BIOS 635: Logistic Regression

Kevin Donovan

1/28/2021

Review

- Homework 2 due on 2/4 at 11PM through GitHub Classroom
- Article Evaluation 1 assigned, due on 2/9 through GitHub Classroom
- Last lecture: discussed k-nearest neighbor and linear regression

Classification

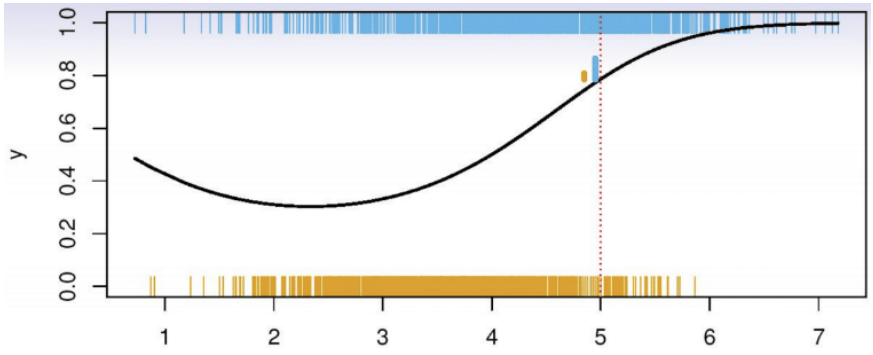
Let response $Y \in \{0, 1\}$

Goal: Predict Y using features X

What to model?:

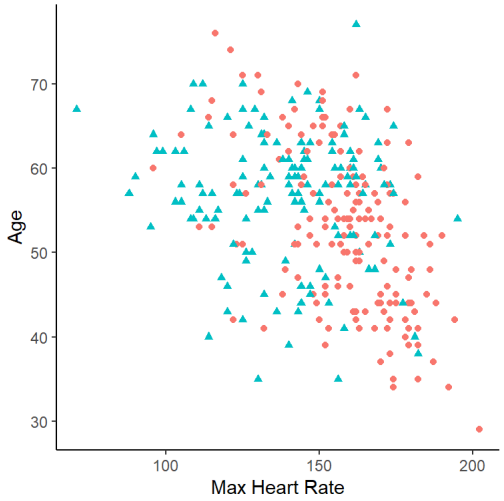
Let $p_k(x) = \Pr(Y = k|X = x)$, $k = 0, 1$

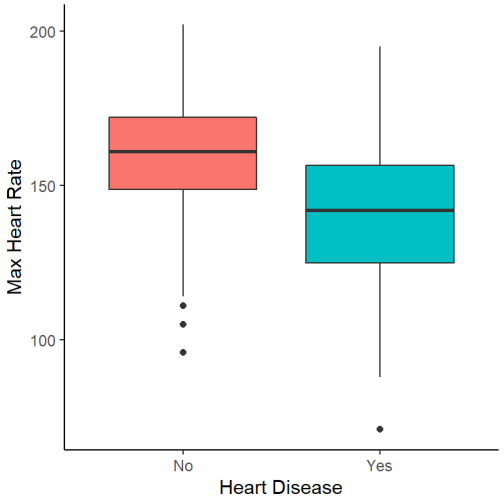
Denoted as the **conditional class probabilities** at x

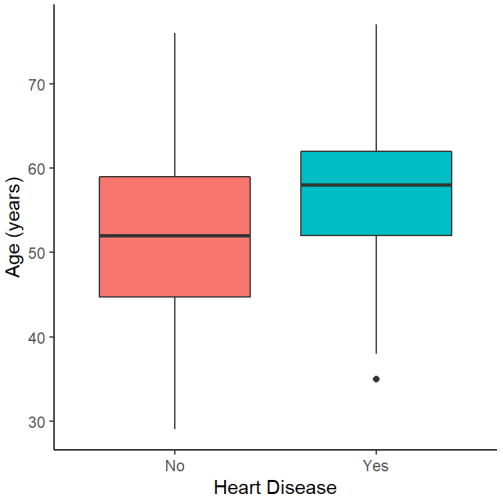


Example: Heart Disease

Goal: Predict diagnosis of heart disease in patients







Classification: Regression

Recall: Linear regression model

$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$

$E(Y|X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ as $\mathrm{E}(\epsilon|X) = \mathrm{E}(\epsilon) = 0$

