

Take Home Exam II

Xingche Guo

December 1, 2017

1 Introduction

The main purpose of this study is to learn the different parametric methods, (e.g. **Wald's Theory**, **Inversion of Likelihood Ratio**, and **Parametric Bootstrap**), for constructing confident intervals of the parameters in Inverse Gaussian distribution under different conditions (e.g. differences in sample size n and in dispersion parameter λ). On the other hand, this research gives me a chance to practice conducting a Monte Carlo study and to learn how to compare methods by different criteria.

I'm interested in doing this because this study is very meaningful. The results tells me the pros and cons of these methods under different conditions, which might give me a better idea of selecting methods to conduct confidence intervals. This study also shows how the sample size n and dispersion parameter λ affect the confidence interval features. Besides, it sheds light on determining how big the Monte Carlo sample size M and bootstrap sample size N should be, which are all very useful in my later research.

2 Methods

2.1 Method 1

The density function of an Inverse Gaussian distribution can be written as:

$$f(y|\mu, \lambda) = \sqrt{\frac{\lambda}{2\pi y^3}} \exp\left(-\frac{\lambda(y - \mu)^2}{2\mu^2 y}\right); \quad y > 0; \mu > 0; \lambda > 0$$

Thus, the joint density of iid r.v. Y_1, \dots, Y_n is:

$$f(\mathbf{y}|\mu, \lambda) = \prod_{i=1}^n f(y_i|\mu, \lambda)$$

The log likelihood function of the joint density is:

$$\begin{aligned}
l(\mu, \lambda; \mathbf{y}) &= \sum_{i=1}^n \log f(y_i | \mu, \lambda) \\
&= \sum_{i=1}^n \left(\frac{1}{2} \log \frac{\lambda}{2\pi y_i^3} - \frac{\lambda(y_i^2 - 2y_i\lambda + \mu^2)}{2\mu^2 y_i} \right) \\
&= \frac{n}{2} \log \lambda - \frac{\lambda}{2\mu^2} \sum_{i=1}^n y_i + \frac{n\lambda}{\mu} - \frac{\lambda}{2} \sum_{i=1}^n \frac{1}{y_i} - \frac{1}{2} \sum_{i=1}^n \log 2\pi y_i^3
\end{aligned}$$

Compute the MLE by solving:

$$\begin{cases} \frac{\partial l}{\partial \mu} = 0 \\ \frac{\partial l}{\partial \lambda} = 0 \end{cases}$$

The MLE of μ and λ is:

$$\begin{cases} \hat{\mu} = \frac{\sum_{i=1}^n y_i}{n} \\ \hat{\lambda} = \left(\frac{\sum_{i=1}^n \frac{1}{y_i}}{n} - \frac{n}{\sum_{i=1}^n y_i} \right)^{-1} \end{cases}$$

The Fisher's Information Matrix is:

$$I(\mu, \lambda) = \begin{pmatrix} \frac{n\lambda}{\mu^3} & 0 \\ 0 & \frac{n}{2\lambda^2} \end{pmatrix}$$

By Wald's theory, the 95% Confidence Interval of μ and λ is:

$$\begin{cases} \mu \in \hat{\mu} \pm 1.96 \times \sqrt{\frac{\hat{\mu}^3}{n\hat{\lambda}}} \\ \lambda \in \hat{\lambda} \pm 1.96 \times \sqrt{\frac{2\hat{\lambda}^2}{n}} \end{cases}$$

2.2 Method 2

For λ , using the profile likelihood thought, define: $l_1(\lambda) = l(\hat{\mu}, \lambda)$. By Wilk's theorem, we have:

$$-2(l_1(\lambda_0) - l_1(\hat{\lambda})) \sim \chi_1^2, \text{ when } n \rightarrow \infty.$$

Thus, by the inversion of likelihood ratio statistics, the 95% Confidence Interval of λ is:

$$\{\lambda \in R^+ : -2(l_1(\lambda) - l_1(\hat{\lambda})) \leq \chi_1^2(0.95)\}$$

Define:

$$T_\lambda = -2(l_1(\lambda) - l_1(\hat{\lambda}))$$

Then:

$$\begin{aligned} T_\lambda &= -2[(\frac{n}{2}\log\lambda + \frac{n\lambda}{2\hat{\mu}} - \frac{\lambda}{2}\sum_{i=1}^n \frac{1}{y_i}) - (\frac{n}{2}\log\hat{\lambda} + \frac{n\hat{\lambda}}{2\hat{\mu}} - \frac{\hat{\lambda}}{2}\sum_{i=1}^n \frac{1}{y_i})] \\ &= (\sum_{i=1}^n \frac{1}{y_i} - \frac{n}{\hat{\mu}})(\lambda - \hat{\lambda}) - n\log \frac{\lambda}{\hat{\lambda}} \end{aligned}$$

For μ , notice that:

$$\frac{\partial l}{\partial \lambda} = 0 \rightarrow \lambda = (\frac{\hat{\mu}}{\mu^2} - \frac{2}{\mu} + \frac{\sum_{i=1}^n \frac{1}{y_i}}{n})^{-1}$$

So, we define:

$$\tilde{\lambda} \equiv \tilde{\lambda}(\mu) = (\frac{\hat{\mu}}{\mu^2} - \frac{2}{\mu} + \frac{\sum_{i=1}^n \frac{1}{y_i}}{n})^{-1}$$

Then define: $l_2(\mu) = l(\mu, \tilde{\lambda})$, similarly, the 95% Confidence Interval of μ is:

$$\{\mu \in R^+ : -2(l_2(\mu) - l_2(\hat{\mu})) \leq \chi_1^2(0.95)\}$$

Define:

$$T_\mu = -2(l_2(\mu) - l_2(\hat{\mu}))$$

Then:

$$T_\mu = -2[(\frac{n}{2}\log\tilde{\lambda} - \frac{\tilde{\lambda}}{2\mu^2}\sum_{i=1}^n y_i + \frac{n\tilde{\lambda}}{\mu} - \frac{\tilde{\lambda}}{2}\sum_{i=1}^n \frac{1}{y_i}) - (\frac{n}{2}\log\hat{\lambda} + \frac{n\hat{\lambda}}{2\hat{\mu}} - \frac{\hat{\lambda}}{2}\sum_{i=1}^n \frac{1}{y_i})]$$

To draw a conclusion, the 95% Confidence Interval of μ and λ is:

$$\begin{cases} \mu \in \{\mu \in R^+ : T_\mu \leq \chi_1^2(0.95)\} \\ \lambda \in \{\lambda \in R^+ : T_\lambda \leq \chi_1^2(0.95)\} \end{cases}$$

where T_μ and T_λ are defined above.

These intervals were computed by evaluating the normed profile likelihood for a sequence of values between lb and ub with spacing of 0.001, where lb and ub are functions with respect to the sample size n and the confidence interval width computed by the first method.

For example, for the CI of μ , the CI computed by the first method is: $\hat{\mu} \pm 1.96 \times \sqrt{\frac{\hat{\mu}^3}{n\hat{\lambda}}}$. Then we can let:

$$lb = \hat{\mu} - 1.96 \times \sqrt{\frac{\hat{\mu}^3}{n\hat{\lambda}}} \times C_\mu(n)$$

$$ub = \hat{\mu} + 1.96 \times \sqrt{\frac{\hat{\mu}^3}{n\hat{\lambda}}} \times C_{\mu}(n)$$

Where $C_{\mu}(n)$ is a function of n . By simulation, I decide to use:

$$C_{\mu}(n) = \begin{cases} 15, & n \leq 20 \\ 10, & 20 < n \leq 50 \\ 5, & 50 < n \leq 100 \\ 3, & 100 < n \leq 200 \\ 2, & n > 200 \end{cases} \quad ; \quad C_{\lambda}(n) = \begin{cases} 5, & n \leq 20 \\ 3, & 20 < n \leq 50 \\ 2, & n > 50 \end{cases}$$

Notice that this bound-decide method will get a narrower CI width of μ compared with the theoretical one when $n = 10$. However, since the true coverage rate is very close to 0.95, the narrower CI width will lead to a better performance. On the other hand, our bound-decide method would be much faster when n become larger than the bound-decide method that is unified for different sample size n .

2.3 Method 3

By Maximum likelihood and parametric bootstrap using the comparison function: $h(\theta, \hat{\theta}) = n^{1/2}(\hat{\theta} - \theta)$. We can write down the algorithm as follow:

- Step1: Calculate $\hat{\mu}$ and $\hat{\lambda}$ (the MLE of μ and λ using the original data).
- Step2: Generate observations $\mathbf{y}_m^* = (y_{1m}^*, \dots, y_{nm}^*)$, $m = 1, \dots, N$, from the fitted model $Inv.Gauss(\hat{\mu}, \hat{\lambda})$. (N will be decided later).
- Step3: Calculate μ_m^* and λ_m^* (the MLE of μ and λ using the m th generated data).
- Step4: Calculate $\mu_{[(N+1)(0.025)]}^*$, $\mu_{[(N+1)(0.975)]}^*$, $\lambda_{[(N+1)(0.025)]}^*$, $\lambda_{[(N+1)(0.975)]}^*$ (e.g. $\mu_{[(N+1)(0.025)]}^*$ is the $[(N+1)(0.025)]$ th order statistics of $\{\mu_m^*\}_{m=1}^N$).

