CS 240 Final Project Report

Amer Nour Eddin | 213171245

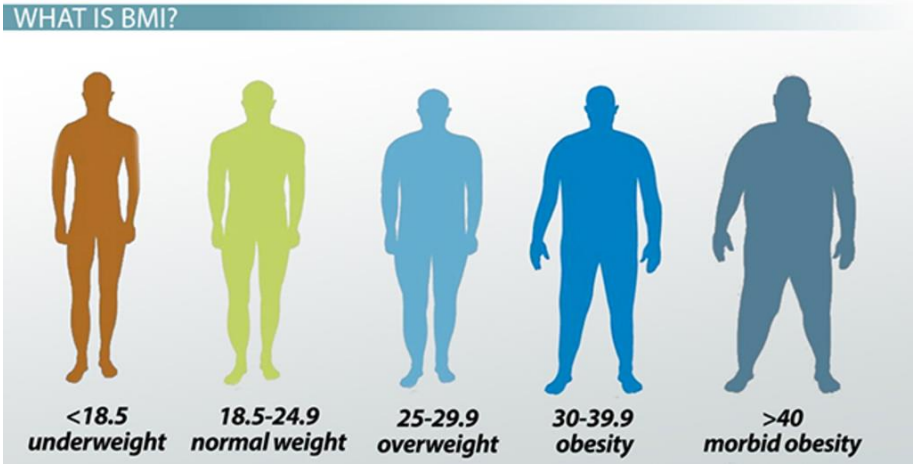
PART 1

I choose the data of the baseball players (Master.csv) which contains a general information about all the players like the (name, Id, birthdate place, weight, height etc.).

Also I choose the (Salaries.csv) data which contains the players Ids and each player salary throughout the years.

My intention is the following:

I will calculate the BMI (Body Mass Index) for each player, which is calculated using the player's weight and height → *BMI = weight (lb.) / (height (in))^2 x 703*



My question is there any relation between the player's BMI and his average salary throughout the years?

"If the player's BMI value is HIGHER than 26 then his salary value will be larger than other players"

PART 2

These are the two main columns that I used:

```
In [5]: FilteredPlayers
```

Out[5]:

	playerID	nameFirst	nameLast	weight	height
0	aardsda01	David	Aardsma	215.0	75.0
1	aaronha01	Hank	Aaron	180.0	72.0
2	aaronto01	Tommie	Aaron	190.0	75.0
3	aasedo01	Don	Aase	190.0	75.0
4	abadan01	Andy	Abad	184.0	73.0
5	abadfe01	Fernando	Abad	220.0	73.0
6	abadijo01	John	Abadie	192.0	72.0
7	abbated01	Ed	Abbaticchio	170.0	71.0
8	abbeybe01	Bert	Abbey	175.0	71.0
9	abbeych01	Charlie	Abbey	169.0	68.0

```
In [456]: FilteredSalaries
```

Out[456]:

	yearID	playerID	salary
0	1985	barkele01	870000
1	1985	bedrost01	550000
2	1985	benedbr01	545000
3	1985	campri01	633333
4	1985	ceronri01	625000
5	1985	chambch01	800000
6	1985	dedmoje01	150000
7	1985	forstte01	483333
8	1985	garbege01	772000
9	1985	harpete01	250000

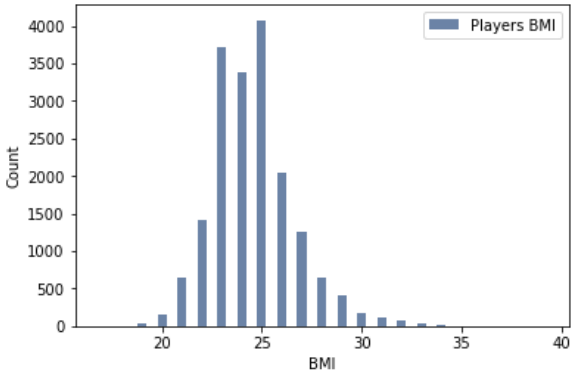
```
In [4]: #cleaning and filtering
FilteredPlayers = Players.drop(['birthYear','birthDay','birthCountry','birthState', 'birthCity',
                'birthMonth','deathYear','deathMonth','deathDay','deathCountry',
                'deathState','deathCity','nameGiven','bats','throws','debut',
                'finalGame','retroID','bbrefID'],axis=1).dropna()
FilteredSalaries = Salaries.drop(['teamID', 'lgID', ],axis=1).dropna()
```