For binary Y, $\mathrm{E}(Y|X) = \Pr(Y = 1|X)$

$\Pr(Y = 1|X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

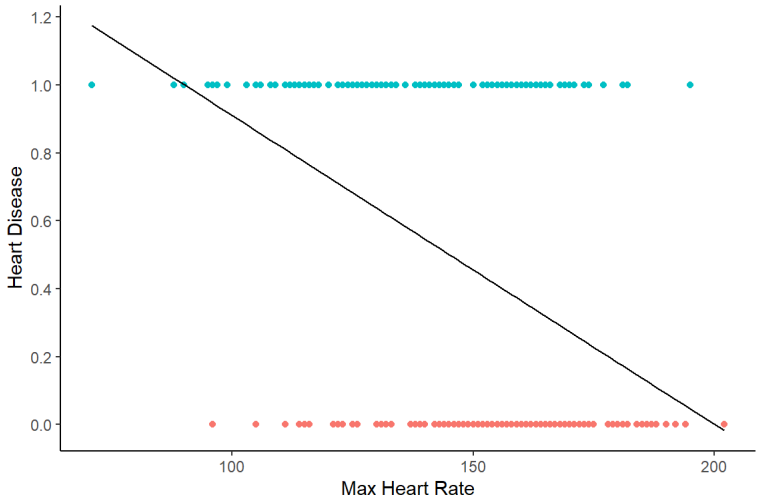
Denoted *linear probability model*

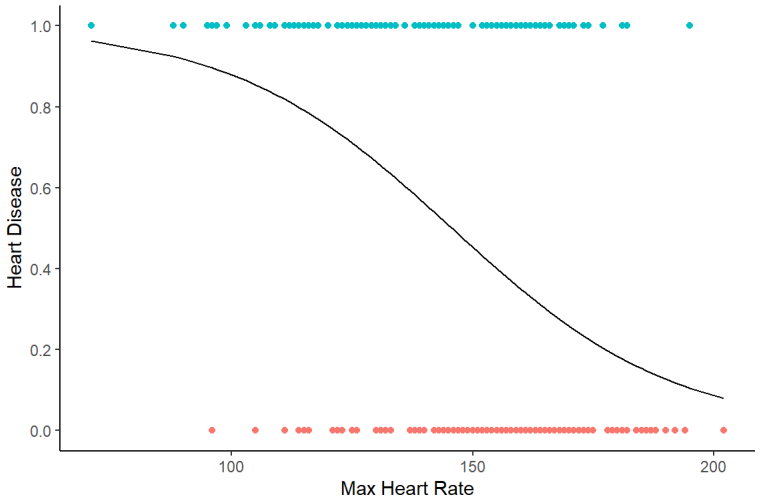
Limitation: $0 \leq \Pr(Y = 1|X) \leq 1$ but linear function not constrained

Classification: Regression

Consider:

Model probability of heart disease as function of max heart rate





Logistic Regression

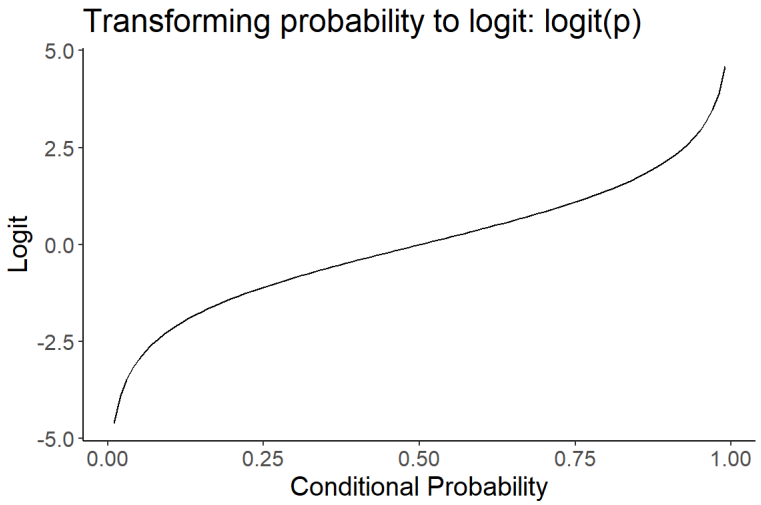
Model: Single Feature X

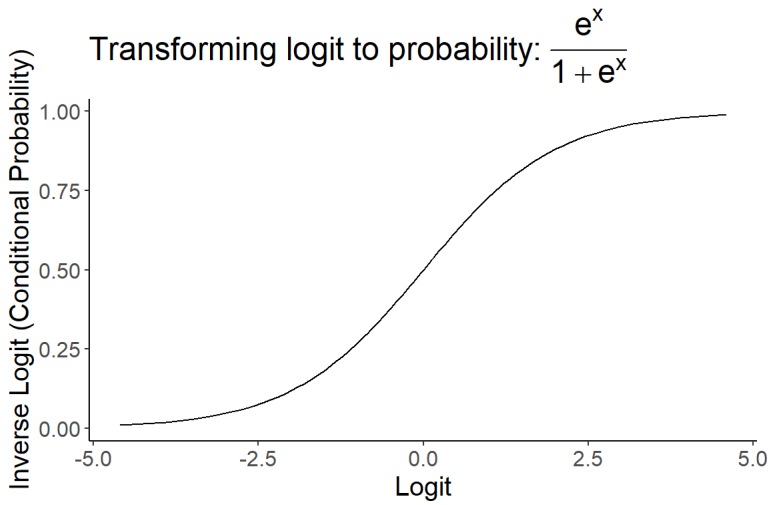
$$\text{logit}[p(Y=1|X)] = \beta_0 + \beta_1 X$$

where $\text{logit}(p) = \ln(\frac{p}{1-p}) = \log(\text{odds}[p(Y=1|X)])$

In terms of conditional probability:

$$p(Y=1|X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$





Logistic Regression

Estimation: Maximum Likelihood

Specify distribution for $Y|X$ as Y is **binary**

Observation i's distribution: $f(y_i|x_i) = p(x_i)^{y_i}[1-p(x_i)]^{1-y_i}$

implies $f(1|x_i) = p(x_i)$ and $f(0|x_i) = 1 - p(x_i)$

where $p(x_i) = \Pr(Y_i = 1|X_i)$

Maximum Likelihood

For n independent observations, have joint distribution of sample:

$$f(y|x) = \prod_{i=1}^{n=1} p(x_i)^{y_i}[1-p(x_i)]^{1-y_i}$$

$$\log(f) = \sum_{i=1}^{n} y_i \log[p(x_i)] + (1-y_i)\log[1-p(x_i)]$$

$$= \sum_{i=1}^{n_1} \log[p(x_i)] + \sum_{j=1}^{n_0} \log[1-p(x_i)]$$

$$\text{where } n_1 = \sum_{i=1}^{n}(y_i) \text{ and } n_0 = n - n_1$$

modeling $\text{logit}[p(Y=1|X)] = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Maximum Likelihood Example

Intuition:

Find estimates of β which best match with observed data, assuming data generated from specified likelihood

