Planning Document

1. Bootstrap functions (five functions):

a. Non-parametric percentile method

Resample by sample, don't know the distribution

The upper limit: $(1-\alpha/2)(b+1)\%th$

The lower limit: $\alpha/2(b+1)\%th$

Met problem: the determination of percentile
Solution: use quantile()

b. Non-parametric BCa method

Resample by sample, don't know the distribution
The upper limit: as lecture notes
The lower limit: as lecture notes
Met problem: Understanding the calculation of α hat
Solution: understand get.ahat()

c. Parametric method

Resample by the parameter
Know the distribution, estimate then fit the model
Generate resamples by the distribution
Use the resamples to estimate again

i. By percentile method
ii. By Bca

Met problem: Poisson how to deal with standard cdf, etc..
Solution:
It seems no standard Poisson distribution, so I used the mean of estimated boot.mean, but the performance of this way is not good at all. A large number of estimated means are filtered. Thus, I am thinking whether BCa is only available for non-parametric bootstrap.
Finally, R document in RStudio of boot.ci() and the following reference proved my assumption
https://www.umassmed.edu/contentassets/a7bd41506c5a4308b401e312a7ff59fc/bootstrap-overview.pdf
So I delete the BCa part of parametric method, the original code is in deleted_BCa_Parametric.r

d. (optional) Read the book An Introduction to bootstrap methods with applications to R to find other methods

An introduction to bootstrap methods with applications to R / Michael R. Chernick, Robert A. LaBudde.
Chernick, Michael R.
Hoboken, N.J. : Wiley, c2011.

Finally chose the balanced bootstrap with percentile method
Here is the reference: http://www.maths.qmul.ac.uk/~bb/CTS_Chapter4_5-8_Students.pdf
It is interesting, let's see its performance

Met Problem: how to control the number of a certain value sampled by sample(), to make the number less than or equal to N
Solution 1: it is very difficult to control it, so I control it manually.
Solution 2: it is still very difficult to control the N, so I give up to use sample() without limiting N. The results show it is okay to generate the estimated mean but not appropriate to generate a CI. Consequently, I will not analysis the CI under this method.

2. Simulation study:

a. What need to be compared:

i. Different distributions (Normal or Poisson) with a big sample size by the same method with the same alpha (simulation 1 vs simulation 2, simulation 5 vs simulation 6, simulation 9 vs simulation 10,)

ii. Different distributions (Normal or Poisson) with a small sample size by the same method with the same alpha
(simulation 3 vs simulation 4, simulation 7 vs simulation 8, simulation 11 vs simulation 12)
iii. The same distribution with different sample sizes by the same method with the same alpha
(simulation 1 vs simulation 3, simulation 2 vs simulation 4)
(simulation 5 vs s 7, s6 vs s8)
(s9 vs s11, s10 vs s12)
iv. The same distribution with the same sample size by different methods with the same alpha
(s1 vs s5 vs s9)
(s2 vs s6 vs s10)
(s3 vs s7 vs s11)
(s4 vs s8 vs s12)
v. The same distribution with the same sample size by the same method with different alphas
(s1, s13, s19) (s2,14, 20) (5, 15, 21) (6,16, 22) (9,17, 23)(10,18,24)

Bootstrap methods need to be checked:

1) NonPara.percentileMethod ()
2) NonPara.BCaMethod ()
3) Para.percentileMethod ()

Met Problem & solution : Notice that 1000 observations Are still Small, Thus Change 10000 to big And 1000 to small

Function Simulation No.	Bootstrap Method	Sample size	alpha	Distribution	Comment with round 10	Comment with round 1000
1	1	10000	0.05	normal	True mean always in the CIs	True mean always in the CIs
2	1	10000	0.05	poisson	True mean always in the CIs	True mean always in the CIs
3	1	1000	0.05	normal	True mean never in the CIs	True mean never in the CIs
4	1	1000	0.05	poisson	True mean always in the CIs	True mean always in the CIs
5	2	10000	0.05	normal	True mean always in the CIs	True mean always in the CIs
6	2	10000	0.05	poisson	True mean always in the CIs	True mean always in the CIs
7	2	1000	0.05	normal	True mean never in the CIs	True mean never in the CIs
8	2	1000	0.05	poisson	True mean always in the CIs	True mean always in the CIs
9	3	10000	0.05	normal	True mean always in the CIs	True mean always in the CIs
10	3	10000	0.05	poisson	True mean always in the CIs	-
11	3	1000	0.05	normal	True mean never in the CIs	-
12	3	1000	0.05	poisson	True mean always in the CIs	-
13	1	10000	0.01	normal	True mean always in the CIs	-
14	1	10000	0.01	poisson	True mean always in the CIs	-
15	2	10000	0.01	normal	True mean always in the CIs	-

