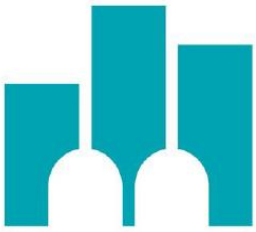


İSTANBUL ŞEHİR ÜNİVERSİTESİ

CS 240

EXPLORATORY DATA ANALISYS

BASKETBALL DATA ANALYSIS PROJECT

Abstract

In the following document I'm doing a data analysis among a data of basketball players. The aim that willing to find is whether there is a relationship between two variables of the data which are the number of Defense Rebounds and assists gained by each player

Mamoun Tawakol
215832785

SECTION 1

From the data available under the basketball_players excel file its clear that after giving the identification of the player id, year, stint, team id and league id followed by many distinct posts in which each particular player credited a defense rebound on the baseball field. Therefore, the performance metric that we are most interested in is the number of defense rebounds for each player. Also in the total number of points in a game of each player on the court.

- What is the relationship between different performance metrics?
- Could it be any strong negative or positive relationship between variables?
- What are the characteristics of variables in datasets?

Givin the three previous questions, I am going to attempt to scrutinize into the relationship between the number defenses rebounds and the number of assists through each game played. The two columns 'dRebounds' and ‘assists’.

Hypothsis: If the number of defense rebounds are high then number of assists are high as well for each particular player.

This means that in other words if a player got many drebounds on a game that person would make many assists too.

Null hypothesis: If a player does not get many defense rebounds in a game is it the case that he does not get many assists as well.

Hence, we will attempt to see if there is a significant statistical relationship between these two variables, which would display if a player is a good at getting drebounds the very player is good at getting many assists as well.

A game of basketball is played between two teams, each composed of five players. The objective is to shoot a basketball (approximately 9.4 inches (24 cm) in diameter) through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high that is mounted to a backboard at each end of the court. Rebounds are divided into two main categories: "offensive rebounds", which the ball is recovered by the offensive side and does not change possession, and "defensive rebounds", in which the defending team gains possession. However, the assists is the number of threw balls to the player who scored a goal. A player gain credit for assists and defense rebounds if he applied one of them in the game.

SECTION 2

	playerID	year	stint	tmID	lgID	GP	GS	minutes	points	oRebounds	...	PostBlocks	PostTurnovers	PostPF	PostfgAttempted	PostfgMade	Po
0	abramjo01	1946	1	PIT	NBA	47	0	0	527	0	...	0	0	0	0	0	
1	aubucch01	1946	1	DTF	NBA	30	0	0	65	0	...	0	0	0	0	0	
2	bakerno01	1946	1	CHS	NBA	4	0	0	0	0	...	0	0	0	0	0	
3	baltihe01	1946	1	STB	NBA	58	0	0	138	0	...	0	0	3	10	2	
4	barrjo01	1946	1	STB	NBA	58	0	0	295	0	...	0	0	0	0	0	
5	baumhfr01	1946	1	CLR	NBA	45	0	0	631	0	...	0	0	0	0	0	
6	beckemo01	1946	1	PIT	NBA	17	0	0	108	0	...	0	0	0	0	0	
7	beckemo01	1946	2	BOS	NBA	6	0	0	13	0	...	0	0	0	0	0	
8	beckemo01	1946	3	DTF	NBA	20	0	0	41	0	...	0	0	0	0	0	
9	beendha01	1946	1	PRO	NBA	58	0	0	713	0	...	0	0	0	0	0	
10	biasaha01	1946	1	TRH	NBA	6	0	0	6	0	...	0	0	0	0	0	

```
data = pd.read_csv("basketball_players.csv", low_memory=False)
print data["dRebounds"]
print data["assists"]
```

```
23733      5
23734     21
23735     26
23736     55
23737     85
23738     16
23739      6
23740      0
23741     27
23742     17
23743      5
23744     22
23745     44
23746     75
23747     21
23748    134
23749      3
23750     32
Name: assists, Length: 23751, dtype: int64
```

```
data = pd.read_csv("basketball_players.csv", low_memory=False)
print data["dRebounds"]
print data["assists"]
```

```
23732    0
23733    0
23734    0
23735    0
23736    0
23737    0
23738    0
23739    0
23740    0
23741    0
23742    0
23743    0
23744    0
23745    0
23746    0
23747    0
23748    0
23749    0
23750    0
Name: dRebounds, Length: 23751, dtype: int64
```