Then the 95% Confidence Interval of μ and λ is:

$$\begin{cases} \mu \in (2\hat{\mu} - \mu_{[(N+1)(0.975)]}^*, 2\hat{\mu} - \mu_{[(N+1)(0.025)]}^*) \\ \lambda \in (2\hat{\lambda} - \lambda_{[(N+1)(0.975)]}^*, 2\hat{\lambda} - \lambda_{[(N+1)(0.025)]}^*). \end{cases}$$

2.4 Method 4

By Maximum likelihood and parametric bootstrap using the comparison function: $h(\theta, \hat{\theta}) = n^{1/2}(\frac{\theta}{\hat{\theta}} - 1)$. The algorithm is the same as in Method 3.

The 95% Confidence Interval of μ and λ is:

$$\begin{cases} \mu \in (\frac{\hat{\mu}^2}{\mu_{[(N+1)(0.975)]}^*}, \frac{\hat{\mu}^2}{\mu_{[(N+1)(0.025)]}^*}) \\ \lambda \in (\frac{\hat{\lambda}^2}{\lambda_{[(N+1)(0.975)]}^*}, \frac{\hat{\lambda}^2}{\lambda_{[(N+1)(0.025)]}^*}). \end{cases}$$

2.5 Method 5

By Maximum likelihood and parametric bootstrap using the comparison function: $h(\theta, \hat{\theta}) = \frac{\hat{\theta} - \theta}{V(\hat{\theta})^{1/2}}$. Only step 4 of the algorithm will change. We now need to calculate $(\frac{\theta^* - \hat{\theta}}{V(\theta^*)^{1/2}})_{[(N+1)(0.025)]}$ and $(\frac{\theta^* - \hat{\theta}}{V(\theta^*)^{1/2}})_{[(N+1)(0.975)]}$ instead of $\theta_{[(N+1)(0.025)]}^*$ and $\theta_{[(N+1)(0.975)]}^*$, where θ denote μ or λ .

Thus, the 95% Confidence Interval of μ and λ is:

$$\begin{cases} \mu \in (\hat{\mu} - [V(\hat{\mu})]^{1/2}(\frac{\mu^* - \hat{\mu}}{V(\mu^*)^{1/2}})_{[(N+1)(0.975)]}, \hat{\mu} - [V(\hat{\mu})]^{1/2}(\frac{\mu^* - \hat{\mu}}{V(\mu^*)^{1/2}})_{[(N+1)(0.025)]}) \\ \lambda \in (\hat{\lambda} - [V(\hat{\lambda})]^{1/2}(\frac{\lambda^* - \hat{\lambda}}{V(\lambda^*)^{1/2}})_{[(N+1)(0.975)]}, \hat{\lambda} - [V(\hat{\lambda})]^{1/2}(\frac{\lambda^* - \hat{\lambda}}{V(\lambda^*)^{1/2}})_{[(N+1)(0.025)]}). \end{cases}$$

Where:

$$V(\tilde{\mu}) = \frac{\tilde{\mu}^3}{n\tilde{\lambda}} ; \quad V(\tilde{\lambda}) = \frac{2\tilde{\lambda}^2}{n}.$$

($\tilde{\mu}$ denotes $\hat{\mu}$ or μ^* , $\tilde{\lambda}$ denotes $\hat{\lambda}$ or λ^*).

Note that:

$$\begin{aligned} & \hat{\lambda} - [V(\hat{\lambda})]^{1/2}(\frac{\lambda^* - \hat{\lambda}}{V(\lambda^*)^{1/2}})_{[(N+1)(0.975)]} \\ &= \hat{\lambda} - \sqrt{\frac{2}{n}}\hat{\lambda}(\frac{\lambda^* - \hat{\lambda}}{\sqrt{\frac{2}{n}}\lambda^*})_{[(N+1)(0.975)]} \\ &= \hat{\lambda} - \hat{\lambda}(1 - \frac{\hat{\lambda}}{\lambda^*})_{[(N+1)(0.975)]} \\ &= \hat{\lambda} - \hat{\lambda}(1 - \frac{\hat{\lambda}}{\lambda_{[(N+1)(0.975)]}^*}) \\ &= \frac{\hat{\lambda}^2}{\lambda_{[(N+1)(0.975)]}^*}. \end{aligned}$$

Similarly:

$$\begin{aligned} & \hat{\lambda} - [V(\hat{\lambda})]^{1/2} \left(\frac{\lambda^* - \hat{\lambda}}{V(\lambda^*)^{1/2}} \right)_{[(N+1)(0.025)]} \\ &= \frac{\hat{\lambda}^2}{\lambda_{[(N+1)(0.025)]}^*}. \end{aligned}$$

Hence, method 4 and method 5 will have the same Confidence Interval for λ given the same seed of randomness.

2.6 Method 6

By Method of Moment estimation and parametric bootstrap using the comparison function: $h(\theta, \hat{\theta}) = n^{1/2}(\hat{\theta} - \theta)$. The algorithm will be similar, the only difference is to replace all the $\hat{\mu}$ and $\hat{\lambda}$ by μ_{MME} and λ_{MME} .

$$\begin{cases} \mu_{MME} = \frac{1}{n} \sum_{i=1}^n Y_i \\ \lambda_{MME} = \frac{\mu_{MME}^3}{\frac{1}{n} \sum_{i=1}^n Y_i^2 - \mu_{MME}^2}. \end{cases}$$

Thus, the 95% Confidence Interval of μ and λ is:

$$\begin{cases} \mu \in (2\mu_{MME} - \mu_{MME,[(N+1)(0.975)]}^*, 2\mu_{MME} - \mu_{MME,[(N+1)(0.025)]}^*) \\ \lambda \in (2\lambda_{MME} - \lambda_{MME,[(N+1)(0.975)]}^*, 2\lambda_{MME} - \lambda_{MME,[(N+1)(0.025)]}^*). \end{cases}$$

2.7 R Functions and Packages Used

- Generate Inverse Gaussian distribution: `statmod::rinvgauss(x, mean, shape)`.
- All the other R functions are written by me (In Appendix).

2.8 Choose Bootstrap — N

There are too many combinations of experiments: $\lambda \in \{2, 4, 8, 12\}$, $n \in \{10, 25, 50, 100, 500\}$, $method \in \{3, 4, 5, 6\}$. It's not efficient to find the best N for all these combinations. So, I use the thought of "Latin Square Design" to compute the confidence interval width of the following combinations when $N \in \{50, 100, 200, 300, \dots, 10000\}$. The results are in figure[1] and figure[2] in the Appendix.

| | method3 | method4 | method5 | method6 |
|----------------|---------|---------|---------|---------|
| $\lambda = 2$ | n=10 | n=50 | n=100 | n=500 |
| $\lambda = 4$ | n=50 | n=100 | n=500 | n=10 |
| $\lambda = 8$ | n=100 | n=500 | n=10 | n=50 |
| $\lambda = 12$ | n=500 | n=10 | n=50 | n=100 |

The figure[1] and figure[2] show that the CI widths converge for most of the combinations when $N=2000$. Thus, we pick the bootstrap N to be 2000.

2.9 Choose Monte Carlo — M

In order to make the comparison between different combinations of experiments, we need to find one unified M . In this case, I compute:

- the coverage rate of μ : given $n = 100$ and $method = 1$ for the four different λ (figure[3]);
- the coverage rate of μ : given $\lambda = 4$ and $method = 1$ for the five different n (figure[4]);
- the coverage rate of μ : given $\lambda = 4$ and $n = 10$ for the six different methods (figure[5]);

I also compute the coverage rate of λ in the same conditions, the results are similar. By checking the figures, I find the coverage rates become stable when $\sqrt{M} = 100$, which means $M = 10000$. So, I choose M to be 10000 for all the cases.

2.10 Monte Carlo Study Algorithm

————to Calculate the 3 Criteria of Confidence Intervals Computed by each Methods. (when $\mu_0 = 5$, $n_0 \in \{10, 25, 50, 100, 500\}$, $\lambda_0 \in \{2, 4, 8, 12\}$, $method \in \{1, 2, 3, 4, 5, 6\}$).

```

for ( $i \in 1 : 10000$ ) {
  ♦ Generate  $n_0$  Inverse Gaussian r.v. of  $\mu = \mu_0$  and  $\lambda = \lambda_0$ 
  by R function statmod::rinvgauss( $n$ ,  $\mu$ ,  $\lambda$ ).
  ♦ Compute 95% CI:  $I_\mu^{(i)}$ ,  $I_\lambda^{(i)}$  using the algorithm in Section 2.1—2.6
  and the R function in Appendix 5.1. ( $N = 2000$  when using method 3—6)
}

```

◆ Compute the actual coverage rate by:

$$cv_\mu = \frac{1}{10000} \sum_{i=1}^{10000} I(\mu_0 \in I_\mu^{(i)}); \quad cv_\lambda = \frac{1}{10000} \sum_{i=1}^{10000} I(\lambda_0 \in I_\lambda^{(i)}).$$

◆ Compute the median width simply by the R median function.

