Contents

[Class 26 R Logistic Regression 1](#_Toc89889961)

[Class 27 R Logistic Regression: Odds Ratio And Inferences 1](#_Toc89889962)

[Class 28 R Logistic Regression Assessing the model 1](#_Toc89889963)

[STOR 455 Class 29 R Multiple Logistic Regression 1](#_Toc89889964)

[STOR 455 Class 30 Multiple Logistic Regression (Again) 1](#_Toc89889965)

[STOR 455 Class 31 R Multiple Logistic Regression Again Agian 1](#_Toc89889966)

[**STOR 455 Homework 6** 1](#_Toc89889967)

[STOR 455 Practice Exam Solutions 7](#_Toc89889968)

[**STOR 455 Exam #2** 19](#_Toc89889969)

[**Should STOR have Undergraduate Learning Assitants? (YES! Starting Fall 2022…)** 19](#_Toc89889970)

## Class 26 R Logistic Regression

library(titanic)  
data("titanic\_train")  
head(titanic\_train)

## PassengerId Survived Pclass  
## 1 1 0 3  
## 2 2 1 1  
## 3 3 1 3  
## 4 4 1 1  
## 5 5 0 3  
## 6 6 0 3  
## Name Sex Age SibSp Parch  
## 1 Braund, Mr. Owen Harris male 22 1 0  
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female 38 1 0  
## 3 Heikkinen, Miss. Laina female 26 0 0  
## 4 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35 1 0  
## 5 Allen, Mr. William Henry male 35 0 0  
## 6 Moran, Mr. James male NA 0 0  
## Ticket Fare Cabin Embarked  
## 1 A/5 21171 7.2500 S  
## 2 PC 17599 71.2833 C85 C  
## 3 STON/O2. 3101282 7.9250 S  
## 4 113803 53.1000 C123 S  
## 5 373450 8.0500 S  
## 6 330877 8.4583 Q

**Logistic Regression** In all of our regression models (so far) the response variable, Y, has been quantitative. What if we want to model a categorical response?

**Categorical Response Variables** - Ways you can think about categorical response variables - WE will only focus on binary responses

* Binary Response: Whether or not a person smokes and Success of a medical treatment, where Y is divided into NOn-smoker vs smoker and X is divided into Durvies vs Dies
* Ordinal Response: Opinion Poll responses: Where Y = Agree, Netural, and Disagree
* Nominal Response: Political preference; where y = Democrat, Republican, independent

**Binary Logistic Regression** - Response variable (Y) is categorical with just two categories (yes/no or success/failure or 0/1 …). - One approach: Code the response Y as a (0,1) dummy (indicator) variable. - Assume we have a single quantitative predictor X.

**Titanic Survival** Y = Survived (0 = no; 1 = yes) X = Fare (ticket cost in dollars) - Want to predict if the people on the titance survived based on how much they paid

# amkes a table that tells you how many people survived out of the titanic overall   
# Just how many people survived total vs died  
table(titanic\_train$Survived)

##   
## 0 1   
## 549 342

# Survival related to ticket   
# SO a table on if survived based on what type of ticket class they bought   
# Shows a rough relatioship between the pclass and the others   
table(titanic\_train$Survived, titanic\_train$Pclass)

##   
## 1 2 3  
## 0 80 97 372  
## 1 136 87 119

# The below caompres the survided to teh class,   
  
# the first calss,   
# third class a lot didn't survived   
# Wouldnt an underlying variable be that there are more people buying lower class tickets than higher class tickets? I think people should look at the proportion   
# WEll, based on teh propprtions, it still looks like the upper class surived more, hmm wonder why

Below: low pvale; we ave strong ev to say that it si nonzero and there is some linear relationship

* if we look at teh model we can plot survied by fare;

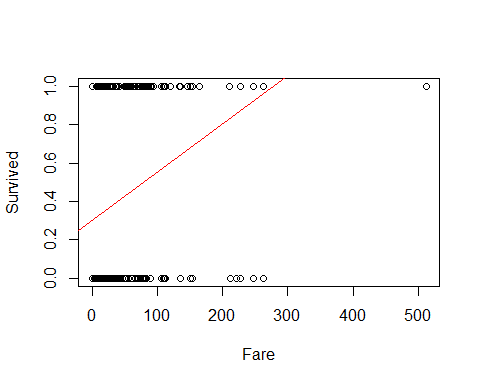
Titanic\_mod=lm(Survived ~ Fare, data=titanic\_train)  
summary(Titanic\_mod)

##   
## Call:  
## lm(formula = Survived ~ Fare, data = titanic\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.9653 -0.3391 -0.3222 0.6044 0.6973   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.3026994 0.0187849 16.114 < 2e-16 \*\*\*  
## Fare 0.0025195 0.0003174 7.939 6.12e-15 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4705 on 889 degrees of freedom  
## Multiple R-squared: 0.06621, Adjusted R-squared: 0.06516   
## F-statistic: 63.03 on 1 and 889 DF, p-value: 6.12e-15

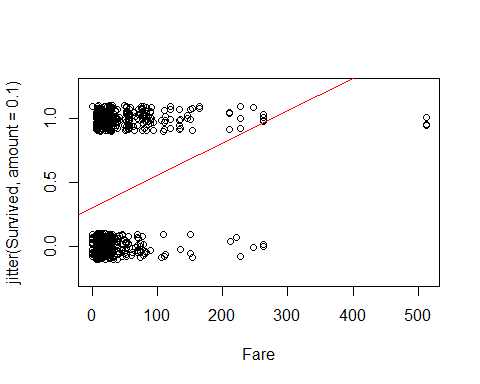
plotting surivied by fare; hard to see how dense it is; there are a lot of calues on top of the m on the bottom;

if we jitter teh data, it moves it up and down a random amount so we can see the visual difference; it doens’t chnage the data, it just changes the visual of it.

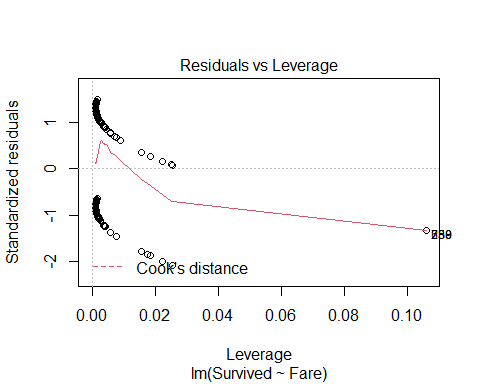
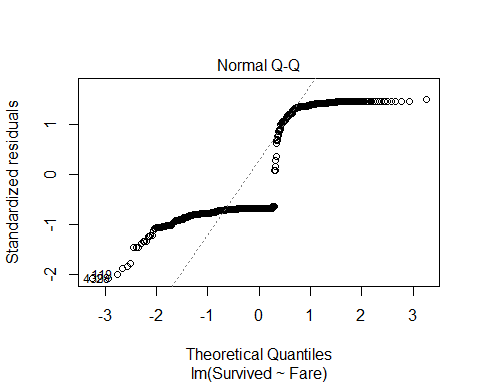
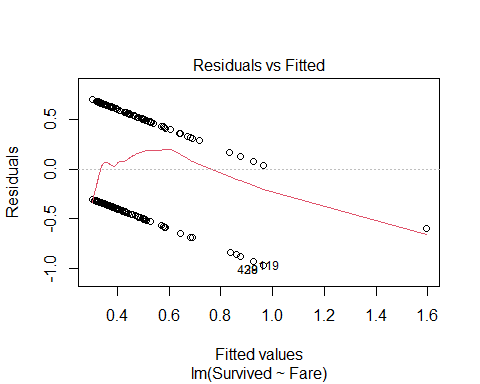
plot(Survived ~ Fare, data=titanic\_train)  
abline(Titanic\_mod, col="red")

 Jitter shows that there is a trend; there appeares to be if you paied more for your ticket, you appear to survied more - residual analysis helps see the difference in plottin gof the things

plot(  
 jitter(Survived, amount=0.1) ~ Fare,   
 ylim = c(-0.25,1.25),   
 data=titanic\_train  
 )  
  
 abline(Titanic\_mod, col="red")



plot(Titanic\_mod, c(1, 2, 5))



The aove shows that teh resultial by fitted has a path residuals are NOT normally distributed and the cook’s distance doesn’t hae any poitns of influence;

but bottom line the model doesn’t work very well

**Binary Logistic Regression Model** Y = Binary response X = Quantitative predictor π = proportion of 1’s (yes, success,…) at any x Probability form 𝜋=𝑒^(𝛽\_𝑜+𝛽\_1 𝑥)/(1+𝑒^(𝛽\_𝑜+𝛽\_1 𝑥) ) - curve(exp(B0+B1*x)/(1+exp(B0+B1*x)),add=TRUE, col=“red”)

below is a model of it by fare ; yo umaek teh family binomial What does it mean when you make the family binomial? Does that make it binary? - No it’s not binary, it means that it is a squared plot

Titanic\_logitmod = glm(Survived ~ Fare, family = binomial, data=titanic\_train)  
# Darws a curve that has a curve in teh middle with a similar likelihood of surviving or not suriving   
# We are looking at the model and predicting teh pi outcome   
# The probabiltiy of that outcome   
# Predict prob of 0 - 1someone who fits this fare would survive based ont eh model we have created   
# WE have to use teh glm function to tdo that   
  
summary(Titanic\_logitmod)

##   
## Call:  
## glm(formula = Survived ~ Fare, family = binomial, data = titanic\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4906 -0.8878 -0.8531 1.3429 1.5942   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.941330 0.095129 -9.895 < 2e-16 \*\*\*  
## Fare 0.015197 0.002232 6.810 9.79e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1186.7 on 890 degrees of freedom  
## Residual deviance: 1117.6 on 889 degrees of freedom  
## AIC: 1121.6  
##   
## Number of Fisher Scoring iterations: 4

# this can be aline if we want it to be   
# But we have to look at it differently

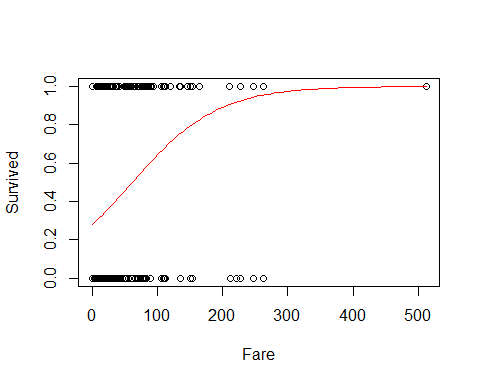
ABove, we are going to claim that the above is a linear model; we;re going to replot the data with teh jitter data; then see if the curve function equation looks nicer

we will learn later where teh curve function is coming from

we want to plot the curve = (exp(B0+B1*x))/(1+exp(B0+B1*x)) (Can see this formula filled in below)

**Predicting Proportion of “Success”** In regression the model predicts the mean Y for any combination of predictors. - What’s the “mean” of a 0/1 indicator variable? 𝑦̄=(∑𝑦\_𝑖 )/𝑛=(#” of ” 1′𝑠)/(#” of trials” )=“Proportion of "success"” - Goal for this model: Predict the “true” proportion of success, π, at any value of the predictor.

plot(Survived ~ Fare, data=titanic\_train)  
  
B0 = summary(Titanic\_logitmod)$coef[1]  
B1 = summary(Titanic\_logitmod)$coef[2]  
  
curve(exp(B0+B1\*x)/(1+exp(B0+B1\*x)),add=TRUE, col="red")



# predicitng the changes of dying based ont eh ticket you bought   
# WE say that there is about a 20% chance of dying if you payed a ceratin amount. THat's what the red line says; at what price of your ticket would oyu have X precentage of curiviing or dying

set.seed(10012020)  
passenger = titanic\_train[sample(nrow(titanic\_train),1),]  
passenger

## PassengerId Survived Pclass Name Sex Age SibSp  
## 622 622 1 1 Kimball, Mr. Edwin Nelson Jr male 42 1  
## Parch Ticket Fare Cabin Embarked  
## 622 0 11753 52.5542 D19 S

# We are randomly selecting one person so that we can check the residuals for a random value

predict(Titanic\_logitmod, passenger, type="response")

## 622   
## 0.4643927

# This is telling us, what do we predict a person who bought a certain amount's chance of surviving?   
# This looks at how much they paid for their ticket and tells us where on teh red curve we would expect this dude to fall   
# So thsi tells us that the dude has a 46% chance of surviving if he paid X amount for his ticket

**Binary Logistic Regression Model** Y = Binary response X = Quantitative predictor π = proportion of 1’s (yes, success,…) at any x Probability form: 𝜋=𝑒^(𝛽\_𝑜+𝛽\_1 𝑥)/(1+𝑒^(𝛽\_𝑜+𝛽\_1 𝑥) ) Logit form: log⁡(𝜋/(1−𝜋))=𝛽\_0+𝛽\_1 𝑥 **NOTE** The logit form can be solved to be in linear form, which is why we can use linear regression rules with it.

**Binary Logistic Regression Model** **Probability Form:** P(X) = ((e(Bo+B1X))/(e(B0+B1X)+1)) **Logit Form:** ln(p/(1-p)) = B0 + B1X

**Example: Golf Putts** Build a model to predict the proportion of putts made (success) based on length (in feet). Data are in Putts1 of Stat2Data.

library(Stat2Data)  
  
data("Putts1")  
head(Putts1)

## Length Made  
## 1 3 1  
## 2 3 1  
## 3 3 1  
## 4 3 1  
## 5 3 1  
## 6 3 1

**Logistic Regression for Putting**

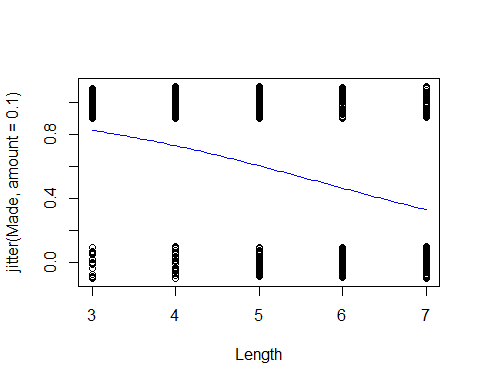
modPutt=glm(Made~Length,family=binomial,data=Putts1)  
summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

# pvalues are small   
# So we like these realtionships, but we should plot it to see what it actually loks like and that is done below

logit = function(B0, B1, x)  
 {  
 exp(B0+B1\*x)/(1+exp(B0+B1\*x))  
}  
# THis function will make the curve that we need on the curve above   
#SO , this is the same as above

B0 = summary(modPutt)$coef[1]  
B1 = summary(modPutt)$coef[2]  
  
plot(jitter(Made,amount=0.1)~Length,data=Putts1)  
# These lines overall plot what the data looks like   
curve(exp(B0+B1\*x)/(1+exp(B0+B1\*x)),add=TRUE, col="red")  
# This line is the line that we would use to predict someone will make a put based on the distance from the hole.   
  
# Can also use the logit function   
curve(logit(B0, B1, x), add = TRUE, col = "blue")



# THis will make the same line as above

logit = function(B0, B1, x)  
 {  
 exp(B0+B1\*x)/(1+exp(B0+B1\*x))  
}  
# THis function will make the curve that we need on the curve above   
#SO , this is the same as above

**Golf Putts Probabilities** 𝜋̂=𝑒(3.257−0.5661𝐿𝑒𝑛𝑔𝑡ℎ)/(1+𝑒(3.257−0.5661𝐿𝑒𝑛𝑔𝑡ℎ) ) Where phat = 𝑝̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑡𝑟𝑖𝑎𝑙𝑠)

**Golf Putts Probabilities** Length: 3,4,5,6,7 phat: 0.835, 0.739, 0.565, 0.488, 0.328 pihat: 0.826, 0.730, 0.605, 0.465, 0.330

Making a table of those that are made vs failed at different lengths

Putts.table = table(Putts1$Made, Putts1$Length)  
Putts.table

##   
## 3 4 5 6 7  
## 0 17 31 47 64 90  
## 1 84 88 61 61 44

p.hat = as.vector(Putts.table[2,]/colSums(Putts.table))  
# Make it a vector because we want to be able to use it with dataframes   
p.hat

## [1] 0.8316832 0.7394958 0.5648148 0.4880000 0.3283582

pi.hat=0  
  
# Compare the predictions, so 3 - 7 feet; so from 3 ft to 4 ft to 5 ft to 6 ft etc.   
# Will make the pihat for each of these   
# Pihat = the probability of sucess at a certain feet distance   
# Pi = success/trials   
for(i in 3:7)  
 {  
 pi.hat[i-2] = logit(B0, B1, i)  
 }  
  
pi.hat

## [1] 0.8261256 0.7295364 0.6049492 0.4650541 0.3304493

# Makea a dataframe that tells you the pihat values and the p hat values   
# We dont know the difference btween pi and p hat   
Putts = data.frame(  
 "Length" = c(3:7),   
 "p.hat" = p.hat,   
 "pi.hat" = pi.hat)  
  
head(Putts)

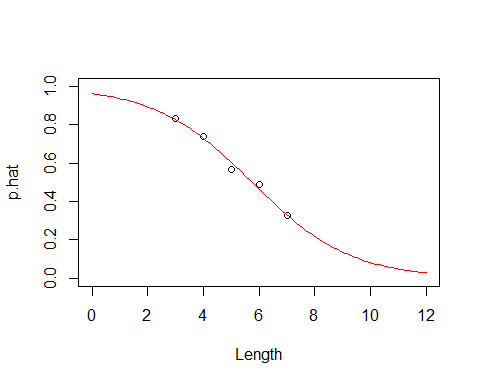
## Length p.hat pi.hat  
## 1 3 0.8316832 0.8261256  
## 2 4 0.7394958 0.7295364  
## 3 5 0.5648148 0.6049492  
## 4 6 0.4880000 0.4650541  
## 5 7 0.3283582 0.3304493

* Probability form of puttin gmodel\*
* etended from 0 - 12; the points on teh graph are the actual proprtions that were made; the p hat values;

the line shows the pi hat values; teh line on the 4 = what we predict; teh dot = the actual value - these are close adnthis is how we test teh linear model - we are going to put it back to the logit form so that we cna put this on a line and we want to see if teh data fits teh lie - if it doens’t then we will have to do transformations - it gets a lot math-y-er we agoig to spend next class talking more about the math how to lok at hypothesis testing and anova doesn’t make sence here anymore because teh residual doesn’t work each point represntes differen combination of data points.

**Probability Form of Putting Model**

plot(p.hat~Length,ylim=c(0,1), xlim=c(0,12), data=Putts)  
curve(logit(B0, B1, x),add=TRUE, col="red")



## Class 27 R Logistic Regression: Odds Ratio And Inferences

library(Stat2Data)  
  
data("Putts1")  
head(Putts1)

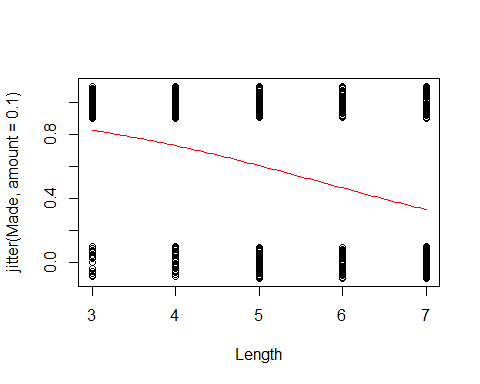
## Length Made  
## 1 3 1  
## 2 3 1  
## 3 3 1  
## 4 3 1  
## 5 3 1  
## 6 3 1

**Logistic Regression for Putting**

# Use glm for different types of graphs  
modPutt=glm(Made~Length,family=binomial,data=Putts1)  
summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

B0 = summary(modPutt)$coef[1]  
B1 = summary(modPutt)$coef[2]  
  
plot(jitter(Made,amount=0.1)~Length,data=Putts1)  
curve(exp(B0+B1\*x)/(1+exp(B0+B1\*x)),add=TRUE, col="red")



**Golf Putts Probabilities** 𝜋̂=𝑒(3.257−0.5661𝐿𝑒𝑛𝑔𝑡ℎ)/(1+𝑒(3.257−0.5661𝐿𝑒𝑛𝑔𝑡ℎ) ) 𝑝̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑡𝑟𝑖𝑎𝑙𝑠) - THis is also a part of Class 26

# This makes a table so we can then make the proportion of success for the golf putts probabilities   
  
Putts.table = table(Putts1$Made, Putts1$Length)  
Putts.table

##   
## 3 4 5 6 7  
## 0 17 31 47 64 90  
## 1 84 88 61 61 44

# Proportion made for each distance   
# takes the probabilities from the putts table and see the proportion for each distance so it's the total for distance 3  
# P(success)/Ntrials  
p.hat = as.vector(Putts.table[2,]/colSums(Putts.table))  
p.hat

## [1] 0.8316832 0.7394958 0.5648148 0.4880000 0.3283582

logit = function(B0, B1, x)  
 {  
 exp(B0+B1\*x)/(1+exp(B0+B1\*x))  
 }

pi.hat=0  
  
for(i in 3:7)  
 {  
 pi.hat[i-2] = logit(B0, B1, i)  
 }  
  
pi.hat

## [1] 0.8261256 0.7295364 0.6049492 0.4650541 0.3304493

Putts = data.frame(  
 "Length" = c(3:7),   
 "p.hat" = p.hat,   
 "pi.hat" = pi.hat)  
  
head(Putts)

## Length p.hat pi.hat  
## 1 3 0.8316832 0.8261256  
## 2 4 0.7394958 0.7295364  
## 3 5 0.5648148 0.6049492  
## 4 6 0.4880000 0.4650541  
## 5 7 0.3283582 0.3304493

he above is all review from last class that we didn’t get to p-hat = the 3:7, length is 3:7; this sis from teh putts data,

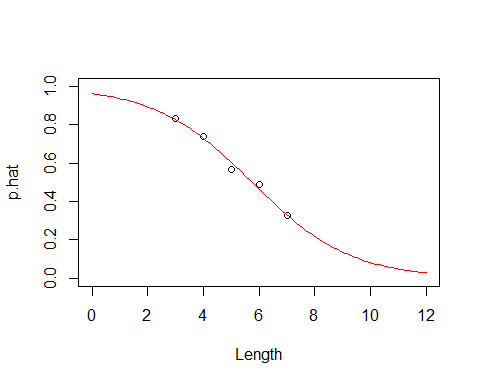
the below plots the curve ontop of it

logit, logs the thing, i think; see th formula above code

if we change x limits, it shows the smaller vs bigger graph

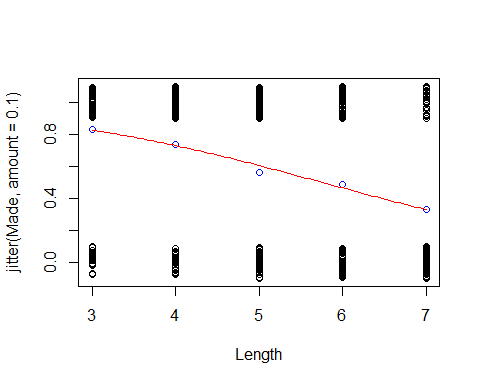
**Probability Form of Putting Model**

plot(p.hat~Length,ylim=c(0,1), xlim=c(0,12), data=Putts)  
# Shows the proportion we are predicting to the prediction plot that we have   
curve(logit(B0, B1, x),add=TRUE, col="red")

 Putts1 = a different dataset; we want to putthe same points on this differently

the blue points are the mean values for each value of x

plot(jitter(Made,amount=0.1)~Length,data=Putts1)  
  
points(p.hat~Length, data=Putts, col='blue')  
  
curve(logit(B0, B1, x),add=TRUE, col="red")

 THink about the Odds of something happening; - the odds vs probability - the odds against are 4:1; they expect those one horse to lose 4/5 vs 1/5 of the time

if pi = proportion of yes (success 1, etc)

the odds of yes = P(pi)/(P(1-pi)) odds of yes = P(yes)/P(no)

logit form = log(odds of yes) = B0 + B1X Below adds 2 new columns to teh dataframe, this messes with teh data

one is probability of it happened over teh probaility that it doesnt haveppen

pi = predicted

**Odds** The odds against a certain horse winning a race are 4 to 1.  
- What does that mean? – 4 losses for every 1 win – P(Win) = 1/5 – P(Loss) = 4/5

𝑂𝑑𝑑𝑠= (𝑃(𝑊𝑖𝑛))/(𝑃(𝐿𝑜𝑠𝑠))=(1/5)/(4/5)=1/4

**Odds** If pi = proportion of “yes” (success, 1, ….) the odds of yes are(is)

(𝑃(𝑦𝑒𝑠))/(𝑃(𝑛𝑜))=𝜋/(1−𝜋)

With a little bit of algebra… 𝑜𝑑𝑑𝑠=𝜋/(1−𝜋)⇔𝜋=𝑜𝑑𝑑𝑠/(1+𝑜𝑑𝑑𝑠)

**Odds and Logistic Regression** Logit form: log⁡(𝜋/(1−𝜋))=𝛽\_𝑜+𝛽\_1 𝑋 -The logistic model assumes a linear relationship between the predictor and log(odds). - log⁡( 𝑜𝑑𝑑𝑠)=𝛽\_𝑜+𝛽\_1 𝑋

**Logit Form of Putting Model**

**Back to Putting Data** Since we have lots of putts, we can estimate 𝑝̂ (proportion of putts made) at each length 𝑝̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑡𝑟𝑖𝑎𝑙𝑠) and the odds (𝑜𝑑𝑑𝑠)̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑚𝑖𝑠𝑠𝑒𝑑)=𝑝̂/(1−𝑝̂ ) and find log⁡((𝑜𝑑𝑑𝑠̂) at each length.