PART 3

I computed the BMI values and add it to the players data frame as the following…

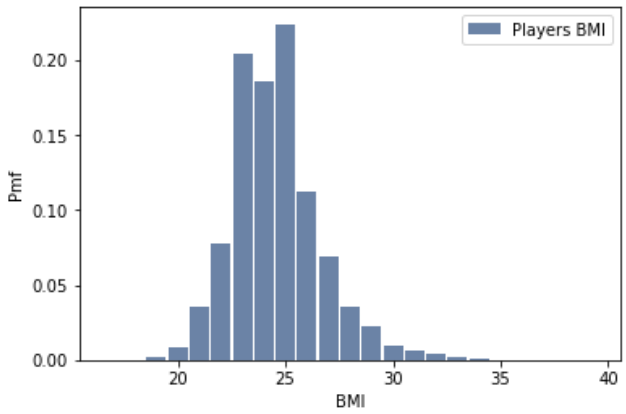
Also I made the histogram chart for the BMI values.

```
n [6]: # Computing the BMI value for each player and add it to the Players dataframe
FilteredPlayers['BMI'] = (FilteredPlayers['weight']*703/FilteredPlayers['height']**2).astype(int)

# #BMI VALUES ABOVE 26
# Above26 = FilteredPlayers[FilteredPlayers.BMI >26.0]
# #BMI VALUES BELOW 26
# Below26 = FilteredPlayers[FilteredPlayers.BMI <26.0]

#plotting the histogram for the BMI values for all players
hist = thinkstats2.Hist(FilteredPlayers.BMI, label='Players BMI')
thinkplot.Hist(hist, width = 0.45)
thinkplot.Config(xlabel='BMI', ylabel='Count')
```



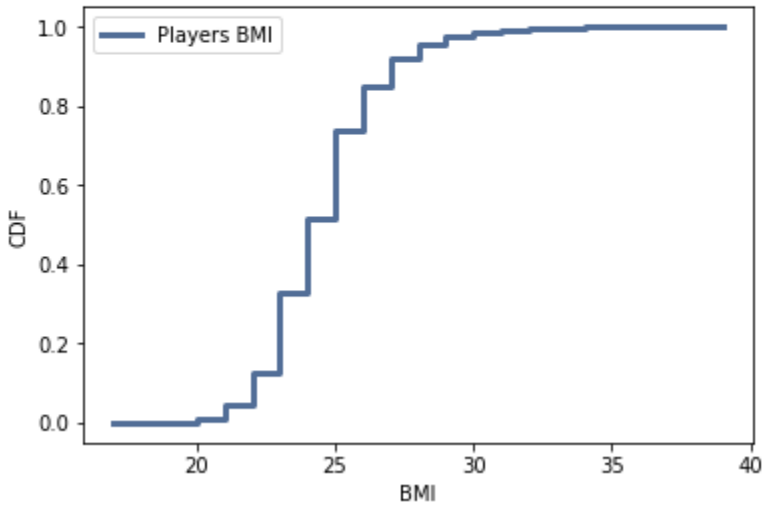
Probability Mass Function (PMF)

```
In [7]: #plotting PMF probability mass function
pmf1 = thinkstats2.Pmf(FilteredPlayers.BMI, label='Players BMI')
thinkplot.Hist(pmf1)
thinkplot.Config(xlabel='BMI', ylabel='Pmf')
```



Cumulative Distribution Function (CDF)

```
#plotting the CDF
cdf = thinkstats2.Cdf(FilteredPlayers.BMI, label='Players BMI')
thinkplot.Cdf(cdf)
thinkplot.Config(xlabel='BMI', ylabel='CDF', loc='upper left')
```

