Homework 03

Inhwan Ko

November 16, 2021

Modeling Cy Young winners using logit

```
data <- read.csv("cyyoung.csv")

# Get nice colors
col <- brewer.pal(5, "Set1")
blue <- col[2]
orange <- col[5]
```

Answer a.

```
m1 <- cy ~ era + winpct
m1data <- extractdata(m1, data, na.rm=TRUE)

y <- data$cy
x1 <- m1data[,2:ncol(m1data)] %>% as.matrix()


# Likelihood function for logit
llk.logit <- function(param,y,x) {
  os <- rep(1,length(x[,1]))
  x <- cbind(os,x)
  b <- param[ 1 : ncol(x) ]
  xb <- x%*%b
  sum( y*log(1+exp(-xb)) + (1-y)*log(1+exp(xb)))
}

# Fit logit model using optim
ls.result <- lm(y~x1)  # use ls estimates as starting values
stval <- ls.result$coefficients  # initial guesses
logit.m1 <- optim(stval,llk.logit,method="BFGS",hessian=T,y=y,x=x1)
                  # call minimizer procedure
pe.m1 <- logit.m1$par   # point estimates
vc.m1 <- solve(logit.m1$hessian)  # var-cov matrix
se.m1 <- sqrt(diag(vc.m1))    # standard errors
ll.m1 <- -logit.m1$value  # likelihood at maximum

# Alternative estimation technique:  GLM
glm.m1 <- glm(m1, data=data, family="binomial")
```

```
glm.m1 %>%
  tidy() %>%
  mutate(estimate.glm = estimate,
         estimate.optim = pe.m1,
         std.error.glm = std.error,
         std.error.optim = se.m1) %>%
  select(-estimate, -std.error, -statistic, -p.value) %>%
  pander::pander()
```

term	estimate.glm	estimate.optim	std.error.glm	std.error.optim
(Intercept)	1.342	1.342	3.289	3.29
era	-2.11	-2.112	0.5126	0.513
winpct	6.171	6.18	3.91	3.912

```
tibble(logll.glm = logLik(glm.m1),
         logll.optim = ll.m1) %>%
  pander::pander()
```