◆ Compute the out of bound probability by:

$$P_\mu = \frac{1}{10000} \sum_{i=1}^{10000} I(\text{lower bound of } I_\mu^{(i)} \leq 0); \quad P_\lambda = \frac{1}{10000} \sum_{i=1}^{10000} I(\text{lower bound of } I_\lambda^{(i)} \leq 0).$$

The code can be found in Appendix 5.2.

3 Results

The **coverage rate**, **median CI width** and the **out of bound probability** of all the combinations of experiments ($\lambda \in \{2, 4, 8, 12\}$, $n \in \{10, 25, 50, 100, 500\}$, and $method \in \{1, 2, 3, 4, 5, 6\}$) are shown in table[1] –table[6] in the Appendix. There are also four figures about the coverage rate and median CI width for μ and λ (figure[6] — figure[9]).

4 Discussion and Conclusions

4.1 Performance of Methods When n Different

For confidence intervals of μ :

When the sample size n small, the actual coverage rate of method 1, 3, and 6 are small, (i.e. when $n = 10$, $\lambda = 2$: $cv1 = 0.7941$, $cv3 = 0.7339$, $cv6 = 0.7042$) while the coverage rate of method 2 are very close to 0.95 (i.e. when $n = 10$, $\lambda = 2$: $cv2 = 0.9324$). However, the median width of method 2 is too large compared to the other methods (i.e. when $n = 10$, $\lambda = 2$: $med.wid2 = 45.7270$, $med.wid1 = 7.4000$). I guess the coverage rate of method 2 is close to 0.95 because the CI computed by method 2 is too wide so that the true μ falls into most of the CI, while the coverage rate of method 1, 3, and 6 is small means the CI computed by these methods are not wide enough to capture enough information. So, by the coverage rate and median width trade-off, I prefer to pick

method 4 and method 5 (i.e. when $n = 10$, $\lambda = 2$: $cv4 = 0.8583$, $cv5 = 0.8860$, $med.wid4 = 8.4691$, $med.wid5 = 13.4934$). Besides, the out of bound probability of method 4 and method 5 is always 0, which also means method 4 and method 5 are better.

When the sample size n large, the differences between the different methods become less. when $n = 500$, the differences of the first five methods are nearly negligible (The coverage rate of method 6 when $n = 500$ is a little small than the first five methods) and all of the coverage rate of the six methods are approaching to 0.95. So, the first five methods (even method 6) are applicable. In this case, I prefer method 1 because it's the fastest method.

For confidence intervals of λ :

Firstly, we should notice a fact that the confidence interval computed by method 4 and method 5 are the same give the same seed. (It has been discussed in section 2.5.) Also, the results of method 1 and method 2 perform very similar (nearly the same) by checking table [2], [4], and [6].

When the sample size n small, method 1, 2, 4, and 5 have the similar median CI width, method 3 and method 6 have a much larger median CI width than the front four methods. (i.e. when $n = 10$, $\lambda = 2$: $med.wid1 = 4.1874$, $med.wid2 = 4.276$, $med.wid4\&5 = 3.9031$, $med.wid3 = 7.5531$, $med.wid6 = 17.6629$). On the other hand, the out of bound probability of method 3 and method 6 are very large while the other four methods always have 0 probability (i.e. when $n = 10$, $\lambda = 2$: $P(out\ of\ bound\ of\ method\ 3) = 1$, $P(out\ of\ bound\ of\ method\ 6) = 1$). In the table [2], I notice that the coverage rate of method 4 and 5 are closer to 0.95 than method 1 and 2, while the confidence intervals are narrower. In this case, I recommend to choose method 4 or method 5. (Method 4 is a little faster).

When the sample size n large, the first five methods outperform the method 6 (i.e. when $n = 500$, $\lambda = 2$: the median width of the first fives methods are about 0.497, while $med.wid6 = 1.3649$). Besides, the out of bound probability of method 6 when $n = 100$, $\lambda = 2$ is still big (i.e. $P=0.1697$). As a result, first five methods are applicable. In this case, I prefer method 1 because it's the fastest.

4.2 Performance of Methods When λ Different

The change of λ will not affect the performance of the confidence interval for λ , however, it does have influence on the confidence interval for μ . It can be found in table [1], [3] and [5] that the

performance of the confidence interval for μ becomes better when λ increase (i.e. the coverage rate is closer to 0.95, the out of bound probability is smaller).

By checking table [1], I found the coverage rate of method 2 and method 5 for μ is closer to 0.95 than the other four methods when λ is large and n is small (e.g. when $n = 10$, $\lambda = 12$: $cv2 = 0.9282$, $cv5 = 0.9376$, the other methods are all less than 0.9). At the same time, the median CI width of these two methods are not too large compared to the other four methods, the out of bound probability is 0 for these two methods. So, I would like to pick method 2 or method 5 for conducting CI of μ when λ large and n small.

The method 4 is also applicable since the out of bound probability is 0 and the coverage rate is higher than method 1, 3 and 6, though it's not as good as method 2 and method 5.

4.3 Checking the figure [6]—[9]

The discussion above is based on the table [1]—[6], we can get the same conclusion by check the figure [6]—[9], which are more clear than the tables.

For the CI for μ , note that the coverage rate given by the method 2, 4, and 5 converge faster than method 1, 3 and 6. However, the median width computed by method 2 is much larger than the other ones when λ is small. Thus, the method 4 and 5 is preferable. Method 2 is applicable when λ is large. When n is large, the coverage rate and median width for all of the methods become very close.

For the CI for λ , note that the median width for method 1, 2, 4, and 5 are very close and they are smaller than the median width for method 3 and 6. On the other hand, the coverage rate computed by method 4 and 5 are closest to 0.95 among all methods. Thus, method 4 and 5 are better. When n is large, the coverage rate and median width for the first 5 methods become very close.

4.4 Conclusions

For the confidence interval of μ , the following methods have good performance (* The method in bracket means: applicable but not as good as the other ones.):

| | n small | n large |
|-----------------|----------------|----------------------|
| λ small | method 4 5 | method 1 2 3 4 5 (6) |
| λ large | method 2 (4) 5 | method 1 2 3 4 5 (6) |

For the confidence interval of λ , the following methods have good performance:

| n small | n large |
|------------|------------------|
| method 4 5 | method 1 2 3 4 5 |

Over all, my favorite method is method 5 since it has a good performance in any condition. Method 4 is also a reasonable choice. However, I recommend to use method 1 when the sample size is large since it's much faster than the other ones.

5 Appendix

5.1 R Functions of Method 1-6

```
##### Method1 MLE & Wald Theory #####
mle_mu <- function(x) return( mean(x) )
mle_la <- function(x) return( 1 / ( mean(1/x) - 1/mean(x) ) )

ci_mu_1 <- function(x){
  sigma <- sqrt( (mle_mu(x))^3/(length(x)*mle_la(x)) )
  lb <- mle_mu(x) - 1.96*sigma
  ub <- mle_mu(x) + 1.96*sigma
  return(c(lb,ub))
}

ci_la_1 <- function(x){
  sigma <- sqrt( (2*(mle_la(x))^2)/length(x) )
  lb <- mle_la(x) - 1.96*sigma
  ub <- mle_la(x) + 1.96*sigma
  return(c(lb,ub))
}

##### Method2 MLE & Inverse LRT #####

## lambda
LR_la <- function(la,x){
  n <- length(x)
  return((sum(1/x) - (n/mle_mu(x)) )*(la - mle_la(x)) - n*log((la)/mle_la(x)))
}

ci_la_2 <- function(x){
  me <- mle_la(x)
  ci <- ci_la_1(x)
```

```

n <- length(x)
if (n <= 20) b <- 5*(me-ci[1])
if (n>20 & n <=50) b <- 3*(me-ci[1])
if (n>50) b <- 2*(me-ci[1])
st <- max(me-b,0.001)
ed <- me+b
XX <- seq(st,ed,0.001)
LR <- LR_la( XX, x )
C <- qchisq(0.95,1)
index1 <- which(LR < C)[1]
index2 <- which(LR < C)[length(which(LR < C))]
return(c(XX[index1],XX[index2]))
}

```

```

##mu
til_la <- function(mu,x){
  Y <- (mle_mu(x))/(mu^2) - (2/mu) + mean(1/x)
  return(1/Y)
}

```

```

LR_mu <- function(mu,x){
  n <- length(x)
  L1 <- (n/2)*log(til_la(mu,x)) - ((til_la(mu,x))/(2*mu^2))*sum(x) +
    (n*til_la(mu,x))/mu - (til_la(mu,x)/2)*sum(1/x)
  L2 <- (n/2)*log(mle_la(x)) + (n*mle_la(x))/(2*mle_mu(x)) -
    (mle_la(x)/2)*sum(1/x)
  return((-2)*(L1-L2))
}

```

```

ci_mu_2 <- function(x){
  me <- mle_mu(x)
  ci <- ci_mu_1(x)
  n <- length(x)
  if (n <= 20) b <- 15*(me-ci[1])

