Class 6

BUS 696

Prof. Jonathan Hersh

BUS 696: Class 6 Announcements

1. Pset 4 solutions
2. Problem Set 5 Posted
3. Final Project
    - Project details posted
4. Questions?

BUS 696: Class 6 Outline

1. AI in the News
2. Review
3. Lift Charts
4. Calibration Plots
5. Severe Class Imbalance Issues
6. Leave-One-Out Cross-Validation
7. K-Fold Cross-Validation
8. Bootstrapping

Review: Sensitivity and Specificity

		True default status		
		No	**Yes**	
Predicted default status (cutoff p>0.6)	**No**	TN = 485	FN = 11	N^* = 495
	Yes	FP = 1	TP = 3	P^*= 5
		N = 486	P = 14	

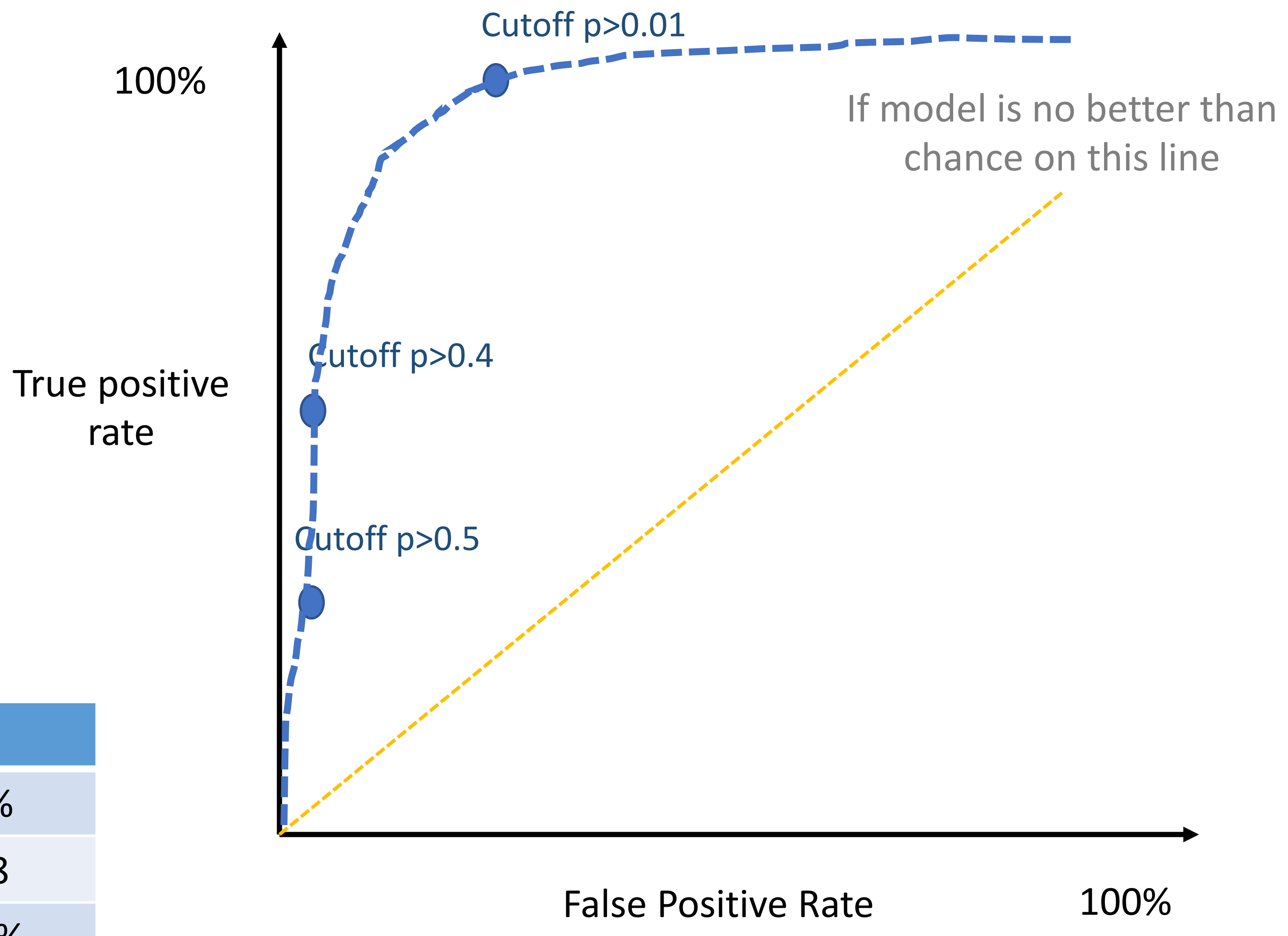
- **Sensitivity:** True positive rate (aka 1 – power or recall)
  - TP/P = 3 / 14 = 21.4%
- **Specificity:** True negative rate
  - TN/N = 485 / 486= 99.7%

- **False positive rate** (aka Type I error, 1 - Specificity)
  - FP/N = 1/486= 0.002%

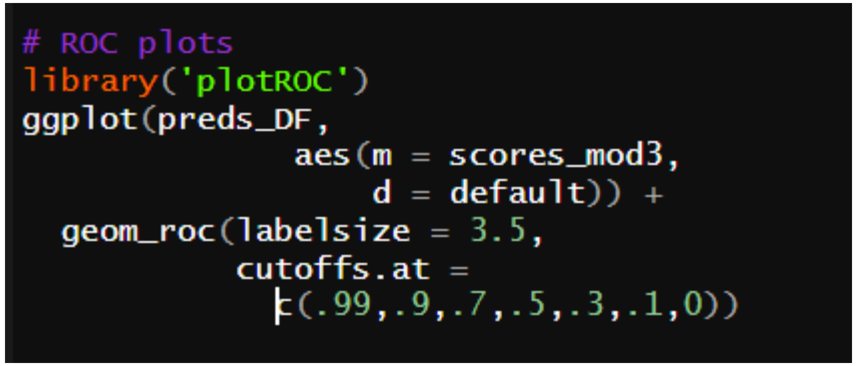
ROC Curve

- Can we show consequences of FPs and FNs as we vary the cutoff probability to assign classes?
- Idea of a ROC (Receiver Operator Curve) plot

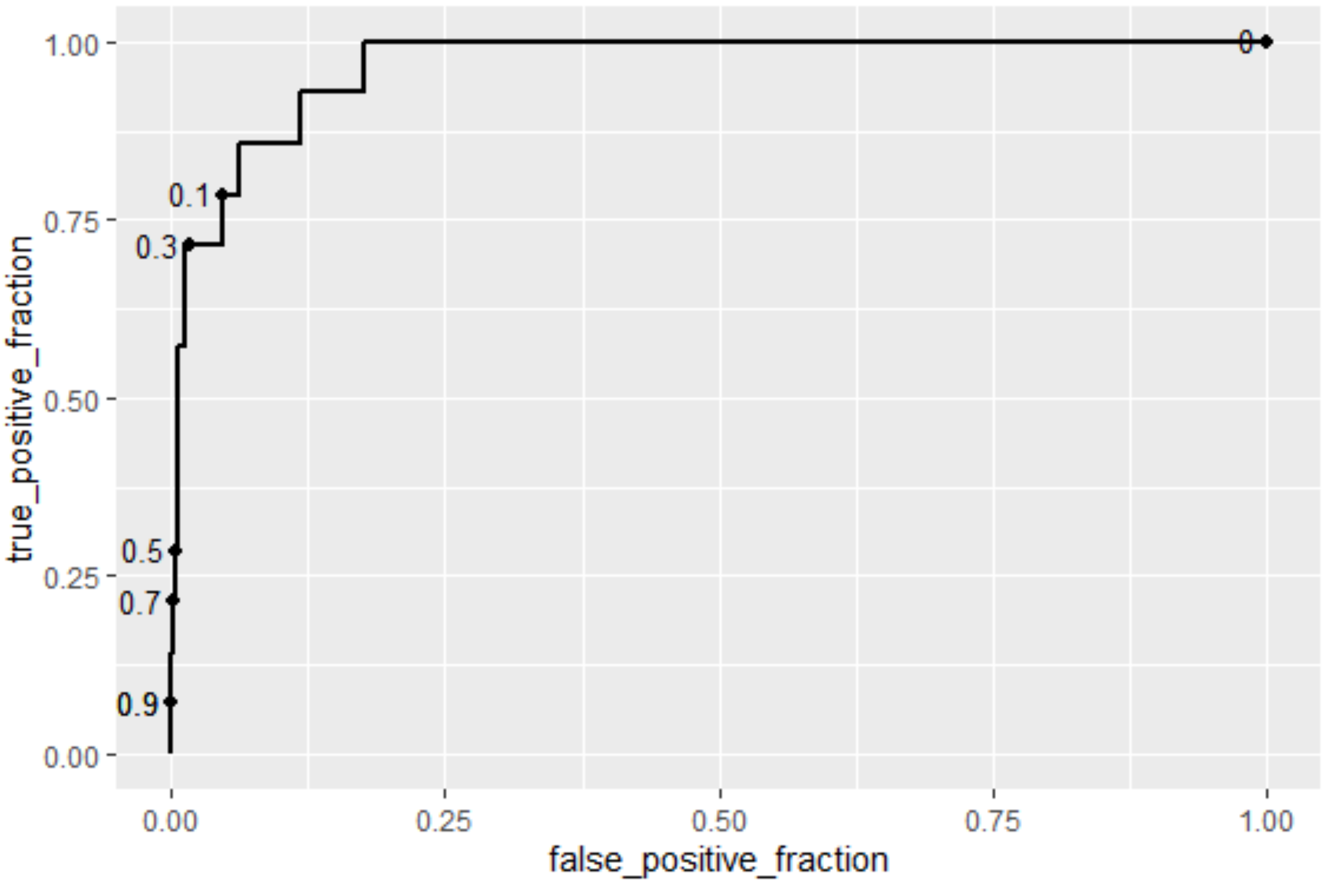
Cutoff	TPR	FPR
0.01	100%	22.6%
0.4	57%	0.008
0.5	21.4%	0.004%
0.6	21.4%	0.002%



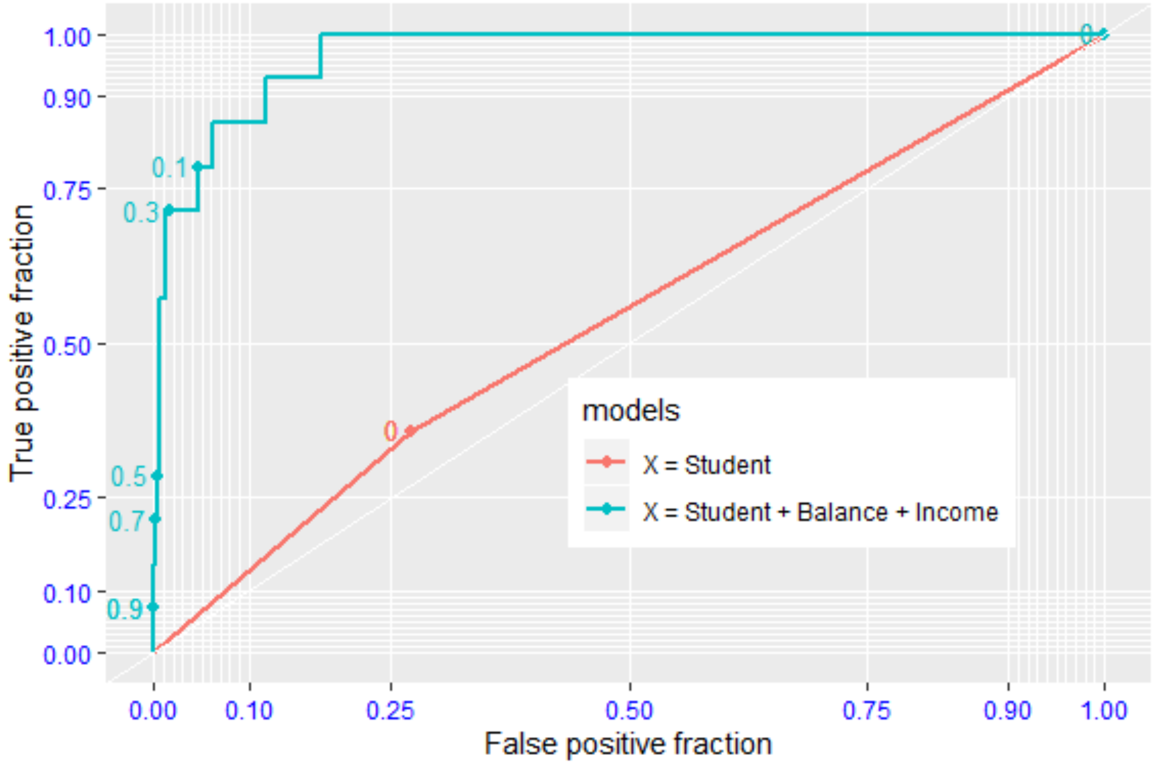
ROC Curves in R



```
# ROC plots
library('plotROC')
ggplot(preds_DF,
              aes(m = scores_mod3,
                  d = default)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at =
             c(.99,.9,.7,.5,.3,.1,0))
```

