MODEL ASSESSMENT

Dr. Aric LaBarr
Institute for Advanced Analytics

COMPARING MODELS

Purpose of Modeling

- Statistical models are created for two different purposes – estimation and prediction.
  - **Estimation:** Quantifying the expected change in response associated with predictors (relationships).
  - **Prediction:** Use the model to predict new response.

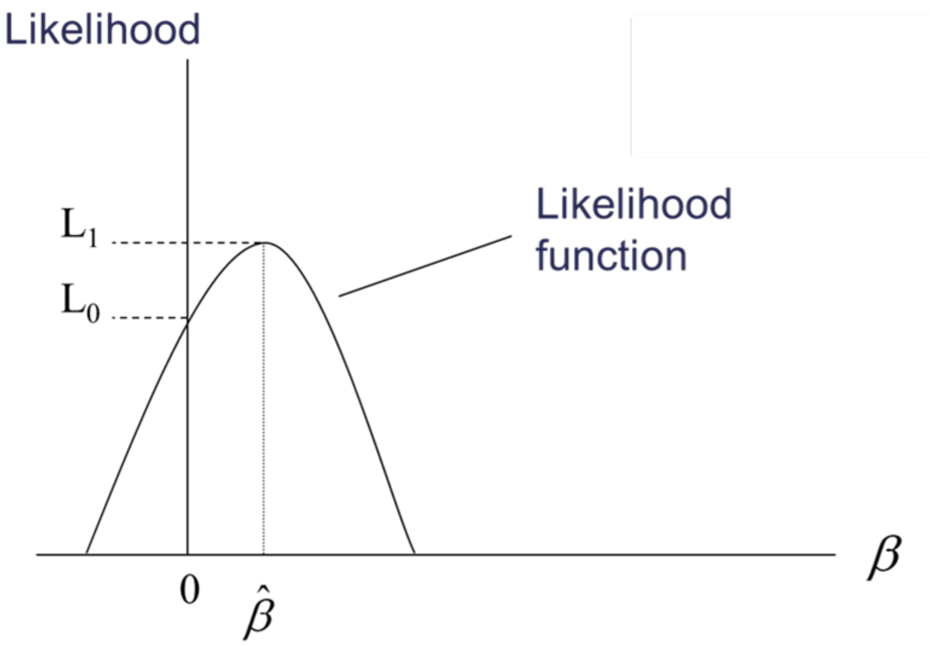
- Won’t necessarily agree!

Deviance/Likelihood Measures

- AIC and BIC approximate out-of-sample prediction error by applying a penalty for model complexity:
  - AIC – crude, large-sample approximation of leave-one-out cross-validation.
  - BIC – favors smaller models/penalizes model complexity more.
- Lower values “better” than higher.
- No amount of lower is “better” enough.
- May not always agree, but neither is necessarily better.

Deviance/Likelihood Measures

- Number of "pseudo"-R^2 quantities for logistic regression.
- Higher values indicate "better" model.
- Generalized / Nagelkerke R^2 - how much better than intercept only model?
- Unlike linear regression, there is **no interpretation** on these.



$$R_G^2 = 1 - \left(\frac{L_0}{L_1}\right)^{\frac{2}{n}}$$

Deviance and Likelihood Measures

```
AIC(logit.model)

[1] 1287.964

BIC(logit.model)

[1] 1394.86

PseudoR2(logit.model, which = "Nagelkerke")

Nagelkerke 0.7075796
```

ASSESSING PREDICTIVE POWER

What is a Good Logistic Model?

- Logistic regression is a **model for probability of an event** – NOT the occurrence of an event.
- Logistic regression **can** be a classification model as well.

- Good model should reflect both of these, but importance of one over the other depends on the problem.

Discrimination vs. Calibration

- **Discrimination** – ability to separate the events from the non-events. How good is model at distinguishing the 1's from the 0's.
- **Calibration** – how well predicted probabilities agree with the actual frequency of the outcomes. Are predicted probabilities systematically too low/high?

- **May not agree with each other!**

ASSESSING PREDICTIVE POWER

Probability Based Metrics

Coefficient of Discrimination

- Want model to assign a higher probability to events and lower probability to non-events.
- **Coefficient of discrimination** (or **discrimination slope**) is the difference in average predicted probability between 1's and 0's:

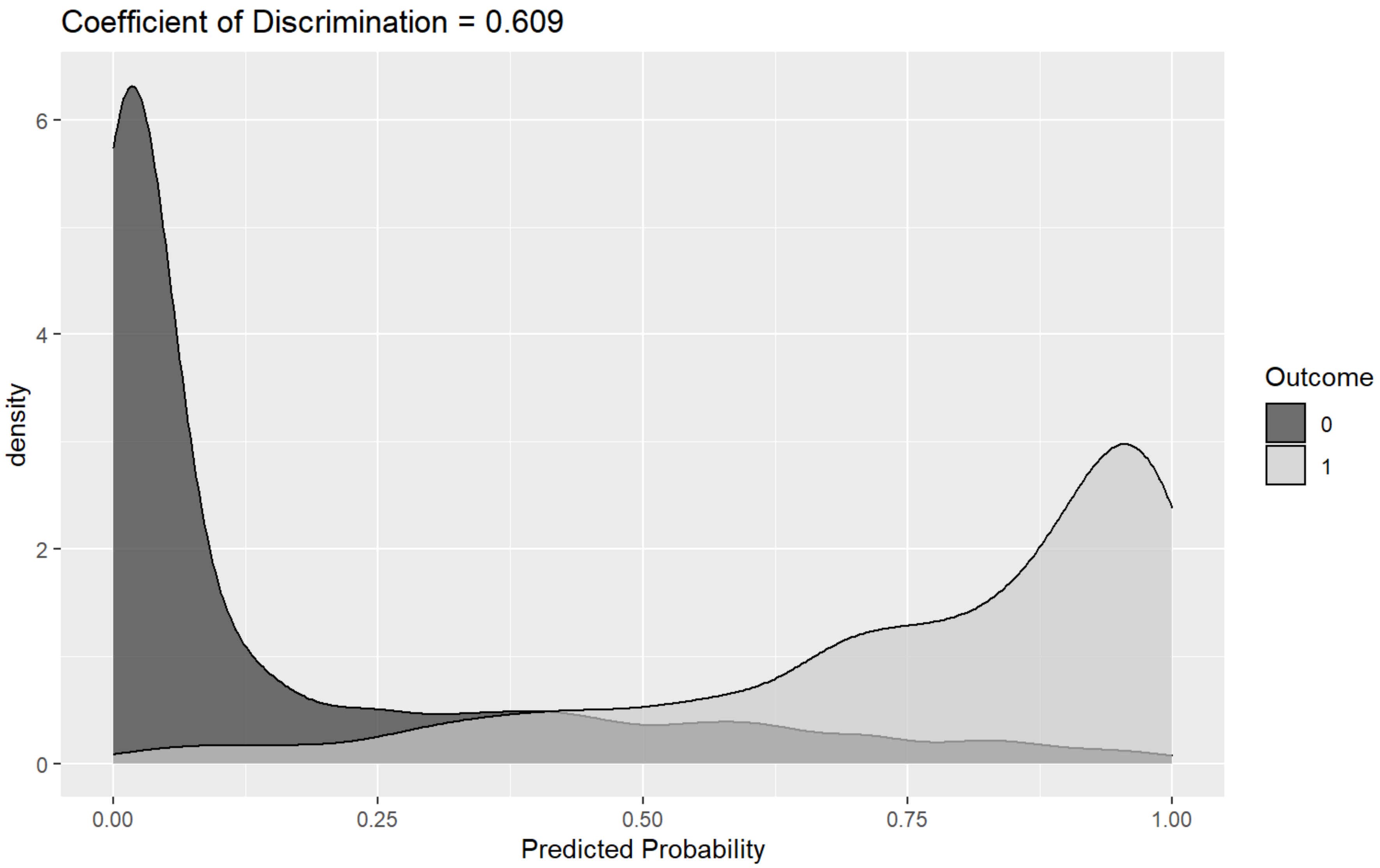
$$D = \bar{\hat{p}}_1 - \bar{\hat{p}}_0$$

- Able to compare with histograms as well.

Discrimination Slope

```
train$p_hat <- predict(logit.model, type = "response")
p1 <- train$p_hat[train$Bonus == 1]
p0 <- train$p_hat[train$Bonus == 0]

coef_discrim <- mean(p1) - mean(p0)

ggplot(train, aes(p_hat, fill = factor(Bonus)))
 + geom_density(alpha = 0.7)
 + scale_fill_grey()
 + labs(x = "Predicted Probability", fill = "Outcome",
        title = paste("Coefficient of Discrimination = ",
        round(coef_discrim, 3), sep = ""))
```

Discrimination Slope



Rank-order Statistics

- How well does the model order predictions?

- **Concordance:** for a pair of subjects with and without the event, the one **with the event** had the **higher** predicted probability.
- **Discordance:** for a pair of subjects with and without the event, the one **with the event** had the **lower** predicted probability.
- **Tied:** for a pair of subjects with and without the event, they both have the **same** predicted probability.

Concordance

- **Interpretation** – For all possible (1,0) pairs, the model assigned the higher predicted probability to the observation with the event *concordance%* of the time.
- Common metrics based on concordance:
  - c-statistic: $c = Concordance\ \% + \frac{1}{2} Tied\ \%$
  - Somers' D (Gini): $D_{xy} = 2c - 1$
  - Kendall's τ_a: $\tau_a = \frac{\#\text{concordant} - \#\text{discordant}}{\frac{n(n-1)}{2}}$

Rank-order Statistics – R

```
library(Hmisc)

somers2(train$p_hat, train$Bonus)

        C         Dxy          n    Missing
0.9428394   0.8856789       2051          0
```

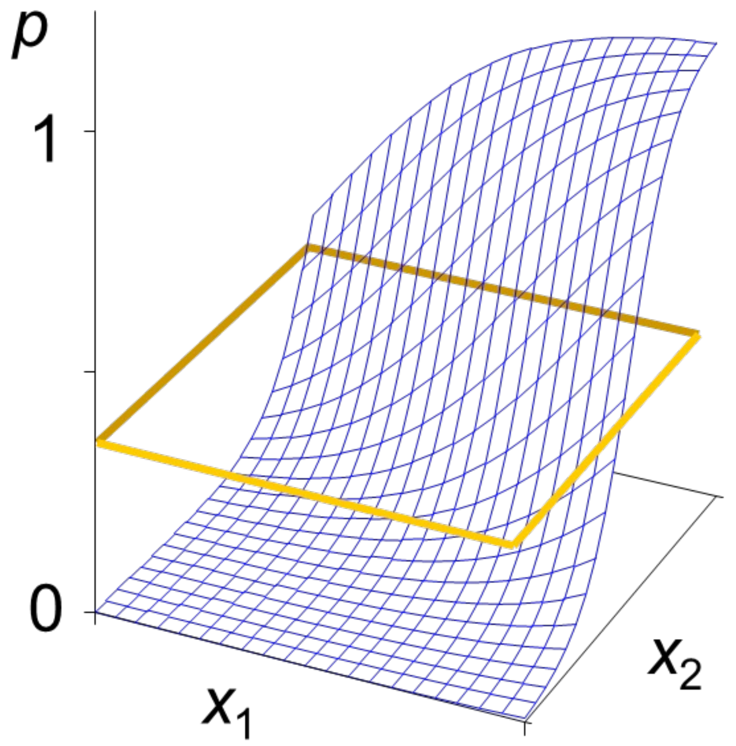
ASSESSING PREDICTIVE POWER

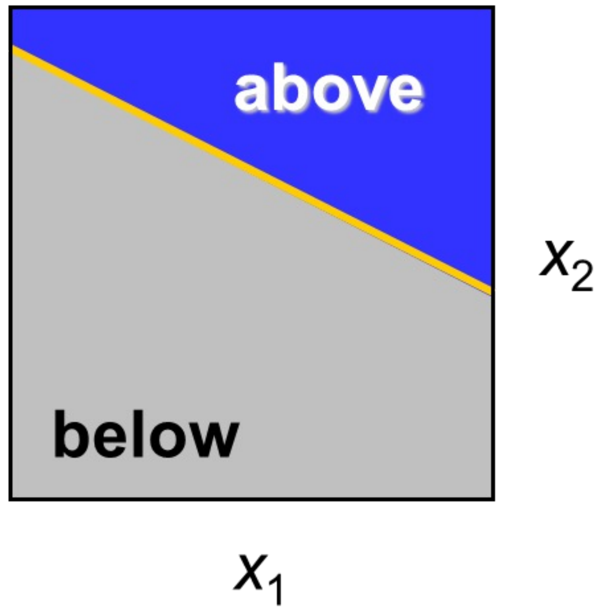
Classification Based Metrics

Classification

- Want model to correctly classify events and non-events.
- **Classification** forces the model to predict $\hat{y}_i = 1$ or $\hat{y}_i = 0$ based on whether the predicted probability exceeds some threshold – for example, $\hat{y}_i = 1$ if $\hat{p}_i > 0.5$.
- Strict classification-based measures completely discard any information about the actual quality of the model's predicted probabilities.

