Bootstrapping example on the titanic dataset

Tom Blain

20/10/2022

```
library("readr") #For read_csv
library("dplyr")
library("knitr") # For kable
library("coxed")

data <- read_csv(("TitanicFull.csv"))
data<-as.data.frame(data)
```

Hopefully anyone reading this will have seen the titanic dataset before, For some introduction and background to the dataset, please see https://www.kaggle.com/competitions/titanic/overview

In this experiment we are interested in studying the average age of adult passengers on the titanic. We will apply bootstrapping techniques on a random sample of the population. Then we will filter the data to survivors and separate by sex.

```
head(data)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                 Name    Sex Age SibSp Parch
## 1                            Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                             Heikkinen, Miss. Laina female  26     0     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                           Allen, Mr. William Henry   male  35     0     0
## 6                                   Moran, Mr. James   male  NA     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171    7.25  <NA>        S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282   7.925  <NA>        S
## 4           113803    53.1  C123        S
## 5           373450    8.05  <NA>        S
## 6           330877  8.4583  <NA>        Q
```

```
data <- subset(data, data$Age >= 18)
age_data <- data$Age
```

```r
age_data <- age_data[!is.na(age_data)]
c("True mean", mean(age_data))
```

```
## [1] "True mean"        "33.5831946755408"
```

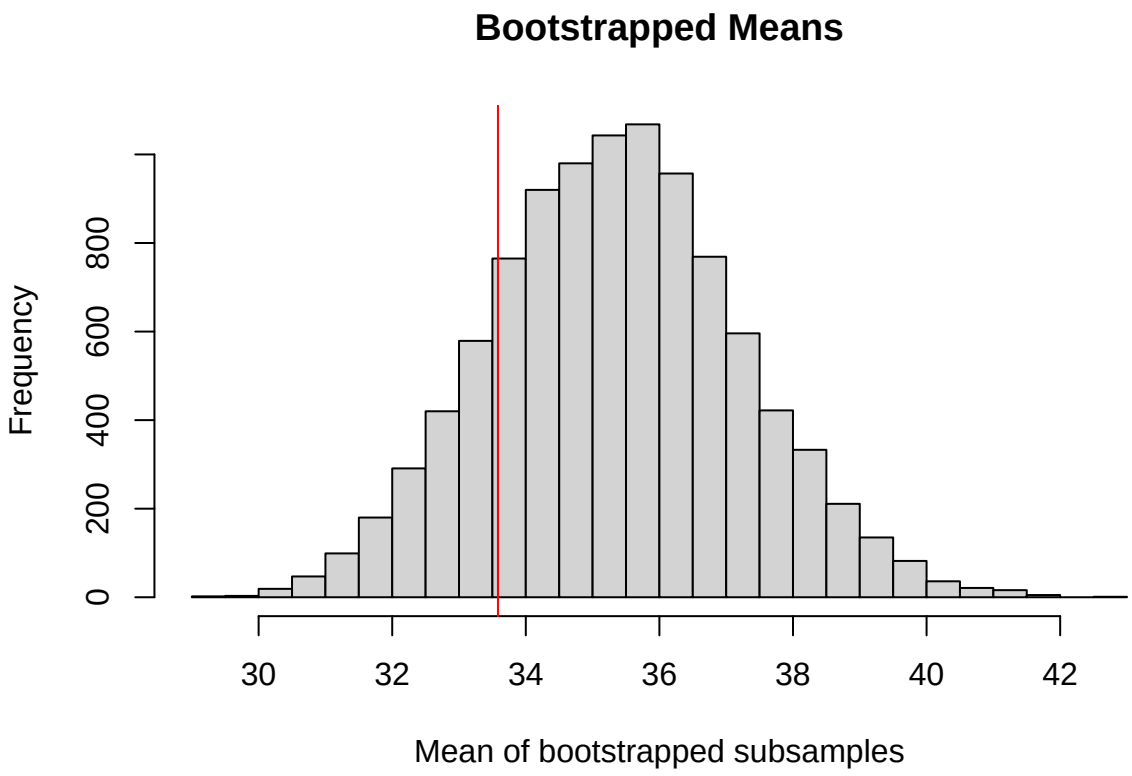
We know the true mean of the data. For our bootstrapping experiment, we will take a random sample of "n" values from the population.

```r
n <- 50
sample_age <- sample(age_data, n, replace = FALSE)

c("Sample mean", mean(sample_age))
```

```
## [1] "Sample mean" "35.37"
```

