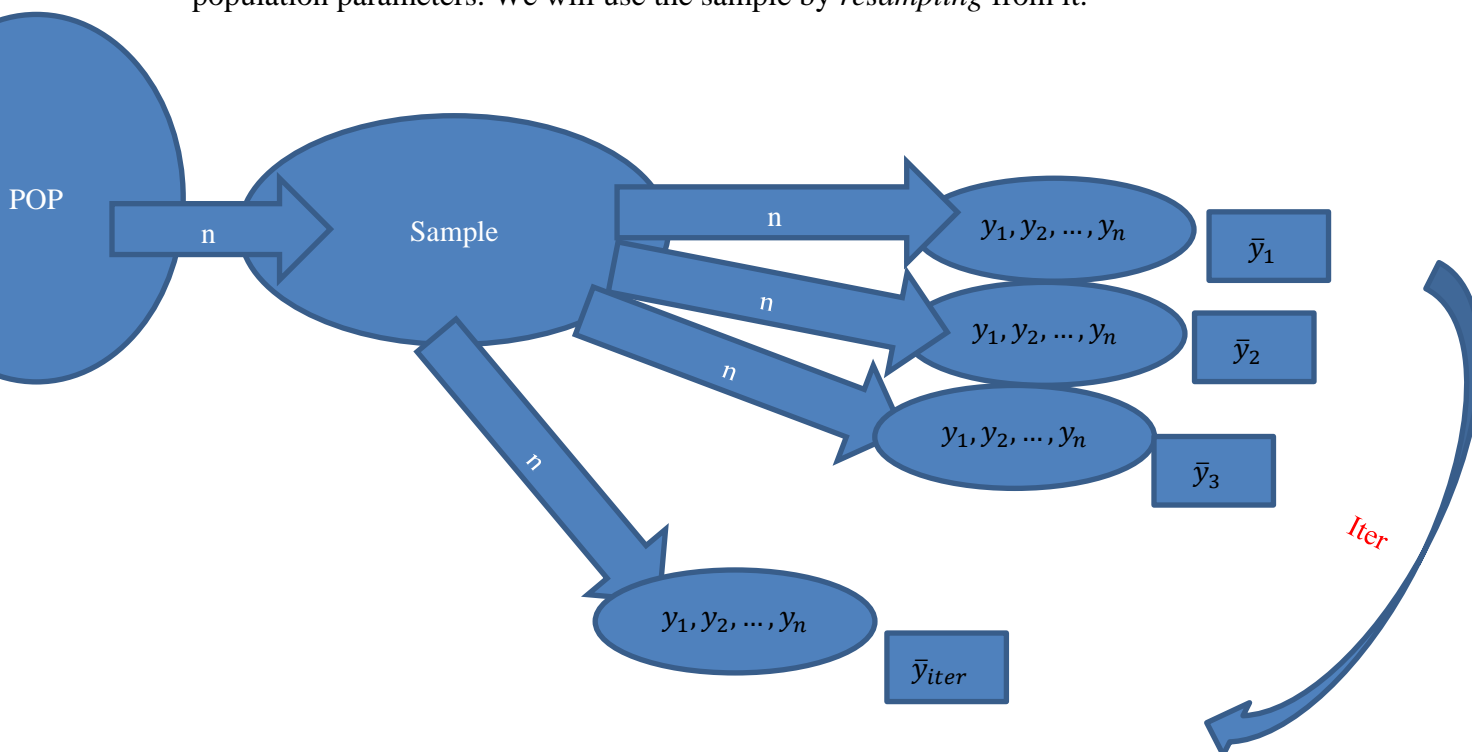# MATH 4753 Laboratory 9 BOOTSTRAP and confidence intervals

In this lab we will learn about two types of estimate:

1. Point

2. Interval

These estimates are important and will be studied in detail as we progress through chapter 7 of MS. We will use a simulation technique called the bootstrap. This gets its name from the idea that we derive everything from what we have at hand – that is we pull ourselves up from our bootstraps. Specifically this means that we take one sample from the population of interest and assume that the sample is representative, that is, it contains information that is summative of the population parameters. We will use the sample by *resampling* from it.