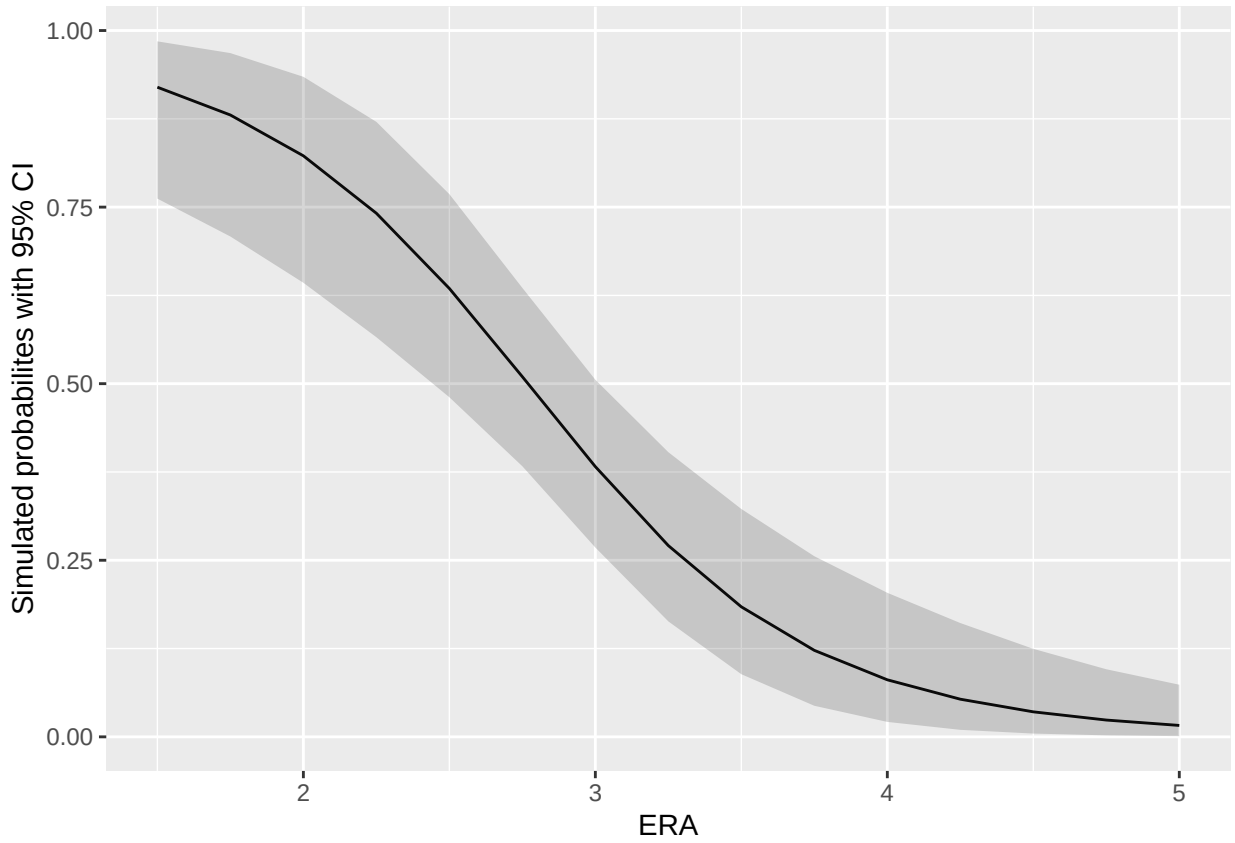
logll.glm	logll.optim
-46.22	-46.22

Answer b.:

```
x <- tibble(os = rep(1, 15),
            era = seq(1.5, 5, 0.25),
            winpct = rep(data$winpct %>% mean(), 15))

simbeta <- mvrnorm(10000, pe.m1, vc.m1)

xb <- x %>% as.matrix() %*% t(simbeta)

inverse.logit <- function(xb){
1 / (1 + exp (- xb))}

pi <- inverse.logit(xb)
dim(pi)
```

```
## [1]    15 10000
```

```
tibble(era=seq(1.5, 5, 0.25),
       pe=rowMeans(pi),
       lower=matrixStats::rowQuantiles(pi, probs=0.025),
       upper=matrixStats::rowQuantiles(pi, probs=0.975)) %>%
  ggplot(aes(x=era, y=pe, ymax=upper, ymin=lower)) +
  geom_line() +
  geom_ribbon(alpha=0.2, linetype=0) +
  labs(y="Simulated probabilites with 95% CI", x="ERA")
```



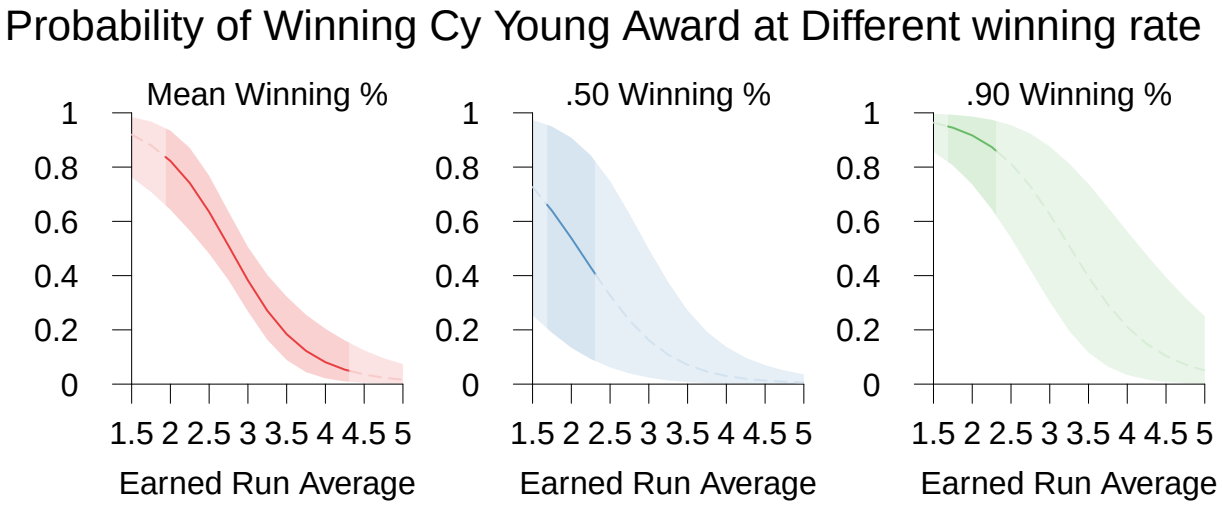
Answer c.:

```
sims <- 10000

xhyp <- seq(1.5, 5, 0.25)
nscen <- length(xhyp)

winpct.base <-
winpct.0.5  <-
winpct.0.9  <-
  cfMake(m1, m1data, nscen)


for (i in 1:nscen) {
  winpct.base <- cfChange(winpct.base, "era", x = xhyp[i], scen = i)
  winpct.0.5  <- cfChange(winpct.0.5,  "era", x = xhyp[i], scen = i)
  winpct.0.9  <- cfChange(winpct.0.9,  "era", x = xhyp[i], scen = i)

  winpct.0.5  <- cfChange(winpct.0.5, "winpct", x=0.5, scen=i)
  winpct.0.9  <- cfChange(winpct.0.9, "winpct", x=0.9, scen=i)
}


sims.base <- logitsimev(winpct.base, simbeta, ci=0.95)
```

```
sims.0.5  <- logitsimev(winpct.0.5,  simbeta, ci=0.95)
sims.0.9  <- logitsimev(winpct.0.9,  simbeta, ci=0.95)
```