**Golf Putts Odds** (𝑜𝑑𝑑𝑠)̂=(# 𝑚𝑎𝑑𝑒)/(# 𝑚𝑖𝑠𝑠𝑒𝑑)=𝑝̂/(1−𝑝̂ ) (from sample) (𝑜𝑑𝑑𝑠)̂=𝜋̂/(1−𝜋̂ ) (from logistic regression) Length: 3,4,5,6,7 oddshat(from sample): 4.94,2.84,1.30,0.95,0.49 oddshat(from regression): 4.75,2.70,1.53,0.87,0.49

Putts$p.Odds = Putts$p.hat/(1-Putts$p.hat)  
Putts$pi.Odds = Putts$pi.hat/(1-Putts$pi.hat)  
  
head(Putts)

## Length p.hat pi.hat p.Odds pi.Odds  
## 1 3 0.8316832 0.8261256 4.9411765 4.751277  
## 2 4 0.7394958 0.7295364 2.8387097 2.697355  
## 3 5 0.5648148 0.6049492 1.2978723 1.531320  
## 4 6 0.4880000 0.4650541 0.9531250 0.869348  
## 5 7 0.3283582 0.3304493 0.4888889 0.493539

**Plot for Putts Data** Plot log⁡((𝑜𝑑𝑑𝑠̂) versus Length (3, 4, 5, 6, 7) Add a line with intercept and slope from the logistic model.

The below code does something

the line tells you how the probaility chanes as teh rate of other things change.

so we need to think fo teh odds ratio, a common way to compare two groiups is to look at a ratio of their odds

odds ratio (OR) = Odd.R = Odd1/Odd2

odds using data from 4 ft = 2.84 odds using data from 3 feet = 4.94

odds ratio ( 4 to 3) = 2.84/4.94 = 0.57

the odds of making a putt from 4 feet are 57% of the odds of making from 3 feet

*Interpreting Slope Using Odds Ratio*

log(Odds) = B0+B1X -> odds = e^(B0+B1\*X)

*CI for Slope and ODds Ratio* - Using teh SE for the slope, find a CI for B1 with:

B-hat1 +/- z-star \* SE

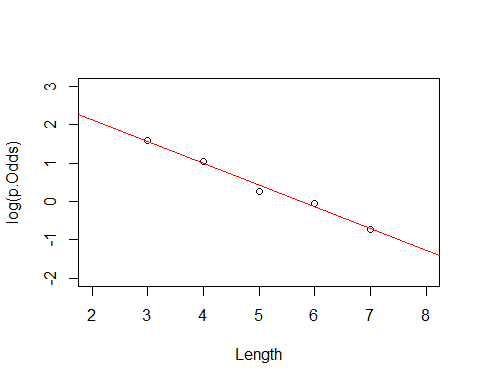
the above will get you theCI for teh odds ratio (E^B1) exponentiate the CI for B1

ex:

CI for slope = (0.5, 0.6) CI for OR = e^0.5, e^0.6 = (0.497, 0.648)

**Logit Form of Putting Model**

plot(log(p.Odds)~Length, data=Putts, xlim=c(2,8), ylim=c(-2,3))  
abline(B0, B1, col="red")



**Odds Ratio** A common way to compare two groups is to look at the ratio of their odds “Odds Ratio”=” OR “=(”Odd” “s” \_1)/(“Odd” “s” \_2 )

**Putting Data** Odds using data from 4 feet = 2.84 Odds using data from 3 feet = 4.94 Odds ratio (4 ft to 3 ft) = 2.84/4.94=0.57 The odds of making a putt from 4 feet are 57% of the odds of making from 3 feet.

**Odds Ratios for Putts** From fitted logistic: Length: 3,4,5,6,7, pihat: 0.826,0.730,0.605,0.465,0.331 odds hat: 4.75,2.70,1.53,0.84,0.49

To Odds Ratio Length: 4-3 ft, 5-4 ft, 6-5 ft, 7-6 ft Odds Ratio: 0.57, 0.57, 0.57, 0.57

In a logistic model, the odds ratio when changing the predictor by one is constant.

**Odds Ratios for Putts** From samples at each distance: Length: 3,4,5,6,7 phat: 0.832, 0.739, 0.565, 0.488, 0.328 oddshat: 4.94, 2.84, 1.30, 0.95, 0.49

To Odds Ratio: Length: 4-3ft, 5-4 ft, 6-5ft, 7-6ft Odds Ratio: 0.57, 0.46, 0.73, 0.51

**Interpreting “Slope” using Odds Ratio** log⁡(𝑜𝑑𝑑𝑠)=𝛽\_0+𝛽\_1 𝑥goes to 𝑜𝑑𝑑𝑠=𝑒^(𝛽\_0+𝛽\_1 𝑥) What happens when we increase x by one? 𝑒^(𝛽\_0+𝛽\_1 (𝑥+1))=𝑒^(𝛽\_0+𝛽\_1 𝑥)∙𝑒^(𝛽\_1 ) When we increase x by one, the odds increase/decrease by a factor of 𝑒^(𝛽\_1 ) (odds ratio). For putts: The odds of making a putt decrease by a factor of 0.57 (𝑒^(−0.566)) for every extra foot of length.

**CI for Slope and Odds Ratio** Using the SE for the slope, find a CI for β1 with 𝛽̂\_1±𝑧^∗∙𝑆𝐸 <- this is just the formula for confidence intervals To get CI for the odds ratio (𝑒^(𝛽\_1 )) exponentiate the CI for β1

CI for slope: −0.566±1.96(0.06747) =(−0.698,−0.434)

CI for OR: 〖(𝑒〗(−0.698),𝑒(−0.438)) =(0.497, 0.648)

SE\_B1 = summary(modPutt)$coef[2,2]  
exp(B1 - SE\_B1\*qnorm(0.975))

## [1] 0.4973894

exp(B1 + SE\_B1\*qnorm(0.975))

## [1] 0.6479761

in practice we are not going to use teh confint.default function; because the default forces the thing to use z scores; and teh not by default uses some lo glikelihoods to get this thing

**Similar tests/measures for logistic regression?** Recall: “Ordinary” Regression lm(formula = Active ~ Rest)

Coefficients: Estimate Std. Error t value Pr(>|t|)  
(Intercept) 8.75340 5.60773 1.561 0.12  
Rest 1.18387 0.08214 14.413 <2e-16 \*\*Rest, above, tests for individual coefficients\*

*For the first three italics, there are for comparing the models* Residual standard error: *14.39* on 310 degrees of freedom Multiple R-squared: *0.4012*, Adjusted R-squared: *0.3993* F-statistic: *207.7* on 1 and 310 DF, p-value: *< 2.2e-16* <- test overall fit

**Test for Individual Coefficients** Ho: Bi = 0 Ha: Bi != 0

t.s. = Bhat/SEofBhat (R will give you all of these variables) Interpret as with individual t-tests in ordinary regression P-value = 2P( Z > |t.s.| )

Estimate Std. Error z value Pr(>|z|)

(Intercept) 3.25684 0.36893 8.828 <2e-16  ***Length -0.56614 0.06747 -8.391 <2e-16***

**Estimating Parameters in Ordinary Regression** Coefficients are chosen to minimize the sum of the squared errors in the observed sample. (Least Squares Estimation) 𝑆𝑆𝐸=〖Σ(𝑦−𝑦̂)〗^2 **WE WANT A SMALL SSE**

**Test for Overall Fit** Ho: B1 = 0 -> log(odds) = Bo Ha: B1 != 0 -> log(Odds) = Bo + B1X

How much “better” does the linear model do than one with a constant? Is it “significantly” better?

**Maximizing the Likelihood of the Sample** - Suppose that there are three decks of cards: 1. Standard 52 card deck 2. Euchre deck (9, 10, J, Q, K, A) 3. Deck with all red cards If two cards were drawn from a deck (without replacement), a Jack of Hearts, then a King of Hearts, from which deck do you think that there were chosen?

* Suppose that there are three decks of cards:

1. Standard 52 card deck; (1/52)(1/51)≈“0.000377”
2. Euchre deck (9, 10, J, Q, K, A); (1/24)(1/23)≈“0.001812”
3. Deck with all red cards; (1/26)(1/25)≈“0.001538”

**Estimating Parameters in Logistic Regression** Parameters are chosen to maximize the likelihood of the observed sample. (Maximum Likelihood Estimation) If the ith data point is YES (yi=1), calculate 𝜋̂\_𝑖 If the ith data point is NO (yi=0), calculate 1−𝜋̂\_𝑖

Likelihood:𝐿=∏〖𝜋̂\_𝑖〗^(𝑦\_𝑖 ) (1−𝜋̂\_𝑖 )^(1−𝑦\_𝑖 ) **WE WANT A HIGH LIKELIHOOD**

**Test for Overall Fit** Length: 3,4,5,6,7, Made: 84,88,61,61,44 Missed: 17,31,47,64,90 Ratio: 0.826, 0.730, 0.605, 0.465, 0.330

𝐿=∏〖𝜋̂\_𝑖〗^(𝑦\_𝑖 ) (1−𝜋̂\_𝑖 )^(1−𝑦\_𝑖 ) Ho: B1 = 0 -> log(odds) = Bo L = .576^338\*(1-.576)^249

Ha: B1 != 0 -> log(Odds) = Bo + B1X L = (0.826^84 \* 0.174^17) \* (0.730^88 \* 0.270^31) \* (0.605^61 \* 0.395^47) \* (0.465^61 \* 0.535^64) \* (0.330^44 \* 0.670^90)

exp(confint.default(modPutt))

## 2.5 % 97.5 %  
## (Intercept) 12.6006177 53.5133410  
## Length 0.4973894 0.6479761

exp(confint(modPutt))

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 12.7974573 54.4505172  
## Length 0.4960611 0.6464444

just keep in mind the units for the CI

Similar tests/measures for logistic regression

recall: “Ordinary” regression

*Test for idividual coeff*

Ho: Bi = 0  
Ha: Bi -/= 0

t.s = B-hati/SE(B-hati)

R WIll give you all of these numbers

interpret as with individual t tests in ordinary regression

p-value = 2P(Z>abs(t.s))

*Estimating Parameters in ORd Regression*

Coeff are chosen to min the sum of the squared errors in teh observed sample (LEast Squares Estimation

SEE = sum(y-y-hat)^2

We want a small SSE

*Test for Overall Fit* H0: B1 = 0 Ha: B1 =/= 0 log(odds) = B0 log(odds) = B0 + B1X; these are competing models

how much better does the lienar mdoel do than one with a constatst? IS sit sig better?

*Estimating Parameters in Logistic Regression* Parameters are chosen to max the likelihood of the observed sample (MAx likelihood estimation)

If teh ith data poin is YES (yi = 1), calc pi-hati

If teh ith data point is No (yi = 0), calc 1-pi-hati

We want L to be big

THis is where the table(Putts1$MAde) starts

summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

**−2 ln⁡(𝐿) for Constant (H0) Model** For a constant model:  
𝐿\_0=𝜋̂^(#𝑦𝑒𝑠) 〖(1−𝜋̂)〗^(𝑛−#𝑦𝑒𝑠) log⁡(𝐿\_0 )=#𝑦𝑒𝑠∙log⁡𝜋̂ )+#𝑛𝑜∙log-pihat

Combining all putts: 338 made out of 587 𝜋hat =338/587=0.5758 𝐿\_0=〖0.5758〗^338 〖0.4242〗^249

log⁡(𝐿\_0 )=338 log⁡(0.576)+249 log⁡(0.424)=−400.1 〖−2log〗⁡(𝐿\_0 )=800.2

**Putts1: Made~Length** lmodPutt=glm(Made~Length,family=binomial,data=Putts1) summary(lmodPutt)

**Example: Golf Putts** 𝐿=∏〖𝜋̂\_𝑖〗^(𝑦\_𝑖 ) (1−𝜋̂\_𝑖 )^(1−𝑦\_𝑖 ) 𝐿=〖0.826〗^84 〖0.174〗^17 〖0.730〗^88 〖0.270〗31⋯〖0.330〗44 〖0.670〗^90 log⁡(𝐿)=84 log⁡(0.826)+17 log⁡(0.174)+⋯ +44 log⁡(0.330)+90 log⁡(0.670)=−359.9 Coefficients are chosen to get 𝑙𝑜𝑔(𝐿) as big as possible 〖−2log〗⁡(𝐿)=718. 8 <- Minimize residual deviance

* How much “improvement” with the predictor?
* Compare the null deviance with the residual deviance; subtract the two to get your Gstatistic

lmodPutt=glm(Made~Length,data=Putts1,family=binomial,data=Putts1) summary(lmodPutt)

Coefficients: Estimate Std. Error z value Pr(>|z|) (Intercept) 3.25684 0.36893 8.828 <2e-16 Length -0.56614 0.06747 -8.391 <2e-16

Null deviance: 800.21 on 586 degrees of freedom

Residual deviance: 719.89 on 585 degrees of freedom

−2 l𝑜𝑔⁡(𝐿\_0 )−(−2 log⁡(𝐿) )=800.2−719.99=80.3 This difference is called the G statistic.

**Evaluating Overall Fit** Test for overall fit (Similar to regression ANOVA) t.s. = G = improvement in –2log(L) over a model with just a constant term Compare to y2 with k d.f. (chi-square) - k = number of predictiors

The null sys tat it doens’t matter how far we are from teh hole, while teh laternative says that it does matter

Bo = 0 Bo =/= 0

table(Putts1$Made)

##   
## 0 1   
## 249 338

338/(338+249)

## [1] 0.5758092

L.null = (.576)^338\*(1-.576)^249  
L.null

## [1] 1.725431e-174

-2\*log(L.null)

## [1] 800.2087

if the distance matters, then teh difference lengts =will ahev different values

we first calc how you got the sample from 3 ft putts

based on data, we made 0.73 putts at 3ft, then the probabiltiy of making it

the log(L) below is a little bigger than thte above L.null, which means taht we like the second L better

L = 0.826^84\*0.174^17\*0.730^88\*0.270^31\*.605^61\*.395^47\*.465^61\*.535^64\*0.330^44\*0.670^90  
L

## [1] 4.765502e-157

-2\*log(L)

## [1] 719.8889

do things with chi-squared; it lieks chi squared, so we like log?

we cna look at this like a chi squared

how likeily is it to get this on a chi squared distribution

summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

1-pchisq(80.3,1)

## [1] 0

summary(modPutt)

##   
## Call:  
## glm(formula = Made ~ Length, family = binomial, data = Putts1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8705 -1.1186 0.6181 1.0026 1.4882   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.25684 0.36893 8.828 <2e-16 \*\*\*  
## Length -0.56614 0.06747 -8.391 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 800.21 on 586 degrees of freedom  
## Residual deviance: 719.89 on 585 degrees of freedom  
## AIC: 723.89  
##   
## Number of Fisher Scoring iterations: 4

G = summary(modPutt)$null.deviance - summary(modPutt)$deviance  
  
1 - pchisq(G,1)

## [1] 0

the below gives you how likly we would see this by chacne; we if small p value; then we can reject Ho

**Evaluating Overall Fit** Ho: Bi = 0 Ha: Bi != 0

log⁡(𝜋/(1−𝜋))=𝛽\_𝑜+𝛽\_1 𝑋

anova(modPutt, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: Made  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 586 800.21   
## Length 1 80.317 585 719.89 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Class 28 R Logistic Regression Assessing the model

library(Stat2Data)  
library(readr)  
  
GoldenBalls <- read\_csv("https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/GoldenBalls.csv")  
  
logit = function(B0, B1, x)  
 {  
 exp(B0+B1\*x)/(1+exp(B0+B1\*x))  
 }

**Checking Linearity** Three methods depending on the type of dataset: - Datasets with a binary predictor – nothing to check! - Datasets with a quantitative predictor with many response values for each predictor - Datasets with a quantitative predictor with many values for the predictor but few response values for each predictor value.

*Example Golf Putts* We looked at the log odds formula there wre no big deviations

what about the Datsets with binary predicotrs? We are going to look at golden balls; split or steal the money how likeyly are they to split or steal based on teh age of teh contestant?

head(GoldenBalls)

## # A tibble: 6 x 2  
## Over40 Split  
## <dbl> <dbl>  
## 1 1 1  
## 2 1 1  
## 3 1 1  
## 4 1 1  
## 5 1 1  
## 6 1 1

table(GoldenBalls$Split, GoldenBalls$Over40)

##   
## 0 1  
## 0 195 76  
## 1 187 116

Claim: There is a difference in the proportion of people who would split or steal based on their age.

He table above shows the table for if the preson is over40 and has split or not; so row, column; where row = if they split and coloumn = age over or below 40. over forty = 1; under = 0

we see that there is a low p value below that says that we have evidence to say there is a realtionship between age and their saying yes or no to steal

then we pull out teh coeffs. we plot the raw data with some jitter, but we dont think tis super needed right now.

if we want to make a logit plot,then we want to find out where the table is, why do we table 2 and coloumn; p-hat will give you the samplel porp[tion, he is just pulling from the table

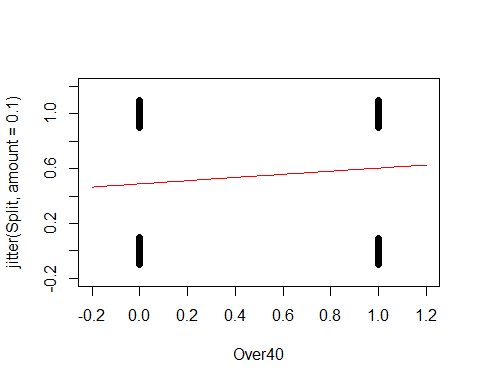
then goldenball odds is just loged

**Golden Balls: Logistic Regression**

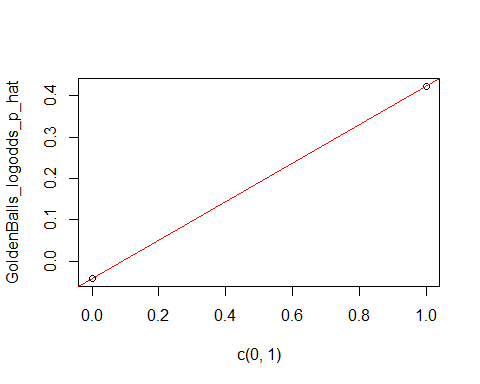
GBmod = glm(Split~Over40, data=GoldenBalls, family=binomial)  
# Make logistic model  
summary(GBmod)

##   
## Call:  
## glm(formula = Split ~ Over40, family = binomial, data = GoldenBalls)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.361 -1.160 1.004 1.195 1.195   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.04189 0.10235 -0.409 0.68233   
## Over40 0.46475 0.17960 2.588 0.00966 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 793.95 on 573 degrees of freedom  
## Residual deviance: 787.17 on 572 degrees of freedom  
## AIC: 791.17  
##   
## Number of Fisher Scoring iterations: 4

# See summary of logistic mdoel   
  
B0 = summary(GBmod)$coef[1] # Pull out intercept  
B1 = summary(GBmod)$coef[2] # Pull out slope  
  
#Plot the GBMod data  
plot(jitter(Split,amount=0.1)~Over40,data=GoldenBalls, xlim=c(-.2, 1.2), ylim=c(-.2, 1.2) )  
# Plot the GBMod  
curve(logit(B0, B1, x),add=TRUE, col="red")



GoldenBalls\_table=table(GoldenBalls$Split, GoldenBalls$Over40)  
GoldenBalls\_p\_hat=as.vector(GoldenBalls\_table[2,]/colSums(GoldenBalls\_table))  
GoldenBalls\_logodds\_p\_hat = log(GoldenBalls\_p\_hat/(1-GoldenBalls\_p\_hat))  
  
plot(GoldenBalls\_logodds\_p\_hat~c(0,1))  
abline(B0, B1, col="red")



**Quantitative predictor: Few response values for each predictor** - This process breaks down if there are not many values of the response, but the previous process can be mimicked by manipulating the data first.

* We can manipulate the data by:

1. Slicing the x-axis into intervals.
2. Compute the average x-value and empirical logit foreach slice
3. Plot the values as before

Medical School Acceptance Dataset: Is GPA a useful predictor for acceptance to medical school?

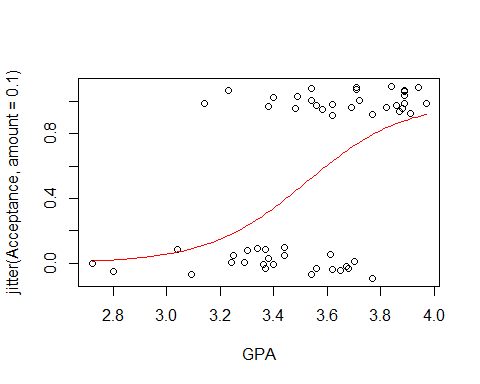
data("MedGPA")  
head(MedGPA)

## Accept Acceptance Sex BCPM GPA VR PS WS BS MCAT Apps  
## 1 D 0 F 3.59 3.62 11 9 9 9 38 5  
## 2 A 1 M 3.75 3.84 12 13 8 12 45 3  
## 3 A 1 F 3.24 3.23 9 10 5 9 33 19  
## 4 A 1 F 3.74 3.69 12 11 7 10 40 5  
## 5 A 1 F 3.53 3.38 9 11 4 11 35 11  
## 6 A 1 M 3.59 3.72 10 9 7 10 36 5

WE’re hoping that therea re different cases for the low, medium and high; if they didn’t follow the logit curve, ten they would be weird and you cant use logit

below we are predicting the mdoel with acceptabnce and gpa

MedGPA.glm = glm(Acceptance~GPA, data=MedGPA, family = binomial)  
  
B0 = summary(MedGPA.glm)$coef[1]  
B1 = summary(MedGPA.glm)$coef[2]  
  
plot(jitter(Acceptance,amount=0.1)~GPA,data=MedGPA)  
curve(logit(B0, B1, x),add=TRUE, col="red")

 It makes sense that teh logit model fits here, the ost people who get in have high gpa it looks like its a jittered acceptabnce rate we dont know other things about these people, there are probablyother reasons they arent getting or they gota ccepted into med school

below we are shorting based on how good their gpa is, this is just ordering it by gpa; so its going to look at the levels **Create a new dataframe with the predictor sorted smallest to largest**

sorted.MedGPA = MedGPA[order(MedGPA$GPA),]  
GPA = sorted.MedGPA$GPA  
Acceptance = sorted.MedGPA$Acceptance  
  
#we want to pull out GPA so we can just work with that; so we do gpa = sorted.medgpa$gpa   
#then we also want to know if thery got acepted, so see teh Acceptance object in R   
  
# Select a number of “slices” or groups for the data and find the mean value of the predictor for each slice  
# WE slect slices so we can look at the mean of the groups   
# We want to see if our model follows the means of the groups well   
  
groups = 5  
group.size = 11  
  
GPA.means = 0  
Acceptance.sums = 0  
  
for(i in 1:groups){  
 GPA.means[i] = mean(  
 GPA[((group.size\*i)-(group.size-1)):(group.size\*i)])  
 }  
  
GPA.means

## [1] 3.130909 3.410000 3.585455 3.734545 3.905455

above does a loop where it takes teh mean of teh first 11 elements, tehn it’s teh second 11 elements; so its not easy to read this, there is a nicer way to do this:

library(TTR)  
runMean(GPA, 11)

## [1] NA NA NA NA NA NA NA NA  
## [9] NA NA 3.130909 3.189091 3.240909 3.270909 3.297273 3.319091  
## [17] 3.334545 3.349091 3.366364 3.380000 3.396364 3.410000 3.426364 3.441818  
## [25] 3.457273 3.473636 3.490000 3.506364 3.525455 3.541818 3.558182 3.570909  
## [33] 3.585455 3.597273 3.610000 3.623636 3.636364 3.650000 3.661818 3.671818  
## [41] 3.685455 3.699091 3.717273 3.734545 3.751818 3.769091 3.786364 3.803636  
## [49] 3.820000 3.836364 3.851818 3.864545 3.880000 3.893636 3.905455

the above will do the same thing as teh loop, but it’s from the TTR package this gives us all of the 11, we jsut want the 11th, 22th, 33rd, and 44th values; how do we get that? look below

runMean(GPA, 11)[seq(11,length(GPA),11)]

## [1] 3.130909 3.410000 3.585455 3.734545 3.905455

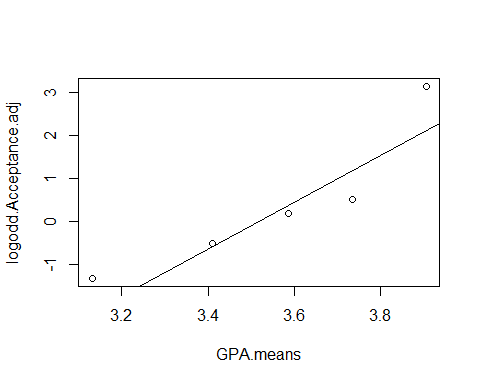
THe above gives you what we want

Accepted sums **Find the number of acceptances for each slice**

for(i in 1:groups){  
 Acceptance.sums[i] =sum(   
 Acceptance[((group.size\*i)-(group.size-1)):(group.size\*i)])  
 }  
  
Acceptance.sums

## [1] 2 4 6 7 11

# “Fudge” the proportions slightly and find the log of the predicted odds  
# Why “fudge”?  
# Proportions of 0 and 1 cause issues.  
  