```

```

if (n>20 & n <=50) b <- 10*(me-ci[1])
if (n>50 & n<=100) b <- 5*(me-ci[1])
if (n>100 & n<=200) b <- 3*(me-ci[1])
if(n>200) b <- 2*(me-ci[1])
st <- max(me-b,0.00001)
ed <- me+b
XX <- seq(st,ed,0.001)
LR <- LR_mu( XX, x )
C <- qchisq(0.95,1)
index1 <- which(LR < C)[1]
index2 <- which(LR < C)[length(which(LR < C))]
return(c(XX[index1],XX[index2]))
}

```

Method3 MLE & Parametric bootstrap 1

```

ci_boot_mu_1 <- function(x,N){
  library(statmod)
  mu <- mle_mu(x)
  la <- mle_la(x)
  n <- length(x)
  X <- matrix(rinvgauss(n*N,mu,la),ncol = N)
  mu1 <- apply(X,2,mle_mu)
  mu1 <- sort(mu1)
  qmu1 <- mu1[ floor( (N+1)*(1-(0.05/2))) ]
  qmu2 <- mu1[ floor( (N+1)*(0.05/2) ) ]
  lb <- 2*mu-qmu1
  ub <- 2*mu-qmu2
  return(c(lb,ub))
}

```

```

ci_boot_la_1 <- function(x,N){
  library(statmod)

```

```

mu <- mle_mu(x)
la <- mle_la(x)
n <- length(x)
X <- matrix(rinvgauss(n*N,mu,la),ncol = N)
la1 <- apply(X,2,mle_la)
la1 <- sort(la1)
qla1 <- la1[ floor( (N+1)*(1-(0.05/2))) ]
qla2 <- la1[ floor( (N+1)*(0.05/2) ) ]
lb <- 2*la-qla1
ub <- 2*la-qla2
return(c(lb,ub))
}

```

Method4 MLE & Parametric bootstrap 2

```

ci_boot_mu_2 <- function(x,N){
  library(statmod)
  mu <- mle_mu(x)
  la <- mle_la(x)
  n <- length(x)
  X <- matrix(rinvgauss(n*N,mu,la),ncol = N)
  mu1 <- apply(X,2,mle_mu)
  mu1 <- sort(mu1)
  qmu1 <- mu1[ floor( (N+1)*(1-(0.05/2))) ]
  qmu2 <- mu1[ floor( (N+1)*(0.05/2) ) ]
  lb <- mu^2/qmu1
  ub <- mu^2/qmu2
  return(c(lb,ub))
}

```

```

ci_boot_la_2 <- function(x,N){
  library(statmod)
  mu <- mle_mu(x)
  la <- mle_la(x)

```

```

n <- length(x)
X <- matrix(rinvgauss(n*N,mu,la),ncol = N)
la1 <- apply(X,2,mle_la)
la1 <- sort(la1)
qla1 <- la1[ floor( (N+1)*(1-(0.05/2))) ]
qla2 <- la1[ floor( (N+1)*(0.05/2) ) ]
lb <- la^2/qla1
ub <- la^2/qla2
return(c(lb,ub))
}

```

Method5 MLE & Parametric bootstrap 3

```

Vmu <- function(x){
  n <- length(x)
  la <- mle_la(x)
  mu <- mle_mu(x)
  return((mu^3)/(n*la))
}

```

```

Vla <- function(x){
  n <- length(x)
  la <- mle_la(x)
  return((2*la^2)/n)
}

```

```

ci_boot_mu_3 <- function(x,N){
  library(statmod)
  mu <- mle_mu(x)
  la <- mle_la(x)
  n <- length(x)
  X <- matrix(rinvgauss(n*N,mu,la),ncol = N)
  mu1 <- apply(X,2,mle_mu) - mu
  V1 <- apply(X,2,Vmu)
}

```



```

mu11 <- sort(mu1/sqrt(V1))
qmu1 <- mu11[ floor( (N+1)*(1-(0.05/2))) ]
qmu2 <- mu11[ floor( (N+1)*(0.05/2) ) ]
lb <- mu-sqrt(Vmu(x))*qmu1
ub <- mu-sqrt(Vmu(x))*qmu2
return(c(lb,ub))
}

```

```

ci_boot_la_3 <- function(x,N){
  library(statmod)
  mu <- mle_mu(x)
  la <- mle_la(x)
  n <- length(x)
  X <- matrix(rinvgauss(n*N,mu,la),ncol = N)
  la1 <- apply(X,2,mle_la) - la
  V1 <- apply(X,2,Vla)
  la11 <- sort(la1/sqrt(V1))
  qla1 <- la11[ floor( (N+1)*(1-(0.05/2))) ]
  qla2 <- la11[ floor( (N+1)*(0.05/2) ) ]
  lb <- la-sqrt(Vla(x))*qla1
  ub <- la-sqrt(Vla(x))*qla2
  return(c(lb,ub))
}

```

Method6 MME & Parametric bootstrap 4

```

mme_mu <- function(x) return(mean(x))
mme_la <- function(x) return( ((mean(x))^3)/(mean(x^2)-(mean(x))^2) )

```

```

ci_boot_mu_4 <- function(x,N){
  library(statmod)
  mu <- mme_mu(x)
  la <- mme_la(x)

```

```

n <- length(x)
X <- matrix(rinvgauss(n*N,mu,la),ncol = N)
mu1 <- apply(X,2,mme_mu)
mu1 <- sort(mu1)
qmu1 <- mu1[ floor( (N+1)*(1-(0.05/2))) ]
qmu2 <- mu1[ floor( (N+1)*(0.05/2) ) ]
lb <- 2*mu-qmu1
ub <- 2*mu-qmu2
return(c(lb,ub))
}

```

```

ci_boot_la_4 <- function(x,N){
  library(statmod)
  mu <- mme_mu(x)
  la <- mme_la(x)
  n <- length(x)
  X <- matrix(rinvgauss(n*N,mu,la),ncol = N)
  la1 <- apply(X,2,mme_la)
  la1 <- sort(la1)
  qla1 <- la1[ floor( (N+1)*(1-(0.05/2))) ]
  qla2 <- la1[ floor( (N+1)*(0.05/2) ) ]
  lb <- 2*la-qla1
  ub <- 2*la-qla2
  return(c(lb,ub))
}

```

5.2 R Functions of 3 criteria

```

criteria <- function(M,n = 10, mu = 5,lambda = 2,

```

```

para = c("mu","lambda"), method = c(1,2,3,4,5,6)){
  if (method == 1){
    ci_mu <- function(x) ci_mu_1(x)
    ci_la <- function(x) ci_la_1(x)
  }
  if (method == 2){
    ci_mu <- function(x) ci_mu_2(x)
    ci_la <- function(x) ci_la_2(x)
  }
  if (method == 3){
    ci_mu <- function(x) ci_boot_mu_1(x,2000)
    ci_la <- function(x) ci_boot_la_1(x,2000)
  }
  if (method == 4){
    ci_mu <- function(x) ci_boot_mu_2(x,2000)
    ci_la <- function(x) ci_boot_la_2(x,2000)
  }
  if (method == 5){
    ci_mu <- function(x) ci_boot_mu_3(x,2000)
    ci_la <- function(x) ci_boot_la_3(x,2000)
  }
  if (method == 6){
    ci_mu <- function(x) ci_boot_mu_4(x,2000)
    ci_la <- function(x) ci_boot_la_4(x,2000)
  }
  set.seed(520520)
  library(statmod)
  if (para == "mu"){
    IG <- matrix(rinvgauss(M*n,mu,lambda), ncol=M)
    CImu <- apply(IG,2,ci_mu)
    ID <- (CImu[1,]<mu)&(CImu[2,]>mu)
    CI <- t(CImu)
  }
}

```

```

if (para == "lambda"){
  IG <- matrix(rinvgauss(M*n,mu,lambda), ncol=M)
  CI1a <- apply(IG,2,ci_la)
  ID <- (CI1a[1,<lambda)&(CI1a[2,>lambda)
  CI <- t(CI1a)
}

rate <- mean(ID)
width <- CI[,2]-CI[,1]
med_width <- median(width)
prob <- mean(CI[,1]<0)
result <- list(rate = rate,medianWidth = med_width, Prob = prob)
return(result)
}