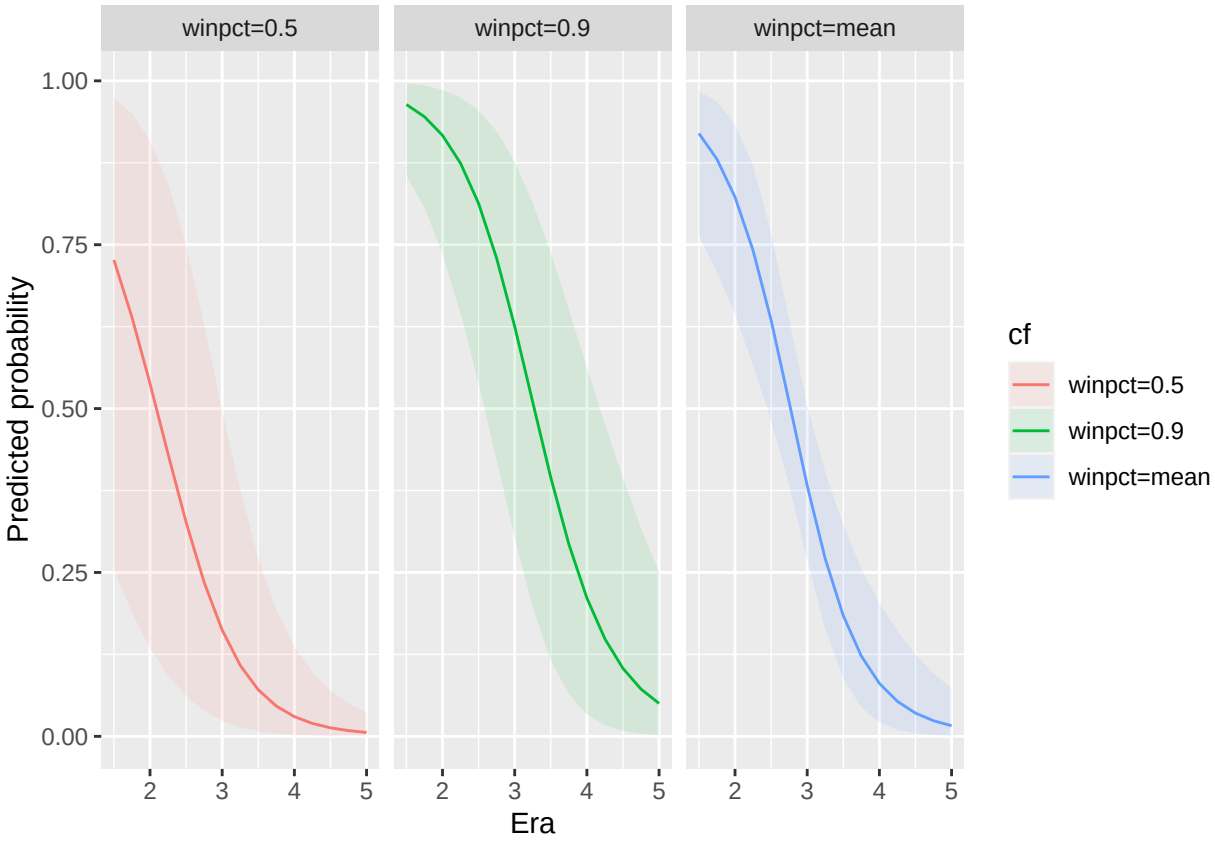
Plot by `tile`

```
trace.fun <- function(sim.result, coln, cfmake, plt){
 lineplot(x=xhyp,
          y=sim.result$pe,
          lower=sim.result$lower,
          upper=sim.result$upper,
          col=col[coln],
          extrapolate=list
            (data=m1data[,2:ncol(m1data)],
             cfact=cfmake$x[,2:ncol(cfmake$x)],
             omit.extrapolated=FALSE),
          plot=plt)
}


trace.base <-
  trace.fun(sim.result = sims.base,
            coln = 1,
            cfmake = winpct.base,
            plt = 1)

trace.0.5 <-
  trace.fun(sim.result = sims.0.5,
            coln = 2,
            cfmake = winpct.0.9,
            plt = 2)

trace.0.9 <-
  trace.fun(sim.result = sims.0.9,
            coln = 3,
            cfmake = winpct.0.9,
            plt = 3)


# Plot traces using tile
tile(trace.base,
             trace.0.5,
             trace.0.9,
             xC = c(1,3),
             limits=matrix(c(1.50,5.00,0,1,1.50,5.00,0,1,1.50,5.00,0,1),
                           nrow=3,
                           ncol=4, byrow=TRUE),
             xaxistitle=list(labels=c("Earned Run Average",
                                      "Earned Run Average",
                                      "Earned Run Average")),
             columntitle=list(labels=c("Mean Winning %",".50 Winning %", ".90 Winning %")),
             maintitle=list(labels="Probability of Winning Cy Young Award at Different winning rate"))
```



Plot by ggplot

```
df.c <- tibble(pe = c(sims.base$pe, sims.0.5$pe, sims.0.9$pe),
               lower = c(sims.base$lower, sims.0.5$lower, sims.0.9$lower),
               upper = c(sims.base$upper, sims.0.5$upper, sims.0.9$upper),
               cf = c(rep("winpct=mean",15),rep("winpct=0.5",15),rep("winpct=0.9",15)),
               era = rep(xhyp,3))

df.c %>%
  ggplot(aes(x = era, y = pe, colour = cf))+
  geom_line() +
  geom_ribbon(aes(ymin = lower, ymax = upper, fill = cf), linetype = 0, alpha = 0.1)+
  labs(x = "Era",
       y = "Predicted probability") +
  facet_wrap(~cf)
```



The plots shows that the lower `era`, earnd run aveages is more likely to win Cy young, and the higher winning rates prone to be awarded Cy Young. Because of sample size, `winpct = 0.5/0.9` have lagrge the 95% CI.(Shadow line may suggest lack of data).