Fitting in R: `glm` function

```
glm_fit <-
  glm(heart_disease~MAX_Heart_Rate+Age,
             family=binomial(),
            data=heart_data)

# Raw output
summary(glm_fit)
```

```
##
## Call:
## glm(formula = heart_disease ~ MAX_Heart_Rate + Age, family = binomial(),
##     data = heart_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1108  -0.9418  -0.5842   1.0657   2.0774
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     4.839597   1.498563   3.229  0.00124 **
## MAX_Heart_Rate -0.040711   0.006808  -5.980 2.24e-09 ***
## Age             0.019706   0.015339   1.285  0.19889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 417.98  on 302  degrees of freedom
## Residual deviance: 359.24  on 300  degrees of freedom
## AIC: 365.24
##
## Number of Fisher Scoring iterations: 3
```

```
# Format output
tidy(glm_fit) %>%
  mutate(p.value=ifelse(p.value<0.005, "<0.005",
```

```
                    as.character(round(p.value, 3))),
       term=fct_recode(factor(term),
                       "Intercept"="(Intercept)",
                       "Max Heart Rate"=
                       "MAX_Heart_Rate")) %>%
flextable() %>%
set_header_labels("term"="Variable",
                  "estimate"="Estimate",
                  "std.error"="Std. Error",
                  "statistic"="Z Statistic",
                  "p.value"="P-value") %>%
autofit()
```

Variable	Estimate	Std. Error	Z Statistic	P-value
Intercept	4.840	1.4986	3.2	<0.005
Max Heart Rate	-0.041	0.0068	-6.0	<0.005
Age	0.020	0.0153	1.3	0.199

Generalized Linear Models

Logistic regression models are an example of a *generalized linear model* (GLM)

GLM: Extension of standard linear model to handle general distributions

Examples:

- Normally distributed residuals (linear regression)
- Binary outcomes (logisitic regression)
- Categorical outcomes (multinomial logisitc regression)
- Count outcomes Poisson regression)
- Rates (beta regression)

Generalized Linear Models

Structure of model:

1. Choose conditional distribution $f(y|x)$

- Linear regression: $f(y|x) \sim \text{Normal}(\mu_{y|x}, \sigma^2)$
  - $\mu_{y|x} = \text{E}(Y|X); \sigma^2 = \text{Var}(Y|X) = \text{Var}(\epsilon)$
- Logistic regression: $f(y|x) \sim \text{Binomial}[p(x)]$
  - $p(x) = \mu_{y|x} = \Pr(Y = 1|X)$

Generalized Linear Models

2. Choose *link function* $g(\mu_{y|x}) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

- Linear regression: $g(\mu_{y|x}) = \mu_{y|x}$
- Logistic regression $g(\mu_{y|x}) = \log(\frac{\mu_{y|x}}{1-\mu_{y|x}})$
- **Idea**: $g(\mu_{y|x}) : \mathcal{X}_\mu \to (-\infty, \infty)$

3. Construct likelihood and fit

- Assuming independent observations:
  - $f(y|x) = \prod_{i=1}^n f(y_i|x_i)$

Logistic Regression

Let's go back to heart disease example:

Variable	Estimate	Std. Error	Z Statistic	P-value
Intercept	4.840	1.4986	3.2	<0.005
Max Heart Rate	-0.041	0.0068	-6.0	<0.005
Age	0.020	0.0153	1.3	0.199

Estimated model: $\hat{\Pr}[Y = 1|HeartRate, Age] = \frac{e^{4.84 - 0.041 HeartRate + 0.02 Age}}{1 + e^{4.84 - 0.041 HeartRate + 0.02 Age}}$

Interpretation:

1. $\hat{\beta}_0 = 4.840$

- $\hat{\Pr}[Y = 1|HeartRate = 0, Age = 0] = \frac{e^{4.84}}{1 + e^{4.84}}$

2. $\hat{\beta}_1 = -0.041$

- $\rightarrow$ Probability of heart disease **decreases** as `max heart rate` **increases** (holding age fixed)

3. $\hat{\beta}_1 = 0.02$

- $\rightarrow$ Probability of heart disease **increase** as age **increases** (holding `max heart rate` fixed)
  - *P-value = 0.2 $\rightarrow$ age may not be useful predictor*

Logistic Regression

Intercept:

- `heart rate` = 0 and/or age = 0 doesn't make sense

Solution: center at means

- `heart rate` - μ = 0 $\rightarrow$ `heart rate` = μ

Variable	Estimate	Std. Error	Z Statistic	P-value
Intercept	-0.178	0.1276	-1.4	0.162
Max Heart Rate centered	-0.041	0.0068	-6.0	<0.005
Age (years) centered	0.020	0.0153	1.3	0.199

Interpretation:

I. $\hat{\beta}_0 = -0.18$

$$\hat{\Pr}[Y = 1 | HeartRate = 0, Age = 0] = \frac{e^{-0.18}}{1 + e^{-0.18}}$$

Slopes $\hat{\beta}_1, \hat{\beta}_2$ not changed

Logistic Regression

Model-based estimated probabilities (non-centered):

For patient with `heart rate`=150 and age=65 years

$$\hat{\Pr}[Y = 1|HeartRate = 150, Age = 65] = \frac{e^{4.84 - 0.041*150 + 0.02*65}}{1 + e^{4.84 - 0.041*150 + 0.02*60}} = 0.4975$$

Based on $\hat{\Pr}[Y = 1|HeartRate, Age]$ can create predicted response $\hat{Y}$ by thresholding

Logistic Regression: Confounding

Example: Credit Card Default Rate

- Consider predicting if a person defaults on their loan based on
  - *Student Status (Student or Not Student):*