Logistic Discrimination





Classification Table

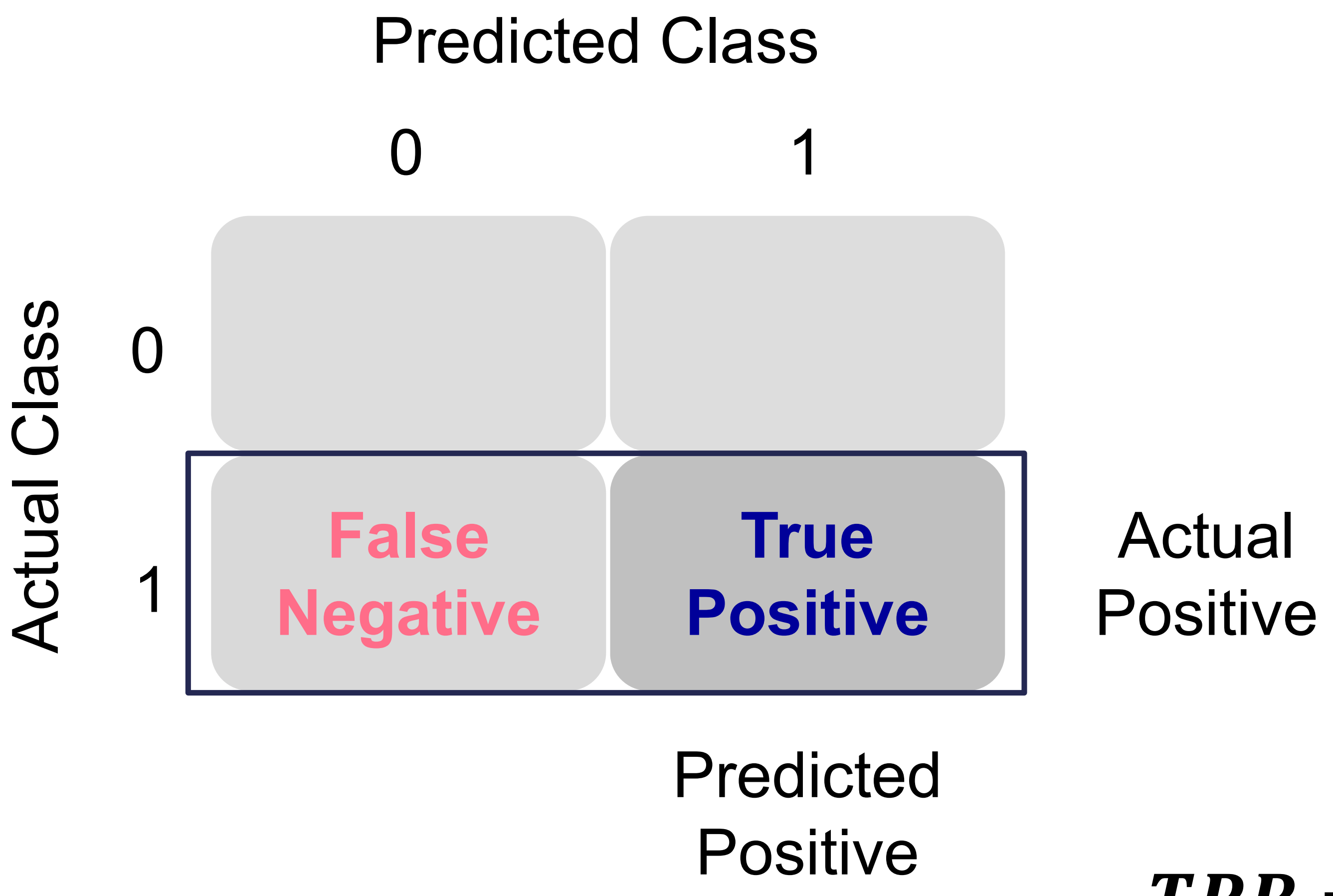
Predicted Class

		0	1	
Actual Class	0	**True Negative**	**False Positive**	Actual Negative
	1	**False Negative**	**True Positive**	Actual Positive
		Predicted Negative	Predicted Positive	

ASSESSING PREDICTIVE POWER

Sensitivity vs. Specificity

Sensitivity / Recall



$$TPR = \frac{TP}{TP + FN}$$

Specificity

Predicted Class

Actual Class	0	1	
0	**True Negative**	**False Positive**	Actual Negative
1			
	Predicted Negative		

$$TNR = \frac{TN}{TN + FP}$$

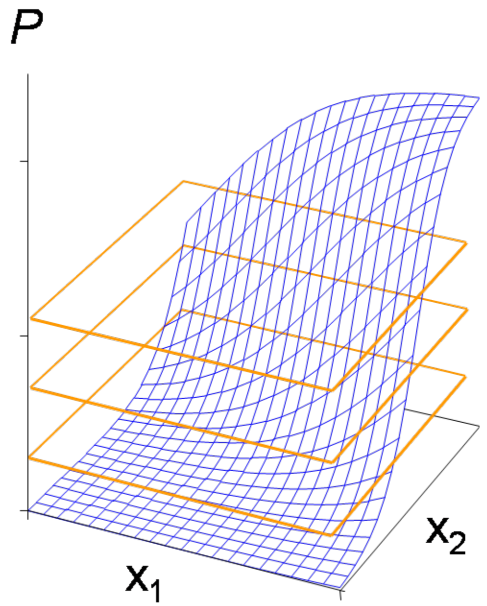
1 – Specificity

Predicted Class

Actual Class		0	1	
	0	True Negative	False Positive	Actual Negative
	1			
		Predicted Negative		

$$FPR = \frac{FP}{TN + FP}$$

Classification Changes with Cut-off



response	$\hat{P}$	cutoff=.5	cutoff=.25
0	.32	0	1
1	.40	0	1
1	.92	1	1
0	.06	0	0
1	.52	1	1
1	.39	0	1
1	.22	0	0
0	.17	0	0
0	.13	0	0
⋮	⋮	⋮	⋮
1	.75	1	1

Best Cut-off?

- **Always** consider the cost of false positives and false negatives when doing classification.
- When **NOT** considering costs, many different techniques to “optimal” cut-off.
- **Youden J statistic** (or **Youden’s index**):

$$J = \text{sensitivity} + \text{specificity} - 1$$

- “Optimal” – false positives and false negatives are weighed equally, so select cut-off that produces highest Youden *J* statistic.

Classification Table

```
train <- train %>%
  mutate(Bonus_hat = ifelse(p_hat > 0.5, 1, 0))

table(train$Bonus_hat, train$Bonus)

     0   1
0 1062 127
1  149 713
```

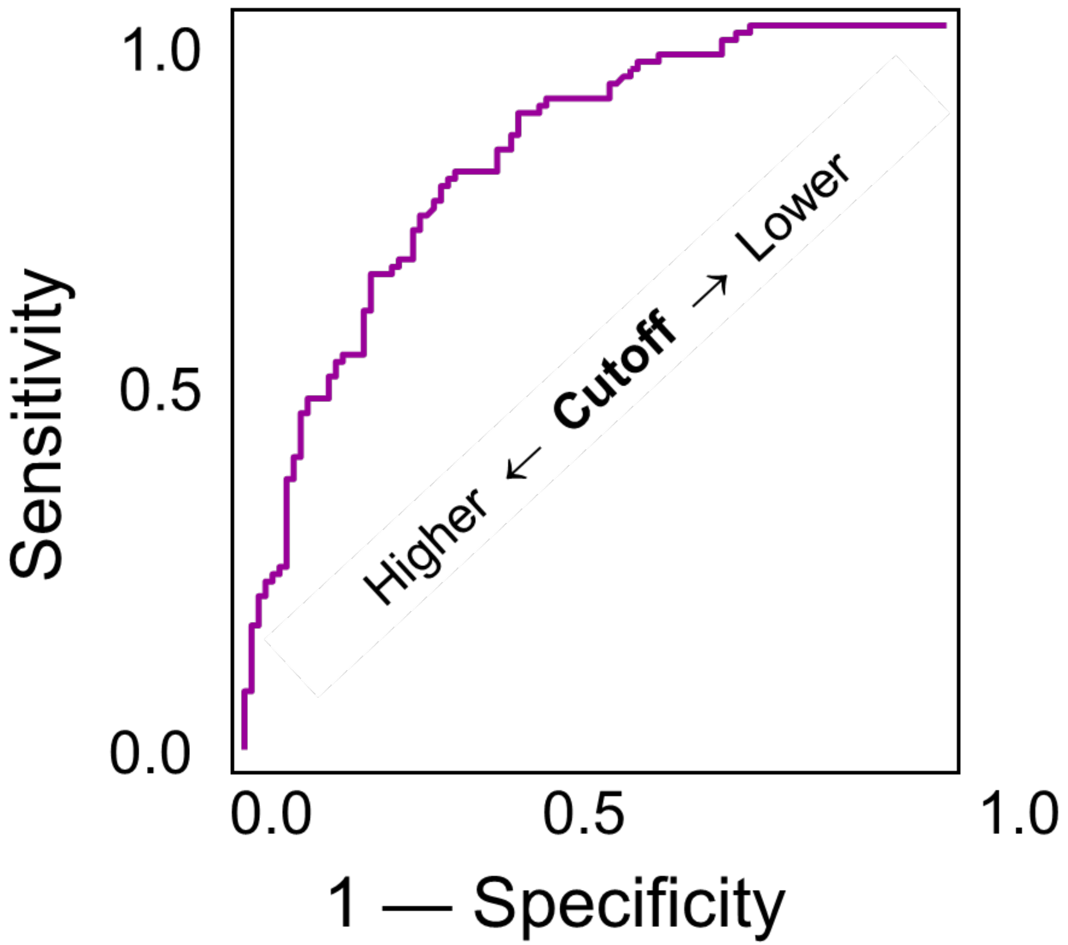
Youden Index

```
library(ROCit)

logit_meas <- measureit(train$p_hat, train$Bonus, measure = c("ACC", "SENS",
"SPEC"))

print(logit_meas)
```

Prints out metrics (Cutoff, Depth, TP, FP, TN, FN, and ones listed above) for every possible cut-off!

Output not shown here.