Acceptance.prop.adj = (Acceptance.sums +0.5)/(group.size+1)  
  
logodd.Acceptance.adj = log(Acceptance.prop.adj/(1-Acceptance.prop.adj))  
  
# Plot the logodds of the adjusted proportions by the means of the predictor variables and a linear model  
# ASk self if the data ppears linear and if the group numbers matter (YES! THEY DO)  
plot(logodd.Acceptance.adj~GPA.means)  
abline(B0,B1)

 above does it in loop format, but you can do it better with teh ttr package

acceptance.sums = runSum(Acceptance, 11)  
acceptance.sums

## [1] NA NA NA NA NA NA NA NA NA NA 2 2 2 2 3 2 2 2 2 2 3 4 5 6 6  
## [26] 6 6 6 6 6 7 7 6 5 4 5 4 5 5 6 7 6 6 7 8 9 9 10 10 10  
## [51] 10 10 11 11 11

THis is saysing tha ttherea re not enought o give me an 11 people sum; the above gives you a running cumumlaitve sum for 11, then 12, then 13 then etc

so to get what we want do below:

acceptance.sums = runSum(Acceptance, 11)[seq(11,length(GPA), 11)]  
acceptance.sums

## [1] 2 4 6 7 11

we need to fudge so we tell R to nto give us exactly 0 and not eactly 1 we get log of 0 andlog of 1 aer issues;

we are just goint o add 0.5 to the sums and divivde by the group size sot aht we never get exactly 1 or a little less than that.

look at teh group size and numbers way above.

WE’re a bit lost because he deviated adn I’m not paying as much attention as I could

acceptance.propo.ad = (Acceptance.sums + 0.5)/(group.size + 1)  
acceptance.propo.ad

## [1] 0.2083333 0.3750000 0.5416667 0.6250000 0.9583333

logodd.accept.ad = log(acceptance.propo.ad/(1-acceptance.propo.ad))  
logodd.accept.ad

## [1] -1.3350011 -0.5108256 0.1670541 0.5108256 3.1354942

below, we want to make sure we load stat2data

we give it teh raw data, so that it does the thing taht we want itt to; we want to group it by 5, but i dont know why

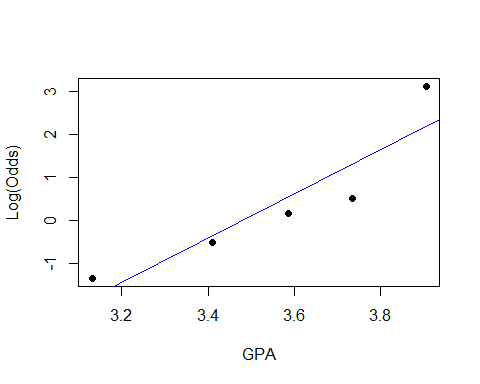
we choose 5 because that’s how we split teh data earlier in the code;

this will give you the same plot, but we the othere xtra work in teh past; its a fsater way to do the thing

there may be issues with this on teh data; if you try and slice the data ti migth overlap and cause errors; certain groupsing might work differently for different data; it defpending on teh datatset

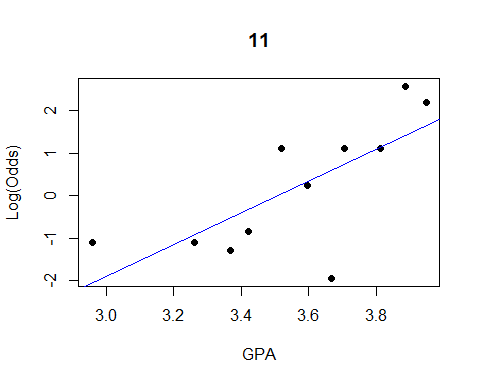
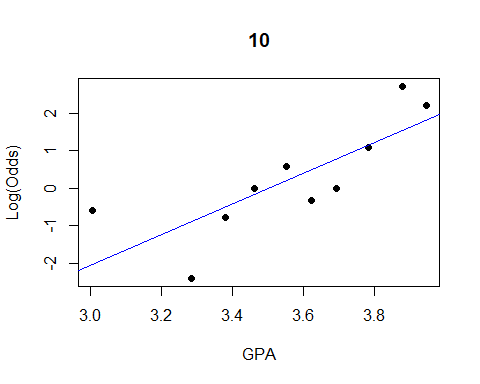
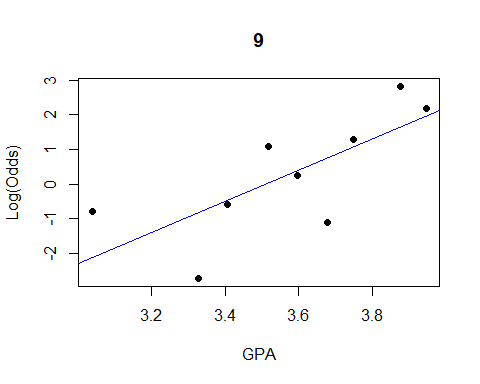
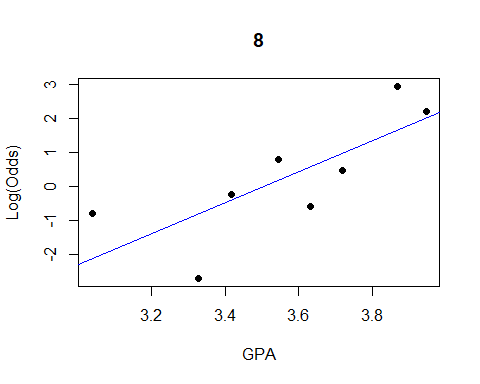
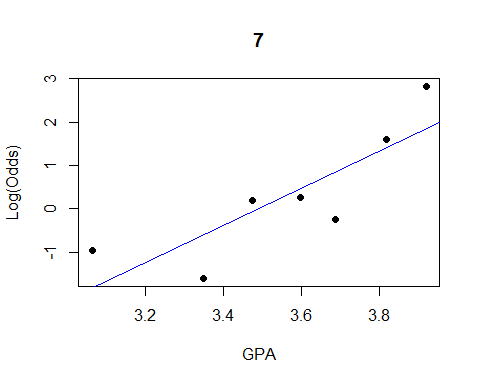
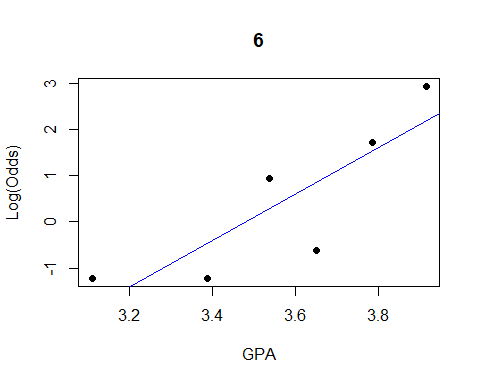
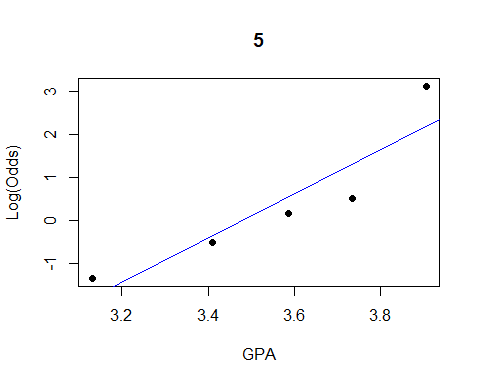
you just haev to ttiral and error it **Check linear conditions**

emplogitplot1(Acceptance~GPA, data=MedGPA, ngroups=5)

 this will check the groups of the other names; this is how you test other nimber grouping, which is useful

these arent residuals, but we think about them that way; if tehre are different patterns, we could try transformations; the logit plot works teh same as teh full things we did in the other classes **This checks different grouping types**

for(j in 5:11){emplogitplot1(Acceptance~GPA, data=MedGPA, ngroups=j, main=j)}



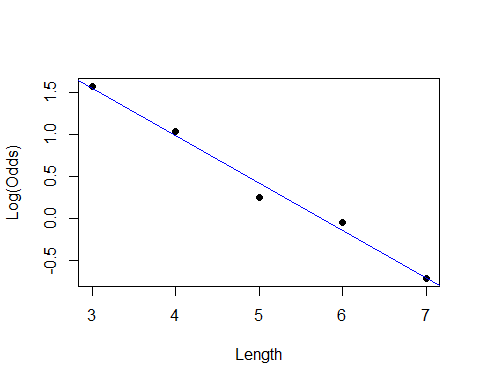
below shows you a shortcut on how to do the long thing int eh short; with teh logitplot function

ngroups = all shows that all possible predictor wvalue and will make teh own group

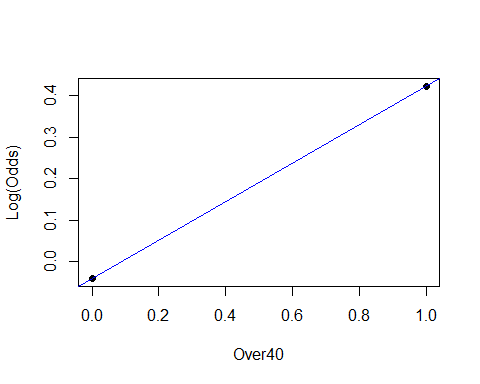
that doesnt work for medgpa data beause the grouping are different; there are vary few outcomes for each logodd for med gpa and it doesnt tell us much aout ht eoutcomes

the goldenballgraph is less exciting because it’s a graph with a thing between two lines

data("Putts1")  
emplogitplot1(Made~Length, data=Putts1, ngroups="all")



emplogitplot1(Split~Over40, data=GoldenBalls, ngroups="all")



## STOR 455 Class 29 R Multiple Logistic Regression

library(Stat2Data)  
library(leaps)  
  
source("https://raw.githubusercontent.com/JA-McLean/STOR455/master/scripts/ShowSubsets.R")  
  
logit = function(B0, B1, x)  
{  
 exp(B0+B1\*x)/(1+exp(B0+B1\*x))  
}

**Categorical Predictors with Multiple Categories in Logistic Regression** Example: Predicting survival in an intensive care unit (ICU) Response: Survive = 0 for dead and 1 for lived Predictor: AgeGroup = 1 for YOung, 2 for middle, 3 for old

data("ICU")  
head(ICU)

## ID Survive Age AgeGroup Sex Infection SysBP Pulse Emergency  
## 1 4 0 87 3 1 1 80 96 1  
## 2 8 1 27 1 1 1 142 88 1  
## 3 12 1 59 2 0 0 112 80 1  
## 4 14 1 77 3 0 0 100 70 0  
## 5 27 0 76 3 1 1 128 90 1  
## 6 28 1 54 2 0 1 142 103 1

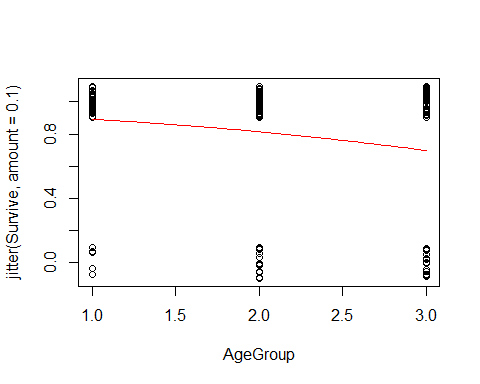
**Categorical Predictors with Multiple Categories in Logistic Regression** - Two approaches: 1. **Method #1:** Logistic regression for Survive with AgeGroup as a quantitative predictor. 2. **Method #2:** Use dummy (indicator) variables for the age categories as predictors in a logistic regression model for Survive.

**Method #1: AgeGroup as Quantitative Pred**

ICUmod = glm(Survive~AgeGroup, data=ICU, family=binomial)  
  
summary(ICUmod)

##   
## Call:  
## glm(formula = Survive ~ AgeGroup, family = binomial, data = ICU)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1120 0.4769 0.6414 0.6414 0.8484   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.7566 0.5732 4.809 1.52e-06 \*\*\*  
## AgeGroup -0.6399 0.2414 -2.651 0.00802 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 200.16 on 199 degrees of freedom  
## Residual deviance: 192.66 on 198 degrees of freedom  
## AIC: 196.66  
##   
## Number of Fisher Scoring iterations: 4

B0 = summary(ICUmod)$coef[1]  
B1 = summary(ICUmod)$coef[2]  
  
plot(jitter(Survive,amount=0.1)~AgeGroup,data=ICU)  
curve(logit(B0, B1, x),add=TRUE, col="red")

 The above is a log mod that predicets survive y age group with fam = bi; if we don’t tell it fam = bi, then it will only give us a line and we wont get teh curve we want

if it’s non zero, tehn there is a change in teh log odds based on surviving based on teh age group when we plot this we can look at it and see the coeffs.

**Method #1: AgeGroup as Quantitative Pred**

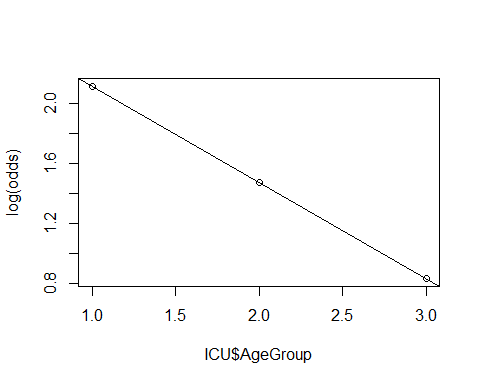
pi = logit(B0, B1, ICU$AgeGroup)  
head(pi)

## [1] 0.6977833 0.8925107 0.8140745 0.6977833 0.6977833 0.8140745

odds = pi/(1-pi)  
head(odds)

## [1] 2.308884 8.303252 4.378498 2.308884 2.308884 4.378498

plot(log(odds)~ICU$AgeGroup)  
abline(B0,B1)

 The above shows what we are predicting ; the odds are teh predicitng/ 1-odds predicted.

ploting the logodds with teh mdel on top of it

plotting teh log odds against teh other things here.

the predicts are right on this line; as we goes from young to middle to old we follow this ration

we miht not have this be true we might be forcing a relationship

its like when we were looking at active vs resting heartrate

its assuming a consistent rate of chaneg between age groups

if we lok at how the data actuallyuly looks with teh table; we can see that the actual counts are

we want to see the proportions are they different from teh predicted values and how much?

so we are going to make a table that are the proportions

so 54/59; etc etc. the prop.table will make this prop table for us

we want to lok at teh column proportion for those who surived adn that’s why we have a 2 in the code below

**Two-way Table: Survive by AgeGroup**

# Two way table of Counts  
ICU.table = table(ICU$Survive, ICU$AgeGroup)  
ICU.table

##   
## 1 2 3  
## 0 5 17 18  
## 1 54 60 46

# Two way table of Column Proportions  
ICU.prop.table = prop.table(ICU.table,2)  
ICU.prop.table

##   
## 1 2 3  
## 0 0.08474576 0.22077922 0.28125000  
## 1 0.91525424 0.77922078 0.71875000

# Two way table of Column logodds  
logodds.ICU.table = log(ICU.prop.table/(1-ICU.prop.table))  
logodds.ICU.table

##   
## 1 2 3  
## 0 -2.3795461 -1.2611312 -0.9382696  
## 1 2.3795461 1.2611312 0.9382696

above we can see in teh actual data ,the ic propo will tell us teh proportions; we have

if we plot all these together then we get a log odds table; that lets us plot it all together logodds proportion/ 1-proprotion

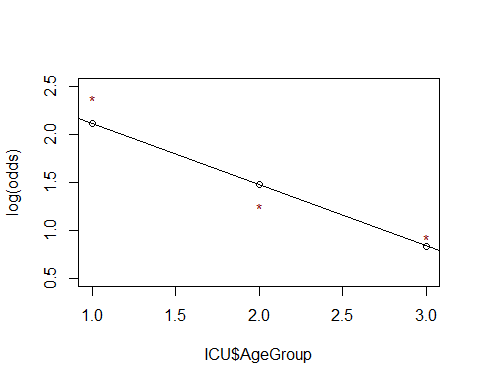
we want to be able to plot this, but it wont work well in a table format, so we need to make thi a dataframe.

we want teh columsn transposed; t = transponse **Two-way Table: Survive by AgeGroup**

logodds.ICU.df = t(as.data.frame.matrix(logodds.ICU.table))  
head(logodds.ICU.df)

## 0 1  
## 1 -2.3795461 2.3795461  
## 2 -1.2611312 1.2611312  
## 3 -0.9382696 0.9382696

plot(log(odds)~ICU$AgeGroup, ylim=c(.5, 2.5))  
abline(B0,B1)  
points(logodds.ICU.df[,2], col="dark red",pch="\*")



the above pulls out all the log odds rows and makes them red so they stand out this is so that you can do somethign else

what if we wanted two age groups; we could make 1 age group for young, and one for old; and if its’ not either then it has to be middle; but we have used this to be 1 = young and the other is middle, then old would be both = 0 **Method #2: Survive ~ Middle + Old**

ICUmod.2 = glm(Survive~factor(AgeGroup), data=ICU, family=binomial)  
summary(ICUmod.2)

##   
## Call:  
## glm(formula = Survive ~ factor(AgeGroup), family = binomial,   
## data = ICU)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2218 0.4208 0.7063 0.7063 0.8127   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.3795 0.4675 5.090 3.57e-07 \*\*\*  
## factor(AgeGroup)2 -1.1184 0.5422 -2.063 0.03915 \*   
## factor(AgeGroup)3 -1.4413 0.5439 -2.650 0.00805 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 200.16 on 199 degrees of freedom  
## Residual deviance: 191.59 on 197 degrees of freedom  
## AIC: 197.59  
##   
## Number of Fisher Scoring iterations: 5

**Dummy Indicators for Multiple Categories** For a categorical predictor with k levels, we should use k − 1 dummy indicators. X = 1 if group 1, 0 if otherwise Xi-1 = 1 if in group k-1, 0 if otherwise

What happens to Group #k? That is teh reference group

Constant term is an estimate for Group #k and other coefficients are the differences from it.

* The coef for age 2 and 3 are the log odds for each in relation to the survive
* we dont want to lok at certain age groups we want ot know if age group as a whole is a good predicotr the ines dont give us that

**Binary Logistic Regression Model** Y = Binary response X1,X2,…,Xk = Multiple predictors π = proportion of 1’s at any x1, x2, …, xk Equivalent forms of the logistic regression model: Logit form: log⁡(𝜋/(1−𝜋))=𝛽\_0+𝛽\_1 𝑥\_1+𝛽\_2 𝑥\_2+⋯+𝛽\_𝑘 𝑥\_𝑘

Probability form: 𝜋=𝑒^(𝛽\_𝑜+𝛽\_1 𝑥\_1+𝛽\_2 𝑥\_2+⋯+𝛽\_𝑘 𝑥\_𝑘 )/(1+𝑒^(𝛽\_𝑜+𝛽\_1 𝑥\_1+𝛽\_2 𝑥\_2+⋯+𝛽\_𝑘 𝑥\_𝑘 ) )

y = binary response; X1, X2, Xk = mult predictor

pi = propotion of 1 at any xi

this is equal to the log reg mod

log form = log(pi/1-pi) = B0\_B1X1+B2X2 +…BkXk

prob form = pi = (e(B0+B1X1+…+BkXk)/1-e(same as num))

we can also use anova below to do the hypothesisi test; there aren’t teh samekind of residuals

the chisq thing will tell it;

recall nested f-test basic idea: Is teh improvement (reduction in SEE) Sig for teh number of extra preditores?

compare full model to reduced model = use t.s. = F - ratio (interpret similar to ANOVA)

**Interpreting Individual Tests** Similar issues to ordinary regression: - Is the predictor helpful, given the other predictors are already in the model? - Beware of problems due to multicollinearity. - Try to keep the model simple.

**G-Test for Overall Fit** H0:β1=β2=…=βk=0 vs. Ha: Some βi ≠ 0 t.s. = G = improvement in –2log(L) over a model with just a constant term Compare to 2 with k d.f.

