#330 Final Project

formule1<-as.formula("re\_30 ~ age + female + nchronic + rehabtransfer + tmpm + totchg +

cm\_aids + cm\_alcohol + cm\_anemdef + cm\_arth + cm\_bldloss +

cm\_chf + cm\_chrnlung + cm\_coag + cm\_depress + cm\_dm + cm\_dmcx +

cm\_drug + cm\_htn\_c + cm\_hypothy + cm\_liver + cm\_lymph + cm\_lytes +

cm\_mets + cm\_neuro + cm\_obese + cm\_para + cm\_perivasc + cm\_psych +

cm\_pulmcirc + cm\_renlfail + cm\_tumor + cm\_ulcer + cm\_valve +

cm\_wghtloss + dx003811 + dx00389 + dx004100 + dx004102 +

dx00413 + dx004185 + dx00539 + dx007054 + dx007070 + dx007071 +

dx01550 + dx01579 + dx01624 + dx01629 + dx01809 + dx01919 +

dx020300 + dx020410 + dx020500 + dx023770 + dx024901 + dx025000 +

dx025001 + dx025002 + dx025013 + dx025042 + dx025072 + dx025080 +

dx025082 + dx025083 + dx02724 + dx02761 + dx027801 + dx027949 +

dx02809 + dx028260 + dx028419 + dx02849 + dx02851 + dx02853 +

dx02859 + dx02863 + dx02866 + dx028800 + dx028983 + dx029181 +

dx029530 + dx029532 + dx029570 + dx029574 + dx029580 + dx029590 +

dx029620 + dx029650 + dx02967 + dx029680 + dx029689 + dx02989 +

dx030000 + dx030151 + dx030300 + dx030301 + dx030390 + dx030391 +

dx030393 + dx030400 + dx030401 + dx030410 + dx030470 + dx030471 +

dx030480 + dx030490 + dx030501 + dx030550 + dx030560 + dx030590 +

dx030593 + dx03083 + dx030981 + dx0311 + dx03159 + dx033829 +

dx03384 + dx034590 + dx034839 + dx03484 + dx03569 + dx035782 +

dx036589 + dx036960 + dx037000 + dx04010 + dx04019 + dx040300 +

dx040391 + dx041400 + dx041401 + dx041519 + dx042090 + dx04241 +

dx042731 + dx04280 + dx042823 + dx043814 + dx044421 + dx04476 +

dx04581 + dx045821 + dx04589 + dx04829 + dx049121 + dx049322 +

dx0496 + dx05070 + dx05119 + dx05363 + dx056729 + dx056789 +

dx05679 + dx05680 + dx056962 + dx056969 + dx05712 + dx05715 +

dx05718 + dx05771 + dx05793 + dx05849 + dx05852 + dx05855 +

dx05856 + dx0591 + dx06146 + dx07038 + dx07100 + dx071915 +

dx071941 + dx072402 + dx073300 + dx073313 + dx073390 + dx07580 +

dx078039 + dx078093 + dx07812 + dx07823 + dx07824 + dx078321 +

dx07837 + dx078559 + dx078650 + dx078720 + dx078839 + dx078959 +

dx07907 + dx07960 + dx079902 + dx08052 + dx080606 + dx08072 +

dx081200 + dx081201 + dx081209 + dx082020 + dx082300 + dx082302 +

dx082312 + dx082322 + dx082382 + dx082392 + dx08241 + dx08244 +

dx08246 + dx08248 + dx08249 + dx08250 + dx08251 + dx082523 +

dx082525 + dx082535 + dx083104 + dx08500 + dx085011 + dx08505 +

dx08509 + dx085182 + dx085202 + dx085220 + dx085221 + dx085223 +

dx085226 + dx08600 + dx08602 + dx086320 + dx086355 + dx086404 +

dx086501 + dx086503 + dx08670 + dx086802 + dx08750 + dx08921 +

dx08971 + dx09013 + dx09042 + dx090441 + dx09047 + dx09100 +

dx09120 + dx09130 + dx092820 + dx09348 + dx09552 + dx099529 +

