R1 project

Group 10

Christopher Comora Nataliya Peshekhodko Zhuoning Liu

2020-February-08

Abstract

This project will involve a simulation study and a real data analysis with horse-kick death data. The goal of this project is in making inference for the mean of the distribution. Several different ways of constructing confidence intervals will be constructed.

1. General Procedure

Setting up variables.

```
n=10
lambda = 1
B = 1000
alpha = 0.05
```

1.a Generate a random sample of size 10 with parameter λ=1.

```
data = rpois (n, lambda)
```

1.b Create and save a 95% CI for the mean of the population using the approximate confidence interval 1.

```
lambda_hat = mean (data)

lower_bound_ci1 = lambda_hat - qnorm(1-alpha/2)*sqrt(var(data)/length(data))
upper_bound_ci1 = lambda_hat + qnorm(1-alpha/2)*sqrt(var(data)/length(data))
```

Calculated CI is

```
c(lower_bound_ci1, upper_bound_ci1)
```

```
## [1] 0.1733393 0.8266607
```

1.c Create and save a 95% CI for the mean of the population using the approximate confidence interval 2.

```
lower_bound_ci2 = lambda_hat - qnorm(1-alpha/2)*sqrt(lambda_hat/length(data))
upper_bound_ci2 = lambda_hat + qnorm(1-alpha/2)*sqrt(lambda_hat/length(data))
```

Claculated CI is

```
c(lower_bound_ci2, upper_bound_ci2)
```

```
## [1] 0.06173873 0.93826127
```

1.d Create and save 95% CIs for the mean of the population using the parametric bootstrap method. ($B = 1000$)

Replicating randomly generated data from Poisson distribution `B = 1000` times

```
bootData <- replicate (n=B, expr = rpois (n, lambda_hat))
means<-colMeans(bootData)
```

Calculated CI is

```
c(quantile (means, 0.025), quantile (means, 0.975))
```

```
##  2.5% 97.5%
##   0.1   1.0
```

1.e Create and save 95% CIs for the mean of the population using the nonparametric bootstrap method. ($B = 1000$)

```
lambda_vec <- rep(NA, B)
for(i in 1:B){
sel <- sample(1:n, n, replace=TRUE)
  bootstrap_x <- data[sel]
  lambda_vec[i] <- mean(bootstrap_x) }
```

Calculated CI is

```
quantile(lambda_vec,c(0.025,0.975))
```

```
##  2.5% 97.5%
##   0.2   0.8
```

2. Repeat the general procedure `N=5000` times

```
N = 5000
```

Create and save a 95% CI for the mean of the population using the approximate confidence interval 1. Repeat this action `N = 5000` times. Sample size `n=` 10.

```
means <- c()
lower_bound <- c()
upper_bound <- c()

for (j in 1:N){
  data = rpois (n, lambda)
  means[j] = mean (data)
  lower_bound[j] = means[j] - qnorm(1-alpha/2)*sqrt(var(data)/length(data))
  upper_bound[j] = means[j] + qnorm(1-alpha/2)*sqrt(var(data)/length(data))
}
```

Save calculated CIs in data frame

```
ci_1 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Create and save a 95% CI for the mean of the population using the approximate confidence interval 2. Repeat this action `N` = 5000 times. Sample size `n`= 10.

```
lower_bound <- c()
upper_bound <- c()
for (j in 1:N) {
  lower_bound[j] = means[j] - qnorm(1-alpha/2)*sqrt(means[j]/length(data))
  upper_bound[j] = means[j] + qnorm(1-alpha/2)*sqrt(means[j]/length(data))
}
```

Save calculated CIs in data frame

```
ci_2 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Create and save a 95% CI for the mean of the population using parametric bootstrap. Repeat this action `N` = 5000 times. Sample size `n`= 10, bootstrap size `B` = 1000.

```
means <- c()
lower_bound <- c()
upper_bound <- c()

for (j in 1:N) {
  bootData <- replicate (n=B, expr = rpois (n, lambda_hat))
  means<-colMeans(bootData)
  lower_bound[j] = quantile (means, 0.025)
  upper_bound[j] = quantile (means, 0.975)
}
```

Save calculated CIs in data frame.

```
ci_3 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Create and save a 95% CI for the mean of the population using nonparametric bootstrap. Repeat this action `N` = 5000 times. Sample size `n`= 10, bootstrap size `B` = 1000.

