**6304 Module 3 Live Lecture**

**R Script File**

**rm(list=ls())**

**library(rio)**

**gpa50=import("Module 3 Data Sets.xlsx",**

**sheet="GPAs 50 Stacked")**

**colnames(gpa50)=tolower(make.names(colnames(gpa50)))**

**attach(gpa50)**

**t.test(gpas50,mu=2.1,alternative = c("two.sided"))**

**results=t.test(gpas50,mu=2.1,alternative = c("two.sided"))**

**results**

**names(results)**

**results$p.value**

**results$conf.int**

**results$conf.int[1]**

**results$conf.int[2]**

**width=results$conf.int[2]-results$conf.int[1]**

**width**

**# t Distribution**

**curve(dnorm(x,0,1),from=-4,to=4,col="red",lwd=3,**

**main="Comparing Normal and t Distributions")**

**curve(dt(x,2),from=-4,to=4,lwd=3,add=TRUE)**

**abline(v=-1,lwd=3,col="blue")**

**pnorm(-1,0,1)**

**pt(-1,2)**

**# How're the Confidence Intervals Different?**

**curve(dnorm(x,0,1),from=-4,to=4,col="red",lwd=3,**

**main="Comparing Confidence Intervals,**

**Normal and t")**

**curve(dt(x,2),from=-4,to=4,lwd=3,add=TRUE)**

**qnorm(.025,0,1)**

**# Z Confidence Intervals**

**abline(v=qnorm(.025,0,1),lwd=3,col="blue")**

**abline(v=qnorm(.975,0,1),lwd=3,col="blue")**

**qt(.025,2)**

**# t Confidenced Intervals, df=2**

**abline(v=qt(.05,2),lwd=3,col="dark green")**

**abline(v=qt(.95,2),lwd=3,col="dark green")**

**# How does the t distribtuion change shape?**

**curve(dnorm(x,0,1),from=-4,to=4,lwd=3,col="red",**

**main="One Normal and Several t Distributions")**

**for(i in 2:10) {**

**curve(dt(x,i),from=-4,to=4,lwd=3,add=TRUE)**

**}**

**# How do confidence interval widths change**

**# with higher df?**

**for(i in 2:10){**

**abline(v=qt(.025,i),lwd=3,col="dark green")**

**abline(v=qt(.975,i),lwd=3,col="dark green")**

**}**

**abline(v=qnorm(.025,0,1),lwd=3,col="red")**

**abline(v=qnorm(.975,0,1),lwd=3,col="red")**

**# Changing data sets for small sample size.**

**GPA15=import("Module 3 Data Sets.xlsx",sheet="GPAs 15 Stacked")**

**# Properly format column names.**

**colnames(GPA15)=tolower(make.names(colnames(GPA15)))**

**# Confidence interval output dumped to screen.**

**t.test(GPA15$student.gpa)**

**# Conduct a t test and place output in an object.**

**ginger=t.test(GPA15$student.gpa)**

**ginger**

**names(ginger)**

**ginger$conf.int**

**ginger$conf.int[1]**

**# Comparing t values v. Z values?**

**# These are t values used for confidence intervals.**

**# Arguments are one-tail area of coverage and degrees of freedom.**

**qt(.975,14)**

**qt(.975,nrow(GPA15)-1)**

**qnorm(.975)**

**qt(.975,10)**

**qt(.975,100)**

**qt(.975,1000)**

**qt(.975,10000)**

**qt(.975,100000)**

**qt(.975,1000000)**

**# Looking at a graph of t and Z values.**

**# t values quickly approach Z, but never are precisely equal.**

**# Asymptotic**

**maryann=data.frame()**

**for(i in 1:10000){**

**maryann[i,1]=i**

**maryann[i,2]=qt(.975,i)**

**}**

**colnames(maryann)=c("n","t.value")**

**plot(maryann,type="l",col="red",ylim=c(0,5),**

**main="t values v. Z values by Sample Size")**

**abline(h=qnorm(.975,0,1),col="blue")**

**qnorm(.975,0,1)**

**maryann[nrow(maryann),2]**

**# Hypothesis Test in R**

**# Sewer Pipe Data**

**library(rio)**

**pipe=import("Module 3 Data Sets.xlsx",sheet="Sewer Pipe")**

**colnames(pipe)=tolower(make.names(colnames(pipe)))**

**mean(pipe$breaking.point)**

**sd(pipe$breaking.point)**

**hist(pipe$breaking.point,col="red",**

**main="Sample Pipe Breaking Points")**

**plot(density(pipe$breaking.point),lwd=3,**

**main="Density Plot, Sample Pipe Breaking Points")**

**abline(v=mean(pipe$breaking.point),lwd=3,col="blue")**

**abline(v=2400,col="red",lwd=3)**

**results=t.test(pipe$breaking.point,**

**mu=2400,alternative = "greater")**

**results**

**results=t.test(pipe$breaking.point,**

**mu=2400,alternative = "two.sided")**

**results**

**# Hypothesis Test**

**# GPA Data**

**big.set=import("Module 3 Data Sets.xlsx",**

**sheet="GPAs 50 Stacked")**

**small.set=import("Module 3 Data Sets.xlsx",sheet="GPAs 15 Stacked")**

**hist(big.set$GPAs50,col="red",xlim=c(0,4),**

**main="Grade Point Averages",**

**xlab="Range of GPAs at USF")**

**hist(small.set$`Student GPA`,col="blue",add=TRUE)**

