CS 240 Exploratory Data Analysis
Project
Furkan Kara - 213900015

SECTION 1

1. Is there any relationship between coaches losing rate and win rate?
2. Is there any relationship between coaches' mission time and win rate?
3. Are there any strong negative or positive relationship between winning rate and losing rate?
4. Is there any strong negative or positive relationship between coaches win rate and mission time?
5. What are the characteristics of variables in coaches' statistic? What are the average, standard deviation, variance, max, min of these variables? And these values describe what?

1st question is selected. When win rate increase or will be good, coaches losing rate how changing, Also, are there any relationship between losing rate and win rate?

Ho: $\beta1 = \beta2 = 0$ (null hypothesis says that no linear relationship between two variables)

Ha: $\beta1 \neq \beta2 \neq 0$ ($\beta1$: win rate, $\beta2$: losing rate)

This question has been selected because looking relationship of win rate and losing rate are important to analyze coaches' performance. This is the question of description; Is there any linear relationship between win rate and mission time for each different coaches' performance?

SECTION 2

```
In [4]: coaches = pd.read_csv(r"C:\Users\FURKAN\Desktop\NJHH\Week 12\basketball_coaches.csv")
        coaches
```

```
In [17]: won=coaches['won']
         won=won.dropna()
         stint=coaches['stint']
         stint=stint.dropna()
```

Won and lost is important parameters of coaches' performance that will be used. won, describes number of victories. lost, describes number of match loses. .dropna() has been used to removing NaN variables.

SECTION 3

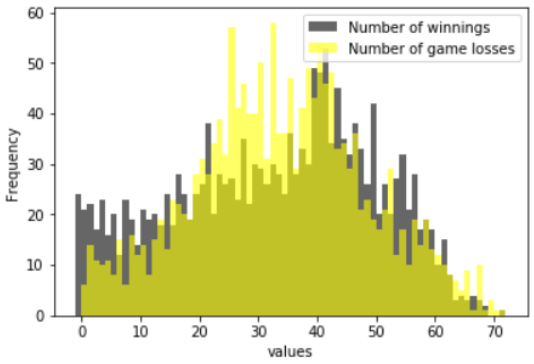
'5 descriptive statistic of won: standart deviation: ' + str(won.std()) + ', mean: ' + str(won.mean()) + ', median: ' +str(won.median())+', min: '+str(won.min())+', max: '+str(won.max())

```
'5 descriptive statistic of won: standart deviation: 17.011147846862432,
mean: 33.29821428571429, median: 36.0, min: 0.0, max: 72.0'
```

```
'5 descriptive statistic of lost: standart deviation: ' + str(lost.std())
+ ', mean: ' + str(lost.mean()) + ', median: ' +str(lost.median())+', min:
'+str(lost.min())+', max: '+str(lost.max())
```

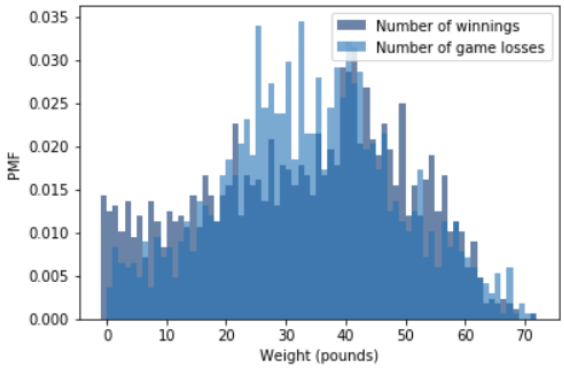
```
'5 descriptive statistic of lost: standart deviation: 14.715967024642866,
mean: 33.286904761904765, median: 33.0, min: 0.0, max: 71.0'
```

```
won_hist = thinkstats2.Hist(won, label='Number of winnings')
lost_hist = thinkstats2.Hist(lost, label='Number of game losses')
width =1
thinkplot.PrePlot(2)
thinkplot.Hist(won_hist , align='right', width=width,color='black')
thinkplot.Hist(lost_hist, align='left', width=width,color='yellow')
thinkplot.Config(xlabel='values', ylabel='Frequency')
```