Null deviance: 200.16 on 199 degrees of freedom

Residual deviance: 191.59 on 197 degrees of freedom 𝐺=200.16−191.59=8.57

1-pchisq(8.57,2) [1] 0.01377362 <- Reject H0

**Method #2: Survive ~ Middle + Old** Coefficients: Estimate Std. Error z value Pr(>|z|)  
(Intercept) **2.3795** 0.4675 5.090 3.57e-07 \*\*\* <- Log(oods) young factor(AgeGroup)2 **-1.1184** 0.5422 -2.063 0.03915 \*  
factor(AgeGroup)3 **-1.4413** 0.5439 -2.650 0.00805 \*\*

The factor age group bolded = the change in log(odds) for middle and old compared to young

anova(ICUmod.2, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: Survive  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 199 200.16   
## factor(AgeGroup) 2 8.5721 197 191.59 0.01376 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Recall: Nested F-test** Purpose: Test a subset of predictors Ex: 𝑌=𝛽1𝑋1+𝛽2𝑋2+𝛽3𝑋3+𝛽4𝑋4+𝛽5𝑋5 + 𝜀  
𝐻0:𝛽3=𝛽4=𝛽5=0 vs. 𝐻𝑎: 𝑆𝑜𝑚𝑒 𝛽𝑖 ≠ 0 for i>2

Basic idea: Is the improvement (reduction in SSE) “significant” for the number of extra predictors? i.e. Compare “full” model to “reduced” model

t.s.= F-ratio (interpret similar to ANOVA)

**Nested LRT for Logistic Regression(Likelihood Ratio Test)** Purpose: Test a subset of predictors Ex: log⁡(𝑜𝑑𝑑𝑠)=𝛽1𝑋1+𝛽2𝑋2+𝛽3𝑋3+𝛽4𝑋4+𝛽5𝑋5  
𝐻0:𝛽3=𝛽4=𝛽5=0 vs. 𝐻𝑎: 𝑆𝑜𝑚𝑒 𝛽𝑖 ≠ 0 for i>2

Basic idea: Is the improvement, change in –2log⁡(𝐿), “significant” for the number of extra predictors? i.e. Compare “reduced” model to “full” model

𝜒^2=–2log⁡(𝐿𝑅𝑒𝑑𝑢𝑐𝑒𝑑) – (–2log⁡(𝐿𝐹𝑢𝑙𝑙))

Chi-square d.f.=#extra predictors tested

**Comparing Full to Reduced Models** ICUMod 3 = full and ICUMod2 = reduced

𝐻0:𝛽3=0 vs. 𝐻𝑎: 𝛽3 ≠ 0

ICUmod.3 = glm(Survive~factor(AgeGroup)+Emergency, data=ICU, family=binomial)  
summary(ICUmod.3)

##   
## Call:  
## glm(formula = Survive ~ factor(AgeGroup) + Emergency, family = binomial,   
## data = ICU)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4388 0.2632 0.4469 0.8536 1.0137   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.7771 0.8801 5.428 5.7e-08 \*\*\*  
## factor(AgeGroup)2 -1.4317 0.5527 -2.590 0.009585 \*\*   
## factor(AgeGroup)3 -1.8557 0.5606 -3.310 0.000931 \*\*\*  
## Emergency -2.5234 0.7538 -3.347 0.000816 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 200.16 on 199 degrees of freedom  
## Residual deviance: 171.16 on 196 degrees of freedom  
## AIC: 179.16  
##   
## Number of Fisher Scoring iterations: 6

use anova for a drop in dev test;

this tells us

ICU mod 2 = reduced and 3 = full with emergency

we are going to see that just the two models you get the two residuals deviationces, it tells you df difference; its teh 1 bc its jstthe emerg var; the

doesnt give a p value ebcause we didnt give it a test

if we tell it the test is chisq, then we will get teh pvaleu

there are small values and they are different; tehre are different assumptions being made; it prob wont change the decision, but ti could be difference value thatn teh summaru **Drop in Deviance Test**

1 - pchisq(summary(ICUmod.2)$deviance - summary(ICUmod.3)$deviance, 1)

## [1] 6.187652e-06

#Reject H0 (p-value= 6.187652e-06). The Emergency term significantly improves the model.  
# This is also often called a “Drop-in-Deviance” test.

anova(ICUmod.2, ICUmod.3, test="Chisq")

## Analysis of Deviance Table  
##   
## Model 1: Survive ~ factor(AgeGroup)  
## Model 2: Survive ~ factor(AgeGroup) + Emergency  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 197 191.59   
## 2 196 171.16 1 20.429 6.188e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Example: Predicting Medical School Acceptance** Data: MedGPA  
Accept Status: A=accepted to medical school or D=denied admission Acceptance Indicator for Accept: 1=accepted or 0=denied Sex F=female or M=male BCPM Bio/Chem/Physics/Math grade point average GPA College grade point average VR Verbal reasoning (subscore) PS Physical sciences (subscore) WS Writing sample (subcore) BS Biological sciences (subscore) MCAT Score on the MCAT exam (sum of CR+PS+WS+BS) Apps Number of medical schools applied to

Goal: Find the “best” model for Acceptance using some or all of these predictors.

NOw, what if instead i did the anova of mod3; with a test = chisq; that is going to give su s a table tha tdeos teh test but compares with teh factor with teh null and tehn comp emergenc withw factor age grouo it everytime i add a thing then it des a nested test

useful only if you want to test things in order

if we want to test different order tehnw e have to do something difference. ’

data(MedGPA)  
head(MedGPA)

## Accept Acceptance Sex BCPM GPA VR PS WS BS MCAT Apps  
## 1 D 0 F 3.59 3.62 11 9 9 9 38 5  
## 2 A 1 M 3.75 3.84 12 13 8 12 45 3  
## 3 A 1 F 3.24 3.23 9 10 5 9 33 19  
## 4 A 1 F 3.74 3.69 12 11 7 10 40 5  
## 5 A 1 F 3.53 3.38 9 11 4 11 35 11  
## 6 A 1 M 3.59 3.72 10 9 7 10 36 5

**Criteria to Compare Models for Ordinary Multiple Regression** - Look for large R2 – But R2 is always best for the model with all predictors - Look for large adjusted R2 – Helps factor in the number of predictors in the model - Look at individual t-tests – Might be susceptible to multicollinearity problems

*-How to Choose Models to Compare for Ordinary Multiple Regression?* Method #1: All Subsets! Consider all possible combinations of predictors. How many are there? Pool of k predictors -> 2𝑘−1 subsets

Advantage: Find the best model for your criteria Disadvantage: LOTS of computation

* Note: requires leaps package

all = regsubsets(Acceptance~., data=MedGPA[,2:11])

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =  
## force.in, : 1 linear dependencies found

## Reordering variables and trying again:

ShowSubsets(all)

## SexM BCPM GPA VR PS WS BS MCAT Apps Rsq adjRsq Cp  
## 1 ( 1 ) \* 30.57 29.24 11.08  
## 2 ( 1 ) \* \* 39.37 36.99 5.34  
## 3 ( 1 ) \* \* \* 43.75 40.37 3.49  
## 4 ( 1 ) \* \* \* \* 46.40 42.02 3.16  
## 5 ( 1 ) \* \* \* \* \* 48.87 43.55 2.98  
## 6 ( 1 ) \* \* \* \* \* \* 49.59 43.16 4.35  
## 7 ( 1 ) \* \* \* \* \* \* \* 49.99 42.38 6.00  
## 8 ( 1 ) \* \* \* \* \* \* \* \* 49.99 41.10 8.00

# This “works” in the sense that it runs, but creates a linear not a logistic model…

Will learn later how to automate the chosing the best model for other types of models

## STOR 455 Class 30 Multiple Logistic Regression (Again)

library(readr)  
library(bestglm)  
library(Stat2Data)  
  
insurance <- read\_csv("https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/insurance.csv")

* Looking at something like mallowCp and see which one is most likely based on teh precitors and a few other things -

1. Only the resposne and possible predictors variables should be withing the datagrame
2. te response variable must be the last column in teh dataframe.

We need to tell R. We need to think what we don’t want in the model ; if we have accept anad acceptance in teh model,then we are going to get some errors abecause they are the same things; there are issues because the logistical model wouldn’t work as well because it would be a straight vertial line.

WE could chosoe teh specific columns we want or choose teh ones we dont’ want with negative index ;

**Example: Predicting Medical School Acceptance** Data: MedGPA  
Accept Status: A=accepted to medical school or D=denied admission Acceptance Indicator for Accept: 1=accepted or 0=denied Sex F=female or M=male BCPM Bio/Chem/Physics/Math grade point average GPA College grade point average VR Verbal reasoning (subscore) PS Physical sciences (subscore) WS Writing sample (subcore) BS Biological sciences (subscore) MCAT Score on the MCAT exam (sum of CR+PS+WS+BS) Apps Number of medical schools applied to

Find the “best” model for Acceptance using some or all of these predictors.

data(MedGPA)  
head(MedGPA)

## Accept Acceptance Sex BCPM GPA VR PS WS BS MCAT Apps  
## 1 D 0 F 3.59 3.62 11 9 9 9 38 5  
## 2 A 1 M 3.75 3.84 12 13 8 12 45 3  
## 3 A 1 F 3.24 3.23 9 10 5 9 33 19  
## 4 A 1 F 3.74 3.69 12 11 7 10 40 5  
## 5 A 1 F 3.53 3.38 9 11 4 11 35 11  
## 6 A 1 M 3.59 3.72 10 9 7 10 36 5

below shows how to set accetance to null, that deltes teh accept vars.

best glm wants the response in the specific part of the dataframe, wants it in teh last section ; if your thing is named soemthign sepciifc, it sometimes acts differently, but mostly this is different.

THe second part of the code below reorders teh columns with teh response value last so that the glm is better. THere are other ways that you can do this, but this is for consistency.

**bestglm for Model Selection** Requirements to use bestglm() 1. Only the response and possible predictor variables should be within the dataframe

MedGPA.1 = within(MedGPA, {Accept = NULL}) #delete Accept variable  
head(MedGPA.1)

## Acceptance Sex BCPM GPA VR PS WS BS MCAT Apps  
## 1 0 F 3.59 3.62 11 9 9 9 38 5  
## 2 1 M 3.75 3.84 12 13 8 12 45 3  
## 3 1 F 3.24 3.23 9 10 5 9 33 19  
## 4 1 F 3.74 3.69 12 11 7 10 40 5  
## 5 1 F 3.53 3.38 9 11 4 11 35 11  
## 6 1 M 3.59 3.72 10 9 7 10 36 5

Above we could have just overwritten the thing; but this is easy to make the running the cell a lot of times and then it will be fine.

using the best glm fucntion; just like when makign teh lienar model, we need to tell it which family of functions to draw from; it’s going to look at a LSRL if we dno’t tell it otherwise

family = binomial tells you to make it logistics.

1. The response variable must be the last column in the dataframe.

MedGPA.2 = MedGPA.1[,c(2:10,1)] #reorder columns with response last  
#bestglm for Model Selection  
head(MedGPA.2)

## Sex BCPM GPA VR PS WS BS MCAT Apps Acceptance  
## 1 F 3.59 3.62 11 9 9 9 38 5 0  
## 2 M 3.75 3.84 12 13 8 12 45 3 1  
## 3 F 3.24 3.23 9 10 5 9 33 19 1  
## 4 F 3.74 3.69 12 11 7 10 40 5 1  
## 5 F 3.53 3.38 9 11 4 11 35 11 1  
## 6 M 3.59 3.72 10 9 7 10 36 5 1

Tell em about teh BIC and BICQ DO the same thing, but same it as an object

The best nmodels will tell you how many best models there arel

the top rowis the best model that you would like

the next four best models are the other best models

BIC = the baysian information criteria ; we are going to use it like mallowCp

calculated like: klog(n) - 2log(L(alpha)); n = sample size k = number of predictors alpha = set of all paramets L(alpha) = probability of obtraining the data which you have, supposing the modelbeing tested was given

*SMaller values indicate preferred models*

tells you we got teh data ttha we give given the model

there si going to be a best, but there migh tnot be a stat difference between teh things;

it’s saying on ce we take teh neg 2log, that its nmmore likely that it gen teh data thta we got the value is based on teh samp size and num predictors it could still bea g odoo number if we have different predictors

most are within 0-2 BIC, so there are not much difference between them .

there sin’t much difference between teh models its easier to get teh data, but it’s not very stats; E don’t really know that, but these look pretty similar

**bestglm for Model Selection** BIC = Bayesian Information Criteria

bestglm(MedGPA.2, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

## BIC  
## BICq equivalent for q in (0.407407122288894, 0.830512766582046)  
## Best Model:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -39.4708940 12.2144951 -3.231480 0.001231510  
## SexM -2.8403423 1.1580871 -2.452616 0.014182182  
## GPA 5.3344003 2.4807386 2.150327 0.031529326  
## PS 1.0247592 0.4722984 2.169728 0.030027451  
## WS -0.7177605 0.3496614 -2.052730 0.040098780  
## BS 1.7914617 0.6434984 2.783941 0.005370279

**Bayesian Information Criteria** k log(n)- 2log(L(θ̂))

n : sample size k : number of predictors θ : set of all parameters. L(θ̂) :probability of obtaining the data which you have, supposing the model being tested was a given.

Selection criteria, similar to Mallow’s Cp Smaller values indicate preferred models

**Comparing Models by BIC** Change in BIC; Evidence against hiher BIC 0-2; Little 2-6; POsitive 6-10; Strong greater than 10; Very strong

MedGPA.2.bestglm = bestglm(MedGPA.2, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

MedGPA.2.bestglm$BestModels

## Sex BCPM GPA VR PS WS BS MCAT Apps Criterion  
## 1 TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE FALSE 51.35809  
## 2 TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE FALSE 52.67338  
## 3 TRUE FALSE TRUE TRUE FALSE TRUE FALSE TRUE FALSE 52.81895  
## 4 TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE 52.85687  
## 5 TRUE TRUE FALSE FALSE TRUE TRUE TRUE FALSE FALSE 53.46655

**Example: Predicting Survival** Data: ICU  
ID Patient ID code Survive 1=patient survived to discharge or 0=patient died Age Age (in years) AgeGroup 1= young (under 50), 2= middle (50-69), 3 = old (70+) Sex 1=female or 0=male Infection 1=infection suspected or 0=no infection SysBP Systolic blood pressure (in mm of Hg) Pulse Heart rate (beats per minute) Emergency 1=emergency admission or 0=elective admission

Find the “best” model for Survival using some or all of these predictors.

data("ICU")  
head(ICU)

## ID Survive Age AgeGroup Sex Infection SysBP Pulse Emergency  
## 1 4 0 87 3 1 1 80 96 1  
## 2 8 1 27 1 1 1 142 88 1  
## 3 12 1 59 2 0 0 112 80 1  
## 4 14 1 77 3 0 0 100 70 0  
## 5 27 0 76 3 1 1 128 90 1  
## 6 28 1 54 2 0 1 142 103 1

#Requirements to use bestglm()  
#1. Only the response and possible predictor variables should be within the dataframe  
ICU.1 <- within(ICU, {ID = NULL}) #delete ID variable  
  
#delete ID variable  
# WHy do we delete teh ID Variable? We probably don't need it because each row = the incident number  
  
#2. The response variable must be the last column in the dataframe.  
#reorder columns with response last; column 1 is now the survived column because the ID column was deleted.  
ICU.2 = ICU.1[,c(2:8,1)] #reorder columns with response last  
  
# AgeGroup is Treated as Quantitative   
head(ICU.2)

## Age AgeGroup Sex Infection SysBP Pulse Emergency Survive  
## 1 87 3 1 1 80 96 1 0  
## 2 27 1 1 1 142 88 1 1  
## 3 59 2 0 0 112 80 1 1  
## 4 77 3 0 0 100 70 0 1  
## 5 76 3 1 1 128 90 1 0  
## 6 54 2 0 1 142 103 1 1

bestglm(ICU.2, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

## BIC  
## BICq equivalent for q in (0.0293273190867612, 0.516811637275042)  
## Best Model:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 5.3032038 0.9752351 5.437872 5.392065e-08  
## AgeGroup -0.8430258 0.2515652 -3.351123 8.048461e-04  
## Emergency -2.5144865 0.7576616 -3.318746 9.042257e-04

# THis tells me that teh best variable to predict survived is Emergency  
  
bestglm(ICU.2, family=binomial)$BestModels

## Morgan-Tatar search since family is non-gaussian.

## Age AgeGroup Sex Infection SysBP Pulse Emergency Criterion  
## 1 FALSE TRUE FALSE FALSE FALSE FALSE TRUE 183.3483  
## 2 FALSE TRUE FALSE FALSE TRUE FALSE TRUE 183.4829  
## 3 TRUE FALSE FALSE FALSE FALSE FALSE TRUE 183.6723  
## 4 TRUE FALSE FALSE FALSE TRUE FALSE TRUE 183.7191  
## 5 FALSE TRUE FALSE TRUE FALSE FALSE TRUE 186.7208

# The criteria doesn't change very much between teh first three models   
# Criteria is teh BIC; we want this to be low   
  
#THe data is teaching Age group as a numerical verabiel, we need to cahnge it to a cateorical variable if we want to look at each age group

*BElow is how to make agegroup a categorical variable* We are reassingin tee variable age group as teh factor of age group, so this breaks it into whatever age groups that are under agegroup category.

ICU\_factor\_AgeGroup = ICU.2   
ICU\_factor\_AgeGroup$AgeGroup = factor(ICU\_factor\_AgeGroup$AgeGroup)  
  
head(ICU\_factor\_AgeGroup)

## Age AgeGroup Sex Infection SysBP Pulse Emergency Survive  
## 1 87 3 1 1 80 96 1 0  
## 2 27 1 1 1 142 88 1 1  
## 3 59 2 0 0 112 80 1 1  
## 4 77 3 0 0 100 70 0 1  
## 5 76 3 1 1 128 90 1 0  
## 6 54 2 0 1 142 103 1 1

below is running the log model on the log model, but wiht age group sections differentiated.

bestglm(ICU\_factor\_AgeGroup, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

## Note: factors present with more than 2 levels.

## BIC  
## Best Model:  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Age 1 1.149 1.1486 8.004 0.00515 \*\*   
## Emergency 1 2.581 2.5811 17.987 3.42e-05 \*\*\*  
## Residuals 197 28.270 0.1435   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

now it’s not using age group; but it’s giving more datapoints there is a change in the amount of predictirs we dont expect the below to be the same as the above ones, because we added more varaibles by levling teh age group

this goes with us a warning: “Factors rpesent with more than 2 level” it’s saying one thing is more than 2 levels; we its telling us taht there are more to teh columns that they give us

ICU\_factor\_AgeGroup\_bestglm = bestglm(ICU\_factor\_AgeGroup, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

## Note: factors present with more than 2 levels.

ICU\_factor\_AgeGroup\_bestglm$BestModels

## Age AgeGroup Sex Infection SysBP Pulse Emergency Criterion  
## 1 TRUE FALSE FALSE FALSE FALSE FALSE TRUE 183.6723  
## 2 TRUE FALSE FALSE FALSE TRUE FALSE TRUE 183.7191  
## 3 FALSE TRUE FALSE FALSE FALSE FALSE TRUE 187.0545  
## 4 FALSE TRUE FALSE FALSE TRUE FALSE TRUE 187.3861  
## 5 TRUE FALSE FALSE TRUE FALSE FALSE TRUE 187.4172

Below is making the age groups, assigning numbers; so if tha agegroup was 2, then put a 1, if it was 3, then put a 1, then the last code removes the agegroup column because we don’t need age group anymore since we included teh dummy predictors in the first two lines of code below.

below is what bestglm is doing. This looked at the data tiself. IT didn’t look at atransformation if we ignore tranformation, then we have the ebst model here.

But should we ignore tranofmromatio?

NOt always.

#Requirements to use bestglm()  
# 3. Create dummy variables for non binary categorical variables.  
  
ICU.2$AgeGroup2 = ifelse(ICU.2$AgeGroup==2,1,0)  
ICU.2$AgeGroup3 = ifelse(ICU.2$AgeGroup==3,1,0)  
ICU.3 <- within(ICU.2, {AgeGroup = NULL}) #delete AgeGroup variable  
ICU.4 = ICU.3[,c(1:6,8,9,7)] #reorder columns with response last  
  
head(ICU.4)

## Age Sex Infection SysBP Pulse Emergency AgeGroup2 AgeGroup3 Survive  
## 1 87 1 1 80 96 1 0 1 0  
## 2 27 1 1 142 88 1 0 0 1  
## 3 59 0 0 112 80 1 1 0 1  
## 4 77 0 0 100 70 0 0 1 1  
## 5 76 1 1 128 90 1 0 1 0  
## 6 54 0 1 142 103 1 1 0 1

bestglm(ICU.4, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

## BIC  
## BICq equivalent for q in (0.0343073257857045, 0.505855168373752)  
## Best Model:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 5.50876248 1.03351106 5.330144 9.813508e-08  
## Age -0.03401617 0.01069436 -3.180759 1.468899e-03  
## Emergency -2.45353515 0.75256981 -3.260209 1.113300e-03

# Comparing Models by BIC  
ICU.4.bestglm = bestglm(ICU.4, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

ICU.4.bestglm$BestModels

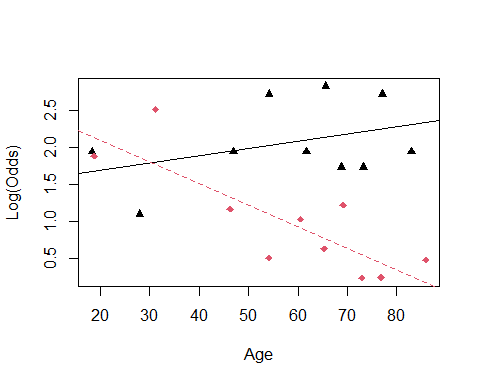
## Age Sex Infection SysBP Pulse Emergency AgeGroup2 AgeGroup3 Criterion  
## 1 TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE 183.6723  
## 2 TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE 183.7191  
## 3 FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE 187.0545  
## 4 FALSE FALSE FALSE TRUE FALSE TRUE TRUE TRUE 187.3861  
## 5 TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE 187.4172

We are assuming that age has teh same impact on surivial as old people; so age in general causes teh same surivial results.

WE can guess tho; if older peopple come in that is going to be different than if younger people are going in for an enermcy. WE can do that with an emperical logit plot.

THis logitplot will help us split by a factor for those brough tin with emergency and not emergency. if you run into errors with emplogitplot, then you can just factor the variables and sometimes that helps. Factor the last variable only, if that doesnt work, then factor others **bestglm for Model Selection**

emplogitplot2(Survive~Age+factor(Emergency), data=ICU.4, ngroups=10)

 We are assuming that age has teh same impact on surivial as old people; so age in general causes teh same surivial results.

WE can guess tho; if older peopple come in that is going to be different than if younger people are going in for an enermcy. WE can do that with an emperical logit plot.

THis logitplot will help us split by a factor for those brough tin with emergency and not emergency. if you run into errors with emplogitplot, then you can just factor the variables and sometimes that helps. Factor the last variable only, if that doesnt work, then factor others

ICU.4$EMAGE = ICU.4$Age\*ICU.4$Emergency  
head(ICU.4)

## Age Sex Infection SysBP Pulse Emergency AgeGroup2 AgeGroup3 Survive EMAGE  
## 1 87 1 1 80 96 1 0 1 0 87  
## 2 27 1 1 142 88 1 0 0 1 27  
## 3 59 0 0 112 80 1 1 0 1 59  
## 4 77 0 0 100 70 0 0 1 1 0  
## 5 76 1 1 128 90 1 0 1 0 76  
## 6 54 0 1 142 103 1 1 0 1 54

ICU.5 = ICU.4[,c(1:8,10,9)] # THis moves surive to teh end of the columns, so that wecan keep doing the code with bestgml.   
head(ICU.5)

## Age Sex Infection SysBP Pulse Emergency AgeGroup2 AgeGroup3 EMAGE Survive  
## 1 87 1 1 80 96 1 0 1 87 0  
## 2 27 1 1 142 88 1 0 0 27 1  
## 3 59 0 0 112 80 1 1 0 59 1  
## 4 77 0 0 100 70 0 0 1 0 1  
## 5 76 1 1 128 90 1 0 1 76 0  
## 6 54 0 1 142 103 1 1 0 54 1

x = bestglm(ICU.5, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

x$BestModels

## Age Sex Infection SysBP Pulse Emergency AgeGroup2 AgeGroup3 EMAGE  
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## 2 FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE  
## 3 FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE  
## 4 TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## 5 FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE  
## Criterion  
## 1 179.1958  
## 2 179.2383  
## 3 183.0363  
## 4 183.6723  
## 5 183.6742

Someitmes best subsets isn’t as useful as we think so; for example: when you have categorical variables, soemtimes they are not immediately reflected through the best mdoels

*Use bestglm when you have binary categorical variables and when you have quantitative variables* IF you want to add interections adn transformations, then it will cause issues

THe ICU dataset was really nice, ti was really clean and easy to work with, but the below is less clean; insurance. How much you pay is based on a huge amount of htings; WHo elese do we haev data on that is like you and how much do we think that you and them are going toet in an accident and cost us moeny

index is just a number, there are 8k people; target flag = accident or no ltager amount= insurance costs - the first 6 rows, the first frow only 1 there are a lot of other types of things; red care = more insuance; previous thing; own a home, etc. so much we could deal with when making this.

THere are some probelms: 1. a lot of the these money variables, are character vectors and not numerical 2. some variabels are not binary, which is okay, but we also see thery’re saved as characters - characters and factors are different, and bestglm doesn’t like characters, they like factors.

WIll look back at this next class

*How to find the variables for the logistical regression models* - Bestglm - backgwards - formard - stepwise

head(insurance)

## # A tibble: 6 x 26  
## INDEX TARGET\_FLAG TARGET\_AMT KIDSDRIV AGE HOMEKIDS YOJ INCOME PARENT1  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <chr>   
## 1 1 0 0 0 60 0 11 $67,349 No   
## 2 2 0 0 0 43 0 11 $91,449 No   
## 3 4 0 0 0 35 1 10 $16,039 No   
## 4 5 0 0 0 51 0 14 <NA> No   
## 5 6 0 0 0 50 0 NA $114,986 No   
## 6 7 1 2946 0 34 1 12 $125,301 Yes   
## # ... with 17 more variables: HOME\_VAL <chr>, MSTATUS <chr>, SEX <chr>,  
## # EDUCATION <chr>, JOB <chr>, TRAVTIME <dbl>, CAR\_USE <chr>, BLUEBOOK <chr>,  
## # TIF <dbl>, CAR\_TYPE <chr>, RED\_CAR <chr>, OLDCLAIM <chr>, CLM\_FREQ <dbl>,  
## # REVOKED <chr>, MVR\_PTS <dbl>, CAR\_AGE <dbl>, URBANICITY <chr>

**Issues with Insurance Data for bestglm**

* Stepwise Regression (Linear Regression) Basic idea: Alternate forward selection and backward elimination

1. Use forward selection to choose a new predictor and check its significance.
2. Use backward elimination to see if predictors already in the model can be dropped.