POP

n

Sample

n

$y_1, y_2, \ldots, y_n$

$\bar{y}_1$

n

$y_1, y_2, \ldots, y_n$

$\bar{y}_2$

n

$y_1, y_2, \ldots, y_n$

$\bar{y}_3$

n

$y_1, y_2, \ldots, y_n$

$\bar{y}_{iter}$

Iter

The R function `sample()` will be used to do the resampling. If the sample is of size n then the bootstrap re-sampling will create samples of size n *with replacement*. The following is a function that will facilitate the bootstrap procedure.

```r
myboot<-function(iter=10000,x,fun="mean",alpha=0.05,...){

#Notice where the ... is repeated in the code
n=length(x)    #sample size

#Now sample with replacement
y=sample(x,n*iter,replace=TRUE) #A

# Make a matrix with all the resampled values
rs.mat=matrix(y,nr=n,nc=iter,byrow=TRUE)
xstat=apply(rs.mat,2,fun)
# xstat is a vector and will have iter values in it
ci=quantile(xstat,c(alpha/2,1-alpha/2)) #B
# Nice way to form a confidence interval
# A histogram follows
#  The object para will contain the parameters used to make the
histogram
para=hist(xstat,freq=FALSE,las=1,main="Histogram  of  Bootstrap  sample
statistics",...)

#mat will be a matrix that contains the data, this is done so that I
can use apply()
mat=matrix(x,nr=length(x),nc=1,byrow=TRUE)

#pte is the point estimate
#This uses whatever fun is
pte=apply(mat,2,fun)
abline(v=pte,lwd=3,col="Black")# Vertical line
segments(ci[1],0,ci[2],0,lwd=4)        #Make the segment for the ci
text(ci[1],0,paste("(",round(ci[1],2),sep=""),col="Red",cex=3)
text(ci[2],0,paste(round(ci[2],2),")",sep=""),col="Red",cex=3)

# plot the point estimate 1/2 way up the density
text(pte,max(para$density)/2,round(pte,2),cex=3)

return(list(ci=ci,fun=fun,x=x))# Some output to use if necessary
}
```
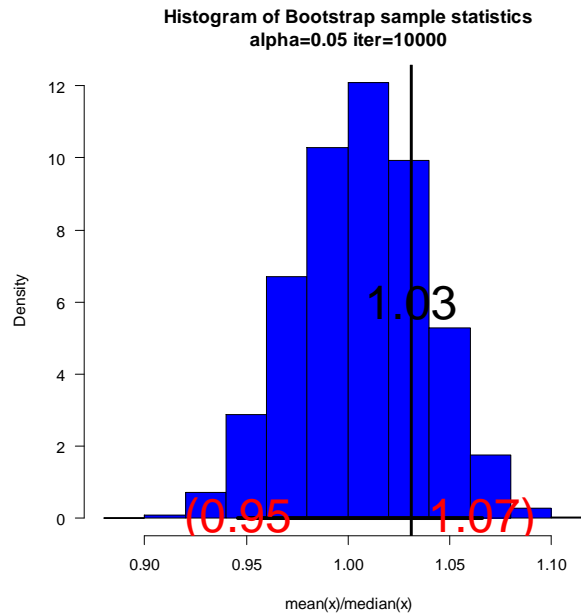
The above code will be used to make a number of plots, point and interval estimates and useful output in the form of a list.

Histogram of Bootstrap sample statistics
alpha=0.05 iter=10000

1.03

(0.95   1.07)

Density

mean(x)/median(x)

Tasks
All output should be made through RMD. Make sure you knit to html and upload to CANVAS

Note: All plots you are asked to make should be recorded through RMD.