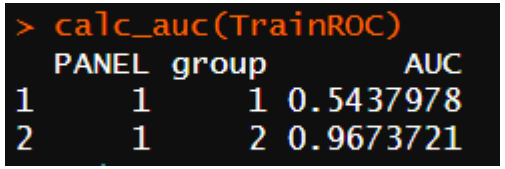


AUC = Area Under Curve



- Better models lie up and to the left
- AUC calculates how much total area is under a particular curve

- E.g. model with just X = student is much worse than X = student + balance + income



```
> calc_auc(TrainROC)
  PANEL group       AUC
1     1     1 0.5437978
2     1     2 0.9673721
```

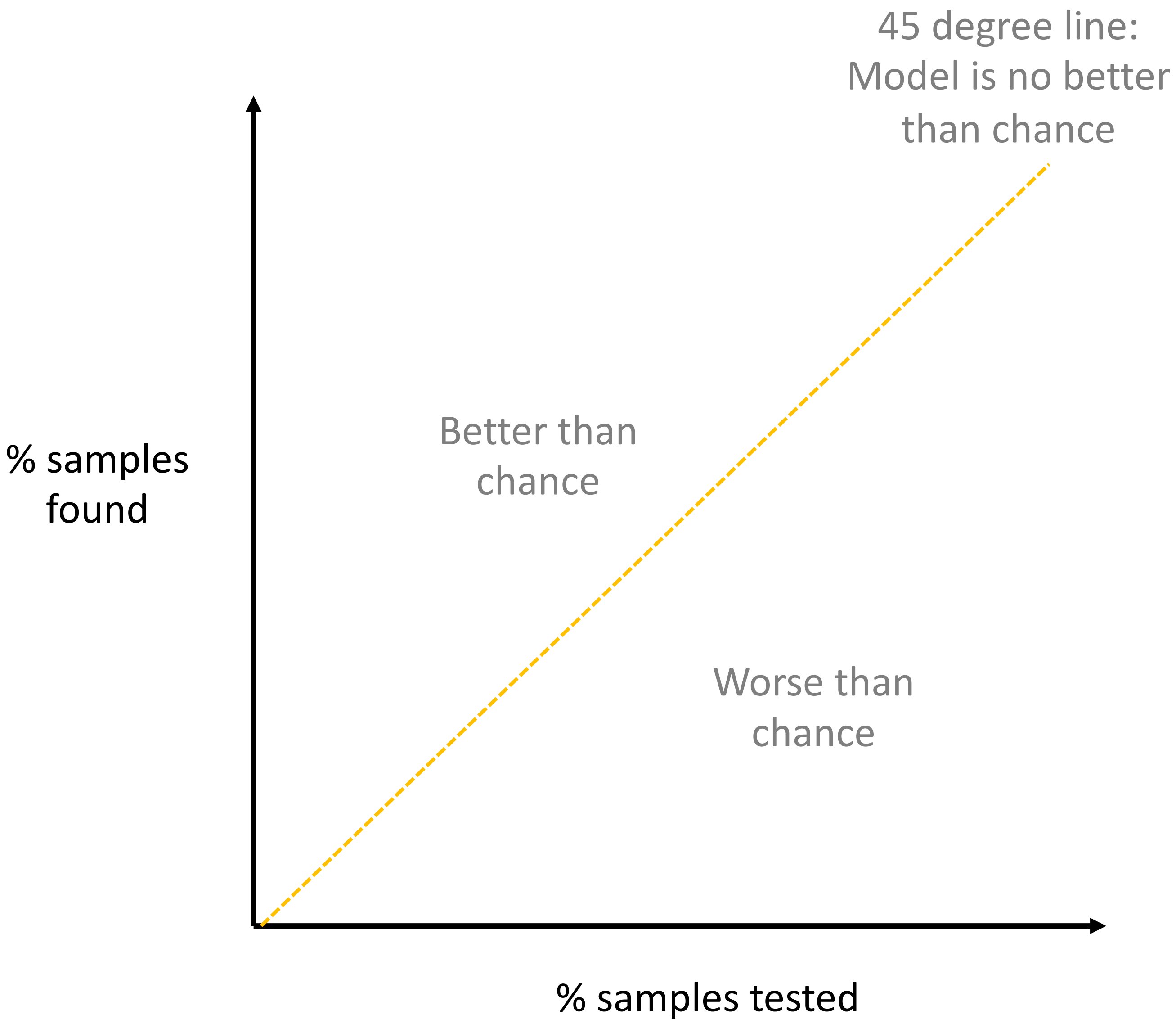
BUS 696: Class 6 Outline

1. AI in the News
2. Review
3. **Lift Charts**
4. Calibration Plots
5. Severe Class Imbalance Issues
6. Leave-One-Out Cross-Validation
7. K-Fold Cross-Validation
8. Bootstrapping

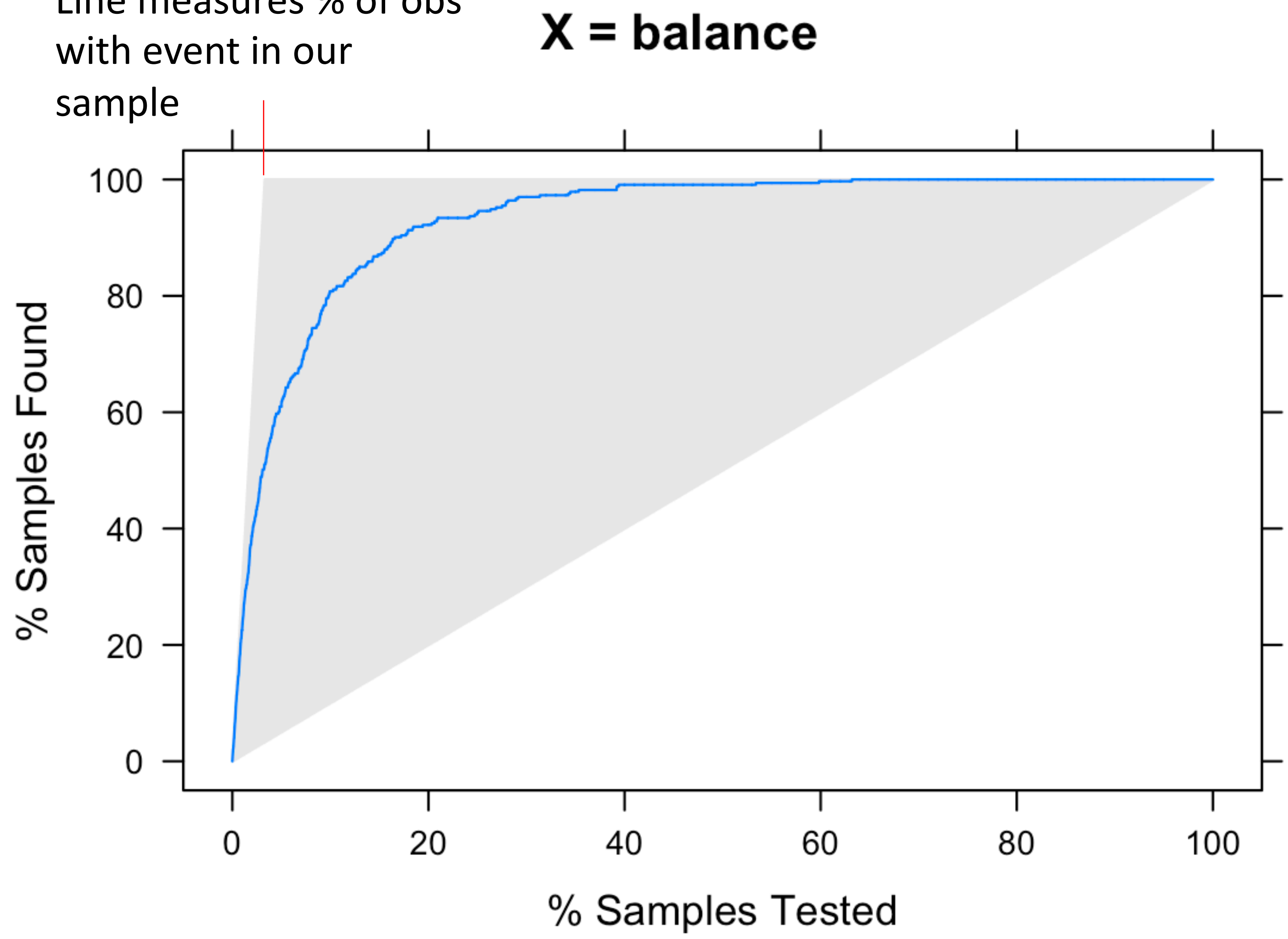
Lift Chart

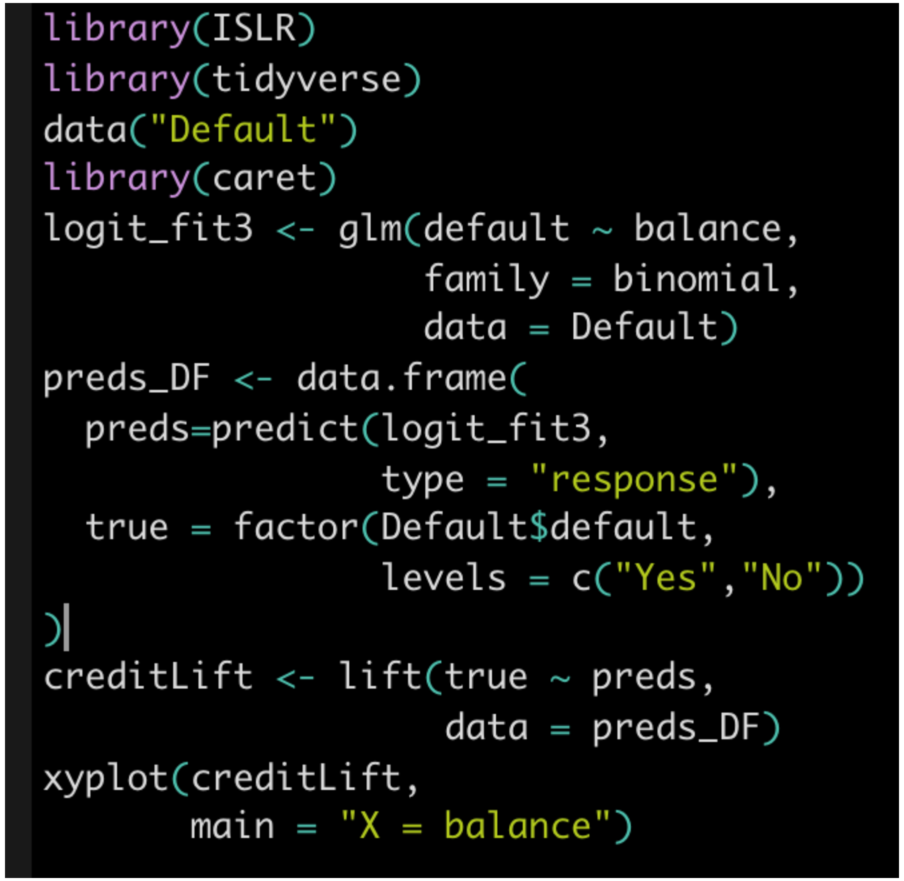
- How useful is our predictive model?

