STOR 455 Class 5 Transformations

# message=FALSE, warning=FALSE suppress warnings and messages from appearing in knitted files  
  
library(readr)  
library(Stat2Data)

**What to do when regression assumptions are violated?** *Examples of violations:* 1. Nonlinear patterns in residuals 2. Heteroscedasticity (nonconstant variance) 3. Lack of normality in residuals 4. Outliers: influential points, large residuals

**Data Transfomrations** Can be used to: 1. Address non-linear patterns 2. Stabilize variance 3. Remove skewness from residuals 4. Minimize effects of outliers

**Common Transformations** - Log - Square root - Exponentiation - Power function - Reciprocal

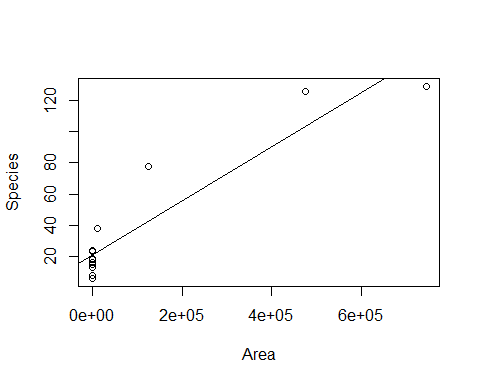
*Example: Mammal Species* Y = Number of mammal species on an island X = Area of the island

Data on fourteen islands in Southeast Asia are stored in SpeciesArea (in Stats2Data)

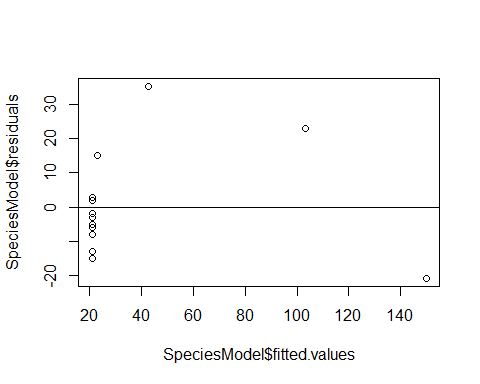
*Notes* - Make sure you check your working environment

**Why is log of something useful?** - Log goes up and then trails off forever - If have a big right skew, or extreme values or outliers in the right side, it helps squeeze the outliers back in - If we have constant variance issues, with fanning patterns, use log - DOesn’t effect low values very much, but helps smoosh bigger data

# Pull in the data   
data("SpeciesArea")  
  
# The log of the species are going to be the most useful   
# Plot the data, make a linear model   
# Want to predict the number of species on each island, based on teh island area  
plot(Species~Area, data=SpeciesArea) # Just the scatterplot, see that is appears to follow a log, but we'll keep make the other models to jsut see how bad it violates the other conditions of linear  
  
SpeciesModel=lm(Species~Area, data=SpeciesArea)  
abline(SpeciesModel) # See that the line deosn't work for us



# Doing Residual analysis   
plot(SpeciesModel$residuals~SpeciesModel$fitted.values)  
abline(0,0)



# When looking at plots, you can see that it's pretty bad   
# Residual analysis can also be done with plot(mod1, 1:2)

# Tells you which point is biggests value of residual   
max(SpeciesModel$residuals)

## [1] 35.24152

# Tells you where the value is in the table   
which.max(SpeciesModel$residuals)

## 3   
## 3

# Gives you the row of the max, this is the number that you got from which.max above  
SpeciesArea[3,]

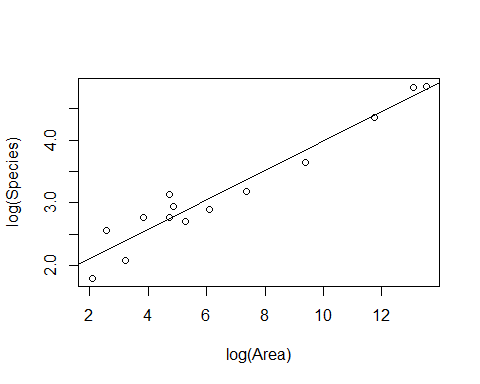
## Name Area Species logArea logSpecies  
## 3 Java 125628 78 11.7411 4.35671