Here the two columns of interest ‘dRebounds’ and ‘assists’ are being read from the csv file and consequently printed as shown above. Since, we have got pretty huge data sets if there is any relation or behavior of the data at certain regions that is related between the columns shall be examined elaborately.

```
dRebounds = data.dRebounds.dropna()
assists = data.assists.dropna()
```

```
dRebounds.values
```

```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

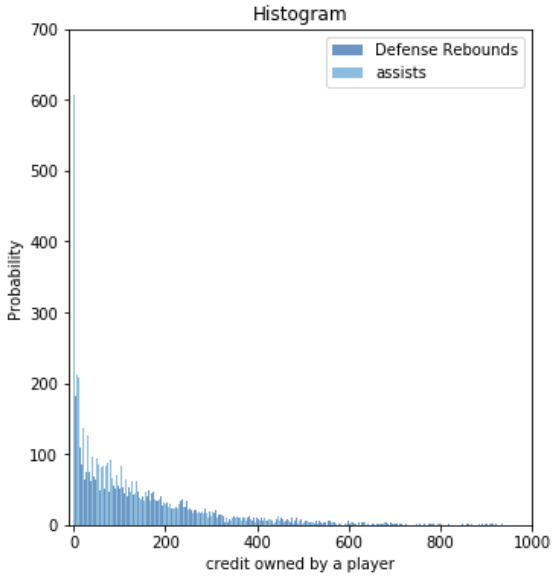
```
assists.values
```

```
array([ 35,  20,   0, ..., 134,   3,  32], dtype=int64)
```

Cleaning the data dropping none values in the columns. Moreover, these are the numpy arrays of the data used for the calculation of the p-value and other statistics.

SECTION 3

```
# Histogrma
hist_dReounds = thinkstats2.Hist(dRebounds, label = "Defense Rebounds")
hist_assists = thinkstats2.Hist(assists, label = "assists")
thinkplot.preplot(2, cols=2)
width = 0.5
thinkplot.Hist(hist_dReounds, align = "right", width = 0.5)
thinkplot.Hist(hist_assists, align = "left", width = 0.5)
thinkplot.show(title='Histogram', xlabel = "credit owned by a player", ylabel = "Probability", axis = [-10,1000, 0, 700])
```

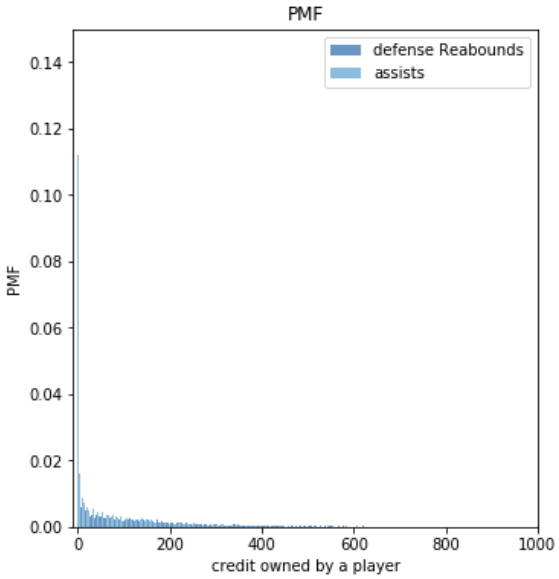


The first graph

The first graph to be plot is a Histogram of the data in interest. First the Hist module of the thinkstats2 is used then PlePlot of two for the two columns, then it's plotted and configured.

The histogram as a plot lets us to discover, and show, the underlying frequency distribution (shape) of the set close in magnitude two data groups' data. This allows the inspection of the data for its underlying distribution. So the distribution of the data in the histogram is defined by the frequency.