- One way of judging this is to say if we order observations by model predictions (highest to lowest) how many observations will we have to "test" before we find all observations with the event



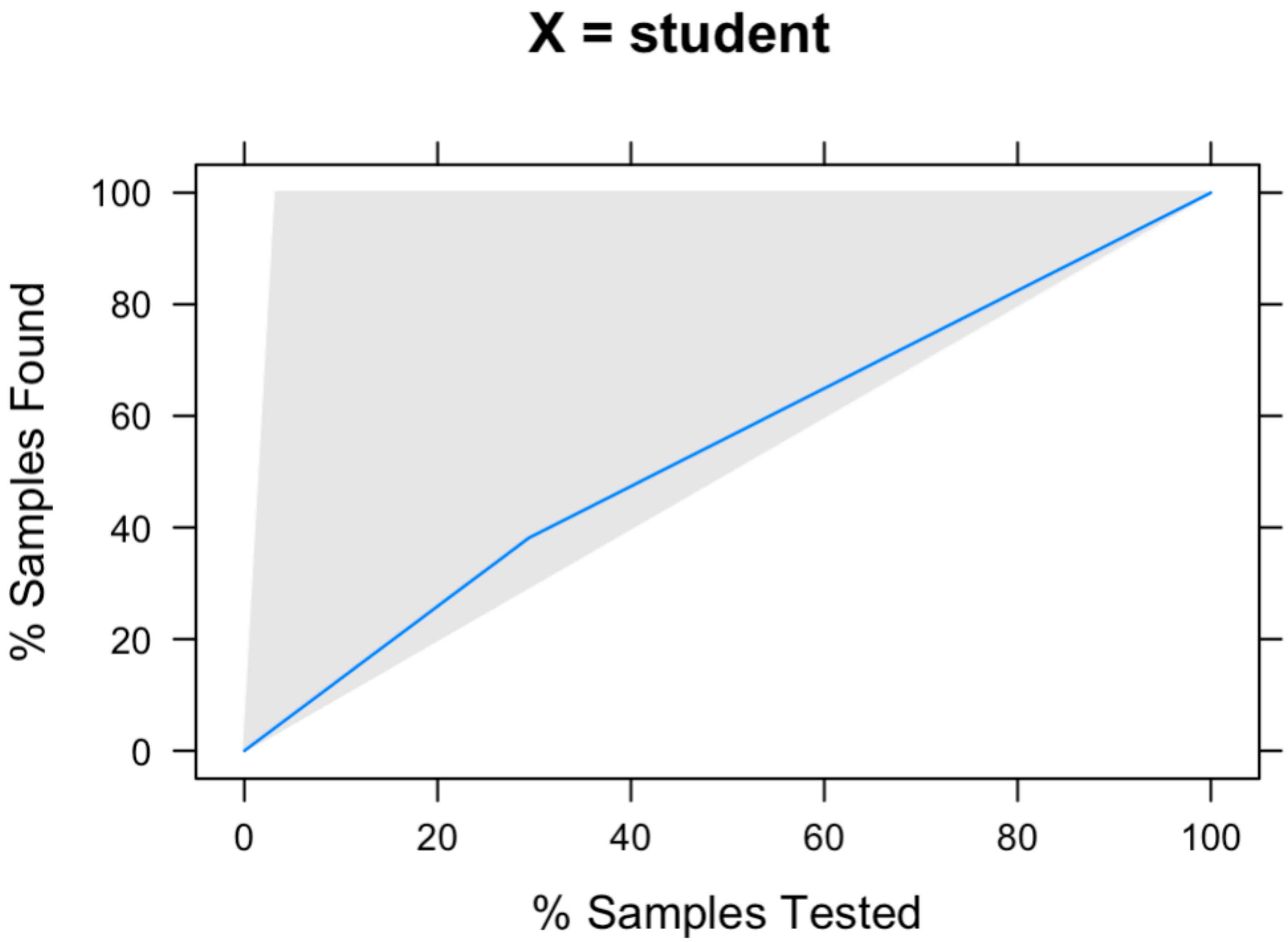
Line measures % of obs with event in our sample

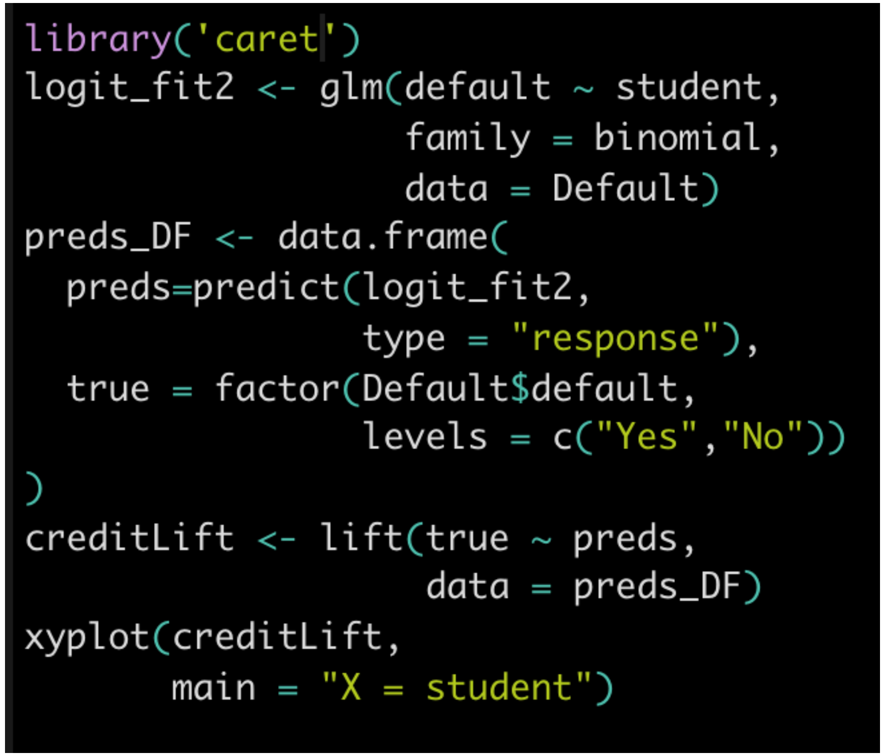




```
library(ISLR)
library(tidyverse)
data("Default")
library(caret)
logit_fit3 <- glm(default ~ balance,
                  family = binomial,
                  data = Default)
preds_DF <- data.frame(
  preds=predict(logit_fit3,
                type = "response"),
  true = factor(Default$default,
                levels = c("Yes","No"))
)
creditLift <- lift(true ~ preds,
                   data = preds_DF)
xyplot(creditLift,
       main = "X = balance")
```

Lift Chart





```
library('caret')
logit_fit2 <- glm(default ~ student,
                  family = binomial,
                  data = Default)
preds_DF <- data.frame(
  preds=predict(logit_fit2,
                type = "response"),
  true = factor(Default$default,
                levels = c("Yes","No"))
)
creditLift <- lift(true ~ preds,
                   data = preds_DF)
xyplot(creditLift,
       main = "X = student")
```

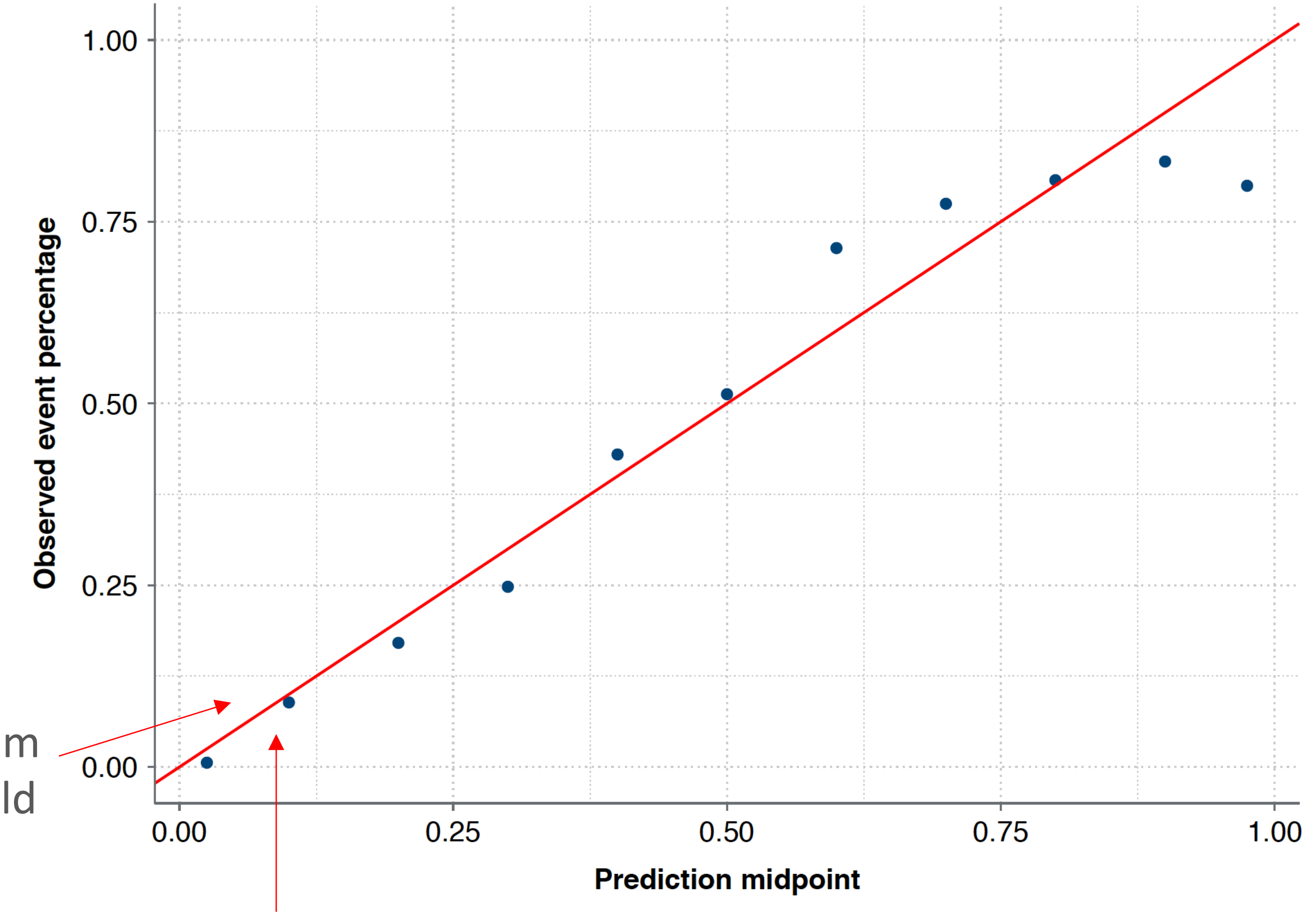
BUS 696: Class 6 Outline

1. AI in the News
2. Review
3. Lift Charts
4. **Calibration Plots**
5. Severe Class Imbalance Issues
6. Leave-One-Out Cross-Validation
7. K-Fold Cross-Validation
8. Bootstrapping

Calibration plot

- Idea of a calibration plot:

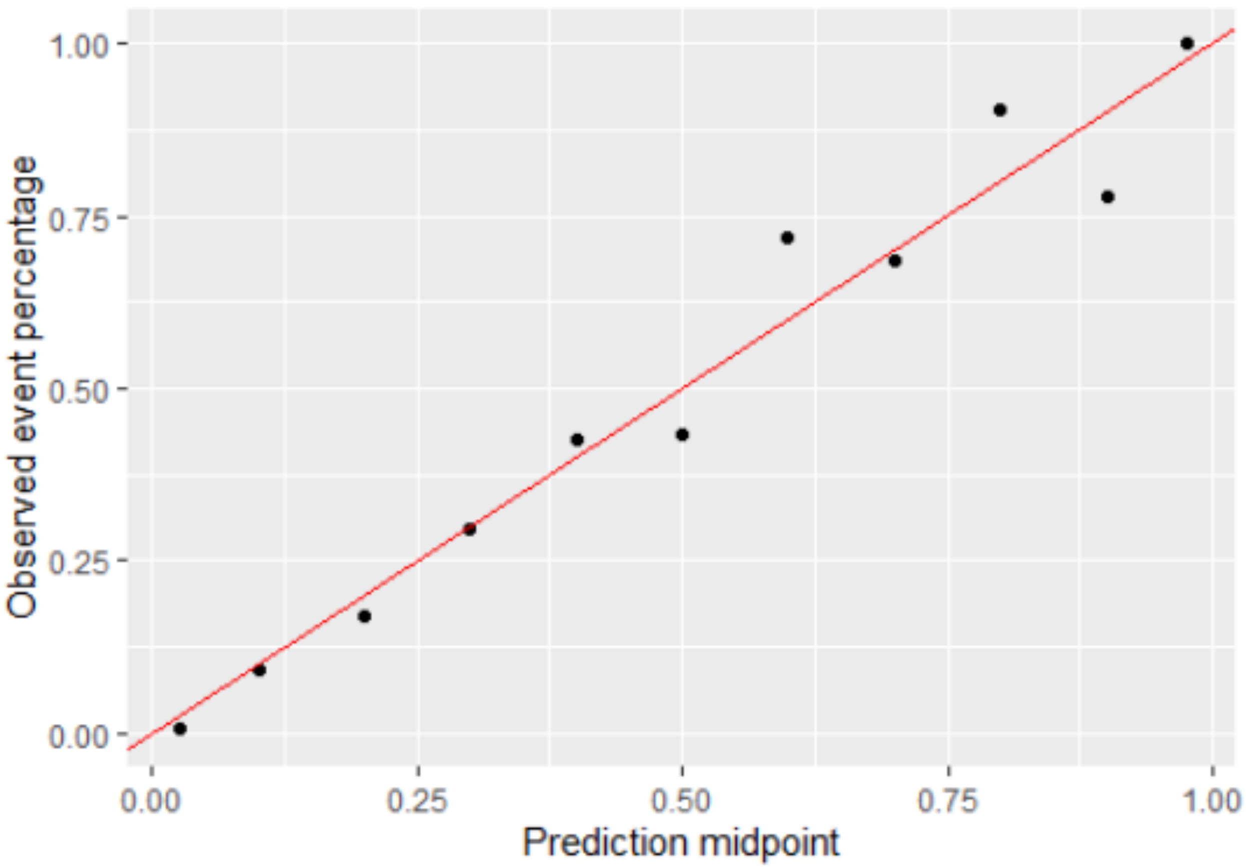
- A well-calibrated prediction is one where if we give an X% probability of an event occurring, X/100 of the time the event happens.

