The Impact of Variable Leakage Rates on OE Ratios a Logistic Model Simulation

A cohort of **51 hospitals** with 1,000 randomized cases each is generated with a dichotomous outcome variable **y = 30 day readmission status** and a single dichotomous **risk factor x**. Overall, the **prevalence of y is 20%** *(readmission rate)* and the **prevalence of x is 10%**. Varying proportions of each of hospital’s readmissions occur at non-index hospitals (referred to as) **Leakage Rates = 0% through 50%.** A Spearman rank (rho) **correlation between y and x = 0.45** is used to establish the strength of association of y with x in order to achieve an overall **ROC score of approx. 0.66** for a logistic model: *glm(y ~ x,family = binomial(link=‘logit’))*.

**The above is resampled 1000 times**

- Note: the x and y prevalences (p & q), leakage rates, # hospitals, sample sizes (n.sim) and rho can all be customized.

rm(list=ls())  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.2 v dplyr 1.0.8  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## Warning: package 'dplyr' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

##################################################################  
# Routine for generating sample random dataset for testing Logistic model  
# Generates two variables - y: response variable; x: risk factor  
# q: ~Prevalence of y (set by user)  
# p: ~Prevalence of x (set by user)  
# rho: Spearman rank correlation  
# used as strength of association of y with x, (set by user).  
# Higher the abs(rho) => higher ROC for glm(y ~ x,family = binomial(link='logit'))  
# Four hospital sets of 1,000 cases generated with varying leakage rates% (0,10,20,40)  
##################################################################  
rm(list=ls())  
library(tidyverse)  
library(reshape2)

## Warning: package 'reshape2' was built under R version 4.1.2

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

# This snippet generates the data, called once for each hospital  
# All hospitals: y prevalence (q) ~ 20%; x prevalence (p) ~10%, N=1000  
#-----------------------------------------  
  
# Compute Pr(0,0) from rho, p=Pr(X=1), and q=Pr(Y=1).  
a <- function(rho, p, q) {  
 a.0 <- rho \* sqrt(p\*q\*(1-p)\*(1-q)) + (1-p)\*(1-q)  
 prob <- c(`(0,0)`=a.0, `(1,0)`=1-q-a.0, `(0,1)`=1-p-a.0, `(1,1)`=a.0+p+q-1)  
 if (min(prob) < 0) {  
 print(prob)  
 stop("Error: a probability is negative.")  
 }  
 u <- sample.int(4, n.sim \* n, replace=TRUE, prob=prob)  
 return(u)  
}  
#-----------------------------------------  
iterations = 100  
staticEnctern = 100  
set.seed(21)  
hospitals <- 51  
hospid <- c(0:(hospitals-1))  
#hospLeakageRate <- seq(0,0.50,0.50/(hospitals-1)) # Readmissions % at non-index facilities (0 to 50%)  
hospEnctrs <- rep(staticEnctern,hospitals) # Each hosp with with 1000 cases, change if desired  
p <- 0.1 # Single risk factor 'x' prevalance   
q <- 0.3 # Overall readmission event rate   
rho <- 0.45; # Spearman rank (rho) correlation between y and x   
n.sim <- sum(hospEnctrs)  
n <- 1 # don't change  
  
  
for (i in 1:iterations) {  
 u <- a(rho, p, q) # Randomize  
 cohort <- data.frame(iteration=rep(i,n.sim),  
 hosp = rep(c(hospid),c(hospEnctrs)),  
 #lkg = rep(c(hospLeakageRate),c(hospEnctrs)),  
 lkg = rep(c(hospid/100),c(hospEnctrs)),  
 x = 1 - u %% 2,  
 y = floor((u-1)/2),  
 y\_ndxhosp = floor((u-1)/2)\*unlist(lapply(rep(c(1-hospid/100),c(hospEnctrs)), function(x) rbinom(n=1, size=1,prob=x)))  
 )  
 m1 <- glm(y ~ x , data = cohort, family = binomial(link='logit'))  
 #pROC::roc(response = cohort$y, predictor = fitted(m1))  
 #coef(summary(m1))[,'Pr(>|z|)']  
 m1$coefficients  
 b0=m1$coefficients[1]  
 b1=m1$coefficients[2]  
 cohort$expected <- exp(b0+b1\*cohort$x)/(1+exp(b0+b1\*cohort$x))  
 cohort$AUC <-0\*cohort$y + pROC::roc(response = cohort$y, predictor = fitted(m1))$auc[1]  
 # Initialize vectors  
 if(i==1){   
 smry <- cohort %>% group\_by(hosp, iteration) %>%  
 summarize(Obs\_rate=mean(y)  
 , Exp\_rate = mean(expected)  
 , Obs\_rate\_same = mean(y\_ndxhosp)  
 , OE\_Ratio = mean(y)/mean(expected)  
 , OE\_Ratio\_same = mean(y\_ndxhosp )/mean(expected)  
 , OE\_Ratio\_same\_adj = Obs\_rate\_same/ mean((1-lkg)\*expected)  
 , OE\_Ratio\_same\_adj\_delta = mean(y)/mean(expected) - Obs\_rate\_same/ mean((1-lkg)\*expected)  
 , Rate\_Ratio = (Obs\_rate\_same/ mean((1-lkg)\*expected)) / (mean(y)/mean(expected))  
 , AUC = mean(AUC)  
 )  
 }  
 if(i>1){  
 hold <- cohort %>% group\_by(hosp, iteration) %>%  
 summarize(Obs\_rate=mean(y)  
 , Exp\_rate = mean(expected)  
 , Obs\_rate\_same = mean(y\_ndxhosp)  
 , OE\_Ratio = mean(y)/mean(expected)  
 , OE\_Ratio\_same = mean(y\_ndxhosp )/mean(expected)  
 , OE\_Ratio\_same\_adj = Obs\_rate\_same/ mean((1-lkg)\*expected)  
 , OE\_Ratio\_same\_adj\_delta = mean(y)/mean(expected) - Obs\_rate\_same/ mean((1-lkg)\*expected)  
 , Rate\_Ratio = (Obs\_rate\_same/ mean((1-lkg)\*expected)) / (mean(y)/mean(expected))  
 , AUC = mean(AUC)  
 )  
 smry<-rbind(smry,hold)   
 }   
  