Answer d.:

I assume strikeout to walk ratio (K/BB) is an important factor of pitcher. Also, the more a team wins, the player in the team is likely to win the Cy Young award. Also, it would be better to control different league.

```
data <-
  data %>%
  mutate(kbb = strikeout/walks)

m2 <- cy ~ era + kbb + twinpct + natleag + playoffs
m2data <- extractdata(m2, data, na.rm=TRUE)


x2 <- m2data[,2:ncol(m2data)] %>% as.matrix()

# Fit logit model using optim
ls.result <- lm(y~x2)
stval2 <- ls.result$coefficients
logit.m2 <- optim(stval2,llk.logit,method="BFGS",hessian=T,y=y,x=x2)

pe.m2 <- logit.m2$par    # point estimates
```

```
vc.m2 <- solve(logit.m2$hessian)  # var-cov matrix
se.m2 <- sqrt(diag(vc.m2))    # standard errors
ll.m2 <- -logit.m2$value  # likelihood at maximum


glm.m2 <- glm(m2, data=data, family="binomial")
```

(i): A likelihood ratio test

```
k.m1 <- length(pe.m1)
k.m2 <- length(pe.m2)

lr.test <- 2 * (ll.m2 - ll.m1)
lr.test.p <- pchisq(lr.test, df = (k.m2 - k.m1), lower.tail=FALSE)
```

The likelihood ratio is 0.74.

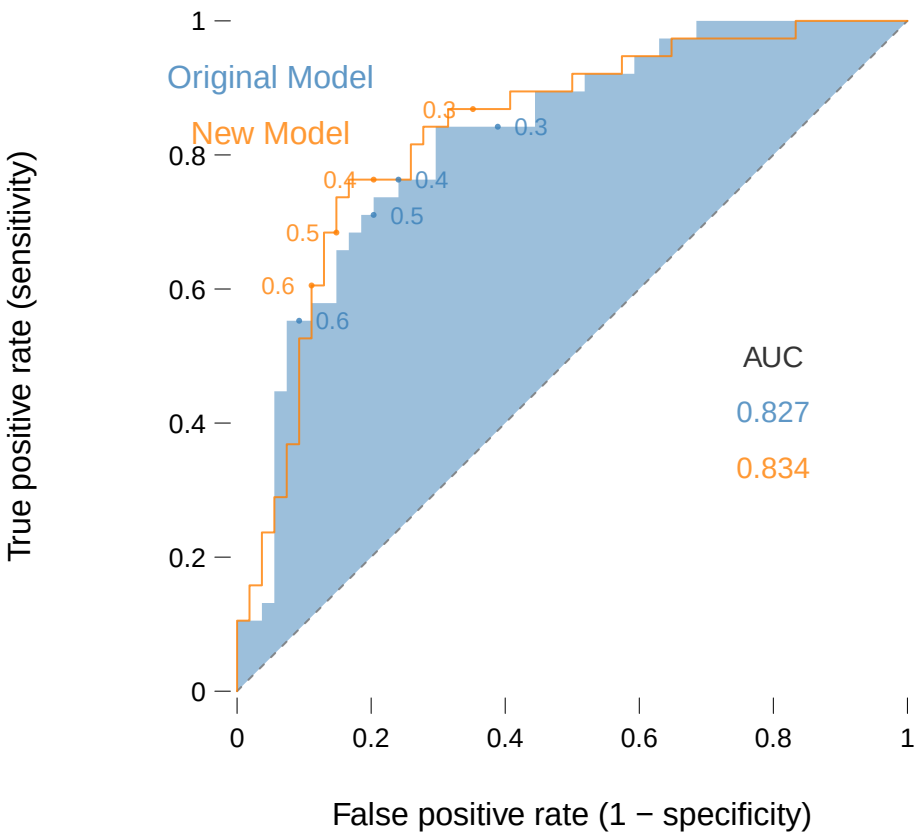
(ii): BIC and AIC

```
bic.m1 <- log(nrow(m1data))*k.m1 - 2*ll.m1
bic.m2 <- log(nrow(m2data))*k.m2 - 2*ll.m2
bic.test <- bic.m2 - bic.m1

aic.m1 <- 2*k.m1 - 2*ll.m1
aic.m2 <- 2*k.m2 - 2*ll.m2
aic.test <- aic.m2 - aic.m1
```

The BIC test result is 12.33 and the AIC test result is 4.76. Both results are positive, suggesting the original model is the better.

