STOR 455 Class 8 Inference for Regression SLope

library(readr)  
library(Stat2Data)  
library(metRology)  
  
DistanceHome <- read\_csv("https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/DistanceHome.csv")  
Domestic=subset(DistanceHome,Distance<250)  
  
source("https://raw.githubusercontent.com/JA-McLean/STOR455/master/scripts/CIPIPlot.R")   
# Function that is not built into the thing   
# The source will allow you to call the function as you want

*NOtes* - If the data is packed around the line, then we can predict that the data will be by the line

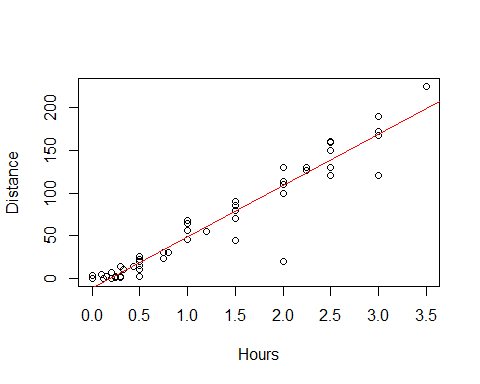
**Inference for Slope and Intercept** - Find a confidence interval of plausible values for the parameter. - Test a hypothesis about a possible value for the parameter

**Bootstrap Distribution** - THink of poopulation - Estimate the distribution and variability (SE) of β ̂\_𝑖 from the bootstraps - WE ar elooking atht the distnace data, - THink of the population and there is some population slope here - If I knew everyone’s distance from home, this would be the realsiton they have; we know the sample but not the population - It would be nice if we could take a lot of samples ad keep calculating that regression line (then you could see teh distribution) - Think of seeds that fall from a tree; from each seed that falls from teh tree we get a slope from that line. - We cant keep taking from teh ample, and we need to find where the actual population vlaue is

* Hypothesis what the population might look like based on this sample
* Maybe the population is just a lot of copies of this sample
* If we take samples form teh bootstrap poulation, what variability is there?
* We are making many copies of them
* We just have out 54 key people with our data, taking one out and seeing where they fall on the regression line, putting htem back and then pulling someone else – If you do this a bunch of times, sometimes you ll get a sample that is like the population or not; youll get a range of values

**Simple Linear Regression** *See below*

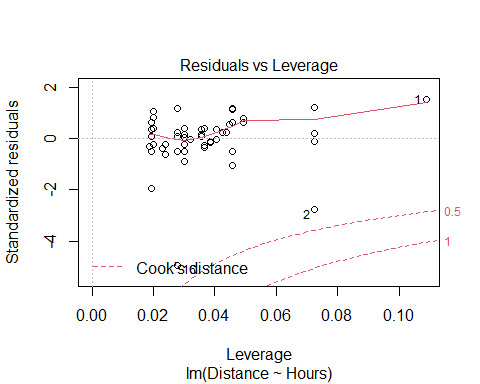
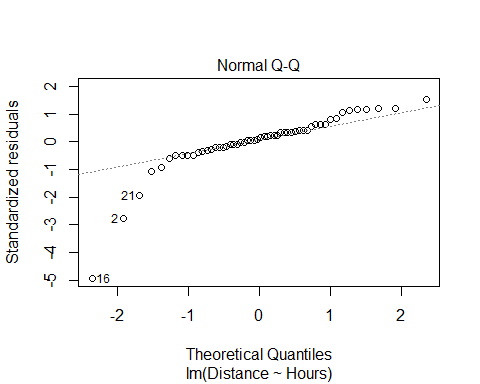
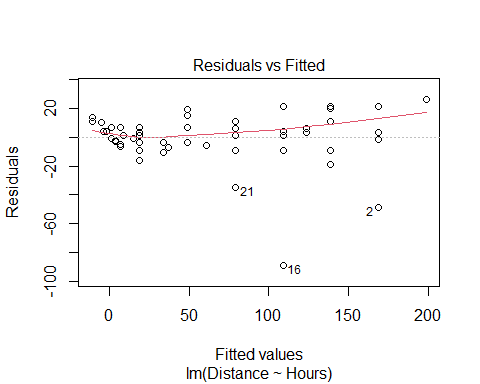
moddist=lm(Distance~Hours, data=Domestic)  
# Hours per distnace; predict distance form home based on hours it takes you to get home   
  
plot(Distance~Hours, data=Domestic)  
abline(moddist, col="red")



summary(moddist)