```
 means <- c()
 lower_bound <- c()
 upper_bound <- c()

 for (j in 1:N) {
   lambda_vec <- rep(NA, B)
   for(i in 1:B){
     sel <- sample(1:n, n, replace=TRUE)
     bootstrap_x <- data[sel]
     lambda_vec[i] <- mean(bootstrap_x)  }
   lower_bound[j] <- quantile(lambda_vec, 0.025)
   upper_bound[j] <- quantile(lambda_vec, 0.975)}
```

Save calculated CIs in data frame.

```
 ci_4 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

3. Summary about different method execution

Calculate proportion of times each method contains the true value of the `mean=` 1.

```
ci_1_prop <- nrow (subset (ci_1, ci_1$lower_bound<=lambda &
                             ci_1$upper_bound >=lambda))/nrow(ci_1)
ci_2_prop <- nrow (subset (ci_2, ci_2$lower_bound<=lambda &
                             ci_2$upper_bound >=lambda))/nrow(ci_2)
ci_3_prop <- nrow (subset (ci_3, ci_3$lower_bound<=lambda &
                             ci_3$upper_bound >=lambda))/nrow(ci_3)
ci_4_prop <- nrow (subset (ci_4, ci_4$lower_bound<=lambda &
                             ci_4$upper_bound >=lambda))/nrow(ci_4)
```

Calculate the average length of the CIs created by the procedure.

```
ci_1_avg <- mean (ci_1$upper_bound - ci_1$lower_bound)
ci_2_avg <- mean (ci_2$upper_bound - ci_2$lower_bound)
ci_3_avg <- mean (ci_3$upper_bound - ci_3$lower_bound)
ci_4_avg <- mean (ci_4$upper_bound - ci_4$lower_bound)
```

Organize all calculated data into the table.

```
dt <- data.frame (c ("Approximate 1", "Approximate 2", "Parametric Bootstrap",
                     "Nonparametric Bootstrap"),
                  c(ci_1_prop, ci_2_prop, ci_3_prop, ci_4_prop),
                  c (ci_1_avg, ci_2_avg, ci_3_avg, ci_4_avg))
colnames(dt) <- c ("Method", "Prop Containing", "Avg Length")
kable(dt)
```

Method	Prop Containing	Avg Length
Approximate 1	0.9066	1.1918767
Approximate 2	0.9264	1.2210844
Parametric Bootstrap	0.8702	0.8869175
Nonparametric Bootstrap	1.0000	1.1048470

4. Repeat steps above for diffrent sample sizes

```
n=30
```

Create and save a 95% CI for the mean of the population using the approximate confidence interval 1. Repeat this action `N = 5000` times. Sample size `n=` 30.

```
data = rpois (n, lambda)
lambda_hat = mean (data)
approx_var = var (data)

means <- c()
lower_bound <- c()
upper_bound <- c()

for (j in 1:N){
  data = rpois (n, lambda)
  means[j] = mean (data)
  lower_bound[j] = means[j] - qnorm(1-alpha/2)*sqrt(var(data)/length(data))
  upper_bound[j] = means[j] + qnorm(1-alpha/2)*sqrt(var(data)/length(data))
}
```

Save calculated CIs to the data frame.

```
ci_1 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Create and save a 95% CI for the mean of the population using the approximate confidence interval 2. Repeat this action `N = 5000` times. Sample size `n=` 30.

```
lower_bound <- c()
upper_bound <- c()
for (j in 1:N) {
  lower_bound[j] = means[j] - qnorm(1-alpha/2)*sqrt(means[j]/length(data))
  upper_bound[j] = means[j] + qnorm(1-alpha/2)*sqrt(means[j]/length(data))
}
```

Save calculated CIs to the data frame.

```
ci_2 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Create and save a 95% CI for the mean of the population using parametric bootstrap. Repeat this action `N = 5000` times. Sample size `n=` 30, bootstrap size `B = 1000`.

```
means <- c()
lower_bound <- c()
upper_bound <- c()

for (j in 1:N) {
  bootData <- replicate (n=B, expr = rpois (n, lambda_hat))
  means<-colMeans(bootData)
  lower_bound[j] = quantile (means, 0.025)
  upper_bound[j] = quantile (means, 0.975)
}
```

Save calculated CIs to the data frame.

```
ci_3 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Create and save a 95% CI for the mean of the population using nonparametric bootstrap. Repeat this action `N` = 5000 times. Sample size `n=` 30, bootstrap size `B` = 1000.