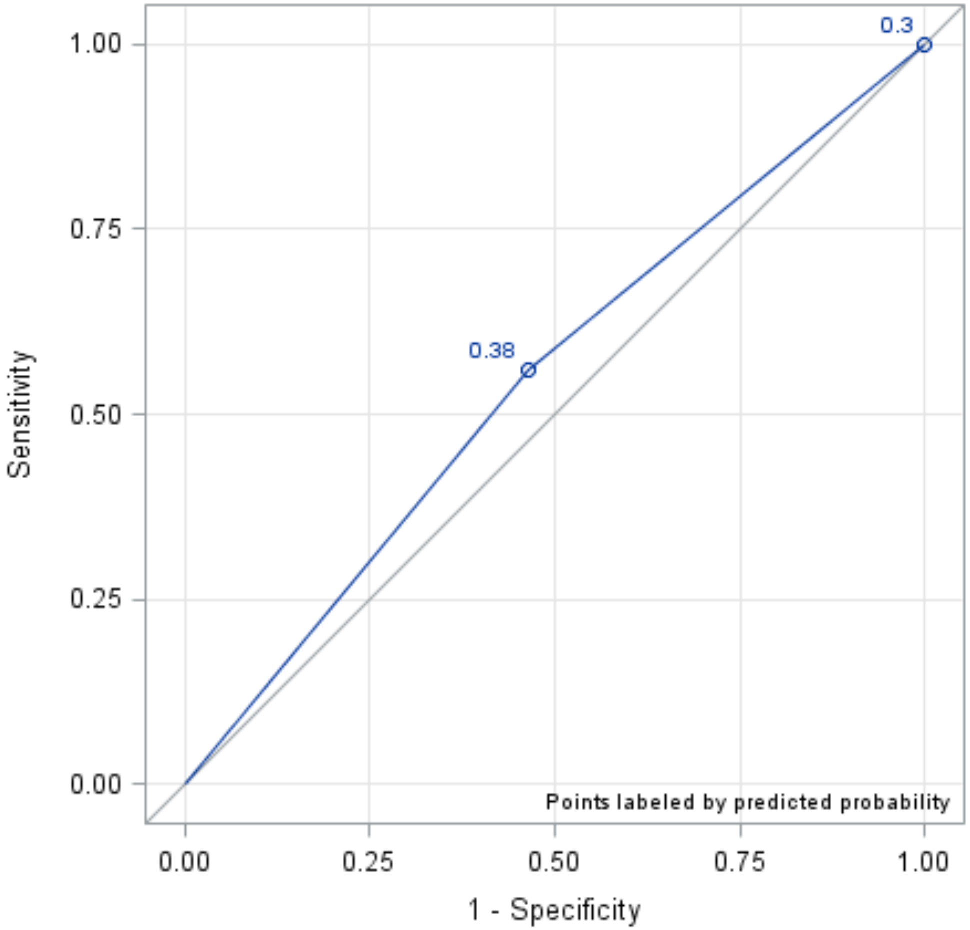
ROC Curve



- **ROC curve** plots *TPR* vs. *FPR* for a grid of thresholds.
- **Area under the curve** (AUC or AUROC) summarizes the overall quality of ROC curve – equivalent to c-statistic.
- Want high sensitivity and high specificity.

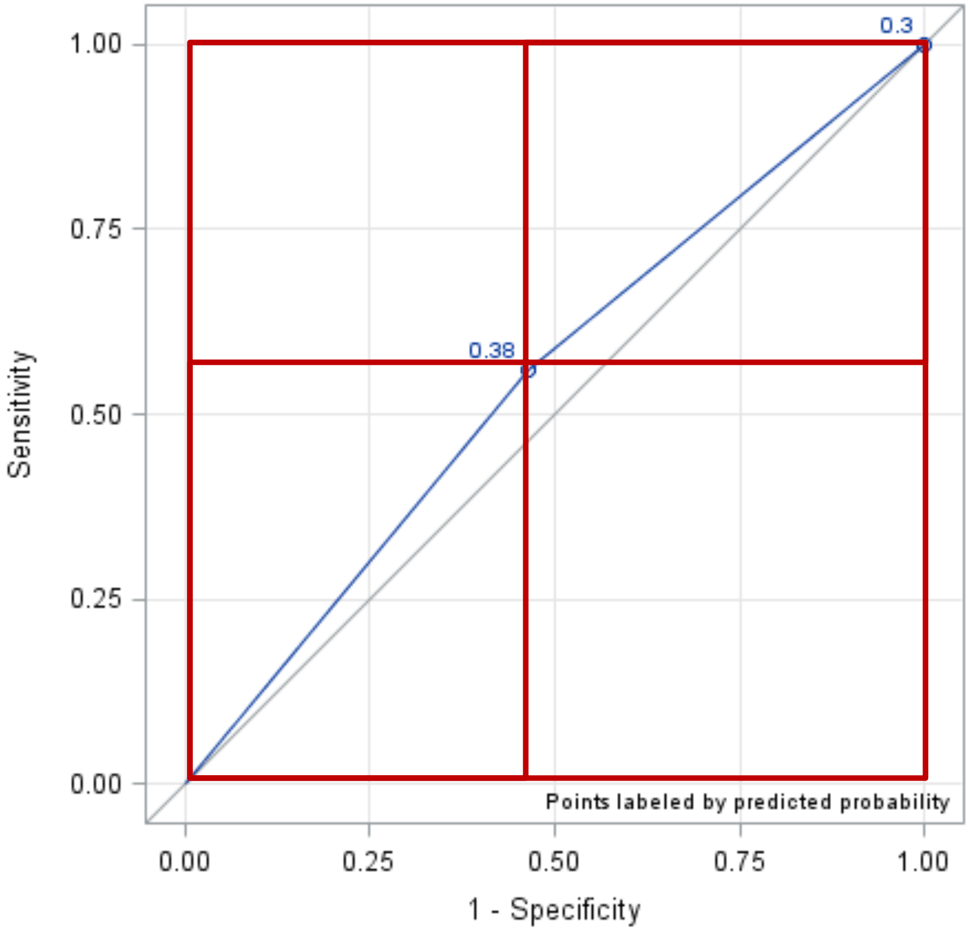
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



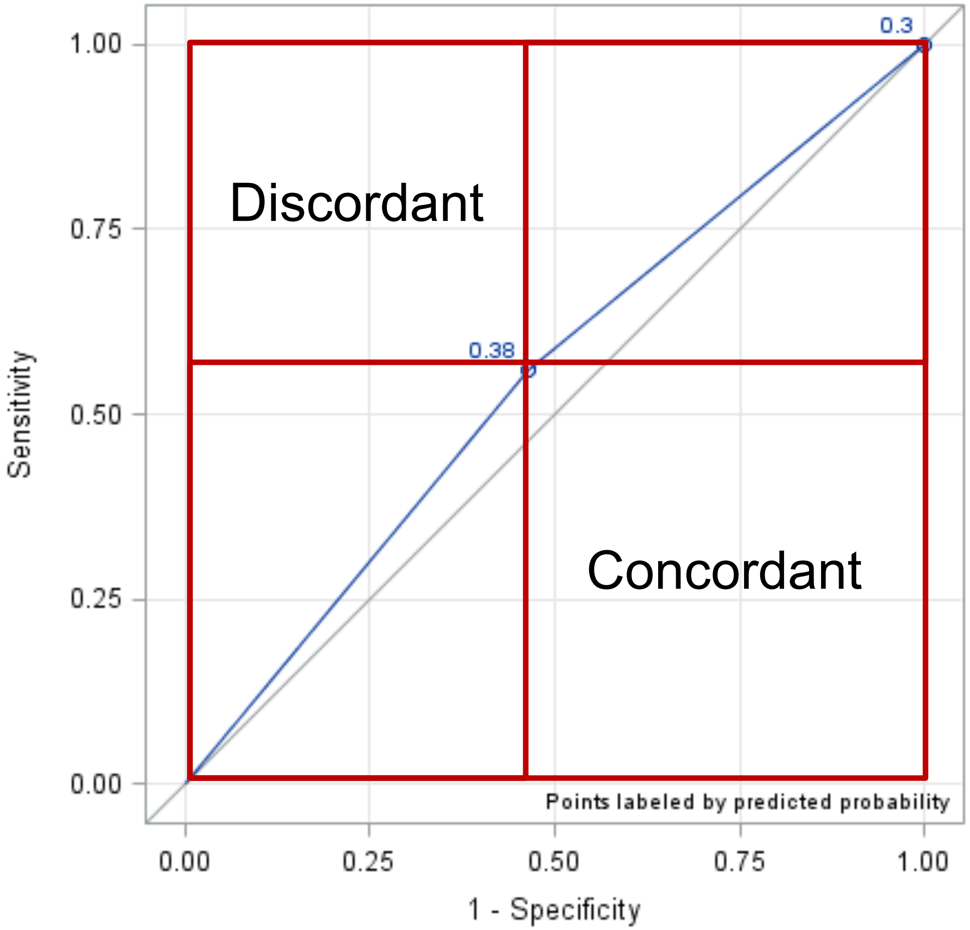
Area Under the ROC Curve

$$AUC = \%\, Concordant + \frac{1}{2}(\%\, Tied)$$



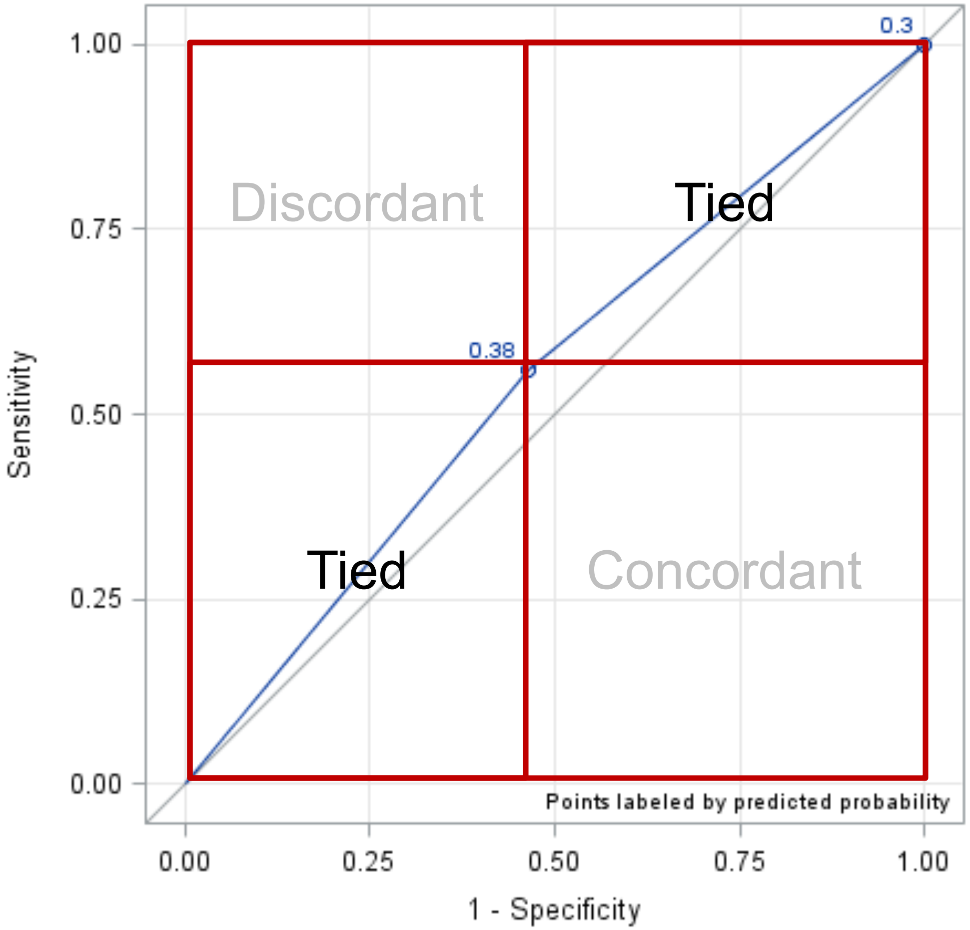
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