16	2	10000	0.01	poisson	True mean always in the CIs	-
17	3	10000	0.01	normal	True mean always in the CIs	-
18	3	10000	0.01	poisson	True mean always in the CIs	-

Met Problem:

a) it is very bad CI coverage, there must be something wrong
b) The BCa CI takes so much time, one function almost need 1-2 hours with round 1000
c) The way of generating data in testing seems to be wrong
d) The sample size 1000 should not be a small number

Solution :

a) Fix the way of generating data
b) Misunderstand: asked one of my classmate, she said I should compare the true mean, not the population mean
c) Stop the above tests, it had already cost me 20 hours and results were useless

New testing:

Round 10:

```
> simulation1()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.183903859039906"
[1] "The mean of the dataset:  3.8405292932576  The mean of the boostrap:  3.
83723965406251"
[1] "The difference of the means:  0.00328963919508851"
> simulation2()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0796775000000003"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.0046397715
3361"
[1] "The difference of the means:  0.00136022846638717"
> simulation3()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.150341291957213"
[1] "The mean of the dataset:  5.66277710116312  The mean of the boostrap:  5
.66219866159703"
[1] "The difference of the means:  0.000578439566096378"
```

```
> simulation4()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0330725000000007"
[1] "The mean of the dataset:  4  The mean of the boostrap:  3.99771275"
[1] "The difference of the means:  0.00228725000000018"
```

```
> simulation5()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0370573672079422"
[1] "The mean of the dataset:  3.96810585865152  The mean of the boostrap:  3
.96785205263476"
[1] "The difference of the means:  0.000253806016763836"
> simulation6()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0846688634926114"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.0054173046
6453"
[1] "The difference of the means:  0.000582695335473282"
> simulation7()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0266326998688902"
[1] "The mean of the dataset:  4.33255542023262  The mean of the boostrap:  4
.33283406427261"
[1] "The difference of the means:  -0.000278644039985565"
> simulation8()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0318773574922502"
[1] "The mean of the dataset:  4  The mean of the boostrap:  3.99793135729"
[1] "The difference of the means:  0.0020686427100034"
```

```
> simulation9()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0375727979349874"
[1] "The mean of the dataset:  3.96810585865152  The mean of the boostrap:  3
.9679555492692"
[1] "The difference of the means:  0.000150309382322522"
> simulation10()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0744957043401682"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.0056943997
4462"
[1] "The difference of the means:  0.000305600255384064"
> simulation11()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0251525882369181"
[1] "The mean of the dataset:  4.33255542023262  The mean of the boostrap:  4
.33190512512384"
[1] "The difference of the means:  0.000650295108789223"
> simulation12()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0334605111445869"
[1] "The mean of the dataset:  4  The mean of the boostrap:  3.99918823837655
"
[1] "The difference of the means:  0.000811761623447982"
```

```
> simulation13()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.244833460337704"
[1] "The mean of the dataset:  3.8405292932576  The mean of the boostrap:  3.83743575397134"
[1] "The difference of the means:  0.00309353928625899"
> simulation14()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.109008499999999"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.00466663457661"
[1] "The difference of the means:  0.00133336542338736"
> simulation15()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0476996603981088"
[1] "The mean of the dataset:  3.96810585865152  The mean of the boostrap:  3.96763389818624"
[1] "The difference of the means:  0.000471960465278976"
> simulation16()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.108728372910087"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.00488802020063"
[1] "The difference of the means:  0.00111197979937305"
> simulation17()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0502265198452911"
[1] "The mean of the dataset:  3.96810585865152  The mean of the boostrap:  3.96797427963354"
[1] "The difference of the means:  0.000131579017981222"
> simulation18()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.100229728283109"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.00573177263866"
[1] "The difference of the means:  0.000268227361336137"
> simulation19()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.213835808008336"
[1] "The mean of the dataset:  3.8405292932576  The mean of the boostrap:  3.83737041690612"
[1] "The difference of the means:  0.00315887635147316"
> simulation20()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.100531"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.00465603564155"
[1] "The difference of the means:  0.00134396435845208"
> simulation21()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0435692942424497"
[1] "The mean of the dataset:  3.96810585865152  The mean of the boostrap:  3.96767049098436"
[1] "The difference of the means:  0.000435367667161124"
> simulation22()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.100567173277188"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.00507945511979"
[1] "The difference of the means:  0.000920544880209206"
> simulation23()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0437712945543871"
[1] "The mean of the dataset:  3.96810585865152  The mean of the boostrap:  3.967971073167"
[1] "The difference of the means:  0.000134785484515731"
> simulation24()
[1] "The percentage that the boostrap CI contains the ture mean is:  1"
[1] "The percentage the ture mean is smaller than the CI:  0"
[1] "The percentage the ture mean is bigger than the CI:  0"
[1] "The mean of the length of the CI:  0.0871194288924162"
[1] "The mean of the dataset:  4.006  The mean of the boostrap:  4.00572576583094"
[1] "The difference of the means:  0.000274234169062026"
```

