and minimum values, means, variances and standard deviations.

```
In [4]: #Showing maximum and minimum values, means, variances and standard deviations (general info)
print ("Minumum value of the West is %.2f" %west.min())
print ("Maximum of the West is %.2f" %west.max())
print ("Mean of the West is %.2f" %west.mean())
print ("Variance of the West is %.2f" %west.var())
print ("Standart deviation of the West is %.2f" %west.std())

print ("Minumum value of the East is %.2f" %east.min())
print ("Maximum of the East is %.2f" %east.max())
print ("Mean of the East is %.2f" %east.mean())
print ("Variance of the East is %.2f" %east.var())
print ("Standart deviation of the East is %.2f" %east.std())
```

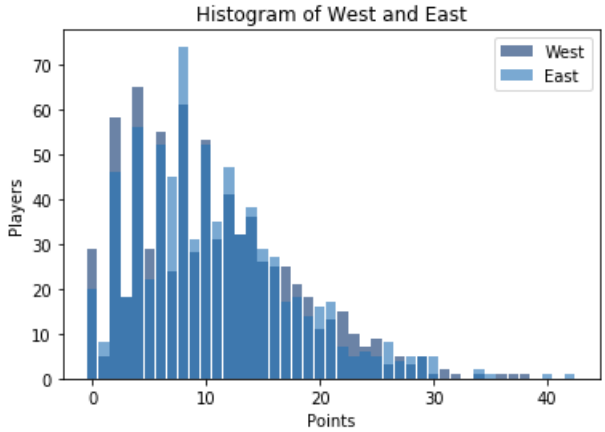
```
Minumum value of the West is 0.00
Maximum of the West is 38.00
Mean of the West is 10.65
Variance of the West is 49.58
Standart deviation of the West is 7.04
Minumum value of the East is 0.00
Maximum of the East is 42.00
Mean of the East is 10.79
Variance of the East is 47.13
Standart deviation of the East is 6.87
```

- Histogram

-West and East datasets are analysed by thinkstats2.Hist() function.

- thinkplot.Hist() function plotted the analysed data.

```
In [4]: #analyzing columns, then plotting them
hist_west_1 = thinkstats2.Hist(west,label="West")
hist_east_1 = thinkstats2.Hist(east,label="East")

thinkplot.Hist(hist_west_1)
thinkplot.Hist(hist_east_1)

thinkplot.Show(xlabel = 'Points',ylabel='Players',title='Histogram of West and East')
```




```
<matplotlib.figure.Figure at 0xd4da090>
```

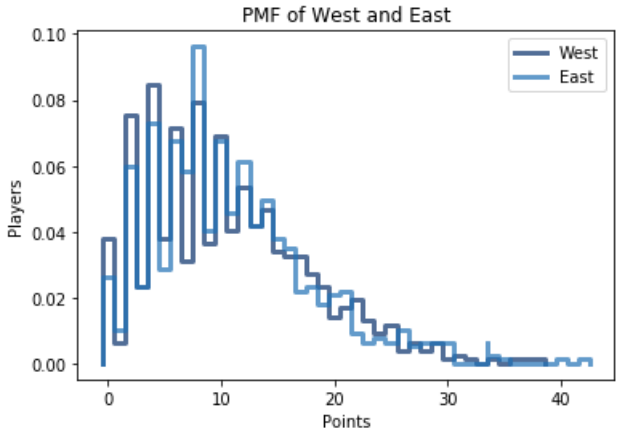
-In both East and West conferences' teams number of players for each score are similar.

- PMF

  -West and East datasets are analysed by thinkstats2.Pmf() function.

  - thinkplot.Pmf() function plotted the analysed data.

```
In [5]: #analyzing columns with probability mass function, then plotting them
        pmf_west_1 = thinkstats2.Pmf(west,label="West")
        pmf_east_1 = thinkstats2.Pmf(east,label="East")

        thinkplot.Pmf(pmf_west_1)
        thinkplot.Pmf(pmf_east_1)

        thinkplot.Show(xlabel = 'Points',ylabel='Players',title='PMF of West and East')
```




```
<matplotlib.figure.Figure at 0xd45a870>
```