(iii): in-sample ROC curves

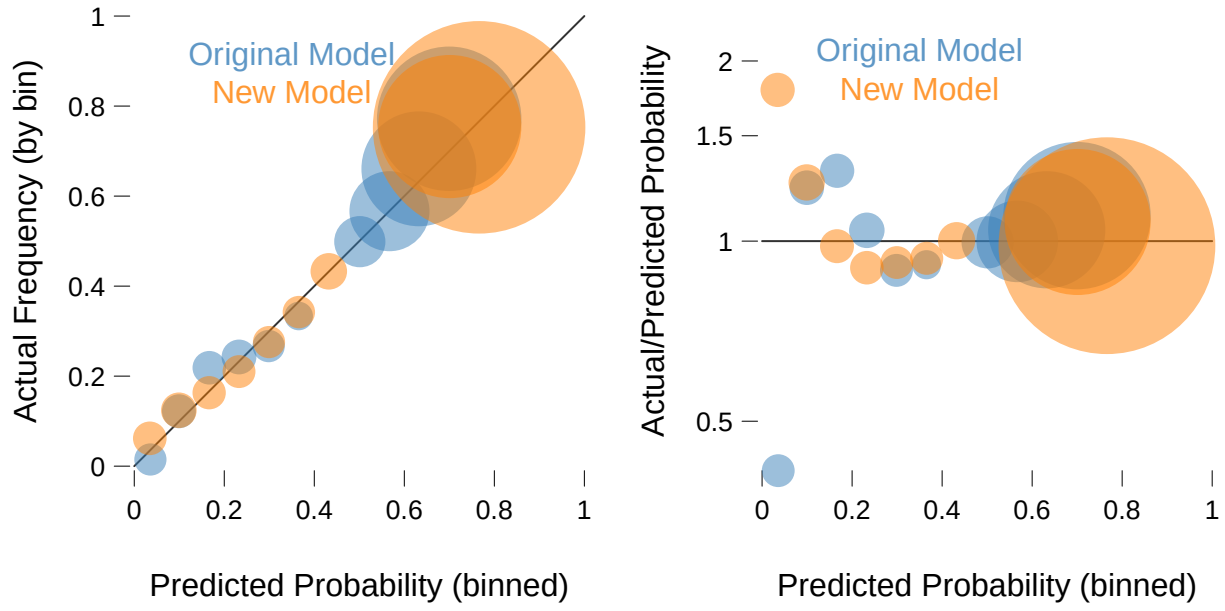
```
binned.m1 <- binPredict(glm.m1, col = blue, bins = 15, label = "Original Model", quantiles=F)
binned.m2 <- binPredict(glm.m2, col = orange, bins = 15, label = "New Model", quantiles=F)

plot(binned.m1, binned.m2, display = "roc",
thresholds = c(0.6, 0.5, 0.4, 0.3),
labx = 0.05)
```



AUC of new model is higher than original model, suggesting the new model has a better fitness.

(iv): insample Actual versus Predicted plots

```
plot(binned.m1, binned.m2, display=c("avp", "evp"), hide=TRUE, labx=0.35,
     totalarea=0.01)
```

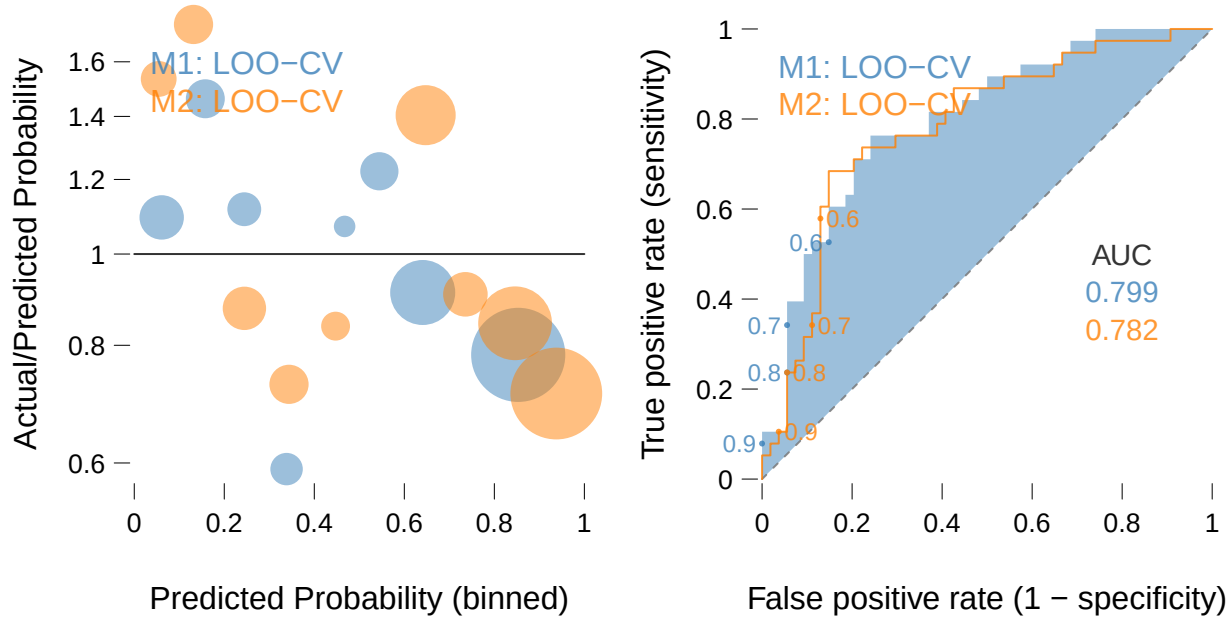


It is hard to indentify the differences between two models.