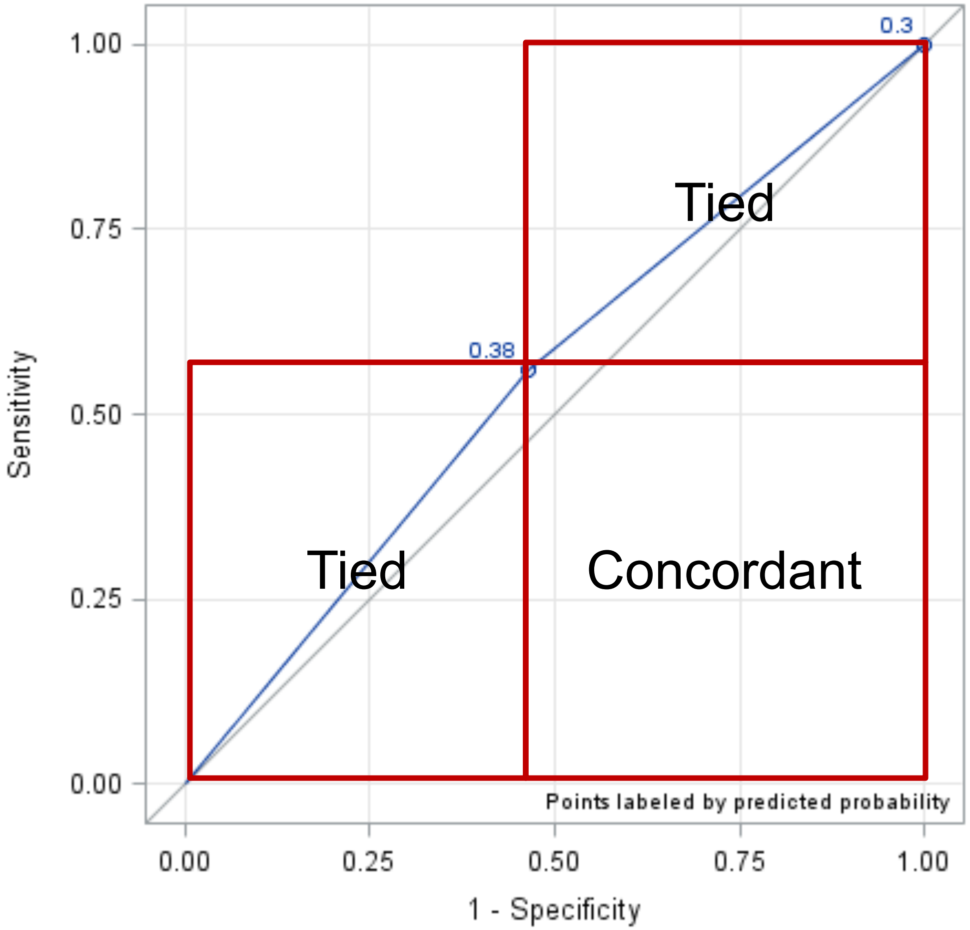
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



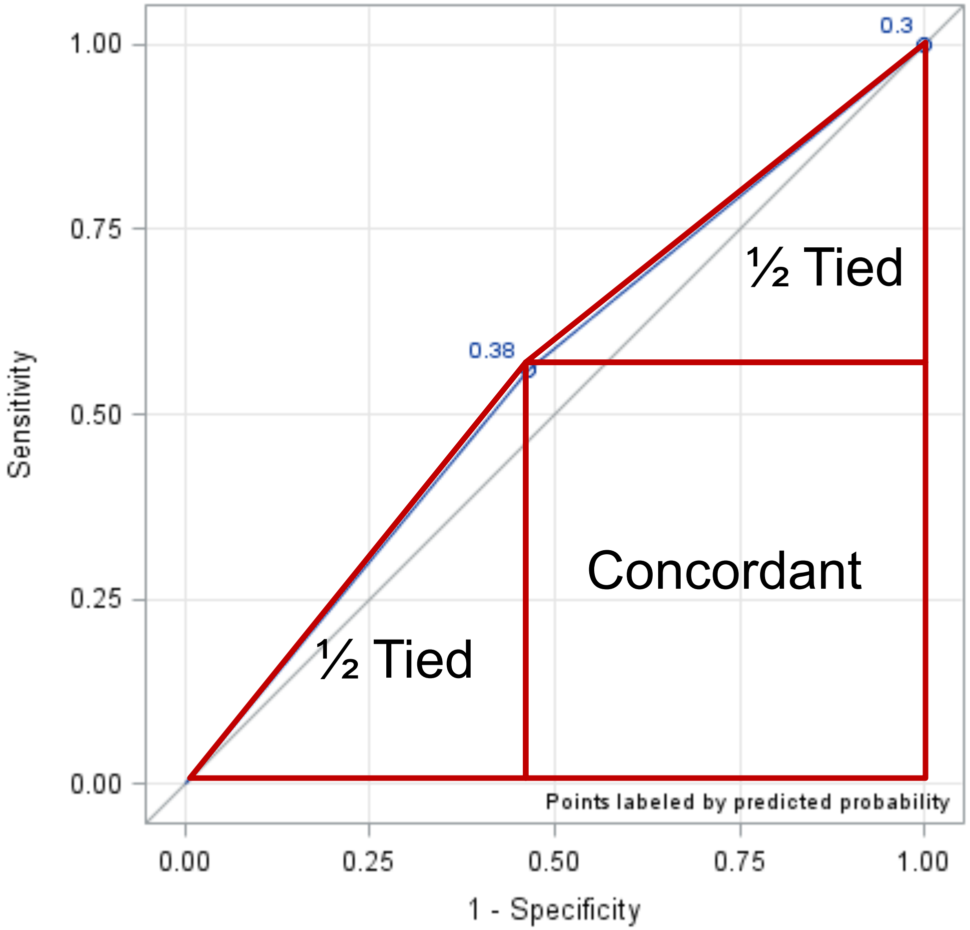
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



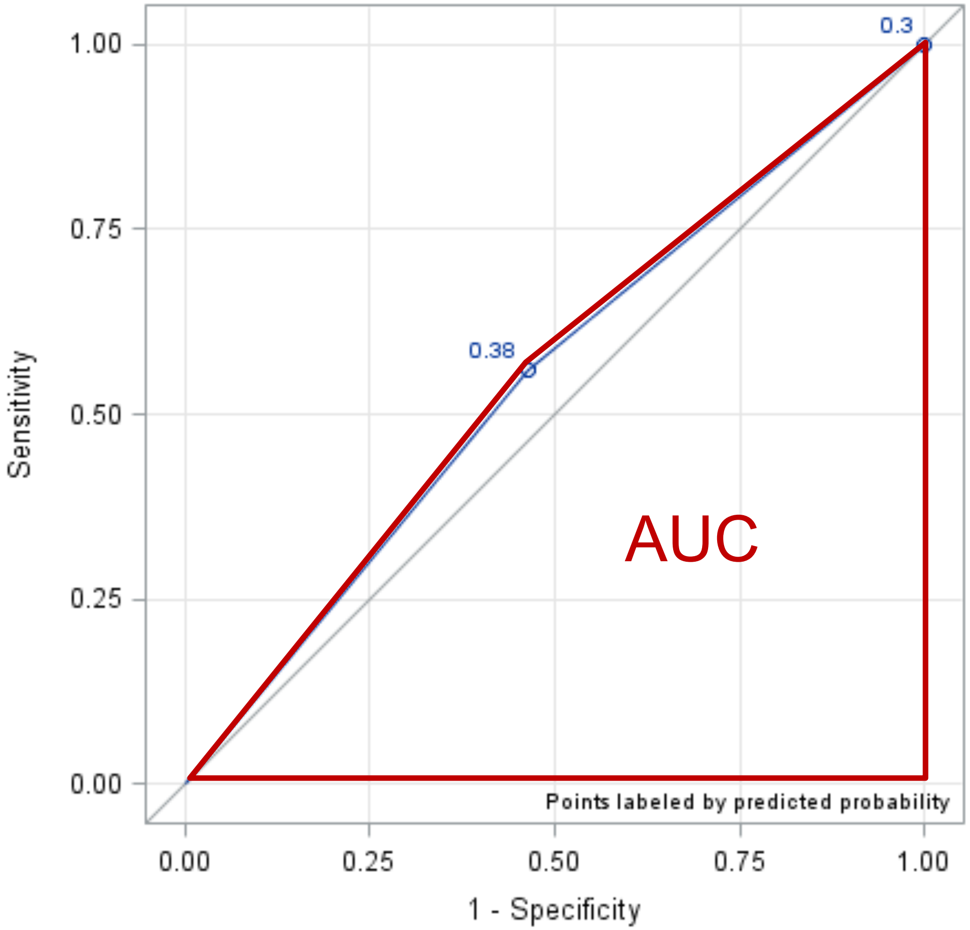
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



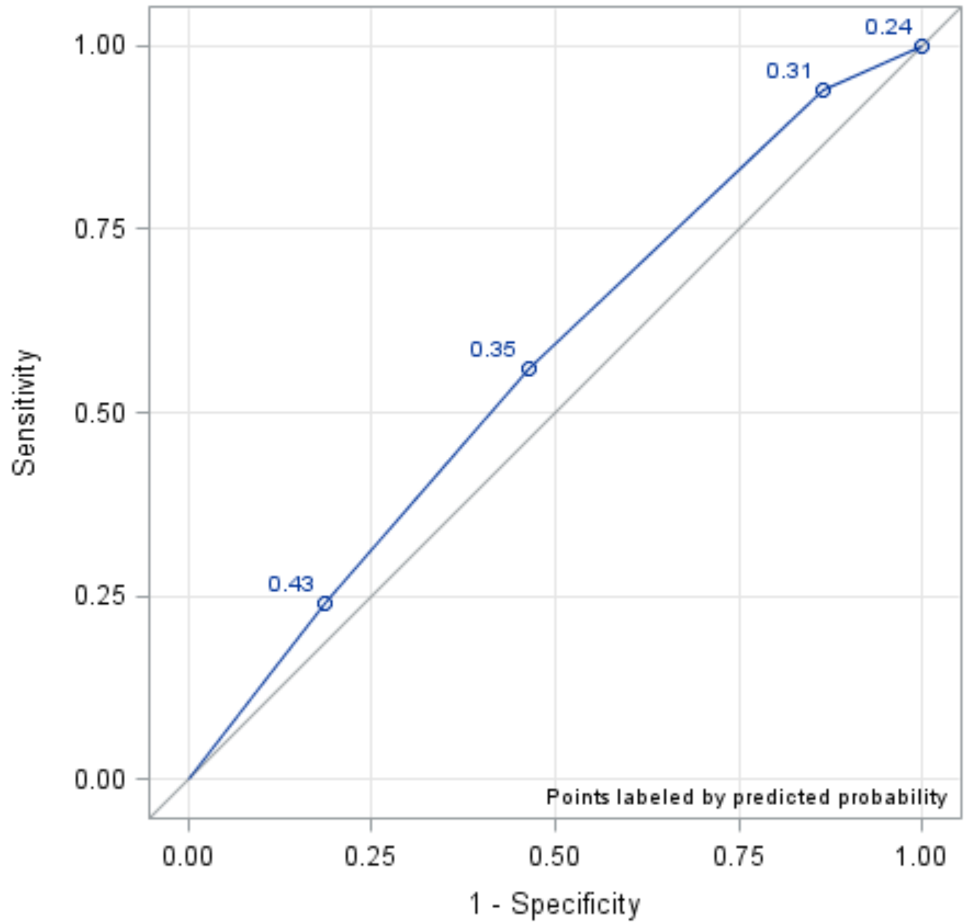
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



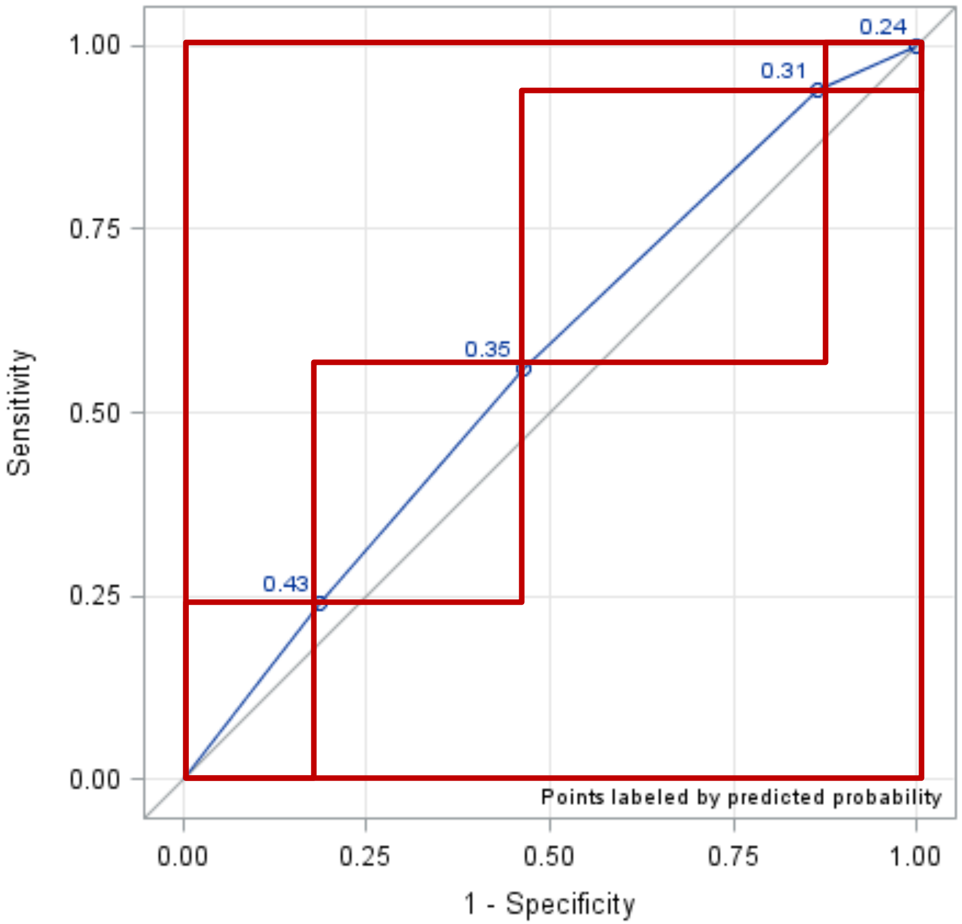
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