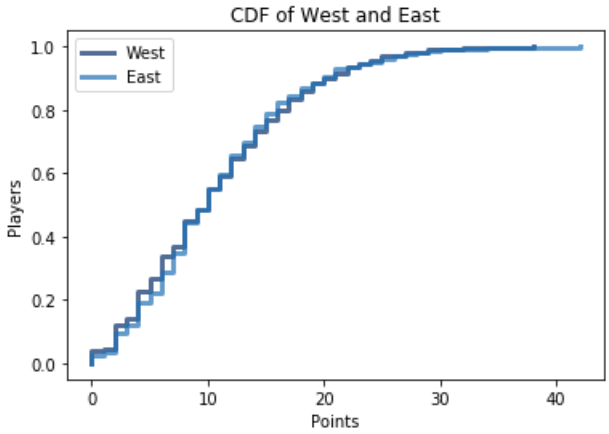
--In both East and West conferences' teams' number of players for each score are similar. Mid-score is 10. Number East conference players is higher than number West conference players on mid-score.

- CDF

-West and East datasets are analysed by thinkstats2.Cdf() function.

- thinkplot.Cdf() function plotted the analysed data.

```
In [7]: #analyzing columns with cumulative distribution function, then plotting them
        cdf_west_1 = thinkstats2.Cdf(west,label="West")
        cdf_east_1 = thinkstats2.Cdf(east,label="East")

        thinkplot.Cdf(cdf_west_1)
        thinkplot.Cdf(cdf_east_1)

        thinkplot.Show(xlabel = 'Points',ylabel='Players',title='CDF of West and East')
```




```
<matplotlib.figure.Figure at 0xe2f5490>
```

- Graphic show us percentile of the points of both West and East conference players.

## Part 4:

-Estimating means of bot conferences' data to see average points of conferences.

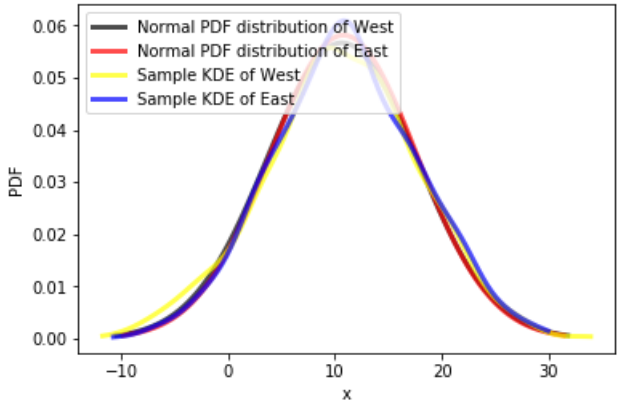
-Calculating standard deviations and medians to use in normal distribution analyse.

In [10]:
```
#showing means and standard deviations
mean_of_west, std_of_west = west.mean(), west.std()
mean_of_east, std_of_east = east.mean(), east.std()

print ('Mean of West: ' +str(mean_of_west) +', Std of West: '+ str(std_of_west))
print ('Mean of East: ' +str(mean_of_east) +', Std of East: '+ str(std_of_east))

#analyzing pdf values
pdf_west = thinkstats2.NormalPdf(mean_of_west, std_of_west)
pdf_east = thinkstats2.NormalPdf(mean_of_east, std_of_east)

#fuction to calculate median via using cdf
def Median(x):
    cdf = thinkstats2.Cdf(x)
    return cdf.Value(0.5)

#defining medians via using Median(x) function
median_west = Median(pdf_west)
median_east = Median(pdf_east)

#showing values
print ('Median :' + str(median_west))
print ('Median :' + str(median_east))

print ('Density of Pdf of West :' +str(pdf_west.Density(mean_of_west+ std_of_west)))
print ('Density of Pdf of East :' +str(pdf_east.Density(mean_of_east+ std_of_east)))
```

```
Mean of West: 10.6540962289, Std of West: 7.04154242958
Mean of East: 10.7932379714, Std of East: 6.86515628971
Median :10.6540962289
Median :10.7932379714
Density of Pdf of West :0.0343633127172
Density of Pdf of East :0.0352462077057
```

-Comparing real pdf values and estimated pdf values.