```

5.3 Some Tables and Figures

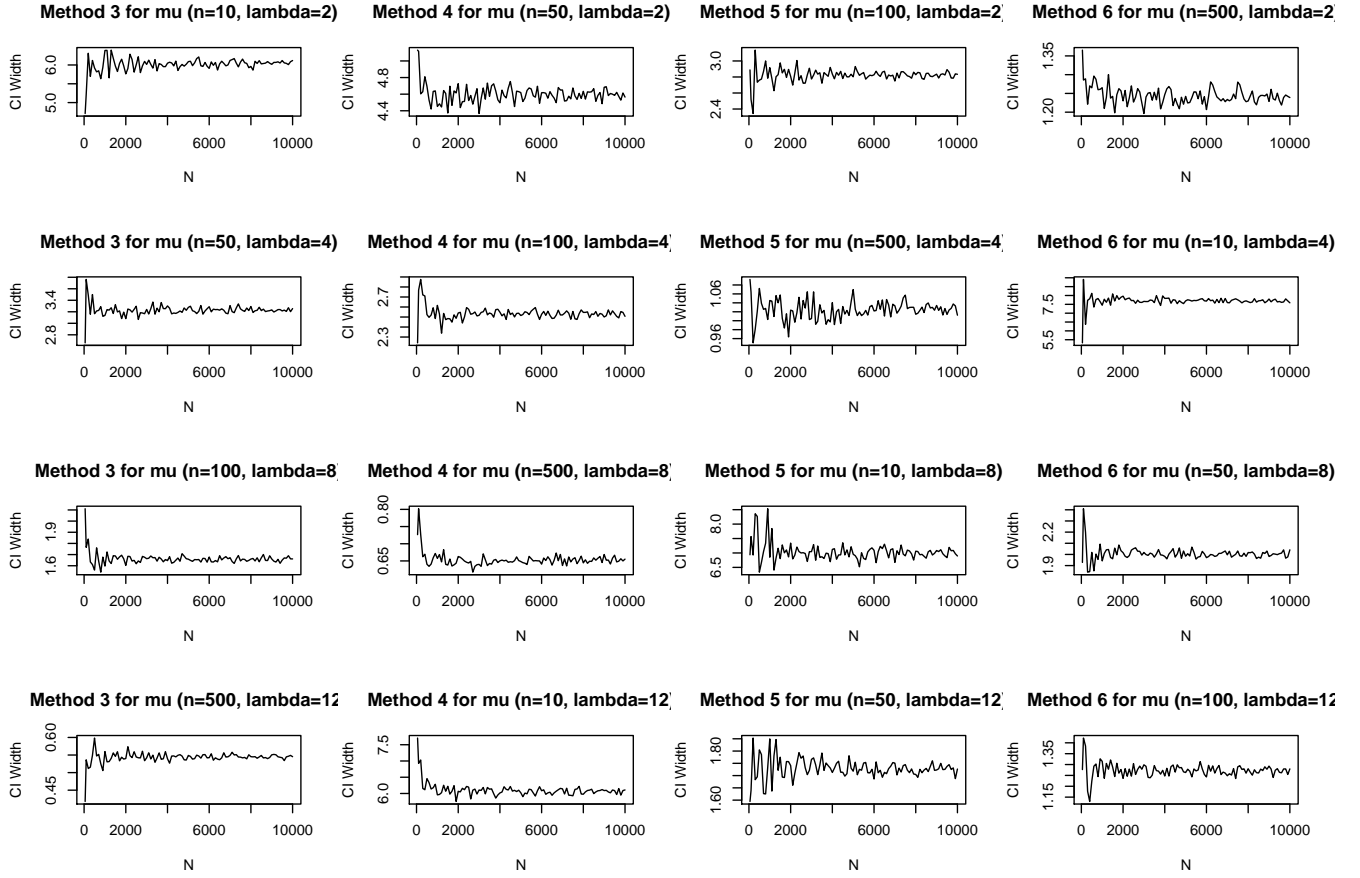


Figure 1: Choose bootstrap N using the CI width of μ

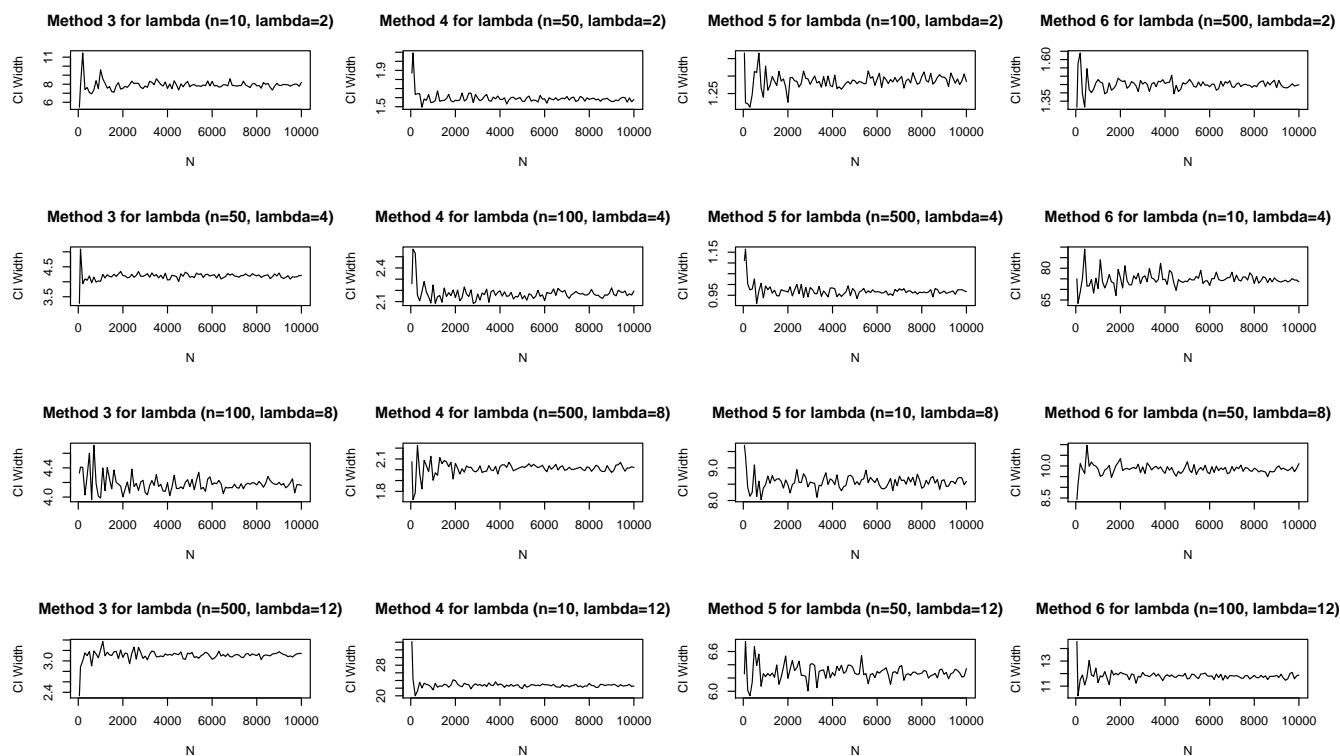


Figure 2: Choose bootstrap N using the CI width of λ

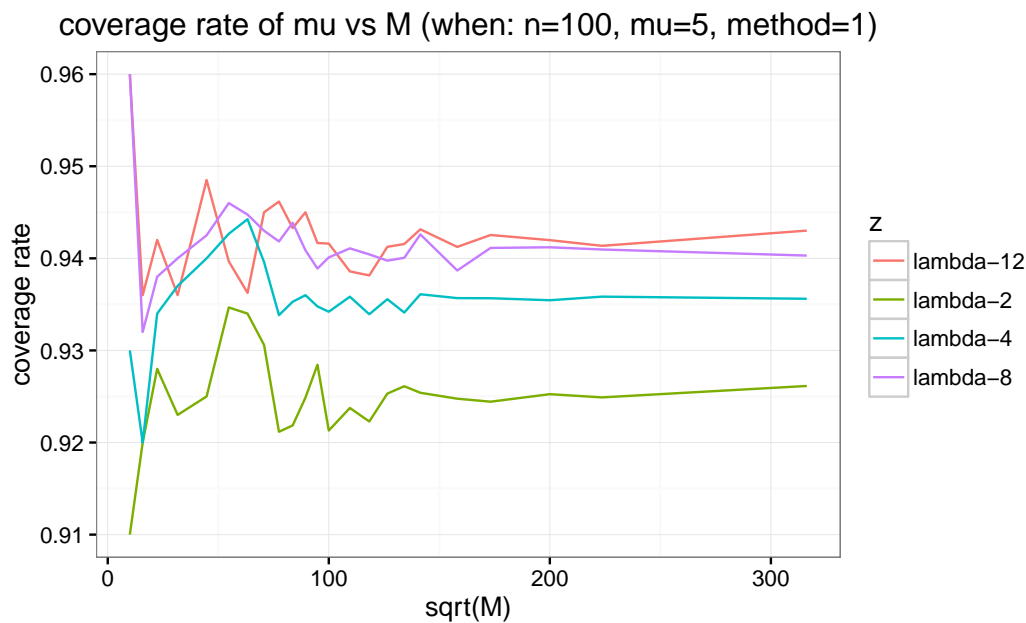


Figure 3

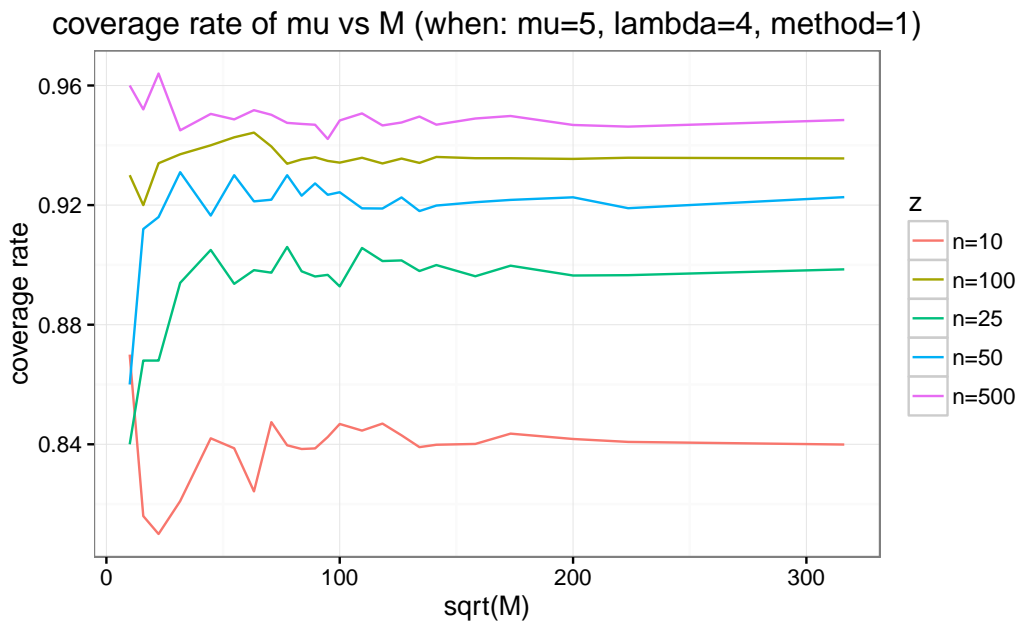


Figure 4

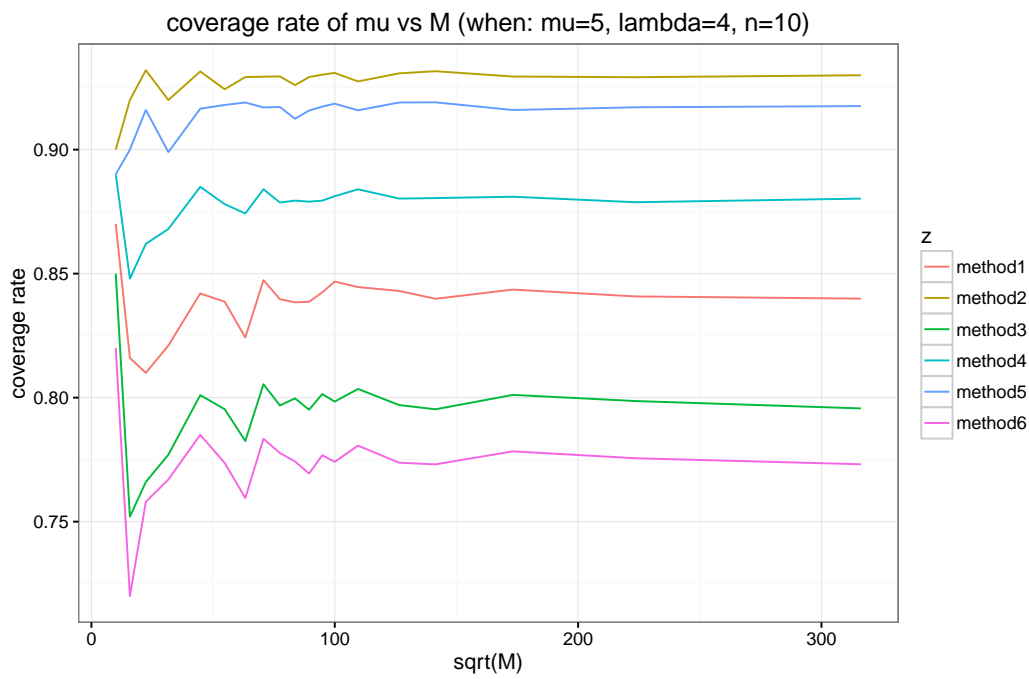


Figure 5

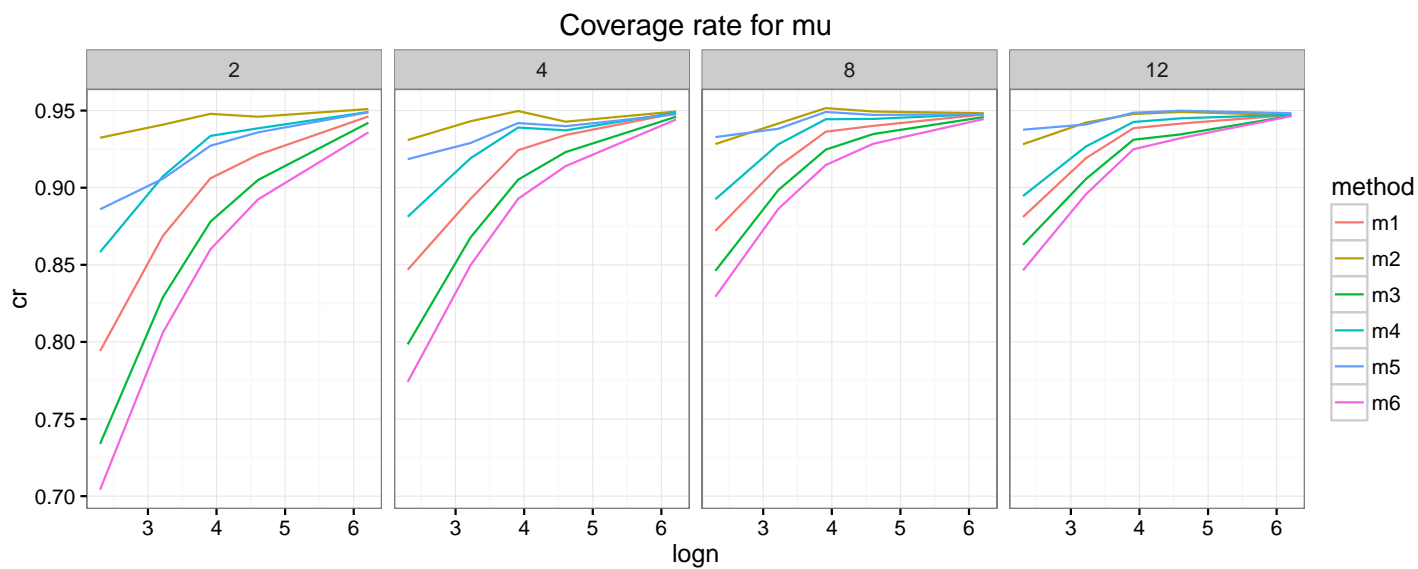


Figure 6

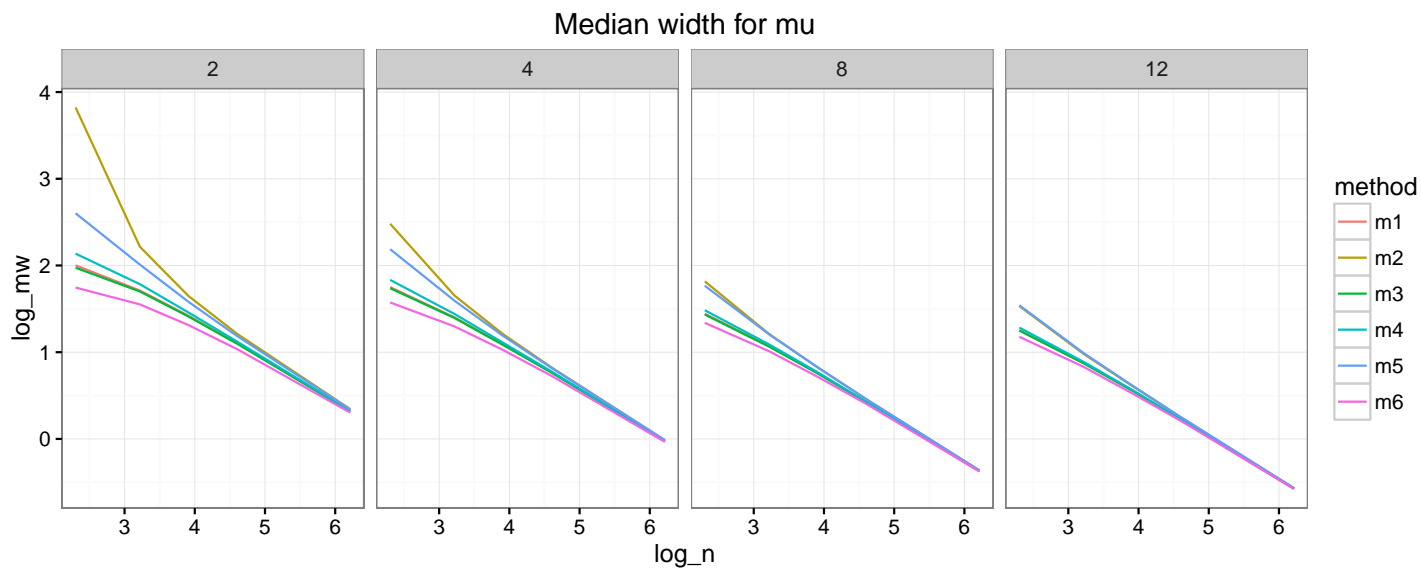


Figure 7

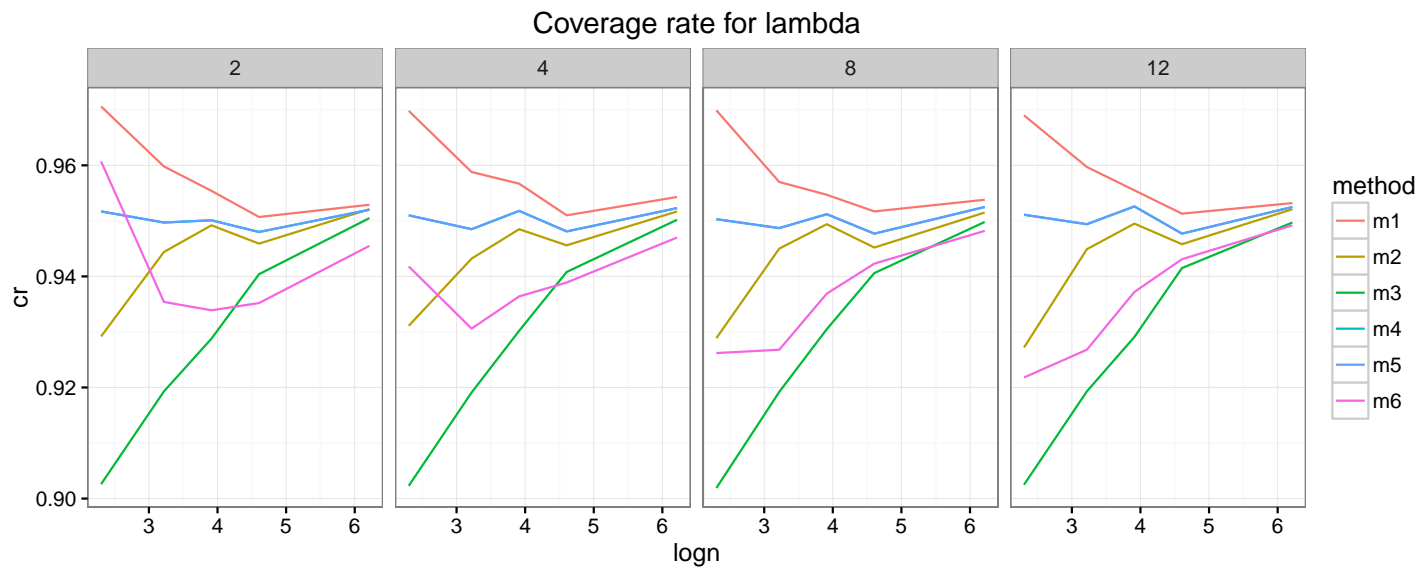


Figure 8

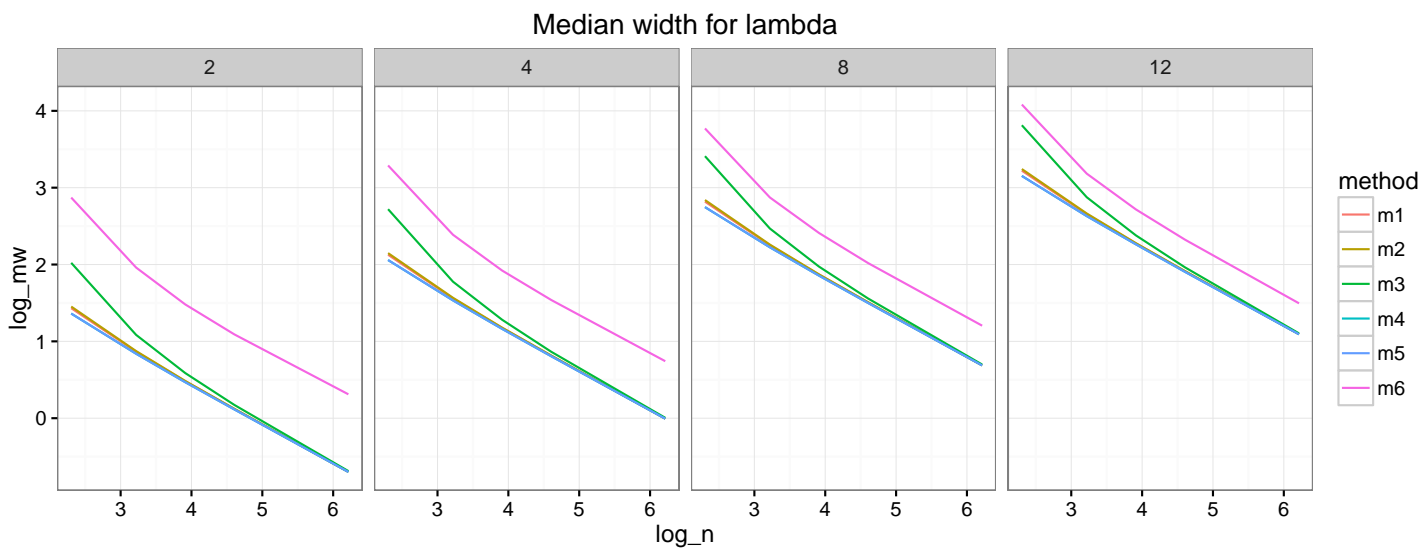


Figure 9

| "lambda" | "n" | "method1" | "method2" | "method3" | "method4" | "method5" | "method6" |
|----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|
| 2 | 10 | 0.7941 | 0.9324 | 0.7339 | 0.8583 | 0.886 | 0.7042 |
| 2 | 25 | 0.8688 | 0.9409 | 0.829 | 0.9074 | 0.9058 | 0.8061 |
| 2 | 50 | 0.9061 | 0.9479 | 0.878 | 0.9336 | 0.9272 | 0.8601 |
| 2 | 100 | 0.9213 | 0.946 | 0.905 | 0.9385 | 0.9359 | 0.8924 |
| 2 | 500 | 0.9462 | 0.951 | 0.9421 | 0.9491 | 0.9489 | 0.9359 |
| 4 | 10 | 0.8468 | 0.9309 | 0.7984 | 0.8812 | 0.9185 | 0.7741 |
| 4 | 25 | 0.8928 | 0.9431 | 0.8678 | 0.9191 | 0.929 | 0.8498 |
