Chapter 7 Notebook

Code taken from Scott Jones and “Practical Data Science with R” and broken into chunks based on the sections. This is so I may step through each piece and create a notebook for easier reading.

load("C:/Users/turne/Downloads/psub.RData") #Substitute your own location  
dtrain <- subset(psub, ORIGRANDGROUP >= 500)  
dtest <- subset(psub, ORIGRANDGROUP < 500)  
model <- lm(log(PINCP,base=10) ~ AGEP + SEX + COW + SCHL, data=dtrain)  
dtest$predLogPINCP <- predict(model, newdata= dtest)  
dtrain$predLogPINCP <- predict(model, newdata = dtrain)  
head(dtest$predLogPINCP)

## [1] 4.640264 4.465395 4.429962 4.555424 4.235032 4.379018

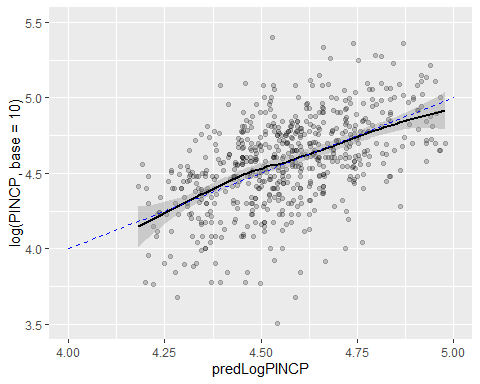
library(ggplot2)  
ggplot(data=dtest, aes(x=predLogPINCP, y=log(PINCP, base=10))) +   
 geom\_point(alpha=0.2, color="black") +  
 geom\_smooth(aes(x=predLogPINCP,  
 y=log(PINCP, base=10)), color="black") +   
 geom\_line(aes(x=log(PINCP, base=10),  
 y=log(PINCP, base=10)), color="blue", linetype=2) +  
 scale\_x\_continuous(limits=c(4,5)) +  
 scale\_y\_continuous(limits=c(3.5,5.5))

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 9 rows containing non-finite values (stat\_smooth).

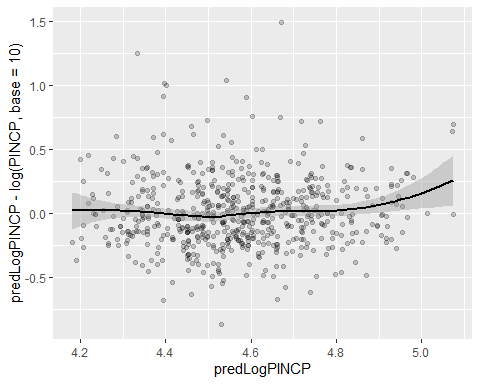
## Warning: Removed 9 rows containing missing values (geom\_point).

## Warning: Removed 67 rows containing missing values (geom\_path).



ggplot(data=dtest, aes(x=predLogPINCP,  
 y=predLogPINCP-log(PINCP,base=10))) +   
 geom\_point(alpha=.2, color="black") +  
 geom\_smooth(aes(x=predLogPINCP,  
 y=predLogPINCP-log(PINCP,base=10)),  
 color="black")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



rsq <- function(y,f) {1 - sum((y-f)^2)/sum((y-mean(y))^2) }  
rsq(log(dtrain$PINCP, base=10), predict(model, newdata=dtrain))

## [1] 0.3382568

rsq(log(dtest$PINCP, base=10), predict(model, newdata=dtest))

## [1] 0.2605496

rmse<- function(y,f) {sqrt(mean( (y-f)^2 )) }  
rmse(log(dtrain$PINCP, base=10), predict(model, newdata=dtrain))

## [1] 0.2651856

rmse(log(dtest$PINCP, base=10), predict(model, newdata=dtest))

## [1] 0.2752171

summary(log(dtrain$PINCP, base=10) - predict (model, newdata=dtrain))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.29220 -0.14153 0.02458 0.00000 0.17632 0.62532

summary(log(dtest$PINCP, base=10) - predict (model, newdata=dtest))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.494246 -0.165262 0.018922 -0.004636 0.175471 0.868075

load("C:/Users/turne/Downloads/NatalRiskData(1).RData") #Substitute your own location  
train <- sdata[sdata$ORIGRANDGROUP<= 5, ]  
test <- sdata[sdata$ORIGRANDGROUP>5, ]

complications <- c("ULD\_MECO","ULD\_PRECIP","ULD\_BREECH")  
riskfactors <- c("URF\_DIAB","URF\_CHYPER","URF\_PHYPER","URF\_ECLAM")  
y <- "atRisk"  
x <- c("PWGT",  
 "UPREVIS",  
 "CIG\_REC",  
 "GESTREC3",  
 "DPLURAL",  
 complications,  
 riskfactors)  
fmla<- paste(y, paste(x, collapse="+"), sep="~")