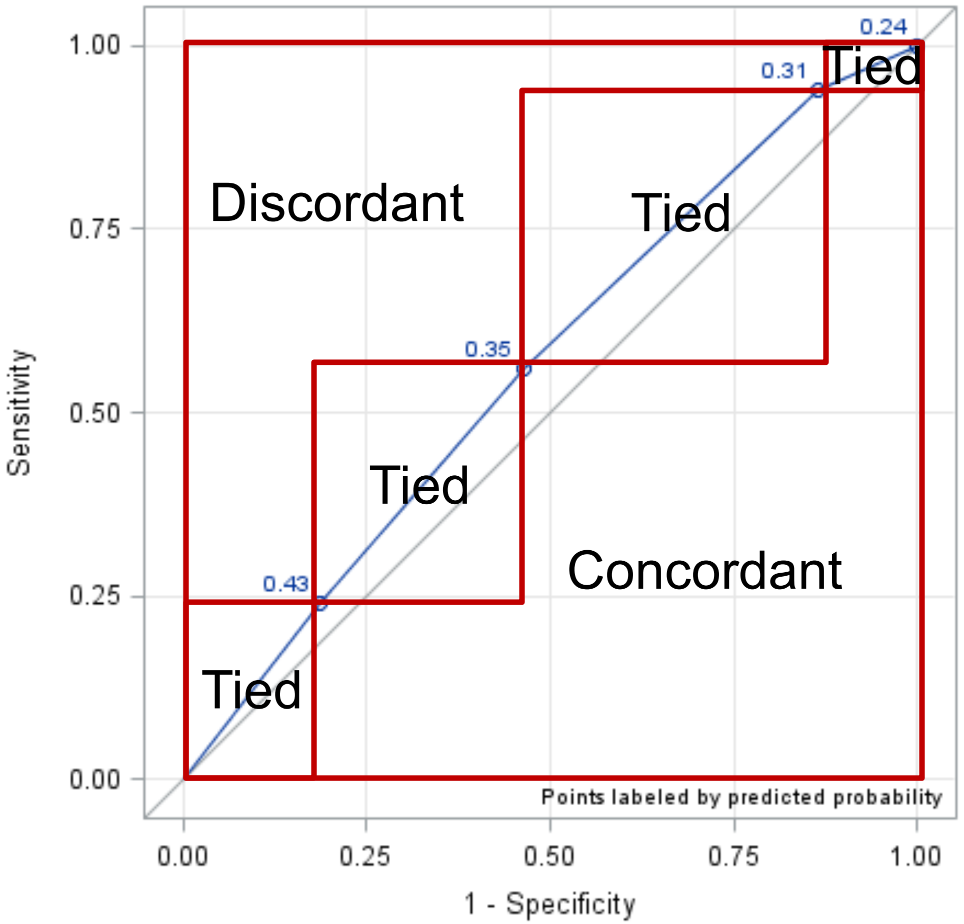
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



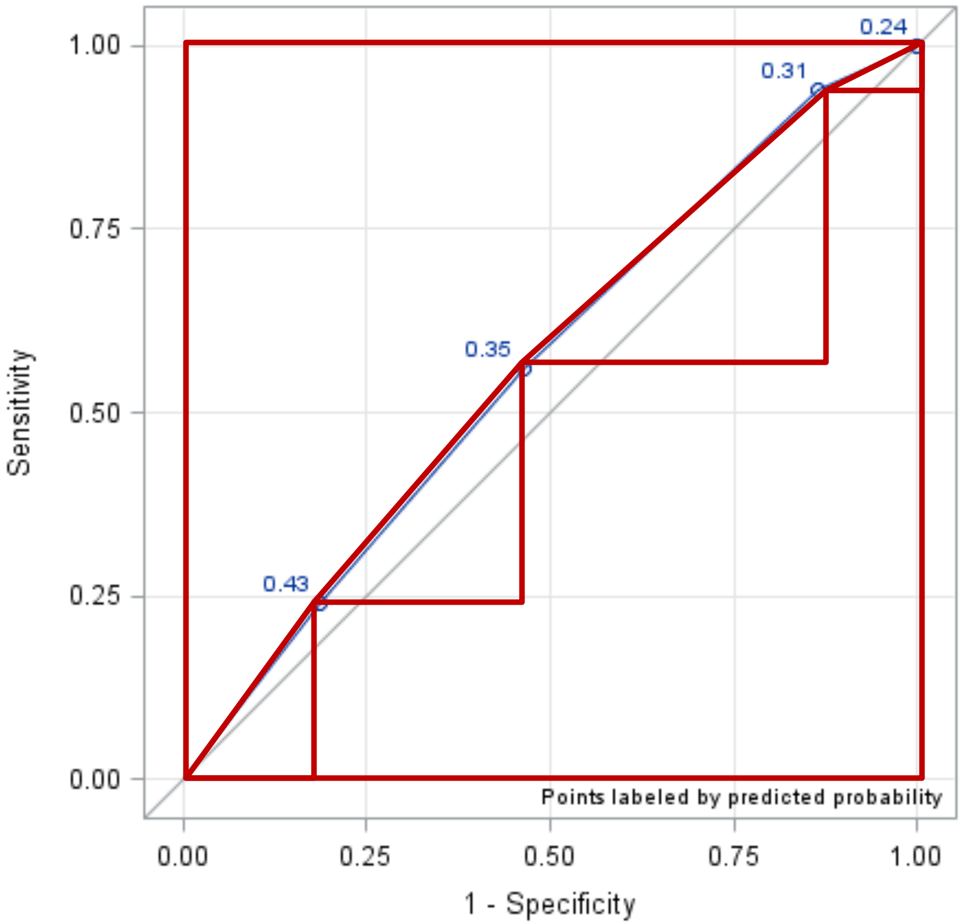
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



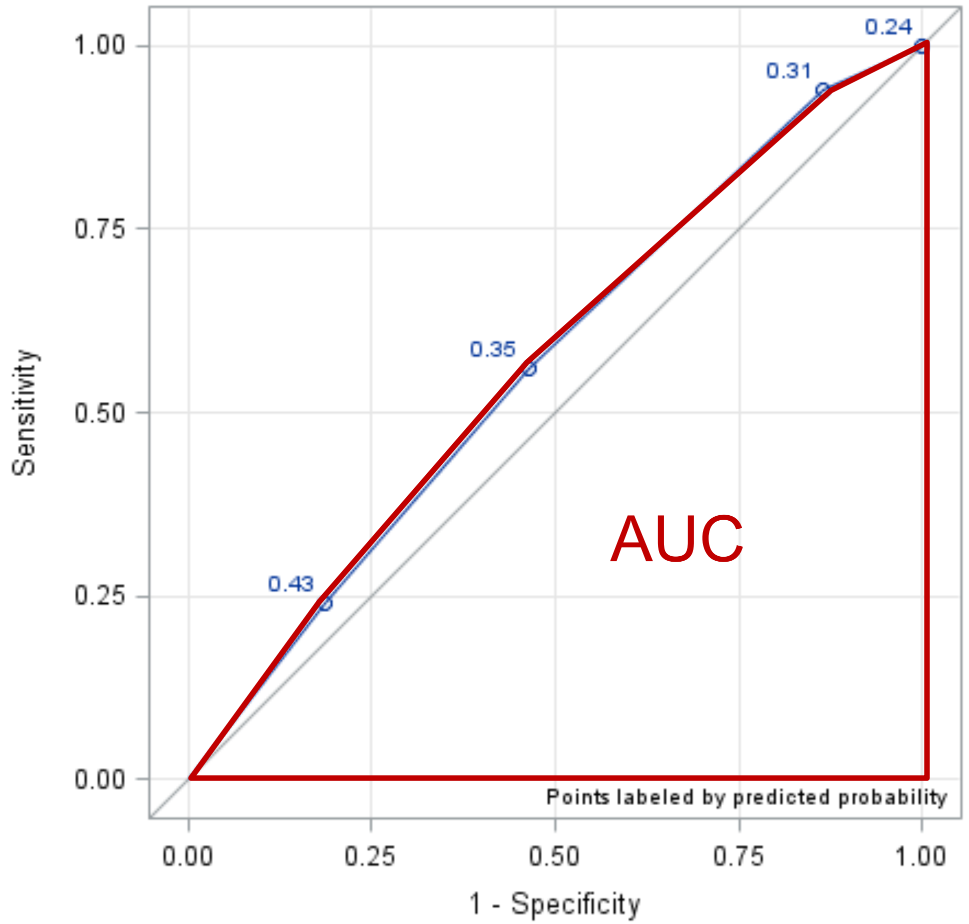
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