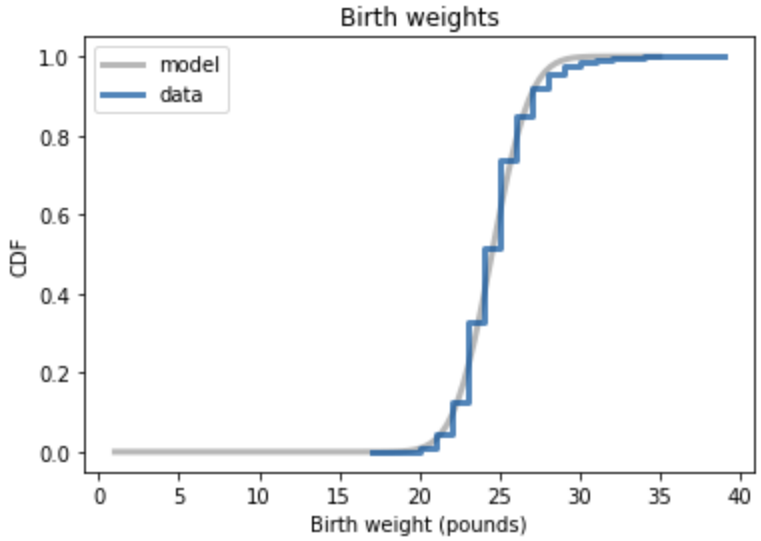


PART 4

I used the Normal distribution for modeling as I saw that it is the most that fits the data shape

```
#Modeling
#Normal distirbution
mu, var = thinkstats2.TrimmedMeanVar(bmi, p=0.01)
print('Mean, Var', mu, var)

# plot the model
sigma = np.sqrt(var)
print('Sigma', sigma)
xs, ps = thinkstats2.RenderNormalCdf(mu, sigma, low=1, high=35)

thinkplot.Plot(xs, ps, label='model', color='0.6')

# plot the data
cdf = thinkstats2.Cdf(bmi, label='data')

thinkplot.PrePlot(1)
thinkplot.Cdf(cdf)
thinkplot.Config(title='Birth weights',
                 xlabel='Birth weight (pounds)',
                 ylabel='CDF')
```

```
Mean, Var 24.5332249201 3.54447172292
Sigma 1.88267674414
```



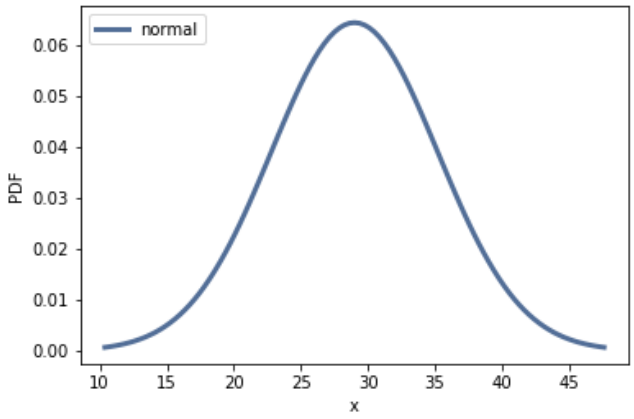

```
BMIs = pd.Series(g2.BMI)

# Here are the mean and standard deviation of the players BMI values
mean, std = BMIs.mean(), BMIs.std()

mean, std
```

```
(29.0, 6.2048368229954285)
```

```
#NormalPdf returns a Pdf object that represents the normal distribution with the given parameters.
pdf = thinkstats2.NormalPdf(mean, std)

#Density returns a probability density, which doesn't mean much by itself.

pdf.Density(mean + std)
```

```
0.038997113287876944
```

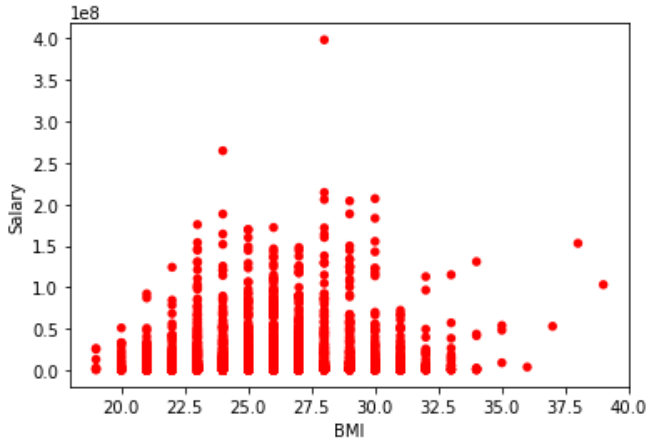
```
# Plotting the PDF with normal distirbution
thinkplot.Pdf(pdf, label='normal')
thinkplot.Config(xlabel='x', ylabel='PDF')
```

