

Mustafa Enes Tatlı

214061096

Cs 240 final project

**Part 1 ideas**

* Which leagues are have the lowest hit scores?
* Are there any relationship between high salary and beyzboll player?
* What is the correlation between the strikeouts in 2003 and 2005.

I created the hypothesis and my hypothesis is *“There is a strong correlation between the strikeouts in 2003 and 2005”.*

from \_\_future\_\_ import print\_function, division

%matplotlib inline

import numpy as np

import thinkstats2

import thinkplot

import matplotlib

import pandas as pd

The import the data is require this examining on the project.

Part 2

df = pd.read\_csv('Pitching.csv') #####I took the pitching datas.####

ten = df.SO[df.yearID == 2003].dropna()

eleven = df.SO[df.yearID == 2005].dropna()

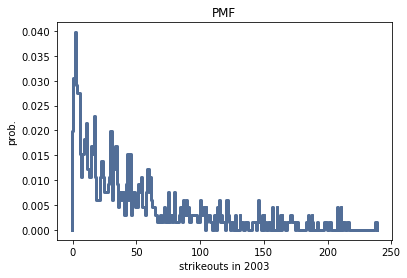
I used “Pitching.csv” file. Then I defined the columns that I am going to use. I will compare the strikeouts in 2003 and 2005, I create two variables called ten and eleven. I cleaned the empty data.

Part 3

hist\_eleven = thinkstats2.Hist(eleven)

thinkplot.Hist(hist\_eleven)

thinkplot.Show(xlabel='strikeouts in 2003',ylabel='Frequency', title='Histogram')

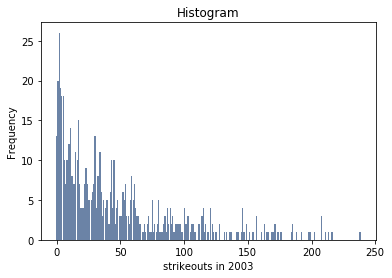


Pmf function to find the probability of each number of strikeouts. I show that the graph the same steps on histogram graph. In this graph it is easy to compare the values. As the number of strike outs in 2003 increase, the probability of players decrease.

pmf\_eleven =thinkstats2.Pmf(eleven)

thinkplot.Pmf(pmf\_eleven)

thinkplot.Show(xlabel="strikeouts in 2003", ylabel ='prob.',title="PMF")



Central point is approximately 25, the modes is 0 and 50, there is no outliers, the tails

İf the increase the strikeouse in 2003, the frequency is lower.

cdf\_eleven = thinkstats2.Cdf(eleven)

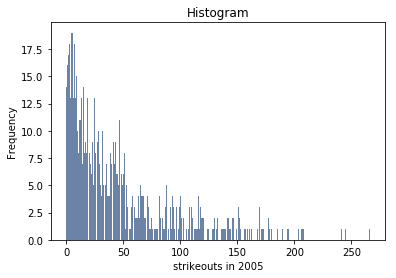
thinkplot.Cdf(cdf\_eleven)

thinkplot.Show(xlabel="strikeouts in 2003", ylabel='cdf', loc='upper left',title='cdf')

hist\_ten = thinkstats2.Hist(ten)

thinkplot.Hist(hist\_ten)

thinkplot.Show(xlabel='strikeouts in 2005',ylabel='Frequency', title='Histogram')

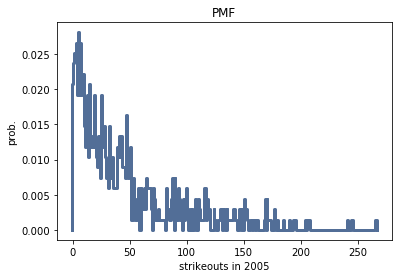


The likelihood of a strikeout in 2005 has a frequency of almost uninterrupted intervals between 0-50. Strikeout probability has a popular frequency between 0-10 and regularly decreasing frequency after. Frequencies between 50-150 are often less or less when the frequency is reduced.