```
# PMF
pmf_dRebounds = thinkstats2.Pmf(dRebounds, label = "defense Reabounds")
pmf_assists = thinkstats2.Pmf(assists, label = "assists")
thinkplot.preplot(2, cols=2)
thinkplot.Hist(pmf_dRebounds, align = "right", width= width)
thinkplot.Hist(pmf_assists, align = "left", width= width)
thinkplot.show(title='PMF', xlabel = "credit owned by a player", ylabel = "PMF", axis = [-10,1000, 0, 0.15])
```

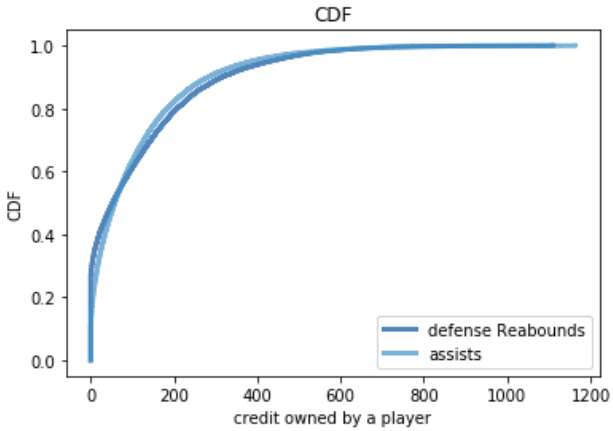


The second graph (CDF)

The second graph to be plot is the Pmf which follows exactly the same procedure, however applying the Pmf module.

The Probability Mass Function provides us with the probabilities for the discrete random variables. "Random variables" are variables from experiments or in our case from the data set.