(v): cross-validation

```
loocv <- function (obj) {
    data <- obj$data
    m <- dim(data)[1]
    form <- formula(obj)
    fam <- obj$family$family
    loo <- rep(NA, m)
    for (i in 1:m) {
        i.glm <- glm(form, data = data[-i, ], family = fam)
        loo[i] <- predict(i.glm, newdata = data[i,], family = fam, type = "response")
    }
    loo
}

predCVm1 <- loocv(glm.m1)
predCVm2 <- loocv(glm.m2)

# Make cross-validated AVP and ROC plots; note use of newpred input in binPredict
binnedM1cv <- binPredict(glm.m1, newpred=predCVm1, col=blue, bins = 10,
                         label="M1: LOO-CV", quantiles=F)
```

```
binnedM2cv <- binPredict(glm.m2, newpred=predCVm2, col=orange, bins = 10,
                         label="M2: LOO-CV", quantiles=F)

plot(binnedM1cv, binnedM2cv, display=c("evp","roc"), hide=TRUE, thresholds=c(0.9, 0.8, 0.7, 0.6),labx=0
```



These plots also suggest the original model is the better.

Answer e.:

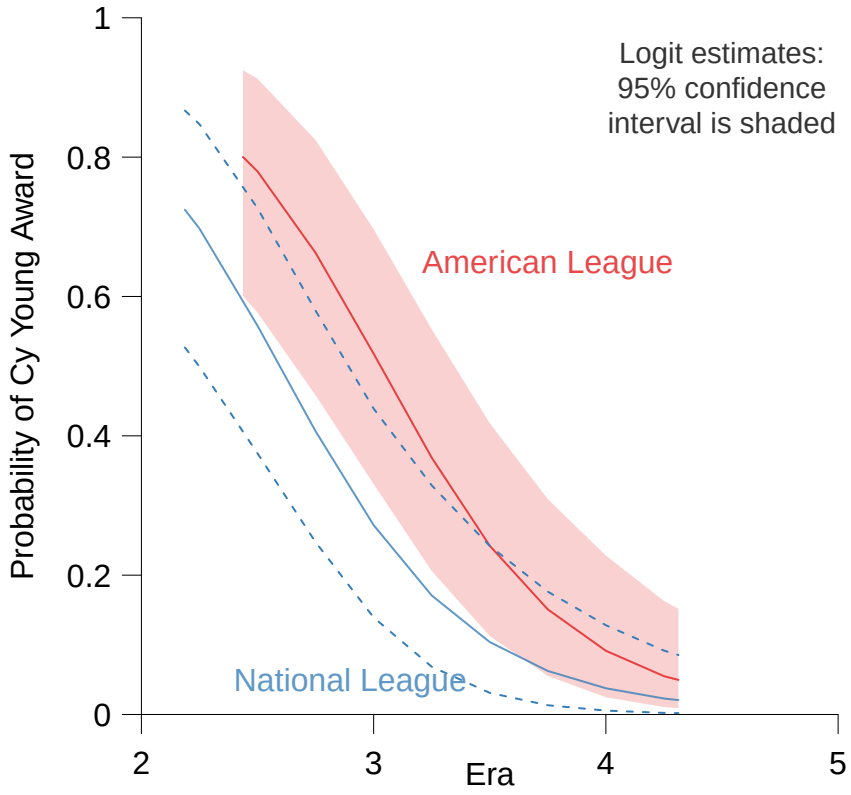
Cy Young is awarded to one players from each league. I am interested that how the leagues differenciate the other factors such as `era`.

```
simbeta.m2 <- mvrnorm(sims, pe.m2, vc.m2)
xhyp <- seq(1.5, 5, 0.25)
nscen <- length(xhyp)

aleague.sim <-
nleague.sim <-
  cfMake(m2, m2data, nscen)

for (i in 1:nscen) {

  aleague.sim <- cfChange(aleague.sim, "era", x = xhyp[i], xpre= xhyp[i], scen = i)
  aleague.sim <- cfChange(aleague.sim, "natleag", x = 0,  scen = i)
```

```
  nleague.sim <- cfChange(nleague.sim, "era", x = xhyp[i], xpre= xhyp[i], scen = i)
  nleague.sim <- cfChange(nleague.sim, "natleag", x = 1,  xpre = 0, scen = i)
}

# Simulate expected probabilities for all scenarios
aleague.prob <- logitsimev(aleague.sim, simbeta.m2, ci=0.95)
nleague.prob <- logitsimev(nleague.sim, simbeta.m2, ci=0.95)


leagueFD <- logitsimfd(nleague.sim, simbeta.m2, ci=0.95)
leagueRR <- logitsimrr(nleague.sim, simbeta.m2, ci=0.95)
```

```
# Set up lineplot traces of expected probabilities
aleague.Trace <- lineplot(x=xhyp,
                          y=aleague.prob$pe,
                          lower=aleague.prob$lower,
                          upper=aleague.prob$upper,
                          col=col[1],
                          extrapolate=list(data=m2data[,2:ncol(m2data)],
                            cfact=aleague.sim$x[,2:ncol(aleague.sim$x)],
                            omit.extrapolated=TRUE),
                          plot=1)

nleague.Trace <- lineplot(x=xhyp,
                          y=nleague.prob$pe,
                          lower=nleague.prob$lower,
                          upper=nleague.prob$upper,
                          col=col[2],
                          ci = list(mark="dashed"),
                          extrapolate=list(data=m2data[,2:ncol(m2data)],
                            cfact=nleague.sim$x[,2:ncol(nleague.sim$x)],
                            omit.extrapolated=TRUE),
                          plot=1)


# Set up traces with labels and legend
labelTrace <- textTile(labels=c("American League", "National League"),
                       x=c( 3.75, 2.9),
                       y=c( 0.65, 0.05),
                       col=col,
                       plot=1)

legendTrace <- textTile(labels=c("Logit estimates:", "95% confidence", "interval is shaded"),
                        x=c(4.5, 4.5, 4.5),
                        y=c(0.95, 0.9, 0.85),
                        cex=0.9,
                        plot=1)

# Plot traces using tile
tile(aleague.Trace,
     nleague.Trace,
     labelTrace,
     legendTrace,
```