**Is there a package in R to automate this process?** Yes! The stepAIC function in the MASS package can be used. - But we dont learn how to use it yet

Your task is to investigate the stepAIC function to determine how it can be used to determine the best logistic regression model using the insurance data

Currency\_Convert <- function(Field){  
 Field <- as.numeric(gsub("\\$|,","", Field))  
}  
  
#Change factors to numbers  
insurance$HOME\_VAL\_num = Currency\_Convert(insurance$HOME\_VAL)  
insurance$INCOME\_num = Currency\_Convert(insurance$INCOME)  
insurance$BLUEBOOK\_num = Currency\_Convert(insurance$BLUEBOOK)  
insurance$OLDCLAIM\_num = Currency\_Convert(insurance$OLDCLAIM)  
  
#remove unneeded variables  
insurance.1 = within(insurance,   
 {INDEX = NULL  
 TARGET\_AMT = NULL  
 HOME\_VAL = NULL  
 INCOME = NULL   
 BLUEBOOK = NULL  
 OLDCLAIM = NULL})  
  
  
head(insurance.1)

## # A tibble: 6 x 24  
## TARGET\_FLAG KIDSDRIV AGE HOMEKIDS YOJ PARENT1 MSTATUS SEX EDUCATION   
## <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr>   
## 1 0 0 60 0 11 No z\_No M PhD   
## 2 0 0 43 0 11 No z\_No M z\_High School  
## 3 0 0 35 1 10 No Yes z\_F z\_High School  
## 4 0 0 51 0 14 No Yes M <High School   
## 5 0 0 50 0 NA No Yes z\_F PhD   
## 6 1 0 34 1 12 Yes z\_No z\_F Bachelors   
## # ... with 15 more variables: JOB <chr>, TRAVTIME <dbl>, CAR\_USE <chr>,  
## # TIF <dbl>, CAR\_TYPE <chr>, RED\_CAR <chr>, CLM\_FREQ <dbl>, REVOKED <chr>,  
## # MVR\_PTS <dbl>, CAR\_AGE <dbl>, URBANICITY <chr>, HOME\_VAL\_num <dbl>,  
## # INCOME\_num <dbl>, BLUEBOOK\_num <dbl>, OLDCLAIM\_num <dbl>

insurance.2 = insurance.1[,c(2:24,1)]   
head(insurance.2)

## # A tibble: 6 x 24  
## KIDSDRIV AGE HOMEKIDS YOJ PARENT1 MSTATUS SEX EDUCATION JOB TRAVTIME  
## <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr> <chr> <dbl>  
## 1 0 60 0 11 No z\_No M PhD Profe~ 14  
## 2 0 43 0 11 No z\_No M z\_High Sc~ z\_Blu~ 22  
## 3 0 35 1 10 No Yes z\_F z\_High Sc~ Cleri~ 5  
## 4 0 51 0 14 No Yes M <High Sch~ z\_Blu~ 32  
## 5 0 50 0 NA No Yes z\_F PhD Doctor 36  
## 6 0 34 1 12 Yes z\_No z\_F Bachelors z\_Blu~ 46  
## # ... with 14 more variables: CAR\_USE <chr>, TIF <dbl>, CAR\_TYPE <chr>,  
## # RED\_CAR <chr>, CLM\_FREQ <dbl>, REVOKED <chr>, MVR\_PTS <dbl>, CAR\_AGE <dbl>,  
## # URBANICITY <chr>, HOME\_VAL\_num <dbl>, INCOME\_num <dbl>, BLUEBOOK\_num <dbl>,  
## # OLDCLAIM\_num <dbl>, TARGET\_FLAG <dbl>

#Sad trombone, because best gml wont run here  
insurance.2 = as.data.frame(insurance.2)  
#bestglm(insurance.2, family=binomial)

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

insurance.2.1 = insurance.2 %>% mutate\_if(is.character, factor)  
head(insurance.2.1)

## KIDSDRIV AGE HOMEKIDS YOJ PARENT1 MSTATUS SEX EDUCATION JOB  
## 1 0 60 0 11 No z\_No M PhD Professional  
## 2 0 43 0 11 No z\_No M z\_High School z\_Blue Collar  
## 3 0 35 1 10 No Yes z\_F z\_High School Clerical  
## 4 0 51 0 14 No Yes M <High School z\_Blue Collar  
## 5 0 50 0 NA No Yes z\_F PhD Doctor  
## 6 0 34 1 12 Yes z\_No z\_F Bachelors z\_Blue Collar  
## TRAVTIME CAR\_USE TIF CAR\_TYPE RED\_CAR CLM\_FREQ REVOKED MVR\_PTS CAR\_AGE  
## 1 14 Private 11 Minivan yes 2 No 3 18  
## 2 22 Commercial 1 Minivan yes 0 No 0 1  
## 3 5 Private 4 z\_SUV no 2 No 3 10  
## 4 32 Private 7 Minivan yes 0 No 0 6  
## 5 36 Private 1 z\_SUV no 2 Yes 3 17  
## 6 46 Commercial 1 Sports Car no 0 No 0 7  
## URBANICITY HOME\_VAL\_num INCOME\_num BLUEBOOK\_num OLDCLAIM\_num  
## 1 Highly Urban/ Urban 0 67349 14230 4461  
## 2 Highly Urban/ Urban 257252 91449 14940 0  
## 3 Highly Urban/ Urban 124191 16039 4010 38690  
## 4 Highly Urban/ Urban 306251 NA 15440 0  
## 5 Highly Urban/ Urban 243925 114986 18000 19217  
## 6 Highly Urban/ Urban 0 125301 17430 0  
## TARGET\_FLAG  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 1

#Sadder trombone; because bestglm won't run   
insurance.2.1 = as.data.frame(insurance.2.1)  
#bestglm(insurance.2.1, family=binomial)

## STOR 455 Class 31 R Multiple Logistic Regression Again Agian

library(readr)  
library(bestglm)  
library(MASS)  
  
insurance <- read\_csv("https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/insurance.csv")

**Issues with Insurance Data for bestglm**

head(insurance)

## # A tibble: 6 x 26  
## INDEX TARGET\_FLAG TARGET\_AMT KIDSDRIV AGE HOMEKIDS YOJ INCOME PARENT1  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <chr>   
## 1 1 0 0 0 60 0 11 $67,349 No   
## 2 2 0 0 0 43 0 11 $91,449 No   
## 3 4 0 0 0 35 1 10 $16,039 No   
## 4 5 0 0 0 51 0 14 <NA> No   
## 5 6 0 0 0 50 0 NA $114,986 No   
## 6 7 1 2946 0 34 1 12 $125,301 Yes   
## # ... with 17 more variables: HOME\_VAL <chr>, MSTATUS <chr>, SEX <chr>,  
## # EDUCATION <chr>, JOB <chr>, TRAVTIME <dbl>, CAR\_USE <chr>, BLUEBOOK <chr>,  
## # TIF <dbl>, CAR\_TYPE <chr>, RED\_CAR <chr>, OLDCLAIM <chr>, CLM\_FREQ <dbl>,  
## # REVOKED <chr>, MVR\_PTS <dbl>, CAR\_AGE <dbl>, URBANICITY <chr>

Predict if people are going to get into a car accident How likely that these people get in a car accident based on teh data we are working with

# If it sees a dollar sign or a comma, it changes things to a number like as.numeric   
# So this makes us just have numbers   
# Want 4 new variables in teh dataframe   
Currency\_Convert <- function(Field){  
 Field <- as.numeric(gsub("\\$|,","", Field))  
}  
  
#Change factors to numbers  
insurance$HOME\_VAL\_num = Currency\_Convert(insurance$HOME\_VAL)  
insurance$INCOME\_num = Currency\_Convert(insurance$INCOME)  
insurance$BLUEBOOK\_num = Currency\_Convert(insurance$BLUEBOOK)  
insurance$OLDCLAIM\_num = Currency\_Convert(insurance$OLDCLAIM)  
  
#remove unneeded variables  
# Got rid of them because it would be not known at teh time we wanted to make prediction   
# But we now technically have this data   
# So we are jus getting rid of it   
insurance.1 = within(insurance,   
 {INDEX = NULL  
 TARGET\_AMT = NULL  
 HOME\_VAL = NULL  
 INCOME = NULL   
 BLUEBOOK = NULL  
 OLDCLAIM = NULL})  
  
  
head(insurance.1)

## # A tibble: 6 x 24  
## TARGET\_FLAG KIDSDRIV AGE HOMEKIDS YOJ PARENT1 MSTATUS SEX EDUCATION   
## <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr>   
## 1 0 0 60 0 11 No z\_No M PhD   
## 2 0 0 43 0 11 No z\_No M z\_High School  
## 3 0 0 35 1 10 No Yes z\_F z\_High School  
## 4 0 0 51 0 14 No Yes M <High School   
## 5 0 0 50 0 NA No Yes z\_F PhD   
## 6 1 0 34 1 12 Yes z\_No z\_F Bachelors   
## # ... with 15 more variables: JOB <chr>, TRAVTIME <dbl>, CAR\_USE <chr>,  
## # TIF <dbl>, CAR\_TYPE <chr>, RED\_CAR <chr>, CLM\_FREQ <dbl>, REVOKED <chr>,  
## # MVR\_PTS <dbl>, CAR\_AGE <dbl>, URBANICITY <chr>, HOME\_VAL\_num <dbl>,  
## # INCOME\_num <dbl>, BLUEBOOK\_num <dbl>, OLDCLAIM\_num <dbl>

# Now we have numeric data where we need it

insurance.2 = insurance.1[,c(2:24,1)]   
# For best glm, we want the last column of the data as teh predicted, so we need that to be at t eh end of the dataframe   
# We want target flag at the end   
head(insurance.2)

## # A tibble: 6 x 24  
## KIDSDRIV AGE HOMEKIDS YOJ PARENT1 MSTATUS SEX EDUCATION JOB TRAVTIME  
## <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr> <chr> <dbl>  
## 1 0 60 0 11 No z\_No M PhD Profe~ 14  
## 2 0 43 0 11 No z\_No M z\_High Sc~ z\_Blu~ 22  
## 3 0 35 1 10 No Yes z\_F z\_High Sc~ Cleri~ 5  
## 4 0 51 0 14 No Yes M <High Sch~ z\_Blu~ 32  
## 5 0 50 0 NA No Yes z\_F PhD Doctor 36  
## 6 0 34 1 12 Yes z\_No z\_F Bachelors z\_Blu~ 46  
## # ... with 14 more variables: CAR\_USE <chr>, TIF <dbl>, CAR\_TYPE <chr>,  
## # RED\_CAR <chr>, CLM\_FREQ <dbl>, REVOKED <chr>, MVR\_PTS <dbl>, CAR\_AGE <dbl>,  
## # URBANICITY <chr>, HOME\_VAL\_num <dbl>, INCOME\_num <dbl>, BLUEBOOK\_num <dbl>,  
## # OLDCLAIM\_num <dbl>, TARGET\_FLAG <dbl>

#Sad trombone  
insurance.2 = as.data.frame(insurance.2)  
# bestglm(insurance.2, family=binomial)  
# This doesn't work because the structure of teh dataframe is maniuplated a bit weird; have to make it a dataframe; it still wont make it work all the time, but it might fix it   
# THis time it wont fix this issue

# So to fix that, we will use dplyr to make all the character vectors as factors   
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

insurance.2.1 = insurance.2 %>% mutate\_if(is.character, factor)  
head(insurance.2.1)

## KIDSDRIV AGE HOMEKIDS YOJ PARENT1 MSTATUS SEX EDUCATION JOB  
## 1 0 60 0 11 No z\_No M PhD Professional  
## 2 0 43 0 11 No z\_No M z\_High School z\_Blue Collar  
## 3 0 35 1 10 No Yes z\_F z\_High School Clerical  
## 4 0 51 0 14 No Yes M <High School z\_Blue Collar  
## 5 0 50 0 NA No Yes z\_F PhD Doctor  
## 6 0 34 1 12 Yes z\_No z\_F Bachelors z\_Blue Collar  
## TRAVTIME CAR\_USE TIF CAR\_TYPE RED\_CAR CLM\_FREQ REVOKED MVR\_PTS CAR\_AGE  
## 1 14 Private 11 Minivan yes 2 No 3 18  
## 2 22 Commercial 1 Minivan yes 0 No 0 1  
## 3 5 Private 4 z\_SUV no 2 No 3 10  
## 4 32 Private 7 Minivan yes 0 No 0 6  
## 5 36 Private 1 z\_SUV no 2 Yes 3 17  
## 6 46 Commercial 1 Sports Car no 0 No 0 7  
## URBANICITY HOME\_VAL\_num INCOME\_num BLUEBOOK\_num OLDCLAIM\_num  
## 1 Highly Urban/ Urban 0 67349 14230 4461  
## 2 Highly Urban/ Urban 257252 91449 14940 0  
## 3 Highly Urban/ Urban 124191 16039 4010 38690  
## 4 Highly Urban/ Urban 306251 NA 15440 0  
## 5 Highly Urban/ Urban 243925 114986 18000 19217  
## 6 Highly Urban/ Urban 0 125301 17430 0  
## TARGET\_FLAG  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 1

#Sadder trombone  
insurance.2.1 = as.data.frame(insurance.2.1)  
#bestglm(insurance.2.1, family=binomial)  
# When trying again, it doesn't appear to work still; so we have to try something different   
# NEw error, Error: p = 23, much be <= 15   
# tells us that there are too many variables, so we can't run it

When best glm doesn’t work, let’s use different stepwise. We will do AIC Stepwise

full\_insurance = glm(TARGET\_FLAG~., data=insurance.1, family="binomial")  
summary(full\_insurance)

##   
## Call:  
## glm(formula = TARGET\_FLAG ~ ., family = "binomial", data = insurance.1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5538 -0.7033 -0.3906 0.6213 3.1736   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.474e-01 3.122e-01 -0.472 0.636775   
## KIDSDRIV 3.194e-01 7.059e-02 4.524 6.06e-06 \*\*\*  
## AGE -3.903e-03 4.681e-03 -0.834 0.404452   
## HOMEKIDS 2.313e-02 4.275e-02 0.541 0.588473   
## YOJ -9.356e-03 9.775e-03 -0.957 0.338476   
## PARENT1Yes 4.109e-01 1.266e-01 3.246 0.001170 \*\*   
## MSTATUSz\_No 4.266e-01 1.006e-01 4.240 2.24e-05 \*\*\*  
## SEXz\_F -2.076e-01 1.287e-01 -1.613 0.106647   
## EDUCATIONBachelors -3.762e-01 1.325e-01 -2.840 0.004508 \*\*   
## EDUCATIONMasters -4.433e-01 2.140e-01 -2.072 0.038312 \*   
## EDUCATIONPhD 9.120e-02 2.635e-01 0.346 0.729208   
## EDUCATIONz\_High School -1.126e-03 1.068e-01 -0.011 0.991590   
## JOBDoctor -8.849e-01 3.279e-01 -2.699 0.006958 \*\*   
## JOBHome Maker -3.239e-01 1.669e-01 -1.941 0.052309 .   
## JOBLawyer -1.491e-01 2.120e-01 -0.703 0.481916   
## JOBManager -1.075e+00 1.650e-01 -6.519 7.07e-11 \*\*\*  
## JOBProfessional -2.843e-01 1.418e-01 -2.006 0.044906 \*   
## JOBStudent -3.615e-01 1.528e-01 -2.365 0.018010 \*   
## JOBz\_Blue Collar -1.931e-01 1.204e-01 -1.603 0.108890   
## TRAVTIME 1.569e-02 2.192e-03 7.159 8.15e-13 \*\*\*  
## CAR\_USEPrivate -8.292e-01 1.060e-01 -7.822 5.18e-15 \*\*\*  
## TIF -5.225e-02 8.544e-03 -6.115 9.64e-10 \*\*\*  
## CAR\_TYPEPanel Truck 6.955e-01 1.948e-01 3.570 0.000357 \*\*\*  
## CAR\_TYPEPickup 5.556e-01 1.154e-01 4.816 1.47e-06 \*\*\*  
## CAR\_TYPESports Car 1.107e+00 1.466e-01 7.553 4.27e-14 \*\*\*  
## CAR\_TYPEVan 5.678e-01 1.497e-01 3.793 0.000149 \*\*\*  
## CAR\_TYPEz\_SUV 8.265e-01 1.257e-01 6.575 4.88e-11 \*\*\*  
## RED\_CARyes -2.278e-01 1.032e-01 -2.208 0.027231 \*   
## CLM\_FREQ 2.004e-01 3.320e-02 6.036 1.58e-09 \*\*\*  
## REVOKEDYes 8.521e-01 1.075e-01 7.930 2.19e-15 \*\*\*  
## MVR\_PTS 1.161e-01 1.587e-02 7.312 2.63e-13 \*\*\*  
## CAR\_AGE -3.920e-03 8.897e-03 -0.441 0.659514   
## URBANICITYz\_Highly Rural/ Rural -2.306e+00 1.244e-01 -18.537 < 2e-16 \*\*\*  
## HOME\_VAL\_num -1.420e-06 4.287e-07 -3.312 0.000927 \*\*\*  
## INCOME\_num -3.429e-06 1.432e-06 -2.394 0.016680 \*   
## BLUEBOOK\_num -2.257e-05 6.102e-06 -3.699 0.000216 \*\*\*  
## OLDCLAIM\_num -1.310e-05 4.574e-06 -2.864 0.004186 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 6990.9 on 6044 degrees of freedom  
## Residual deviance: 5362.3 on 6008 degrees of freedom  
## (2116 observations deleted due to missingness)  
## AIC: 5436.3  
##   
## Number of Fisher Scoring iterations: 5

#More sad trombone  
#stepAIC(full\_insurance)  
# Says that some blank values are messing us up, so lets get rid of them

insurance.forstepAIC = na.omit(insurance.1) # Gets rid of the na values   
  
full\_insurance.2 = glm(TARGET\_FLAG~., data=insurance.forstepAIC, family="binomial")  
  
# Given a full model - Backwards selection is the default  
stepAIC(full\_insurance.2)

## Start: AIC=5436.27  
## TARGET\_FLAG ~ KIDSDRIV + AGE + HOMEKIDS + YOJ + PARENT1 + MSTATUS +   
## SEX + EDUCATION + JOB + TRAVTIME + CAR\_USE + TIF + CAR\_TYPE +   
## RED\_CAR + CLM\_FREQ + REVOKED + MVR\_PTS + CAR\_AGE + URBANICITY +   
## HOME\_VAL\_num + INCOME\_num + BLUEBOOK\_num + OLDCLAIM\_num  
##   
## Df Deviance AIC  
## - CAR\_AGE 1 5362.5 5434.5  
## - HOMEKIDS 1 5362.6 5434.6  
## - AGE 1 5363.0 5435.0  
## - YOJ 1 5363.2 5435.2  
## <none> 5362.3 5436.3  
## - SEX 1 5364.9 5436.9  
## - RED\_CAR 1 5367.1 5439.1  
## - INCOME\_num 1 5368.1 5440.1  
## - OLDCLAIM\_num 1 5370.6 5442.6  
## - PARENT1 1 5372.8 5444.8  
## - HOME\_VAL\_num 1 5373.3 5445.3  
## - EDUCATION 4 5382.1 5448.1  
## - BLUEBOOK\_num 1 5376.2 5448.2  
## - MSTATUS 1 5380.1 5452.1  
## - KIDSDRIV 1 5382.6 5454.6  
## - CLM\_FREQ 1 5398.3 5470.3  
## - TIF 1 5401.0 5473.0  
## - JOB 7 5425.0 5485.0  
## - TRAVTIME 1 5413.7 5485.7  
## - MVR\_PTS 1 5416.3 5488.3  
## - REVOKED 1 5424.1 5496.1  
## - CAR\_USE 1 5424.5 5496.5  
## - CAR\_TYPE 5 5444.9 5508.9  
## - URBANICITY 1 5842.8 5914.8  
##   
## Step: AIC=5434.47  
## TARGET\_FLAG ~ KIDSDRIV + AGE + HOMEKIDS + YOJ + PARENT1 + MSTATUS +   
## SEX + EDUCATION + JOB + TRAVTIME + CAR\_USE + TIF + CAR\_TYPE +   
## RED\_CAR + CLM\_FREQ + REVOKED + MVR\_PTS + URBANICITY + HOME\_VAL\_num +   
## INCOME\_num + BLUEBOOK\_num + OLDCLAIM\_num  
##   
## Df Deviance AIC  
## - HOMEKIDS 1 5362.8 5432.8  
## - AGE 1 5363.2 5433.2  
## - YOJ 1 5363.4 5433.4  
## <none> 5362.5 5434.5  
## - SEX 1 5365.1 5435.1  
## - RED\_CAR 1 5367.3 5437.3  
## - INCOME\_num 1 5368.4 5438.4  
## - OLDCLAIM\_num 1 5370.8 5440.8  
## - PARENT1 1 5373.0 5443.0  
## - HOME\_VAL\_num 1 5373.3 5443.3  
## - BLUEBOOK\_num 1 5376.3 5446.3  
## - MSTATUS 1 5380.3 5450.3  
## - EDUCATION 4 5387.3 5451.3  
## - KIDSDRIV 1 5382.8 5452.8  
## - CLM\_FREQ 1 5398.4 5468.4  
## - TIF 1 5401.3 5471.3  
## - JOB 7 5425.2 5483.2  
## - TRAVTIME 1 5413.8 5483.8  
## - MVR\_PTS 1 5416.5 5486.5  
## - REVOKED 1 5424.2 5494.2  
## - CAR\_USE 1 5424.6 5494.6  
## - CAR\_TYPE 5 5445.2 5507.2  
## - URBANICITY 1 5843.0 5913.0  
##   
## Step: AIC=5432.76  
## TARGET\_FLAG ~ KIDSDRIV + AGE + YOJ + PARENT1 + MSTATUS + SEX +   
## EDUCATION + JOB + TRAVTIME + CAR\_USE + TIF + CAR\_TYPE + RED\_CAR +   
## CLM\_FREQ + REVOKED + MVR\_PTS + URBANICITY + HOME\_VAL\_num +   
## INCOME\_num + BLUEBOOK\_num + OLDCLAIM\_num  
##   
## Df Deviance AIC  
## - YOJ 1 5363.5 5431.5  
## - AGE 1 5364.0 5432.0  
## <none> 5362.8 5432.8  
## - SEX 1 5365.4 5433.4  
## - RED\_CAR 1 5367.6 5435.6  
## - INCOME\_num 1 5368.6 5436.6  
## - OLDCLAIM\_num 1 5371.1 5439.1  
## - HOME\_VAL\_num 1 5373.8 5441.8  
## - BLUEBOOK\_num 1 5376.6 5444.6  
## - PARENT1 1 5377.6 5445.6  
## - MSTATUS 1 5380.7 5448.7  
## - EDUCATION 4 5387.8 5449.8  
## - KIDSDRIV 1 5390.0 5458.0  
## - CLM\_FREQ 1 5398.8 5466.8  
## - TIF 1 5401.6 5469.6  
## - JOB 7 5425.4 5481.4  
## - TRAVTIME 1 5414.0 5482.0  
## - MVR\_PTS 1 5416.9 5484.9  
## - REVOKED 1 5424.8 5492.8  
## - CAR\_USE 1 5425.1 5493.1  
## - CAR\_TYPE 5 5445.7 5505.7  
## - URBANICITY 1 5843.1 5911.1  
##   
## Step: AIC=5431.49  
## TARGET\_FLAG ~ KIDSDRIV + AGE + PARENT1 + MSTATUS + SEX + EDUCATION +   
## JOB + TRAVTIME + CAR\_USE + TIF + CAR\_TYPE + RED\_CAR + CLM\_FREQ +   
## REVOKED + MVR\_PTS + URBANICITY + HOME\_VAL\_num + INCOME\_num +   
## BLUEBOOK\_num + OLDCLAIM\_num  
##   
## Df Deviance AIC  
## - AGE 1 5365.0 5431.0  
## <none> 5363.5 5431.5  
## - SEX 1 5366.1 5432.1  
## - RED\_CAR 1 5368.3 5434.3  
## - INCOME\_num 1 5369.8 5435.8  
## - OLDCLAIM\_num 1 5372.0 5438.0  
## - HOME\_VAL\_num 1 5374.5 5440.5  
## - BLUEBOOK\_num 1 5377.4 5443.4  
## - PARENT1 1 5377.9 5443.9  
## - EDUCATION 4 5388.3 5448.3  
## - MSTATUS 1 5382.8 5448.8  
## - KIDSDRIV 1 5390.4 5456.4  
## - CLM\_FREQ 1 5399.6 5465.6  
## - TIF 1 5402.6 5468.6  
## - JOB 7 5425.5 5479.5  
## - TRAVTIME 1 5414.6 5480.6  
## - MVR\_PTS 1 5417.9 5483.9  
## - REVOKED 1 5425.6 5491.6  
## - CAR\_USE 1 5426.2 5492.2  
## - CAR\_TYPE 5 5446.8 5504.8  
## - URBANICITY 1 5843.5 5909.5  
##   
## Step: AIC=5431  
## TARGET\_FLAG ~ KIDSDRIV + PARENT1 + MSTATUS + SEX + EDUCATION +   
## JOB + TRAVTIME + CAR\_USE + TIF + CAR\_TYPE + RED\_CAR + CLM\_FREQ +   
## REVOKED + MVR\_PTS + URBANICITY + HOME\_VAL\_num + INCOME\_num +   
## BLUEBOOK\_num + OLDCLAIM\_num  
##   
## Df Deviance AIC  
## <none> 5365.0 5431.0  
## - SEX 1 5367.1 5431.1  
## - RED\_CAR 1 5369.7 5433.7  
## - INCOME\_num 1 5371.0 5435.0  
## - OLDCLAIM\_num 1 5373.6 5437.6  
## - HOME\_VAL\_num 1 5376.9 5440.9  
## - BLUEBOOK\_num 1 5380.7 5444.7  
## - MSTATUS 1 5383.5 5447.5  
## - PARENT1 1 5383.9 5447.9  
## - EDUCATION 4 5390.0 5448.0  
## - KIDSDRIV 1 5391.4 5455.4  
## - CLM\_FREQ 1 5401.0 5465.0  
## - TIF 1 5404.0 5468.0  
## - TRAVTIME 1 5415.8 5479.8  
## - JOB 7 5428.4 5480.4  
## - MVR\_PTS 1 5420.1 5484.1  
## - REVOKED 1 5427.4 5491.4  
## - CAR\_USE 1 5427.8 5491.8  
## - CAR\_TYPE 5 5447.1 5503.1  
## - URBANICITY 1 5846.8 5910.8