Met Problem: the alpha cannot reach ideal situations. The simulation functions start to be hard to use, need to be improved. Still confused about comparing which mean

Solution:

a) I think I still should compare the population mean, or the bootstrap is meanless

b) ~~Always 1 or 0 proportion because I forgot to delete set.seed~~

c) This is not about deleting set.seed. I cannot delete it. It is very important.
If I delete it, the proportion "seems" better, not only 0 or 1. But it is wrong. Our aim is to use bootstrap samples to estimate the mean. **Thus, I must generate the same sample.** Thus, to deal with this problem, I must separate sample and resamples to make set.seed not influence resamples. At beginning, I just make them in different functions in the same R file. But it did not work. So I have to make the process of generating the sample in one r file

named simulation_drive_v2.r and left the resample functions in bootstrapFunction.r and the simulation functions in simulation_v2.r.

Conclusion: set.seed() seems easy and boring, but it really matters for statistical inferences. I used almost five hours to debug just because this "simple" problem.

New testings: (improving the function also improves the efficiency of testing, only 8 tests needed)

Function Simulation No.	Bootstrap Method	Sample size	alpha	Distribution	Comment with round 10	Comment with round 100	**B = 10**	**B = 100**
1	1, 2, 3	50	0.05	normal				
2	1, 2, 3	50	0.05	poisson				
3	1, 2, 3	10	0.05	normal				
4	1, 2 , 3	10	0.05	poisson				
5	1, 2, 3	50	0.01	normal				
6	1,2 , 3	50	0.01	poisson				
7	1, 2, 3	50	0.02	normal				
8	1, 2 ,3	50	0.02	poisson				

I should compare B, this is the thing I ignored.

a. Steps:
   i. Set a seed
   ii. Generate samples/sample
   iii. Run bootstrap functions/function
   iv. Compare outcomes from bootstrap functions and the samples
   v. Classify outcomes and give conclusion
      1) Text conclusion:
         a) The coverage of true mean
         b) Equal tail coverage
         c) The length of the CI
      2) Graphic conclusion:
         a) (optional) The plots of each bootstrap's estimators' distribution (not very necessary I think. It seems to be a kind of wasting resources if I want to loop several times, so I want to give up this option)
            Finally I added this function:
            The code is simple:
            hist(estimators)
            abline(v = truemean, col = "green")
   vi. Other improvements:
      1) use csv to make the data easy to be checked, the format should be

Method	CI coverage	Smaller	Bigger	truemean	bootmean	n	B	**round**	difference	length	distribution
1	0.9	0	0.1	2	2.8	20	100		0.32423	1.123	normal
2	1										poisson
3	0										
1	0.8										
…	…	…	…	…	…	…	…	…	…	…	…

(1) NonPara.percentileMethod ()
(2) NonPara.BCaMethod ()
(3) Para.percentileMethod ()

Met Problem:
Write.csv() cannot use append… This is really not convenient!!!
Forgot alpha
Solution:
Use write.table. Deal with add column name's warning for a long time…

Add alpha.set

How to analyse this data set? Back to 2.a and add B and round's effects.

