Michael Gleyzer, Mathew Witek and Harish Pillai
Assignment 6
10/31/2019

```
hurricane.mat <- read.csv('C:\\Users\\mgleyzer\\Downloads\\Hurricane cat 4 and 5
corrected.csv' )
# delete the NA values in the first two rows
hurricane.mat1 <- hurricane.mat[-c(1,2),]
# function takes the hurricane.mat1, fits a linear model,
And makes a prediction based on given bootstrapped values of the statistic
lmfunc <- function(vec0,mat=hurricane.mat1){
  # Coerce hurricane.mat1 into a data frame
  mat00<-as.data.frame(hurricane.mat1)
  # Make a data frame to be used for the linear model fitting
  # The rows of the matrix to be used in the linear fit
# Determined by the bootstrapped indices
  mat0<-as.data.frame(hurricane.mat1[vec0,])
  # Fit a linear model with predictors(AMO , decade)
  # and response variable as cat45, using the mat0 data
  lm.str<-lm(cat45~AMO+decade,data=mat0)
  # Get the 13th row which corresponds to the 1990 decade
  predict.mat<-mat00[c(13,13) , ]
  # Predict new response values using fitted linear model and prediction
  # matrix
out<-predict(lm.str,predict.mat)
# Return the column with the predicted response variable
out[1]}

Jackknife<-function(vec0,statfunc=sd){
  # Assign the length of vec0 to n1
  n1<-length(vec0)
  # Initialize the variable jackvec
  jackvec<-NULL
  # Calculate the initial standard deviation of vec0
  mu0<-statfunc(vec0)
  # Iterate the length of vec0
  for(i in 1:n1){
    # Find the standard deviation of
    # vec0, when omitting the ith element
    mua<-statfunc(vec0[-i])
    # Create vector jackvec composed of jackknifed elements.
    # Those elements represent the bootstrapped values' individual effects on standard
deviation
```

```
  jackvec<-c(jackvec, n1*(mu0)-(n1-1)*mua)}
 # Find the bias of each jackknife estimate compared to the original
 # estimate
 jackmean<- mean(jackvec)
 jackbias<-mean(jackvec)-mu0
 # Find the standard deviation of the jackknifed effect estimates
 jacksd<-sd(jackvec)
 # List out all the results
 list(mu0=mu0,jackbias=jackbias,jacksd=jacksd)}


My.bootstrap.ci<-function(data.str,nboot=10000,alpha=0.05, stat.func = lmfunc){
 #Assign length of the data(no. of rows) to n0
 n0<-length(data.str[,1])
 # Make vec0 the vector holding the indices of the rows
 vec0<-c(1:n0)
 #Assign to mean0 the value returned by the lmfunc function
 mean0<-lmfunc(vec0)
 #Find the standard deviation of the data using the jackknife function. Assign it to
variable
 #sd0
 sd0<-Jackknife(vec0 , statfunc = stat.func) $ jacksd
 #Initialize the bootvec object
 bootvec<-NULL
 #Iterate 10000 times
 for( i in 1:nboot){
  #Create bootstrap vector of data
  #By sampling with replacement from given vector v0
  vecb<-sample(vec0,replace=T)
  #Apply the linear model function to the bootstrap vector
  meanb<-stat.func(vecb)
  #Find the standard deviation of the bootstrap vector
  sdb<-Jackknife(vecb , statfunc = stat.func) $ jacksd
  #Bootstrap vector of test statistics(to be used for confidence intervals)
  bootvec<-c(bootvec,(meanb-mean0)/(sdb/sqrt(n0)))}
 #Determine lower quantile of bootvec test statistic data
 lq<-quantile(bootvec,alpha/2)
 #Determine upper quantile of bootvec test statistic data
 uq<-quantile(bootvec,1-alpha/2)
 #Find lower bound for normal distribution confidence interval
 LB<-mean0-(sd0/sqrt(n0))*uq
 #Find upper bound for normal distribution confidnce interval
 UB<-mean0-(sd0/sqrt(n0))*lq
```

```
 #Find bootstrapped lower bound for confidence interval
 NLB<-mean0-(sd0/sqrt(n0))*qt(1-alpha/2,n0-1)
 #Find bootstrapped upper bound for confidence interval
 NUB<-mean0+(sd0/sqrt(n0))*qt(1-alpha/2,n0-1 )
 #Store both confidence intervals(normal theory, bootstrapped) in a list
 list(bootstrap.confidence.interval=c(LB,UB),normal.confidence.interval=c(NLB,NUB))}
```

```
My.bootstrap.ci(hurricane.mat1 , nboot= 10000, alpha = 0.05, stat.func = lmfunc)
```

Results:

```
$bootstrap.confidence.interval
      15       15
14.13991 18.38291

$normal.confidence.interval
      15       15
13.75571 18.10972
```