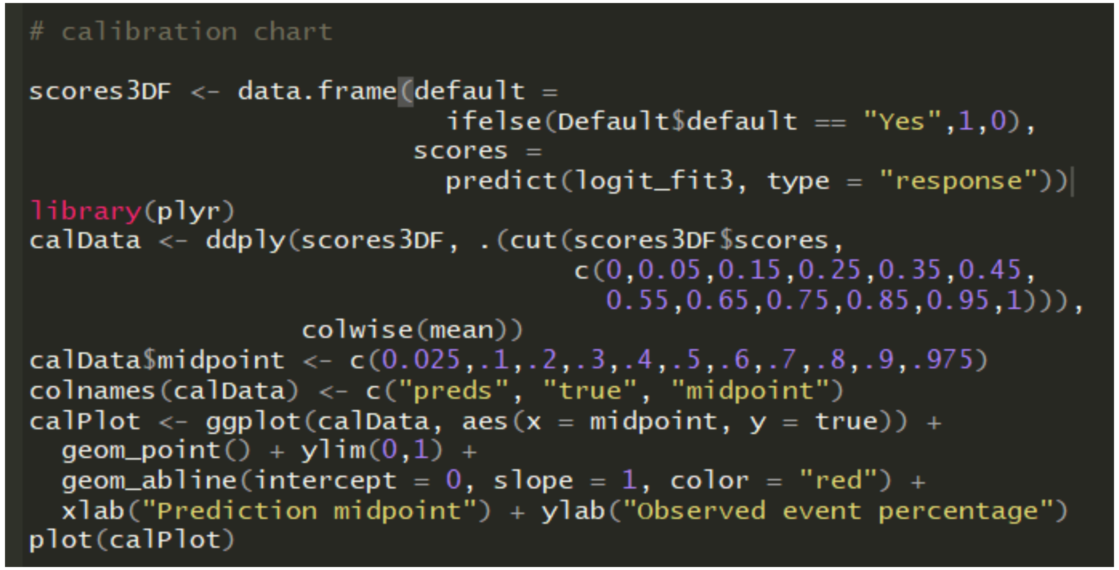


For 1/10 of them the event should actually occur

All obs where we predict 10% chance of event occuring

Calibration plot





```
# calibration chart

scores3DF <- data.frame(default =
                          ifelse(Default$default == "Yes",1,0),
                        scores =
                          predict(logit_fit3, type = "response"))
library(plyr)
calData <- ddply(scores3DF, .(cut(scores3DF$scores,
                                  c(0,0.05,0.15,0.25,0.35,0.45,
                                    0.55,0.65,0.75,0.85,0.95,1))),
                 colwise(mean))
calData$midpoint <- c(0.025,.1,.2,.3,.4,.5,.6,.7,.8,.9,.975)
colnames(calData) <- c("preds", "true", "midpoint")
calPlot <- ggplot(calData, aes(x = midpoint, y = true)) +
  geom_point() + ylim(0,1) +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  xlab("Prediction midpoint") + ylab("Observed event percentage")
plot(calPlot)
```

BUS 696: Class 6 Outline

1. AI in the News
2. Review
3. Lift Charts
4. Calibration Plots
5. **Severe Class Imbalance Issues**
6. Leave-One-Out Cross-Validation
7. K-Fold Cross-Validation
8. Bootstrapping

Severe class imbalance

- Severe class imbalance occurs when one class highly outnumbers another
    - e.g. 98% non-defaults and 2% defaults
- Even fancy models can't solve this

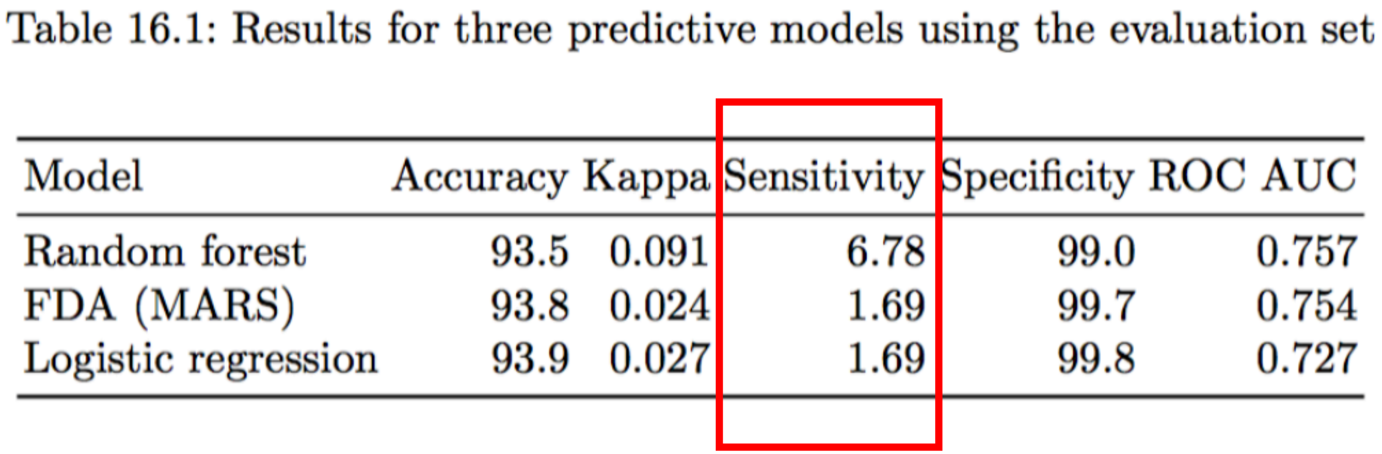
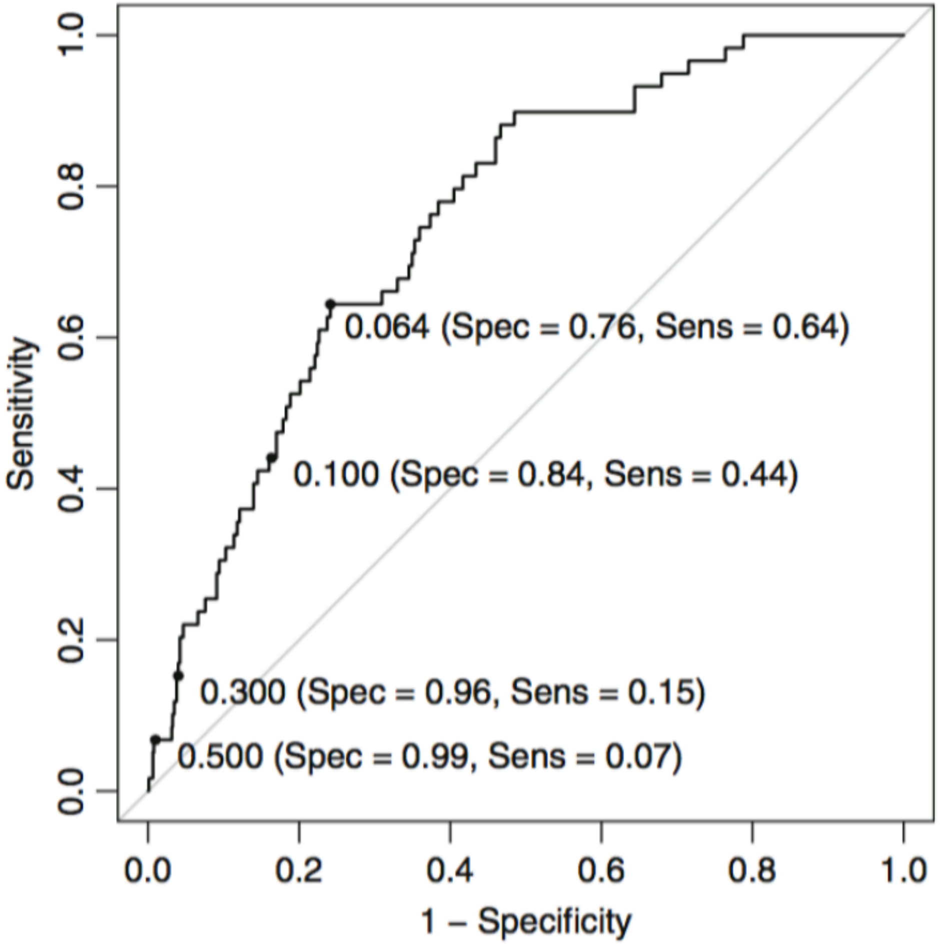


Table 16.1: Results for three predictive models using the evaluation set

Model	Accuracy	Kappa	Sensitivity	Specificity	ROC AUC
Random forest	93.5	0.091	6.78	99.0	0.757
FDA (MARS)	93.8	0.024	1.69	99.7	0.754
Logistic regression	93.9	0.027	1.69	99.8	0.727

Low sensitivity => low TP/P = True Positive Rate

Remedy 1: alternate score cutoff



Remedies 2,3&4 under-sample majority class, oversample minority class, SMOTE interpolate minority class

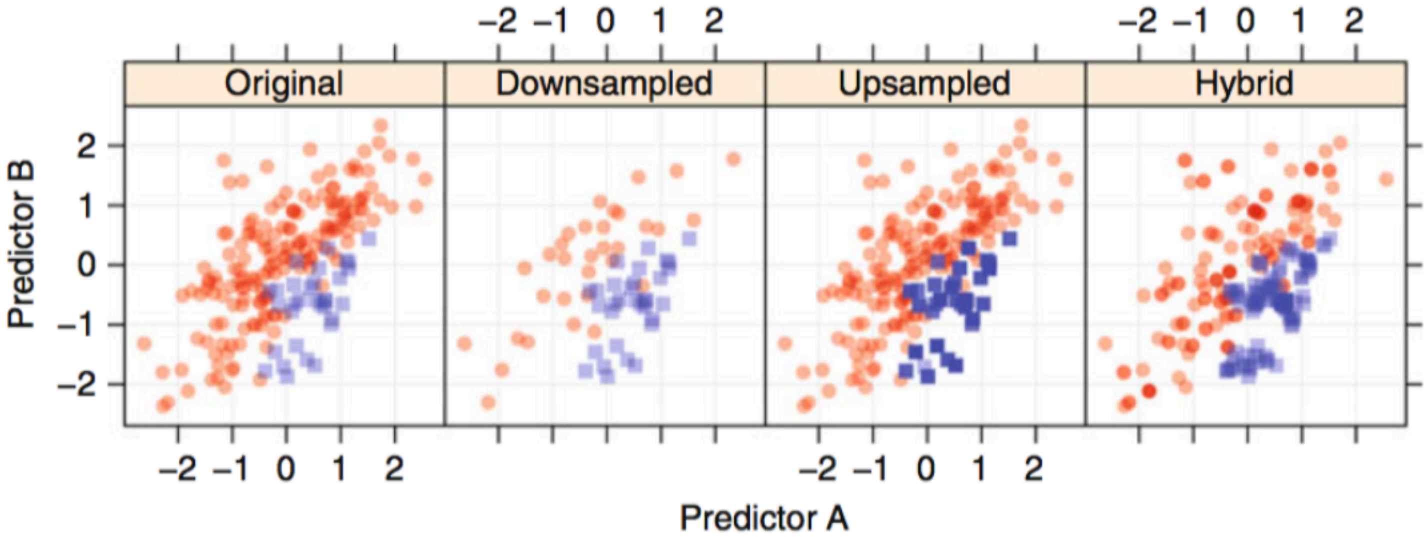
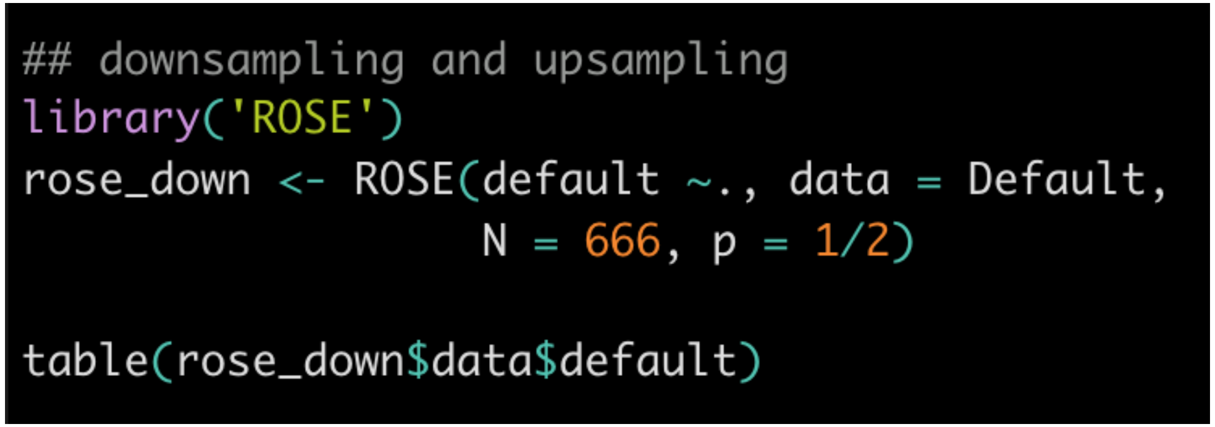