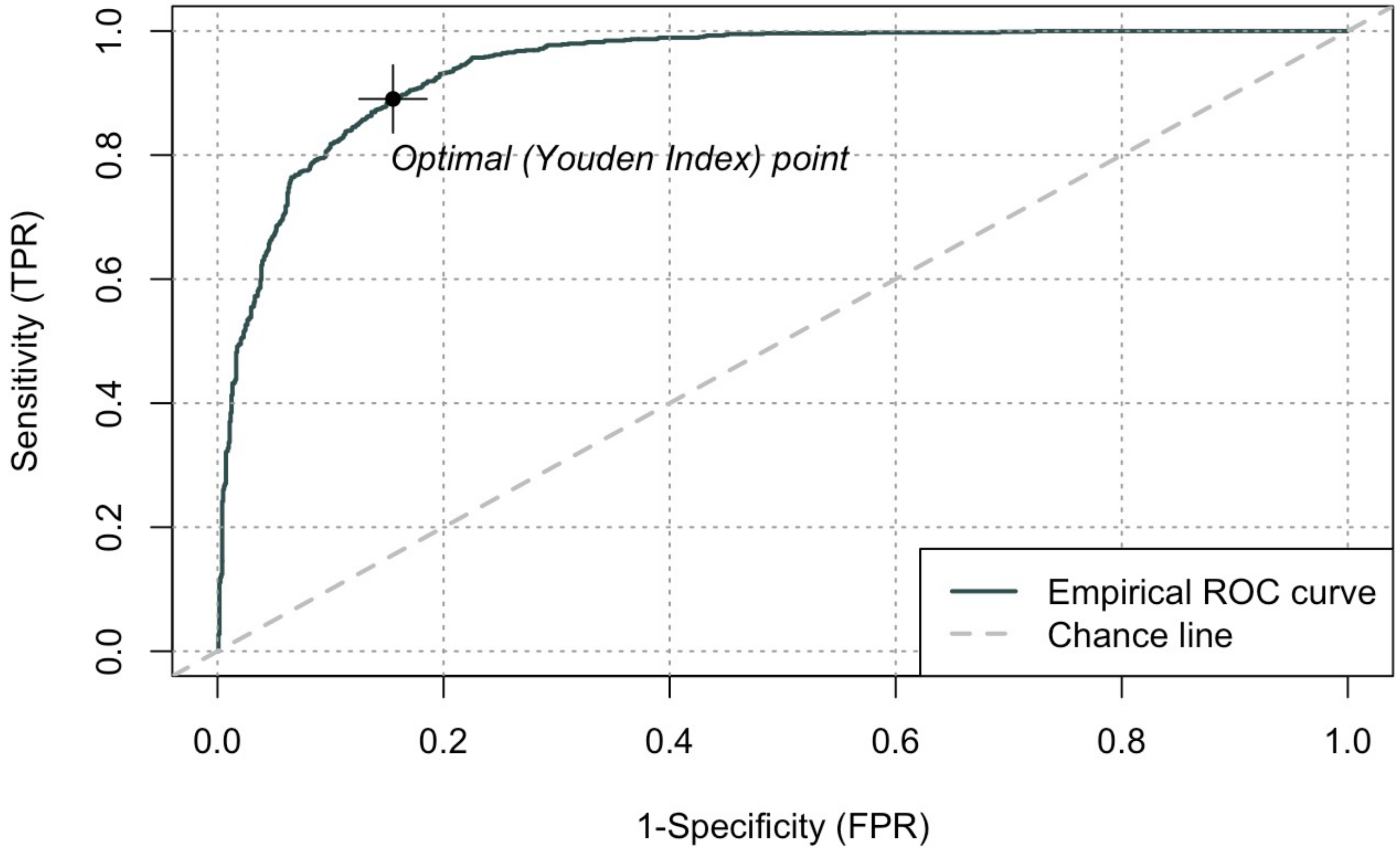
Area Under the ROC Curve

$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$



ROC Curve

```
logit_roc <- rocit(train$Bonus, train$p_hat)
plot(logit_roc)
```



ROC Curve

```
plot(logit_roc)$optimal

    value       FPR       TPR    cutoff
0.7352326 0.1552436 0.8904762 0.4229724
```

Optimal cut-off that maximizes Youden index

$$J = \text{sensitivity} + \text{specificity} - 1$$

ROC Curve

```
summary(logit_roc)

Method used: empirical
Number of positive(s): 840
Number of negatives(s): 1211
Area under curve: 0.9428
```

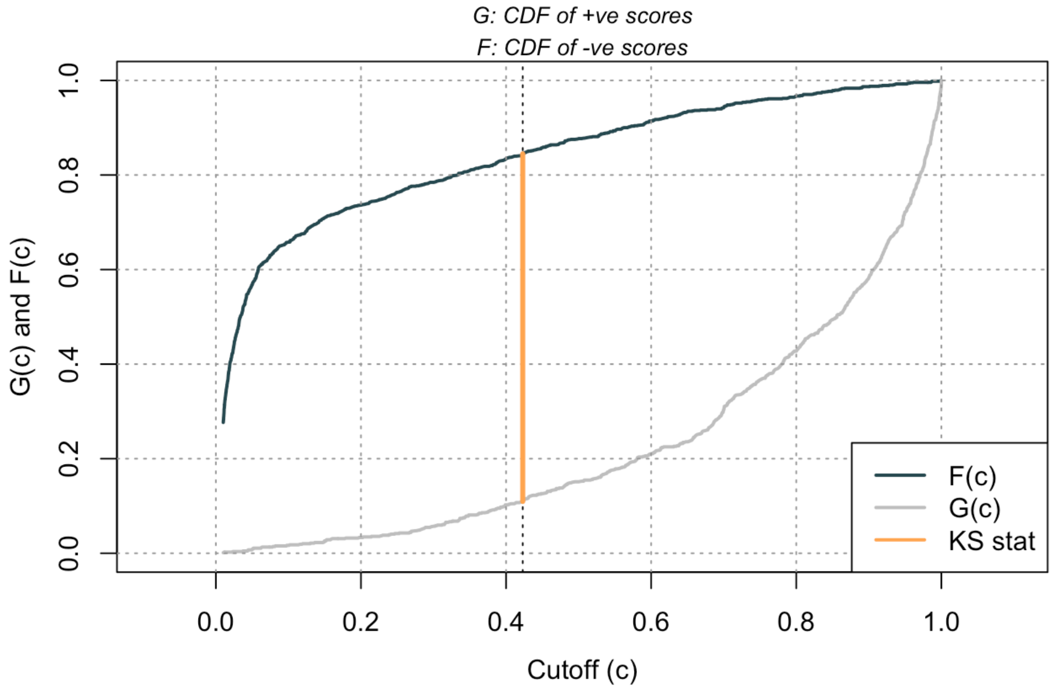
$$AUC = \%\,Concordant + \frac{1}{2}(\%\,Tied)$$

ASSESSING PREDICTIVE POWER

KS Statistic

K-S Statistic

- Very popular measure in banking and finance industries.
- The Two-Sample K-S statistic can determine if there is a difference between two cumulative distribution functions.
- Has a corresponding hypothesis test, with **D test statistic** (used for model comparison), and p-value.



K-S Statistic or Youden?

- D test statistic is used for model comparison.

$$D = \max(TPR - FPR)$$
$$= \max(Sensitivity + Specificity - 1)$$
$$= \max(Youden\ J)$$

- Mathematically **equivalent** to Youden’s J statistic.

Best Cut-off?

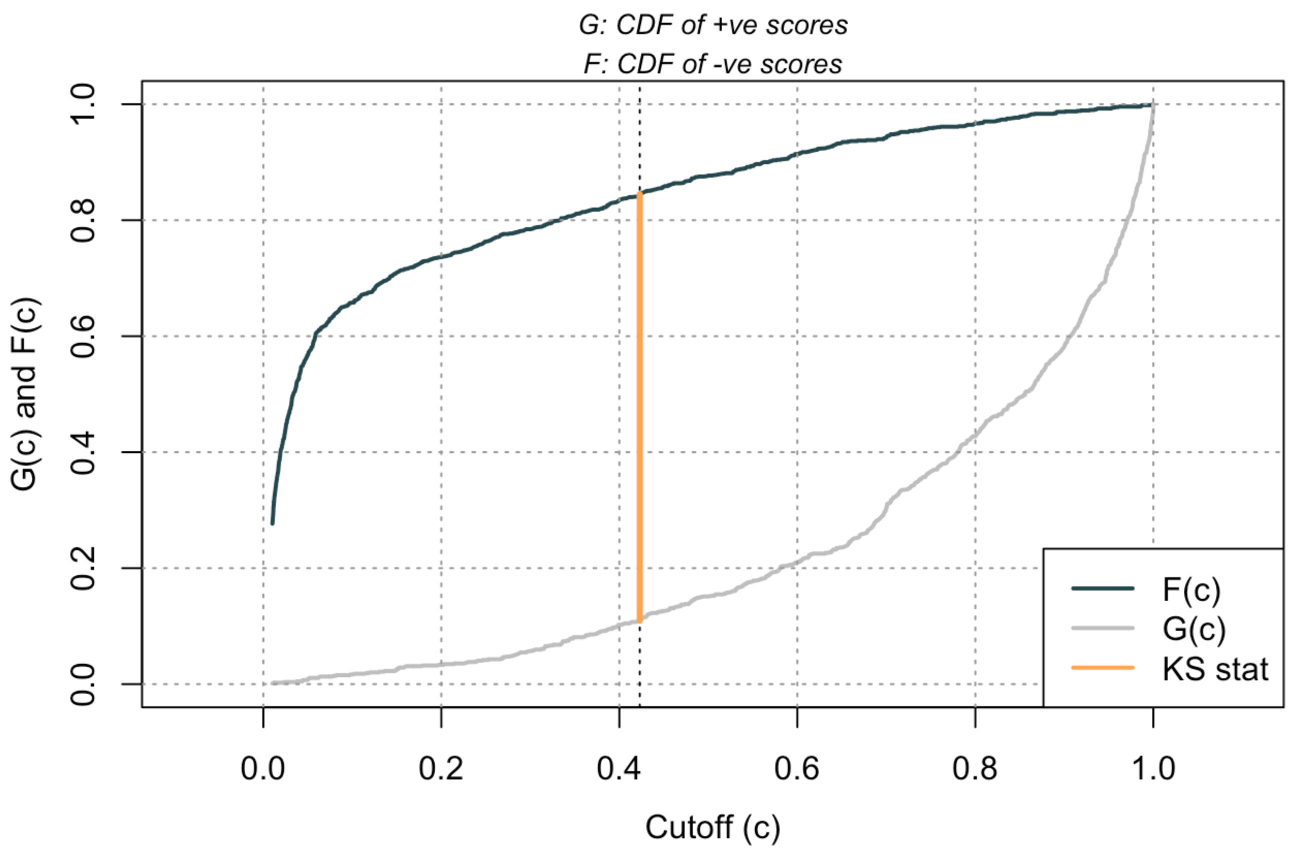
- **Always** consider the cost of false positives and false negatives when doing classification.
- When **NOT** considering costs, many different techniques to “optimal” cut-off.
- $\boldsymbol{KS}$ **statistic D** (maximum difference between TPR and FPR):

$$D = \max_{depth}(TPR - FPR)$$

- “Optimal” – select cut-off that produces highest D statistic (same as Youden’s).

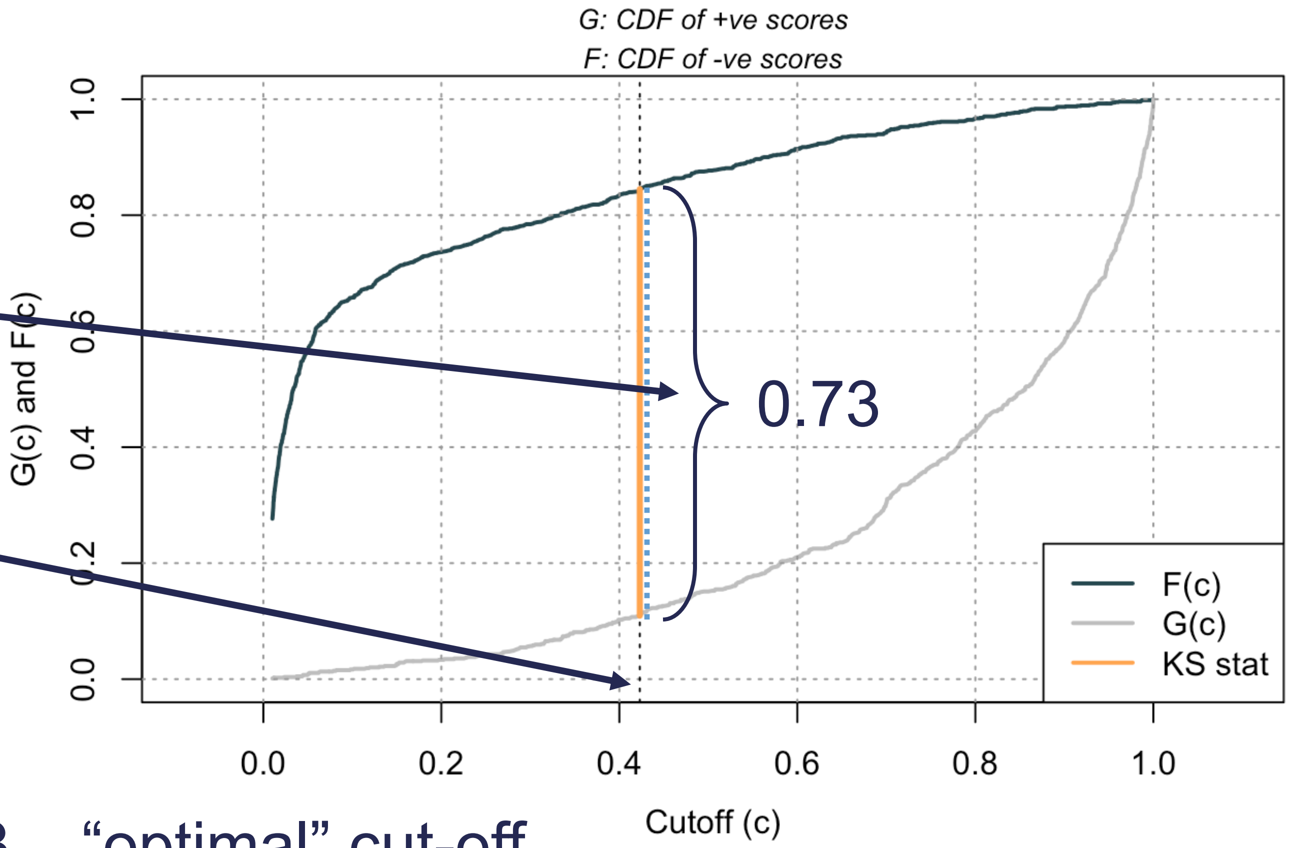
K-S Statistic

```
ksplot(logit_roc)

ksplot(logit_roc)$`KS Stat`
[1] 0.7352326

ksplot(logit_roc)$`KS Cutoff`
[1] 0.4229724
```



