Muhammed Fatih KARAKELLE

214101829

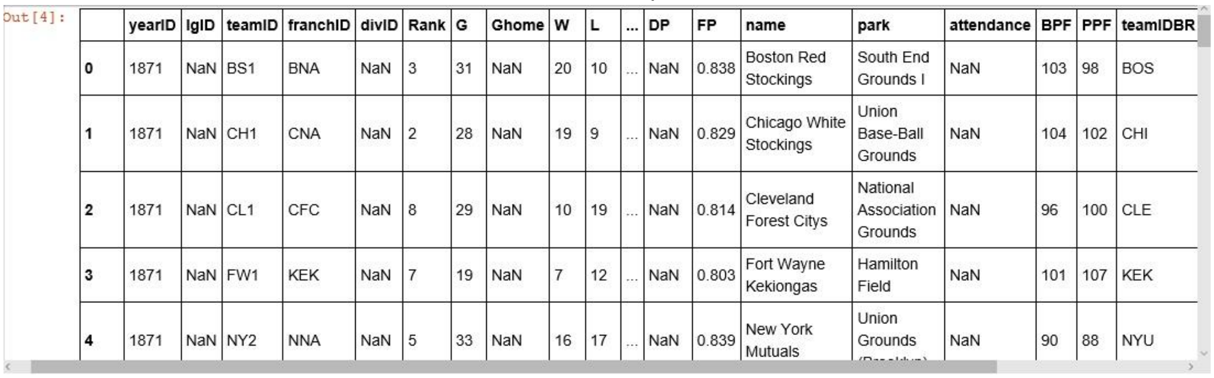
CS 240 - ASSIGNMENT

After brainstorming, I found 3 questions:

1) Is there a relationship between the positions in final standings and wins after 2000?
2) Is there a relationship between positions in final standings and lose?
3) Is players' salaries increase as time goes on?

I tried to analyse the first question. I opened my csv file which is 'Teams.csv' in Jupiter. In the csv file the values are between 1871 and 2016. However, in this time period the rules of the game have changed. For example, the number of the games that teams play have changed. We can see that the maximum number of game that a team played is 33 in 1871 but in 2016 this number is 162. That's why I specified my time period between 2000 and 2016. First, I make a histogram of the number of victories of the teams. Then, I made Pmf and Cdf tables of the wins. To be able to see the relationship between position in final standings and wins after 2000 I wrote a correlation function. I was expecting negative correlation because while the number of wins increase, the number in the league decrease. In the league table, the ordering starts from 1 but the team in the first position of the league has more victory than the team at the last position of the league. After find the correlation coefficient I made a scatter plot to see the relationship between two variables clearly. To make the chart clearer I wrote jittering function and I added random noise to the chart. After jittering my chart, I examine the shape of the distribution. After that I made hypothesis testing to see if it happened by chance or the effect of it is real.

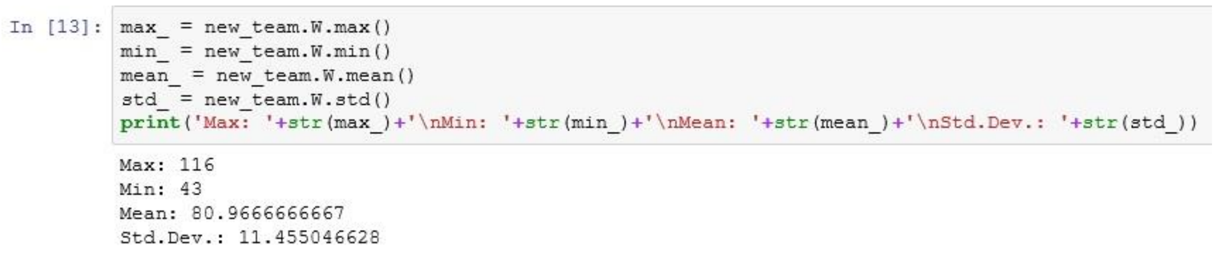
1) First I opened my ‘Teams.csv’ file in Jupiter notebook. The table that I will work has 48 columns and 2835 rows. However, I will only care the parts happened after 2000. The number of rows became 480 after I specialize the data.