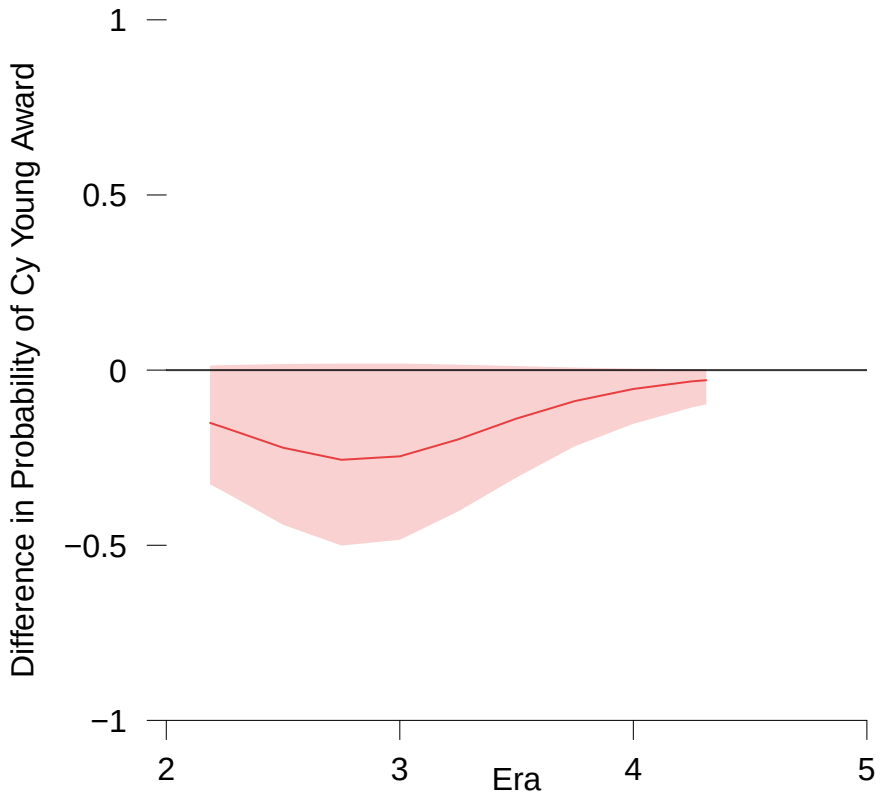
```
    limits=c(2,5,0,1),
    xaxis=list(at=c(2,3,4,5)),
    xaxistitle=list(labels="Era"),
    yaxistitle=list(labels="Probability of Cy Young Award"),
    width=list(null=5,yaxistitle=4,yaxis.labelspace=-0.5)
    )
```



Since the American league has Dh system which increases the risk of homerun, the result confirmed that pitchers in AL could be awarded Cy young with higher `era`.

