Project 6: Statistics and Bootstrapping

EE 511 – Section Thursday 9 am

Name: Junquan Yu

Student ID #: 3372029142

1. Problem Statement

The attached datasheet represents a set of 100 independent samples from some population.

a) Compute the sample mean m, and the sample variance s^2.

b) Use the data to generate a discrete approximation to the Cumulative Distribution Function – the empirical distribution, $F_{X^*}(x)$. Plot this distribution.

c) By splitting the data into equal size intervals (0-5, 6-10, etc), generate a discrete approximation to the distribution and determine the values of the Probability Mass Function for this discrete approximation.

d) Use the bootstrapping technique to generate M bootstrap samples based on the empirical distribution found in part b) and compute the sample mean and sample variance for each Bootstrap sample. Use $M = 50$ and $M = 100$.

e) Find the value of the MSE of the sample mean.

$$MSE_F(m) = E_F[(\mu - m)^2]$$

And compare to the variance of the sample means based on the bootstrap samples.

f) Calculate the (population) variance of the empirical distribution – call this We could evaluate

$$MSE_{F^*}(s^2) = E_F[(s^2 - \sigma^2_{F^*})^2]$$

By computing s^2 for all possible n^n samples that can be generated from the empirical distribution. That is a formidable computational task, so we consider only a (random) subset of such samples – i.e. the set of Bootstrap samples in part d) and use the sample variances found in part d) to estimate the MSE.

37.12	8.45	28.96	0.27	36.22
2.78	3.98	32.79	0.14	24.87
1.33	33.25	19.91	30.43	25.84
33.55	31.10	1.86	30.57	5.34
45.39	28.67	7.12	35.38	1.92
9.25	12.55	27.49	33.72	2.30
28.32	30.92	32.62	24.10	33.56
35.62	27.88	20.71	36.62	24.03
28.00	31.44	33.32	5.01	1.30
4.56	2.28	11.33	0.24	8.53
5.27	18.52	7.63	31.03	4.06
12.83	15.43	8.75	4.65	5.21
7.90	26.48	6.81	32.20	25.69
18.18	4.48	30.33	1.68	28.44
23.26	3.35	0.17	8.90	13.29
31.54	26.16	22.79	6.89	27.92
30.99	6.93	13.27	10.08	28.95
13.40	4.57	34.10	0.76	36.40
0.60	39.74	1.11	2.40	1.05
34.10	29.95	1.94	0.16	1.43

2. Theoretical Exploration or Analysis

In statistics, bootstrapping is any test or metric that relies on random sampling with replacement. Bootstrapping allows assigning measures of accuracy (defined in terms of bias, variance, confidence intervals, prediction error or some other such measure) to sample estimates.[1][2] This technique allows estimation of the sampling distribution of almost any statistic using random sampling methods.[3][4] Generally, it falls in the broader class of resampling methods.

Bootstrapping is the practice of estimating properties of an estimator (such as its variance) by measuring those properties when sampling from an approximating distribution. One standard choice for an approximating distribution is the empirical distribution function of the observed data. In the case where a set of observations can be assumed to be from an independent and identically distributed population, this can be implemented by constructing a number of resamples with replacement, of the observed dataset (and of equal size to the observed dataset).

It may also be used for constructing hypothesis tests. It is often used as an alternative to statistical inference based on the assumption of a parametric model when that assumption is in doubt, or where parametric inference is impossible or requires complicated formulas for the calculation of standard errors.

3. Simulation Methodology

For the problem a:

I just typed the all of the sample data from the homework PDF into an excel file called "Data Samples" and loaded these data into the Matlab with the built-in function xlsread( ). Then I used the function mean( ) and var( ) to calculate the sample mean and sample variance respectively.

For the problem b and problem c:

I created several equal-sized intervals to accommodate these data and checked every number in the data. If the number I checked belongs to the certain interval, then the PMF counts for that interval plus one. In addition, if the I checked is less than the up bound of certain interval, then Fx counts for that interval plus one. After traversing all the number, I can get the Fx for these data, which is Fx=Fx counts/200. Similarly, the PMF for these data is PMF=PMF counts/200.

For the problem d:

I generated the bootstrap samples with the function randsample( ) and then used the functions mean( ) and var( ) to calculate the mean and variance of these bootstrap samples.

For the problem e:

I created a loop structure to calculate the MSE of the sample means using the sample means from the problem d. Then I compared it to the variance of the sample means based on the bootstrap samples.