dx099591 + dx099652 + dx099679 + dx099832 + dx099932 + dx0V0254 +

dx0V0481 + dx0V065 + dx0V1011 + dx0V1043 + dx0V110 + dx0V111 +

dx0V1204 + dx0V1251 + dx0V1254 + dx0V1255 + dx0V148 + dx0V1553 +

dx0V1581 + dx0V4321 + dx0V433 + dx0V4361 + dx0V440 + dx0V443 +

dx0V4512 + dx0V462 + dx0V463 + dx0V4972 + dx0V5419 + dx0V5865 +

dx0V600 + dx0V6284 + dx0V641 + dx0V642 + dx0V643 + dx0V6549 +

dx0V850 + dx0V8524 + dx0V9081 + pr00109 + pr00206 + pr00353 +

pr00379 + pr0043 + pr00481 + pr01641 + pr01682 + pr02103 +

pr02171 + pr02309 + pr02757 + pr03327 + pr03491 + pr03523 +

pr03845 + pr0387 + pr03889 + pr03893 + pr03895 + pr03897 +

pr04543 + pr04573 + pr04575 + pr04610 + pr04639 + pr04675 +

pr04821 + pr05252 + pr05384 + pr0540 + pr05492 + pr05581 +

pr05717 + pr06149 + pr07672 + pr07675 + pr07786 + pr07807 +

pr07809 + pr07812 + pr07815 + pr07816 + pr07817 + pr07818 +

pr07819 + pr07855 + pr07901 + pr07906 + pr07915 + pr07916 +

pr07918 + pr07926 + pr07931 + pr07932 + pr07935 + pr07936 +

pr07962 + pr07966 + pr07976 + pr07978 + pr08147 + pr08152 +

pr08184 + pr08191 + pr08345 + pr08388 + pr08401 + pr08415 +

pr08417 + pr08471 + pr08472 + pr08628 + pr08659 + pr08665 +

pr08667 + pr08669 + pr08686 + pr08721 + pr08853 + pr08872 +

pr08949 + pr08968 + pr09353 + pr09354 + pr09356 + pr09357 +

pr09462 + pr09607 + pr09788 + pr09904 + pr09905 + zip\_1 +

zip\_2 + zip\_3 + dispu\_AMA + dispu\_HomeHeal ")

library(readr)

re30fin <- read\_csv("~/re30fin.csv")

install.packages("MCMCpack")

library("MCMCpack")

#level1 model

posterior.level1<-MCMClogit(formula = formu, b0=0, B0=.001,data=re30fin)

summary.level1<-summary(posterior.level1)

coefficient.level1<-summary.level1$statistics[,1]

coefficient.level1<-as.matrix(coefficient.level1)

x.new<-matrix(1,nrow=166285,ncol=1)

x.level1<-as.matrix(cbind(x.new,re30fin[,3:387]))

pre.level1<- x.level1 %\*% coefficient.level1

prob.level1<-1/(1+exp(-pre.level1))

re30fin$prob.level1<-prob.level1

sum(re30fin$re\_30==1)

hist(prob.level1)

#posterior.level2<- brm(formula =

# time | cens(censored) ~ age \* sex + disease + (1 + age|patient),

# data = kidney, family = lognormal(),

# prior = c(set\_prior("normal(0,5)", class = "b"),

# set\_prior("cauchy(0,2)", class = "sd"),

# set\_prior("lkj(2)", class = "cor")), warmup = 1000,

# iter = 5000, chains = 4, control = list(adapt\_delta = 0.95))

#level2

install.packages("brms")

library("brms")

colnames(re30fin)[388]<-"HOSPID"

colnames(re30fin)[389]<-"Hvolume"

colnames(re30fin)[390]<-"Hbedsize1"

colnames(re30fin)[391]<-"Hbedsize3"

summary(re30fin$HOSPID)

level2<-brm(re\_30 ~ prob.level1 + HOSPID , data = re30fin)

#prior=get\_prior(HOSPID~Hvolume+Hbedsize1+Hbedsize3,data=re30fin,family=poisson)

#nlform <- bf(re\_30 ~ prob.level1 + HOSPID, HOSPID ~ Hvolume + Hbedsize1 + Hbedsize3, nl = TRUE)