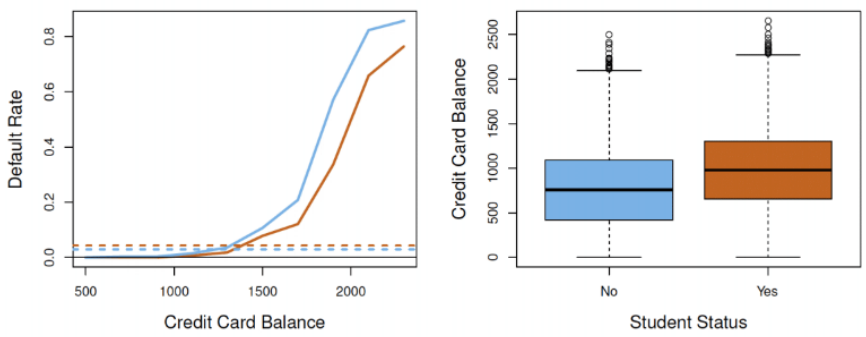
	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-3.5041	0.0707	-49.55	< 0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

- Now consider adding features: `credit balance` and `income`

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-10.8690	0.4923	-22.08	< 0.0001
`balance`	0.0057	0.0002	24.74	< 0.0001
`income`	0.0030	0.0082	0.37	0.7115
`student[Yes]`	-0.6468	0.2362	-2.74	0.0062

- Why did `student`'s coefficient change so much? **Confounding**

Logistic Regression: Confounding

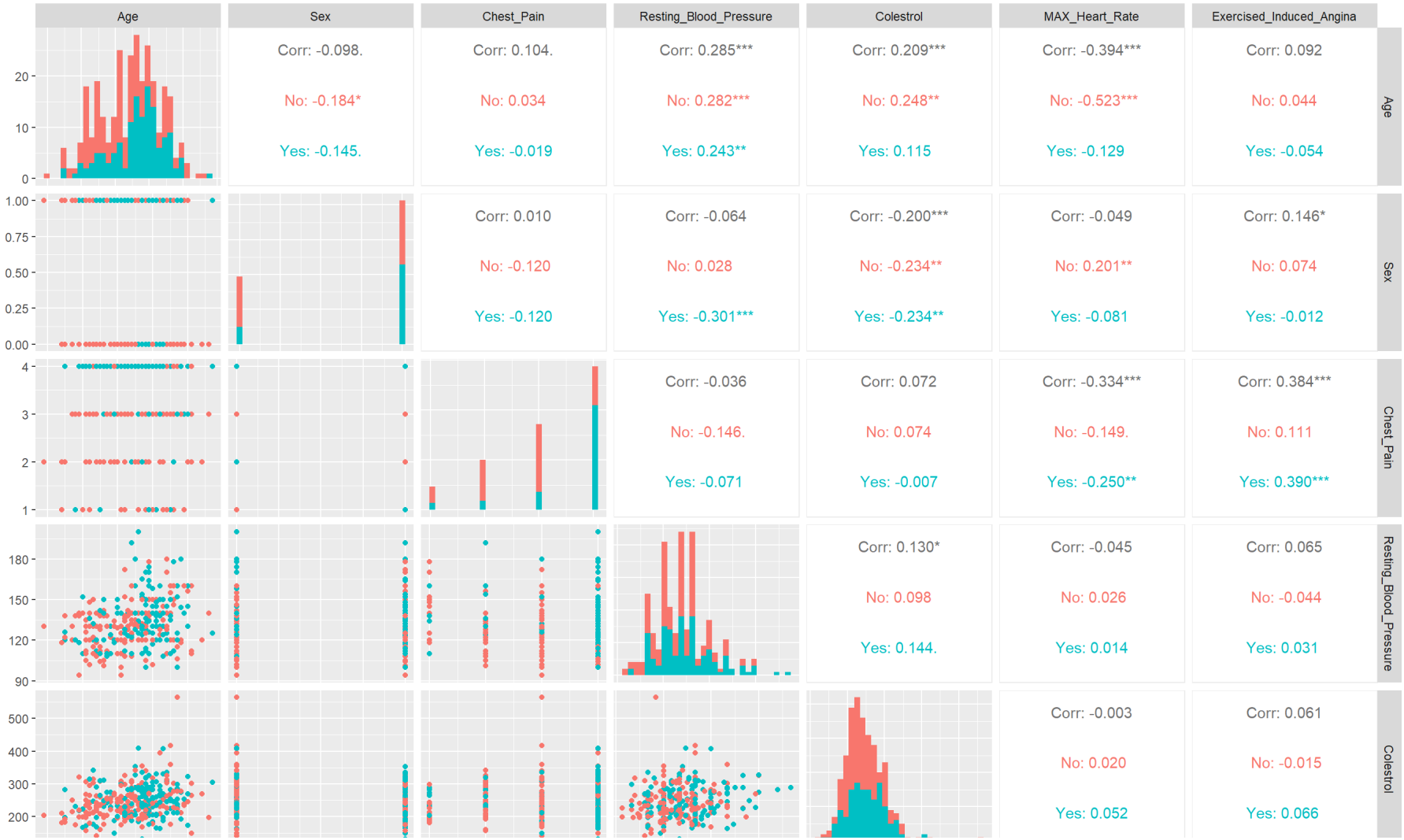


- Being a student $\rightarrow$ higher balance (more loans)
  - $\rightarrow$ *higher marginal default rate vs non-students*
  - *But is it the higher balance or simply them being students leading to defaulting more often?*
  - *Need to compare students and non-students* ***controlling*** *for balance to answer this*
  - *Can be done using regression*

Logistic Regression: Prediction

Example: Model probability of heart disease as function of many features

```
ggpairs(heart_data, columns =
          c("Age", "Sex", "Chest_Pain", "Resting_Blood_Pressure", "Colestrol",
            "MAX_Heart_Rate", "Exercised_Induced_Angina"),
        ggplot2::aes(colour=heart_disease),
        diag = list(continuous="barDiag"))
```