##   
## Call: glm(formula = TARGET\_FLAG ~ KIDSDRIV + PARENT1 + MSTATUS + SEX +   
## EDUCATION + JOB + TRAVTIME + CAR\_USE + TIF + CAR\_TYPE + RED\_CAR +   
## CLM\_FREQ + REVOKED + MVR\_PTS + URBANICITY + HOME\_VAL\_num +   
## INCOME\_num + BLUEBOOK\_num + OLDCLAIM\_num, family = "binomial",   
## data = insurance.forstepAIC)  
##   
## Coefficients:  
## (Intercept) KIDSDRIV   
## -3.905e-01 3.307e-01   
## PARENT1Yes MSTATUSz\_No   
## 4.730e-01 4.146e-01   
## SEXz\_F EDUCATIONBachelors   
## -1.861e-01 -3.995e-01   
## EDUCATIONMasters EDUCATIONPhD   
## -4.969e-01 4.064e-02   
## EDUCATIONz\_High School JOBDoctor   
## -7.396e-03 -9.037e-01   
## JOBHome Maker JOBLawyer   
## -2.872e-01 -1.650e-01   
## JOBManager JOBProfessional   
## -1.091e+00 -2.958e-01   
## JOBStudent JOBz\_Blue Collar   
## -3.133e-01 -2.020e-01   
## TRAVTIME CAR\_USEPrivate   
## 1.557e-02 -8.324e-01   
## TIF CAR\_TYPEPanel Truck   
## -5.237e-02 7.121e-01   
## CAR\_TYPEPickup CAR\_TYPESports Car   
## 5.525e-01 1.090e+00   
## CAR\_TYPEVan CAR\_TYPEz\_SUV   
## 5.778e-01 8.112e-01   
## RED\_CARyes CLM\_FREQ   
## -2.236e-01 2.001e-01   
## REVOKEDYes MVR\_PTS   
## 8.551e-01 1.170e-01   
## URBANICITYz\_Highly Rural/ Rural HOME\_VAL\_num   
## -2.308e+00 -1.469e-06   
## INCOME\_num BLUEBOOK\_num   
## -3.471e-06 -2.370e-05   
## OLDCLAIM\_num   
## -1.326e-05   
##   
## Degrees of Freedom: 6044 Total (i.e. Null); 6012 Residual  
## Null Deviance: 6991   
## Residual Deviance: 5365 AIC: 5431

# trace=FALSE will show only the final model, not each step.  
# Looks at new insurance data   
# Big model at teh end   
stepAIC(full\_insurance.2, trace=FALSE)

##   
## Call: glm(formula = TARGET\_FLAG ~ KIDSDRIV + PARENT1 + MSTATUS + SEX +   
## EDUCATION + JOB + TRAVTIME + CAR\_USE + TIF + CAR\_TYPE + RED\_CAR +   
## CLM\_FREQ + REVOKED + MVR\_PTS + URBANICITY + HOME\_VAL\_num +   
## INCOME\_num + BLUEBOOK\_num + OLDCLAIM\_num, family = "binomial",   
## data = insurance.forstepAIC)  
##   
## Coefficients:  
## (Intercept) KIDSDRIV   
## -3.905e-01 3.307e-01   
## PARENT1Yes MSTATUSz\_No   
## 4.730e-01 4.146e-01   
## SEXz\_F EDUCATIONBachelors   
## -1.861e-01 -3.995e-01   
## EDUCATIONMasters EDUCATIONPhD   
## -4.969e-01 4.064e-02   
## EDUCATIONz\_High School JOBDoctor   
## -7.396e-03 -9.037e-01   
## JOBHome Maker JOBLawyer   
## -2.872e-01 -1.650e-01   
## JOBManager JOBProfessional   
## -1.091e+00 -2.958e-01   
## JOBStudent JOBz\_Blue Collar   
## -3.133e-01 -2.020e-01   
## TRAVTIME CAR\_USEPrivate   
## 1.557e-02 -8.324e-01   
## TIF CAR\_TYPEPanel Truck   
## -5.237e-02 7.121e-01   
## CAR\_TYPEPickup CAR\_TYPESports Car   
## 5.525e-01 1.090e+00   
## CAR\_TYPEVan CAR\_TYPEz\_SUV   
## 5.778e-01 8.112e-01   
## RED\_CARyes CLM\_FREQ   
## -2.236e-01 2.001e-01   
## REVOKEDYes MVR\_PTS   
## 8.551e-01 1.170e-01   
## URBANICITYz\_Highly Rural/ Rural HOME\_VAL\_num   
## -2.308e+00 -1.469e-06   
## INCOME\_num BLUEBOOK\_num   
## -3.471e-06 -2.370e-05   
## OLDCLAIM\_num   
## -1.326e-05   
##   
## Degrees of Freedom: 6044 Total (i.e. Null); 6012 Residual  
## Null Deviance: 6991   
## Residual Deviance: 5365 AIC: 5431

final\_model\_backwards=stepAIC(full\_insurance.2, trace=FALSE)   
  
summary(final\_model\_backwards)

##   
## Call:  
## glm(formula = TARGET\_FLAG ~ KIDSDRIV + PARENT1 + MSTATUS + SEX +   
## EDUCATION + JOB + TRAVTIME + CAR\_USE + TIF + CAR\_TYPE + RED\_CAR +   
## CLM\_FREQ + REVOKED + MVR\_PTS + URBANICITY + HOME\_VAL\_num +   
## INCOME\_num + BLUEBOOK\_num + OLDCLAIM\_num, family = "binomial",   
## data = insurance.forstepAIC)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5691 -0.7024 -0.3901 0.6201 3.1495   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.905e-01 2.287e-01 -1.707 0.087770 .   
## KIDSDRIV 3.307e-01 6.390e-02 5.176 2.27e-07 \*\*\*  
## PARENT1Yes 4.730e-01 1.089e-01 4.342 1.41e-05 \*\*\*  
## MSTATUSz\_No 4.146e-01 9.568e-02 4.334 1.47e-05 \*\*\*  
## SEXz\_F -1.861e-01 1.272e-01 -1.464 0.143274   
## EDUCATIONBachelors -3.995e-01 1.231e-01 -3.245 0.001174 \*\*   
## EDUCATIONMasters -4.969e-01 1.905e-01 -2.609 0.009094 \*\*   
## EDUCATIONPhD 4.064e-02 2.461e-01 0.165 0.868809   
## EDUCATIONz\_High School -7.396e-03 1.063e-01 -0.070 0.944554   
## JOBDoctor -9.037e-01 3.273e-01 -2.761 0.005755 \*\*   
## JOBHome Maker -2.872e-01 1.569e-01 -1.831 0.067098 .   
## JOBLawyer -1.650e-01 2.112e-01 -0.781 0.434854   
## JOBManager -1.091e+00 1.641e-01 -6.650 2.93e-11 \*\*\*  
## JOBProfessional -2.958e-01 1.410e-01 -2.098 0.035938 \*   
## JOBStudent -3.133e-01 1.446e-01 -2.166 0.030305 \*   
## JOBz\_Blue Collar -2.020e-01 1.201e-01 -1.682 0.092562 .   
## TRAVTIME 1.557e-02 2.189e-03 7.114 1.13e-12 \*\*\*  
## CAR\_USEPrivate -8.324e-01 1.059e-01 -7.861 3.82e-15 \*\*\*  
## TIF -5.237e-02 8.537e-03 -6.134 8.56e-10 \*\*\*  
## CAR\_TYPEPanel Truck 7.121e-01 1.944e-01 3.664 0.000249 \*\*\*  
## CAR\_TYPEPickup 5.525e-01 1.153e-01 4.793 1.65e-06 \*\*\*  
## CAR\_TYPESports Car 1.090e+00 1.451e-01 7.512 5.84e-14 \*\*\*  
## CAR\_TYPEVan 5.778e-01 1.494e-01 3.866 0.000110 \*\*\*  
## CAR\_TYPEz\_SUV 8.112e-01 1.247e-01 6.508 7.63e-11 \*\*\*  
## RED\_CARyes -2.236e-01 1.031e-01 -2.168 0.030141 \*   
## CLM\_FREQ 2.001e-01 3.318e-02 6.032 1.62e-09 \*\*\*  
## REVOKEDYes 8.551e-01 1.074e-01 7.965 1.65e-15 \*\*\*  
## MVR\_PTS 1.170e-01 1.585e-02 7.383 1.54e-13 \*\*\*  
## URBANICITYz\_Highly Rural/ Rural -2.308e+00 1.244e-01 -18.552 < 2e-16 \*\*\*  
## HOME\_VAL\_num -1.469e-06 4.267e-07 -3.443 0.000575 \*\*\*  
## INCOME\_num -3.471e-06 1.424e-06 -2.439 0.014744 \*   
## BLUEBOOK\_num -2.370e-05 6.031e-06 -3.930 8.51e-05 \*\*\*  
## OLDCLAIM\_num -1.326e-05 4.569e-06 -2.903 0.003700 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 6990.9 on 6044 degrees of freedom  
## Residual deviance: 5365.0 on 6012 degrees of freedom  
## AIC: 5431  
##   
## Number of Fisher Scoring iterations: 5

none = glm(TARGET\_FLAG~1, data=insurance.forstepAIC, family="binomial")  
  
# Tell it to do forward and stepwise below for an AIC model   
final\_model\_forwards = stepAIC(none, scope=list(upper=full\_insurance.2), direction="forward", trace=FALSE)  
final\_model\_both = stepAIC(none, scope=list(upper=full\_insurance.2), direction = "both", trace=FALSE)

summary(final\_model\_forwards)

##   
## Call:  
## glm(formula = TARGET\_FLAG ~ URBANICITY + JOB + MVR\_PTS + CAR\_TYPE +   
## MSTATUS + REVOKED + CAR\_USE + TRAVTIME + TIF + KIDSDRIV +   
## INCOME\_num + CLM\_FREQ + EDUCATION + BLUEBOOK\_num + PARENT1 +   
## HOME\_VAL\_num + OLDCLAIM\_num + RED\_CAR + SEX, family = "binomial",   
## data = insurance.forstepAIC)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5691 -0.7024 -0.3901 0.6201 3.1495   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.905e-01 2.287e-01 -1.707 0.087770 .   
## URBANICITYz\_Highly Rural/ Rural -2.308e+00 1.244e-01 -18.552 < 2e-16 \*\*\*  
## JOBDoctor -9.037e-01 3.273e-01 -2.761 0.005755 \*\*   
## JOBHome Maker -2.872e-01 1.569e-01 -1.831 0.067098 .   
## JOBLawyer -1.650e-01 2.112e-01 -0.781 0.434854   
## JOBManager -1.091e+00 1.641e-01 -6.650 2.93e-11 \*\*\*  
## JOBProfessional -2.958e-01 1.410e-01 -2.098 0.035938 \*   
## JOBStudent -3.133e-01 1.446e-01 -2.166 0.030305 \*   
## JOBz\_Blue Collar -2.020e-01 1.201e-01 -1.682 0.092562 .   
## MVR\_PTS 1.170e-01 1.585e-02 7.383 1.54e-13 \*\*\*  
## CAR\_TYPEPanel Truck 7.121e-01 1.944e-01 3.664 0.000249 \*\*\*  
## CAR\_TYPEPickup 5.525e-01 1.153e-01 4.793 1.65e-06 \*\*\*  
## CAR\_TYPESports Car 1.090e+00 1.451e-01 7.512 5.84e-14 \*\*\*  
## CAR\_TYPEVan 5.778e-01 1.494e-01 3.866 0.000110 \*\*\*  
## CAR\_TYPEz\_SUV 8.112e-01 1.247e-01 6.508 7.63e-11 \*\*\*  
## MSTATUSz\_No 4.146e-01 9.568e-02 4.334 1.47e-05 \*\*\*  
## REVOKEDYes 8.551e-01 1.074e-01 7.965 1.65e-15 \*\*\*  
## CAR\_USEPrivate -8.324e-01 1.059e-01 -7.861 3.82e-15 \*\*\*  
## TRAVTIME 1.557e-02 2.189e-03 7.114 1.13e-12 \*\*\*  
## TIF -5.237e-02 8.537e-03 -6.134 8.56e-10 \*\*\*  
## KIDSDRIV 3.307e-01 6.390e-02 5.176 2.27e-07 \*\*\*  
## INCOME\_num -3.471e-06 1.424e-06 -2.439 0.014744 \*   
## CLM\_FREQ 2.001e-01 3.318e-02 6.032 1.62e-09 \*\*\*  
## EDUCATIONBachelors -3.995e-01 1.231e-01 -3.245 0.001174 \*\*   
## EDUCATIONMasters -4.969e-01 1.905e-01 -2.609 0.009094 \*\*   
## EDUCATIONPhD 4.064e-02 2.461e-01 0.165 0.868809   
## EDUCATIONz\_High School -7.396e-03 1.063e-01 -0.070 0.944554   
## BLUEBOOK\_num -2.370e-05 6.031e-06 -3.930 8.51e-05 \*\*\*  
## PARENT1Yes 4.730e-01 1.089e-01 4.342 1.41e-05 \*\*\*  
## HOME\_VAL\_num -1.469e-06 4.267e-07 -3.443 0.000575 \*\*\*  
## OLDCLAIM\_num -1.326e-05 4.569e-06 -2.903 0.003700 \*\*   
## RED\_CARyes -2.236e-01 1.031e-01 -2.168 0.030141 \*   
## SEXz\_F -1.861e-01 1.272e-01 -1.464 0.143274   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 6990.9 on 6044 degrees of freedom  
## Residual deviance: 5365.0 on 6012 degrees of freedom  
## AIC: 5431  
##   
## Number of Fisher Scoring iterations: 5

summary(final\_model\_both)

##   
## Call:  
## glm(formula = TARGET\_FLAG ~ URBANICITY + JOB + MVR\_PTS + CAR\_TYPE +   
## MSTATUS + REVOKED + CAR\_USE + TRAVTIME + TIF + KIDSDRIV +   
## INCOME\_num + CLM\_FREQ + EDUCATION + BLUEBOOK\_num + PARENT1 +   
## HOME\_VAL\_num + OLDCLAIM\_num + RED\_CAR + SEX, family = "binomial",   
## data = insurance.forstepAIC)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5691 -0.7024 -0.3901 0.6201 3.1495   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.905e-01 2.287e-01 -1.707 0.087770 .   
## URBANICITYz\_Highly Rural/ Rural -2.308e+00 1.244e-01 -18.552 < 2e-16 \*\*\*  
## JOBDoctor -9.037e-01 3.273e-01 -2.761 0.005755 \*\*   
## JOBHome Maker -2.872e-01 1.569e-01 -1.831 0.067098 .   
## JOBLawyer -1.650e-01 2.112e-01 -0.781 0.434854   
## JOBManager -1.091e+00 1.641e-01 -6.650 2.93e-11 \*\*\*  
## JOBProfessional -2.958e-01 1.410e-01 -2.098 0.035938 \*   
## JOBStudent -3.133e-01 1.446e-01 -2.166 0.030305 \*   
## JOBz\_Blue Collar -2.020e-01 1.201e-01 -1.682 0.092562 .   
## MVR\_PTS 1.170e-01 1.585e-02 7.383 1.54e-13 \*\*\*  
## CAR\_TYPEPanel Truck 7.121e-01 1.944e-01 3.664 0.000249 \*\*\*  
## CAR\_TYPEPickup 5.525e-01 1.153e-01 4.793 1.65e-06 \*\*\*  
## CAR\_TYPESports Car 1.090e+00 1.451e-01 7.512 5.84e-14 \*\*\*  
## CAR\_TYPEVan 5.778e-01 1.494e-01 3.866 0.000110 \*\*\*  
## CAR\_TYPEz\_SUV 8.112e-01 1.247e-01 6.508 7.63e-11 \*\*\*  
## MSTATUSz\_No 4.146e-01 9.568e-02 4.334 1.47e-05 \*\*\*  
## REVOKEDYes 8.551e-01 1.074e-01 7.965 1.65e-15 \*\*\*  
## CAR\_USEPrivate -8.324e-01 1.059e-01 -7.861 3.82e-15 \*\*\*  
## TRAVTIME 1.557e-02 2.189e-03 7.114 1.13e-12 \*\*\*  
## TIF -5.237e-02 8.537e-03 -6.134 8.56e-10 \*\*\*  
## KIDSDRIV 3.307e-01 6.390e-02 5.176 2.27e-07 \*\*\*  
## INCOME\_num -3.471e-06 1.424e-06 -2.439 0.014744 \*   
## CLM\_FREQ 2.001e-01 3.318e-02 6.032 1.62e-09 \*\*\*  
## EDUCATIONBachelors -3.995e-01 1.231e-01 -3.245 0.001174 \*\*   
## EDUCATIONMasters -4.969e-01 1.905e-01 -2.609 0.009094 \*\*   
## EDUCATIONPhD 4.064e-02 2.461e-01 0.165 0.868809   
## EDUCATIONz\_High School -7.396e-03 1.063e-01 -0.070 0.944554   
## BLUEBOOK\_num -2.370e-05 6.031e-06 -3.930 8.51e-05 \*\*\*  
## PARENT1Yes 4.730e-01 1.089e-01 4.342 1.41e-05 \*\*\*  
## HOME\_VAL\_num -1.469e-06 4.267e-07 -3.443 0.000575 \*\*\*  
## OLDCLAIM\_num -1.326e-05 4.569e-06 -2.903 0.003700 \*\*   
## RED\_CARyes -2.236e-01 1.031e-01 -2.168 0.030141 \*   
## SEXz\_F -1.861e-01 1.272e-01 -1.464 0.143274   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 6990.9 on 6044 degrees of freedom  
## Residual deviance: 5365.0 on 6012 degrees of freedom  
## AIC: 5431  
##   
## Number of Fisher Scoring iterations: 5

anova(final\_model\_backwards, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: TARGET\_FLAG  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 6044 6990.9   
## KIDSDRIV 1 42.25 6043 6948.6 8.032e-11 \*\*\*  
## PARENT1 1 121.66 6042 6826.9 < 2.2e-16 \*\*\*  
## MSTATUS 1 32.03 6041 6794.9 1.517e-08 \*\*\*  
## SEX 1 1.26 6040 6793.7 0.261667   
## EDUCATION 4 142.63 6036 6651.0 < 2.2e-16 \*\*\*  
## JOB 7 90.04 6029 6561.0 < 2.2e-16 \*\*\*  
## TRAVTIME 1 10.04 6028 6550.9 0.001530 \*\*   
## CAR\_USE 1 86.88 6027 6464.1 < 2.2e-16 \*\*\*  
## TIF 1 39.38 6026 6424.7 3.488e-10 \*\*\*  
## CAR\_TYPE 5 121.74 6021 6302.9 < 2.2e-16 \*\*\*  
## RED\_CAR 1 0.97 6020 6302.0 0.324811   
## CLM\_FREQ 1 230.56 6019 6071.4 < 2.2e-16 \*\*\*  
## REVOKED 1 80.01 6018 5991.4 < 2.2e-16 \*\*\*  
## MVR\_PTS 1 89.60 6017 5901.8 < 2.2e-16 \*\*\*  
## URBANICITY 1 476.35 6016 5425.5 < 2.2e-16 \*\*\*  
## HOME\_VAL\_num 1 26.75 6015 5398.7 2.317e-07 \*\*\*  
## INCOME\_num 1 9.38 6014 5389.3 0.002193 \*\*   
## BLUEBOOK\_num 1 15.73 6013 5373.6 7.304e-05 \*\*\*  
## OLDCLAIM\_num 1 8.60 6012 5365.0 0.003368 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

G = final\_model\_backwards$null.deviance - final\_model\_backwards$deviance  
Gdf = final\_model\_backwards$df.null - final\_model\_backwards$df.residual  
  
1-pchisq(G, Gdf)

## [1] 0

# how likely this random person that we made up would be to get into an accident   
  
some\_person = data.frame(   
   
 KIDSDRIV = 0,   
 KIDSDRIV = "No",  
 PARENT1 = "No",  
 MSTATUS = "z\_No",   
 SEX="z\_F",   
 EDUCATION = "Masters",   
 JOB= "Professional",   
 TRAVTIME = 15,   
 CAR\_USE = "Private",   
 TIF = 5,   
 CAR\_TYPE = "z\_SUV",   
 RED\_CAR = "no",   
 CLM\_FREQ = 0,   
 REVOKED = "No",   
 MVR\_PTS = 0,   
 URBANICITY = "Highly Urban/ Urban",   
 HOME\_VAL\_num = 258000,  
 INCOME\_num = 82000,   
 BLUEBOOK\_num = 16400,   
 OLDCLAIM\_num = 0  
)

# Predicts logodds  
# Using the final backwards model, this predicts how likely that the person we made up will get into an accident Put type = REsponse if you want probabilitiy   
  
predict(final\_model\_backwards, some\_person)

## 1   
## -2.05658

odds = exp(predict(final\_model\_backwards, some\_person))  
  
odds/(1+odds)

## 1   
## 0.1133892

# Gets teh odds of getting into an accident   
# Same as the predict probability function below

# Predicts probability  
# this predicts the probability that the person we made up will get into an accident  
predict(final\_model\_backwards, some\_person, type = "response")

## 1   
## 0.1133892

predict(final\_model\_forwards, some\_person, type = "response")

## 1   
## 0.1133892

predict(final\_model\_both, some\_person, type = "response")

## 1   
## 0.1133892

## STOR 455 Homework 6

Are Emily and Greg More Employable Than Lakisha and Jamal?

Bertrand, M., & Mullainathan, S. (2004). Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination. *American Economic Review, 94*(4), pp. 991-1013.

