CS 240 PROJECT

Elif Nur Kalkan/216291637

After brainstorming, I concentrated on 3 questions:

1. Is there a relationship between the number of prizes won by the number of years played by the player?
2. Is there a relationship between the duration of the player's stay in the play and the points he has received?
3. Is there a relationship between the player's age and the score?

I will analyse the second question. I opened my csv file which is “basketball_players_allstar” in Jupyter. Firstly I import the necessary libraries then open my data. Then I specified data which I am gonna work. I did this with .loc command. So, I can present which columns I want.

```
In [4]: myspecificdata = mydata.loc[:,['minutes', 'points']]
        myspecificdata
```

Out[4]:

	minutes	points
0	28	11.0
1	18	10.0
2	13	4.0
3	14	10.0
4	27	10.0
5	32	21.0
6	23	11.0
7	37	25.0
8	32	20.0
9	23	15.0
10	30	17.0
11	23	21.0
12	36	22.0

```
In [7]: from __future__ import print_function, division
        %matplotlib inline
        import first
        import pandas as pd
        import numpy as np
        import thinkstats2
        import thinkplot
```

```
In [3]: mydata=pd.read_csv("basketball_player_allstar.csv")
        mydata
```

Out[3]:

	player_id	last_name	first_name	season_id	conference	league_id	games_played	minutes	points	o_rebounds	...
0	abdulka01	Abdul-Jabbar	Kareem	1978	West	NBA	1	28	11.0	NaN	...
1	abdulka01	Abdul-Jabbar	Kareem	1969	East	NBA	1	18	10.0	NaN	...
2	abdulka01	Abdul-Jabbar	Kareem	1988	West	NBA	1	13	4.0	NaN	...
3	abdulka01	Abdul-Jabbar	Kareem	1987	West	NBA	1	14	10.0	NaN	...
4	abdulka01	Abdul-Jabbar	Kareem	1986	West	NBA	1	27	10.0	NaN	...
5	abdulka01	Abdul-Jabbar	Kareem	1985	West	NBA	1	32	21.0	NaN	...
6	abdulka01	Abdul-Jabbar	Kareem	1984	West	NBA	1	23	11.0	NaN	...
7	abdulka01	Abdul-	Kareem	1983	West	NBA	1	27	25.0	NaN	...

Then, i used describe command to see their statistics like mean, standard deviation(std) etc. To see how much similarity between them. I calculated some statistics to see the relationship between the duration of the player's stay and the points he has received. We can see the maximum points 42.00 taking in 99 minutes.

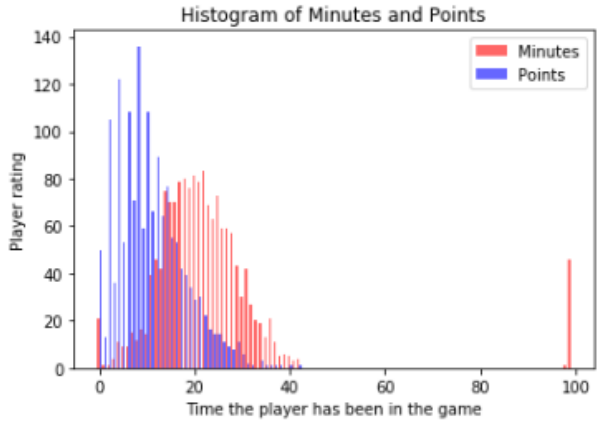
```
In [5]: myspecificdata.describe()
```

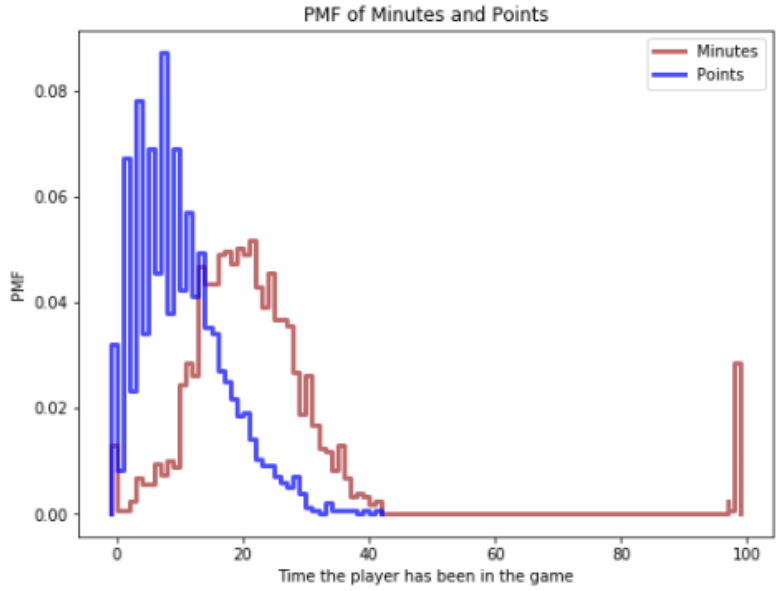
Out[5]:

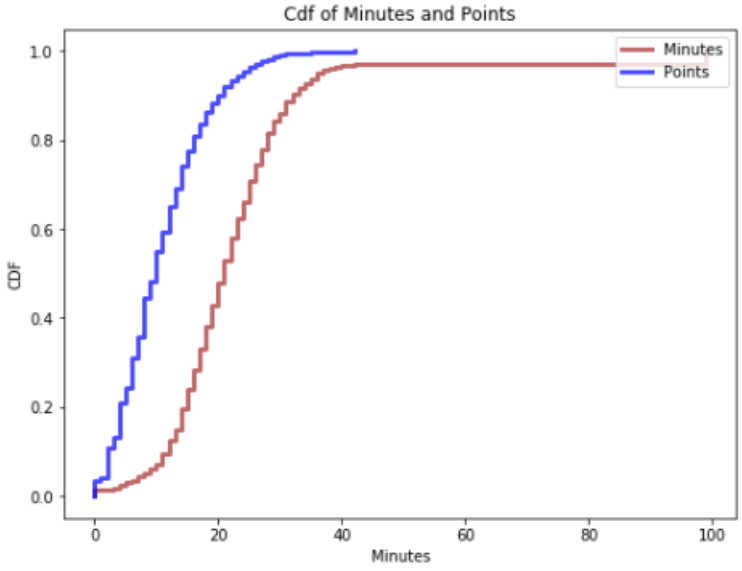
	minutes	points
count	1609.000000	1562.000000
mean	23.103170	10.749680
std	15.221598	6.962659
min	0.000000	0.000000
25%	16.000000	6.000000
50%	21.000000	10.000000
75%	27.000000	15.000000
max	99.000000	42.000000

I used these codes to create histogram , CDF and PMF:

```
In [32]: histogram_of_minutes  = thinkstats2.Hist(myspecificdata.minutes,label="Minutes")
histogram_of_points = thinkstats2.Hist(myspecificdata.points,label="Points")

thinkplot.Hist(histogram_of_minutes, color='Red', width=0.45, align='right')
thinkplot.Hist(histogram_of_points, color='Blue', width=0.45, align='left')
thinkplot.Show(xlabel='Time the player has been in the game', ylabel='Player rating' ,loc='upper right'

histogram_of_minutes1 = thinkstats2.Pmf(myspecificdata.minutes,label="Minutes")
histogram_of_points1 = thinkstats2.Pmf(myspecificdata.points,label="Points")
thinkplot.PrePlot(2)

thinkplot.Pmf(histogram_of_minutes1, color='Brown', align='right')
thinkplot.Pmf(histogram_of_points1, color='Blue', align='right')
thinkplot.Show(xlabel='Time the player has been in the game', ylabel='PMF',loc='upper right',title='PMF

histogram_of_minutes2 = thinkstats2.Cdf(myspecificdata.minutes)
histogram_of_points2 = thinkstats2.Cdf(myspecificdata.points)

thinkplot.Cdf(histogram_of_minutes2, color='Brown',label="ER")
thinkplot.Cdf(histogram_of_points2, color='Blue',label="W")
thinkplot.Show(xlabel='Minutes', ylabel='CDF',loc='upper right',title='Cdf of Minutes and Points')
```


Histogram of Minutes and Points
Minutes
Points
Player rating
Time the player has been in the game


PMF of Minutes and Points
Minutes
Points
PMF
Time the player has been in the game


Cdf of Minutes and Points
Minutes
Points
CDF
Minutes

I made a histogram for the time the player has been in the game. From the histogram chart we can understand the the most time the player has been between 16 and 22 minutes.

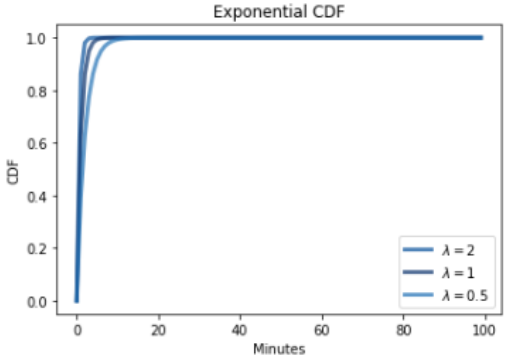
Then I made a probabilty mass function for the time the player has been in the game. Probability

mass function tell us about the time the player has been in the game. As we see the in histogram the most time that player in the game 16 and 22. From PMF, we can see that highest probabilty of the time that player in the game is between 16 and 22 is nearly 0,5.