Now from this sample, we can subsample with replacement n values, known as bootstrapping. R has a built in function for this, known as boot(), We can also use sample() with replace set to true,

```r
bootstrap <- function(sample_age, mean_subsamples){
  subsample <- sample(sample_age, n, replace = TRUE)
  mean_subsamples <- append(mean_subsamples, mean(subsample))
  return(mean_subsamples)
  }

iterations <- 10000
bootstrap_means <- c()
for(i in 1:iterations){
  bootstrap_means <- bootstrap(sample_age, bootstrap_means)
}

hist(bootstrap_means, breaks = 20, xlab = "Mean of bootstrapped subsamples", ylab = "Frequency", main =
abline(v = mean(age_data), col = "red")
#text(locator(), labels = "True pop mean")
coords <- locator()
```




```
c("The True population mean is",mean(age_data))
```

```
## [1] "The True population mean is" "33.5831946755408"
```

```
c("Our bootstrapped estimate of the population mean from a random sample is",mean(bootstrap_means))
```

```
## [1] "Our bootstrapped estimate of the population mean from a random sample is"
## [2] "35.355003"
```

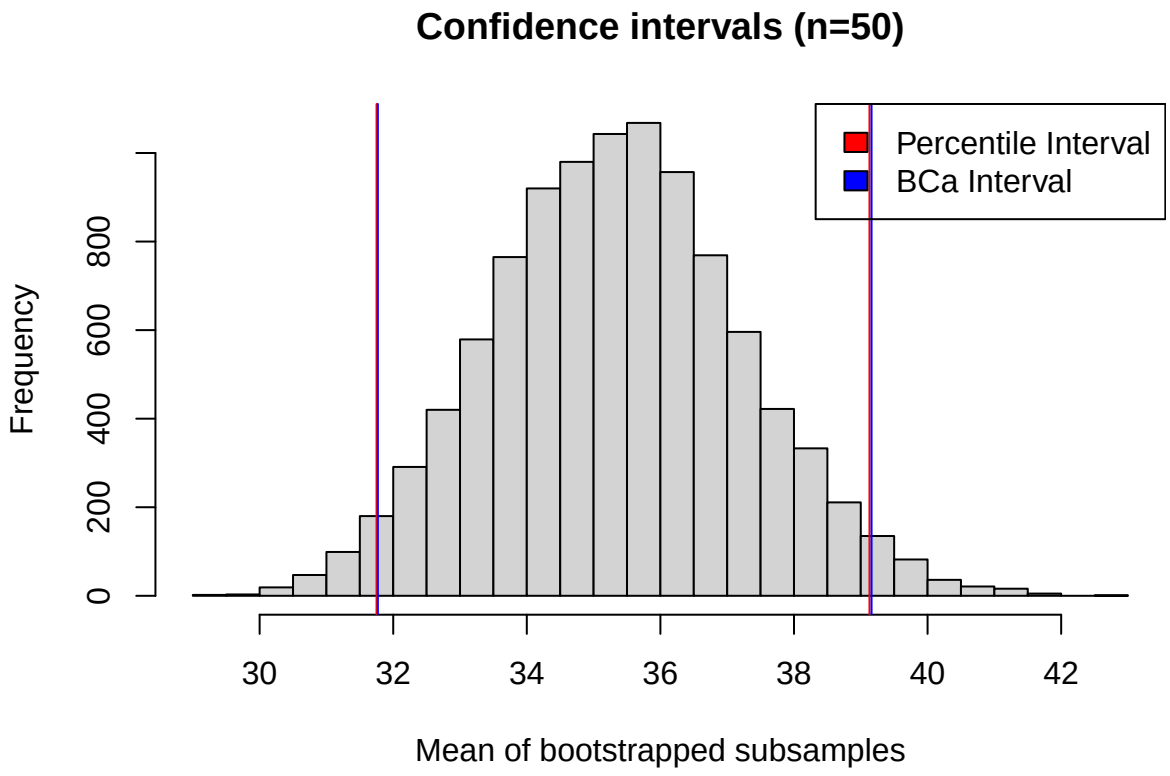
```
CI <- 0.95
orderedmean <- sort(bootstrap_means)
lower <- orderedmean[iterations*((1-CI)/2)]
upper <- orderedmean[iterations - (iterations*(1- CI)/2)]
c("Percentile CI",lower,upper)
```

```
## [1] "Percentile CI" "31.75"         "39.13"
```

```
bca <- bca(bootstrap_means, CI)
c("BCa CI", bca)
```

```
## [1] "BCa CI"           "31.77"            "39.1609870057047"
```