Out[4]:

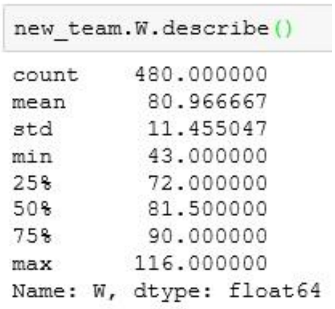
	yearID	lgID	teamID	franchID	divID	Rank	G	Ghome	W	L	...	DP	FP	name	park	attendance	BPF	PPF	teamIDBR
0	1871	NaN	BS1	BNA	NaN	3	31	NaN	20	10	...	NaN	0.838	Boston Red Stockings	South End Grounds I	NaN	103	98	BOS
1	1871	NaN	CH1	CNA	NaN	2	28	NaN	19	9	...	NaN	0.829	Chicago White Stockings	Union Base-Ball Grounds	NaN	104	102	CHI
2	1871	NaN	CL1	CFC	NaN	8	29	NaN	10	19	...	NaN	0.814	Cleveland Forest Citys	National Association Grounds	NaN	96	100	CLE
3	1871	NaN	FW1	KEK	NaN	7	19	NaN	7	12	...	NaN	0.803	Fort Wayne Kekiongas	Hamilton Field	NaN	101	107	KEK
4	1871	NaN	NY2	NNA	NaN	5	33	NaN	16	17	...	NaN	0.839	New York Mutuals	Union Grounds	NaN	90	88	NYU

2) I will use Rank which is the position in final standings and W which is the number of victory columns in the table.



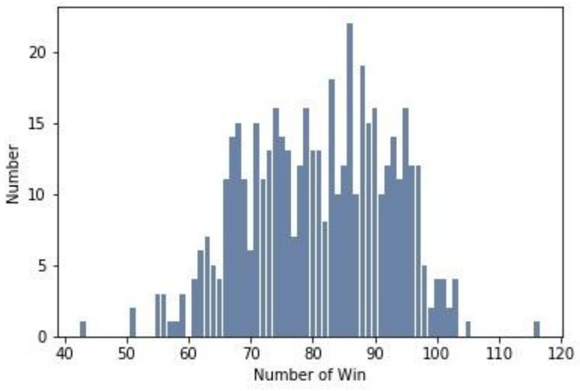
```
In [13]: max_  = new_team.W.max()
         min_  = new_team.W.min()
         mean_  = new_team.W.mean()
         std_  = new_team.W.std()
         print('Max: '+str(max_)+'\nMin: '+str(min_)+'\nMean: '+str(mean_)+'\nStd.Dev.: '+str(std_))

         Max: 116
         Min: 43
         Mean: 80.9666666667
         Std.Dev.: 11.455046628
```

3) I calculated some statistics about the win column. The maximum number for the winning games is 116. The minimum number is 43. The mean of that column is 80.97 which means the average of the winning numbers. Standard deviation is 11.46.

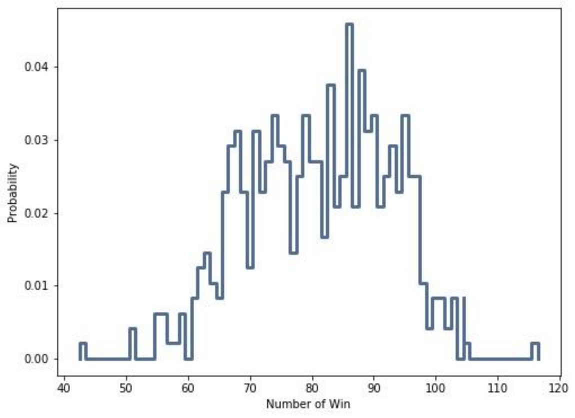


```
new_team.W.describe()

count     480.000000
mean       80.966667
std        11.455047
min        43.000000
25%        72.000000
50%        81.500000
75%        90.000000
max       116.000000
Name: W, dtype: float64
```