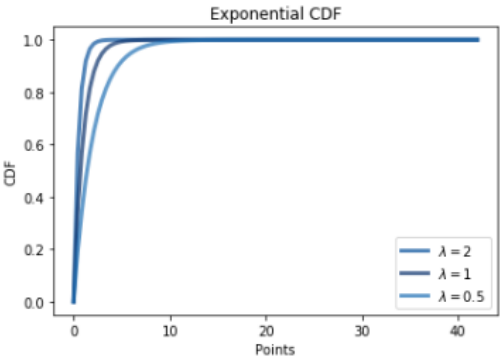
After that I made a cumulative distribution function whic is starting from 0 and goes to 1 .

After i show these distributions i picked exponential distribution to model my data . Here's what the exponential CDF looks like with a range of parameters.

In [35]:
```
thinkplot.PrePlot(1)
for lam in [2.0,1,0.5]:
    xs, ps = thinkstats2.RenderExpoCdf(lam,myspecificdata.minutes.min(), myspecificdata.minutes.max())
    label = r'$\lambda=%g$' % lam
    thinkplot.Plot(xs, ps, label=label)

thinkplot.Config(title='Exponential CDF', xlabel='Minutes', ylabel='CDF',
                 loc='lower right')
```



In [36]:
```
thinkplot.PrePlot(1)
for lam in [2.0,1,0.5]:
    xs, ps = thinkstats2.RenderExpoCdf(lam,myspecificdata.points.min(), myspecificdata.points.max())
    label = r'$\lambda=%g$' % lam
    thinkplot.Plot(xs, ps, label=label)

thinkplot.Config(title='Exponential CDF', xlabel='Points', ylabel='CDF',
                 loc='lower right')
```



After i represent functions of my data, i calculated correlation between these two variables to observe how much they have relation or do they have relation. Correlation takes values between 1 and -1. If value get closer 1, it means there is positive correlation. If value of correlation get -1, it means there is still correlation but it is negative.

In [8]:
```
def Cov(xs, ys, meanx=None, meany=None):
    xs = np.asarray(xs)
    ys = np.asarray(ys)

    if meanx is None:
        meanx = np.mean(xs)
    if meany is None:
        meany = np.mean(ys)

    cov = np.dot(xs-meanx, ys-meany) / len(xs)
    return cov
```

In [9]:
```
def Corr(xs, ys):
    xs = np.asarray(xs)
    ys = np.asarray(ys)

    meanx, varx = thinkstats2.MeanVar(xs)
    meany, vary = thinkstats2.MeanVar(ys)

    corr = Cov(xs, ys, meanx, meany) / np.sqrt(varx * vary)
    return corr
```

In [10]:
```
Corr(myspecificdata.minutes, myspecificdata.points)
```

Out[10]:
```
nan
```

In [11]:
```
class HypothesisTest(object):

    def __init__(self, data):
        self.data = data
        self.MakeModel()
        self.actual = self.TestStatistic(data)

    def PValue(self, iters=2000):
        self.test_stats = [self.TestStatistic(self.RunModel())
                           for _ in range(iters)]

        count = sum(1 for x in self.test_stats if x >= self.actual)
        return count / iters

    def TestStatistic(self, data):
        raise UnimplementedMethodException()

    def MakeModel(self):
        pass

    def RunModel(self):
        raise UnimplementedMethodException()
```

In [12]:
```
class DiffMeansPermute(thinkstats2.HypothesisTest):

    def TestStatistic(self, data):
        myspecificdata.R, myspecificdata.H = data
        test_stat = abs(myspecificdata.R.std() - myspecificdata.H.std())
        return test_stat

    def MakeModel(self):
        myspecificdata.R, myspecificdata.H = self.data
        self.n, self.m = len(myspecificdata.R), len(myspecificdata.H)
        self.pool = np.hstack((myspecificdata.R, myspecificdata.H))

    def RunModel(self):
        np.random.shuffle(self.pool)
        data = self.pool[:self.n], self.pool[self.n:]
        return data
```

In [13]:
```
data = myspecificdata.minutes, myspecificdata.points
ht = DiffMeansPermute(data)
pvalue = ht.PValue()
pvalue
```

```
C:\ProgramData\Anaconda2\lib\site-packages\ipykernel_launcher.py:9: UserWarning: Pandas doesn't allow columns to be c
reated via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access
  if __name__ == '__main__':
```

Out[13]:
```
0.0
```

In conclusion, I used difference means permute function but i replaced means with standard deviation in test statistic part. Then, i got pvalue as a 0. We can understand from here there is no relationship between the duration of the player's stay in the play and the points he has received