| 4 | 50 | 0.9243 | 0.9497 | 0.9052 | 0.939 | 0.9419 | 0.8928 |
| 4 | 100 | 0.9342 | 0.9428 | 0.9231 | 0.9372 | 0.9399 | 0.914 |
| 4 | 500 | 0.9483 | 0.9494 | 0.946 | 0.9487 | 0.9476 | 0.9442 |
| 8 | 10 | 0.872 | 0.9283 | 0.8461 | 0.8925 | 0.9328 | 0.8293 |
| 8 | 25 | 0.9137 | 0.9416 | 0.8984 | 0.9281 | 0.9382 | 0.8862 |
| 8 | 50 | 0.9363 | 0.9516 | 0.9247 | 0.9444 | 0.9491 | 0.9147 |
| 8 | 100 | 0.9401 | 0.9494 | 0.9348 | 0.9446 | 0.9472 | 0.9285 |
| 8 | 500 | 0.9475 | 0.9483 | 0.9458 | 0.9478 | 0.9476 | 0.9445 |
| 12 | 10 | 0.881 | 0.9282 | 0.863 | 0.8946 | 0.9376 | 0.8464 |
| 12 | 25 | 0.9191 | 0.9422 | 0.9056 | 0.9267 | 0.941 | 0.8957 |
| 12 | 50 | 0.9386 | 0.9478 | 0.9311 | 0.9426 | 0.9486 | 0.9249 |
| 12 | 100 | 0.9416 | 0.9491 | 0.9346 | 0.945 | 0.9499 | 0.9322 |
| 12 | 500 | 0.9472 | 0.9471 | 0.9466 | 0.9473 | 0.9483 | 0.9464 |

Table 1: Coverage Rate for μ

| "lambda" | "n" | "method1" | "method2" | "method3" | "method4" | "method5" | "method6" |
|----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|
| 2 | 10 | 0.9706 | 0.9292 | 0.9026 | 0.9517 | 0.9517 | 0.9607 |
| 2 | 25 | 0.9598 | 0.9444 | 0.9193 | 0.9497 | 0.9497 | 0.9354 |