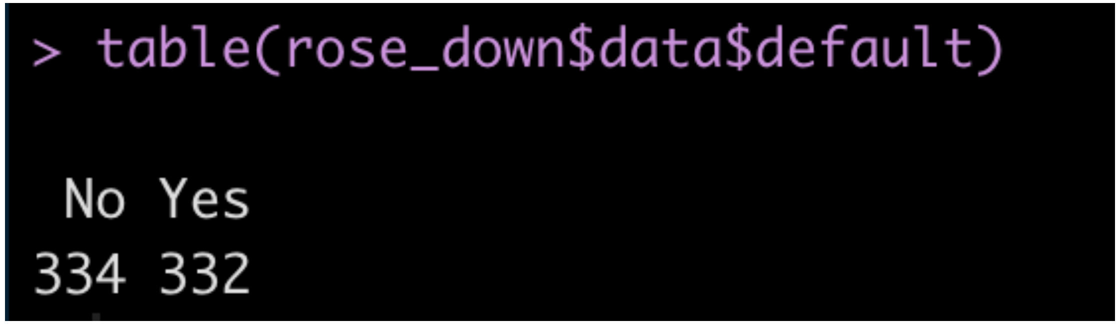
Fig. 16.3: *From left to right*: The original simulated data set and realizations of a down-sampled version, an up-sampled version, and sampling using SMOTE where the cases are sampled and/or imputed

Downsampling using ROSE



```
## downsampling and upsampling
library('ROSE')
rose_down <- ROSE(default ~., data = Default,
                  N = 666, p = 1/2)

table(rose_down$data$default)
```



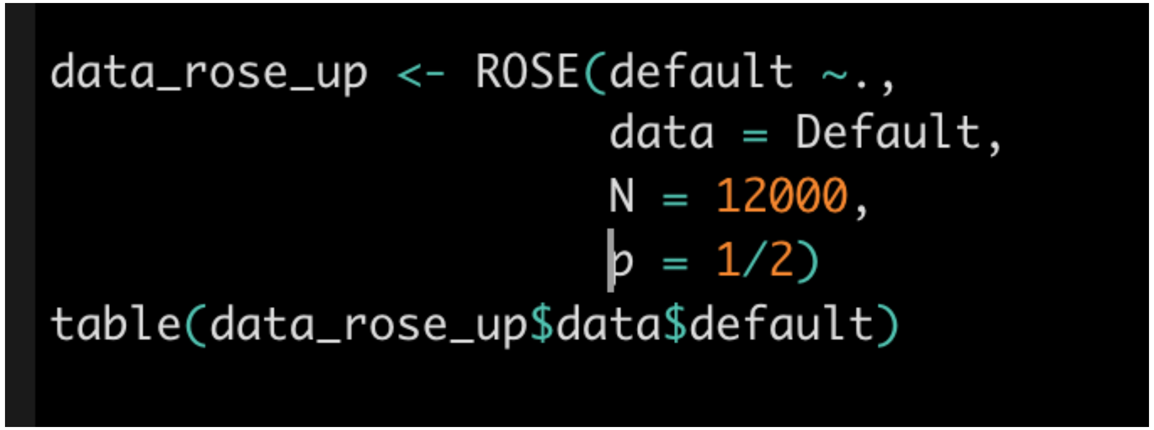
```
> table(rose_down$data$default)

 No Yes
334 332
```

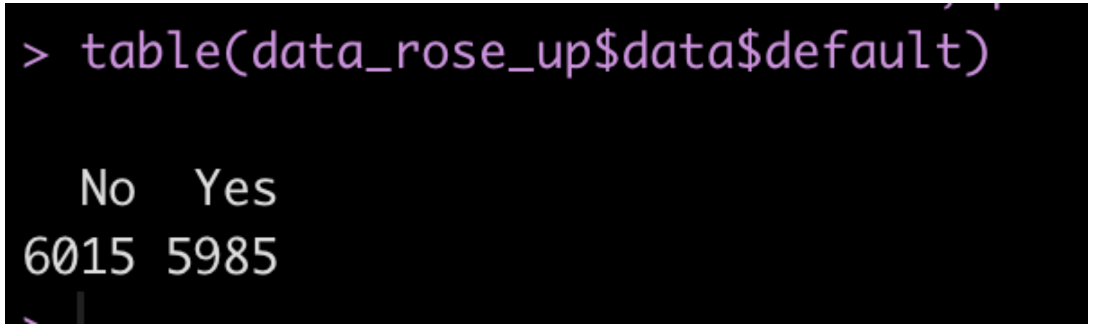
- N = size of dataset desired
- p = probability of event in desired dataset

- Set N = p*nrow(origina datset) to down sample

Upsampling using ROSE



```
data_rose_up <- ROSE(default ~.,
                     data = Default,
                     N = 12000,
                     p = 1/2)
table(data_rose_up$data$default)
```



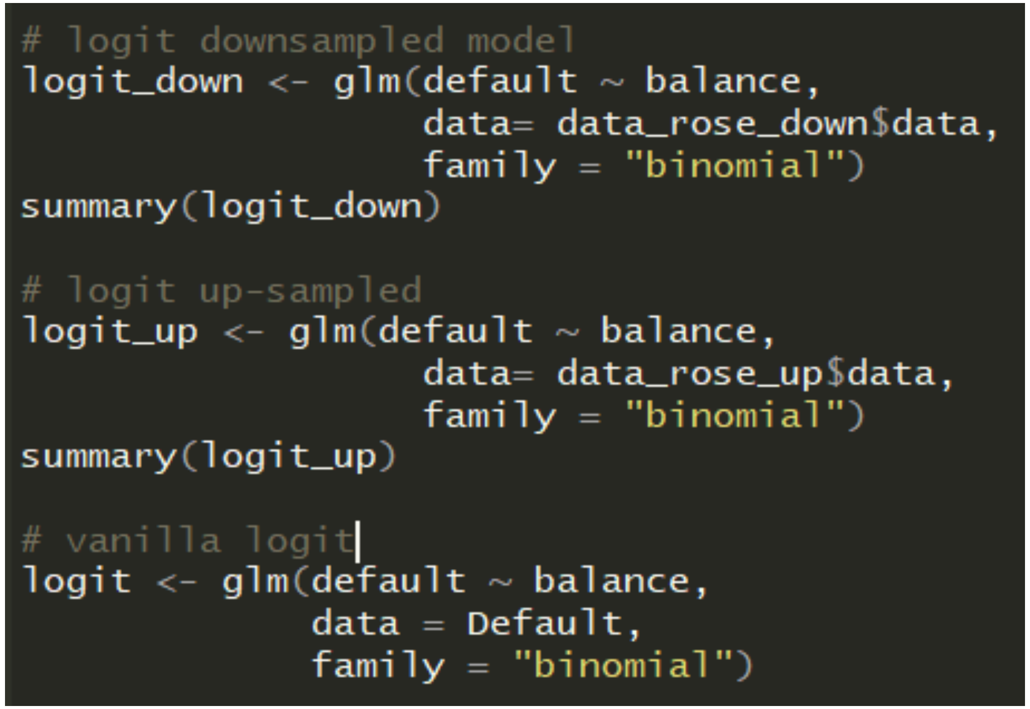
```
> table(data_rose_up$data$default)

  No  Yes 
6015 5985 
```

- N = size of dataset desired
- p = probability of event in desired dataset

- Set N > nrow(origina datset) to up sample

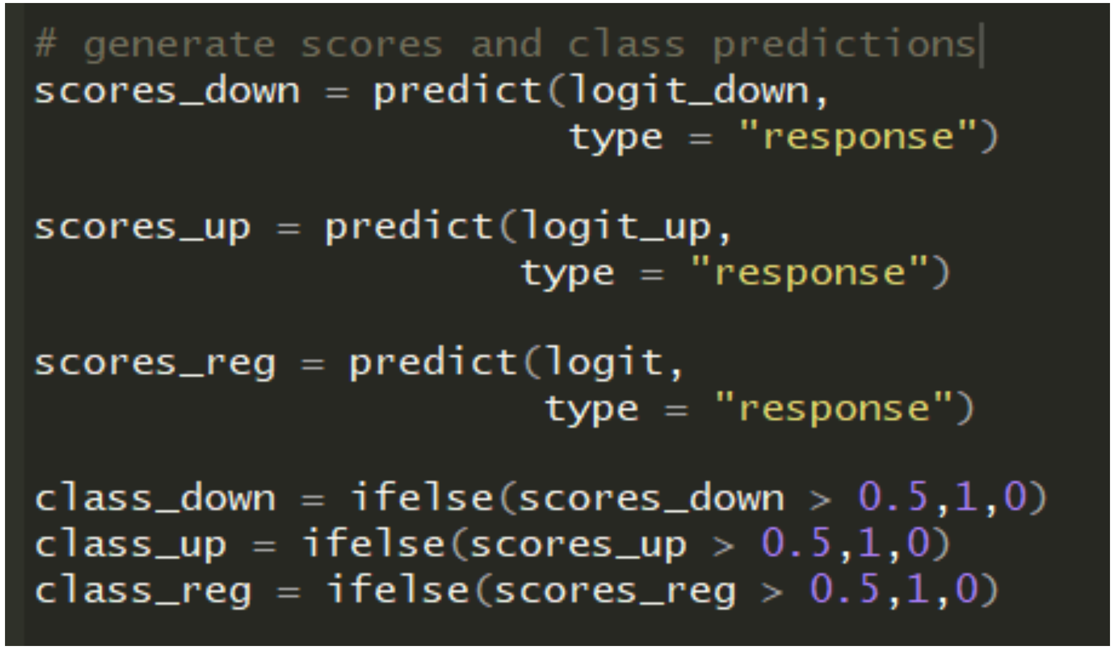
Fit Models on Upsampled/Downsampled Data



```r
# logit downsampled model
logit_down <- glm(default ~ balance,
                  data= data_rose_down$data,
                  family = "binomial")
summary(logit_down)

# logit up-sampled
logit_up <- glm(default ~ balance,
                  data= data_rose_up$data,
                  family = "binomial")
summary(logit_up)

# vanilla logit
logit <- glm(default ~ balance,
             data = Default,
             family = "binomial")
```



```r
# generate scores and class predictions
scores_down = predict(logit_down,
                      type = "response")

scores_up = predict(logit_up,
                    type = "response")

scores_reg = predict(logit,
                     type = "response")

class_down = ifelse(scores_down > 0.5,1,0)
class_up = ifelse(scores_up > 0.5,1,0)
class_reg = ifelse(scores_reg > 0.5,1,0)
```

Model Diagnostics on Up-Sampled/Down-Sampled Models

	Vanilla Logit	Down-Sampled Logit	Up-Sampled Logit
Accuracy	0.872	0.869	0.972
True Positives	307	5282	100
True Negs	274	5149	9625
Sensitivity	0.9	0.88	0.3
Specificity	0.843	0.859	0.996
False Pos Rate	0.157	0.141	0.004

Remedy 5: Adjust Likelihood Cost Parameter

- Another alternative: adjust “cost” of FN in Likelihood estimation
- We won’t cover this, but it’s useful in practice.

BUS 696: Class 6 Outline

1. AI in the News
2. Review
3. Lift Charts
4. Calibration Plots
5. Severe Class Imbalance Issues
6. **Leave-One-Out Cross-Validation**
7. K-Fold Cross-Validation
8. Bootstrapping

Resampling: Test-Validation Set Approach

- Recall: to approximate out of sample error we can set up a test and training split
- Problem: we only get one shot at building a test set. What if we select a weird test set (high variance to MSE_{tset})
- Can always build multiple test sets, but may not have enough observations for multiple test sets

mpg	cyl	Displ
20	4	3
15	6	5
12	4	2.4
10	8	4.6
14	6	3
25	4	2

$\hat{f}(X^{train})$ training

$\hat{y}^{test} = \hat{f}(X^{test})$ test

$$MSE_{tset} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left(y_i^{test} - \hat{y}^{test}\right)^2$$

Resampling: Leave-One-Out Cross-Validation

- Idea of LOOCV: Let’s approximate a bunch of test sets, each of size 1
- We start with estimating a model using every observation except 1.
- Use that model to predict into observation 1.