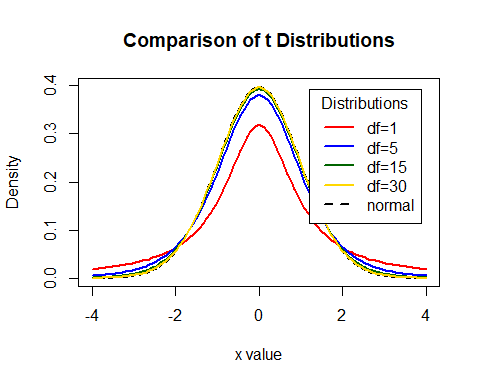
##   
## Call:  
## lm(formula = Distance ~ Hours, data = Domestic)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -88.892 -4.680 2.172 7.082 26.141   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -11.063 4.056 -2.727 0.00868 \*\*   
## Hours 59.977 2.484 24.144 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.26 on 52 degrees of freedom  
## Multiple R-squared: 0.9181, Adjusted R-squared: 0.9165   
## F-statistic: 582.9 on 1 and 52 DF, p-value: < 2.2e-16

# Slope = 59; for every 1 hour people ar eaway from home, then they are about 60 miles more away from home   
# We want to think about; How close is this to the population value?   
# What claims can I make about the population?  
  
plot(moddist, c(1, 2, 5))



# THis whole chunk of code creates a linear model, plots it against the data, and checks the conditions.   
# Pretty linear relaitonship overall: hours away from home, distance increase   
# Residuals have a curve that s alittle ; overall it doesnt seem very bad   
# normal QQ plot, ther eis a problem where one side has a skew   
  
#COok's ditance   
# High leverage, far right   
# High influence, up/down

# Display the Student's t distributions with various  
# degrees of freedom and compare to the normal distribution  
  
x <- seq(-4, 4, length=100)  
hx <- dnorm(x)  
  
degf <- c(1, 5, 15, 30)  
colors <- c("red", "blue", "darkgreen", "gold", "black")  
labels <- c("df=1", "df=5", "df=15", "df=30", "normal")  
  
plot(x, hx, type="l", lty=2, xlab="x value",  
 ylab="Density", main="Comparison of t Distributions")  
  
for (i in 1:4){  
 lines(x, dt(x,degf[i]), lwd=2, col=colors[i])  
}  
  
legend("topright", inset=.05, title="Distributions",  
 labels, lwd=2, lty=c(1, 1, 1, 1, 2), col=colors)



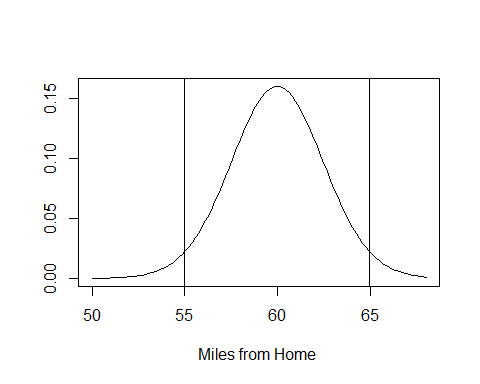
# What the tdist looks like   
#df comes down to sample size and other factors (predicotrs and modeltype)   
# Bigger sample, more degree of freedom   
# For 30 df, a sample of 32, this is pretty much a normal curve   
# As it gests smaller, we are pressing down and pressing out the sides of the graph   
#We are triyng to make this based on what the standard error of our sample is

**CI for Slope or Intercept** - HOw R is doing the math - BUild a distribution where the cente ris where the sample value is adnd we are making a curve on top of that and how far in each direction do we need to go were the middle 95% of the area under the curve is met? Bihar +/- tstar*StandardError or the Bihat Y = Bo + B1X+E - t* comes from a t-distribution with n-2 d.f and depends on the level of confidence - For 1−𝛼 level confidence, use qt(1-α/2,df)in R - **e.g. for 95% confidence and 52 df, qt(0.975,52)** - tstar = tells us how many STDERRORS in each direction we need to go

qt(0.975, 52)

## [1] 2.006647

# T = t distribution   
# area under curve to teh left of the point we want = first arguement; 0.975 = 95% confi int   
# df = 2nd argument; sampel size minus 2   
# Result = 2.00664 ish   
#If i want to see the confidence interval, you have to start at your sample of slope  
#qt gives you tstar  
  
curve(  
 dt.scaled(  
 x,   
 52,  
 mean = summary(moddist)$coef[2,1],  
 sd = summary(moddist)$coef[2,2]  
 ),   
 from = 50, to = 68,  
 xlab = "Miles from Home ",  
 ylab = " "  
 )  
  
  
abline(  
 v=c(  
 qt.scaled(  
 0.025,   
 52,   
 mean = summary(moddist)$coef[2,1],   
 sd = summary(moddist)$coef[2,2]  
 ),  
 qt.scaled(  
 0.975,   
 52,   
 mean = summary(moddist)$coef[2,1],   
 sd = summary(moddist)$coef[2,2]  
 )  
 )  
 )