pmf\_ten =thinkstats2.Pmf(ten)

thinkplot.Pmf(pmf\_ten)

thinkplot.Show(xlabel="strikeouts in 2005", ylabel ='prob.',title="PMF")



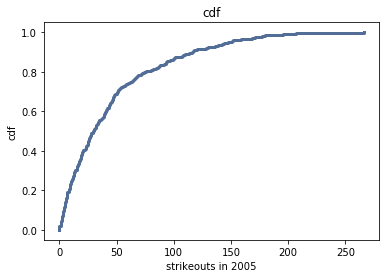
Examination of the PMF chart shows that strikeout possibilities are 50-50% between 0-50. Strikeout rate per game is regularly fluctuating between 50-150, but has unpredictable changes after 150.

cdf\_ten = thinkstats2.Cdf(ten)

thinkplot.Cdf(cdf\_ten)

thinkplot.Show(xlabel="strikeouts in 2005", ylabel='cdf', loc='upper left',title='cdf')

Part 4



The increase curve of strikeout values has a high trend between 0-50, and the probability of strikeout between 0-50 is almost 60%. The likelihood of 50+ strikeouts tends to increase regularly with lower incline.

def PercentileRank(scores, your\_score):

count = 0

for score in scores:

if score <= your\_score:

count += 1

percentile\_rank = 100.0 \* count / len(scores)

return percentile\_rank

PercentileRank(eleven,50)

def PercentileRank(scores, your\_score):

count = 0

for score in scores:

if score <= your\_score:

count += 1

percentile\_rank = 100.0 \* count / len(scores)

return percentile\_rank

PercentileRank(ten,50)

Part 5

histogramone = thinkstats2.Hist(ten,label='Number of Hits in 2003')

histogramtwo = thinkstats2.Hist(eleven,label='Number of Hits in 2005')

width = 0.45

thinkplot.PrePlot(2)

thinkplot.Hist(histogramone, align='right', width=width,color='blue')

thinkplot.Hist(histogramtwo, align='left', width=width)

thinkplot.Show(xlabel='# of Hit in 2003 - # of Hit in 2005', ylabel='Frequency',title='Histogram',loc='upper left')

pmf1 = thinkstats2.Pmf(ten,label='in 2003')

thinkplot.Pmf(pmf1,color='yellow')

pmf2 = thinkstats2.Pmf(eleven,label='in 2005')

thinkplot.Pmf(pmf2)

width = 0.45

thinkplot.PrePlot(2)

thinkplot.Show(xlabel='# of Hit in 2003 - # of Hit in 2005', ylabel='Probability',title='PMF')

cdf1 = thinkstats2.Cdf(ten,label='in 2003')

thinkplot.Cdf(cdf1,color='yellow')

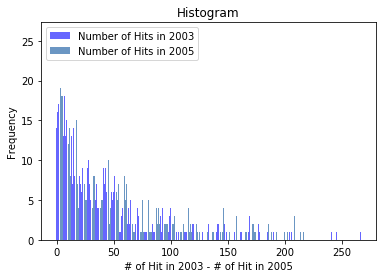
cdf2 = thinkstats2.Cdf(eleven,label='in 2005')

thinkplot.Cdf(cdf2)

width = 0.45

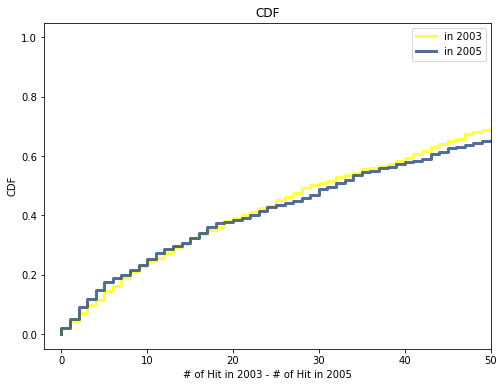
thinkplot.PrePlot(2)

thinkplot.Show(xlabel='# of Hit in 2003 - # of Hit in 2005', ylabel='CDF',title='CDF',xlim=[-2,50])



If strikeout frequencies in 2003 and 2005 are compared, strikeout rates in 2003 are more

unstable. While the rates in 2003 are more uplift, 2005 has more similar values.