```
# CDF
cdf_dRebounds = thinkstats2.Cdf(dRebounds, label = "defense Reabounds")
cdf_assists = thinkstats2.Cdf(assists, label = "assists")
thinkplot.preplot(2)
thinkplot.Cdfs([cdf_dRebounds, cdf_assists])
thinkplot.show(title='CDF', xlabel = "credit owned by a player", ylabel = "CDF")
```



The third graph (CDF)

On the third place is the CDF plotted the same way as the others by just using the CDF module. It does give us a clearer view of the picture just by displaying thee very curvature they form .Hence, we observe how similar the data is and how at a single point the domination of the curves change but till keeps the very similar manner of propagation

Here their relationship is quite precise and tells a lot about the relationship between the two groups.

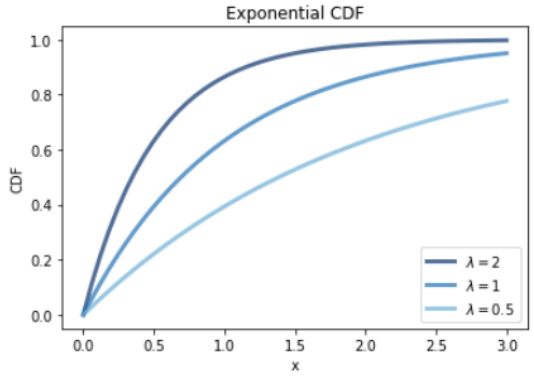
SECTION 4

In the real world, exponential distributions come up when we look at a series of events and measure the times between events, which are called **interarrival times**. If the events are equally likely to occur at any time, the distribution of interarrival times tends to look like an exponential distribution.

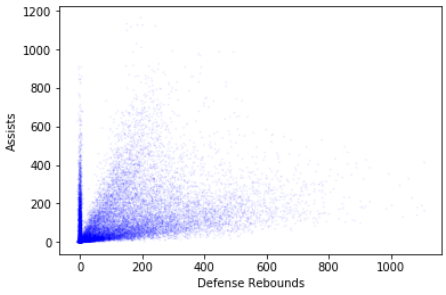
The CDF of the exponential distribution is:

$$CDF(x) = 1 - e^{-\lambda x}$$

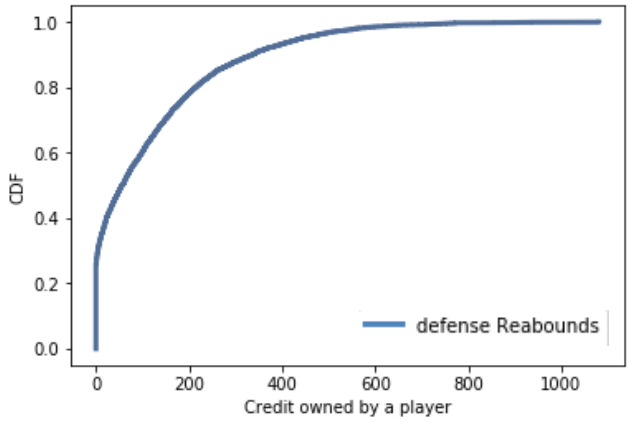
The parameter, λ, determines the shape of the distribution.



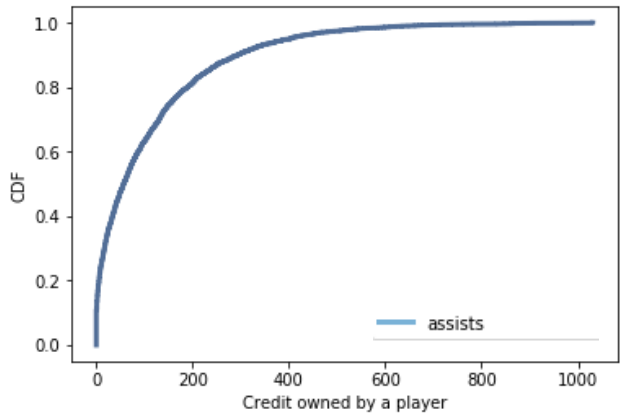

```
jet_dRebounds = Jitter(dRebounds, 2.8)
jet_assists = Jitter(assists, 1.0)

thinkplot.Scatter(jet_dRebounds, jet_assists, alpha=0.1, s=2)
thinkplot.Config(xlabel='Defense Rebounds',
                 ylabel='Assists',
                 legend=False)
```



```
dRebounds_sample = np.random.choice(dRebounds, 5000, replace=True)
cdf_sample_dR = thinkstats2.Cdf(dRebounds_sample, label = 'dRebounds')
thinkplot.Cdf(cdf_sample_dR)
thinkplot.Config(xlabel='Credit owned by a player', ylabel='CDF', loc = 'lower right')
```



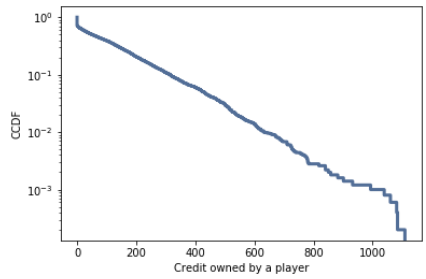

```
assists_sample = np.random.choice(assists, 5000, replace=True)
cdf_sample_as = thinkstats2.Cdf(assists_sample, label = 'Assists')
thinkplot.Cdf(cdf_sample_as)
thinkplot.Config(xlabel='Credit owned by a player', ylabel='CDF', loc = 'lower right')
```



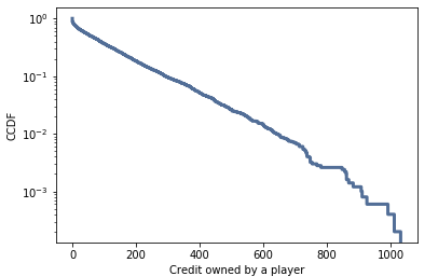
As we can see the shape of the distribution gets closest to the modelling distribution of the exponential distribution function whose λ = 2. The exponential distribution seems to be the only one to encompass the data sets from the two different columns. From the individual CDF again is easy to see the closeness to their best fit modeling exponential function.

Also, as visible from the jitter plot the vast majority of the data pairs of the defense rebound and assists performance seems to be very even hence partially support the initial proposition of the hypothesis.