$\hat{y}^{LOOCV} = \hat{f}_{X^1}(X^1)$

$\hat{f}_{X^{-1}}(X^{-1})$ training

$\hat{y}^{LOOCV}$	mpg	cyl	Displ
18	20	4	3
	15	6	5
	12	4	2.4
	10	8	4.6
	14	6	3
	25	4	2

X^{-1}: X excluding observation 1

Resampling: Leave-One-Out Cross-Validation

- We then exclude observation 2 use observations 1,3,..,n to fit a model.
- We use the estimates from that model to predict into observation 2

$\hat{y}^{LOOCV}$	mpg	cyl	Displ
18	20	4	3
16	15	6	5
	12	4	2.4
	10	8	4.6
	14	6	3
	25	4	2

training

$\hat{y}^{LOOCV} = \hat{f}_{X^{-2}}(X^{-2})$

$\hat{f}_{X^{-2}}(X^{-2})$ training

X^{-2}: X excluding observation 2

Resampling: Leave-One-Out Cross-Validation

- We then exclude observation 2 use observations 1,3,..,n to fit a model.
- We use the estimates from that model to predict into observation 2
- And we proceed in that manner until we have predictions for every observation

training

$\hat{y}^{LOOCV} = \hat{f}_{X^{-3}}(X^{-3})$

$\hat{f}_{X^{-3}}(X^{-3})$ training

$\widehat{y}^{CV}$	mpg	cyl	Displ
18	20	4	3
16	15	6	5
12	12	4	2.4
	10	8	4.6
	14	6	3
	25	4	2

X^{-3}: X excluding observation 3

Resampling: Leave-One-Out Cross-Validation

- We then exclude observation 2 use observations 1,3,..,n to fit a model.
- We use the estimates from that model to predict into observation 2
- And we proceed in that manner until we have predictions for every observation

$\hat{f}_{X^{\{-4\}}}(X^{\{-4\}})$ training

$\hat{y}^{LOOCV} = \hat{f}_{X^{\{-4\}}}(X^{\{4\}})$

training

$\hat{y}^{CV}$	mpg	cyl	Displ
18	20	4	3
16	15	6	5
12	12	4	2.4
11	10	8	4.6
	14	6	3
	25	4	2

$X^{\{-4\}}$: X excluding observation 4

Resampling: Leave-One-Out Cross-Validation

- We then exclude observation 2 use observations 1,3,..,n to fit a model.
- We use the estimates from that model to predict into observation 2
- And we proceed in that manner until we have predictions for every observation

$\hat{f}_{X^{\{-5\}}}(X^{\{-5\}})$ training

$\hat{y}^{LOOCV} = \hat{f}_{X^{\{-5\}}}(X^{\{5\}})$

training

$\hat{y}^{LOOCV}$	mpg	cyl	Displ
18	20	4	3
16	15	6	5
12	12	4	2.4
11	10	8	4.6
15	14	6	3
	25	4	2

$X^{\{-5\}}$: X excluding observation 1

Resampling: Leave-One-Out Cross-Validation

- We then exclude observation 2 use observations 1,3,..,n to fit a model.
- We use the estimates from that model to predict into observation 2
- And we proceed in that manner until we have predictions for every observation

$\hat{f}_{X^{\{-6\}}}(X^{\{-6\}})$ training

$\hat{y}^{LOOCV} = \hat{f}_{X^{\{-6\}}}(X^{\{6\}})$

$\hat{y}^{LOOCV}$	mpg	cyl	Displ
18	20	4	3
16	15	6	5
12	12	4	2.4
11	10	8	4.6
11	14	6	3
22	25	4	2

$X^{\{-6\}}$: X excluding observation 6

Leave-One-Out Cross-Validation

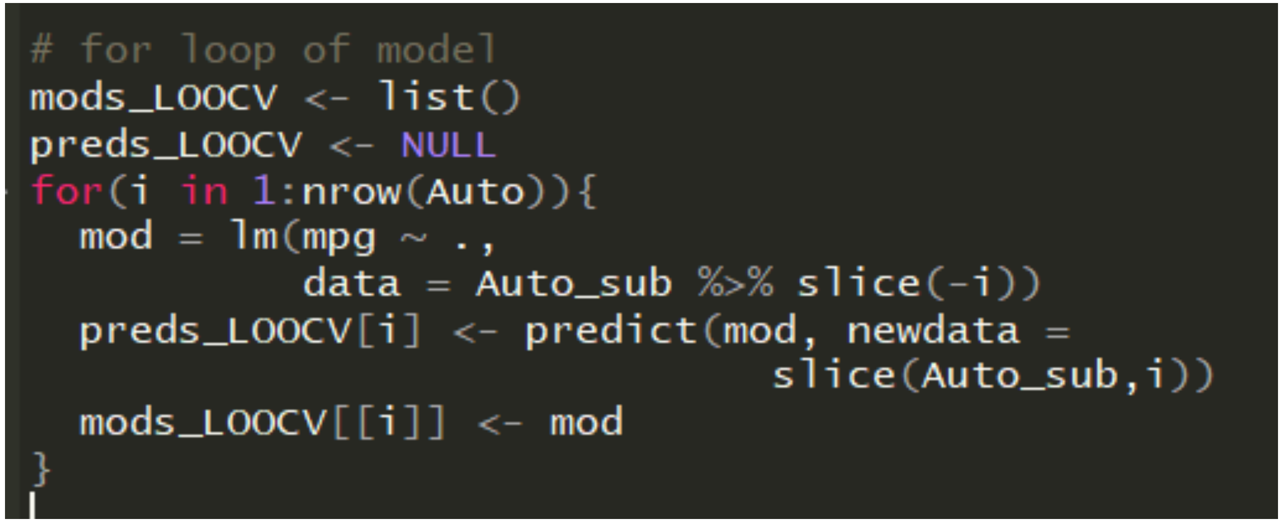
- At the end we have a series of $\hat{y}^{LOOCV}$.
- These were calculated using models that were trained on data excluding this observation
- Kind of like a training set, right? Like N (number of rows of the dataset) training sets.
- We can then calculate MSE_{CV} which is mean-squared-error calculated using ${\hat{y}_i}^{LOOCV}$s.

$\hat{y}^{LOOCV}$	mpg	cyl	Displ
18	20	4	3
16	15	6	5
12	12	4	2.4
11	10	8	4.6
11	14	6	3
22	25	4	2

$$MSE_{LOOCV} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - {\hat{y}_i}^{LOOCV}\right)^2$$

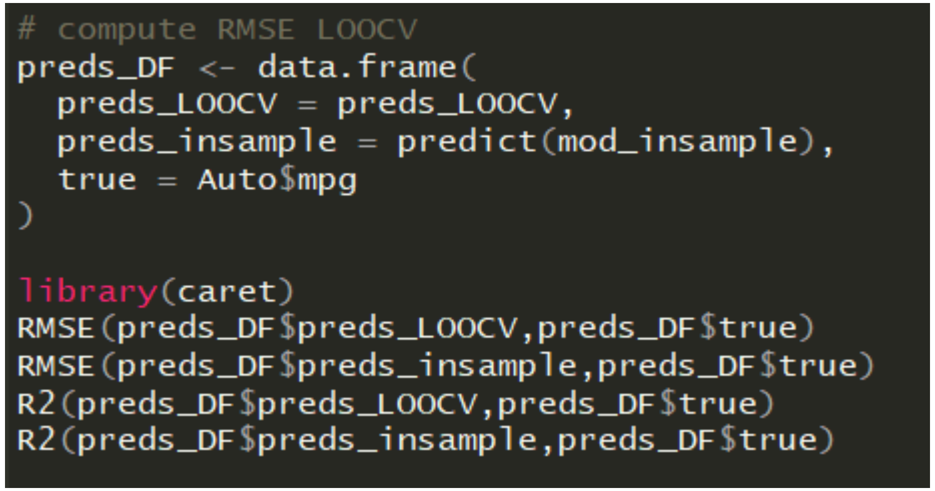
Leave-One-Out Cross-Validation in R

- Many automatic ways to do it (see boot package) but we will try by hand
- In general performance metrics are lower (better) in-sample versus cross-validated



```r
# for loop of model
mods_LOOCV <- list()
preds_LOOCV <- NULL
for(i in 1:nrow(Auto)){
  mod = lm(mpg ~ .,
           data = Auto_sub %>% slice(-i))
  preds_LOOCV[i] <- predict(mod, newdata =
                              slice(Auto_sub,i))
  mods_LOOCV[[i]] <- mod
}
```

	RMSE (pred vs true)	R2 (pred vs true)
In-Sample	3.29	0.82
LOOCV	3.37	0.81



```r
# compute RMSE LOOCV
preds_DF <- data.frame(
  preds_LOOCV = preds_LOOCV,
  preds_insample = predict(mod_insample),
  true = Auto$mpg
)

library(caret)
RMSE(preds_DF$preds_LOOCV,preds_DF$true)
RMSE(preds_DF$preds_insample,preds_DF$true)
R2(preds_DF$preds_LOOCV,preds_DF$true)
R2(preds_DF$preds_insample,preds_DF$true)
```

BUS 696: Class 6 Outline

1. AI in the News
2. Review
3. Lift Charts
4. Calibration Plots
5. Severe Class Imbalance Issues
6. Leave-One-Out Cross-Validation
7. **K-Fold Cross-Validation**
8. Bootstrapping

Resampling: K-Fold Cross-Validation

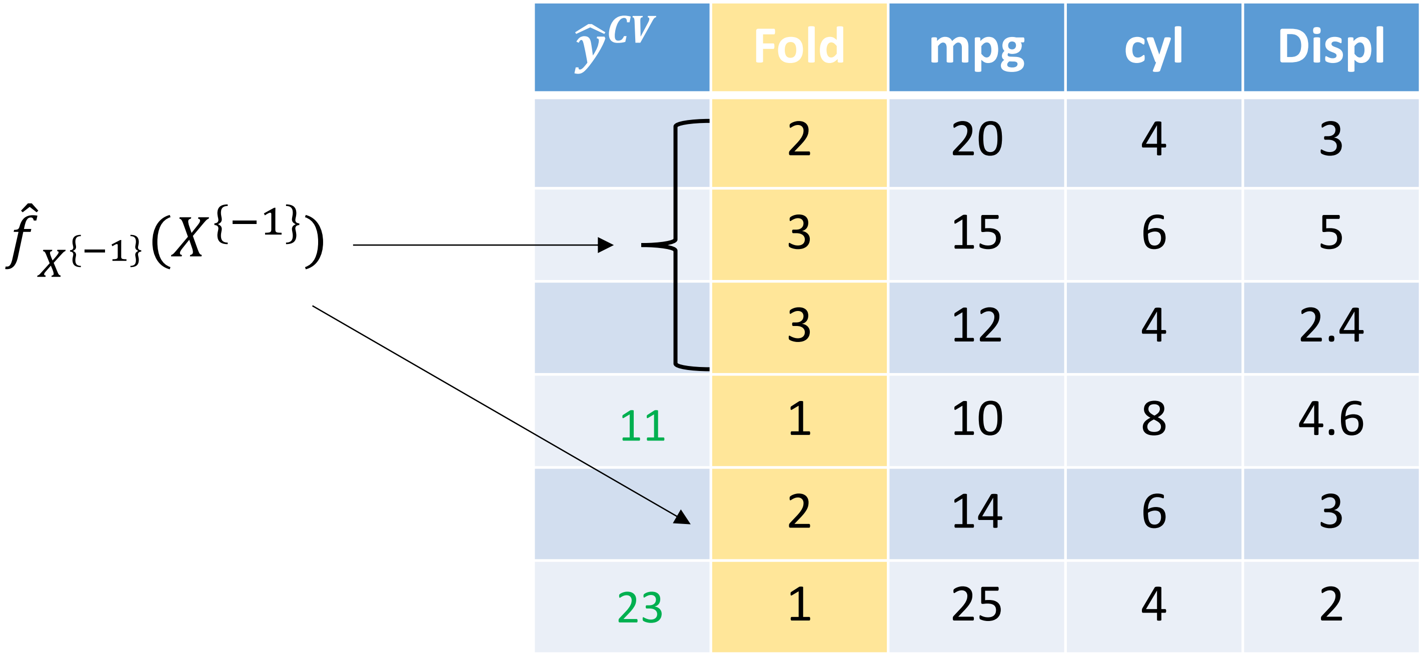
- K-Fold Cross-Validation in contrast randomly partitions the data into k folds.
- We start by estimating a model in folds 2,…,k, and using that predicted model to predict into fold 1.
- This process is repeated until all folds have predictions

$\hat{y}^{CV}$	mpg	cyl	Displ
	20	4	3
	15	6	5
	12	4	2.4
	10	8	4.6
	14	6	3
	25	4	2

Resampling: K-Fold Cross-Validation

- E.g. we set k = 3 and do 3-Fold Cross-Validation

- Fit a model using observations 1-3 and 5 (folds 2-3)
- Then use this model to predict into fold 1.