| 2 | 50 | 0.9554 | 0.9492 | 0.9288 | 0.9501 | 0.9501 | 0.9339 |
| 2 | 100 | 0.9507 | 0.9459 | 0.9404 | 0.948 | 0.948 | 0.9352 |
| 2 | 500 | 0.9529 | 0.9521 | 0.9505 | 0.952 | 0.952 | 0.9455 |
| 4 | 10 | 0.9698 | 0.9311 | 0.9023 | 0.951 | 0.951 | 0.9418 |
| 4 | 25 | 0.9588 | 0.9432 | 0.9191 | 0.9485 | 0.9485 | 0.9306 |
| 4 | 50 | 0.9567 | 0.9485 | 0.9302 | 0.9518 | 0.9518 | 0.9364 |
| 4 | 100 | 0.951 | 0.9456 | 0.9408 | 0.9481 | 0.9481 | 0.9389 |
| 4 | 500 | 0.9543 | 0.9517 | 0.9502 | 0.9523 | 0.9523 | 0.947 |
| 8 | 10 | 0.9699 | 0.9289 | 0.9019 | 0.9503 | 0.9503 | 0.9262 |
| 8 | 25 | 0.957 | 0.945 | 0.9192 | 0.9487 | 0.9487 | 0.9268 |
| 8 | 50 | 0.9547 | 0.9494 | 0.9305 | 0.9512 | 0.9512 | 0.9369 |
| 8 | 100 | 0.9517 | 0.9452 | 0.9406 | 0.9477 | 0.9477 | 0.9423 |
| 8 | 500 | 0.9538 | 0.9515 | 0.9498 | 0.9525 | 0.9525 | 0.9482 |
| 12 | 10 | 0.969 | 0.9272 | 0.9025 | 0.9511 | 0.9511 | 0.9218 |
| 12 | 25 | 0.9597 | 0.9449 | 0.9193 | 0.9494 | 0.9494 | 0.9268 |
| 12 | 50 | 0.9555 | 0.9495 | 0.9291 | 0.9526 | 0.9526 | 0.9372 |
| 12 | 100 | 0.9513 | 0.9458 | 0.9415 | 0.9477 | 0.9477 | 0.9431 |
| 12 | 500 | 0.9532 | 0.9521 | 0.9497 | 0.9525 | 0.9525 | 0.9492 |