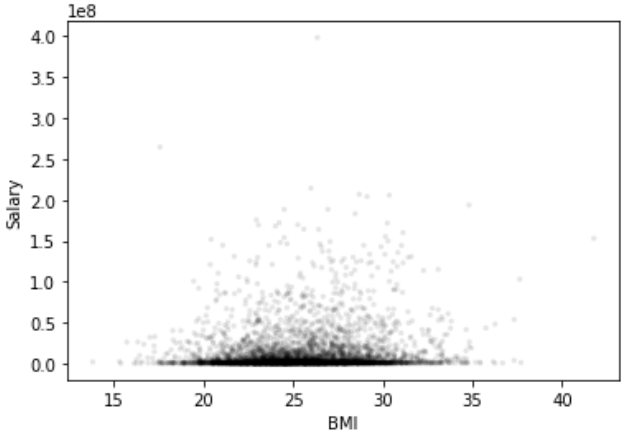


PART 5

I planned to show the correlation between 2 variables which are the BMI values and the salaries of the players and I used the scatter plot in order to visualize this correlation.

```
# Plotting the BMI values with the Salaries as a scatter plot to show the coorolation


# this function will get a random sample for a given dataframe
def SampleRows(df, nrows, replace=False):
    # replace = same row could be chosen more than one or not
    indices = np.random.choice(df.index, nrows, replace=replace)
    print(indices)
    sample = df.loc[indices]
    return sample

gscatter = new_df.groupby('playerID')["salary","BMI","nameFirst","nameLast"].apply(lambda x: x)#.sort_values(by='salary', ascendi
dff = gscatter.groupby(["nameFirst","nameLast","BMI"],as_index=False).sum()
# dff
# the previous data frame returns all the players with their first and last names and their respective BMI values and salaries.

sample = SampleRows(dff, 5000)

bmi, salary = sample.BMI, sample.salary

# sample


thinkplot.Scatter(bmi, salary, alpha=1.0,color='red')
thinkplot.Config(xlabel='BMI',
                 ylabel='Salary'
                )
```





PART 6

Here I tested my hypothesis that "the players with BMI above 26 have higher salary than others"

Using the hypothesis testing methods provided by thinkstats2.

My test statistic was taking the average of the 2 groups (salaries with BMI above 26 and below 26) and taking the absolute difference

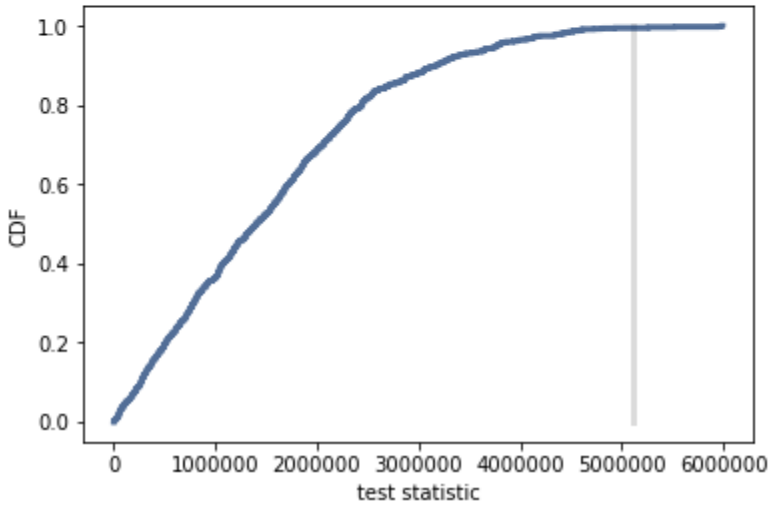
```
def TestStatistic(self, data):
    group1, group2 = data
    test_stat = abs(group1.mean() - group2.mean())
    return test_stat
```

The results was as the following:

```
ht = DiffMeansPermute(data)
pvalue = ht.PValue()
pvalue
"""
the p value is too small so the null hypothesis is rejected
"""
```

```
0.005
```

