ST 503 Hw 6

Robin Baldeo

November 23, 2018

# Question 1 (Exercise 2.2)

#### (A)

q1<- glm(Class~., data = wbca, family=binomial)  
q1.s<- summary(q1)  
  
q1.s

##   
## Call:  
## glm(formula = Class ~ ., family = binomial, data = wbca)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.48282 -0.01179 0.04739 0.09678 3.06425   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 11.16678 1.41491 7.892 2.97e-15 \*\*\*  
## Adhes -0.39681 0.13384 -2.965 0.00303 \*\*   
## BNucl -0.41478 0.10230 -4.055 5.02e-05 \*\*\*  
## Chrom -0.56456 0.18728 -3.014 0.00257 \*\*   
## Epith -0.06440 0.16595 -0.388 0.69795   
## Mitos -0.65713 0.36764 -1.787 0.07387 .   
## NNucl -0.28659 0.12620 -2.271 0.02315 \*   
## Thick -0.62675 0.15890 -3.944 8.01e-05 \*\*\*  
## UShap -0.28011 0.25235 -1.110 0.26699   
## USize 0.05718 0.23271 0.246 0.80589   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 881.388 on 680 degrees of freedom  
## Residual deviance: 89.464 on 671 degrees of freedom  
## AIC: 109.46  
##   
## Number of Fisher Scoring iterations: 8

#deviance  
dev<- q1.s$deviance  
  
#degree of freedom  
df<- q1.s$df[2]  
  
#determine the fit  
pchisq(dev,df, lower.tail = F)

## [1] 1

Using just the deviance 89.464195 and the residual degree of freedom 671 is not enough to determine the fit we must find the p-value from the chi square distribution. Since the p-value is greater than .05 we fail to reject the null and conclude that the binominal is a satisfactory fit.

#### (B)

q2<- step(q1, direction = "backward")

## Start: AIC=109.46  
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +   
## UShap + USize  
##   
## Df Deviance AIC  
## - USize 1 89.523 107.52  
## - Epith 1 89.613 107.61  
## - UShap 1 90.627 108.63  
## <none> 89.464 109.46  
## - Mitos 1 93.551 111.55  
## - NNucl 1 95.204 113.20  
## - Adhes 1 98.844 116.84  
## - Chrom 1 99.841 117.84  
## - BNucl 1 109.000 127.00  
## - Thick 1 110.239 128.24  
##   
## Step: AIC=107.52  
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +   
## UShap  
##   
## Df Deviance AIC  
## - Epith 1 89.662 105.66  
## - UShap 1 91.355 107.36  
## <none> 89.523 107.52  
## - Mitos 1 93.552 109.55  
## - NNucl 1 95.231 111.23  
## - Adhes 1 99.042 115.04  
## - Chrom 1 100.153 116.15  
## - BNucl 1 109.064 125.06  
## - Thick 1 110.465 126.47  
##   
## Step: AIC=105.66  
## Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap  
##   
## Df Deviance AIC  
## <none> 89.662 105.66  
## - UShap 1 91.884 105.88  
## - Mitos 1 93.714 107.71  
## - NNucl 1 95.853 109.85  
## - Adhes 1 100.126 114.13  
## - Chrom 1 100.844 114.84  
## - BNucl 1 109.762 123.76  
## - Thick 1 110.632 124.63

q2.s<- summary(q2)  
  
# the best model with the lowest aic is   
q2.s$call

## glm(formula = Class ~ Adhes + BNucl + Chrom + Mitos + NNucl +   
## Thick + UShap, family = binomial, data = wbca)

The best model with the lowest AIC is with class as response and Adhes, BNucl , Chrom , Mitos ,NNucl, Thick ,and UShap as predictors.