#nlprior <- c(prior(normal(3000, 570), nlpar = "HOSPID"),

# prior(normal(0, 1), nlpar = "Hvolume"),

# prior(normal(0, 1), nlpar = "Hbedsize11"),

# prior(normal(0, 1), nlpar = "Hbedsize3"))

#fit\_level2 <- brm(formula = nlform, data = re30fin, family = gaussian(),

prior = nlprior)

write.csv(re30fin,"forstata.csv")

library(dplyr)

#sum(prob.level1)

#colnames(re30fin)

library(ggplot2)

#ggplot(anore30,aes(y=prob.level1,x=HOSPID,group=HOSPID),width=0.01)+ geom\_boxplot(aes(fill=HOSPID))

#boxplot(anore30$prob.level1~anore30$HOSPID)

#count(re30fin$HOSPID==3178)

#tot <- re30fin %>%

# group\_by(HOSPID) %>%

# summarize(renum = sum(prob.level1), total = sum(x1)) %>%

# mutate(average = renum /total )

#tot<-summarise(group\_by(re30fin,HOSPID),aver=mean(prob.level1,na.rm=T),LCI=quantile(prob.level1,0.05),UCI=quantile(prob.level1,0.95))

#tot

#y.mean<-sum(re30fin$re\_30)/166285

#y.mean

#tot$HOSPID[tot$UCI<y.mean]

#tot$HOSPID[tot$LCI>y.mean]

#plot(tot$aver~tot$HOSPID)

#anore30<-arrange(re30fin,HOSPID)

#pre.level2<-level2predict

#hist(as.matrix(pre.level2))

#re30fin$prob.level2<-pre.level2$E

#re30fin$patient<-1

tot$oi<-tot$re30

#disty<-MASS::fitdistr(pre.level2$E,rgamma,start=list(shape=1,rate=0.01))

#disty

#alpha0<-1

#rate0<-mean(re30fin$re\_30)

#re30fin$prior.gamma<-rgamma(1,alpha0,rate0)

sum(re30fin$prob.logit1)

tot$ode<-tot$pred/tot$oi

tot$ode[tot$ode==Inf]<-1

hist(tot$ode)

mean(tot$ode)

sd(tot$ode)

shapiro.test(tot$ode)

#logit model

model <- glm(formula=formule1,family=binomial(link='logit'),data=re30fin)

coefficient.logit1<-summary(model)

coefficient.logit1<-as.matrix(coefficient.logit1$coefficients[,1])

pre.logit1<- x.level1 %\*% coefficient.logit1

re30fin$prob.logit1<-1/(1+exp(-pre.logit1))

re30fin$prob.logit1

library(dplyr)

tot<-summarise(group\_by(re30fin,HOSPID),ei=sum(prob.logit1,na.rm=T),oi=sum(re\_30),total=sum(patient),LCI=quantile(prob.level1,0.05),UCI=quantile(prob.level1,0.95),std=sd(re\_30))

tot$p<-tot$oi/tot$total

tot$pre.p<-tot$ei/tot$total

tot$ode<-tot$oi/tot$ei

tot$oe<-tot$oi/tot$ei

hist(tot$ode,main="Histogram of O/E ratio for logit model",xlab="O/E ratio")

OE<-mean(tot$oe)

sd(tot$pre.p)

quantile(tot$oe,c(0.025,0.975))

plot(tot$pre.p~tot$p,main="readimssion rate by logistics and data",xlab="data",ylab="prediction from logit")

abline(h=mean(thetaEB))

abline(0,1)

hist(tot$pre.p,main="distribution of predictions",xlab="predicted readimission")

hist(tot$pre.p)

#Empirical Bayes

alpha1<-60

rate1<-60

var1<-1/alpha1^2

thetaEB<-rgamma(637,sum(tot$oi)+alpha1,sum(tot$total)+rate1)

hist(tot$thetaEB1,main="Readmisson by Empirical Bayes",xlab="Estimated Readmission Rate")

mean(thetaEB)

thetadata<-rgamma(637,sum(tot$oi)+60,sum(tot$total)+60)

sd(thetadata)

tot$thetaEB1<-(tot$oi+sum(tot$oi))/(tot$total+sum(tot$total))

plot(tot$thetaEB1~tot$p,main="plot of ThetaEB and Theta from data",xlab="Readmission Rate",ylab="ThetaEB")

abline(h=mean(tot$thetaEB1))

abline(0,1)

