Bootstrapping example on the titanic dataset

Tom Blain

20/10/2022

```
library("readr") #For read_csv
library("dplyr")
```

```
##
## Attaching package: ’dplyr’

## The following objects are masked from ’package:stats’:
##
##     filter, lag

## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
library("knitr") # For kable

data <- read_csv(("01-Data.csv"))
```

```
## Rows: 891 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Name, Sex, Ticket, Cabin, Embarked
## dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
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

```
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  7.2500  <NA>        S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250  <NA>        S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500  <NA>        S
## 6           330877  8.4583  <NA>        Q
```

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

```
## [1] "True mean"        "29.6991176470588"
```

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 <- 30
sample_age <- sample(age_data, n, replace = FALSE)

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

```
## [1] "Sample mean"      "28.8916666666667"
```

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,

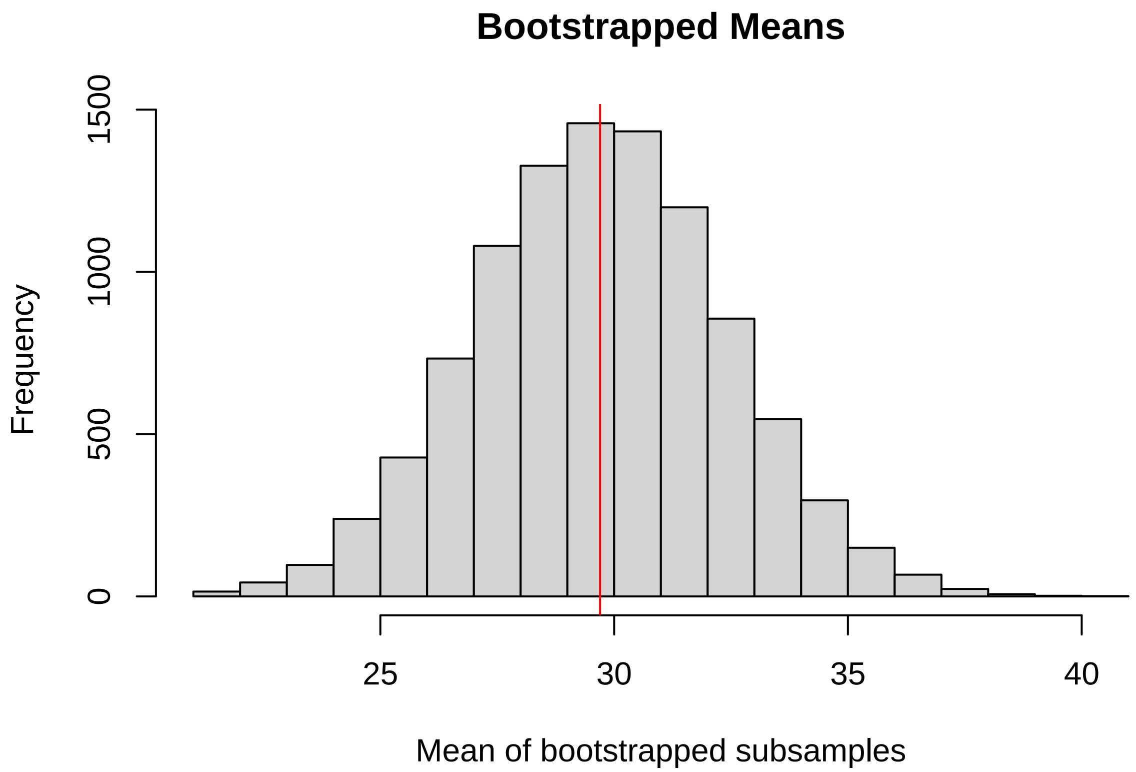
```r
bootstrap <- function(age_data, mean_subsamples){
  subsample <- sample(age_data, n, replace = TRUE)
  mean_subsamples <- append(mean_subsamples, mean(subsample))
  return(mean_subsamples)
  }