#### (C)

# using the parameters from question 1  
#prediction  
pi<- t(as.matrix(q2$coefficients))%\*%as.matrix(c(1,1, 1, 3, 1, 1, 4, 1), nrow= 1)  
pi

## [,1]  
## [1,] 4.834428

#confidience interval  
#using the values from question for prediction  
x0<- as.matrix(c(1,1, 1, 3, 1, 1, 4, 1), nrow= 1)  
  
oo<- eval(q2.s$call)  
  
eta.hat <- sum(x0 \* oo$coefficients)  
p.hat <- ilogit(eta.hat); p.hat

## [1] 0.9921115

Sigma <- (summary(oo))$cov.unscaled  
  
#calucating the se  
se <- sqrt( t(x0) %\*% Sigma %\*% x0 )  
  
#getting the 95% for the prediction   
ci<- ilogit(c(eta.hat - 1.96 \* se, eta.hat + 1.96 \* se))

The Ci is (0.9757467, 0.9974629).

#### (D)

#function to do comparison  
com<- function(o,p){  
 v<- as.numeric(rep(0, length(o)))  
 for(i in 1:length(o)){  
 for(j in 1:length(p)){  
 if(o[i] == p[j]){  
 v[i]= 0  
 break;  
 }else{  
 v[i]= 1  
 }  
 }  
 }  
 return(sum(v))  
}  
  
# malignant  
pre.m<- which(oo$fitted.values<.5)  
or.m<- which(wbca$Class==0)  
  
mi<- com(o= or.m, p = pre.m)  
  
#11 misclassied   
  
#benign  
pre.b<- which(oo$fitted.values>.5)  
or.b<- which(wbca$Class==1)  
  
be<- com(o= or.b, p = pre.b)  
  
#9 misclassfied

With malignant there were 11 and with the benign , there were 9 that were misclassified.

#### (E)

pre.m<- which(oo$fitted.values<.9)  
or.m<- which(wbca$Class==0)  
  
mi<- com(o= or.m, p = pre.m)  
  
#1 misclassied   
  
#benign .9  
pre.b<- which(oo$fitted.values>.9)  
or.b<- which(wbca$Class==1)  
  
be<- com(o= or.b, p = pre.b)  
  
#16 misclassfied

With Malignant there were 1 and with the benign there were 16 that were misclassified. Looking a the .5 cut off and the .9 cut off, I think it is difficult to determine an ideal cut off number. We get very different results with these cut off numbers and choosing the incorrect cut off would results in incorrect classification.

#### (F)

#getting the every 3rd index  
r<- as.numeric(rep(0, nrow(wbca)))  
o<- as.numeric(rep(0, nrow(wbca)))  
for(i in 1:nrow(wbca)){  
 if(i%%3 == 0){  
 r[i]= i  
 }else{  
 o[i]= i  
 }  
}  
#filter out the 0  
r<- r[r>0]  
o<- o[o>0]  
  
test<- wbca[r,,drop= FALSE]  
  
train<- wbca[o,,drop= FALSE]  
  
  
#using training to determine best model with lowest aic  
  
tr<- glm(Class~., data = train, family=binomial)  
summary(tr)

##   
## Call:  
## glm(formula = Class ~ ., family = binomial, data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.98138 -0.00954 0.03310 0.07084 3.07275   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 12.0244 2.0462 5.876 4.19e-09 \*\*\*  
## Adhes -0.4859 0.1555 -3.126 0.00177 \*\*   
## BNucl -0.3732 0.1292 -2.888 0.00388 \*\*   
## Chrom -0.6655 0.2536 -2.625 0.00868 \*\*   
## Epith 0.1779 0.2148 0.828 0.40744   
## Mitos -0.6075 0.5103 -1.190 0.23388   
## NNucl -0.5168 0.1828 -2.828 0.00469 \*\*   
## Thick -0.6533 0.2044 -3.197 0.00139 \*\*   
## UShap -0.5291 0.2612 -2.026 0.04280 \*   
## USize 0.2672 0.2320 1.152 0.24947   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 592.796 on 453 degrees of freedom  
## Residual deviance: 57.651 on 444 degrees of freedom  
## AIC: 77.651  
##   
## Number of Fisher Scoring iterations: 9

tr.1<- step(tr, direction = "backward")