```
 means <- c()
 lower_bound <- c()
 upper_bound <- c()

 for (j in 1:N) {
   lambda_vec <- rep(NA, B)
   for(i in 1:B){
     sel <- sample(1:n, n, replace=TRUE)
     bootstrap_x <- data[sel]
     lambda_vec[i] <- mean(bootstrap_x)  }
   lower_bound[j] <- quantile(lambda_vec, 0.025)
   upper_bound[j] <- quantile(lambda_vec, 0.975)}
 ci_4 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Now calculate proportion of times each method contains the true value of the mean `=1` and the average length of the CIs created by the procedure.

```
 ci_1_prop <- nrow (subset (ci_1, ci_1$lower_bound<=lambda &
                             ci_1$upper_bound >=lambda))/nrow(ci_1)
 ci_2_prop <- nrow (subset (ci_2, ci_2$lower_bound<=lambda &
                             ci_2$upper_bound >=lambda))/nrow(ci_2)
 ci_3_prop <- nrow (subset (ci_3, ci_3$lower_bound<=lambda &
                             ci_3$upper_bound >=lambda))/nrow(ci_3)
 ci_4_prop <- nrow (subset (ci_4, ci_4$lower_bound<=lambda &
                             ci_4$upper_bound >=lambda))/nrow(ci_4)

ci_1_avg <- mean (ci_1$upper_bound - ci_1$lower_bound)
ci_2_avg <- mean (ci_2$upper_bound - ci_2$lower_bound)
ci_3_avg <- mean (ci_3$upper_bound - ci_3$lower_bound)
ci_4_avg <- mean (ci_4$upper_bound - ci_4$lower_bound)
```

Now summarized all calculated information

```
dt <- data.frame (c ("Approximate 1", "Approximate 2", "Parametric Bootstrap",
                     "Nonparametric Bootstrap"),
                  c(ci_1_prop, ci_2_prop, ci_3_prop, ci_4_prop),
                  c (ci_1_avg, ci_2_avg, ci_3_avg, ci_4_avg))
colnames(dt) <- c ("Method", "Prop Containing", "Avg Length")
kable(dt)
```

Method	Prop Containing	Avg Length
Approximate 1	0.9362	0.7092224
Approximate 2	0.9358	0.7132659
Parametric Bootstrap	1.0000	0.7464848
Nonparametric Bootstrap	1.0000	0.6265807

```
n=200
```

Repeat steps above for sample size n=200

```
data = rpois (n, lambda)
```

Creating and saving 95% CI for mean using the approximate confidence interval 1 approach and save data to the data frame:

```
lambda_hat = mean (data)

means <- c()
lower_bound <- c()
upper_bound <- c()

for (j in 1:N){
  data = rpois (n, lambda_hat)
  means[j] = mean (data)
  lower_bound[j] = means[j] - qnorm(1-alpha/2)*sqrt(var(data)/length(data))
  upper_bound[j] = means[j] + qnorm(1-alpha/2)*sqrt(var(data)/length(data))
}
ci_1 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Creating and saving 95% CI for mean using the approximate confidence interval 2 approach and save data to the data frame:

```
lower_bound <- c()
upper_bound <- c()
for (j in 1:N) {
  lower_bound[j] = means[j] - qnorm(1-alpha/2)*sqrt(means[j]/length(data))
  upper_bound[j] = means[j] + qnorm(1-alpha/2)*sqrt(means[j]/length(data))
}
ci_2 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Creating and saving 95% CI for mean using the parametric bootstrap approach and save data to the data frame:

```
means <- c()
lower_bound <- c()
upper_bound <- c()

for (j in 1:N) {
  bootData <- replicate (n=B, expr = rpois (n, lambda_hat))
  means<-colMeans(bootData)
  lower_bound[j] = quantile (means, 0.025)
  upper_bound[j] = quantile (means, 0.975)

}
ci_3 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Creating and saving 95% CI for mean using the non-parametric bootstrap approach and save data to the data frame:

```
means <- c()
lower_bound <- c()
upper_bound <- c()

for (j in 1:N) {
  lambda_vec <- rep(NA, B)
  for(i in 1:B){
    sel <- sample(1:n, n, replace=TRUE)
    bootstrap_x <- data[sel]
    lambda_vec[i] <- mean(bootstrap_x)  }
  lower_bound[j] <- quantile(lambda_vec, 0.025)
  upper_bound[j] <- quantile(lambda_vec, 0.975)}
ci_4 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Now calculate proportion of times each method contains the true value of the `mean = 1` and the average length of the CIs created by the procedure.

```
ci_1_prop <- nrow (subset (ci_1, ci_1$lower_bound<=lambda &
                             ci_1$upper_bound>=lambda))/nrow(ci_1)
ci_2_prop <- nrow (subset (ci_2, ci_2$lower_bound<=lambda &
                             ci_2$upper_bound>=lambda))/nrow(ci_2)
ci_3_prop <- nrow (subset (ci_3, ci_3$lower_bound<=lambda &
                             ci_3$upper_bound>=lambda))/nrow(ci_3)
ci_4_prop <- nrow (subset (ci_4, ci_4$lower_bound<=lambda &
                             ci_4$upper_bound>=lambda))/nrow(ci_4)

ci_1_avg <- mean (ci_1$upper_bound - ci_1$lower_bound)
ci_2_avg <- mean (ci_2$upper_bound - ci_2$lower_bound)
ci_3_avg <- mean (ci_3$upper_bound - ci_3$lower_bound)
ci_4_avg <- mean (ci_4$upper_bound - ci_4$lower_bound)
```

Summarize all calculated information

```
dt <- data.frame (c ("Approximate 1", "Approximate 2", "Parametric Bootstrap",
                     "Nonparametric Bootstrap"),
                  c(ci_1_prop, ci_2_prop, ci_3_prop, ci_4_prop),
                  c (ci_1_avg, ci_2_avg, ci_3_avg, ci_4_avg))
colnames(dt) <- c ("Method", "Prop Containing", "Avg Length")
kable(dt)
```

Method	Prop Containing	Avg Length
Approximate 1	0.6724	0.2919475
Approximate 2	0.6680	0.2924987
Parametric Bootstrap	1.0000	0.2911804
Nonparametric Bootstrap	1.0000	0.2697736

5. Advantages and disadvantages of different types of interval

1. **Confidence interval based on Normal distribution with true population variance.** Difficult to use in real life because usually we do not know population variance. If population variance is

known, Confidence Interval estimation is more correct compare with Confidence Interval estimation with approximated variance.

2. **Confidence interval based on Normal distribution with estimated variance.** Relies on CLT and requires sample size to be large. For not large samples could not work correctly.
   Does not require normality from the original data and does not require knowledge about true population variance.

3. **Confidence Interval based on Parametric bootsrap.** Requires assumption about population distribution that could be wrong. If you do not have large enough sample and know population distribution (or made correct prediction), then Parametric bootstarp will work better than Nonparametric. For small samples with good assumption about population distribution, Parametric Bootstrap works better than Nonparametric bootstrap.

4. **Confidence Interval based on Nonparametric bootsrtap** Works very well when sample size is large enough and we do not know population distribution. Big advatange that should not do any guess about population distribution. Disadvantage - samlpe size should be reasonably large.

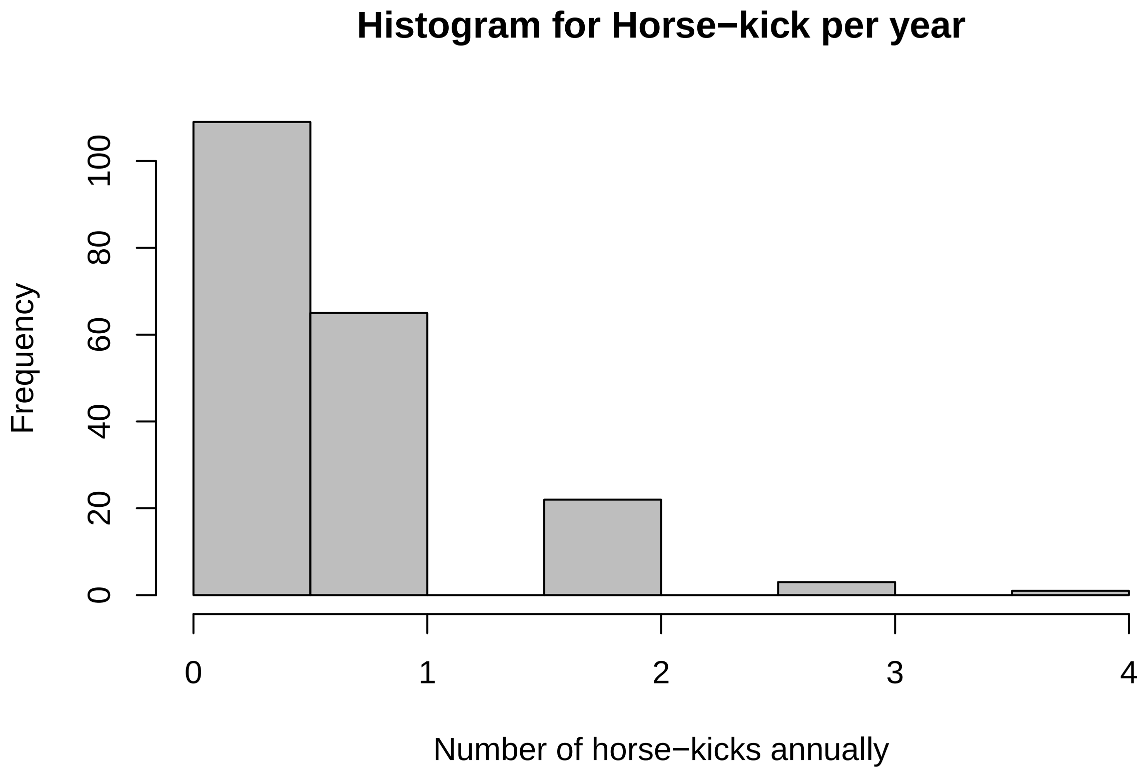
6. Apply the 4 different methods to the horse-kick death data and report the 95% confidence intervals obtained by each method.