```
thinkplot.Cdf(cdf_sample_dR, complement=True)
thinkplot.Config(xlabel='Credit owned by a player',
                 ylabel='CCDF', yscale='log', loc='upper right')
```




```
thinkplot.Cdf(cdf_sample_as, complement=True)
thinkplot.Config(xlabel='Credit owned by a player',
                 ylabel='CCDF', yscale='log', loc='upper right')
```



These are the CCDF of the inter arrivals on a log-*y* scale. It is not exactly straight, which suggests that the exponential distribution is only an approximation to the real one.

SECTION 5

Covariance is useful for some calculations, but it doesn't mean much by itself. The coefficient of correlation is a standardized version of covariance that is easier to interpret.

```
Cov(assists, dRebounds)
```

```
7741.414195699519
```

The Covariance shows how strongly correlated two variables are.

```
np.corrcoef(assists, dRebounds)
```

```
array([[ 1.        ,  0.38601049],
       [ 0.38601049,  1.        ]])
```

Pearson's correlation is not robust in the presence of outliers, and it tends to underestimate the strength of non-linear relationships. The correlation is a single number that describes the degree of relationship between two variables.

```
Corr(assists, dRebounds)
```

```
0.38601048930879406
```

Here, the correlation coefficient shows that correlation is very low which means that the two groups are not very similar which in contrast don’t support the hypothesis.

Spearman's correlation is more robust, and it can handle non-linear relationships as long as they are monotonic.

```
scipy.stats.spearmanr(assists, dRebounds)
```

```
SpearmanrResult(correlation=0.543342652275423, pvalue=0.0)
```

As seen from the scatterplot above the relationship between the two variables don't look very organized so by running a Spearman's correlation the strength and direction of this monotonic relationship is measured and from that we can see the Spearmans correlation is 0.5 which is not very high and not very low but in the middle.

SECTION 6

Hypothsis: If the number of defense rebounds are high then number of assists are high as well for each particular player.

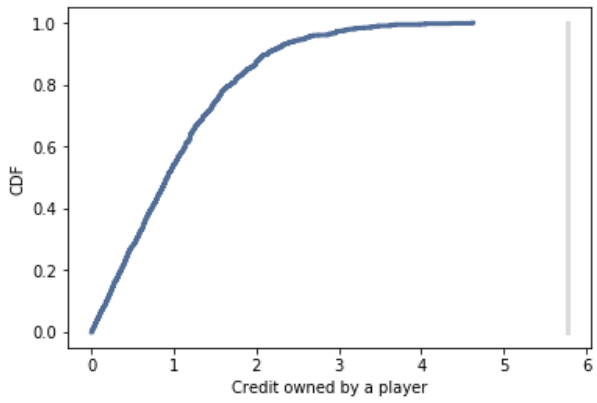
This means that in other words if a player got many drebounds on a game that person would make many assists too.

Null hypothesis: If a player does not get many defense rebounds in a game is it the case that he does not get many assists as well.