Table 2: Coverage Rate for λ

| "lambda" | "n" | "method1" | "method2" | "method3" | "method4" | "method5" | "method6" |
|----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|
| 2 | 10 | 7.4001 | 45.727 | 7.199 | 8.4691 | 13.4934 | 5.7284 |
| 2 | 25 | 5.5346 | 9.1635 | 5.4744 | 5.9553 | 7.4617 | 4.7185 |
| 2 | 50 | 4.1392 | 5.1925 | 4.1 | 4.2945 | 4.856 | 3.7141 |
| 2 | 100 | 3.0102 | 3.3565 | 2.9955 | 3.0705 | 3.2708 | 2.8149 |
| 2 | 500 | 1.381 | 1.41 | 1.3768 | 1.3845 | 1.403 | 1.3548 |
| 4 | 10 | 5.7566 | 11.9495 | 5.6848 | 6.2592 | 8.9142 | 4.8243 |
| 4 | 25 | 4.063 | 5.222 | 4.033 | 4.2235 | 4.9223 | 3.6593 |
| 4 | 50 | 2.9679 | 3.344 | 2.9537 | 3.0301 | 3.2788 | 2.787 |
| 4 | 100 | 2.1491 | 2.276 | 2.1431 | 2.1721 | 2.2606 | 2.0653 |
| 4 | 500 | 0.976 | 0.986 | 0.975 | 0.9778 | 0.9861 | 0.966 |
| 8 | 10 | 4.2347 | 6.1545 | 4.198 | 4.4093 | 5.8501 | 3.8137 |
| 8 | 25 | 2.9183 | 3.354 | 2.898 | 2.9744 | 3.3387 | 2.7506 |
| 8 | 50 | 2.1296 | 2.28 | 2.1281 | 2.1552 | 2.2813 | 2.045 |
| 8 | 100 | 1.5265 | 1.578 | 1.5237 | 1.5339 | 1.5799 | 1.496 |
| 8 | 500 | 0.691 | 0.695 | 0.6901 | 0.6912 | 0.6953 | 0.6866 |
| 12 | 10 | 3.5064 | 4.6245 | 3.488 | 3.6108 | 4.6698 | 3.2474 |
| 12 | 25 | 2.3988 | 2.663 | 2.3907 | 2.4301 | 2.6898 | 2.2949 |
| 12 | 50 | 1.7477 | 1.84 | 1.7438 | 1.7582 | 1.8499 | 1.6899 |
| 12 | 100 | 1.2484 | 1.28 | 1.2462 | 1.2512 | 1.2845 | 1.2299 |
| 12 | 500 | 0.5641 | 0.566 | 0.5635 | 0.564 | 0.5672 | 0.5614 |

Table 3: Median Width for μ

| "lambda" | "n" | "method1" | "method2" | "method3" | "method4" | "method5" | "method6" |
|----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|
| 2 | 10 | 4.1874 | 4.276 | 7.5531 | 3.9031 | 3.9031 | 17.6629 |
| 2 | 25 | 2.3757 | 2.395 | 2.9569 | 2.3133 | 2.3133 | 7.1005 |
| 2 | 50 | 1.6176 | 1.6235 | 1.7996 | 1.5981 | 1.5981 | 4.3961 |
| 2 | 100 | 1.1288 | 1.13 | 1.1883 | 1.1208 | 1.1208 | 2.9763 |
| 2 | 500 | 0.4973 | 0.497 | 0.5024 | 0.4966 | 0.4966 | 1.3649 |
| 4 | 10 | 8.3938 | 8.573 | 15.1829 | 7.8323 | 7.8323 | 26.827 |
| 4 | 25 | 4.7515 | 4.791 | 5.9095 | 4.6316 | 4.6316 | 10.8836 |
| 4 | 50 | 3.2364 | 3.249 | 3.6001 | 3.197 | 3.197 | 6.832 |
| 4 | 100 | 2.2568 | 2.261 | 2.3741 | 2.2398 | 2.2398 | 4.6577 |
| 4 | 500 | 0.9948 | 0.994 | 1.005 | 0.9936 | 0.9936 | 2.0995 |
| 8 | 10 | 16.7491 | 17.1075 | 30.2689 | 15.5985 | 15.5985 | 43.4247 |
| 8 | 25 | 9.4969 | 9.577 | 11.8034 | 9.2449 | 9.2449 | 17.6828 |
| 8 | 50 | 6.4756 | 6.502 | 7.1915 | 6.394 | 6.394 | 11.1394 |
| 8 | 100 | 4.514 | 4.523 | 4.7476 | 4.4786 | 4.4786 | 7.5286 |
| 8 | 500 | 1.9894 | 1.989 | 2.0105 | 1.9872 | 1.9872 | 3.3332 |
| 12 | 10 | 25.0642 | 25.601 | 45.2767 | 23.4108 | 23.4108 | 59.2827 |
| 12 | 25 | 14.2387 | 14.359 | 17.7368 | 13.8751 | 13.8751 | 24.1096 |
| 12 | 50 | 9.7206 | 9.761 | 10.7984 | 9.6105 | 9.6105 | 15.1377 |
| 12 | 100 | 6.7733 | 6.7865 | 7.1215 | 6.7167 | 6.7167 | 10.2105 |
| 12 | 500 | 2.984 | 2.984 | 3.0144 | 2.9794 | 2.9794 | 4.4586 |

Table 4: Median Width for λ

| "lambda" | "n" | "method1" | "method2" | "method3" | "method4" | "method5" | "method6" |
|----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|
| 2 | 10 | 0.296 | 0 | 0.55 | 0 | 0 | 0.2654 |
| 2 | 25 | 0.0041 | 0 | 0.0649 | 0 | 0 | 0.0422 |
| 2 | 50 | 0 | 0 | 0 | 0 | 0 | 0.001 |
| 2 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 500 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 10 | 0.038 | 0 | 0.176 | 0 | 0 | 0.0776 |
| 4 | 25 | 0 | 0 | 0 | 0 | 0 | 0.0014 |
| 4 | 50 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 500 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 10 | 2e-04 | 0 | 0.0074 | 0 | 0 | 0.0076 |
| 8 | 25 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 50 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 500 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 10 | 0 | 0 | 1e-04 | 0 | 0 | 5e-04 |
| 12 | 25 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 50 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 500 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 5: Out of Bound Probability for μ

| "lambda" | "n" | "method1" | "method2" | "method3" | "method4" | "method5" | "method6" |
|----------|-----|-----------|-----------|-----------|-----------|-----------|-----------|
| 2 | 10 | 0 | 0 | 1 | 0 | 0 | 1 |
| 2 | 25 | 0 | 0 | 0.6154 | 0 | 0 | 1 |
| 2 | 50 | 0 | 0 | 0 | 0 | 0 | 0.9607 |
| 2 | 100 | 0 | 0 | 0 | 0 | 0 | 0.1697 |
| 2 | 500 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 10 | 0 | 0 | 1 | 0 | 0 | 1 |
| 4 | 25 | 0 | 0 | 0.6108 | 0 | 0 | 1 |
| 4 | 50 | 0 | 0 | 0 | 0 | 0 | 0.5671 |
| 4 | 100 | 0 | 0 | 0 | 0 | 0 | 0.0027 |
| 4 | 500 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 10 | 0 | 0 | 1 | 0 | 0 | 1 |
| 8 | 25 | 0 | 0 | 0.6097 | 0 | 0 | 1 |
| 8 | 50 | 0 | 0 | 0 | 0 | 0 | 0.0482 |
| 8 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 500 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 10 | 0 | 0 | 1 | 0 | 0 | 1 |
| 12 | 25 | 0 | 0 | 0.6088 | 0 | 0 | 1 |
| 12 | 50 | 0 | 0 | 0 | 0 | 0 | 0.0027 |
| 12 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 500 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 6: Out of Bound Probability for λ