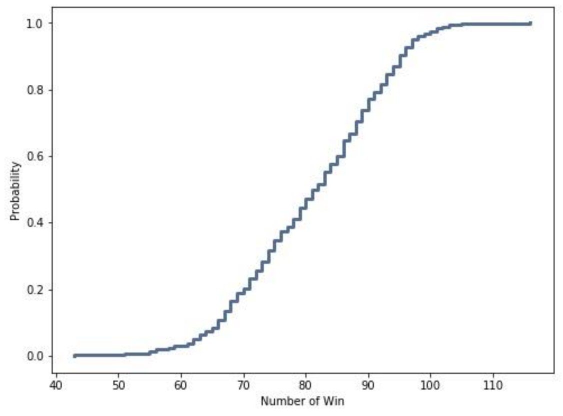
4) I made a histogram for the number of wins. From the histogram chart we can understand that the most number of game winning is between 80 and 90.



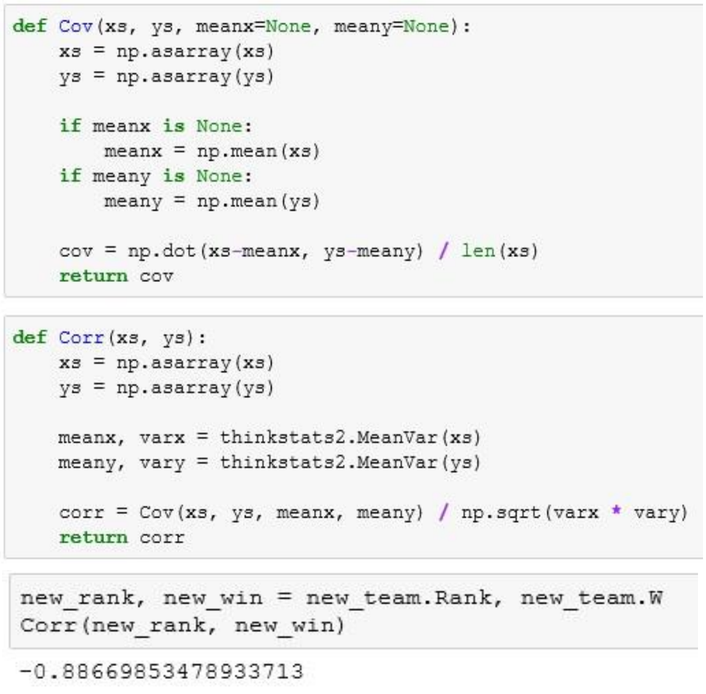
5) Then I made a probability mass function plot for the number of wins. Probability mass function tell us about the probability of the number of wins happen. As we see in the histogram the most happening game winning number is between 80 and 90. From pmf, we can see that the highest probability of the game winning number is again between 80 and 90 which is higher than 0.04.



6) After that I made a cumulative distribution function plot which is starting from 0 and goes to 1 for the number of wins.



7) To be able to see the relationship between the number of the win and rank I used a correlation function. The correlation between winning games and rank is -0.887. Now, what we will understand from this result? As I mentioned above, I expected a negative correlation because we expect that when number of winning game increase the position in the table is decrease. We need to be careful about one key thing. The position is increase but the number of position is decrease. As we know, positions start from 1. That's why position 1 refers the most successful team. This correlation shows us when the number of the



```
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
```

```
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
```