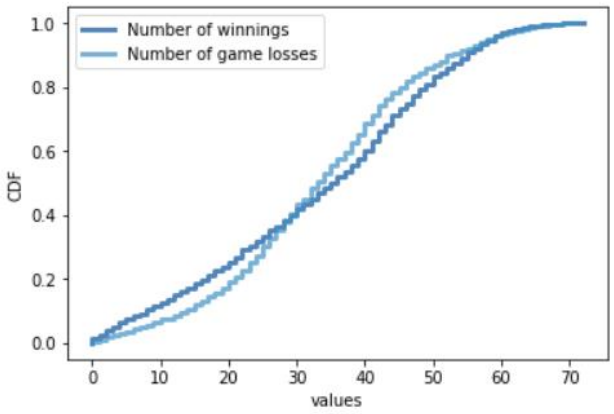
This is the both number of winnings and losses histogram. we can see from these graph, Number of game losses more frequenced values are placed in number of 25,32 and 41. Also, we can say that number of winnings more frequenced values are placed in 40 and nearly 43.

```
won_pmf = thinkstats2.Pmf(won, label='Number of winnings')
lost_pmf = thinkstats2.Pmf(lost, label='Number of game losses')
width =1

# plot PMFs of birth weights for first babies and others
thinkplot.PrePlot(3)
thinkplot.Hist(won_pmf , align='right', width=width)
thinkplot.Hist(lost_pmf, align='left', width=width)
thinkplot.Config(xlabel='Weight (pounds)', ylabel='PMF')
```



We can say that Pmf and histogram are same shape because of probability mass are same as histogram. we can say that both variables are approximately normally distributed.

```
won_cdf = thinkstats2.Cdf(won, label='Number of winnings')
lost_cdf = thinkstats2.Cdf(lost, label='Number of game losses')

thinkplot.PrePlot(2)
thinkplot.Cdfs([won_cdf, lost_cdf])
thinkplot.Config(xlabel='values', ylabel='CDF')
```



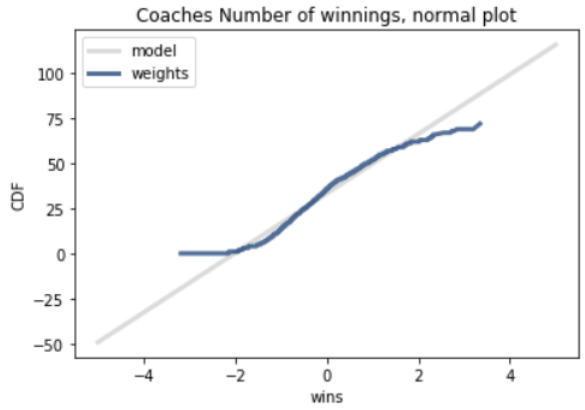
cumulative density function graph describes that both Number of winnings and Number of game losses looks like continuous random variable and they are approximately has same shape.

SECTION 4

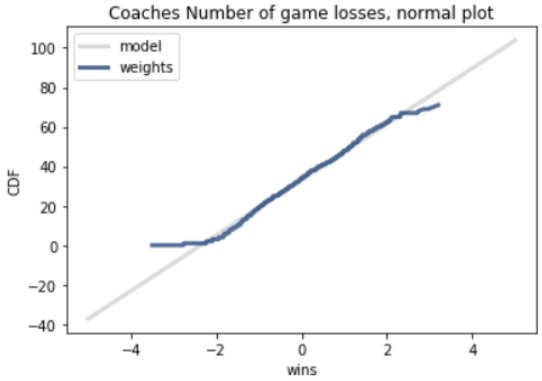
```
def MakeNormalPlot(weights):
    """Generates a normal probability plot of birth weights.

    weights: sequence
    """
    mean, var = thinkstats2.TrimmedMeanVar(weights, p=0.01)
    std = np.sqrt(var)

    xs = [-5, 5]
    xs, ys = thinkstats2.FitLine(xs, mean, std)
    thinkplot.Plot(xs, ys, color='0.8', label='model')

    xs, ys = thinkstats2.NormalProbability(weights)
    thinkplot.Plot(xs, ys, label='weights')
```

```
MakeNormalPlot(won)
thinkplot.Config(title='Coaches Number of winnings, normal plot', xlabel='wins',
                 ylabel='CDF', loc='upper left')
```