**abline(v=mean(big.set$GPAs50),col="green",lwd=3)**

**abline(v=mean(small.set$`Student GPA`),col="orange",lwd=3)**

**results50=t.test(big.set$GPAs50)**

**results50**

**results50=t.test(big.set$GPAs50,mu=2.7,**

**alternative = "greater")**

**results50**

**results50=t.test(big.set$GPAs50,mu=2.75,**

**alternative = "greater")**

**results50**

**results50=t.test(big.set$GPAs50,mu=2.8,**

**alternative = "greater")**

**results50**

**results50=t.test(big.set$GPAs50,mu=2.9,**

**alternative = "greater")**

**results50**

**results50=t.test(big.set$GPAs50,mu=mean(big.set$GPAs50),**

**alternative = "greater")**

**results50**

**gilligan=mean(big.set$GPAs50)-sd(big.set$GPAs50)**

**maryann=mean(small.set$`Student GPA`)-**

**sd(small.set$`Student GPA`)**

**results50=t.test(big.set$GPAs50,**

**mu=gilligan,alternative = "greater")**

**results15=t.test(small.set$`Student GPA`,**

**mu=maryann,alternative="greater")**

**results50**

**results15**

**my.p.values=data.frame(matrix(ncol=3,nrow=0))**

**colnames(my.p.values)=c("test","p50","p15")**

**for(i in seq(1,400,)) {**

**gilligan=t.test(big.set$GPAs50,mu=i/100,**

**alternative="greater")**

**maryann=t.test(small.set$`Student GPA`,mu=i/100,alternative="greater")**

**my.p.values[i,1]=i/100**

**my.p.values[i,2]=gilligan$p.value**

**my.p.values[i,3]=maryann$p.value**

**}**

**plot(my.p.values[,1],my.p.values[,2],type="l",**

**lwd=3,col="red",**

**xlab=c("Test Value"),ylab=c("p Value"),**

**main=c("Changing p Values"))**

**points(my.p.values[,1],my.p.values[,3],type="l",**

**lwd=3,col="blue")**

**abline(.05,0,lwd=3)**

**abline(v=mean(big.set$GPAs50),col="red",lwd=3)**

**abline(v=mean(small.set$`Student GPA`),col="blue",lwd=3)**

**abline(.5,0,lwd=3)**

**# Comparing Means of Two Populations**

**# Independent Sampling**

**# New Data Set – IQ Data**

**iq=import("Module 3 Data Sets.xlsx",sheet="IQ")**

**colnames(iq)=tolower(make.names(colnames(iq)))**

**attach(iq)**

**results=t.test(age.25,age.60,mu=0,**

**alternative = c("two.sided"))**

**results**

**# Comparing Means of Two Populations**

**# Paired Comparisons OR Paired Differences**

**# New Data Set – IQ Data**

**rats=import("Module 3 Data Sets.xlsx",sheet="Rat Pups")**

**colnames(rats)=tolower(make.names(colnames(rats)))**

**attach(rats)**

**results=t.test(male,female,mu=0,**

**alternative=c("two.sided"),paired=TRUE)**

**results**

**grocery=import("Module 3 Data Sets.xlsx",**

**sheet="Grocery")**

**colnames(grocery)=tolower(make.names(colnames(grocery)))**

**attach(grocery)**

**names(grocery)**

**str(grocery)**

**levels(division)**

**division=as.factor(division)**

**str(grocery)**

**levels(division)**

**fairview=subset(grocery,division=="Fairview")**

**summerfield=subset(grocery,division=="Summerfield")**

**set.seed(99)**

**my.fairview=fairview[sample(1:nrow(fairview),18,**

**replace=FALSE),]**

**attach(my.fairview)**

**set.seed(99)**

**my.summerfield=summerfield[sample(1:nrow(summerfield),15,**

**replace=FALSE),]**

**attach(my.summerfield)**

**t.test(my.fairview$customer.penetration,mu=.2,**

**alternative=c("greater"))**

**t.test(my.fairview$customer.penetration,mu=.22,**

**alternative=c("greater"))**

**t.test(my.fairview$deli.sq.ft,my.summerfield$deli.sq.ft,**

**mu=0,alternative=c("two.sided"))**

**t.test(my.summerfield$deli.sales,my.fairview$deli.sales,**

**mu=35000,alternative=c("greater"))**

**my.p.values=data.frame(matrix(ncol=2,nrow=0))**

**colnames(my.p.values)=c("test","p")**

**for(i in 1:1600) {**

**results=t.test(my.fairview$deli.sq.ft,mu=2000+i,**

**alternative=c("two.sided"))**

**x=results$p.value**

**my.p.values[i,1]=i+2000**

**my.p.values[i,2]=x**

**}**

**plot(my.p.values,type="l",lwd=3,col="red",**

**xlab=c("Test Value"),ylab=c("p Value"),**

**main=c("Changing p Values"))**

**abline(.05,0,lwd=3)**

**abline(v=mean(my.fairview$deli.sq.ft),col="blue",lwd=3)**

**boxplot(iq$age.25,iq$age.60,col="red",main="IQ Boxplot")**

**boxplot(iq$age.25,iq$age.60,notch=TRUE,col="red",**

**main="IQ Notched Boxplot")**

**boxplot(male,female,notch=TRUE,col="red",**

**main="Rat Pups Boxplot")**