K-S Statistic

```
ksplot(logit_roc)

ksplot(logit_roc)$`KS Stat`
[1] 0.7352326

ksplot(logit_roc)$`KS Cutoff`
[1] 0.4229724
```



0.423 – “optimal” cut-off

ASSESSING PREDICTIVE POWER

Precision vs. Recall

Sensitivity / Recall

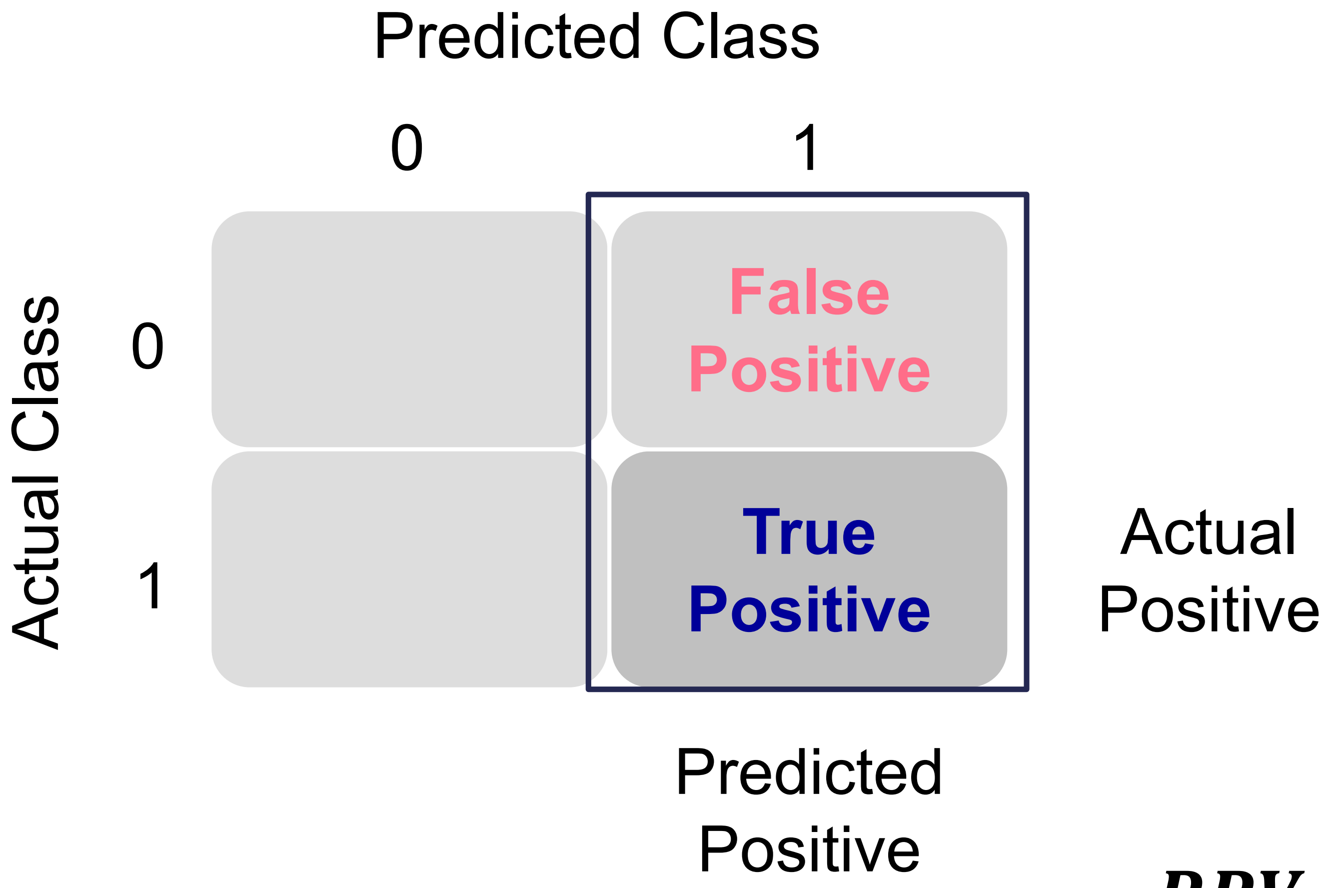
		Predicted Class	
		0	1
Actual Class	0		
	1	**False Negative**	**True Positive**

Actual Positive

Predicted Positive

$$TPR = \frac{TP}{TP + FN}$$

Precision



$$PPV = \frac{TP}{TP + FP}$$

Best Cut-off?

- **Always** consider the cost of false positives and false negatives when doing classification.
- When **NOT** considering costs, many different techniques to "optimal" cut-off.
- $\boldsymbol{F_1}$ **score** (precision-recall version of Youden's Index):

$$F_1 = 2\left(\frac{\text{precision}\times\text{recall}}{\text{precision} + \text{recall}}\right)$$

- "Optimal" – precision and recall are weighed equally, so select cut-off that produces highest F_1 score.

Precision, Recall, F_1

```
library(ROCit)

logit_meas <- measureit(train$p_hat, train$Bonus, measure = c(“PREC”, “REC”,
”FSCR”))

fscore_table <- data.frame(Cutoff = logit_meas$Cutoff, FScore = logit_meas$FSCR)
head(arrange(fscore_table, desc(FScore)), n = 1)
```

```
     Cutoff    FScore
1  0.4229724 0.8423423
```

Optimal cut-off that maximizes F1-score

DOES NOT TYPICALLY MATCH YOUDEN CUT-OFF

Precision & Lift

- Common calculation in marketing.
- Great for interpretation around validity of model ranking / classifying observations correctly.

$$Lift = PPV/\pi_1$$

- The top **depth**% of your customers, based on predicted probability, you get **lift** times as many responses compared to targeting a random sample of **depth**% of your customers.
- **Best seen through an example!**

Lift Chart

```
logit_lift <- gainstable(logit_roc)
print(logit_lift)
```

```
   Bucket Obs  CObs Depth Resp  CResp  RespRate  CRespRate  CCapRate  Lift CLift
1       1 205   205   0.1  200    200     0.976      0.976     0.238 2.382 2.382
2       2 205   410   0.2  190    390     0.927      0.951     0.464 2.263 2.323
3       3 205   615   0.3  167    557     0.815      0.906     0.663 1.989 2.211
4       4 205   820   0.4  134    691     0.654      0.843     0.823 1.596 2.058
5       5 206  1026   0.5   92    783     0.447      0.763     0.932 1.090 1.863
6       6 205  1231   0.6   42    825     0.205      0.670     0.982 0.500 1.636
7       7 205  1436   0.7   12    837     0.059      0.583     0.996 0.143 1.423
8       8 205  1641   0.8    1    838     0.005      0.511     0.998 0.012 1.247
9       9 205  1846   0.9    2    840     0.010      0.455     1.000 0.024 1.111
10     10 205  2051   1.0    0    840     0.000      0.410     1.000 0.000 1.000
```

Lift Chart

```
logit_lift <- gainstable(logit_roc)
print(logit_lift)
```

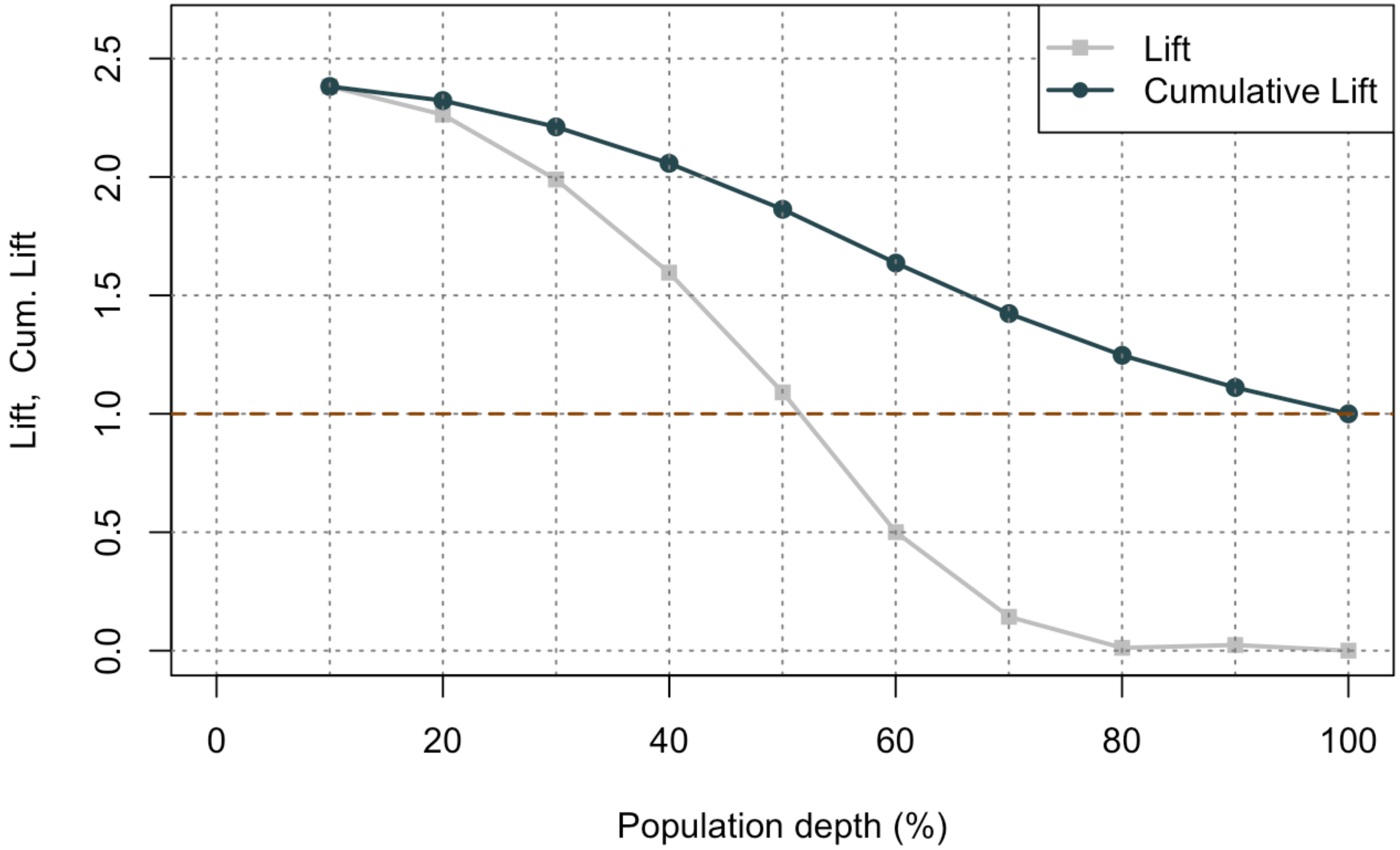
```
   Bucket Obs  CObs Depth Resp  CResp RespRate  CRespRate  CCapRate   Lift  CLift
1       1 205   205   0.1  200    200    0.976      0.976     0.238  2.382  2.382
2       2 205   410   0.2  190    390    0.927      0.951     0.464  2.263  2.323
3       3 205   615   0.3  167    557    0.815      0.906     0.663  1.989  2.211
4       4 205   820   0.4  134    691    0.654      0.843     0.823  1.596  2.058
5       5 206  1026   0.5   92    783    0.447      0.763     0.932  1.090  1.863
6       6 205  1231   0.6   42    825    0.205      0.670     0.982  0.500  1.636
7       7 205  1436   0.7   12    837    0.059      0.583     0.996  0.143  1.423
8       8 205  1641   0.8    1    838    0.005      0.511     0.998  0.012  1.247
9       9 205  1846   0.9    2    840    0.010      0.455     1.000  0.024  1.111
10     10 205  2051   1.0    0    840    0.000      0.410     1.000  0.000  1.000
```