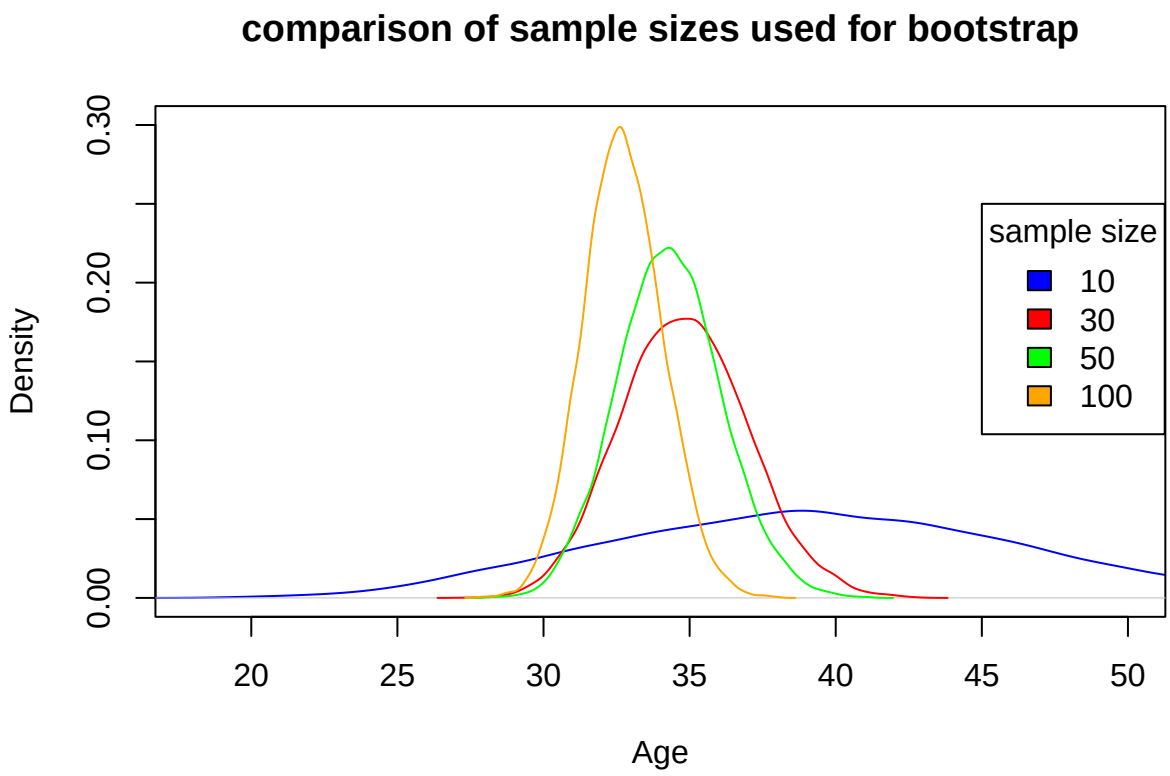
```r
hist(bootstrap_means, breaks = 20, xlab = "Mean of bootstrapped subsamples", ylab = "Frequency", main =
abline(v = lower, col = "red")
abline(v = upper, col = "red")
abline(v = bca[1], col = "blue")
abline(v = bca[2], col = "blue")
legend("topright",c("Percentile Interval", "BCa Interval"), fill = c("red","blue"))
```



lets experiment with the size of the sample from the dataset.

```r
n <- c(10,30,50,100)
colours <- c("blue", "red", "green", "orange", "pink")
for (j in 1:length(n)){
  sample_age <- sample(age_data, n[j], replace = FALSE)
  bootstrap <- function(sample_age, mean_subsamples){
    subsample <- sample(sample_age, n[j], replace = TRUE)
    mean_subsamples <- append(mean_subsamples, mean(subsample))
    return(mean_subsamples)
    }

  iterations <- 10000
  bootstrap_means <- c()
  for(i in 1:iterations){
    bootstrap_means <- bootstrap(sample_age, bootstrap_means)
  }
  if(j == 1){
    plot(density(bootstrap_means), xlab = "Age", xlim = c(18,50),ylim = c(0,0.3), col=colours[j], main =
```

```
  }else{
    lines(density(bootstrap_means), col=colours[j])
  }
}
legend(45, 0.25, legend = n, fill = colours, title = "sample size")
```