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

hist(bootstrap_means, breaks = 20, xlab = "Mean of bootstrapped subsamples", ylab = "Frequency", main =
abline(v = mean(age_data), col = "red", label = "true population mean")
```

```
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "label" is
## not a graphical parameter
```

```
#text(locator(), labels = "True pop mean")
coords <- locator()
```



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

```
## [1] "The True population mean is" "29.6991176470588"
```

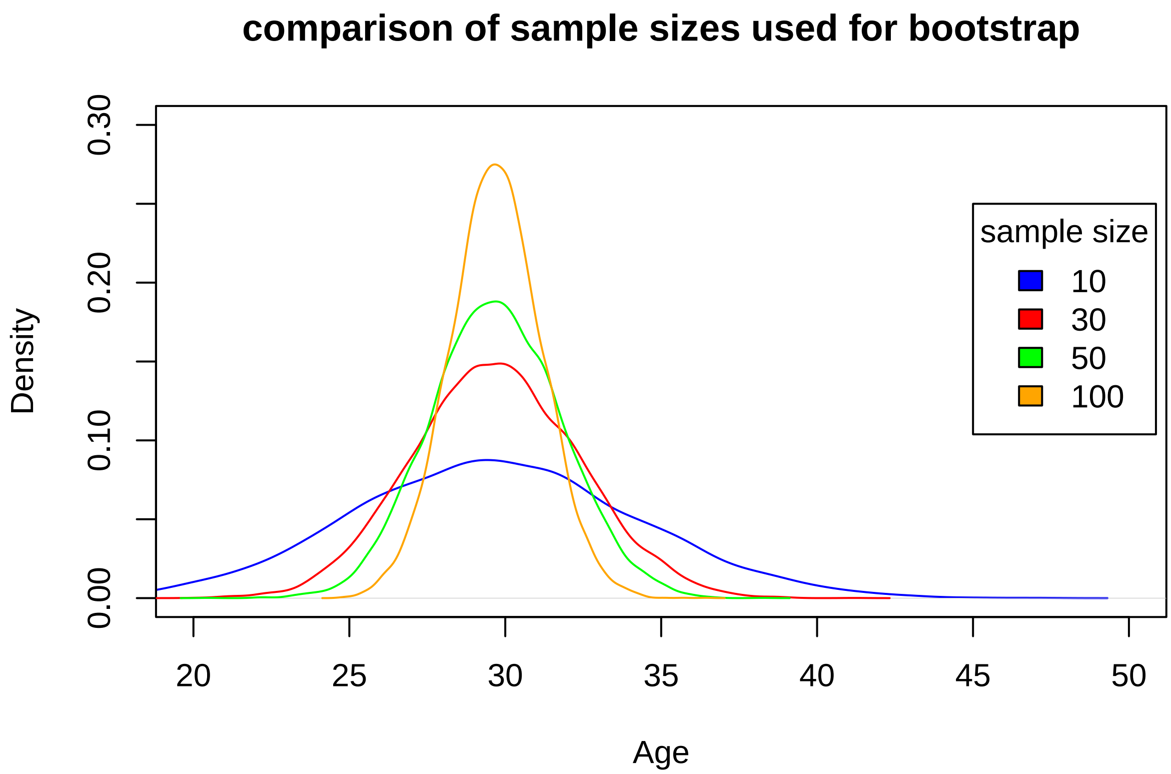
```
c("Our bootstrapped estimate of the population mean from a random sample is",mean(bootstrap_means))
```

```
## [1] "Our bootstrapped estimate of the population mean from a random sample is"
## [2] "29.7196960666667"
```

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

```
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(age_data, mean_subsamples){
    subsample <- sample(age_data, 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(age_data, bootstrap_means)
  }
  if(j == 1){
    plot(density(bootstrap_means), xlab = "Age", xlim = c(20,50),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.