ODEB1<-tot$p/tot$thetaEB1

quantile(ODEB1,c(0.025,0.975))

mean(ODEB1)

OEEB<-tot$p/thetaEB

mean(OEEB)

hist(OEEB,main="O/E ratio by Empirical Bayes")

plot(thetaEB~tot$p,main="plot of ThetaEB and Theta from data",xlab="Readmission Rate")

abline(h=mean(thetaEB))

abline(0,1)

#Non-parametric

npeb<-c()

lam<-tot$oi

lambdabar<-mean(lam)

npeb<-(lam+1)\*(ppois(lam+1,lambda=lam)/ppois(lam,lambda = lam))

hist(npeb,main="Nonparametrics Empirical Bayes for readmissions of Hospitals",xlab= "number of readmissions")

hist(npeb/tot$total,main="Histogram of readminssion rate by NPEB",xlab="Non parametric RE")

mean(tot$oi/npeb)

quantile(tot$oi/npeb,c(0.025,0.975))

hist(tot$oi/npeb,main="histogram of O/E ratio by Nonparametric Bayes",xlab="O/E ratio")

plot(npeb/tot$total~tot$p,main="plot of Nonparmetrics EB ThetaEB and Theta from data",xlab="Readmission Rate")

abline(h=mean(thetaEB))

abline(0,1)

#MCMC

install.packages("rjags")

library(rjags)

model\_string <- "model{

# Likelihood (can't have formulas in distribution functions)

Y ~ dpois(mu)

mu <- N\*lambda

# Prior

lambda ~ dgamma(a, b)

}"

model <- jags.model(textConnection(model\_string),

data = list(Y=Y,N=N,a=a,b=b))

update(model, 10000, progress.bar="none")

samp <- coda.samples(model,

variable.names=c("lambda"),

n.iter=20000, progress.bar="none")

summary(samp)

#James Stein Estimator

#Shrinkage

c<-seq(0,1,by=0.001)

p.bar<-mean(tot$p)

MSE<-c()

ssz<-c()

for (i in 1:length(c)){

z<-c()

z<-p.bar+c[i]\*(tot$p-p.bar)

ssz[i]<-(p.bar-z)^2+(p.bar-tot$p)^2

MSE[i]<-sd(z)

}

plot(ssz)

ssz

sd(tot$pre.p)

zt<-c()

sigma2<-var(tot$p)

p.bar<-mean(tot$p)

ct<-1-(tot$total-3)\*sigma2/sum((tot$p-p.bar)^2)

mean(ct)

zt<-p.bar+ct\*(tot$p-p.bar)

mean(zt)

hist(zt,main="Distribution of JS Estimators for Readmission Rate",xlab="JS Estimators")

OEJS<-mean(tot$p/zt)

hist(tot$p/zt,main="Histogram of O/E ratio by JS Estimators",xlab="O/E ratio")

var(zt)

sd(zt)

sd(tot$p/zt)

plot(zt~tot$p,main="James Stein Estimators and Readmission Rate",xlab="Readmission Rate",ylab="JS Estimators")

abline(h=mean(thetaEB))

abline(0,1)

#Hospital level

tot<-summarise(group\_by(re30fin,HOSPID),ei=sum(prob.logit1,na.rm=T),oi=sum(re\_30),total=sum(patient),LCI=quantile(prob.level1,0.05),UCI=quantile(prob.level1,0.95),std=sd(re\_30),re\_level2=sum(prob.level2))

tot$p<-tot$oi/tot$total

tot$pre.p<-tot$ei/tot$total

tot$pre.p2<-tot$re\_level2/tot$total

tot$ode2<-tot$pre.p2/tot$p

tot$ode2[tot$ode2==Inf]<-1

mean(tot$ode2,na.rm=T)

quantile(tot$ode2,c(0.025,0.975))

hist(tot$ode2,main="Histogram of O/E Ratio by Level 2 model",xlab="O/E Ratio")

hist(tot$pre.p2,main="histogram of Readmission rate from level 2 model ",xlab="Readmission Rate")

plot(tot$pre.p2~tot$p,main="Estimated Readmission by Level 2 and Readmission Rate",xlab="Readmission Rate",ylab="JS Estimators")

abline(h=mean(thetaEB))

abline(0,1)

=