$\hat{f}_{X^{\{-1\}}}(X^{\{-1\}})$

$\hat{y}^{CV}$	Fold	mpg	cyl	Displ
	2	20	4	3
	3	15	6	5
	3	12	4	2.4
11	1	10	8	4.6
	2	14	6	3
23	1	25	4	2

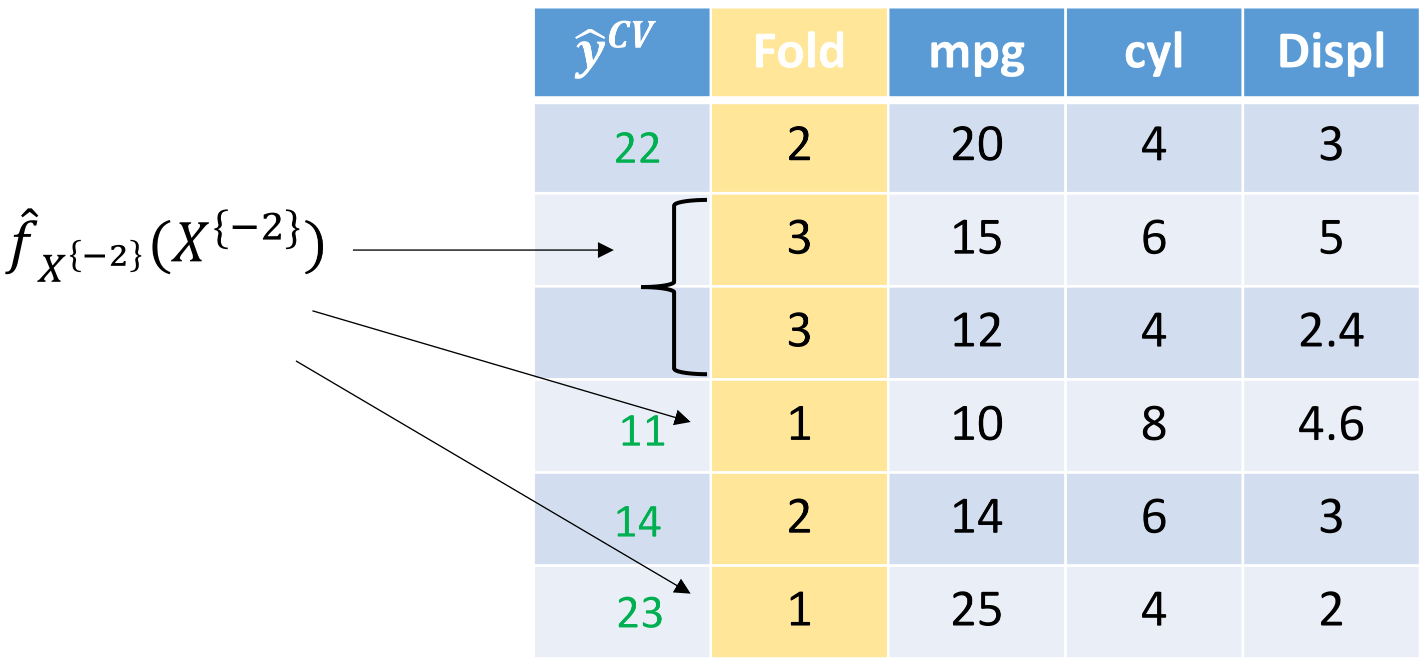
$X^{\{-1\}}$: X excluding fold 1

Resampling: K-Fold Cross-Validation

- E.g. we set k = 3 and do 3-Fold Cross-Validation

- Next we fit a a model using folds 1 & 3

- Then use this model to predict into fold 2.



$\hat{f}_{X^{\{-2\}}}(X^{\{-2\}})$

$\hat{y}^{CV}$	Fold	mpg	cyl	Displ
22	2	20	4	3
	3	15	6	5
	3	12	4	2.4
11	1	10	8	4.6
14	2	14	6	3
23	1	25	4	2

$X^{\{-3\}}$: X excluding fold 1

Resampling: K-Fold Cross-Validation

- E.g. we set k = 3 and do 3-Fold Cross-Validation

- Finally we predict using data in folds 1-2
- Then use this model to predict into fold 3.

$\hat{f}_{X^{\{-3\}}}(X^{\{-3\}})$

$\hat{y}^{CV}$	Fold	mpg	cyl	Displ
22	2	20	4	3
17	3	15	6	5
13	3	12	4	2.4
11	1	10	8	4.6
14	2	14	6	3
23	1	25	4	2

$X^{\{-3\}}$: X excluding fold 1

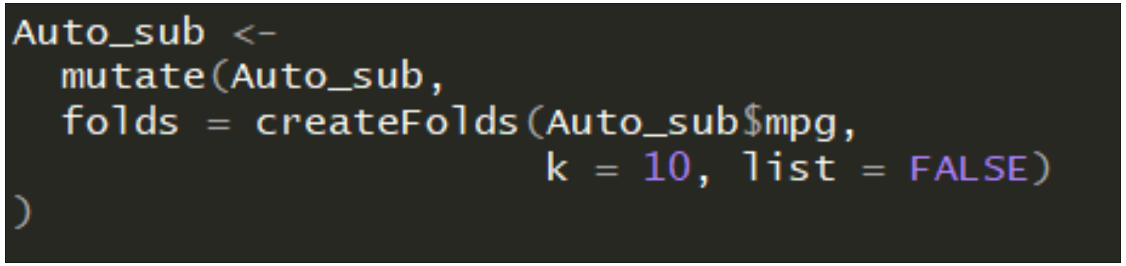
K-Fold Cross-Validation Error

- At the end we have a series of $\hat{y}^{CV}$.
- Again these were calculated using models that were trained on data excluding this observation
- We can then calculate MSE_i which is mean-squared-error in each fold
- We average these to get an average across-fold MSE

$\hat{y}^{CV}$	mpg	cyl	Displ
18	20	4	3
16	15	6	5
12	12	4	2.4
11	10	8	4.6
11	14	6	3
22	25	4	2

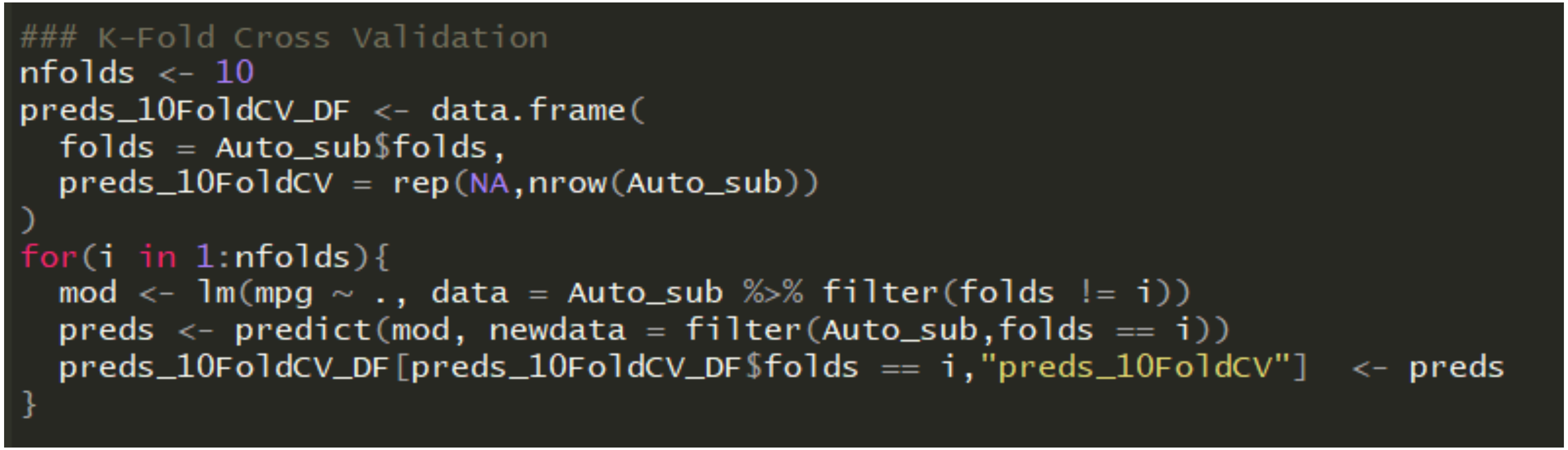
$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

K-Fold Cross-Validation in R



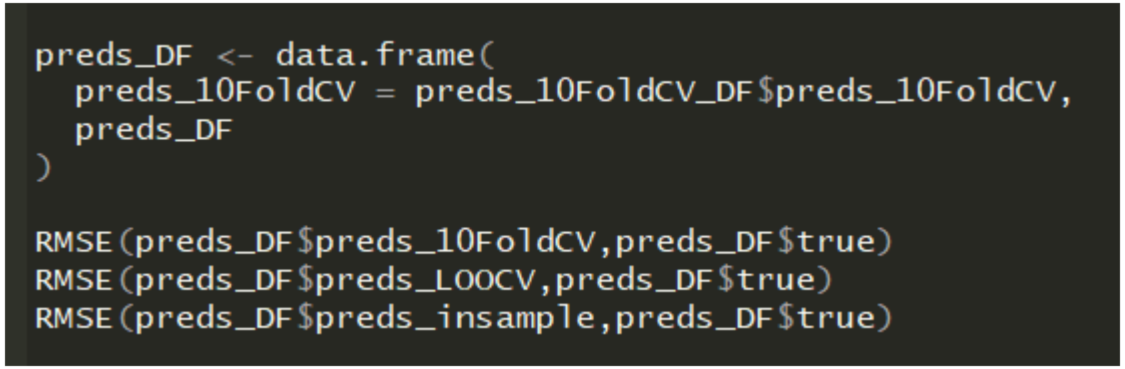
```
Auto_sub <-
  mutate(Auto_sub,
  folds = createFolds(Auto_sub$mpg,
                      k = 10, list = FALSE)
)
```

- Many ways to create folds, but the createFolds() package in `caret` is helpful here.



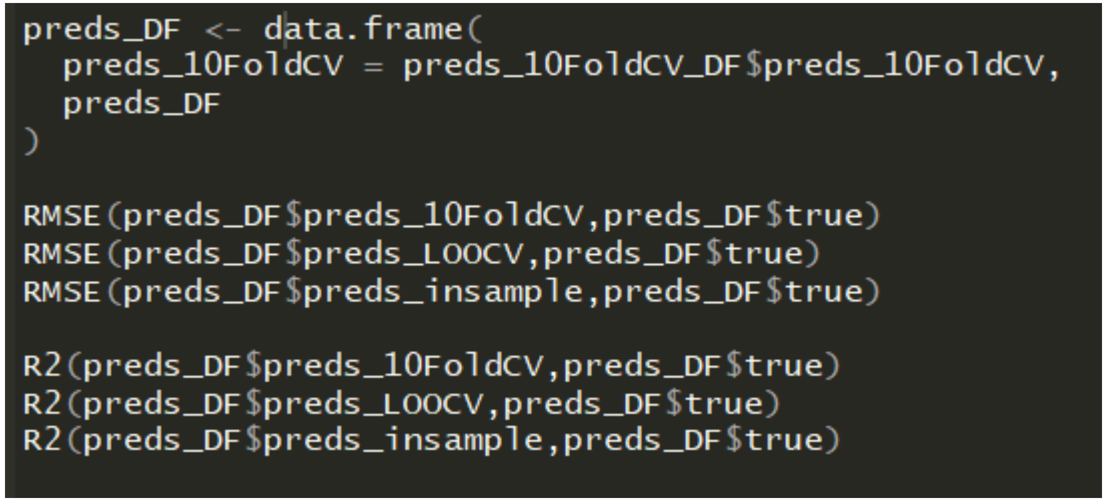
```
### K-Fold Cross Validation
nfolds <- 10
preds_10FoldCV_DF <- data.frame(
  folds = Auto_sub$folds,
  preds_10FoldCV = rep(NA,nrow(Auto_sub))
)
for(i in 1:nfolds){
  mod <- lm(mpg ~ ., data = Auto_sub %>% filter(folds != i))
  preds <- predict(mod, newdata = filter(Auto_sub,folds == i))
  preds_10FoldCV_DF[preds_10FoldCV_DF$folds == i,"preds_10FoldCV"]  <- preds
}
```