*Abstract*

We perform a field experiment to measure racial discrimination in the labor market. We respond with fictitious resumes to help-wanted ads in Boston and Chicago newspapers. To manipulate perception of race, each resume is randomly assigned either a very African American sounding name or a very White sounding name. The results show significant discrimination against African-American names: White names receive 50 percent more callbacks for interviews. We also find that race affects the benefits of a better resume. For White names, a higher quality resume elicits 30 percent more callbacks whereas for African Americans, it elicits a far smaller increase. Applicants living in better neighborhoods receive more callbacks but, interestingly, this effect does not differ by race. The amount of discrimination is uniform across occupations and industries. Federal contractors and employers who list “Equal Opportunity Employer” in their ad discriminate as much as other employers. We find little evidence that our results are driven by employers inferring something other than race, such as social class, from the names. These results suggest that racial discrimination is still a prominent feature of the labor market.

| **Variables** | **Descriptions** |
| --- | --- |
| *call* | Was the applicant called back? (1 = yes; 0 = no) |
| *ethnicity* | indicating ethnicity (i.e., “Caucasian-sounding” vs. “African-American sounding” first name) |
| *sex* | indicating sex |
| *quality* | Indicating quality of resume. |
| *experience* | Number of years of work experience on the resume |
| *equal* | Is the employer EOE (equal opportunity employment)? |

Use the *ResumeNames455* found at the address below:

<https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/ResumeNames455.csv>

library(readr)

ResumeNames455 = read\_csv("https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/ResumeNames455.csv")

1. Construct a logistic model to predict if the job applicant was called back using *experience* as the predictor variable.

mod1 = glm(call~experience, data=ResumeNames455, family = binomial)

summary(mod1)

##

## Call:

## glm(formula = call ~ experience, family = binomial, data = ResumeNames455)

##

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -0.7780 -0.4075 -0.3924 -0.3779 2.3598

##

## Coefficients:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) -2.75960 0.09620 -28.687 < 2e-16 \*\*\*

## experience 0.03908 0.00918 4.257 2.07e-05 \*\*\*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## (Dispersion parameter for binomial family taken to be 1)

##

## Null deviance: 2726.9 on 4869 degrees of freedom

## Residual deviance: 2710.2 on 4868 degrees of freedom

## AIC: 2714.2

##

## Number of Fisher Scoring iterations: 5

1. Plot the raw data and the logistic curve on the same axes.

plot(jitter(call, amount=0.1)~experience, data=ResumeNames455)

logit = function(B0, B1, x){

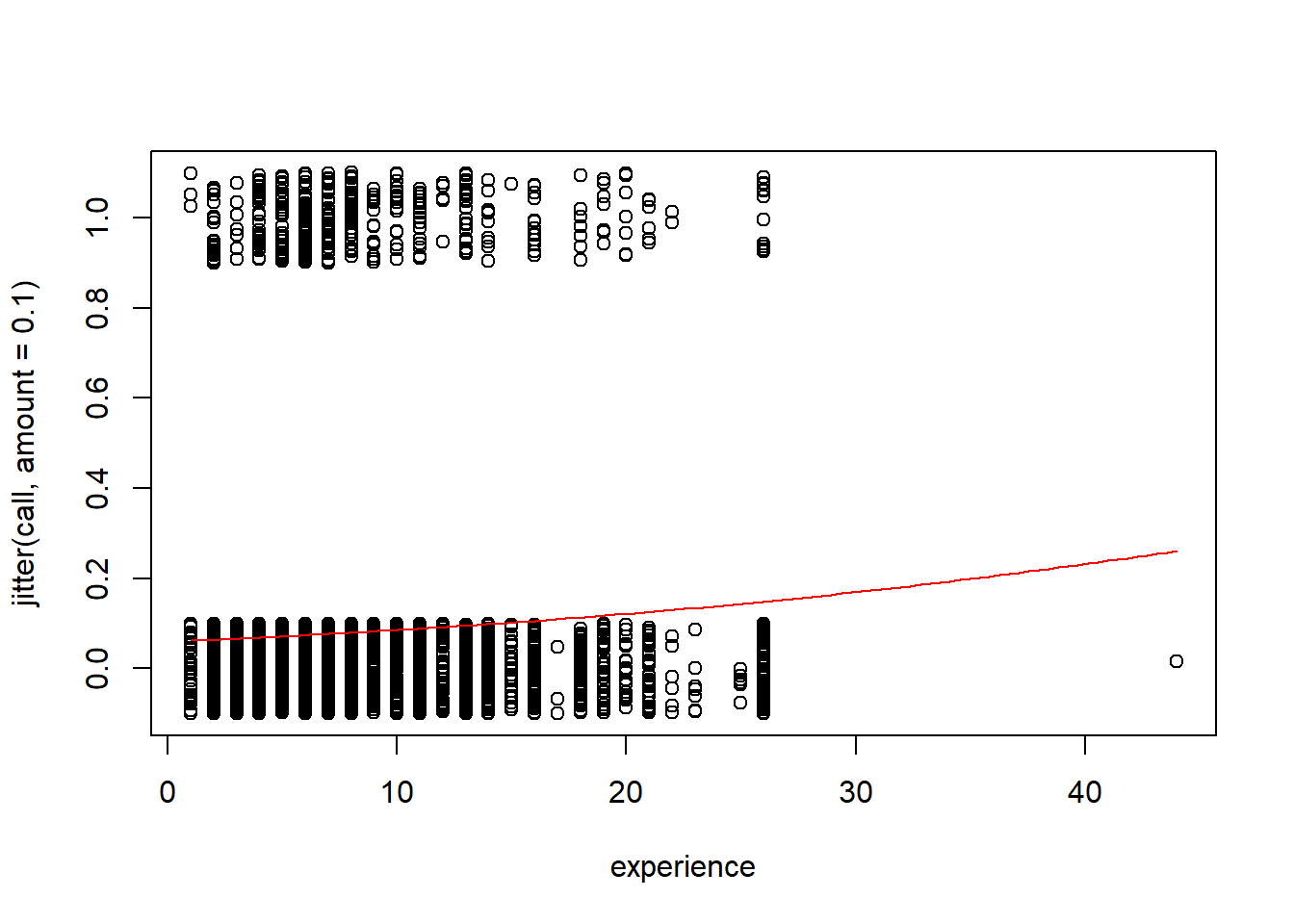
exp(B0 + B1\*x)/(1 + exp(B0 + B1\*x))

}

B0 = summary(mod1)$coef[1]

B1 = summary(mod1)$coef[2]

curve(logit(B0, B1, x), add=TRUE, col="red")



1. For an applicant with 3 years of experience, what does your model predict is the probability of this applicant getting called back?

# without type='response' you are predicting the log(odds)

applicant = data.frame(experience = 3)

predict(mod1, applicant, type="response")

## 1

## 0.06646115

1. Construct an empirical logit plot and comment on the linearity of the data.

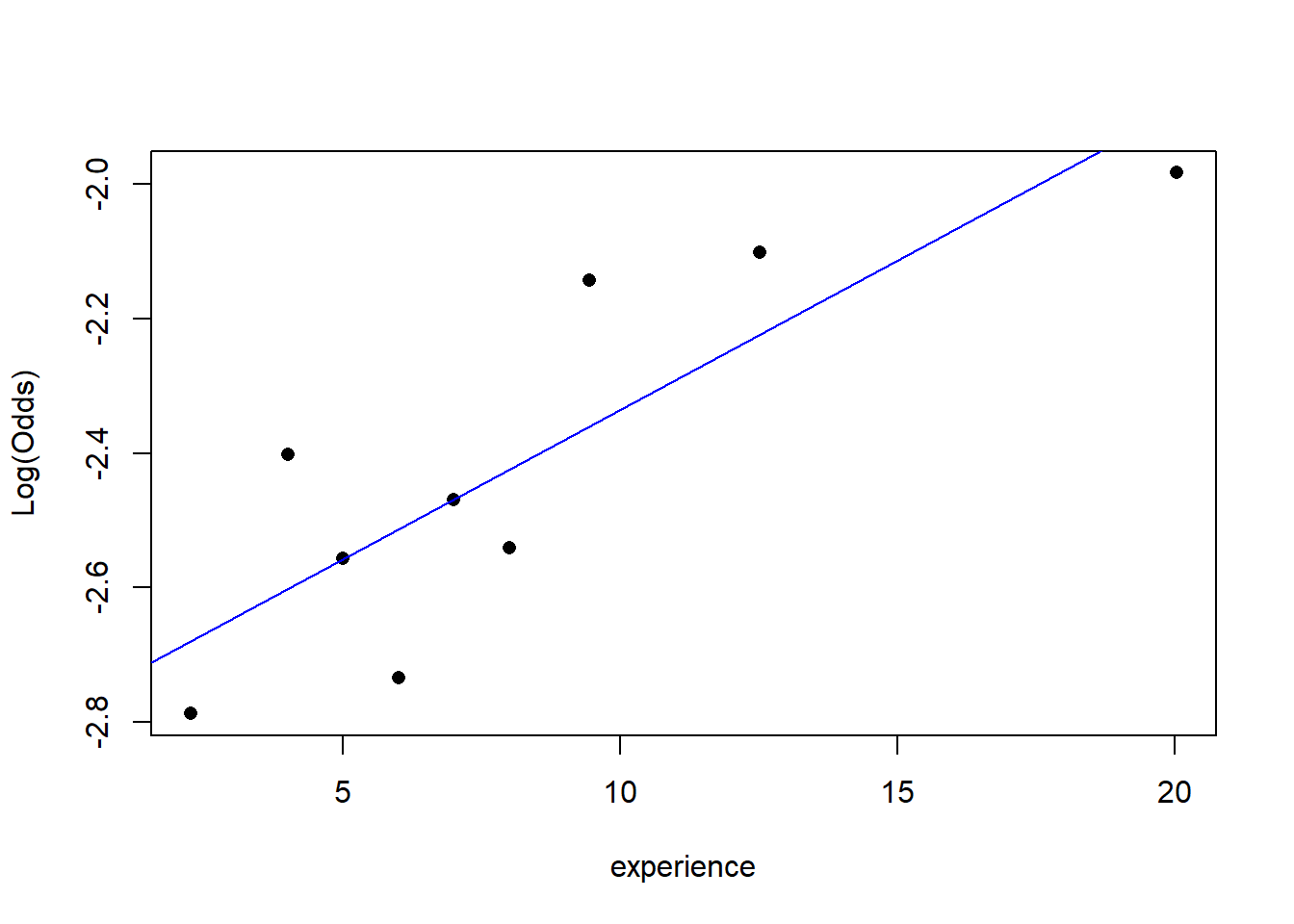
There are no clear nonlinear patterns in the data, so the logistic model seems appropriate

library(Stat2Data)

# ngroups=9 is arbitrarily chosen.

# This data was causes error in the function for many choices of breaks

emplogitplot1(call~experience, data=ResumeNames455, ngroups = 9)



1. Use the model from question #1 to perform a hypothesis test to determine if there is significant evidence of a relationship between *call* and *experience*. Cite your hypotheses, p-value, and conclusion in context.

H0: β1 = 0  
HA: β1 ≠ 0

Since the p-value (2.07e-05 using the summary or 4.298e-05 using log likelihood from anova) is less than 0.05, there is evidence to suggest that the coefficient of the experience term in the binary logistic model is nonzero.

# Either method could be used

summary(mod1)

##

## Call:

## glm(formula = call ~ experience, family = binomial, data = ResumeNames455)

##

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -0.7780 -0.4075 -0.3924 -0.3779 2.3598

##

## Coefficients:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) -2.75960 0.09620 -28.687 < 2e-16 \*\*\*

## experience 0.03908 0.00918 4.257 2.07e-05 \*\*\*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## (Dispersion parameter for binomial family taken to be 1)

##

## Null deviance: 2726.9 on 4869 degrees of freedom

## Residual deviance: 2710.2 on 4868 degrees of freedom

## AIC: 2714.2

##

## Number of Fisher Scoring iterations: 5

anova(mod1, test="Chisq")

|  |
| --- |
|  |

|  | **Df**  **<int>** | **Deviance**  **<dbl>** | **Resid. Df**  **<int>** | **Resid. Dev**  **<dbl>** | **Pr(>Chi)**  **<dbl>** |
| --- | --- | --- | --- | --- | --- |
| NULL | NA | NA | 4869 | 2726.921 | NA |
| experience | 1 | 16.73516 | 4868 | 2710.186 | 4.2977e-05 |

2 rows

1. Construct a confidence interval for the odds ratio for your model and include a sentence interpreting the interval in the context.

We are 95% confident that for each 1 year increase in experience, the odds of getting called back will increase by a factor between approximately 1.02 and 1.06.

exp(confint(mod1))

## Waiting for profiling to be done...

## 2.5 % 97.5 %

## (Intercept) 0.05235403 0.07634629

## experience 1.02097262 1.05841947

1. For each 5-year increase in *experience*, how does your model predict the odds will change for the applicant getting called back?

For each 5 year increase in experience, the odds of getting called back will increase by a factor of approximately 1.22.

exp(summary(mod1)$coef[2,1]) ^ 5

## [1] 1.215796

# or by multiplying before using exp()

exp(5 \* summary(mod1)$coef[2,1])

## [1] 1.215796

In homework #7 we will continue with this data to investigate how the other variables impact an applicant’s chances of being called back.

## STOR 455 Practice Exam Solutions

library(readr)  
library(leaps)  
library(bestglm)  
library(MASS)  
  
BirthWeight <- read\_csv("https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/BirthWeight.csv")  
abalone\_train <- read\_csv("https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/abalone\_train.csv")  
abalone\_test <- read\_csv("https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/abalone\_test.csv")

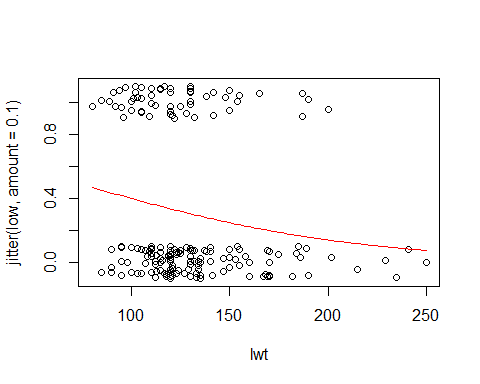
#### Question 1

Low birth weight is an outcome that has been of concern to physicians for years. This is due to the fact that infant mortality rates and birth defect rates are very high for low birth weight babies. Behavior during pregnancy (including diet, smoking habits, and receiving prenatal care) can greatly alter the chances of carrying the baby to term and, consequently, of delivering a baby of normal birth weight. Data were collected at Baystate Medical Center, Springfield, Massachusetts, in 1986 for variables (shown in the table below) that have been shown to be associated with low birth weight in the obstetrical literature.

| Variable | Description |
| --- | --- |
| low | indicator of child’s birth weight less than 2.5 kg. |
| age | mother’s age in years. |
| lwt | mother’s weight in pounds at last menstrual period. |
| race | mother’s race (1 = white, 2 = black, 3 = other). |
| smoke | smoking status during pregnancy. |
| ptl | number of previous premature labours. |
| ht | history of hypertension |
| ui | presence of uterine irritability. |
| ftv | number of physician visits during the first trimester. |
| bwt | child’s birth weight in grams. |

1. Construct and plot a model using the indicator for a child’s low birth weight, *low*, as the response variable, and the mother’s weight in pounds at last menstrual period, *lwt*, and the predictor.

mod1A=glm(low~lwt, data=BirthWeight, family="binomial")  
plot(jitter(low, amount=.1)~lwt, data=BirthWeight)  
  
logit = function(B0, B1, x)  
{  
 exp(B0+B1\*x)/(1+exp(B0+B1\*x))  
}  
  
B0 = summary(mod1A)$coef[1]  
B1 = summary(mod1A)$coef[2]  
  
curve(logit(B0, B1, x),add=TRUE, col="red")



1. Construct a model using the indicator for a child’s low birth weight, *low*, as the response variable, and the mother’s weight in pounds at last menstrual period, age, smoking status during pregnancy, and race as the predictor variables.

mod1B=glm(low~lwt+age+smoke+factor(race), data=BirthWeight, family="binomial")

1. Is there evidence to suggest that the model constructed in part (B) is significantly better than the model constructed in part (A)? Conduct the appropriate hypothesis test. State hypotheses, and provide a conclusion in the context of the data. *6 pts*

Null: The coefficients for the variables age, smoke, and (both dummy) races are 0;  
Alternative: The coefficients for at least one of the variables age, smoke, and race are not 0.  
Statistically significant evidence suggests that at least one of the additional terms has a nonzero coefficient, thus making for a better model than the one with a single predictor.

anova(mod1A, mod1B, test="Chisq")

## Analysis of Deviance Table  
##   
## Model 1: low ~ lwt  
## Model 2: low ~ lwt + age + smoke + factor(race)  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 187 228.69   
## 2 183 214.58 4 14.113 0.006942 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#You may instead examine the summaries of these models to find the difference between the residual deviances, and compare this G statistic to the chi-squared distribution with 4 df.  
  
summary(mod1A)

##   
## Call:  
## glm(formula = low ~ lwt, family = "binomial", data = BirthWeight)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0951 -0.9022 -0.8018 1.3609 1.9821   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.99831 0.78529 1.271 0.2036   
## lwt -0.01406 0.00617 -2.279 0.0227 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 234.67 on 188 degrees of freedom  
## Residual deviance: 228.69 on 187 degrees of freedom  
## AIC: 232.69  
##   
## Number of Fisher Scoring iterations: 4

summary(mod1B)

##   
## Call:  
## glm(formula = low ~ lwt + age + smoke + factor(race), family = "binomial",   
## data = BirthWeight)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5173 -0.9065 -0.5865 1.3035 2.0401   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.332452 1.107672 0.300 0.76407   
## lwt -0.012526 0.006386 -1.961 0.04982 \*   
## age -0.022478 0.034170 -0.658 0.51065   
## smoke 1.054439 0.380000 2.775 0.00552 \*\*  
## factor(race)2 1.231671 0.517152 2.382 0.01724 \*   
## factor(race)3 0.943263 0.416232 2.266 0.02344 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 234.67 on 188 degrees of freedom  
## Residual deviance: 214.58 on 183 degrees of freedom  
## AIC: 226.58  
##   
## Number of Fisher Scoring iterations: 4

G = summary(mod1A)$deviance - summary(mod1B)$deviance  
1 - pchisq(G, 4)

## [1] 0.006941683

1. Use one of the model selection procedures covered in class to determine the best model to predict the indicator for a child’s low birth weight, *low*.

The “best” models produced each way look a bit different, since AIC and BIC values are not directly comparable. Each method has positives and negatives. The models are quite different! BIC penalizes model complexity more heavily, hence the “best” models have fewer terms.

#With bestglm  
  
# Must factor race so it is considered categorical  
  
BirthWeight$race = as.factor(BirthWeight$race)  
  
# Must move low to last column and remove bwt.   
  
# bwt is the baby's birth weight, which will directly correspond  
# to low and cause an error. When predicting if a baby has low   
# birth weight, you won;t know their birth weight first  
  
BirthWeight\_forbestglm = BirthWeight[,c(2:9, 1)]  
head(BirthWeight\_forbestglm)

## # A tibble: 6 x 9  
## age lwt race smoke ptl ht ui ftv low  
## <dbl> <dbl> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 19 182 2 0 0 0 1 0 0  
## 2 33 155 3 0 0 0 0 3 0  
## 3 20 105 1 1 0 0 0 1 0  
## 4 21 108 1 1 0 0 1 2 0  
## 5 18 107 1 1 0 0 1 0 0  
## 6 21 124 3 0 0 0 0 0 0

#This line is sometimes need to restore structure to the dataframe  
BirthWeight\_forbestglm = as.data.frame(BirthWeight\_forbestglm)  
  
bestglm1D = bestglm(BirthWeight\_forbestglm, family=binomial)

## Morgan-Tatar search since family is non-gaussian.

## Note: factors present with more than 2 levels.

bestglm1D$BestModels

## age lwt race smoke ptl ht ui ftv Criterion  
## 1 FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE 231.6256  
## 2 FALSE TRUE FALSE FALSE TRUE TRUE FALSE FALSE 231.6890  
## 3 FALSE TRUE FALSE FALSE FALSE TRUE TRUE FALSE 232.3381  
## 4 FALSE TRUE FALSE TRUE FALSE TRUE FALSE FALSE 232.5830  
## 5 FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE 233.1344

#With stepAIC  
  
BirthWeight\_foretepAIC = BirthWeight[1:9]  
  
# I factored the variable race first above.   
# You could have factored it inside of the glm() function.  
  
mod1D = glm(low~., data=BirthWeight\_foretepAIC, family="binomial")  
none = glm(low~1, data=BirthWeight\_foretepAIC, family="binomial")  
  
# You could have dine this with backwards or forwards as well.  
  
stepAIC(none, scope = list(upper = mod1D), trace=0)

##   
## Call: glm(formula = low ~ ptl + lwt + ht + race + smoke + ui, family = "binomial",   
## data = BirthWeight\_foretepAIC)  
##   
## Coefficients:  
## (Intercept) ptl lwt ht race2 race3   
## -0.08655 0.50321 -0.01591 1.85504 1.32572 0.89708   
## smoke ui   
## 0.93873 0.78570   
##   
## Degrees of Freedom: 188 Total (i.e. Null); 181 Residual  
## Null Deviance: 234.7   
## Residual Deviance: 202 AIC: 218

#### Question 2

Abalone are marine gastropod molluscs, which means they are marine snails.The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope – a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

| Variable | Description |
| --- | --- |
| Sex | M, F |
| Length | longest shell measurement in mm |
| Diameter | perpendicular to length in mm |
| Height | with meat in shell in mm |
| Whole weight | whole abalone in g |
| Shucked weight | weight of meat in g |
| Viscera weight | gut weight (after bleeding) in g |
| Shell weight | after being dried in g |
| Rings | number of rings |

1. Construct a model to predict abalones’ age (using *rings* as the response) with the lowest Mallow’s Cp using any/all of the variables in the *abalone train* dataset. Do not use transformations, or second or greater order terms, or perform an analysis of the residuals.

# model with regsubsets  
  
regsubsets2A=regsubsets(rings~., data=abalone\_train)  
  
source("https://raw.githubusercontent.com/JA-McLean/STOR455/master/scripts/ShowSubsets.R")  
ShowSubsets(regsubsets2A)

## sexM length diameter height weight\_whole weight\_shucked weight\_viscera  
## 1 ( 1 )   
## 2 ( 1 ) \* \*   
## 3 ( 1 ) \* \* \*  
## 4 ( 1 ) \* \* \* \*  
## 5 ( 1 ) \* \* \* \* \*  
## 6 ( 1 ) \* \* \* \* \*  
## 7 ( 1 ) \* \* \* \* \* \*  
## 8 ( 1 ) \* \* \* \* \* \* \*  
## weight\_shell Rsq adjRsq Cp  
## 1 ( 1 ) \* 19.40 19.29 227.63  
## 2 ( 1 ) 35.60 35.41 44.03  
## 3 ( 1 ) 37.49 37.22 24.31  
## 4 ( 1 ) 39.05 38.70 8.45  
## 5 ( 1 ) 39.53 39.10 4.96  
## 6 ( 1 ) \* 39.69 39.17 5.11  
## 7 ( 1 ) \* 39.70 39.09 7.03  
## 8 ( 1 ) \* 39.70 39.00 9.00

mod2A = lm(rings~length+height+weight\_whole+weight\_shucked+weight\_viscera, data=abalone\_train)  
  
# model with step - produces the same best model  
  
none2 = lm(rings~1, data=abalone\_train)  
full = lm(rings~., data=abalone\_train)  
MSE = (summary(full)$sigma)^2  
  
step(none2,scope=list(upper=full),scale=MSE, trace=0)

##   
## Call:  
## lm(formula = rings ~ weight\_shucked + weight\_whole + height +   
## weight\_viscera + length, data = abalone\_train)  
##   
## Coefficients:  
## (Intercept) weight\_shucked weight\_whole height weight\_viscera   
## 4.331 -24.898 14.057 20.404 -14.205   
## length   
## 5.742

#model with forward - produces slightly different model  
  
step(none2,scope=list(upper=full),scale=MSE, direction="forward", trace=0)

##   
## Call:  
## lm(formula = rings ~ weight\_shell + weight\_shucked + weight\_whole +   
## height + weight\_viscera + length, data = abalone\_train)  
##   
## Coefficients:  
## (Intercept) weight\_shell weight\_shucked weight\_whole height   
## 4.534 3.942 -22.957 11.906 18.939   
## weight\_viscera length   
## -12.338 5.383

#model with backward - produces the same best model  
  
step(full,scale=MSE, trace=0)

##   
## Call:  
## lm(formula = rings ~ length + height + weight\_whole + weight\_shucked +   
## weight\_viscera, data = abalone\_train)  
##   
## Coefficients:  
## (Intercept) length height weight\_whole weight\_shucked   
## 4.331 5.742 20.404 14.057 -24.898   
## weight\_viscera   
## -14.205

1. A second dataset, abalone\_test, contains additional data for 500 more abalone. Use this dataset, and your model constructed in part (A), to perform a cross validation analysis of your model. Calcuate and comment on the cross-validation correlation, shrinkage, and analysis of holdout residuals. Does the model constructed in part (A) appear to be similarly effective for predicting the number of rings for abalone?