```
data = dRebounds, assists
ht = DiffMeansPermute(data)
pValue = ht.PValue()
pValue
```

```
0.0
```

```
ht.PlotCdf()
thinkplot.show(xlabel = "Credit owned by a player", ylabel = "CDF")
```



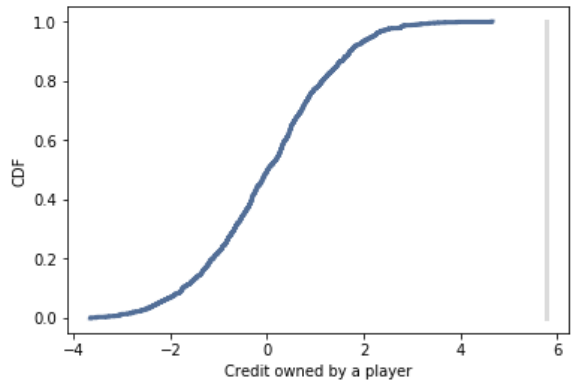
Here, the Test Statistic is difference with the test_stat = abs(group1.mean() - group2.mean()) which the difference between the group's mean values followed by the absolute value taken so that a positive value is obtained at the and because the test statistic will return a single number. The value of the p value is very small approximated to zero which means that the p value is significant and

therefore there is high confidence in discarding the null hypothesis and supporting the proposed hypothesis to be true. Also, from the graph its apparent that the manner of the curviture is exponentially propagating and we observe how far the p value is resulting in a 0.0 value.

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

```
0.0
```

```
ht.PlotCdf()
thinkplot.show(xlabel = "Credit owned by a player", ylabel = "CDF")
```

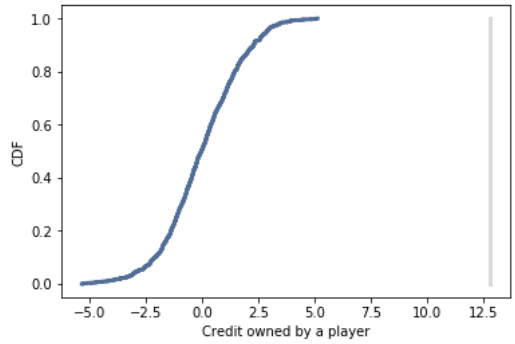


Here, the Test Statistic is DiffMeansOne Sided which uses test_stat = group1.mean() - group2.mean() which takes the mean of the two column of interest dRebounds and assists and subtracts the second from the first one differing with the absence of the absolute value which model the data more as a whole because negative value may affect the final value of the test statistic. If there are great differences between the data by using this test statistic we would be able to detect it to some. The p value is in strong support of the hypothesis by pointing to the null hypothesis as wrong. Negating the null hypothesis means that the p value is statistically significant and there is indeed relationship between the two groups as stated in the hypothesis. In here, the curvature is a bit different in the base from the previous CDF however, the p value is a bit closer but not close enough to two significant figures.

```
ht = DiffStdPermute(data)
pValue = ht.PValue()
pValue
```

```
0.0
```

```
ht.PlotCdf()
thinkplot.show(xlabel = "Credit owned by a player", ylabel = "C
```



In the third try of this experiment another test statistic is implemented using **test_stat = group1.std() - group2.std()** which utilizes standart diviation statistic on the data groups. Standard deviation is a measure of the dispersion of a set of data from its mean. If the data points are further from the mean, there is higher deviation within the data set. Standard deviation is calculated as the square root of variance by determining the variation between each data point relative to the mean.

As obvious from the p value the significance is confirmed once again and that is quite clear from the CDF plot as well where p value is even further confirming the preceding observations.

SECTION 6

Conclusion:

All in all, the aim of the investigation in this report was two prove that there is a significant relation between the two sets of data provided under the naming “dRebounds” and “assists”. From the various kinds of plots an observation was conducted that concludes from the empirical evidence from the proof of the hypothesis analysis that the proposed initially hypothesis is correct. Hence we have found that a basketball player who make defense rebound have a good chance to make an assist. By using different test statistics we were able to show a very strong value for the p-value which consequently directs us to the triteness of the hypothesis. The vacuous distribution plot enabled us to observe the distribution of the two sets of data which additionally helped us imagine how the data is scattered on the plane it occupies. The strong support from all the statistical methods used leads us to conclude that the statistical significance of the observation is present both visually from the alternating kinds of plots and numerically from the statistics applied.