# Just another way to call SpeciesArea[3,]  
SpeciesArea[SpeciesArea$Name=="Java",]

## Name Area Species logArea logSpecies  
## 3 Java 125628 78 11.7411 4.35671

**New Transformation Model**

# New transformation model   
plot(log(Species)~log(Area), data=SpeciesArea)  
SpeciesModel2=lm(log(Species)~log(Area), data=SpeciesArea)  
abline(SpeciesModel2) # Plotting the linear regression line on the scatterplot of the data

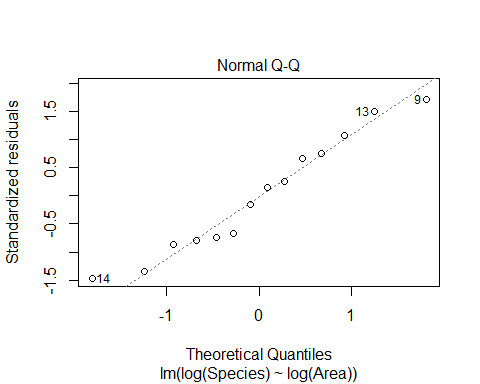
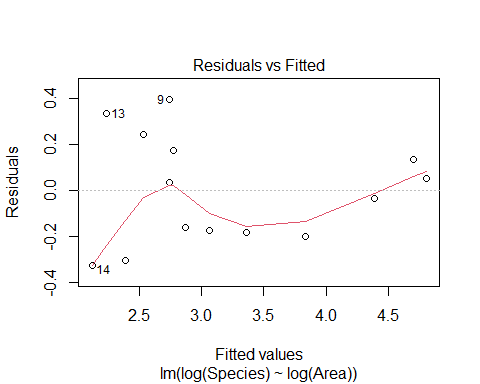


**HOw to interpret the summary table of a transformed linear model** - *Interpret:* For every 1 unit increase in the log(area), there is a 0.2355 increase in log(species)

summary(SpeciesModel2) # Gives you the output of the linear model

##   
## Call:  
## lm(formula = log(Species) ~ log(Area), data = SpeciesArea)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.32280 -0.18071 0.00079 0.16356 0.39534   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.6249 0.1326 12.26 3.81e-08 \*\*\*  
## log(Area) 0.2355 0.0175 13.46 1.34e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2427 on 12 degrees of freedom  
## Multiple R-squared: 0.9379, Adjusted R-squared: 0.9327   
## F-statistic: 181.1 on 1 and 12 DF, p-value: 1.335e-08

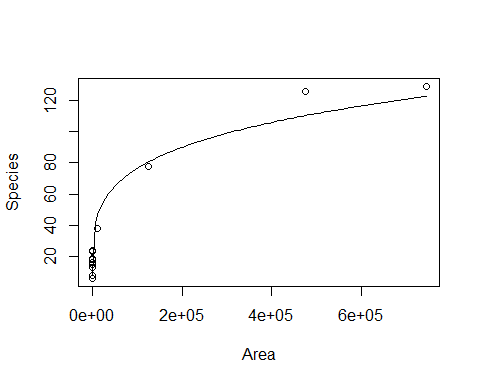
# Look at the coefficients table: Intercept and slope for the Log(area)   
# Intercept (B0) for Log(area) = 1.6249, intercept of the reg line   
# Slope (B1): Log(Area): 0.2335  
# For every 1 unit increase in the log(area), there is a 0.2355 increase in log(species)  
  
# Checking the conditions of the transformed linear model   
plot(SpeciesModel2, 1:2)



**Pulling out the coeffecients of the linear model** - Also shows how to plot the linear model - You need to solve for the same variables before you plot the linear model curve on the base plot because otherwise, it’s in different variables

*BELOW: HOW TO SOLVE OUT FOR LOG OF BOTH SIDES* **IMPORTARNT**

B0 = summary(SpeciesModel2)$coefficients[1,1] # Intercept  
B1 = summary(SpeciesModel2)$coefficients[2,1] # Slope  
  
plot(Species~Area, data=SpeciesArea)  
curve(exp(B0)\*x^B1, add=TRUE) # This is the linear model curve, on the normal data, but solved so that they are in the same units.