Corr: -0.378***
No: -0.172*
Yes: -0.292***
MAX_Heart_Rate
Exercised_Induced_Angina

Logistic Regression: Prediction

- Split data into training and testing set (60:40 split)
- Fit logistic regression model on train, evaluate on test

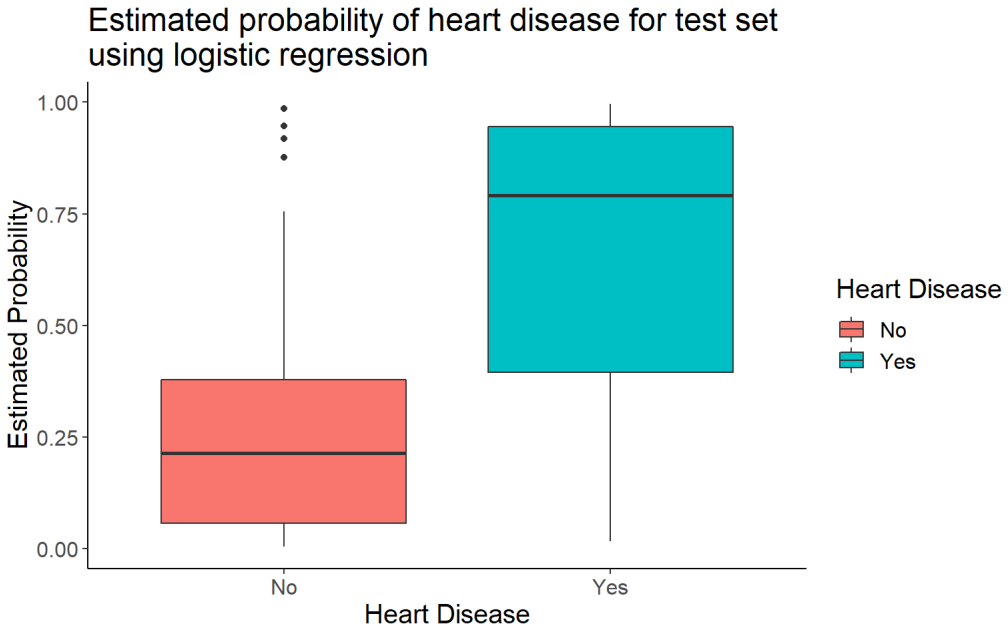
```
# Partition Data
set.seed(12)
train_test_indices <- createDataPartition(heart_data$heart_disease, p=0.6, list = FALSE)
heart_data_train <- heart_data[train_test_indices,]
heart_data_test <- heart_data[-train_test_indices,]

# Train
lm_fit <- glm(formula = heart_disease~Age+Sex+Chest_Pain+Resting_Blood_Pressure+Colestrol+
                MAX_Heart_Rate+Exercised_Induced_Angina,
              data = heart_data_train,
              family = binomial())

summary(lm_fit)
```

```
##
## Call:
## glm(formula = heart_disease ~ Age + Sex + Chest_Pain + Resting_Blood_Pressure +
##     Colestrol + MAX_Heart_Rate + Exercised_Induced_Angina, family = binomial(),
##     data = heart_data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9043  -0.6570  -0.2297   0.5699   2.3395
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -4.267477   2.897706  -1.473  0.14083
## Age                       0.004601   0.025999   0.177  0.85954
## Sex                       2.075197   0.532039   3.900 9.60e-05 ***
## Chest_Pain                0.588048   0.226267   2.599  0.00935 **
## Resting_Blood_Pressure    0.043279   0.013579   3.187  0.00144 **
## Colestrol                 0.006693   0.004119   1.625  0.10417
## MAX_Heart_Rate           -0.048617   0.012144  -4.004 6.24e-05 ***
## Exercised_Induced_Angina  1.552972   0.492859   3.151  0.00163 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 252.46  on 182  degrees of freedom
```

```
## Residual deviance: 152.29  on 175  degrees of freedom
## AIC: 168.29
##
## Number of Fisher Scoring iterations: 5
```

Logistic Regression: Prediction

```r
# Add in test set predictions
heart_data_test$estimated_prob_heart_disease <-
  predict(lm_fit, newdata=heart_data_test, type = "response")

# View test set probabilities
ggplot(data=heart_data_test,
       mapping=aes(x=heart_disease, y=estimated_prob_heart_disease,
                   fill=heart_disease))+
  geom_boxplot()+
  labs(x="Heart Disease", y="Estimated Probability",
       title = "Estimated probability of heart disease for test set\nusing logistic regression",
       fill = "Heart Disease")+
  theme_classic()+
  theme(text = element_text(size=15))
```



Logistic Regression: Prediction

- How to create predicted outcomes using estimated probabilities?
  - *Easy way:*

$$\begin{aligned}\hat{Y}_i &= I[p(X_i) > 0.5] \\ &= \begin{cases} 1 & \text{if } p(X_i) > 0.5 \\ 0 & \text{if } p(X_i) \leq 0.5 \end{cases}\end{aligned}$$

- In heart disease data:

```
# Add in test set predictions
heart_data_test <-
  heart_data_test %>%
  mutate(pred_heart_disease =
           relevel(factor(ifelse(estimated_prob_heart_disease>0.5, "Yes", "No")),
                   ref = "No"))

# Compute confusion matrix
confusionMatrix(data = heart_data_test$pred_heart_disease,
                reference = heart_data_test$heart_disease,
                positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  52  18
##        Yes 13  37
##
##                Accuracy : 0.7417
##                  95% CI : (0.6538, 0.8172)
##     No Information Rate : 0.5417
##     P-Value [Acc > NIR] : 5.135e-06
##
##                   Kappa : 0.4761
##
##  Mcnemar's Test P-Value : 0.4725
##
##             Sensitivity : 0.6727
##             Specificity : 0.8000
```

```
##          Pos Pred Value : 0.7400
##          Neg Pred Value : 0.7429
##              Prevalence : 0.4583
##          Detection Rate : 0.3083
##    Detection Prevalence : 0.4167
##       Balanced Accuracy : 0.7364
##
##        'Positive' Class : Yes
##
```