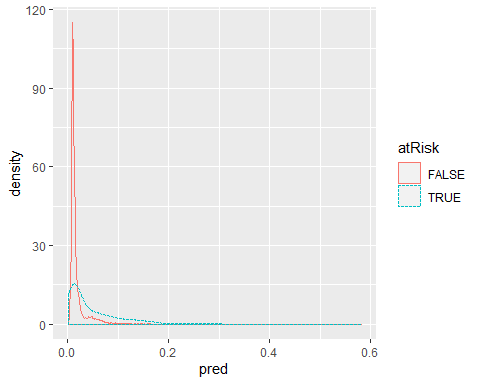
print(fmla)

## [1] "atRisk~PWGT+UPREVIS+CIG\_REC+GESTREC3+DPLURAL+ULD\_MECO+ULD\_PRECIP+ULD\_BREECH+URF\_DIAB+URF\_CHYPER+URF\_PHYPER+URF\_ECLAM"

model <- glm(fmla, data=train, family=binomial(link="logit"))

train$pred <- predict(model, newdata=train, type="response")  
test$pred <- predict(model, newdata=test, type="response")

library(ggplot2)  
ggplot(train, aes(x=pred, color=atRisk, linetype=atRisk)) + geom\_density()



library(ROCR)

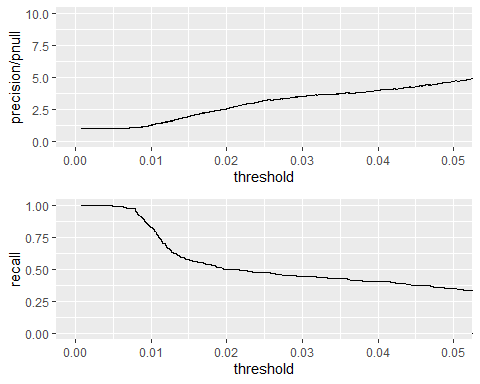
## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

library(grid)  
predObj <- prediction(train$pred, train$atRisk)  
precObj <- performance(predObj, measure = 'prec')  
recObj <- performance(predObj, measure='rec')  
  
precision <- (precObj@y.values)[[1]]  
prec.x <- (precObj@x.values)[[1]]  
recall <- (recObj@y.values)[[1]]  
  
rocFrame <- data.frame(threshold=prec.x, precision=precision, recall=recall)  
  
nplot <- function(plist) {  
 n <- length(plist)  
 grid.newpage()  
 pushViewport(viewport(layout=grid.layout(n,1)))  
 vplayout= function(x,y) {viewport(layout.pos.row=x, layout.pos.col=y)}  
 for (i in 1:n) {  
 print(plist[[i]], vp=vplayout(i,1))  
 }  
}  
   
pnull<- mean(as.numeric(train$atRisk))  
  
p1 <- ggplot(rocFrame, aes(x=threshold)) +  
 geom\_line(aes(y=precision/pnull)) +  
 coord\_cartesian(xlim = c(0,.05), ylim = c(0,10) )  
   
p2 <- ggplot(rocFrame, aes(x=threshold)) +  
 geom\_line(aes(y=recall)) +   
 coord\_cartesian(xlim = c(0,.05) )  
  
nplot(list(p1,p2))

## Warning: Removed 1 rows containing missing values (geom\_path).



ctab.test <- table(pred=test$pred>0.02, atRisk= test$atRisk)  
ctab.test

## atRisk  
## pred FALSE TRUE  
## FALSE 9487 93  
## TRUE 2405 116

precision <- ctab.test[2,2]/sum(ctab.test[2,])  
precision

## [1] 0.04601349

recall <- ctab.test[2,2]/sum(ctab.test[,2])  
recall

## [1] 0.5550239

enrich<- precision/mean(as.numeric(test$atRisk))  
enrich

## [1] 2.664159

coefficients(model)