Lift Chart

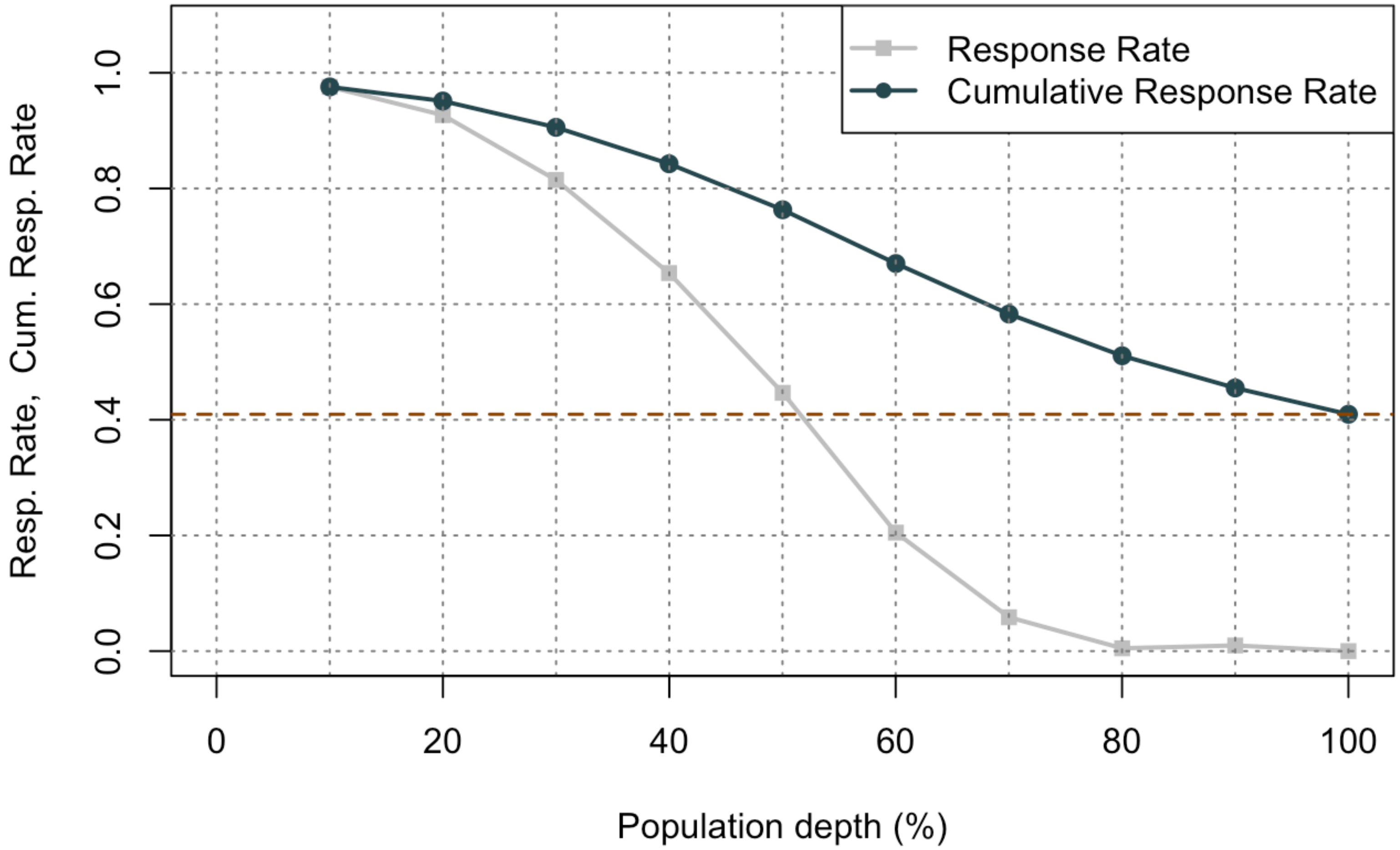
```
logit_lift <- gainstable(logit_roc)
print(logit_lift)
```

```
   Bucket Obs  CObs Depth Resp  CResp  RespRate  CRespRate  CCapRate   Lift  CLift
1       1 205   205   0.1  200    200     0.976      0.976     0.238  2.382  2.382
2       2 205   410   0.2  190    390     0.927      0.951     0.464  2.263  2.323
3       3 205   615   0.3  167    557     0.815      0.906     0.663  1.989  2.211
4       4 205   820   0.4  134    691     0.654      0.843     0.823  1.596  2.058
5       5 206  1026   0.5   92    783     0.447      0.763     0.932  1.090  1.863
6       6 205  1231   0.6   42    825     0.205      0.670     0.982  0.500  1.636
7       7 205  1436   0.7   12    837     0.059      0.583     0.996  0.143  1.423
8       8 205  1641   0.8    1    838     0.005      0.511     0.998  0.012  1.247
9       9 205  1846   0.9    2    840     0.010      0.455     1.000  0.024  1.111
10     10 205  2051   1.0    0    840     0.000      0.410     1.000  0.000  1.000
```

Lift Chart


Lift
Cumulative Lift
Lift, Cum. Lift
0.0
0.5
1.0
1.5
2.0
2.5
0
20
40
60
80
100
Population depth (%)

Response Rate Chart

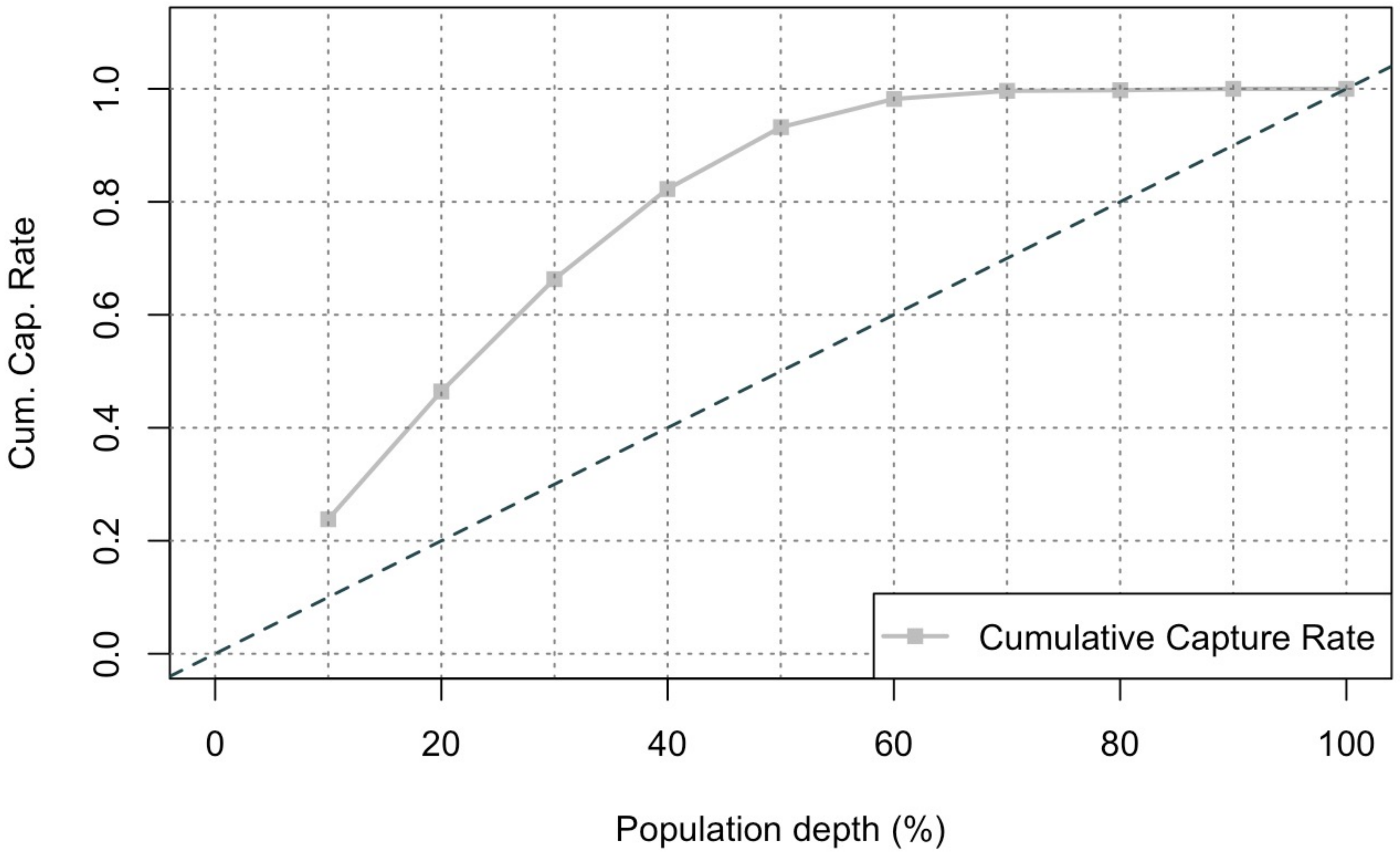


Lift Chart

```
logit_lift <- gainstable(logit_roc)
print(logit_lift)
```

```
   Bucket Obs  CObs Depth  Resp  CResp  RespRate  CRespRate  CCapRate  Lift CLift
1       1 205   205   0.1   200    200     0.976      0.976     0.238 2.382 2.382
2       2 205   410   0.2   190    390     0.927      0.951     0.464 2.263 2.323
3       3 205   615   0.3   167    557     0.815      0.906     0.663 1.989 2.211
4       4 205   820   0.4   134    691     0.654      0.843     0.823 1.596 2.058
5       5 206  1026   0.5    92    783     0.447      0.763     0.932 1.090 1.863
6       6 205  1231   0.6    42    825     0.205      0.670     0.982 0.500 1.636
7       7 205  1436   0.7    12    837     0.059      0.583     0.996 0.143 1.423
8       8 205  1641   0.8     1    838     0.005      0.511     0.998 0.012 1.247
9       9 205  1846   0.9     2    840     0.010      0.455     1.000 0.024 1.111
10     10 205  2051   1.0     0    840     0.000      0.410     1.000 0.000 1.000
```

Cumulative Capture Rate Chart (Gain Chart)



ASSESSING PREDICTIVE POWER

Accuracy vs. Error

Accuracy

Predicted Class

Actual Class		0	1	
	0	**True Negative**	**False Positive**	Actual Negative
	1	**False Negative**	**True Positive**	Actual Positive
		Predicted Negative	Predicted Positive	

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Accuracy

		Predicted Class		
		0	1	
Actual Class	0	**True Negative**	**False Positive**	Actual Negative
	1	**False Negative**	**True Positive**	Actual Positive
		Predicted Negative	Predicted Positive	

$$Accuracy = \frac{TP + TN}{n}$$

Misclassification (Error) Rate

Predicted Class

Actual Class	0	1	
0	True Negative	False Positive	Actual Negative
1	False Negative	True Positive	Actual Positive
	Predicted Negative	Predicted Positive	

$$\boldsymbol{Error = \frac{FP + FN}{n}}$$

Accuracy and Error

- Accuracy and error can be easily fooled so careful focusing only on them.
- If your data has 10% events and 90% non-events, you can have a 90% accurate model by guessing non-events for **every** observation.
- There is more to model building than simply maximizing overall classification accuracy.
- Good numbers to report, but not necessarily to choose models on.

Closing Thoughts on Classification

- Classification is a **decision** that is extraneous to statistical modeling.
- Although logistic regression tends to work well in classification, it is a **probability model** and does not output 1's and 0's.
- Classification assumes cost for each individual is the same.
  - Useful for groups.
  - Careful about single observation decisions.