# 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

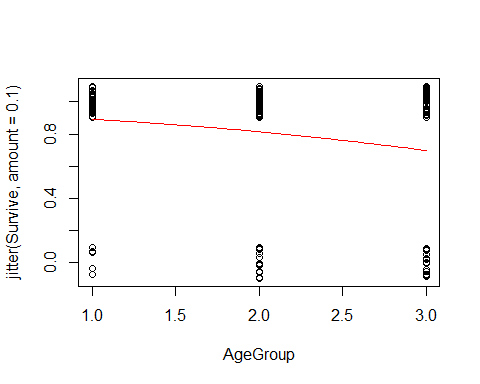
**Categorical Predictors with Multiple Categories in Logistic Regression** - Two approaches: 1. **Method #1:** Logistic regression for Survive with AgeGroup as a quantitative predictor. 2. **Method #2:** Use dummy (indicator) variables for the age categories as predictors in a logistic regression model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

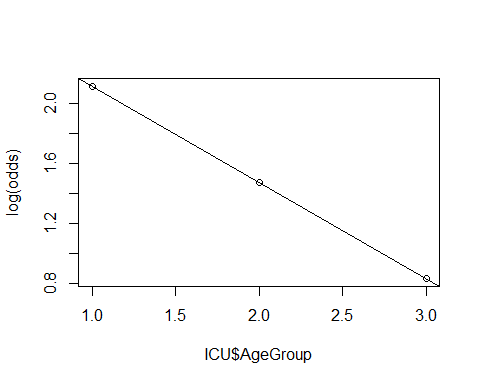
pi = logit(B0, B1, ICU$AgeGroup)  
head(pi)

## [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

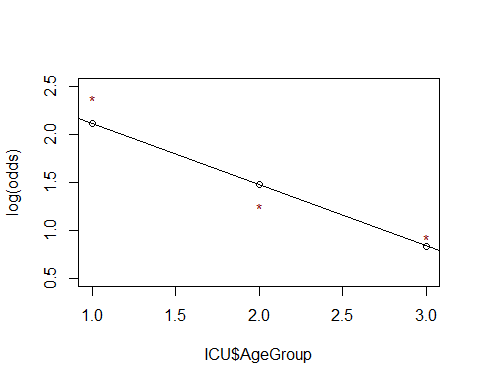
we want to be able to plot this, but it wont work well in a table format, so we need to make thi a dataframe.

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