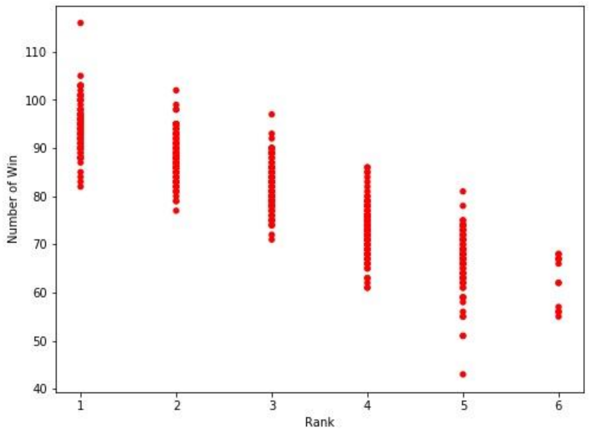
```
new_rank, new_win = new_team.Rank, new_team.W
Corr(new_rank, new_win)
```

```
-0.88669853478933713
```

game winning increase the number of position decrease. So that there is an opposite relationship between rank and win. In the code part we are using covariance to calculate the correlation. We subtract the means from the values and divide in to length of the values. After we found that we divide this to the square root of the multiplication of the variances. This gave us the correlation coefficient.

```
thinkplot.Scatter(new_rank, new_win, alpha=1, color='Red')
thinkplot.Show(xlabel='Rank', ylabel='Number of Win', legend=False)
```

8) To see this relationship better I visualized the correlation with scatter plot. We are using scatter plot to see the relationship between two variables. As we can understand from the table the shape of the distribution in scatter plot is like decreasing. I was expecting a distribution like that because our correlation coefficient was negative.

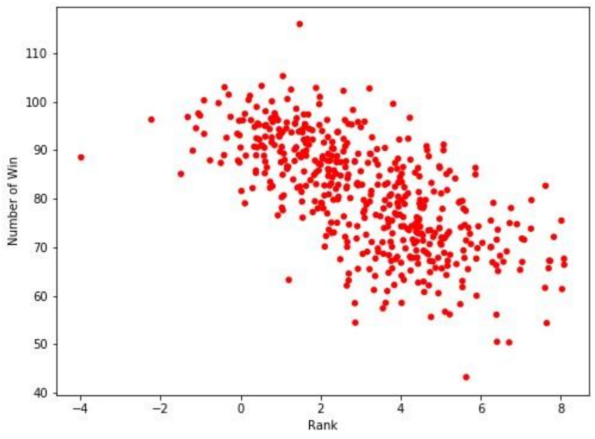


```
def Jitter(values, jitter=0.5):
    n = len(values)
    return np.random.normal(0, jitter, n) + values
```

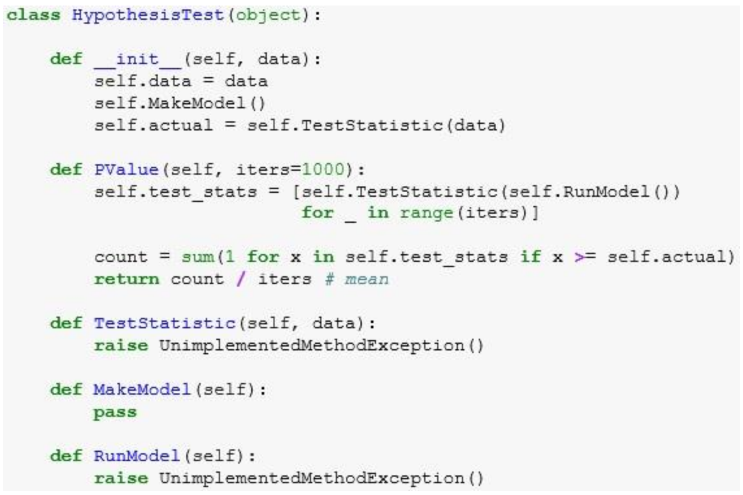
```
new_rank_J = Jitter(new_rank, 1.4)
new_win_J = Jitter(new_win, 0.5)
```

```
thinkplot.Scatter(new_rank_J, new_win_J, alpha=1, color='Red')
thinkplot.Show(xlabel='Rank', ylabel='Number of Win', legend=False)
```

9) To be able to see and interpret better I used jittering. I add random noise to the distribution. After jittering the shape of the distribution became clear. Even we don't know the correlation coefficient we can understand from this chart the correlation is negative.