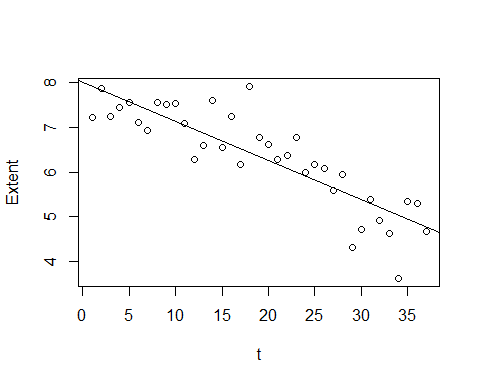
**Artic Sea Ice** The SeaIce data gives information about the amount of sea in the arctic region as measured in September (the time when the amount of ice is at its least) since 1979. The basic research question is to see if we can use time to model the amount of sea ice.

In fact, there are two ways to measure the amount of sea ice: Area and Extent. Area measures the actual amount of space taken up by ice. Extent measures the area inside the outer boundaries created by the ice. If there are areas inside the outer boundaries that are not ice (think about a slice of swiss cheese), then the Extent will be a larger number than the Area. In fact, this is almost always true.

data("SeaIce")  
head(SeaIce)

## Year Extent Area t  
## 1 1979 7.22 4.54 1  
## 2 1980 7.86 4.83 2  
## 3 1981 7.25 4.38 3  
## 4 1982 7.45 4.38 4  
## 5 1983 7.54 4.64 5  
## 6 1984 7.11 4.04 6

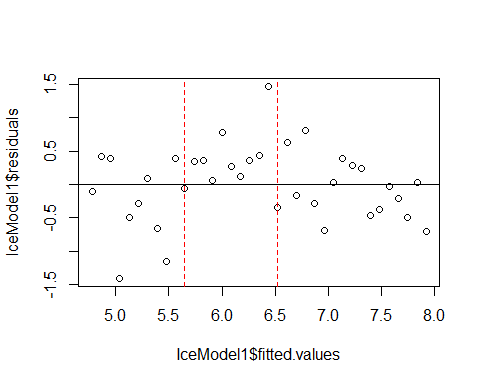
plot(Extent~t, data = SeaIce) # Basic plot of the oringial data   
IceModel1=lm(Extent~t, data = SeaIce) # Linear model of extent by time   
abline(IceModel1) # Plotting the line on the plot



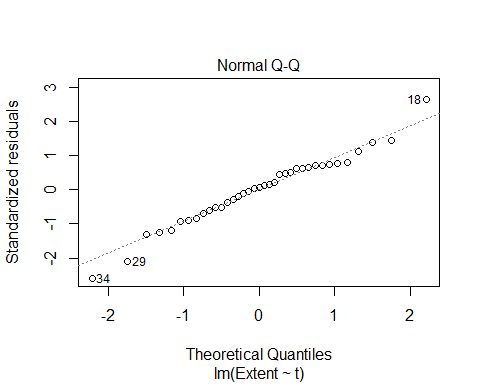
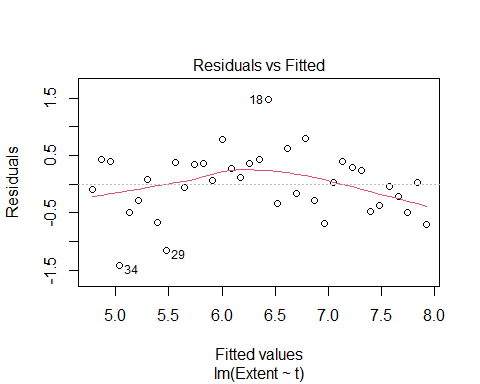
# We see that th eline look spretty good on the data

**Looking at Residuals** - The residual plot has a slight curve - The red line = the benefit of using the plotted of the model itself, rather than the residuals by fitted separate; shows that there is some pattern and curve there - We can see this really well at the middle region - There is some region bt 5 and 6.5 where all the residuals are above the line - So your prediction in that range will always be below what it should be

plot(IceModel1$residuals~IceModel1$fitted.values)  
abline(0,0) #Could also write plot(mod1, 1:2)  
  
# Below used the line to draw two vertical lines where the residual plot was looking weird  
# Will draw 2 vertical lines, one at X of 5.65 and the other 6.52  
abline(v=c(5.65,6.52),   
 col=c("red", "red"),   
 lty=c(2,2), # Look like a dash line   
 lwd=c(1, 1)) #Draws red dashed vertical lines; width