## Start: AIC=77.65  
## Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick +   
## UShap + USize  
##   
## Df Deviance AIC  
## - Epith 1 58.340 76.340  
## - USize 1 58.880 76.880  
## <none> 57.651 77.651  
## - Mitos 1 60.712 78.712  
## - UShap 1 61.450 79.450  
## - Chrom 1 65.983 83.983  
## - BNucl 1 67.373 85.373  
## - NNucl 1 67.538 85.538  
## - Adhes 1 68.073 86.073  
## - Thick 1 71.162 89.162  
##   
## Step: AIC=76.34  
## Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap +   
## USize  
##   
## Df Deviance AIC  
## - USize 1 59.536 75.536  
## <none> 58.340 76.340  
## - Mitos 1 61.264 77.264  
## - UShap 1 61.702 77.702  
## - Chrom 1 66.515 82.515  
## - BNucl 1 67.402 83.402  
## - NNucl 1 67.556 83.556  
## - Adhes 1 68.310 84.310  
## - Thick 1 72.311 88.311  
##   
## Step: AIC=75.54  
## Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap  
##   
## Df Deviance AIC  
## <none> 59.536 75.536  
## - UShap 1 61.894 75.894  
## - Mitos 1 62.329 76.329  
## - Chrom 1 66.762 80.762  
## - NNucl 1 67.576 81.576  
## - BNucl 1 68.332 82.332  
## - Adhes 1 68.359 82.359  
## - Thick 1 72.363 86.363

tr.2<- summary(tr.1)  
  
# the best model with the lowest aic is   
tr\_r<- eval(tr.2$call, tr)  
  
#comparing models  
re<- anova(tr\_r, tr)  
  
pchisq(re$Deviance[2],re$Df[2])

## [1] 0.6103852

#since we have a large p-value greater than alpha then the simpler model is prefered.   
  
#using the test data to the precistion like in part c  
#using the values from question for prediction  
  
oo<- glm(Class ~ Adhes + BNucl + Chrom + Mitos + NNucl + Thick + UShap, family = binomial, data = test)  
  
eta.hat <- sum(x0 \* oo$coefficients)  
p.hat2 <- ilogit(eta.hat);   
p.hat2

## [1] 0.9970556

Here we see the model is identical to the reduced model from part c using the train data. Now we also see that the prediction value using parameters from question 1 (used in part c) 0.9921115 is near identical to the prediction using the test data where prediction is 0.9970556 using the same parameters. Therefore, the process of splitting the data into two parts yields almost the same prediction as part c.

# Question 2 (Exercise 3.1)

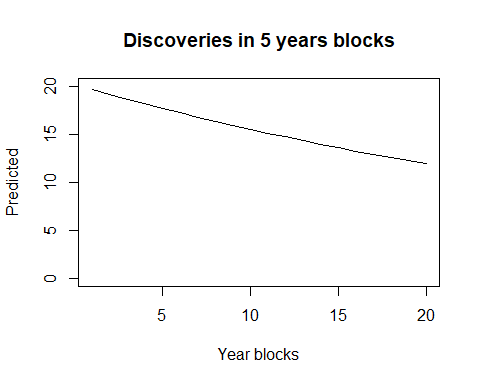
#block the data into groups of 5 X 20 matrix  
m<- matrix(discoveries, ncol = 20)  
  
#variable to hold the sum of the blocks   
rate<- apply(m, 2, sum)  
block<- apply(m, 2, length)  
year<- seq(1:20)  
  
mod\_p1<- glm(rate~ year , family= poisson)  
summary(mod\_p1)

##   
## Call:  
## glm(formula = rate ~ year, family = poisson)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0956 -0.9789 -0.2236 0.7763 3.9335   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.005667 0.111113 27.050 < 2e-16 \*\*\*  
## year -0.026316 0.009918 -2.653 0.00797 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 60.714 on 19 degrees of freedom  
## Residual deviance: 53.625 on 18 degrees of freedom  
## AIC: 147.25  
##   
## Number of Fisher Scoring iterations: 4

mod\_p2<- glm(rate~ offset(log(block))+ year , family= poisson)  
summary(mod\_p2)