Logistic Regression: Prediction

- What if we don't split the data?

```
# Add in train set predictions
heart_data_train$estimated_prob_heart_disease <-
  predict(lm_fit, newdata=heart_data_train, type = "response")

heart_data_train <-
  heart_data_train %>%
  mutate(pred_heart_disease =
          relevel(factor(ifelse(estimated_prob_heart_disease>0.5, "Yes", "No")),
                  ref = "No"))

# Compute confusion matrix
confusionMatrix(data = heart_data_train$pred_heart_disease,
                reference = heart_data_train$heart_disease,
                positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  84  19
##        Yes 15  65
##
##                Accuracy : 0.8142
##                  95% CI : (0.7502, 0.8678)
##     No Information Rate : 0.541
##     P-Value [Acc > NIR] : 8.162e-15
##
##                   Kappa : 0.6245
##
##  Mcnemar's Test P-Value : 0.6069
##
##             Sensitivity : 0.7738
##             Specificity : 0.8485
##          Pos Pred Value : 0.8125
##          Neg Pred Value : 0.8155
##              Prevalence : 0.4590
##          Detection Rate : 0.3552
##    Detection Prevalence : 0.4372
##       Balanced Accuracy : 0.8111
##
##        'Positive' Class : Yes
##
```

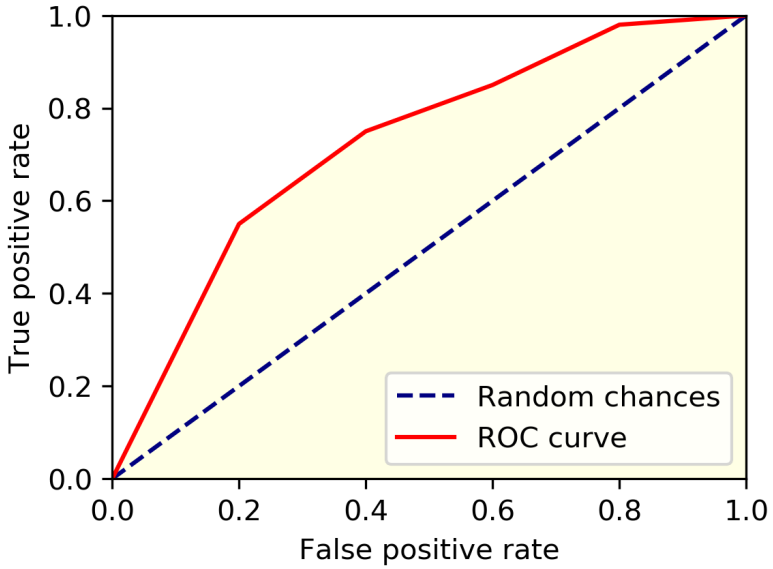
- Not a big difference. **Why?**

Classification: Threshold

- We defined $\hat{Y}$ using a 0.5 probability threshold
  - *Other thresholds may give **better performance***
  - *Infinitely many thresholds $\rightarrow$ how to aggregate?*
  - *Answer: receiver operating characteristic curve (ROC curve)*

ROC Curve

- For given threshold p, ROC curve is function:
  - $ROC(p) = (TPR, FPR)$
  - *Maps each threshold to corresponding true positive rate (TPR) amd false positive rate pair*
  - $TPR = \text{Sensitivity}$, $1 - FPR = \text{Specificity}$
  - *Helps choose "best" threshold based on sensitivity and specificity considerations*



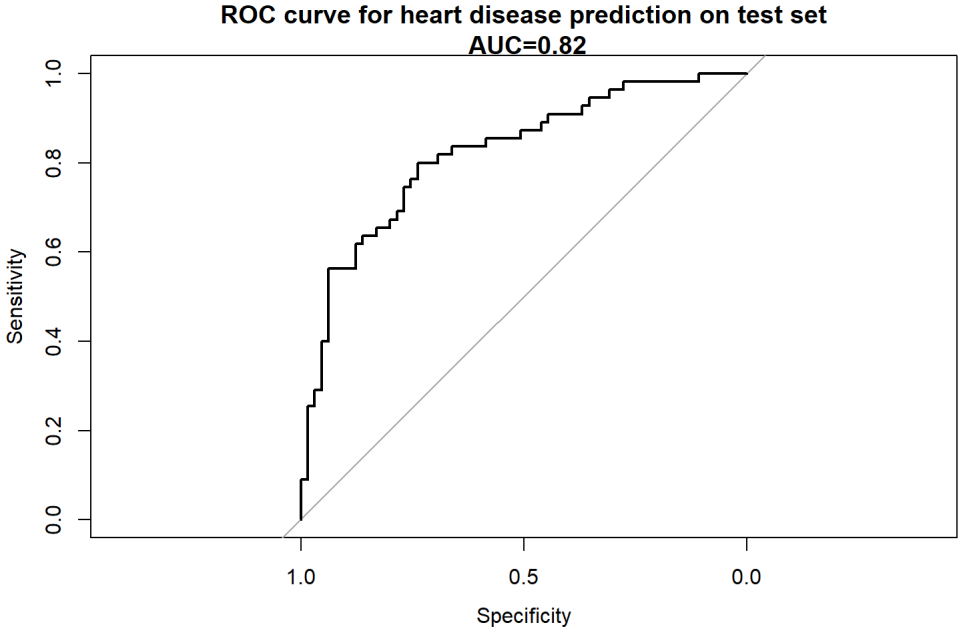
ROC Curve Ex.