(1) B
(2) N
(3) round
(4) Different methods
(5) Different alpha
(6) Different distribution
(7) Others: sample mean

How to show it? Markdown. Emmm….. Not fast and convenient, although it is beautiful, give up

Record:

Normal, α = 0.05, n =50

######summary conlclusions######################### b 10 round 100
1. overall performance worse than B1 round1
2. m1, m2, m3 's not coverager are all balanced
3. coverage m1 and m3 is better
4. no obvious difference of boot means
5. m2's cis' lenght is the smallest

######summary conlclusion#################### b 100 round 100
1. round increases, the coverage seems not better than b = 10, round = 10
2. all mehtods' not coveraged means tend to be bigger than the CIs
3. no obvious difference for coverage
4. boot mean no obvious difference
5. ci lenght no obvious diff

######summary conlclusion#################### b 1000 round 1000
1. coverage becomes better, 0.5, no obvious difference between methods
2. all mehtods' not coveraged means tend to be bigger than the CIs
3. boot mean no obvious difference
4. ci lenght no obvious diff

Poisson, α = 0.05, n =50

######summary conlclusion####################
1. m1 and m3 coverage better mean 0.4, m2 0.37
2. all mehtods' not coveraged means tend to be balanced
3. boot mean m3 the best
4. ci lenght m2 better
######summary conlclusion####################
1. the coverage: m1 the best, m3 varies 0.46
2. m3' not coveraged means tend to be balanced, m1 and m2 tend to be bigger
3. m3 has the best boot mean, m2 the second best(may because 4)
4. ci lenght m2 better
######summary conlclusion####################
1. the coverage almost the same
2. m3 has the best boot mean, m2 the second best
3. ci lenght m3 the best, m1 has the longest

Normal, α = 0.05, n =10

######summary conlclusion#################### B 100
1. m3 best but no obvious difference
2. all mehtods' not coveraged means tend to a little bit bigger set
3. boot mean no obvious difference
4. ci lenght no obvious difference

######summary conlclusion#################### B 1000
1. coverage no obvious difference
2. all mehtods' not coveraged means tend to be balanced
3. boot mean no obvious difference
4. ci lenght m1 and m2 better, no obvious difference

poisson α = 0.05, n =10

######summary conlclusion####################
1. coverage no obvious difference

2. all mehtods' not coveraged means tend to be balanced
3. boot mean no obvious difference, the range of m2's means are the smallest
4. ci lenght m1 and m2 better

######summary conlclusion###################
1. coverage no obvious difference
2. all mehtods' not coveraged means tend to be balanced
3. boot mean m2 best
4. ci lenght no obvious difference

Normal, a = 0.01, n =50
######summary conlclusion###################
1. coverage no obvious difference, m1 and m3 are a little better
2. all mehtods' not coveraged means tend to be balanced
3. boot mean m2 best
4. ci lenght no obvious difference, m2 better a little

###super bad###summary conlclusion###################
1. coverage are very bad, 0.02
2. all mehtods' not coveraged means tend to be bigger than CI, very high
3. boot mean no obvious difference
4. ci lenght no obvious difference

poisson a = 0.01, n =50

######summary conlclusion###################
1. coverage becomes normal
2. all mehtods' not coveraged means tend to be balanced
3. boot mean no obvious difference
4. ci lenght m2 best

######summary conlclusion###################
1. coverage becomes normal
2. all mehtods' not coveraged means tend to be balanced
3. boot mean no obvious difference
4. ci lenght no obvious difference

Normal, a = 0.02, n =50
######summary conlclusion###################
1. coverage becomes normal
2. all mehtods' not coveraged means tend to be smaller than CI
3. boot mean no obvious difference
4. ci lenght no obvious difference

######summary conlclusion###################
1. coverage becomes normal
2. all mehtods' not coveraged means tend to be balanced
3. boot mean no obvious difference
4. ci lenght no obvious difference, m1 and m2 better a little

poisson a = 0.02, n =50

######summary conlclusion###################
1. coverage becomes normal
2. all mehtods' not coveraged means tend to be balanced
3. boot mean no obvious difference
4. ci lenght no obvious difference

######summary conlclusion###################
1. coverage becomes normal
2. all mehtods' not coveraged means tend to be balanced
3. boot mean no obvious difference
4. ci lenght m1 and m2 are better, m1 and m2 no obvious difference