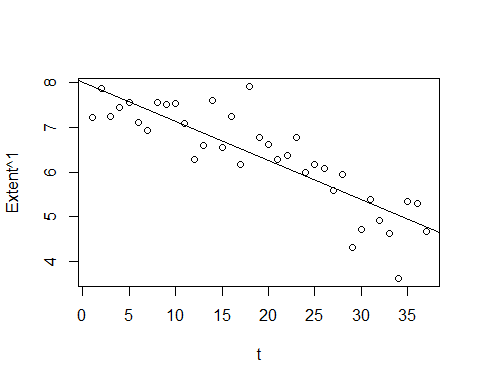


# The abline above shows you where the plot is under predicting   
  
plot(IceModel1, 1:2)

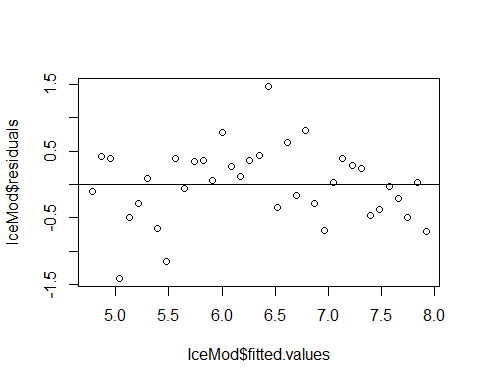


**How does doing transformations change the output?** - Trying exponential to 1

plot(Extent^1~t, data = SeaIce) # Basic plot of the oringial data   
IceMod=lm(Extent^1~t, data = SeaIce) # Linear model of extent by time   
abline(IceMod)



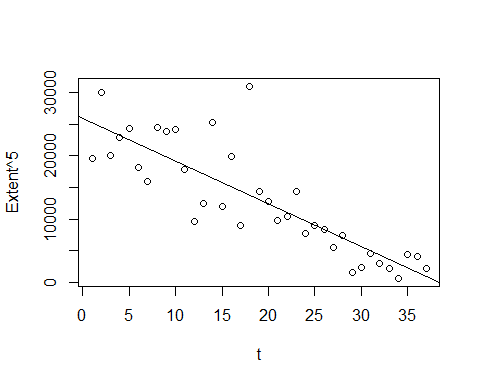
plot(IceMod$residuals~IceMod$fitted.values)  
abline(0,0) #Could also write plot(mod1, 1:2)



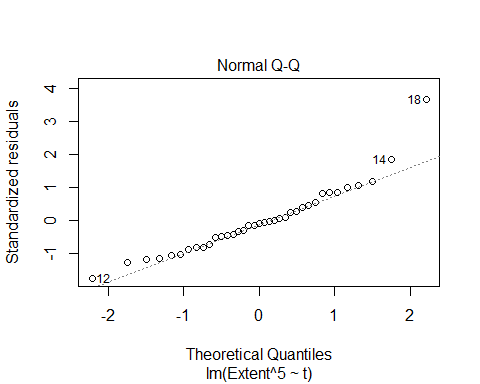
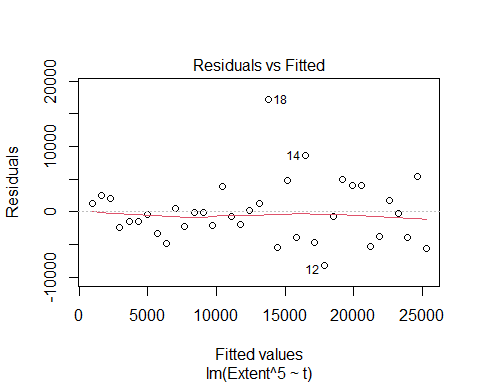
**Trying to raise to the 5th power** - Notice: This looks a bit better on the residual plot - Subjective - No very defined curvature - Look at teh oringal data, we see that there are not some region anymore where all the dots are above or below the line - This might have helped

* We may have also made other problems. See that one middle point that is well above the line? The transformation might have made things worse in different ways

plot(Extent^5~t, data = SeaIce) # Basic plot of the oringial data   
IceMod=lm(Extent^5~t, data = SeaIce) # Linear model of extent by time   
abline(IceMod)