- Heart disease dataset example
- Let's compute the ROC curve using the test set to evaluation differences thresholds

```
# Using pROC, add ROC curve using estimated probabilities of heart disease in test set
roc_obj <-
  roc(response = heart_data_test$heart_disease,
    predictor = heart_data_test$estimated_prob_heart_disease)

# Print obj
roc_obj
```

```
##
## Call:
## roc.default(response = heart_data_test$heart_disease, predictor = heart_data_test$estimated_prob_heart_disease)
##
## Data: heart_data_test$estimated_prob_heart_disease in 65 controls (heart_data_test$heart_disease No) < 55 cases (heart_data_test$heart_disease
## Area under the curve: 0.8215
```

```
# Plot curve
plot(roc_obj, main = paste0("ROC curve for heart disease prediction on test set\n AUC=",
                            round(auc(roc_obj),2)))
```


ROC curve for heart disease prediction on test set
AUC=0.82
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity

ROC Curve Ex.

- For actual use, need to choose threshold
  - *Example: "Elbow Point"/max(Youden's Index)*

```
# Using pROC, add ROC curve using estimated probabilities of heart disease in test set
roc_obj <-
  roc(response = heart_data_test$heart_disease,
    predictor = heart_data_test$estimated_prob_heart_disease)

# Print obj
roc_obj
```

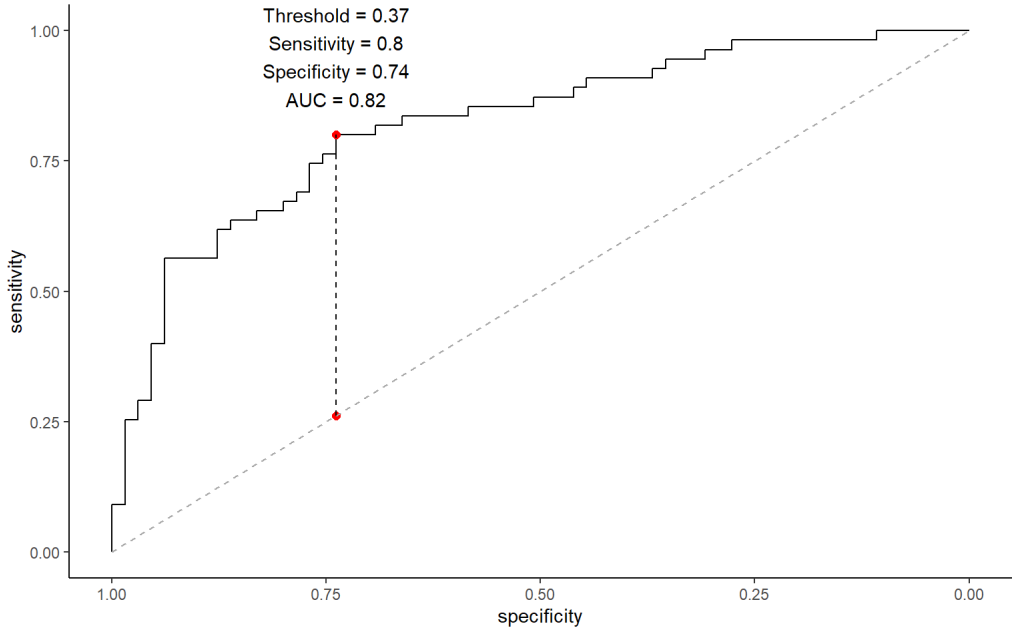
```
##
## Call:
## roc.default(response = heart_data_test$heart_disease, predictor = heart_data_test$estimated_prob_heart_disease)
##
## Data: heart_data_test$estimated_prob_heart_disease in 65 controls (heart_data_test$heart_disease No) < 55 cases (heart_data_test$heart_disease
## Area under the curve: 0.8215
```

```
# Return max Youden's index, with specificity and sensitivity
best_thres_data <-
  data.frame(coords(roc_obj, x="best", best.method = c("youden", "closest.topleft")))

best_thres_data
```

```
##   threshold specificity sensitivity
## 1 0.3748977   0.7384615         0.8
```

```
# Plot curve, add in line at elbow point
data_add_line <-
  data.frame("sensitvity"=c(1-best_thres_data$specificity,
                            best_thres_data$sensitivity),
             "specificity"=c(best_thres_data$specificity,
                            best_thres_data$specificity))

ggroc(roc_obj)+
    geom_point(
    data = best_thres_data,
    mapping = aes(x=specificity, y=sensitivity), size=2, color="red")+
    geom_point(mapping=aes(x=best_thres_data$specificity,
                y=1-best_thres_data$specificity),
                size=2, color="red")+
```

```
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1),
               color="darkgrey", linetype="dashed")+
  geom_text(data = best_thres_data,
            mapping=aes(x=specificity, y=0.95,
                        label=paste0("Threshold = ", round(threshold,2),
                                     "\nSensitivity = ", round(sensitivity,2),
                                     "\nSpecificity = ", round(specificity,2),
                                     "\nAUC = ", round(auc(roc_obj),2))))+
  geom_line(data=data_add_line,
            mapping=aes(x=specificity, y=sensitvity),
            linetype="dashed")+
theme_classic()
```



