11.12题

bootstrap

In [130]:

```
import pandas as pd
import numpy as np

h = [576, 635, 558, 578, 666, 580, 555, 661, 651, 605, 653, 575, 545, 572, 593]
w = [3.39, 3.30, 2.81, 3.03, 3.44, 3.07, 3.00, 3.43, 3.36, 3.13, 3.12, 2.74, 2.76, 2.88, 2.96]


def bootstrap(h,w, n):
    rd = list(n * np.random.rand(n))
    index = []
    xb = []
    for a in rd:
        index.append(int(a))
        xh = np.array(h)[index]
        xw =np.array(w)[index]
    return xh,xw


Nr = len(h)  #same quantities
corr_Nr = []
n = len(h)
for i in range(Nr):
    xh,xw = bootstrap(h,w,n)
    corr_Nr_i = np.corrcoef(xh, xw)[1][0]
    corr_Nr.append(corr_Nr_i)
seb = np.std(corr_Nr)
print("bootstrap estimate of standard error for the correlation co-efficient is:",seb)

#from matplotlib import pyplot as plt
#plt.hist(corr_Nr)
```

```
bootstrap estimate of standard error for the correlation co-efficient is: 0.16479501
956690956
```

由于是自助抽样原因，每次自助抽样产生的相关系数组估计标准误都不同，但与书上的值（Table6.1）接近，且直方图与Figure6.2相似

jackknife

In [59]:

```
len(h)
```

Out[59]:

```
15
```

In [68]:

```
corr_Nr_jackknife = []
for i in range(len(h)):
    lv_index = i#leave out one observation at a time
    lv_h =  np.delete(h,lv_index)
    lv_w = np.delete(w,lv_index)
    corr_Nr_i = np.corrcoef(lv_h, lv_w)[1][0]
    corr_Nr_jackknife.append(corr_Nr_i)
corr_Nr_jackknife  #theta1 - theta15
```

Out[68]:

```
[0.8934870155474253,
 0.7644565165072617,
 0.7558689178272322,
 0.7768176330972215,
 0.7321766454204205,
 0.7806694551467109,
 0.7852310086499684,
 0.7369936433928699,
 0.752535564286768,
 0.7768484704996289,
 0.818983096422017,
 0.7866804671121725,
 0.741270169364169,
 0.7678617198232243,
 0.7798725228771137]
```

根据书上公式

jackknife estimates of bias = (n-1)(theta(.)_head - theta_head)

In [73]:

```
theta_head = np.corrcoef(h, w)[1][0]
theta_head
```

Out[73]:

```
0.7771056872797375
```

In [75]:

```
bias_jack = (len(h)-1)*(np.mean(corr_Nr_jackknife)- theta_head)
bias_jack
```

Out[75]:

```
-0.006376965673733137
```

所以jackknife estimates of bias为-0.006376965673733137

In [86]:

```
se_jack = np.sqrt(((len(h)-1)/len(h))*
                  sum((corr_Nr_jackknife-np.mean(corr_Nr_jackknife))**2))
se_jack
```

Out[86]:

```
0.14226613931668444
```

所以jackknife estimates of standard error 为0.14226613931668444

而上面试验得到的bootstrap estimate of standard error for the correlation co-efficient is: 0.16479501956690956。 而我们的jackknife estimates of standard error接近且略小于Seb。

11.13题

Generate 100 samples X1 , X2, ... X2o from a normal population N(theta, 1) with theta= 1.

In [200]:

```
G_data = np.random.normal(loc=1.0, scale=1.0, size=(100,20))#m = 1,var =1
G_data
```

Out[200]:

```
array([[ 1.9945417 ,  1.20739504,  1.79202964, ...,  1.46577326,
         0.2869103 ,  0.56227091],
       [ 1.45768316, -0.05970277,  0.09069735, ...,  1.158172  ,
         2.07106071,  1.62620244],
       [ 0.44222873,  2.05203196,  0.95833834, ...,  0.0128134 ,
         1.29368225,  0.18705957],
       ...,
       [-0.49347114,  1.93158087,  1.35654829, ...,  0.19379661,
         1.00255705,  1.88790916],
       [ 0.76860743,  2.38734552,  1.56143672, ...,  1.27429396,
         1.18131036,  2.84548439],
       [ 0.26681985,  1.60489419,  2.23786605, ...,  1.94460477,
        -1.34450124,  0.47047007]])
```

a.compute the bootstrap and jackknife estimate of variance and

compute the mean and standard deviation of these variance estimates over the 100 samples.

a.1.1 bootstrap estimate of variance

In [201]:

```
Ve_b = []
for i in range(len(G_data)):#100 Sample
    Ve_b_i = []
    for j in range(200): #one Sample 这里我们bootstrap次数设为 B = 100
        G_i_b = np.random.choice(G_data[i],len(G_data[i]))#随机抽取20个
        theta_head_i = np.mean(G_i_b)
        Ve_b_i.append(theta_head_i)
    Ve_b_j =np.var(Ve_b_i)
    Ve_b.append(Ve_b_j)
```

In [202]:

```
Ve_b[:10]#取100个中 的前10个展示
```

Out[202]:

```
[0.02525797534807077,
 0.03537939014191139,
 0.05705766138494752,
 0.06389060957058217,
 0.04958660965392596,
 0.06219472771625764,
 0.03862448588449225,
 0.056586753012616224,
 0.06747521547757188,
 0.057446827280455574]
```

a.1.2 jackknife estimate of variance

根据书上已经得到的Se_jack（for Theta_head = mean(x)），我们只需要将那个公式平方，就能得到Var_jack

要计算100个Samples中每个Sample的Var_jack只需要循换100次:

In [203]:

```
jack_var_1 = []
for i in range(len(G_data)):#100 Sample
    jack_i_1 = []
    for j in range(len(G_data[i])): #one Sample 每个里做leave out one
        lv_index = j#leave out one observation at a time
        lv_Gdata_i = np.delete(G_data[i],lv_index) #丢弃那个元素
        theta_head_i = np.mean(lv_Gdata_i)  #theta = mean(X)
        jack_i_1.append(theta_head_i)
    jack_var_i = ((len(G_data[i])-1)/len(G_data[i]))*sum((jack_i_1-np.mean(jack_i_1))**2)#基本公式
    jack_var_1.append(jack_var_i)

jack_var_1[:10]#展示前10个
```

Out[203]:

```
[0.027356313845492145,
 0.03506250788915984,
 0.07155246188313429,
 0.06785284537642605,
 0.05049692236796917,
 0.0495718867925081,
 0.04201190890090382,
 0.054645050834143304,
 0.07497800523771138,
 0.04727573763112205]
```

a.2 compute the mean and standard deviation of these variance estimates over the 100 samples.

In [205]:

```
# mean
print("mean of bootstrap estimate of variance : ",np.mean(Ve_b))
print("mean of jack estimate of variance : ",np.mean(jack_var))
#standard deviation
print("standard deviation of bootstrap estimate of variance : ",np.std(Ve_b))
print("standard deviation of jack estimate of variance : ",np.std(jack_var))
```

```
mean of bootstrap estimate of variance :  0.048083016104914764
mean of jack estimate of variance :  0.04670164776776921
standard deviation of bootstrap estimate of variance :  0.017943236145941496
standard deviation of jack estimate of variance :  0.013721515336654186
```

b. Repeat (a) for the statistic theta = mean(X)^2 , and compare the results. Give an explanation for your findings.

b.1.1 bootstrap estimate of variance

In [190]:

```
Ve_b_2 = []
for i in range(len(G_data)):#100 Sample
    Ve_b_i_2 = []
    for j in range(200): #one Sample 这里我们bootstrap次数设为 B = 200
        G_i_b = np.random.choice(G_data[i],len(G_data[i]))#随机抽取20个
        theta_head_i = (np.mean(G_i_b))**2  #这里是唯一变化, theta = mean(X)^2
        Ve_b_i_2.append(theta_head_i)
    Ve_b_j =np.var(Ve_b_i_2)
    Ve_b_2.append(Ve_b_j)
Ve_b_2[:10] #展示前10个
```

Out[190]:

```
[0.13576527813198638,
 0.2193348745662607,
 0.25577856272334953,
 0.4291574204692577,
 0.14106565797190762,
 0.3563872616408253,
 0.18842509138282323,
 0.16159362488173393,
 0.05303773942676729,
 0.0855520094514176]
```

b.1.2 jack estimate of variance

In [191]:

```
jack_var_2 = []
for i in range(len(G_data)):#100 Sample
    jack_i_2 = []
    for j in range(len(G_data[i])): #one Sample 每个里做leave out one
        lv_index = j#leave out one observation at a time
        lv_Gdata_i = np.delete(G_data[i],lv_index) #丢弃那个元素
        theta_head_i = (np.mean(lv_Gdata_i))**2  #theta = mean(X)^2
        jack_i_2.append(theta_head_i)
    jack_var_2_i = ((len(G_data[i])-1)/len(G_data[i]))*sum((jack_i_2-np.mean(jack_i_2))**2)#基本公式
    jack_var_2.append(jack_var_2_i)

jack_var_2[:10]#展示前10个
```

Out[191]:

```
[0.15288622948113773,
 0.25569044537326024,
 0.2872439351391205,
 0.39683295642123645,
 0.14894055499875203,
 0.29266922978619614,
 0.19328071038656724,
 0.1366332279393174,
 0.06075996055377429,
 0.11751993463755704]
```

b.2 compute the mean and standard deviation of these variance estimates over the 100 samples.

In [192]:

```
# mean
print("mean of bootstrap estimate of variance : ",np.mean(Ve_b_2))
print("mean of jack estimate of variance : ",np.mean(jack_var_2))
#standard deviation
print("standard deviation of bootstrap estimate of variance : ",np.std(Ve_b_2))
print("standard deviation of jack estimate of variance : ",np.std(jack_var_2))
```

```
mean of bootstrap estimate of variance :  0.200468094747572 16
mean of jack estimate of variance :  0.20372470086059807
standard deviation of bootstrap estimate of variance :  0.11771926435939083
standard deviation of jack estimate of variance :  0.11343056971768681
```

explanation

前后两组对比试验发现，当theta估计从线性mean(x）变为非线性时mean(x)^2时。前者：jack的standard deviation 很接近bootstrap的standard deviation！后者:

与书上结论一致：The variability of the jackknife estimate is slightly larger than that of the bootstrap for the mean (a linear statistic) but is significantly larger for the correlation coefficient (a nonlinear statistic).对于平均值(线性统计量)，折刀估计的可变性略大于自助法，但对于相关系数(非线性统计量)则显著大于自助法。

In [156]:

```
0.10152309462927 76-0.11292851111407877
```

Out[156]:

```
-0.011405416484801173
```

In [ ]:

```

```