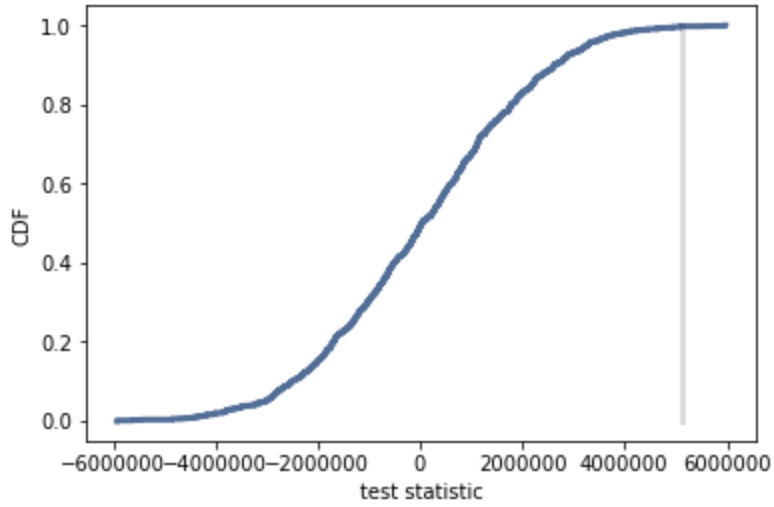
```
ht.PlotCdf()
thinkplot.Config(xlabel='test statistic',
                 ylabel='CDF')
```



```
ht = DiffMeansOneSided(data)
pvalue = ht.PValue()
pvalue

"""
also the hypothesis is not significantly effective
so both cases are unrelated
"""
```

```
0.002
```

```
ht.PlotCdf()
thinkplot.Config(xlabel='test statistic',
                 ylabel='CDF')
```



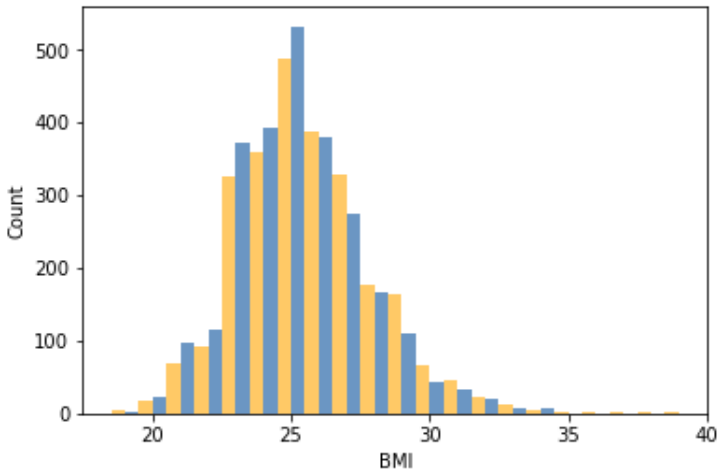
PART 7

My final conclusion was:

There is no real relation or cause and effect in my proposed hypothesis, so I couldn't prove that players with BMI higher than 26 will have higher salary.

And I've done the following to show that:

```
"""
Testing the corrolation
I will affirm that the hypothesis that I made is not true:
I will sort all the salaries by ascending order and I will cut them in halves
(the x highest and x lowest) and I will look at the BMI values for the hieghest salaries
and take plot its histogram, also I will look at the BMI values for the lowest salaries and plot its histogram also,
to see IF THE HIGHEST SALARIES HAS A SPECIAL DESTIRBUTION FOR THE BMI VALUES.
"""

highestSalaries = dff.sort_values(['salary'], ascending=False).reset_index(drop=True).loc[:2569]
lowestSalaries = dff.sort_values(['salary'], ascending=False).reset_index(drop=True).loc[2570:]


thinkplot.PrePlot(2)
hist1 = thinkstats2.Hist(highestSalaries.BMI, label='Highest BMI')
hist2 = thinkstats2.Hist(lowestSalaries.BMI, label='Lowest BMI')
thinkplot.Hist(hist1, width = 0.50,align='right',color='orange')
thinkplot.Hist(hist2, width = 0.50,align='left')
thinkplot.Config(xlabel='BMI', ylabel='Count')
```




```
"""
As we can see from the previous plot that there is no significant difference
between the bmi values for the highest and the lowest salaries.

so there is no relation between bmi and salary that may cause the salary to increase.
"""
```