Generating horse-kick death data, `n=` ’r n‘

```
data <- c(rep(0, 109), rep(1,65), rep(2,22), rep(3,3), rep(4,1))
```

Histogram of the horse-kick data distribution

```
hist (data, main = "Histogram for Horse-kick per year", xlab = "Number of horse-kicks annually", col =
```



Creating and saving 95% CI for mean using the approximate confidence interval 1 approach and save data to the data frame:

```
lambda_hat = mean (data)

means <- c()
lower_bound <- c()
upper_bound <- c()

  data = rpois (n, lambda_hat)
  means = mean (data)
  lower_bound = means - qnorm(1-alpha/2)*sqrt(var(data)/length(data))
  upper_bound = means + qnorm(1-alpha/2)*sqrt(var(data)/length(data))

ci_1 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Creating and saving 95% CI for mean using the approximate confidence interval 2 approach and save data to the data frame:

```
lower_bound <- c()
upper_bound <- c()

  lower_bound = means - qnorm(1-alpha/2)*sqrt(means/length(data))
  upper_bound = means + qnorm(1-alpha/2)*sqrt(means/length(data))

ci_2 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Creating and saving 95% CI for mean using the parametric bootstrap approach and save data to the data frame:

```
means <- c()
lower_bound <- c()
upper_bound <- c()

  bootData <- replicate (n=B, expr = rpois (n, lambda_hat))
  means<-colMeans(bootData)
  lower_bound = quantile (means, 0.025)
  upper_bound = quantile (means, 0.975)

ci_3 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

Creating and saving 95% CI for mean using the non-parametric bootstrap approach and save data to the data frame:

```
 means <- c()
 lower_bound <- c()
 upper_bound <- c()

   lambda_vec <- rep(NA, B)
   for(i in 1:B){
     sel <- sample(1:n, n, replace=TRUE)
     bootstrap_x <- data[sel]
     lambda_vec[i] <- mean(bootstrap_x)  }
   lower_bound <- quantile(lambda_vec, 0.025)
   upper_bound <- quantile(lambda_vec, 0.975)
ci_4 <- data.frame ("lower_bound" = lower_bound, "upper_bound" = upper_bound)
```

95% Confidence Interval depending on method

```
dci <- data.frame (  c("CI 1", "CI 2", "Parametric Bootsrap", "Nonparametric bootstrap"),
                     c( ci_1$lower_bound, ci_2$lower_bound,ci_3$lower_bound, ci_4$lower_bound ),
                     c(ci_1$upper_bound, ci_2$upper_bound, ci_3$upper_bound, ci_4$upper_bound)  )

colnames (dci) <- c ("Method", "Lower Bound", "Upper Bound")
kable(dci)
```

Method	Lower Bound	Upper Bound
CI 1	0.5083642	0.7316358
CI 2	0.5108738	0.7291262
Parametric Bootsrap	0.5000000	0.7250000
Nonparametric bootstrap	0.5100000	0.7350000

Conclusion

For randomly generated data from the Poisson distibution with sample size `n=10`, Parametric distribution works better than nonparametric. As sample size grows `n=30` and `n=200`, Nonparametric bootstrap shows better results: proportion contained true value is higer and confidence interval is more narrow. CI1 and CI2 show pretty much similar results.