For the problem f:

I created a loop structure to calculate the $MSE_{F^*}(s^2)$ using the population variance of the empirical distribution σ^2_{F*}.

4. Experiments and Results

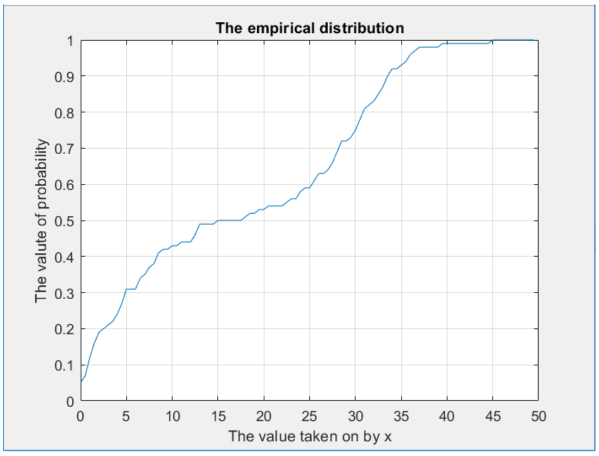
For the problem a:

```
m =

17.647100000000005337597031029873


variance =

177.23229352525245872129744384438
```

It can be seen from the above screenshot that the sample mean $m = 17.6471$ and the sample variance $s^2 = 177.2323$.

For the problem b:

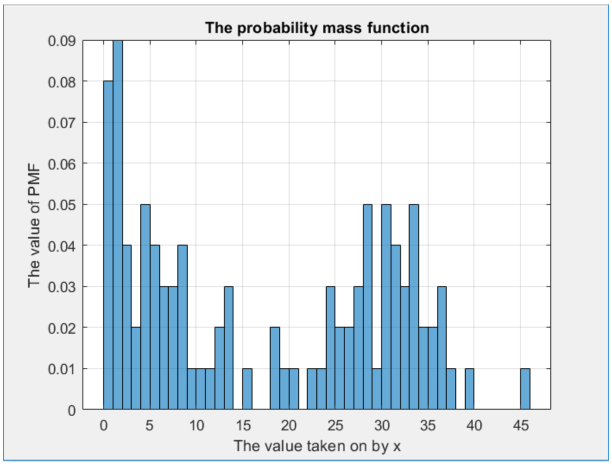
The discrete approximation to the Cumulative Distribution Function – the empirical distribution is showed below:



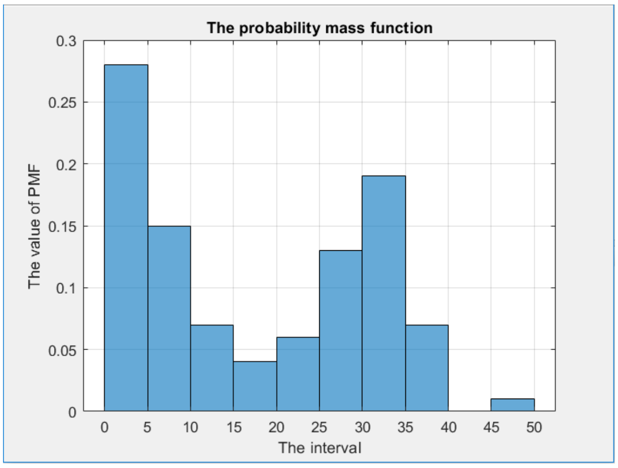
For the problem c:

I split the data into the equal size intervals, which are 1, 5 and 10. Then I simulated for each of them and the results are as follows:

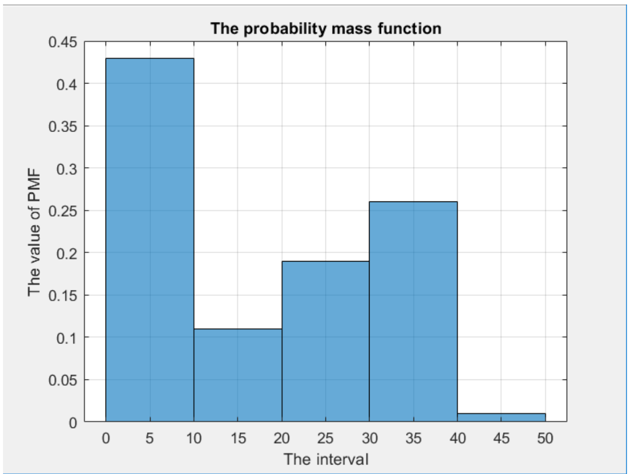
When the size of intervals is set to 1:



When the size of intervals is set to 5:



When the size of intervals is set to 10:



For the problem d:

I simulated and computed the sample mean and the sample variance in the bootstrap samples scenarios with M=50 and 100. The results are as follows:

When 50 bootstrap samples were generated, the sample mean of bootstrap samples is:

```
mean_sample =

  Columns 1 through 10

   18.5936   17.2972   13.8308   20.1544   17.8812   17.6806   18.3582   18.3212   15.5502   16.5604

  Columns 11 through 20

   17.3668   17.6352   21.6174   14.4808   16.3914   16.4014   18.7780   16.7398   15.1470   18.2184

  Columns 21 through 30

   15.8128   17.8026   15.0856   16.0788   20.2522   19.4314   18.3234   16.1736   14.7646   15.8328

  Columns 31 through 40

   22.4832   16.0912   16.2674   15.9010   14.8666   18.6434   16.6956   18.0664   17.4740   17.2222

  Columns 41 through 50

   15.2974   18.5572   15.7214   18.8012   21.3632   13.9778   17.1488   18.0932   17.9882   18.3912
```

When 50 bootstrap samples were generated, the sample variance of bootstrap samples is:

```
var_sample =

  Columns 1 through 10

  227.7178  183.3935  163.6369  192.5490  169.7505  177.3744  188.6601  150.2652  158.7004  164.6407

  Columns 11 through 20

  181.9951  178.5001  177.3251  149.9405  154.3046  156.5397  149.7021  177.3986  190.0071  156.2787

  Columns 21 through 30

  184.2365  198.1244  168.1065  189.7427  166.8585  176.4291  171.7301  122.7201  147.5349  184.1941

  Columns 31 through 40

  166.4695  170.3889  233.4499  177.0201  170.5499  145.5881  178.6318  164.3697  160.3446  164.2131

  Columns 41 through 50

  136.3230  183.8323  156.2007  219.0598  146.5648  185.0397  184.4468  139.3099  156.7120  177.7314
```

When 100 bootstrap samples were generated, the sample mean of bootstrap samples is:

```
mean_sample =

  Columns 1 through 10

   15.6266   19.1835   16.8945   20.7948   16.5950   17.0494   18.7409   18.4271   15.6339   18.7997

  Columns 11 through 20

   18.5432   18.2640   19.1753   17.7046   19.6527   16.7050   17.2861   18.9601   18.6312   18.0751

  Columns 21 through 30

   15.5129   17.4166   17.1560   17.4862   17.8612   17.5184   16.6720   16.9230   17.5616   15.1059

  Columns 31 through 40

   18.6483   18.1704   16.2624   16.6435   20.0513   16.0955   16.9553   17.8439   16.1691   20.0200

  Columns 41 through 50

   17.1182   17.3250   14.9847   17.9329   17.4411   20.2599   16.2732   17.9552   19.3947   17.3113

  Columns 51 through 60

   19.6737   18.2441   16.7023   18.2733   15.3049   18.8122   18.9398   17.9582   19.5341   16.1453

  Columns 61 through 70

   18.1882   16.6882   16.5732   18.9686   16.4736   20.5798   20.5934   16.2990   17.2237   18.2651


  Columns 71 through 80

   17.3544   17.1020   21.5322   17.1022   19.9670   18.7177   17.2033   18.7568   16.7223   19.1559

  Columns 81 through 90

   18.8137   17.1928   17.3364   15.8317   17.3241   18.3636   17.9013   17.5606   16.4760   18.3272

  Columns 91 through 100

   18.3517   17.8380   17.8481   16.9874   20.3652   18.2531   17.7333   18.2312   16.2013   20.8786
```