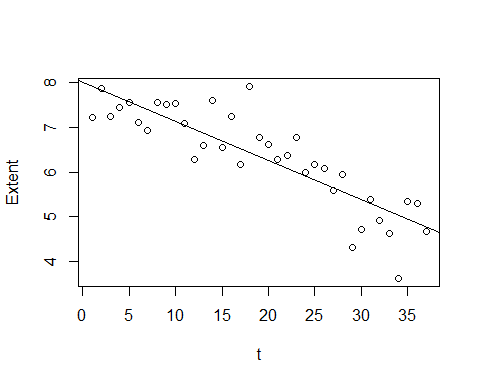
plot(IceMod, 1:2)



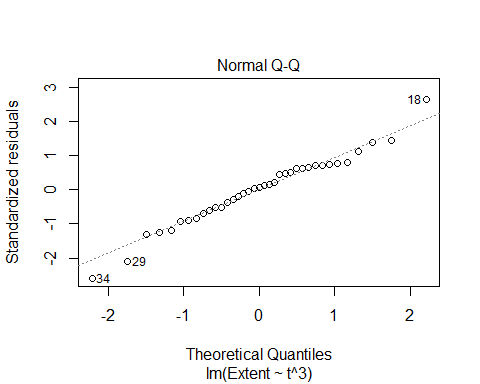
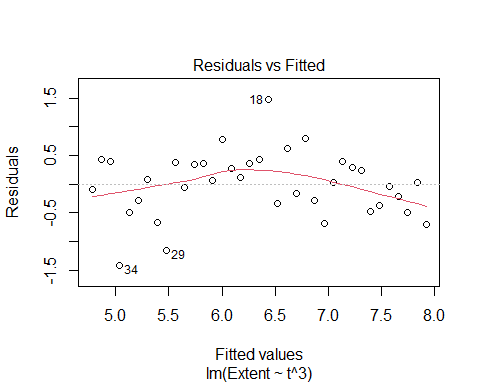
# The reisudal by fitted in teh orinigal plot had a red curve;but in this one the line has appeared to be subdued, but it may be decieving ebcause it plots the range of the data   
# it might just be the extreme case fo 18 appear to stretch it out and tone down what it actualyl looks like when I show it   
# Not as defined as it was before   
  
# The normal QQplot looks pretty good, there the 2 on teh tail that are flying out, but for the small dataset, things appear to fit well   
  
# No constant variance issues here   
  
# Might not be a good model, but it's a better model

**What if we take off the 5th power? and raise time tot he power instead?** - THis gets the curve back in the residual plot, and we don’t want that. so this makes it worse/same - R is ignoring you when you trying to raise the predictor to a power. It thinks that you’re not trying to do that

# THIS IS THE WRONG WAY  
plot(Extent~t^3, data = SeaIce) # Basic plot of the oringial data   
IceMod=lm(Extent~t^3, data = SeaIce) # Linear model of extent by time   
abline(IceMod)



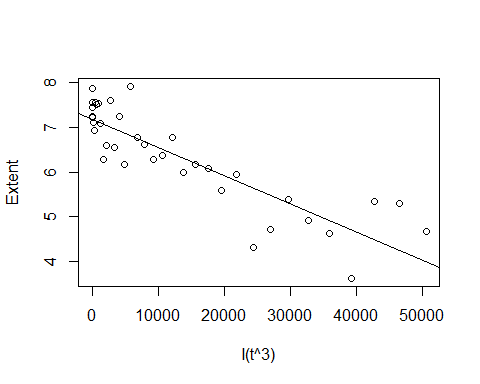
plot(IceMod, 1:2)



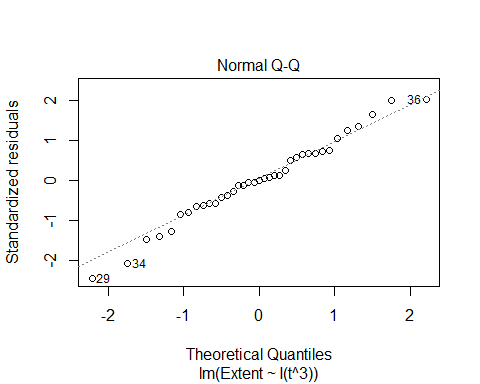
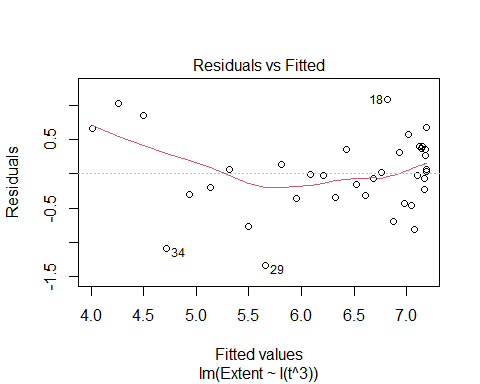
**How to power the predictor**  
-Somethings that work outside of functions may work differently inside of functions. - The carrot that should be raising it, worked on the response, but it won’t work on the predictor, inside of lm - It thinks you’re trying to do an interaction between variables, and you’re not trying to do that.