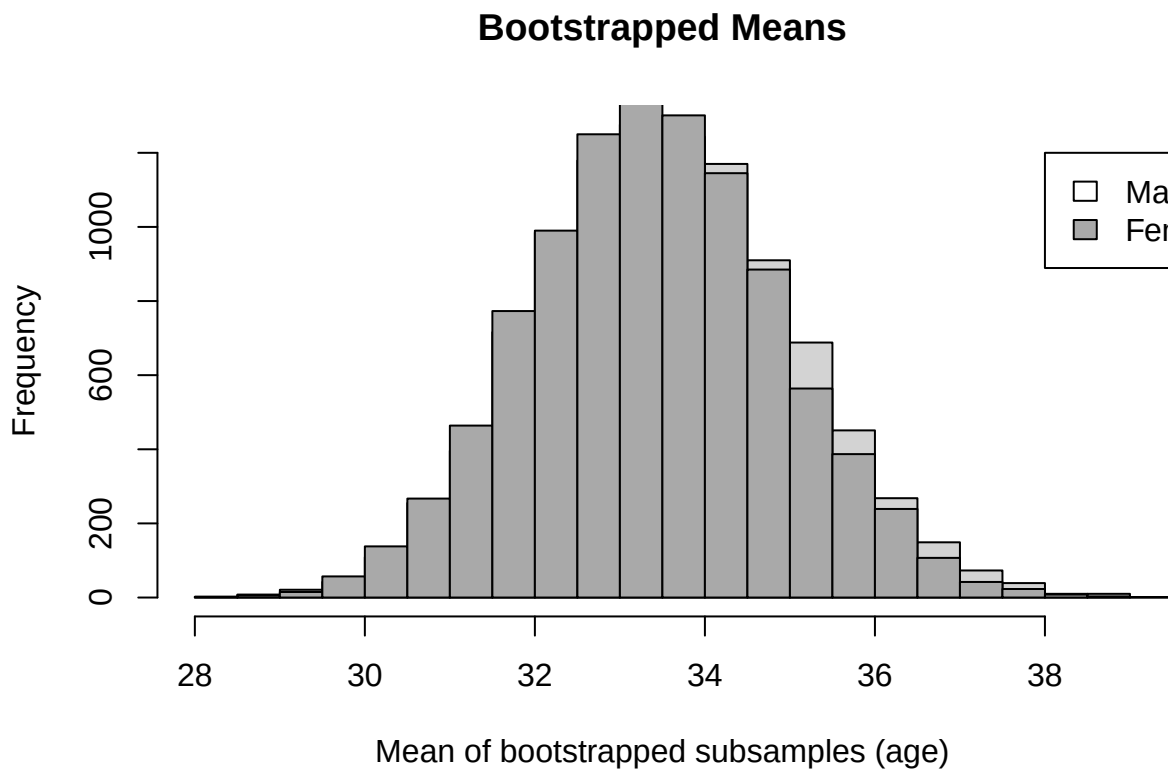


The results here are as expected. This shows the real world issue of choosing the right sample size to give a meaningful confidence interval when weighed up to cost and feasability - we will almost never have access to a whole population dataset.

Now we filter by survived and split the data by sex

```
survived_M <- subset(data, data$Survived == "1" & data$Sex == "male" & data$Age >= 18)
surv_data_M <- survived_M$Age
surv_data_M <- surv_data_M[!is.na(surv_data_M)]

survived_F <- subset(data, data$Survived == "1" & data$Sex == "female" & data$Age >= 18)
surv_data_F <- survived_F$Age
surv_data_F <- surv_data_F[!is.na(surv_data_F)]

n <- 50
sample_age_M <- sample(surv_data_M, n, replace = FALSE)
sample_age_F <- sample(surv_data_F, n, replace = FALSE)

#c("Sample mean", mean(sample_age_M))

bootstrap <- function(sample_age, mean_subsamples){
```

```
  subsample <- sample(sample_age, n, replace = TRUE)
  mean_subsamples <- append(mean_subsamples, mean(subsample))
  return(mean_subsamples)
  }

iterations <- 10000
bootstrap_means_M <- c()
bootstrap_means_F <- c()
for(i in 1:iterations){
  bootstrap_means_M <- bootstrap(sample_age_M, bootstrap_means_M)
  bootstrap_means_F <- bootstrap(sample_age_F, bootstrap_means_F)

}

hist(bootstrap_means_M, breaks = 20, xlab = "Mean of bootstrapped subsamples (age)", ylab = "Frequency"
hist(bootstrap_means_F, breaks = 20, add = TRUE, col = "dark grey")
legend(38, 1200, legend = c("Male", "Female"), fill = c("white", "dark grey"))
```