When 100 bootstrap samples were generated, the sample variance of bootstrap samples is:

```
var_sample =

  Columns 1 through 10

  176.4747  185.4398  169.2663  175.1632  173.3243  192.7230  191.1933  155.7123  173.0485  190.5642

  Columns 11 through 20

  187.3457  163.0218  187.3994  199.6205  194.8217  179.5604  183.1115  160.8484  187.4207  175.6257

  Columns 21 through 30

  164.3265  186.8082  155.8841  175.8319  156.4379  179.6166  170.9359  195.9176  182.0590  173.0535

  Columns 31 through 40

  179.2316  174.1284  164.9895  163.8050  185.9666  189.2978  176.6410  189.5886  179.7289  177.3807

  Columns 41 through 50

  180.2032  182.7750  155.6110  199.5748  176.3708  164.5188  177.2946  171.0566  178.6185  170.7235

  Columns 51 through 60

  171.5854  173.3291  182.1274  185.5790  140.1261  176.5572  174.3465  160.2697  165.4110  177.7353

  Columns 61 through 70

  146.2379  176.5166  169.5793  191.0666  188.5953  156.7463  195.0690  176.8473  177.4771  174.8045
```

```
Columns 71 through 80

187.7380   175.7572   163.9185   169.3489   175.6295   184.5443   210.5725   160.9834   189.2171   199.3133

Columns 81 through 90

192.0668   170.5290   194.2226   183.7002   180.8263   185.2735   200.6945   187.9739   196.2397   179.0103

Columns 91 through 100

170.0927   185.7435   170.0784   169.7846   172.4473   157.0525   187.5548   170.9769   167.7296   176.5803
```

For the problem e:

I simulated 5 times for each of bootstrap samples scenarios with M=50 and 100. The results are as follows:

When 50 bootstrap samples were generated:

Times	1	2	3	4	5
MSE(m)	4.1849	2.5441	2.9722	3.7552	2.7765
Variance	4.1793	2.5948	3.0276	3.8143	2.8082

When 100 bootstrap samples were generated:

Times	1	2	3	4	5
MSE(m)	1.8067	1.7338	1.5645	1.8774	1.8007
Variance	1.8240	1.7432	1.5801	1.8963	1.7569

It can be seen from above tables that the simulation results is very close to the theoretical value about MSE(m) and the MSE(m) and the variance of the sample means are almost the same whether the M=50 or M=100.

For the problem f:

I simulated 5 times for each of bootstrap samples scenarios with M=50 and 100. The results are as follows:

When 50 bootstrap samples were generated:

Times	1	2	3	4	5

$MSE_{F^*}(s^2)$	336.6890	371.5305	325.2111	347.4983	331.0944

When 100 bootstrap samples were generated:

Times	1	2	3	4	5
$MSE_{F^*}(s^2)$	190.6101	151.2661	167.1836	166.4264	160.2743

It can be seen from above tables that the value of $MSE_{F^*}(s^2)$ tends to increase as the value of M decreases.

5. References

1. *Alberto Leon-Garcia. (2008). Probability, Statistics, and Random Processes for Electrical Engineering. Upper Saddle River, NJ 07458. Pearson Education, Inc.*

2. *Zhou Sheng, Shiqian Xie, Chengyi Pan. (2008). Probability Theory and Mathematical Statistics. No.4, Dewai Street, Xicheng District, Beijing. Higher Education Press.*

6. Source Code

```
a=linspace(0,50,101);
Fx=zeros(1,100);
PMF=zeros(1,100);
M=100;
num_input=xlsread('Data Samples.xlsx');
for i=1:100
    num(i)=num_input(i);
    for j=1:100
        if(num(i)<a(j+1))
            Fx(j)=Fx(j)+1;
            if(num(i)>a(j))&&(num(i)<a(j+1))
                PMF(i)=(a(j)+a(j+1))/2;
            end
        end
    end
end
m=mean(num,2)
variance=var(num)
Fx=Fx/100;
```

```
figure(1);
histogram(PMF,'Normalization','probability','BinWidth',10);
title('The probability mass function')
ylabel('The value of PMF')
xlabel('The interval')
grid on
figure(2);
x=linspace(0,49.5,100);
plot(x,Fx);
grid on
title('The empirical distribution')
ylabel('The value of probability')
xlabel('The value taken on by x')
for i=1:M
   sample=randsample(num,M,1);
   mean_sample(i)=mean(sample);
   var_sample(i)=var(sample);
end
mean_sample
var_sample
var_mean=var(mean_sample)

MSE_m=0;
for i=1:M
   MSE(i)=(mean_sample(i)-m)^2;
end
MSE_m=mean(MSE)
MSE_var=0;
for i=1:M
   MSE(i)=(var_sample(i)-variance)^2;
end
MSE_var=mean(MSE)
```