K-Fold Cross-Validation in R



```
preds_DF <- data.frame(
  preds_10FoldCV = preds_10FoldCV_DF$preds_10FoldCV,
  preds_DF
)

RMSE(preds_DF$preds_10FoldCV,preds_DF$true)
RMSE(preds_DF$preds_LOOCV,preds_DF$true)
RMSE(preds_DF$preds_insample,preds_DF$true)
```



```
preds_DF <- data.frame(
  preds_10FoldCV = preds_10FoldCV_DF$preds_10FoldCV,
  preds_DF
)

RMSE(preds_DF$preds_10FoldCV,preds_DF$true)
RMSE(preds_DF$preds_LOOCV,preds_DF$true)
RMSE(preds_DF$preds_insample,preds_DF$true)

R2(preds_DF$preds_10FoldCV,preds_DF$true)
R2(preds_DF$preds_LOOCV,preds_DF$true)
R2(preds_DF$preds_insample,preds_DF$true)
```

	RMSE (pred vs true)	R2 (pred vs true)
In-Sample	3.29	0.82
LOOCV	3.37	0.812
K-Fold CV	3.37	0.812

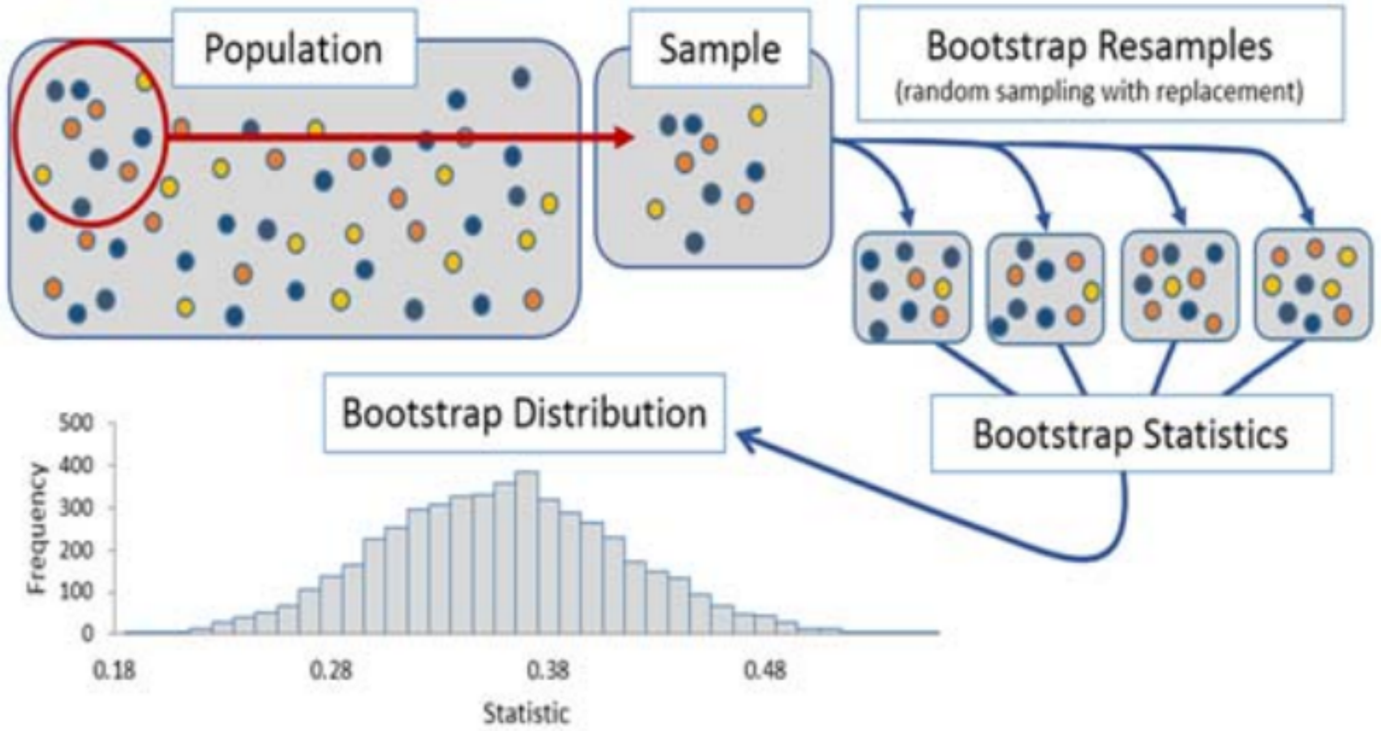
BUS 696: Class 6 Outline

1. AI in the News
2. Review
3. Lift Charts
4. Calibration Plots
5. Severe Class Imbalance Issues
6. Leave-One-Out Cross-Validation
7. K-Fold Cross-Validation
8. **Bootstrapping**

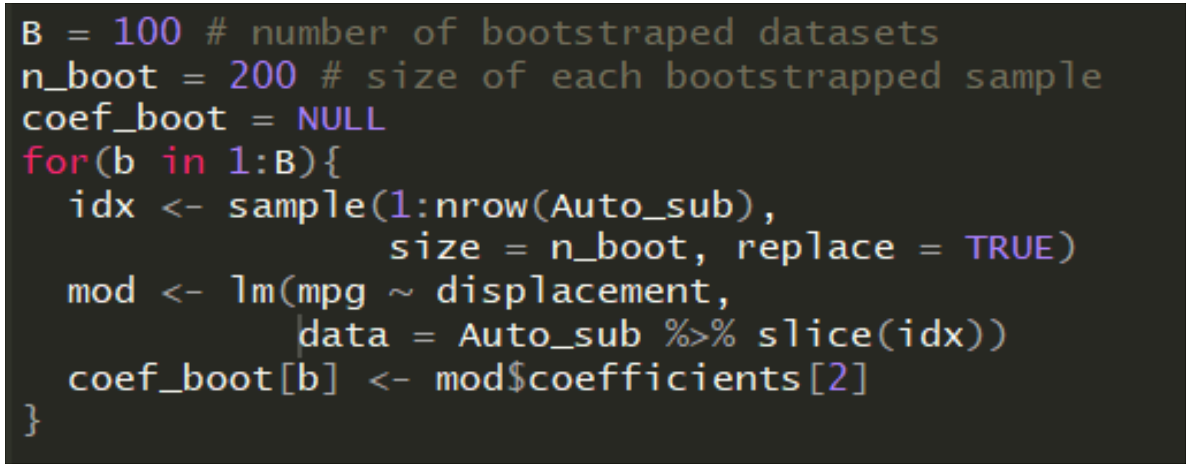
Bootstrap

- The idea of the bootstrap is we take the original data (which is itself a sample from some population of possible data) and generate B bootstrap resamples.

- To do that we sample with replacement the original dataset until we have B bootstrap datasets, each of size n_b



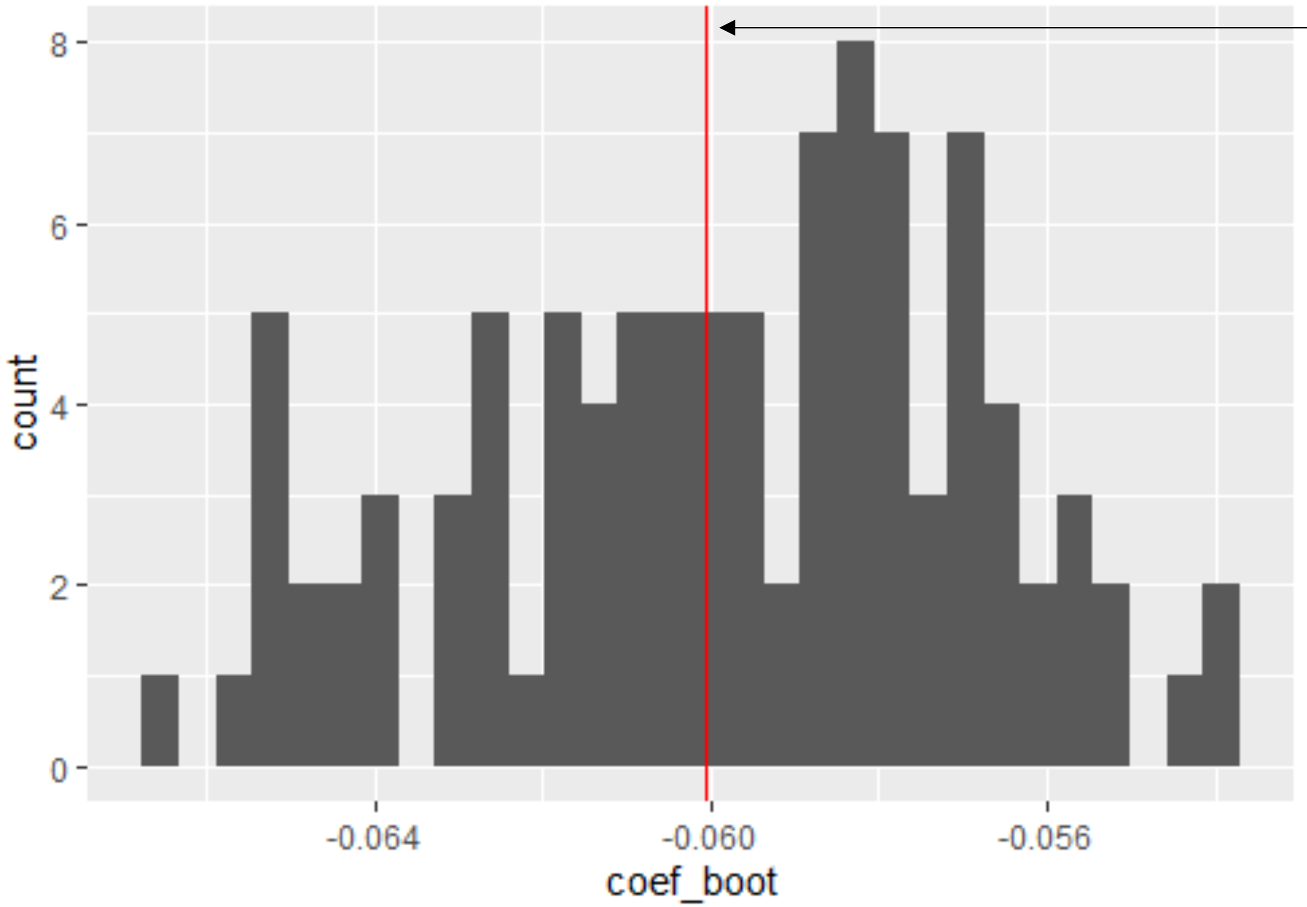
Bootstrapping in R



```
B = 100 # number of bootstraped datasets
n_boot = 200 # size of each bootstrapped sample
coef_boot = NULL
for(b in 1:B){
  idx <- sample(1:nrow(Auto_sub),
                size = n_boot, replace = TRUE)
  mod <- lm(mpg ~ displacement,
            data = Auto_sub %>% slice(idx))
  coef_boot[b] <- mod$coefficients[2]
}
```

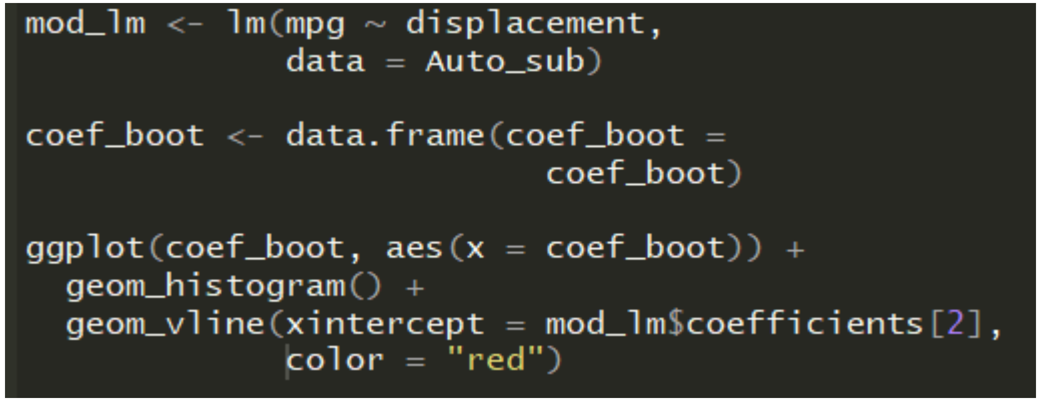
- Again, many ways to do it. First we do it by hand.

Bootstrapping in R



Linear model coefficient on original sample

Each point shows a coefficient from a different bootstrapped sample



```
mod_lm <- lm(mpg ~ displacement,
             data = Auto_sub)

coef_boot <- data.frame(coef_boot =
                          coef_boot)

ggplot(coef_boot, aes(x = coef_boot)) +
  geom_histogram() +
  geom_vline(xintercept = mod_lm$coefficients[2],
             color = "red")
```