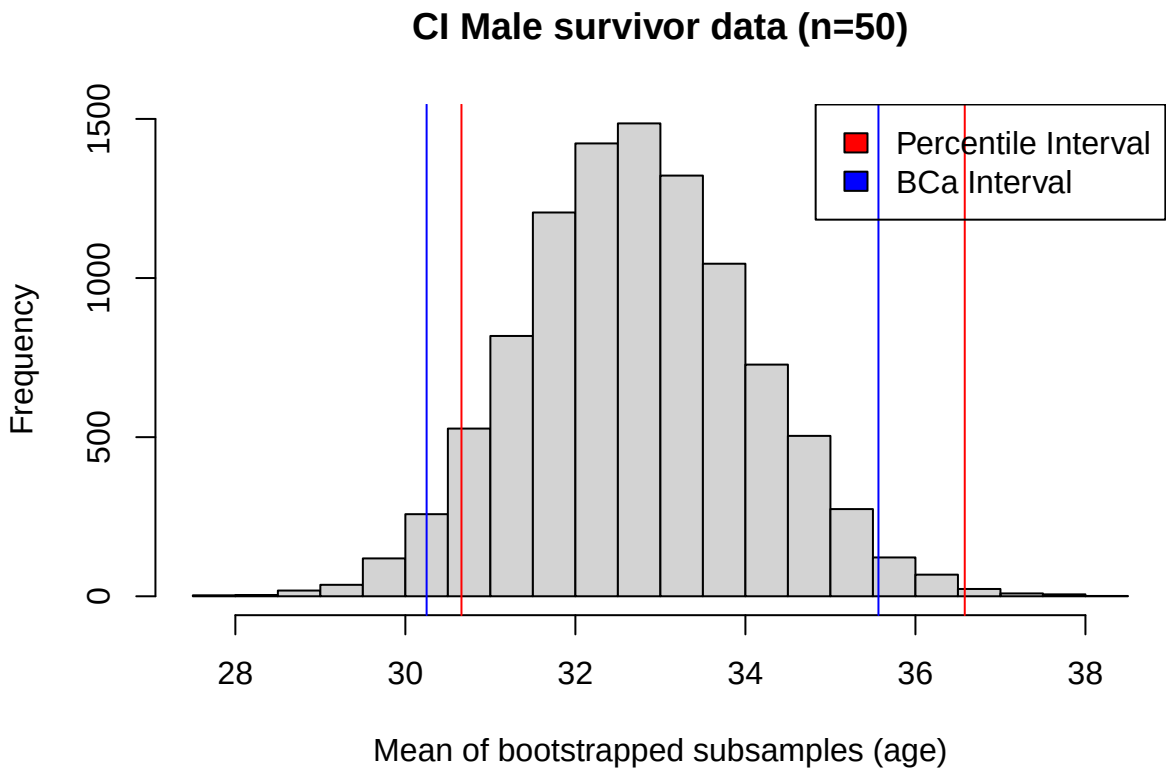

```
CI <- 0.95
orderedmean <- sort(bootstrap_means_M)
lower <- orderedmean[iterations*((1-CI)/2)]
upper <- orderedmean[iterations - (iterations*(1- CI)/2)]
c("Percentile CI",lower,upper)
```

```
## [1] "Percentile CI" "30.66"         "36.58"
```

```r
bca <- bca(bootstrap_means, CI)
c("BCa CI", bca)
```

```
## [1] "BCa CI" "30.25"  "35.565"
```

```r
hist(bootstrap_means, breaks = 20, xlab = "Mean of bootstrapped subsamples (age)", ylab = "Frequency", m
abline(v = lower, col = "red")
abline(v = upper, col = "red")
abline(v = bca[1], col = "blue")
abline(v = bca[2], col = "blue")
legend("topright",c("Percentile Interval", "BCa Interval"), fill = c("red","blue"))
```



There appears to be a large difference in the BCa and percentile intervals. This may indicate bias and/or skewness in the data. More research into the bias and skewness should be conducted.