**Name: Raj Shah**

**PSID: 1499521**

**Assignment 6**

Prob 1) I am looking to make inference about the mean and median miles per gallon for cars in mtcars dataset.

Code:

*library(datasets) #Importing library*

*data(mtcars)*

*mpg<-mtcars$mpg*

*m<-10000*

*my.boot<-function(x,f,b){*

*theta.hat<-f(x)*

*B<-b*

*theta.hat.b<-numeric(B)*

*for (t in 1:B){*

*i <- sample(1:length(x), replace = TRUE)*

*theta.hat.b[t]<-f(x,i)*

*}*

*#return(R)*

*theta.star<-(1/B)\*sum(theta.hat.b)*

*se<-sqrt((1/(B-1))\*sum((theta.hat.b-theta.star)^2))*

*bi<-theta.star-theta.hat*

*ci.norm<-f(theta.hat.b)+c(-1,1)\*qnorm(.975)\*se*

*ci.basic<-theta.star+quantile((theta.hat.b-theta.hat),c(0.025,0.975))*

*ci.percentile<-quantile(theta.hat.b,c(0.025,0.975))*

*result<-matrix(c(bi,se,ci.norm,ci.basic,ci.percentile),ncol=8)*

*colnames(result)<-c("Bias","Std.Error","CI Norm L","CI Norm U","CI Basic L","CI Basic U","CI Perc L","CI Perc U")*

*result<-as.table(result)*

*return(result)*

*#return(c(se,bi,ci.norm,ci.basic,ci.percentile))*

*}*

*my.median<-function(x,indices){*

*return(median(x[indices]))*

*}*

*my.mean<-function(x,indices){*

*return(mean(x[indices]))*

*}*

> my.boot(mpg,my.mean,m)

Bias Std.Error CI Norm L CI Norm U CI Basic L

A -0.00422625 1.04832748 18.03171465 22.14108285 18.07694563

CI Basic U CI Perc L CI Perc U

A 22.20835187 18.08117188 22.21257812

> R.boot<-boot(mpg,statistic=function(x,i) mean(x[i]),R=m)

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = mpg, statistic = function(x, i) mean(x[i]), R = m)

Bootstrap Statistics :

original bias std. error

t1\* 20.09062 -0.0023975 1.052899

> R.boot.ci<-boot.ci(R.boot)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 10000 bootstrap replicates

CALL :

boot.ci(boot.out = R.boot)

Intervals :

Level Normal Basic

95% (18.03, 22.16 ) (18.00, 22.09 )

Level Percentile BCa

95% (18.09, 22.18 ) (18.18, 22.31 )

Calculations and Intervals on Original Scale

> t.test(mpg,mu=0)

One Sample t-test

data: mpg

t = 18.857, df = 31, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

17.91768 22.26357

sample estimates:

mean of x

20.09062

> my.boot(mpg,my.median,m)

Bias Std.Error CI Norm L CI Norm U CI Basic L CI Basic U

A 0.097485 1.267302 16.716133 21.683867 16.647485 21.497485

CI Perc L CI Perc U

A 16.550000 21.400000

> R.boot<-boot(mpg,statistic=function(x,i) median(x[i]),R=m)

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = mpg, statistic = function(x, i) median(x[i]), R = m)

Bootstrap Statistics :

original bias std. error

t1\* 19.2 0.09366 1.2632

> R.boot.ci<-boot.ci(R.boot)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 10000 bootstrap replicates

CALL :

boot.ci(boot.out = R.boot)

Intervals :

Level Normal Basic

95% (16.63, 21.58 ) (17.00, 21.60 )

Level Percentile BCa

95% (16.80, 21.40 ) (15.95, 21.20 )

Calculations and Intervals on Original Scale

No,T.test Cannot be done for median as it is performed for the true value of mean to check the null hypothesis

Condition.