Holdout residual mean relatively close to zero (close is relative)  
Holdout standard deviation is very similar to the standard error of the regression line for the original model constructed from the training data.  
The shape of the holdout residulas is approximately normally distributed, but might indicate a slight bias and the center seems to be shifted left.  
Shrinkage is near 0.10, which isn’t as small as it could be, but suggests that the model predicts the new data similarly as well as the old.

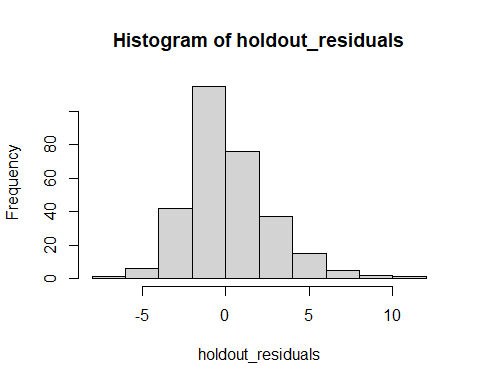
fit = predict(mod2A, abalone\_test)  
  
holdout\_residuals = abalone\_test$rings - fit  
  
mean(holdout\_residuals)

## [1] 0.1286006

sd(holdout\_residuals)

## [1] 2.596266

hist(holdout\_residuals)



summary(mod2A)

##   
## Call:  
## lm(formula = rings ~ length + height + weight\_whole + weight\_shucked +   
## weight\_viscera, data = abalone\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.8210 -1.5879 -0.3408 1.0211 11.9233   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.3314 0.9636 4.495 8.16e-06 \*\*\*  
## length 5.7423 2.4497 2.344 0.019357 \*   
## height 20.4039 5.7006 3.579 0.000369 \*\*\*  
## weight\_whole 14.0574 1.1463 12.263 < 2e-16 \*\*\*  
## weight\_shucked -24.8985 1.5680 -15.879 < 2e-16 \*\*\*  
## weight\_viscera -14.2047 2.8975 -4.902 1.18e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.439 on 694 degrees of freedom  
## Multiple R-squared: 0.3953, Adjusted R-squared: 0.391   
## F-statistic: 90.74 on 5 and 694 DF, p-value: < 2.2e-16

cv\_corr = cor(fit, abalone\_test$rings)  
  
summary(mod2A)$r.squared - cv\_corr^2

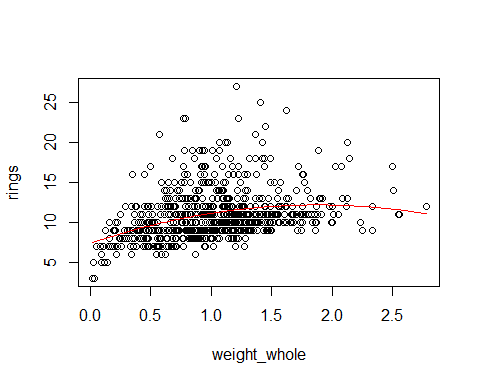
## [1] 0.09549159

1. Linearity is an issue in any abalone model that uses the various measures of weight to predict the number of rings. Would a polynomial model be more appropriate? Contruct a quadratic model using *rings* as the response and the *weight whole* as the predictor. Plot the data and the curve on the same axes. Use the *abalone train* dataset.

mod2C = lm(rings~weight\_whole+I(weight\_whole^2), data=abalone\_train)  
  
summary(mod2C)

##   
## Call:  
## lm(formula = rings ~ weight\_whole + I(weight\_whole^2), data = abalone\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.8056 -1.9640 -0.8028 1.0129 15.4238   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.3778 0.4598 16.045 < 2e-16 \*\*\*  
## weight\_whole 5.1079 0.8353 6.115 1.61e-09 \*\*\*  
## I(weight\_whole^2) -1.3569 0.3569 -3.802 0.000156 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.952 on 697 degrees of freedom  
## Multiple R-squared: 0.1104, Adjusted R-squared: 0.1079   
## F-statistic: 43.26 on 2 and 697 DF, p-value: < 2.2e-16

a = summary(mod2C)$coef[3,1]  
b = summary(mod2C)$coef[2,1]  
c = summary(mod2C)$coef[1,1]  
  
plot(rings~weight\_whole, data=abalone\_train)  
curve(a\*x^2 + b\*x + c, add=TRUE, col="red")



1. Consider a model that uses the log( *rings* ) as the response variable. The predictor variables for the model are *diameter*, *length*, *sex*, and the interactions between *sex* and each other predictor variable. Perform a hypothesis test to determine if the model including the interaction terms is significantly better than a model including the same variables but without the interactions. Include the hypotheses and conclusion.

null: coefficients for the interaction terms are 0;  
alternative: the coefficients for at least one interaction term is nonzero.  
Since the p-value is small, there is significant evidence to suggest that at least one of the interaction terms have a nonzero coefficient.

mod2D1 = lm(log(rings)~length+diameter+sex+length\*sex+diameter\*sex, data=abalone\_train)  
  
mod2D2 = lm(log(rings)~length+diameter+sex, data=abalone\_train)  
  
anova(mod2D2, mod2D1)

## Analysis of Variance Table  
##   
## Model 1: log(rings) ~ length + diameter + sex  
## Model 2: log(rings) ~ length + diameter + sex + length \* sex + diameter \*   
## sex  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 696 40.884   
## 2 694 40.153 2 0.73099 6.3172 0.001911 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## STOR 455 Exam #2

**Directions:** This exam is open books, notes, internet, and all things other than direct communication with others. The *LAdata.csv* dataset is needed to complete the exam. This dataset can be imported from the web address below or from the csv file, also attached in this Sakai assignment. You should complete the exam in this R Notebook, including all code, plots, and explanations. For your submission, you should knit the notebook and submit it as a pdf to Gradescope. If you are unable to knit your exam, you should submit the RMD file to Sakai under the ‘Unable to Knit’ tab. The dataset can be found at GitHub at the address below:

<https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/LAdata.csv>

**Should STOR have Undergraduate Learning Assitants? (YES! Starting Fall 2022…)**

Large introductory STEM courses historically have high failure rates, and failing such courses often leads students to change majors or even drop out of college. Instructional innovations such as the Learning Assistant model can influence this trend by changing institutional norms. In collaboration with faculty who teach large-enrollment introductory STEM courses, undergraduate learning assistants (LAs) use research-based instructional strategies designed to encourage active student engagement and elicit student thinking. These instructional innovations help students master the types of skills necessary for college success such as critical thinking and defending ideas. A study was conducted to investigate the relationship between exposure to LA support in large introductory STEM courses and general failure rates in introductory courses at University of Colorado Boulder.

Alzen, J.L., Langdon, L.S. & Otero, V.K. (2018) A logistic regression investigation of the relationship between the Learning Assistant model and failure rates in introductory STEM courses. *International Journal of STEM Education*, *56*(5). <https://doi.org/10.1186/s40594-018-0152-1>

The *LAdata.csv* dataset represent a subset of the variables examined in this study and a random sample of the data for students in one course, MATH 1300.

| **Variables** | **Descriptions** |
| --- | --- |
| *la\_stud* | Did the student’s course have a learning assistant? (1=yes; 0=no) |
| *sex* | Identified sex of the student (1=male; 0=female) |
| *nonwhite* | Does the student identify as not white? (1=yes; 0=no) |
| *first.gen* | Is the student a first generation college student? (1=yes; 0=no) |
| *finaid\_ever* | Has the student ever receive financial aid? (1=yes; 0=no) |
| *act\_new* | ACT score for the student |
| *hs\_gpa* | High school GPA for the student |
| *credits\_entry* | College credits earned before entering the university |
| *grade* | Grade in the course on a 0.0 to 4.0 scale |
| *fail* | Did the student fail the course? (1=yes; 0=no) |

library(readr)

LAdata <- read\_csv('https://raw.githubusercontent.com/JA-McLean/STOR455/master/data/LAdata.csv')

1. Use the *LAdata.csv* dataset to construct a model to predict if students will *fail* a class using *la\_stud*, *sex*, *nonwhite*, *first.gen*, as well as the interactions between *la\_stud* with each of *sex*, *nonwhite* and *first.gen* as predictors (ie 3 interactions of two variables, with *la\_stud* and each one of the other predictors). Include a summary of this model. *6 pts*

mod1 = glm(fail ~

la\_stud +

nonwhite +

first.gen +

sex +

la\_stud\*nonwhite +

la\_stud\*first.gen +

la\_stud\*sex,

data=LAdata,

family=binomial)

summary(mod1)

##

## Call:

## glm(formula = fail ~ la\_stud + nonwhite + first.gen + sex + la\_stud \*

## nonwhite + la\_stud \* first.gen + la\_stud \* sex, family = binomial,

## data = LAdata)

##

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -0.9828 -0.6412 -0.5567 -0.3909 2.3218

##

## Coefficients:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) -1.22376 0.20397 -6.000 1.98e-09 \*\*\*

## la\_stud -0.44374 0.25337 -1.751 0.0799 .

## nonwhite 0.74711 0.34294 2.179 0.0294 \*

## first.gen -0.09224 0.40773 -0.226 0.8210

## sex -1.30957 0.32798 -3.993 6.53e-05 \*\*\*

## la\_stud:nonwhite -0.55691 0.40915 -1.361 0.1735

## la\_stud:first.gen 0.70932 0.47646 1.489 0.1366

## la\_stud:sex 0.91348 0.38946 2.346 0.0190 \*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## (Dispersion parameter for binomial family taken to be 1)

##

## Null deviance: 959.54 on 1068 degrees of freedom

## Residual deviance: 926.12 on 1061 degrees of freedom

## AIC: 942.12

##

## Number of Fisher Scoring iterations: 5

1. Conduct a hypothesis test at the 0.05 significance level to determine the effectiveness of the *la\_stud* terms in the model constructed in question 1. Cite your hypotheses, p-value, and conclusion in context. *8pts*

*H*0:*β*1=*β*5=*β*6=*β*7=0

*HA*:*At* *least* *one* *of* *β*1,*β*5,*β*6,*β*7≠0

Since the p-value is small (0.04286 < 0.05), there is statistically significant evidence to claim that at least one of the la\_stud terms have a nonzero coefficient.

mod2 = glm(fail ~

nonwhite +

first.gen +

sex,

data=LAdata,

family=binomial)

anova(mod2, mod1, test='Chisq')

|  |
| --- |
|  |

|  | **Resid. Df**  **<dbl>** | **Resid. Dev**  **<dbl>** | **Df**  **<dbl>** | **Deviance**  **<dbl>** | **Pr(>Chi)**  **<dbl>** |
| --- | --- | --- | --- | --- | --- |
| 1 | 1065 | 935.9784 | NA | NA | NA |
| 2 | 1061 | 926.1187 | 4 | 9.859697 | 0.04285856 |

2 rows

# Can also be done this way

G = summary(mod2)$deviance - summary(mod1)$deviance

1 - pchisq(G, 4)

## [1] 0.04285856

1. For non first generation nonwhite female students, what does the model from question 1 predict will be the probability that these students will **pass** the course for courses **with** a learning assistant? What does the model from question 1 predict will be the probability that these students will **pass** the course for courses **without** a learning assistant? *8pts*

s1 = data.frame(sex=0, nonwhite=1, first.gen=0, la\_stud=1)

s2 = data.frame(sex=0, nonwhite=1, first.gen=0, la\_stud=0)

#Probability non first generation nonwhite female student passes with LA

1 - predict(mod1, s1, type='response')

## 1

## 0.8141637

#Probability non first generation nonwhite female student passes without LA

1 - predict(mod1, s2, type='response')

## 1

## 0.6169563

1. Construct a model to predict students’ *grade* in the course using all of the variables (except for *fail*, *hs\_gpa*, and *act\_new*) and including the interactions between *la\_stud* and all of the other variables (except for *fail*, *hs\_gpa*, and *act\_new* and as with question 1, the interactions between *la\_stud* and each one of the other predictors). You should not include transformations, nor a residual analysis. Include a summary of this model. *5 pts*

mod4 = lm(grade~. + la\_stud\*., data=LAdata[-c(6, 7, 10)])

summary(mod4)

##

## Call:

## lm(formula = grade ~ . + la\_stud \* ., data = LAdata[-c(6, 7,

## 10)])

##

## Residuals:

## Min 1Q Median 3Q Max

## -3.2440 -0.6333 0.1551 0.8389 2.2176

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 2.3955419 0.1124618 21.301 < 2e-16 \*\*\*

## la\_stud 0.0026514 0.1343917 0.020 0.984264

## sex 0.4466620 0.1224744 3.647 0.000278 \*\*\*

## nonwhite -0.3225878 0.1495213 -2.157 0.031194 \*

## first.gen 0.1253382 0.1706271 0.735 0.462762

## finaid\_ever -0.4158908 0.1284067 -3.239 0.001237 \*\*

## credits\_entry 0.0272103 0.0084656 3.214 0.001348 \*\*

## la\_stud:sex -0.2341253 0.1478221 -1.584 0.113531

## la\_stud:nonwhite 0.1872411 0.1767231 1.060 0.289606

## la\_stud:first.gen -0.3123343 0.2060254 -1.516 0.129819

## la\_stud:finaid\_ever 0.3500524 0.1542170 2.270 0.023416 \*

## la\_stud:credits\_entry 0.0006063 0.0097250 0.062 0.950300

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 1.106 on 1057 degrees of freedom

## Multiple R-squared: 0.08483, Adjusted R-squared: 0.0753

## F-statistic: 8.907 on 11 and 1057 DF, p-value: 2.621e-15

1. Using **only** the summary output of the model constructed in question 4, What does the coefficient of the *la\_stud* variable (not the interactions, nor the test statistic or p-value) tell you about the relationship between the *la\_stud* and *grade* variables in this model, as well as what specific students this coefficient applies to? *5pts*

The la\_stud coefficient (0.0026514) in the increase in GPA for students in a course with a learning assistant, specifically for students that are female (sex=0), not nonwhite (nonwhite=0), not first generation college students (first.gen=0), have not had financial aid ever (finaid\_ever=0), and have no college credits before entering the university (credits\_entry=0).

1. Perform a **backwards** model selection method (for the lowest Mallow’s Cp) using all of the terms in the model that you constructed in question 4 as possible predictors. Construct this best model and include a summary of this model. *6pts*

MSE = (summary(mod4)$sigma)^2

mod5 = step(mod4, scale=MSE, trace=FALSE)

summary(mod5)

##

## Call:

## lm(formula = grade ~ la\_stud + sex + nonwhite + finaid\_ever +

## credits\_entry + la\_stud:sex + la\_stud:finaid\_ever, data = LAdata[-c(6,

## 7, 10)])

##

## Residuals:

## Min 1Q Median 3Q Max

## -3.1515 -0.6342 0.1717 0.8420 2.2277

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 2.382222 0.108682 21.919 < 2e-16 \*\*\*

## la\_stud 0.019275 0.127257 0.151 0.879639

## sex 0.446036 0.121615 3.668 0.000257 \*\*\*

## nonwhite -0.200140 0.078110 -2.562 0.010536 \*

## finaid\_ever -0.409754 0.122709 -3.339 0.000869 \*\*\*

## credits\_entry 0.028039 0.004157 6.744 2.52e-11 \*\*\*

## la\_stud:sex -0.241861 0.146793 -1.648 0.099725 .

## la\_stud:finaid\_ever 0.314556 0.146807 2.143 0.032369 \*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 1.105 on 1061 degrees of freedom

## Multiple R-squared: 0.08155, Adjusted R-squared: 0.07549

## F-statistic: 13.46 on 7 and 1061 DF, p-value: < 2.2e-16

1. Conduct a hypothesis test at the 0.05 significance level to determine the effectiveness of the *la\_stud* terms in the model constructed in question 6. Cite your hypotheses, p-value, and conclusion in context. *8pts*

*H*0:*β*1=*β*6=*β*7=0

*HA*:*At* *least* *one* *of* *β*1,*β*6,*β*7≠0

Since the p-value is small (0.0447 < 0.05), there is statistically significant evidence to claim that at least one of the la\_stud terms have a nonzero coefficient.

mod7 = lm(grade~sex+nonwhite+finaid\_ever+credits\_entry, data=LAdata)

anova(mod7, mod5)

|  |
| --- |
|  |

|  | **Res.Df**  **<dbl>** | **RSS**  **<dbl>** | **Df**  **<dbl>** | **Sum of Sq**  **<dbl>** | **F**  **<dbl>** | **Pr(>F)**  **<dbl>** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1064 | 1306.490 | NA | NA | NA | NA |
| 2 | 1061 | 1296.602 | 3 | 9.887836 | 2.697048 | 0.0447025 |

2 rows

1. In question 6 I specifically asked you to perform a backwards model selection method rather than a forward or stepwise method. Using the backwards method, *la\_stud* terms were included in your best model. If you had used a forwards or stepwise method, you would find that *la\_stud* terms would **not** be included in your best models. Using the summary of the model from question 6, and knowledge of how the backwards, forwards, and stepwise procedures determine the best model, why would the *la\_stud* terms be included in the backwards model selection output, but not in the forward and stepwise model selection output? *4pts*

backwards will first look to remove the interaction terms and not the individual LA\_stud terms. Since the interaction terms are not all removed from the model in backwards selection, it is never an option to remove the LA\_stud term. For forwards and stepwise, you begin with not predictors. THe interaction terms are not considered unless the individual terms are first added to the model. LA\_stud alone has a high p-value and was never added to the model, so the interaction terms were never considered.

step(mod4, scale=MSE)

## Start: AIC=12

## grade ~ la\_stud + sex + nonwhite + first.gen + finaid\_ever +

## credits\_entry + la\_stud \* (la\_stud + sex + nonwhite + first.gen +

## finaid\_ever + credits\_entry)

##

## Df Sum of Sq RSS Cp

## - la\_stud:credits\_entry 1 0.0048 1292.0 10.004

## - la\_stud:nonwhite 1 1.3721 1293.3 11.123

## <none> 1292.0 12.000

## - la\_stud:first.gen 1 2.8092 1294.8 12.298

## - la\_stud:sex 1 3.0662 1295.0 12.508

## - la\_stud:finaid\_ever 1 6.2977 1298.3 15.152

##

## Step: AIC=10

## grade ~ la\_stud + sex + nonwhite + first.gen + finaid\_ever +

## credits\_entry + la\_stud:sex + la\_stud:nonwhite + la\_stud:first.gen +

## la\_stud:finaid\_ever

##

## Df Sum of Sq RSS Cp

## - la\_stud:nonwhite 1 1.369 1293.3 9.1243

## <none> 1292.0 10.0039

## - la\_stud:first.gen 1 2.830 1294.8 10.3189

## - la\_stud:sex 1 3.075 1295.1 10.5199

## - la\_stud:finaid\_ever 1 6.305 1298.3 13.1617

## - credits\_entry 1 53.907 1345.9 52.1067

##

## Step: AIC=9.12

## grade ~ la\_stud + sex + nonwhite + first.gen + finaid\_ever +

## credits\_entry + la\_stud:sex + la\_stud:first.gen + la\_stud:finaid\_ever

##

## Df Sum of Sq RSS Cp

## - la\_stud:first.gen 1 2.113 1295.5 8.8532

## <none> 1293.3 9.1243

## - la\_stud:sex 1 3.061 1296.4 9.6289

## - nonwhite 1 6.838 1300.2 12.7189

## - la\_stud:finaid\_ever 1 7.168 1300.5 12.9887

## - credits\_entry 1 54.251 1347.6 51.5084

##

## Step: AIC=8.85

## grade ~ la\_stud + sex + nonwhite + first.gen + finaid\_ever +

## credits\_entry + la\_stud:sex + la\_stud:finaid\_ever

##

## Df Sum of Sq RSS Cp

## - first.gen 1 1.134 1296.6 7.7812

## <none> 1295.5 8.8532

## - la\_stud:sex 1 3.322 1298.8 9.5709

## - la\_stud:finaid\_ever 1 5.588 1301.1 11.4249

## - nonwhite 1 6.607 1302.1 12.2582

## - credits\_entry 1 54.412 1349.9 51.3691

##

## Step: AIC=7.78

## grade ~ la\_stud + sex + nonwhite + finaid\_ever + credits\_entry +

## la\_stud:sex + la\_stud:finaid\_ever

##

## Df Sum of Sq RSS Cp

## <none> 1296.6 7.7812

## - la\_stud:sex 1 3.317 1299.9 8.4953

## - la\_stud:finaid\_ever 1 5.610 1302.2 10.3711

## - nonwhite 1 8.023 1304.6 12.3450

## - credits\_entry 1 55.584 1352.2 51.2557

##

## Call:

## lm(formula = grade ~ la\_stud + sex + nonwhite + finaid\_ever +

## credits\_entry + la\_stud:sex + la\_stud:finaid\_ever, data = LAdata[-c(6,

## 7, 10)])

##

## Coefficients:

## (Intercept) la\_stud sex

## 2.38222 0.01927 0.44604

## nonwhite finaid\_ever credits\_entry

## -0.20014 -0.40975 0.02804

## la\_stud:sex la\_stud:finaid\_ever

## -0.24186 0.31456

none = lm(grade~1, data=LAdata[-c(6, 7, 10)])

mod8 = step(none, scope = list(upper=mod4), direction='both')

## Start: AIC=299.28

## grade ~ 1

##

## Df Sum of Sq RSS AIC

## + credits\_entry 1 64.112 1347.6 251.60

## + sex 1 26.302 1385.4 281.18

## + nonwhite 1 12.199 1399.5 292.01

## + finaid\_ever 1 11.344 1400.4 292.66

## + first.gen 1 7.351 1404.4 295.70

## <none> 1411.7 299.28

## + la\_stud 1 2.234 1409.5 299.59

##

## Step: AIC=251.6

## grade ~ credits\_entry

##

## Df Sum of Sq RSS AIC

## + sex 1 20.158 1327.5 237.49

## + finaid\_ever 1 13.418 1334.2 242.90

## + nonwhite 1 11.980 1335.6 244.06

## + first.gen 1 5.499 1342.1 249.23

## <none> 1347.6 251.60

## + la\_stud 1 0.628 1347.0 253.10

## - credits\_entry 1 64.112 1411.7 299.28

##

## Step: AIC=237.49

## grade ~ credits\_entry + sex

##

## Df Sum of Sq RSS AIC

## + finaid\_ever 1 13.289 1314.2 228.73

## + nonwhite 1 11.216 1316.2 230.42

## + first.gen 1 6.512 1321.0 234.23

## <none> 1327.5 237.49

## + la\_stud 1 1.259 1326.2 238.47

## - sex 1 20.158 1347.6 251.60

## - credits\_entry 1 57.968 1385.4 281.18

##

## Step: AIC=228.73

## grade ~ credits\_entry + sex + finaid\_ever

##

## Df Sum of Sq RSS AIC

## + nonwhite 1 7.686 1306.5 224.46

## + first.gen 1 2.550 1311.6 228.66

## <none> 1314.2 228.73

## + la\_stud 1 0.876 1313.3 230.02

## - finaid\_ever 1 13.289 1327.5 237.49

## - sex 1 20.029 1334.2 242.90

## - credits\_entry 1 59.944 1374.1 274.42

##

## Step: AIC=224.46

## grade ~ credits\_entry + sex + finaid\_ever + nonwhite

##

## Df Sum of Sq RSS AIC

## <none> 1306.5 224.46

## + first.gen 1 1.157 1305.3 225.52

## + la\_stud 1 1.079 1305.4 225.58

## - nonwhite 1 7.686 1314.2 228.73

## - finaid\_ever 1 9.759 1316.2 230.42

## - sex 1 19.404 1325.9 238.22

## - credits\_entry 1 59.589 1366.1 270.14