**Week 7 Assignment**

Michael K. Zoucha

College of Science & Information Technology, Bellevue University

DSC 510: Introduction to Programming

Professor Eller

12 October 2021

**Week 7 Assignment**

Exercise 7-1:

Import necessary packages, and grab functions for correlation, covariance, and Spearman’s correlation given in the exercise above.

Text

Description automatically generated

first.MakeFrames() fills data frames for all live births, first births, and not first births. Dropna with a subset of agepreg and totalwgt\_lb drops all records with N/A in either of those fields.

live, firsts, others = first.MakeFrames()  
live = live.dropna(subset=['agepreg', 'totalwgt\_lb'])

Split live data frame into ages and weights, complete correalation / Spearman’s correlation

Corr 0.06883397035410904

SpearmanCorr 0.09461004109658226

ages = live.agepreg  
weights = live.totalwgt\_lb  
print('Corr', Corr(ages, weights))  
print('SpearmanCorr', SpearmanCorr(ages, weights))

Function BinnedPercentiles takes the data frame and bins the mother’s age and the baby’s birth weight before plotting CDF percentile of the binned data.

def BinnedPercentiles(df):  
  
 bins = np.arange(10, 48, 3)  
 indices = np.digitize(df.agepreg, bins)  
 groups = df.groupby(indices)  
  
 ages = [group.agepreg.mean() for i, group in groups][1:-1]  
 cdfs = [thinkstats2.Cdf(group.totalwgt\_lb) for i, group in groups][1:-1]  
  
 thinkplot.PrePlot(3)  
 for percent in [75, 50, 25]:  
 weights = [cdf.Percentile(percent) for cdf in cdfs]  
 label = '%dth' % percent  
 thinkplot.Plot(ages, weights, label=label)  
  
 thinkplot.Config(xlabel="Mother's age (years)",  
 ylabel='Birth weight (lbs)',  
 xlim=[14, 45], legend=True)  
  
BinnedPercentiles(live)

Chart

Description automatically generated

Create scatter plot of ages vs weights using scatterplot and config functions from thinkplot.

thinkplot.Scatter(ages, weights)  
thinkplot.Config(xlabel='Age (years)',  
 ylabel='Birth weight (lbs)',  
 xlim=[10, 45],  
 ylim=[0, 15],  
 legend=False)

Chart, scatter chart

Description automatically generated

Conclusion – There is no obvious relationships between the variables and the scatter plot makes it difficult to see any relationships. With such weak correlations (0.068 and 0.094) the math supports this conclusion.

Exercise 8-1:

Import necessary packages from exercise above. Function Estimate4 creates a random sample then computes the mean errors for median and mean using the randomly generated sample. MeanErrors function from above exercise calculates all errors in an array, then calculates the mean of those errors.

N = sample size = 7

Inters = interations = 100,000

Mu = mean = 0

Sigma = standard deviation = 1

Experiment 1

mean error xbar 0.0007870920521630876

mean error median 0.0018231026636941659

from \_\_future\_\_ import print\_function, division  
  
%matplotlib inline  
  
import numpy as np  
  
import brfss  
  
import thinkstats2  
import thinkplot  
  
def MeanError(estimates, actual):  
  
 errors = [estimate-actual for estimate in estimates]  
 return np.mean(errors)  
  
def Estimate4(n=7, iters=100000):  
  
 mu = 0  
 sigma = 1  
  
 means = []  
 medians = []  
 for \_ in range(iters):  
 xs = [random.gauss(mu, sigma) for i in range(n)]  
 xbar = np.mean(xs)  
 median = np.median(xs)  
 means.append(xbar)  
 medians.append(median)  
  
 print('Experiment 1')  
 print('mean error xbar', MeanError(means, mu))  
 print('mean error median', MeanError(medians, mu))  
  
Estimate4()

Function Estimate5 creates a random sample then computes RMSE using biased and unbiased samples. Function RMSE squares the error for all estimates, takes the mean of those squared errors, then returns the square root of that mean.

N = sample size = 7

Inters = interations = 100,000

Mu = mean = 0

Sigma = standard deviation = 1

Experiment 2

RMSE biased 0.5155813950903438

RMSE unbiased 0.5773697926168189

def RMSE(estimates, actual):  
  
 e2 = [(estimate-actual)\*\*2 for estimate in estimates]  
 mse = np.mean(e2)  
 return np.sqrt(mse)  
  
def Estimate5(n=7, iters=100000):  
  
 mu = 0  
 sigma = 1  
  
 estimates1 = []  
 estimates2 = []  
 for \_ in range(iters):  
 xs = [random.gauss(mu, sigma) for i in range(n)]  
 biased = np.var(xs)  
 unbiased = np.var(xs, ddof=1)  
 estimates1.append(biased)  
 estimates2.append(unbiased)  
  
 print('Experiment 2')  
 print('RMSE biased', RMSE(estimates1, sigma\*\*2))  
 print('RMSE unbiased', RMSE(estimates2, sigma\*\*2))  
  
Estimate5()

Exercise 8-2:

Function SimulateSample creates a random sample of exponential distribution, then calculates the RMSE for the estimates provided.

N = sample size

Lam = exponential distribution

Iters = iterations

def SimulateSample(n, lam=2, iters=1000):  
  
 estimates = []  
 for \_ in range(iters):  
 xs = np.random.exponential(1.0/lam, n)  
 lamhat = 1.0 / np.mean(xs)  
 estimates.append(lamhat)  
  
 stderr = RMSE(estimates, lam)  
 print('standard error', stderr)  
  
 cdf = thinkstats2.Cdf(estimates)  
 ci = cdf.Percentile(5), cdf.Percentile(95)  
 print('confidence interval', ci)  
 thinkplot.Plot([ci[0], ci[0]], [0, 1], color='0.8', linewidth=3)  
 thinkplot.Plot([ci[1], ci[1]], [0, 1], color='0.8', linewidth=3)  
  
 # plot the CDF  
 thinkplot.Cdf(cdf)  
 thinkplot.Config(xlabel='estimate',  
 ylabel='CDF',  
 title='Sampling distribution')  
  
 return stderr, n

trial\_1 (n = 10)

standard error 0.8140119872477184

confidence interval (1.2545003645348562, 3.7011795592017727)

Chart

Description automatically generated

trial\_2 (n = 100)

standard error 0.20474940515111376

confidence interval (1.6993708222949102, 2.3864387619545457)

Chart

Description automatically generated

trial\_3 (n = 1000)

standard error 0.0648685152262111

confidence interval (1.9015615361466522, 2.114929949233723)

Chart

Description automatically generated

Standard Error vs N:

0.8140119872477184 – 10

0.20474940515111376 – 100

0.0648685152262111 – 1000

A screenshot of a computer

Description automatically generated with medium confidence

Conclusion –

As the sample size increases, the RMSE decreases, almost exponentially. The range of the confidence interval also gets smaller as the sample size increases. All confidence intervals contain the actual value of 2.