Prob 2)

Code:

*library(ISwR)*

*data("cystfibr")*

*attach(cystfibr)*

*n<-nrow(cystfibr)*

*cv.err.1<-cv.err.2<-cv.err.3<-numeric(n)*

*for (j in 1:n){*

*y<-pemax[-j];a<-age[-j];s<-sex[-j];w<-weight[-j];b<-bmp[-j];f<-fev1[-j];r<-rv[-j];fr<-frc[-j];t<-tlc[-j]*

*m1<-lm(y~a+s+w+b+f+r+fr+t,data=cystfibr) #model 1*

*yhat1<-m1$coefficients[1]+m1$coefficients[2]\*age[j]+m1$coefficients[3]\*sex[j]+m1$coefficients[4]\*weight[j]+m1$coefficients[5]\*bmp[j]+m1$coefficients[6]\*fev1[j]+m1$coefficients[7]\*rv[j]+m1$coefficients[8]\*frc[j]+m1$coefficients[9]\*tlc[j]*

*cv.err.1[j]<-pemax[j]-yhat1*

*h<-height[-j]*

*m2<-lm(y~a+s+h+w+b,data=cystfibr) #model 2*

*yhat2<-m2$coefficients[1]+m2$coefficients[2]\*age[j]+m2$coefficients[3]\*sex[j]+m2$coefficients[4]\*height[j]+m2$coefficients[5]\*weight[j]+m2$coefficients[6]\*bmp[j]*

*cv.err.2[j]<-pemax[j]-yhat2*

*m3<-lm(y~a+s+f+r+fr+t,data=cystfibr) #model 3*

*yhat3<-m3$coefficients[1]+m3$coefficients[2]\*age[j]+m3$coefficients[3]\*sex[j]+m3$coefficients[4]\*fev1[j]+m3$coefficients[5]\*rv[j]+m3$coefficients[6]\*frc[j]+m3$coefficients[7]\*tlc[j]*

*cv.err.3[j]<-pemax[j]-yhat3*

*}*

*print(c(mean(cv.err.1^2),mean(cv.err.2^2),mean(cv.err.3^2)))*

print(c(mean(cv.err.1^2),mean(cv.err.2^2),mean(cv.err.3^2)))

[1] 981.6799 899.6516 846.9905

**Model 3** is the best as the error is least among the three models.

Prob 3)

Analytical derivation

F(x)=1-(b/x)^a

U=1-(b/x)^a

b^a/x^a=1-u

x=b/((1-u)^a)

f(x)=d/dx(F(x))

f(x)=(2\*b^a/x^3)

Code:

*library("rmutil")*

*n<-1000*

*a<-2*

*b<-2*

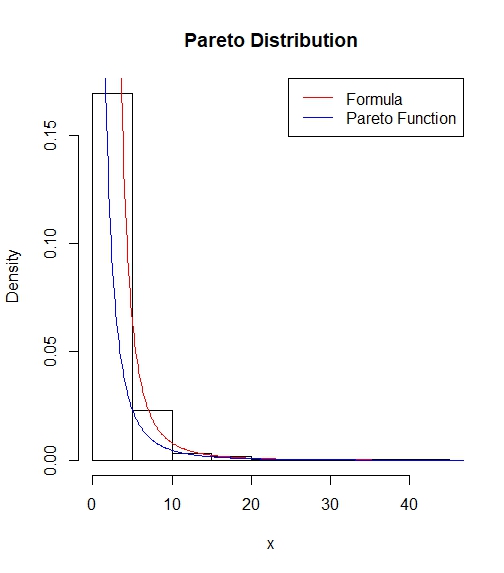
*x<-b/((1-runif(n))^(1/a))*

*hist(x,probability = TRUE,main = "Pareto Distribution")*

*y<-seq(0.1,100,.1)*

*lines(y,(2\*b^a)/(y^3)) #The formula Equation*

*lines(y,dpareto(y,m=a,s=b)) #Comparing with the original function*



Prob 4)

Code:

*m<-10000*

*beta.mc<-function(m,a,b,x){*

*z<-rbeta(m,shape1 = a,shape2 = b)*

*g<-(z<x)*

*cdf<-mean(g)*

*return(cdf)*

*}*

*est.beta<-vector()*

*x<-seq(0.1,0.9,length=9)*

*for (i in x){*

*est.beta<-append(est.beta,beta.mc(m,2,2,i)) #Estimated Beta Numbers*

*}*

*real.beta<-pbeta(x,2,2) #Real Beta Numbers*

*print(round(rbind(x,est.beta,real.beta),3))*

*rm(est.beta)*

> beta.mc(m,2,2,0.1)

[1] 0.0285

> print(round(rbind(x,est.beta,real.beta),3))

[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]

x 0.100 0.200 0.300 0.400 0.500 0.600 0.700 0.800 0.900

est.beta 0.027 0.106 0.219 0.349 0.505 0.651 0.773 0.897 0.972

real.beta 0.028 0.104 0.216 0.352 0.500 0.648 0.784 0.896 0.972