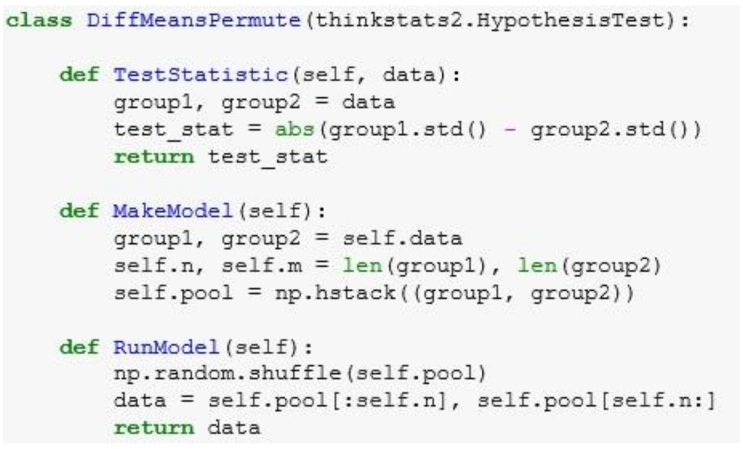


10) The last step is hypothesis testing. I tried to answer the question that if this effects happened by chance or not. It makes our hypothesis more accurate. In the code part we have Pvalue. We will look to Pvalue and try to understand if we should reject our null hypothesis or not.



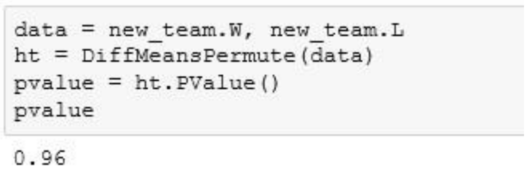
```python
class HypothesisTest(object):

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

    def PValue(self, iters=1000):
        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 # mean

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

    def MakeModel(self):
        pass

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

Generally, we have 0.05 or 5% threshold. If our Pvalue smaller than threshold then it is statistically significant and we can reject our null hypothesis. After I wrote HypothesisTest class I wrote another class that has a name DiffMeansPermute. This class has heritage from the HypothesisTest class. I used this class to see calculate the Pvalue. In



```python
class DiffMeansPermute(thinkstats2.HypothesisTest):

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

    def MakeModel(self):
        group1, group2 = self.data
        self.n, self.m = len(group1), len(group2)
        self.pool = np.hstack((group1, group2))

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

test statistic part I used the absolute differences between the standard deviation of the groups. In the code absolute is used that's why it will be one sided test. In the make model part sizes of the groups are recorded which are m and n. Then, it combines the groups and assign them



```python
data = new_team.W, new_team.L
ht = DiffMeansPermute(data)
pvalue = ht.PValue()
pvalue
```

```
0.96
```

into pool. Run model part simulates the null hypothesis. I found my Pvalue by using these functions. My Pvalue was 0.96. This result is bigger than our threshold which is 0.05. That's why it is not statistically significant. So that we can reject the null hypothesis.

11) As a conclusion, in statistic there are some useful methods that can be used to see the relationship between variables. First I made charts (histogram, pmf, cdf) to see my data. After that I visualized my data with time in scatter plot. It showed the relationship between variables. After that, I calculated correlation which was -0.887 which means the variables inversely correlated. My hypothesis was there is a relationship between the numbers of wins that team's get and their position in the league. My null hypothesis was there is no relationship between them. Then, to see if it happened by chance or not I made hypothesis testing and I found the Pvalue. What I understood from these calculations and from these results, if a team want to be in the higher position in the league they need to get more wins.