Looking at the aggregations of 2003 and 2005, it can be seen that the year 2003 tends to be more stable. This shows that 2003 is more predictable. In 2005, it has more unpredictable values with more variable inclination.

Part 6

def SampleRows(df, nrows, replace=False):

indices = np.random.choice(df.index, nrows, replace=replace)

sample = df.loc[indices]

return sample

ten\_sample = SampleRows(ten, 654)

This code is very important tool the solve this problem the covarience because x column and y

column not equal we need to equal size x and y we put it ten(654).

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

print('Covariance is', (Cov(ten\_sample,eleven)))

def Corr(xs, ys):

xs = np.asarray(xs)

ys = np.asarray(ys)

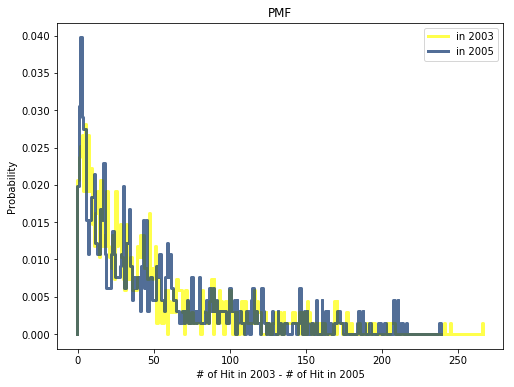
meanx, varx = thinkstats2.MeanVar(xs)

meany, vary = thinkstats2.MeanVar(ys)

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

return corr

print('Correlation is',Corr(ten\_sample,eleven)\*100)



The probabilities in 2003 are less wavy. This proves that probability is more regularly changing and more predictable. However, the strikeout probabilities of 2005 are very fluctuating, suggesting that the probabilities of 2005 are more unpredictable.

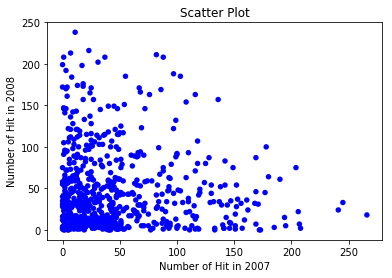
thinkplot.Scatter(ten\_sample, eleven, alpha=1)

thinkplot.Config(xlabel='Number of Hit in 2007',

ylabel='Number of Hit in 2008',

title="Scatter Plot",

legend=False)



class DiffMeansPermute(thinkstats2.HypothesisTest):

def TestStatistic(self, data):

group1, group2 = data

test\_stat = abs(group1.mean() - group2.mean())

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

data = ten.values, eleven.values

ht = DiffMeansPermute(data)

pvalue = ht.PValue()

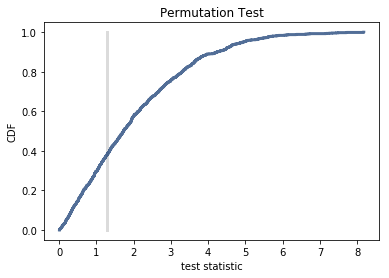
print (pvalue)

ht.PlotCdf()

thinkplot.Config(xlabel='test statistic',

ylabel='CDF',

title='Permutation Test')



The permutation test and the test statistic can be compared so that the break moment can be better analyzed.

Conclusion

All these data are examined, the probability and the tendency of the data given by the functions we use can be understood. For each data, the rate of change, aggregation, incremental increase and tendency of the data for different variables (eg: year) can be calculated. Thus, the analyzed data can be examined in more detail foreseeable. It is easier to comment on the post, for example, by reviewing the data between 2000 and 2010.

References

<https://github.com/AllenDowney/ThinkStats2/tree/master/code>

thinkstats.pdf into lms