In this code mean, variance, standard deviation is used for making normal plots.



it is almost fitted to the line. it means that almost normal plot fitted from the coaches' number of winnings.



It is almost fitted to the line. it means that almost normal plot fitted from the coaches' number of game losses. Actually, number of game losses is better fitted the line than number of winnings

SECTION 5

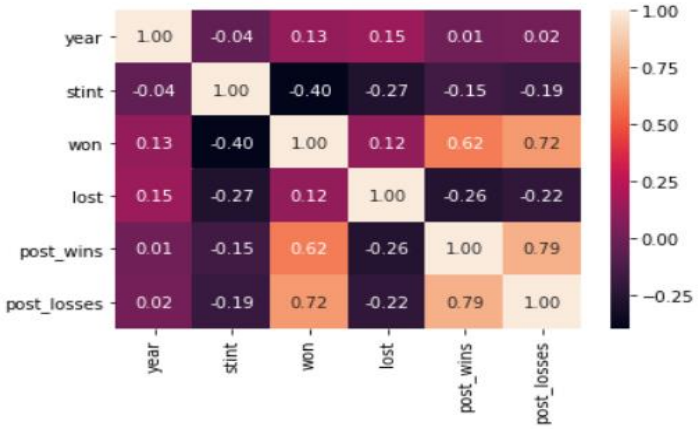
```
def Corr(xs, ys):
    xs = np.asarray(xs)
    ys = np.asarray(ys)

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

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

```
Corr(won, lost)
```

```
0.12096039322422208
```

```
import seaborn as sns
sns.heatmap(coaches.corr(), annot=True, fmt=".2f")
```

this and upper code for heatmap of correlation. Mean, variance and asarray which is like object such as tuples, lists etc., are used from code.



The correlation of won and lost is about 0.12, which is positive correlation. however, we cannot say that there is strong correlation.

SECTION 6

The following function computes the intercept and slope of the least squares fit.

```
from thinkstats2 import Mean, MeanVar, Var, Std, Cov

def LeastSquares(xs, ys):
    meanx, varx = MeanVar(xs)
    meany = Mean(ys)

    slope = Cov(xs, ys, meanx, meany) / varx
    inter = meany - slope * meanx

    return inter, slope
```

Here's the least squares fit to won as a function of lost.

```
inter, slope = LeastSquares(won, lost)
inter, slope
```

```
(29.802574437169778, 0.10464015562029239)
```

The intercept is often easier to interpret if we evaluate it at the mean of the independent variable.

ıntercept: represents the mean valıe in the sufficient group slope: represents the difference in means between groups

```
inter + slope * 25
```

```
32.41857832767709
```

```
slope
```

```
0.10464015562029239
```

Least squares formula used from this code.

Ho: β1= β2=0 (null hypothesis says that no linear relationship between two variables)

Ha: β1 ≠ β2≠ 0 (β1: win rate, β2: losing rate)

I used scipy to look p-values for linearity of two variables.

```
from scipy import stats
slope, intercept, r_value, p_value, std_err = stats.linregress(won,lost)
```

```
p_value
```

p-value is smaller than alpha values so reject null accept alternative which is conclude that there is linear relationship between win and lost

SECTION 7

In overall result of our statistical results shows that relationship of coaches' number of game losses and number of winning rates has positive relationship and not strong relationship. Also, coaches' standard deviations for win higher than lost standard deviation. From there we can understand some coaches' performance are more better than other ones' performance. Number of game losses has more regular distribution than number of winning.