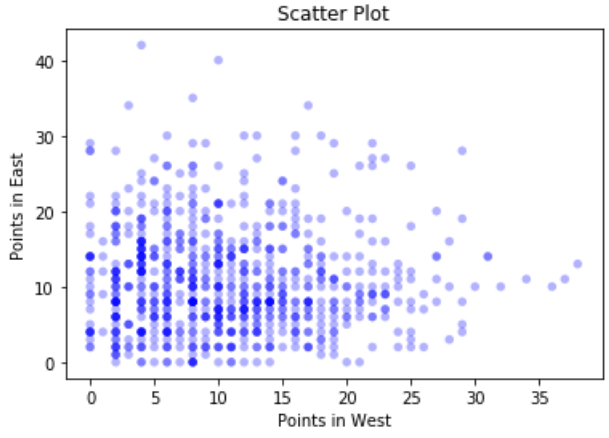
In [11]:
```
##comparing real pdf values and estimated pdf values

#plotting pdf analysis
thinkplot.Pdf(pdf_west, label='Normal PDF distribution of West', color='black')
thinkplot.Pdf(pdf_east, label='Normal PDF distribution of East', color='red')

#defining normals of data
norm_west = np.random.normal(mean_of_west, std_of_west, 1000)
norm_east = np.random.normal(mean_of_east, std_of_east, 1000)

#calculating estimated pdf values
pdf_west = thinkstats2.EstimatedPdf(norm_west)
pdf_east = thinkstats2.EstimatedPdf(norm_east)

#plotting estimated pdf analysis
thinkplot.Pdf(pdf_west, label='Sample KDE of West', color='yellow')
thinkplot.Pdf(pdf_east, label='Sample KDE of East', color='blue')

thinkplot.Show(xlabel='x', ylabel='PDF', loc='upper left')
```




```
<matplotlib.figure.Figure at 0xd81ec50>
```

**Part 4:**

In [15]:
```
#scatter ploting the sample rows
thinkplot.Scatter(west_sample, east, alpha=0.3)
thinkplot.Config(xlabel='Points in West',
                 ylabel='Points in East',
                 title='Scatter Plot',
                 legend=False)
```




-There is no clear condensation, but there is a density at Plot[2:15][4:12].

-Taking random rows as samples to test.

In [14]:
```
#func to take random sample rows from data
def SampleRows(df, nrows, replace=False):
    indices = np.random.choice(df.index, nrows, replace=replace)
    sample = df.loc[indices]
    return sample

west_sample = SampleRows(west, 769)
east_sample = SampleRows(east, 769)
```

-Covariance calculation for east rows and sample west rows.

-Covariance calculation for west rows and sample east rows.

In [16]:
```
#func to calculate covariance
def Covariance(x, y, meanx=None, meany=None):
    x = np.asarray(x)
    y = np.asarray(y)

    if meanx is None:
        meanx = np.mean(x)
    if meany is None:
        meany = np.mean(y)

    covariance = np.dot(x-meanx, y-meany) / len(x)
    return covariance
```

In [17]:
```
#showing covariance between random west data and east data
cov1 = Covariance(west_sample,east)
print('Covariance between random west data and east data is :', cov1)
```

```
Covariance between random west data and east data is : -1.04811274332
```

In [18]:
```
#showing covariance between random east data and west data
cov2 = Covariance(east_sample,west)
print('Covariance between random east data and west data is :', cov2)
```

```
Covariance between random east data and west data is : -0.836149154239
```

-Correlation calculation for east rows and sample west rows.

-Correlation calculation for west rows and sample east rows.

In [19]:
```
#func to calculate correlation
def Correlation(x, y):
    x = np.asarray(x)
    y = np.asarray(y)

    meanx, varx = thinkstats2.MeanVar(x)
    meany, vary = thinkstats2.MeanVar(y)

    correlation = Covariance(x, y, meanx, meany) / np.sqrt(varx * vary)
    return correlation
```

In [20]:
```
#showing covariance between random west data and east data
cor1 = Correlation(west_sample,east)*100
print('Correlation between random west data and east data is :', cor1)
```

```
Correlation between random west data and east data is : -2.17097534769
```

In [21]:
```
#showing covariance between random east data and west data
cor2 = Correlation(east_sample,west)*100
print('Correlation between random east data and west data is :', cor2)
```

```
Correlation between random east data and west data is : -1.73193123776
```