##   
## Call:  
## glm(formula = rate ~ offset(log(block)) + year, family = poisson)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0956 -0.9789 -0.2236 0.7763 3.9335   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.396229 0.111113 12.566 < 2e-16 \*\*\*  
## year -0.026316 0.009918 -2.653 0.00797 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 60.714 on 19 degrees of freedom  
## Residual deviance: 53.625 on 18 degrees of freedom  
## AIC: 147.25  
##   
## Number of Fisher Scoring iterations: 4

plot(mod\_p2$fitted.values , ylim = c(0, 20), type= "l", ylab = "Predicted", xlab="Year blocks", main = "Discoveries in 5 years blocks")



#From the plot the rate of discoveries appears to be on the decrease over the years instead of constant.

From the plot the rate of discoveries appears to be on the decrease over the years instead of constant.

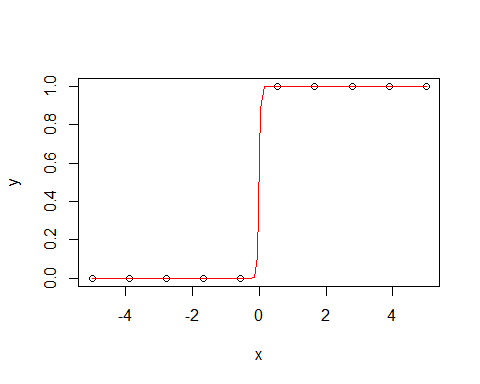
# Question 3

#### (A)

x <- seq(-5, 5, length=10)  
y <- as.numeric(x > 0)  
  
plot(x,y)  
mod<- glm(y~x-1, family = binomial)  
  
summary(mod)

##   
## Call:  
## glm(formula = y ~ x - 1, family = binomial)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.983e-05 -2.110e-08 0.000e+00 2.110e-08 1.983e-05   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## x 40.23 55054.91 0.001 0.999  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1.3863e+01 on 10 degrees of freedom  
## Residual deviance: 7.8648e-10 on 9 degrees of freedom  
## AIC: 2  
##   
## Number of Fisher Scoring iterations: 25

xx <- seq(-5, 5, len=100)  
lines(xx, ilogit(mod$coefficients[1] \* xx), col=2)



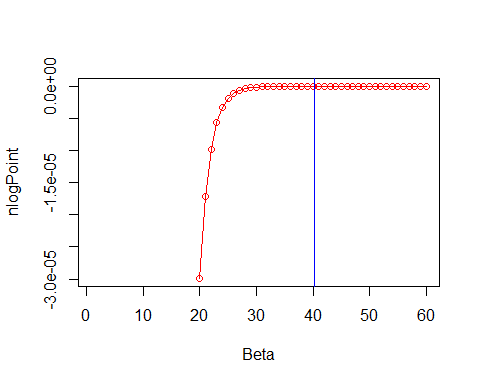
#ci for B  
confint(mod)

## 2.5 % 97.5 %   
## -4205.399 NA

Based on the CI, we can see the breakdown of the MLE. The MLE fails and we cannot utilize the Wald test , therefore we have a lower bound and no upper bound.

#### (B)

loglik <- function(theta) {  
   
 # a <- theta[1]  
 b <- theta[1]  
 #using the logit formula  
 p <- sapply(x, function(p){(exp(1)^(b\*p))/(1+ exp(1)^(b\*p))})  
 o <- sum(dbinom(y, size=1, prob=p, log=TRUE))  
 return(o)  
   
}  
  
nlogPoint<- as.double()  
for( i in 20:60){  
 nlogPoint[i]<- loglik(i)  
}  
  
plot(nlogPoint, xlab = "Beta", col = "red", type = "o")  
  
#vertical line of b-hat  
abline(v=mod$coefficients[1], col = "blue")



Above we see separation from the plot where the plot fits perfectly. This is also evident with the extremely large standard error. Indicating our estimate is junk.