## (Intercept) PWGT UPREVIS   
## -4.41218940 0.00376166 -0.06328943   
## CIG\_RECTRUE GESTREC3< 37 weeks DPLURALtriplet or higher   
## 0.31316930 1.54518311 1.39419294   
## DPLURALtwin ULD\_MECOTRUE ULD\_PRECIPTRUE   
## 0.31231871 0.81842627 0.19172008   
## ULD\_BREECHTRUE URF\_DIABTRUE URF\_CHYPERTRUE   
## 0.74923672 -0.34646672 0.56002503   
## URF\_PHYPERTRUE URF\_ECLAMTRUE   
## 0.16159872 0.49806435

summary(model)

##   
## Call:  
## glm(formula = fmla, family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.9732 -0.1818 -0.1511 -0.1358 3.2641   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.412189 0.289352 -15.249 < 2e-16 \*\*\*  
## PWGT 0.003762 0.001487 2.530 0.011417 \*   
## UPREVIS -0.063289 0.015252 -4.150 3.33e-05 \*\*\*  
## CIG\_RECTRUE 0.313169 0.187230 1.673 0.094398 .   
## GESTREC3< 37 weeks 1.545183 0.140795 10.975 < 2e-16 \*\*\*  
## DPLURALtriplet or higher 1.394193 0.498866 2.795 0.005194 \*\*   
## DPLURALtwin 0.312319 0.241088 1.295 0.195163   
## ULD\_MECOTRUE 0.818426 0.235798 3.471 0.000519 \*\*\*  
## ULD\_PRECIPTRUE 0.191720 0.357680 0.536 0.591951   
## ULD\_BREECHTRUE 0.749237 0.178129 4.206 2.60e-05 \*\*\*  
## URF\_DIABTRUE -0.346467 0.287514 -1.205 0.228187   
## URF\_CHYPERTRUE 0.560025 0.389678 1.437 0.150676   
## URF\_PHYPERTRUE 0.161599 0.250003 0.646 0.518029   
## URF\_ECLAMTRUE 0.498064 0.776948 0.641 0.521489   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2698.7 on 14211 degrees of freedom  
## Residual deviance: 2463.0 on 14198 degrees of freedom  
## AIC: 2491  
##   
## Number of Fisher Scoring iterations: 7

pred <- predict(model, newdata = train, type = 'response')  
 llcomponents <- function(y, py) {  
 y\*log(py) + (1-y)\*log(1-py)  
}  
  
edev <- sign(as.numeric(train$atRisk) - pred) \*  
 sqrt(-2\*llcomponents(as.numeric(train$atRisk), pred))  
  
summary(edev)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.9732 -0.1818 -0.1511 -0.1244 -0.1358 3.2641

loglikelihood <- function(y,py) {  
 sum(y \* log(py) + (1-y)\*log(1-py))  
}  
  
pnull<- mean(as.numeric(train$atRisk))  
  
null.dev<- -2\*loglikelihood(as.numeric(train$atRisk), pnull)  
pnull

## [1] 0.01920912

null.dev

## [1] 2698.716

model$null.deviance

## [1] 2698.716

pred<- predict(model, newdata=train, type="response")  
resid.dev<- -2\*loglikelihood(as.numeric(train$atRisk), pred)  
resid.dev

## [1] 2462.992

model$deviance

## [1] 2462.992

testy<- as.numeric(test$atRisk)  
testpred<- predict(model, newdata=test, type = "response")  
  
pnull.test<- mean(testy)  
  
null.dev.test<- -2\*loglikelihood(testy, pnull.test)  
  
resid.dev.test<- -2\*loglikelihood(testy, testpred)  
  
pnull.test

## [1] 0.0172713

null.dev.test

## [1] 2110.91

resid.dev.test

## [1] 1947.094

df.null <- dim(train)[[1]] -1  
df.model <- dim(train)[[1]] - length(model$coefficients)  
  
df.null

## [1] 14211

df.model

## [1] 14198

delDev<- null.dev - resid.dev  
deldf<- df.null - df.model  
p<- pchisq(delDev, deldf, lower.tail = F)  
  
delDev

## [1] 235.724

deldf

## [1] 13

p

## [1] 5.84896e-43

pr2<- 1-(resid.dev/null.dev)  
  
print(pr2)

## [1] 0.08734674

pr2.test<- 1-(resid.dev.test/null.dev.test)  
print(pr2.test)

## [1] 0.07760427

aic <- 2\*(length(model$coefficients) -  
 loglikelihood(as.numeric(train$atRisk), pred))  
aic

## [1] 2490.992