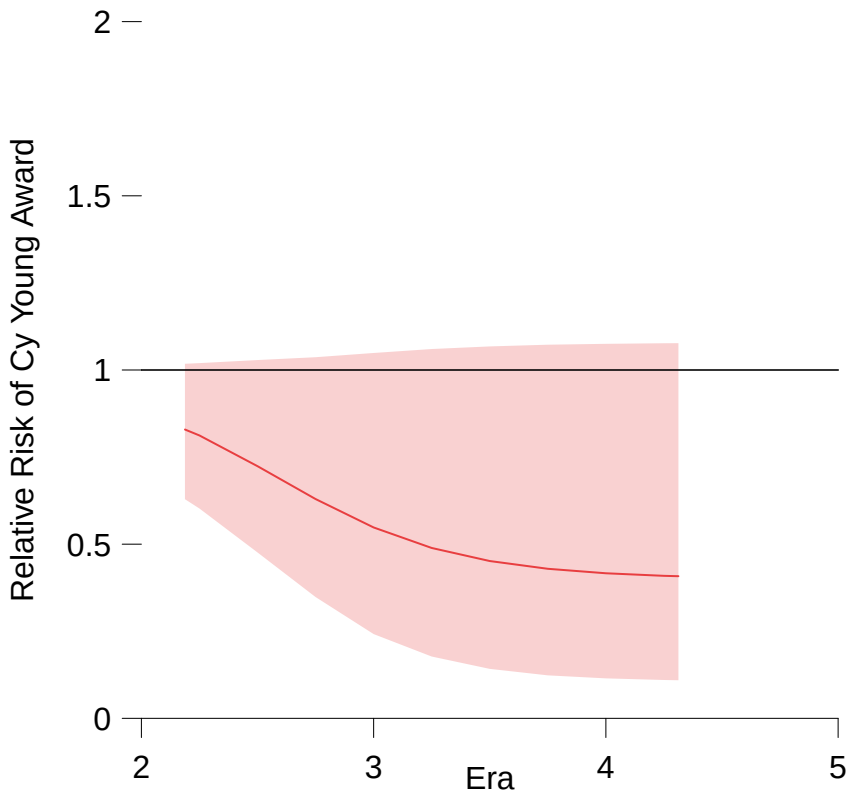
```
# Plot First Difference

# Set up lineplot trace of relative risk
leagueFDTrace <- lineplot(x=xhyp,
                          y=leagueFD$pe,
                          lower=leagueFD$lower,
                          upper=leagueFD$upper,
                          col=col[1],
                          extrapolate=list(data=m2data[,2:ncol(m2data)],
                            cfact=nleague.sim$x[,2:ncol(nleague.sim$x)],
                            omit.extrapolated=TRUE),
                          plot=1)


# Set up baseline: for first difference, this is 0
baseline <- linesTile(x=c(2,5),
```

```
                      y=c(0,0),
                      plot=1)

# Set up traces with labels and legend
labelFDTrace <- textTile(labels=c("Married compared \n to Not Married"),
                       x=c(40),
                       y=c( 0.20),
                       col=col[1],
                       plot=1)

legendFDTrace <- textTile(labels=c("Logit estimates:", "95% confidence", "interval is shaded"),
                        x=c(80, 80, 80),
                        y=c(-0.02, -0.05, -0.08),
                        cex=0.9,
                        plot=1)

# Plot traces using tile
tile(leagueFDTrace,
     labelFDTrace,
     legendFDTrace,
     baseline,
     limits=c(2,5,-1,1),
     xaxis=list(at=c(2,3,4,5)),
     yaxis=list(label.loc=-0.5, major=FALSE),
     xaxistitle=list(labels="Era"),
     yaxistitle=list(labels="Difference in Probability of Cy Young Award"),
     width=list(null=5,yaxistitle=4,yaxis.labelspace=-0.5)

     )
```




```
# Plot Relative Risk

# Set up lineplot trace of relative risk
leagueRRTrace <- lineplot(x=xhyp,
                          y=leagueRR$pe,
                          lower=leagueRR$lower,
                          upper=leagueRR$upper,
                          col=col[1],
                          extrapolate=list(data=m2data[,2:ncol(m2data)],
                          cfact=nleague.sim$x[,2:ncol(nleague.sim$x)],
                          omit.extrapolated=TRUE),
                          plot=1)


# Set up baseline: for relative risk, this is 1
baseRRline <- linesTile(x=c(2,5),
                        y=c(1,1),
                        plot=1)

# Set up traces with labels and legend
labelRRTrace <- textTile(labels=c("Married compared \n to Not Married"),
                         x=c( 55),
                         y=c( 1.25),
                         col=col[1],
                         plot=1)
```

```
legendRRTrace <- textTile(labels=c("Logit estimates:", "95% confidence", "interval is shaded"),
                          x=c(80, 80, 80),
                          y=c(0.98, 0.95, 0.92),
                          cex=0.9,
                          plot=1)

# Plot traces using tile
tile(leagueRRTrace,
     labelRRTrace,
     legendRRTrace,
     baseRRline,
     limits=c(2,5,0,2),
     xaxis=list(at=c(2,3,4,5)),
     yaxis=list(label.loc=-0.5, major=FALSE),
     xaxistitle=list(labels="Era"),
     yaxistitle=list(labels="Relative Risk of Cy Young Award"),
     width=list(null=5,yaxistitle=4,yaxis.labelspace=-0.5))
```



Both the first difference in the probability and the relative risk confirm the differences.

Answer f.:

The logistic regression model assumes that outcome is a binary and an independent. Also, the model assumes that there is a linear relationship between the logit of the outcome and each covariate. However, the distribution of outcome variable, `cy` is not independent since the winners are limited to two players from

each league per year. This means that only two times the number of years players can be awared Cy Young, suggesting zeros are inflating. Furthermore, the winners of Cy Young Award are highly depended on their team. This also shows that the outcome variable is not independent.

Answer g.:

The question asks if we have a sampling bias or not. The expert may consider that all covariates are normally distributed and choose the top tier of the sample. If so, our results would not be changed drastically. The full dataset of all pitchers shows that the variance of our estimates would be small by increasing sample size. However, if the expert's selection is skewed to the specific ability of the pitchers, we may have overfitting problem on our estimation. (though we should take into account that the data is zero-inflated).