#### (C)

The plot show our function bound by 0 and does not meet a maximum but instead stays constant. I think that is the reason why the standard error is so large.

# Question 4

#### (A)

Given and which follows a Gamma distrubtion. Also, given is with which follows a poisson distrubtion. Therefore using the law of total expectation:

Using the law of total varaince:

proven.

#### (B)

The negative binominal is more flexible because the variance of the poisson and the mean of the poisson distribution are the same leaving less room for flexibility and resulting in overdispersion. As a result from the summary output. However, with the negative binominal distribution the mean and the variance is different.

#### (C)

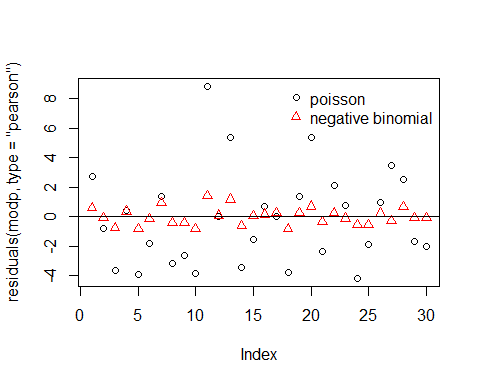
#using model from test book page 64  
modp <- glm(Species ~ .,family=poisson, gala)  
  
summary(modp)

##   
## Call:  
## glm(formula = Species ~ ., family = poisson, data = gala)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.9919 -2.9305 -0.4296 1.3254 7.4735   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.828e+00 5.958e-02 47.471 < 2e-16 \*\*\*  
## Endemics 3.388e-02 1.741e-03 19.459 < 2e-16 \*\*\*  
## Area -1.067e-04 3.741e-05 -2.853 0.00433 \*\*   
## Elevation 2.638e-04 1.934e-04 1.364 0.17264   
## Nearest 1.048e-02 1.611e-03 6.502 7.91e-11 \*\*\*  
## Scruz -6.835e-04 5.802e-04 -1.178 0.23877   
## Adjacent 4.539e-05 4.800e-05 0.946 0.34437   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 3510.73 on 29 degrees of freedom  
## Residual deviance: 313.36 on 23 degrees of freedom  
## AIC: 488.19  
##   
## Number of Fisher Scoring iterations: 5

plot(residuals(modp, type="pearson"))  
  
#negative binominal model  
modnb<- glm(Species ~ .,negative.binomial(1), gala)  
  
summary(modnb)

##   
## Call:  
## glm(formula = Species ~ ., family = negative.binomial(1), data = gala)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3586 -0.5199 -0.1031 0.2427 1.0144   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.479e+00 2.205e-01 11.240 8.07e-11 \*\*\*  
## Endemics 4.901e-02 1.111e-02 4.410 0.000203 \*\*\*  
## Area -2.553e-04 2.620e-04 -0.974 0.340038   
## Elevation 4.206e-06 1.100e-03 0.004 0.996983   
## Nearest 6.177e-03 1.154e-02 0.535 0.597458   
## Scruz -5.246e-04 2.462e-03 -0.213 0.833175   
## Adjacent 9.218e-05 2.746e-04 0.336 0.740096   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.4360555)  
##   
## Null deviance: 54.069 on 29 degrees of freedom  
## Residual deviance: 12.901 on 23 degrees of freedom  
## AIC: 299.91  
##   
## Number of Fisher Scoring iterations: 9

legend("topright", legend = c("poisson","negative binomial"),col=c("black", "red"), pch = c(1,2), bty = "n")  
points(residuals(modnb, type="pearson"), col = 2, pch = 2)  
  
abline(h= 0)



As show in the plot the using the poisson family in the glm(black), we see that the variance is much larger as shown where the points are more spread out from zero. This is different in regards to the negative binominal where the residuals are more clustered around 0.