- Choose threshold based on study-specific **cost-benefit analysis**

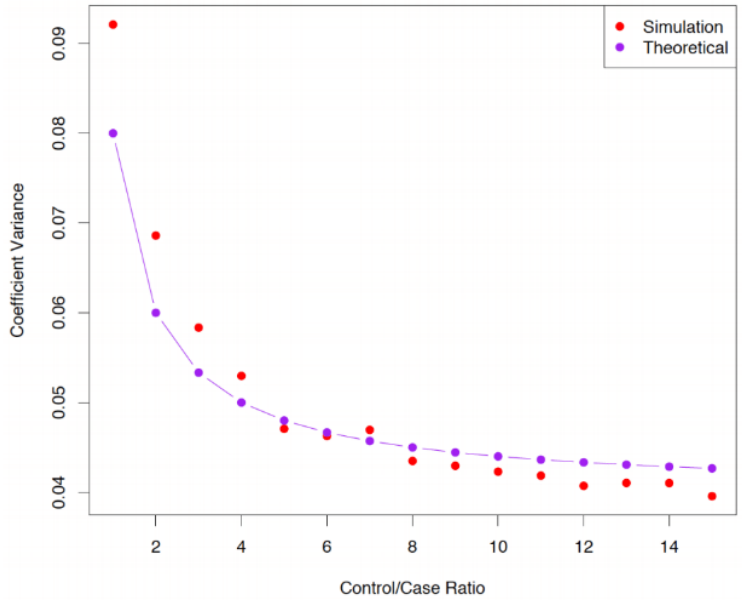
Case-control sampling

- 139 cases and 164 controls - $\hat{\pi} = 0.46$
  - *However, true proportion of heart disease in population is much lower ($\pi = 0.05$)*
- If model is correct, logistic regression allows accurate estimation of β_j, $j > 0$
  - β_0 *estimate* ***not*** *accruate due to baseline prevalance in sample*
- Can **correct** intercept estimate using transformation:

$$\hat{\beta}_0^* = \hat{\beta}_0 + \log(\frac{\pi}{1-\pi}) - \log(\frac{\hat{\pi}}{1-\hat{\pi}})$$

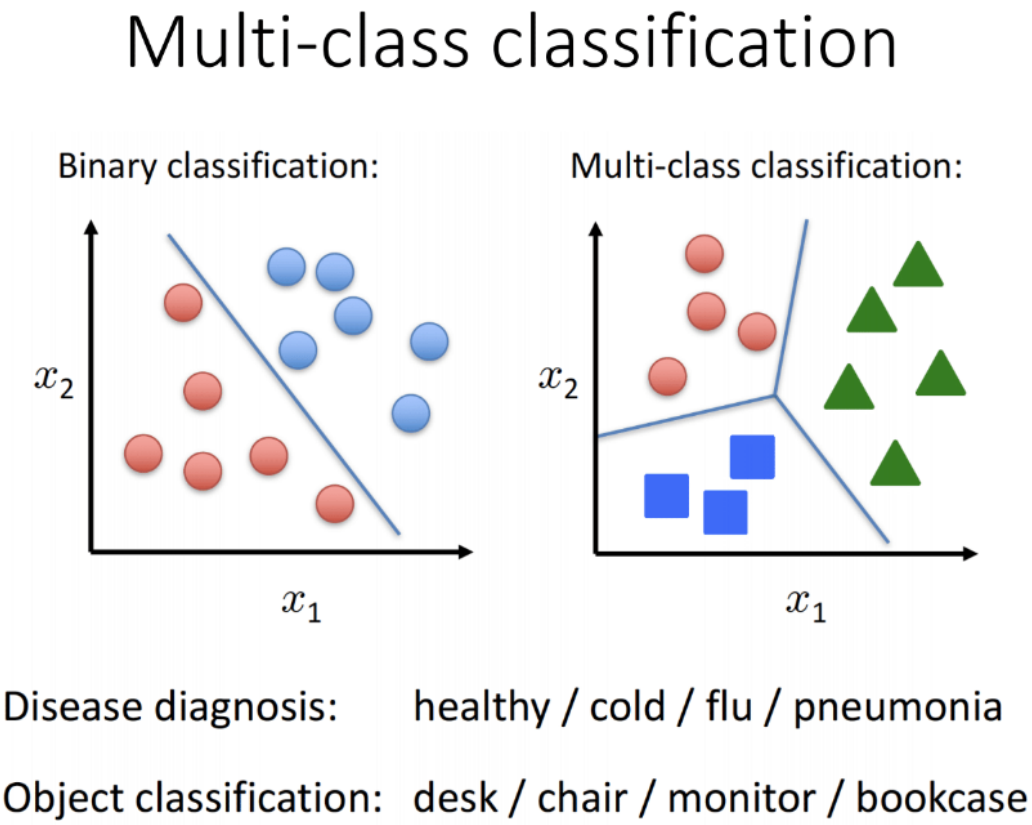
- Case-control sampling done due to cases being rare
  - *i.e., all cases in population selected*
  - *Controls then randomly sampled from population*
  - *How many controls to sample?*

Case-control sampling



Sampling more controls than cases reduces the variance of the parameter estimates. But after a ratio of about 5 to 1 the variance reduction flattens out.

Multi-class classification

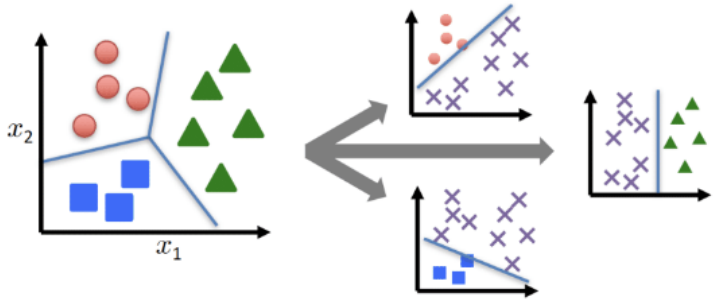


Multi-class classification: logistic regression

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package `glmnet`) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+...+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+...+\beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.



Multiclass logistic regression is also referred to as *multinomial regression*.

Multi-class classification: logistic regression

- Can extend usual two-class logistic regression for multiple-classes
  - *Don't work well in practice*
- Instead discuss other methods which are superior
  - *Ex. Linear discrimminant analysis (LDA)*

Song of the session