**Below: HOW TO Power PREDICTOR PROPERLY** - Still doens’t make a good model in this instance, but it could help in the future

# THIS IS THE RIGHT WAY  
plot(Extent~I(t^3), data = SeaIce) # Basic plot of the oringial data   
IceMod=lm(Extent~I(t^3), data = SeaIce) # Linear model of extent by time   
abline(IceMod)

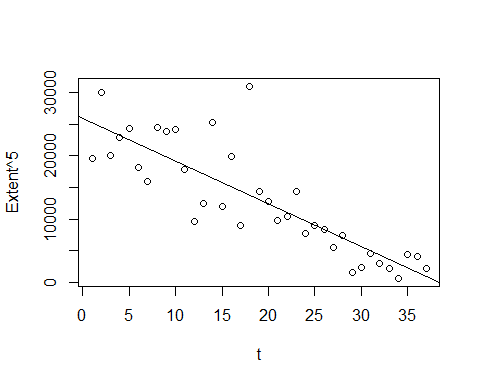


plot(IceMod, 1:2)

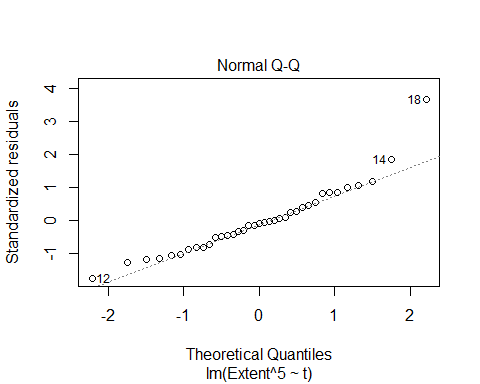
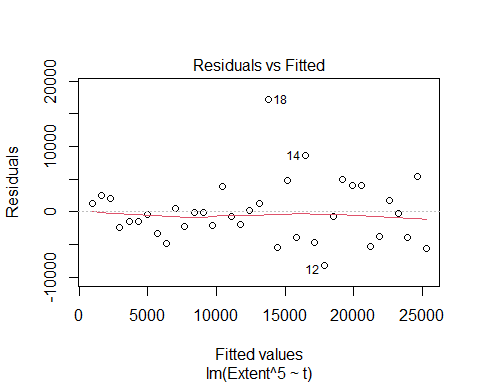


**USING THE BEST MODEL** - WE are using the 5th power model - What if we want to plot this on the orignial raw data?

plot(Extent^5~t, data = SeaIce) # Basic plot of the oringial data   
IceMod=lm(Extent^5~t, data = SeaIce) # Linear model of extent by time   
abline(IceMod)



plot(IceMod, 1:2)



**PLOTTING TRANSFORMATION ON RAW DATA**

plot(Extent~t, data = SeaIce)  
  
summary(IceMod)

##   
## Call:  
## lm(formula = Extent^5 ~ t, data = SeaIce)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8157.2 -3296.7 -438.1 2102.9 17179.4   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 25969.69 1597.27 16.259 < 2e-16 \*\*\*  
## t -676.85 73.29 -9.235 6.51e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4760 on 35 degrees of freedom  
## Multiple R-squared: 0.709, Adjusted R-squared: 0.7007   
## F-statistic: 85.29 on 1 and 35 DF, p-value: 6.512e-11

# Pull out coeff of the transform model   
#INtercept  
B0\_Ice = summary(IceMod)$coefficients[1,1] # Intercept  
   
#Slope  
B1\_Ice = summary(IceMod)$coefficients[2,1] # Slope   
  
# Solve for the curve with math   
curve((B0\_Ice+B1\_Ice\*x)^(1/5), add = TRUE) # Tke 5th root of each side, jsut solved