# IF you want to see the confidence interval   
summary(moddist)$coef[2,1]-qt(0.975, 52)\*summary(moddist)$coef[2,2] #LOwer bound for confidence interval

## [1] 54.99264

summary(moddist)$coef[2,1]+qt(0.975, 52)\*summary(moddist)$coef[2,2] #Upper bound for confidence interval

## [1] 64.96233

#We are predicting that with 95% confidence the solution is between the 54.99-64.96

If we think tha the population might not be normal, then we probably want to do a bootstrap method

**How to find a confidence interval?** -confint(mymodel,level =0.XX) and adjust for the confidence level.

# HOW TO FIND CONFIDENCE INTERVAL; default 95% confidence   
# generally the intercept is not very useful to think about   
# We are predicting taht it could be close to zero   
# WE are mostly looking at the coeffs   
# The hours are about teh same as above   
confint(moddist, level=0.95)

## 2.5 % 97.5 %  
## (Intercept) -19.20166 -2.92358  
## Hours 54.99264 64.96233

**Accuracy of Predictions** Example: It takes a student 2.25 hours to drive from home. How many miles do we predict that thy are away from home? How accurate is that prediction? - Want to make a prediction for a specific case - Wnted regalr prediction, just plug the 2.25 into the regression line - It matterse what you are rtrying to predict - all people or the specific person’s distance from home? - There is a difference; one person has ore variability (Say they’re biking) - If we have the ditribution, ontop of that is some normal curve; most of the people are close to thtat, but they trail off a bit

**Two Forms of Intervals for Regression** 1. Confidence Interval for μY (mean Y) Where is the “true” line for that x? or Where is the average Y for all with that x? 2. Prediction Interval for Individual Y Where are most Y’s for that x?

\_\_CI for μY when X=x\*\_\_ - Predicting in general SSX = ∑▒〖(𝑥\_𝑖 − 𝑥 ̅)〗^2 yhat +/- tstar*standerror*sqrt((1/n)+((xstar-xbar)^2)/SSX)

\_\_Prediction Interval for Individual Y’s when X=x\*\_\_ - predicting for one person yhat +/- tstar*standerror*sqrt(1+ (1/n)+((xstar-xbar)^2)/SSX) Just add 1 in the sqrt

\_\_CI and PI via R when X=x\*\_\_

newx=data.frame(Hours=2.25) # Creat e a new person   
head(newx)

## Hours  
## 1 2.25

predict.lm(moddist, newx, interval="confidence") # Predict mean for all people who are 2.25 away from hoe

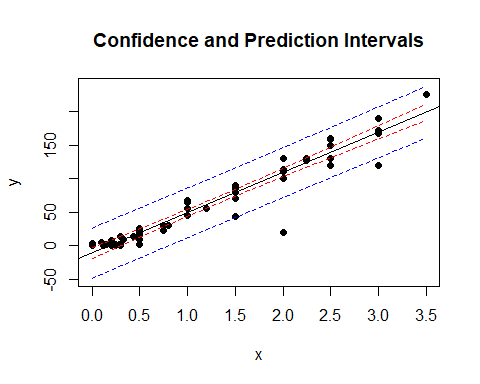
## fit lwr upr  
## 1 123.8867 116.9764 130.797

predict.lm(moddist, newx, interval="prediction") # Distance home for one specific person

## fit lwr upr  
## 1 123.8867 86.59458 161.1789

# Both gives us a fitted value, its a point in the regression lie   
# THis si what we would get if we plugged and chugged   
# One person is going to be a wider range because we want to make sure we get teh one person

CIPIPlot(Domestic$Hours, Domestic$Distance) # Visualize different between confidence and prediction



# calculates   
# For every possible point in teh data, or for the  
# What would be the confidence interveral for that value and what would be the prediction interval

* The red lines are the confidence interval
* if we are trying to predict the mean vlaue for people’s distance away from home based on tehse hours, we will predict the mean is somewhere between the red lines and its tight by the regression line with 95% confidence Th eblue line = much wider
* there’s a lot more variability there
* much wider