- Task 1
    - Make a folder LAB9
    - Download the file "lab9.r"
    - Place this file with the others in LAB9.
    - Start Rstudio
    - Open "lab9.r" from within Rstudio.
    - Go to the "session" menu within Rstudio and "set working directory" to where the source files are located.
    - Issue the function `getwd()`

- Task 2
    - Create your own R file and record the R code you used to complete the lab.
    - In the above code for `myboot()` there are two lines marked A and B. Using any resources available explain what each line does
        - Line A
        - Line B
    - The sample() function should be studied a little further. As used in the myboot() function, each datum in x will be selected with equal probability. Why is this necessary?
    - Issue the following lines in R and record the unique sample you get
        - `set.seed(35) # This will give everyone the same sample`
        - `sam=round(rnorm(20,mean=10,sd=4),2)`

- unique(sample(sam,20,replace=TRUE)) # repeat this line 5Xs
- Explain what you see.
- unique(sample(sam,20,replace=FALSE)) # repeat this line 5Xs
- Explain what you see.
  - Issue sample(sam,21,replace=FALSE) what happens? Why?
- Task 3
  - Using myboot() make 95% (alpha=0.05) bootstrap intervals (iter=10000) for the population mean μ and record the plots when the following samples are used:
    - A) set.seed(39); sam=rnorm(25,mean=25,sd=10)
    - B) set.seed(30); sam=rchisq(20,df=3)
    - C) set.seed(40); sam=rgamma(30,shape=2,scale=3)
    - D) set.seed(10); sam=rbeta(20,shape1=3,shape2=4)
  - In each of the above cases how close is the point estimate to the population value? HINT: You will need to calculate the population mean.
  - In each of the above cases does the interval contain the population value? (HINT: You will need to calculate the population mean. See MS chapter 5 or use Wikipedia)
  - Using myboot() make 80% (alpha=0.20) bootstrap intervals (iter=10000) for the population variance σ^2 and record the plots when the following samples are used:
    - A) set.seed(39); sam=rnorm(25,mean=25,sd=10)
    - B) set.seed(30); sam=rchisq(20,df=3)
    - C) set.seed(40); sam=rgamma(30,shape=2,scale=3)
    - D) set.seed(10); sam=rbeta(20,shape1=3,shape2=4)
- Task 4
  - Adjust myboot() so that it returns as a part of the list the vector containing the statistic (xstat) of interest along with all the other existing quantities.
  - Using the adjusted myboot() function call myboot(x=sam,fun=``median''), where sam=c(1,1,1,2,2,2,2,3,3,3,4,4) - make a barplot of xstat.
  - What is the bootstrap interval estimate for the median? (L,U)
- Task 5
  - Find a 95% interval estimate for the population mean/median (mean divided by median) using each of the following samples
    - A) set.seed(39); sam=rnorm(25,mean=25,sd=10)
    - B) set.seed(30); sam=rchisq(20,df=3)
    - C) set.seed(40); sam=rgamma(30,shape=2,scale=3)
    - D) set.seed(10); sam=rbeta(20,shape1=3,shape2=4)
  - Do the same except make 70% intervals.
- Task 6
  - Issue the command ?distributions in R
  - Choose four distributions that we haven't used this far and make one sample of size 20 from each.
  - Make 80% bootstrap intervals for the mean and variance using each sample. Use iter=10000.
  - Record the graphical output.

- Task 7

- o Use `myboot()` and create bootstrap intervals using two statistics that you find interesting. Use `set.seed(68); sam=rnorm(20,mean=10,sd=4)` as the sample. (You might try some functions like `quantile, IQR, sd, var, mean, median`)

- o Theory demands that if $Y \sim N(\mu, \sigma), then\, P\left(\bar{Y} - z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{Y} + z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha$.
  This defines $\bar{y} - z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{y} + z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}$ to be the $(1 - \alpha)100\%$ confidence interval, where $z_{\frac{\alpha}{2}} =$`qnorm(1-alpha/2,mean=0,sd=1)`, that is the $1 - \alpha/2$ standard normal quantile.

- o Calculate the 95% confidence interval using the above theory using the sample `set.seed(68); sam=rnorm(20,mean=10,sd=4)`

- o Use `myboot()` with the same sample to find a 95% bootstrap interval using the same sample. (Decide on the options you need).

- o How do they compare?

- Task 8
  - o Add myboot2() to your package (make sure it is well documented)
  - o Use whatever data you have installed in your package to create a bootstrap ci obj=ILAS2019::myboot2(x=ddt$DDT)

################## LAB FINISHES HERE ###############################

- Task 9 Extra for experts!
  - o Rewrite myboot() so that it creates interesting plots