}

## Setting levels: control = 0, case = 1

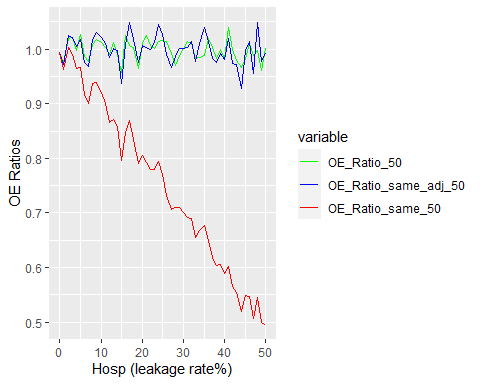
## Setting direction: controls < cases

## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.Setting levels: control = 0, case = 1  
## Setting direction: controls < cases  
## `summarise()` has grouped output by 'hosp'. You can override using the `.groups` argument.

final<-smry %>% group\_by(hosp) %>%  
summarize(cases = n()  
 , Obs\_rate\_avg =mean(Obs\_rate)  
 , Exp\_rate\_avg =mean(Exp\_rate)  
 , Obs\_rate\_same\_avg =mean(Obs\_rate\_same)  
 , OE\_Ratio\_avg =mean(OE\_Ratio)  
 , OE\_Ratio\_25 =quantile(OE\_Ratio, .27)  
 , OE\_Ratio\_50 =quantile(OE\_Ratio, .50)  
 , OE\_Ratio\_75 =quantile(OE\_Ratio, .75)  
 , OE\_Ratio\_same\_avg =mean(OE\_Ratio\_same)  
 , OE\_Ratio\_same\_50 =quantile(OE\_Ratio\_same, .50)  
 , OE\_Ratio\_same\_se =sd(OE\_Ratio\_same)/sqrt(cases)  
 , OE\_Ratio\_same\_adj\_avg =mean(OE\_Ratio\_same\_adj)  
 , OE\_Ratio\_same\_adj\_05 = quantile(OE\_Ratio\_same\_adj, .05)  
 , OE\_Ratio\_same\_adj\_25 = quantile(OE\_Ratio\_same\_adj, .25)   
 , OE\_Ratio\_same\_adj\_50 = quantile(OE\_Ratio\_same\_adj, .50)   
 , OE\_Ratio\_same\_adj\_75 = quantile(OE\_Ratio\_same\_adj, .75)   
 , OE\_Ratio\_same\_adj\_95 = quantile(OE\_Ratio\_same\_adj, .95)  
 , Rate\_Ratio\_05 = quantile(Rate\_Ratio, .05)  
 , Rate\_Ratio\_25 = quantile(Rate\_Ratio, .25)   
 , Rate\_Ratio\_50 = quantile(Rate\_Ratio, .50)   
 , Rate\_Ratio\_75 = quantile(Rate\_Ratio, .75)   
 , Rate\_Ratio\_95 = quantile(Rate\_Ratio, .95)   
 )

Plot the median OE Ratio of **51 hospitals** with 0% to 50% Lakage Rates, respctively.  
Randomaized as follows - 100 iteration, 51 Hospitals  
- 1,000 randomized cases per Hospital - Readmision overall prevalence: 30% each  
- Single dichotomous **risk factor x**  
- Logistic Model: Rho(y,x)=0.45 => Model ROC ~0.65

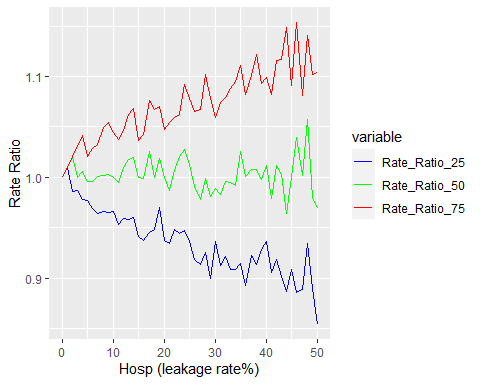
dd\_sub = final[,c("hosp","OE\_Ratio\_50","OE\_Ratio\_same\_adj\_50","OE\_Ratio\_same\_50")]  
dd = melt(dd\_sub, id=c("hosp"))  
ggplot(dd) + geom\_line(aes(x=hosp, y=value, colour=variable)) +  
 xlab("Hosp (leakage rate%)") + ylab("OE Ratios") +  
 scale\_colour\_manual(values=c("green","blue","red"))



dd\_sub = final[,c("hosp","Rate\_Ratio\_25","Rate\_Ratio\_50","Rate\_Ratio\_75")]  
dd = melt(dd\_sub, id=c("hosp"))

## Warning: attributes are not identical across measure variables; they will be  
## dropped

ggplot(dd) + geom\_line(aes(x=hosp, y=value, colour=variable))+   
 scale\_colour\_manual(values=c("blue","green","red")) +   
 xlab("Hosp (leakage rate%)") + ylab("Rate Ratio") +  
 geom\_line(aes(x=hosp, y=value, colour=variable))



AUCs <- smry %>% group\_by(iteration)%>%  
 summarize(AUC=mean(AUC))  
ggplot(data=AUCs,aes(x=AUC )) +  
 geom\_density()

