Session 2 (DAY 2)

Statistical and Machine Learning Models

Workshop on Quantitative Literacy and Statistics

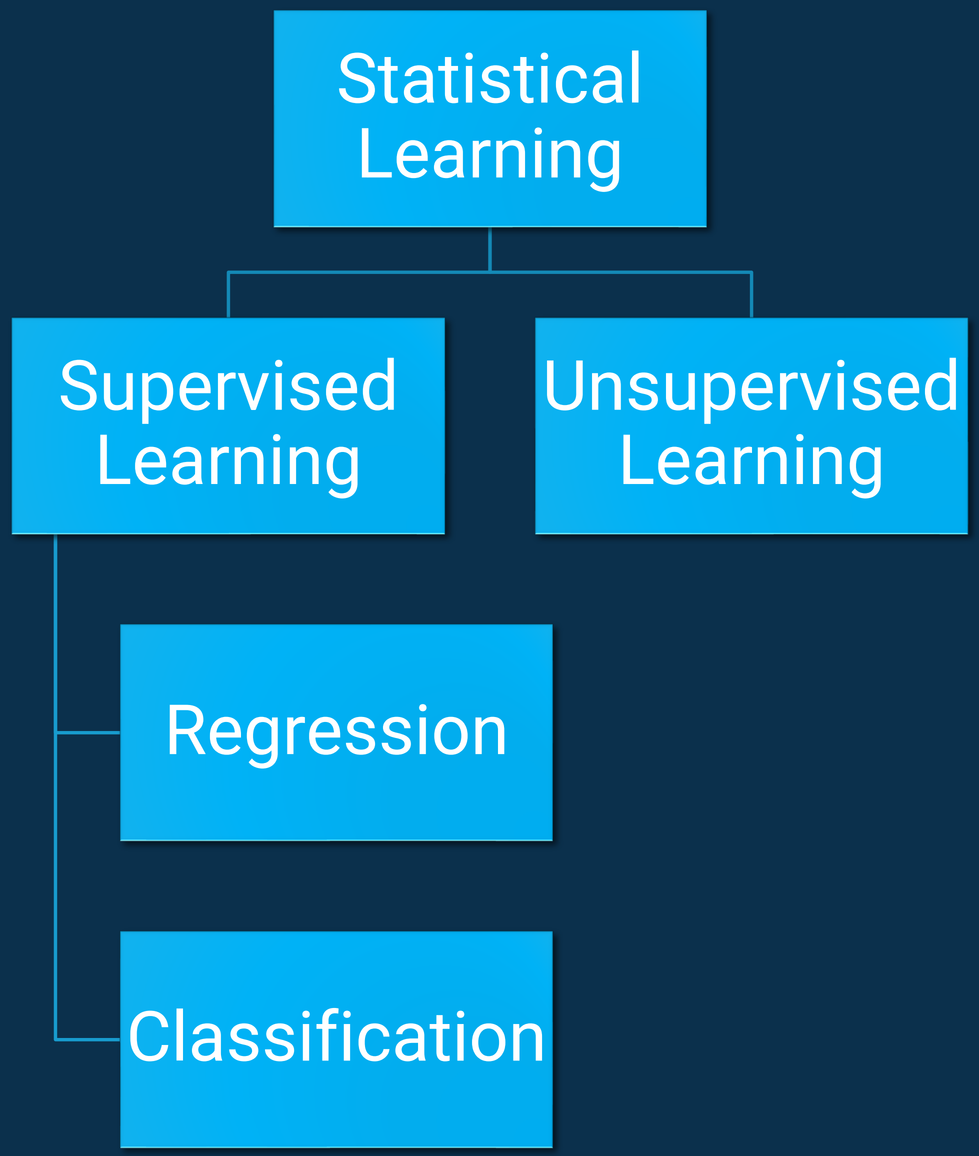
UNIVERSITY OF NEBRASKA AT OMAHA
DATA AND DECISION SCIENCES

Session 2: Part 1

Supervised Learning

Workshop on Quantitative Literacy and Statistics

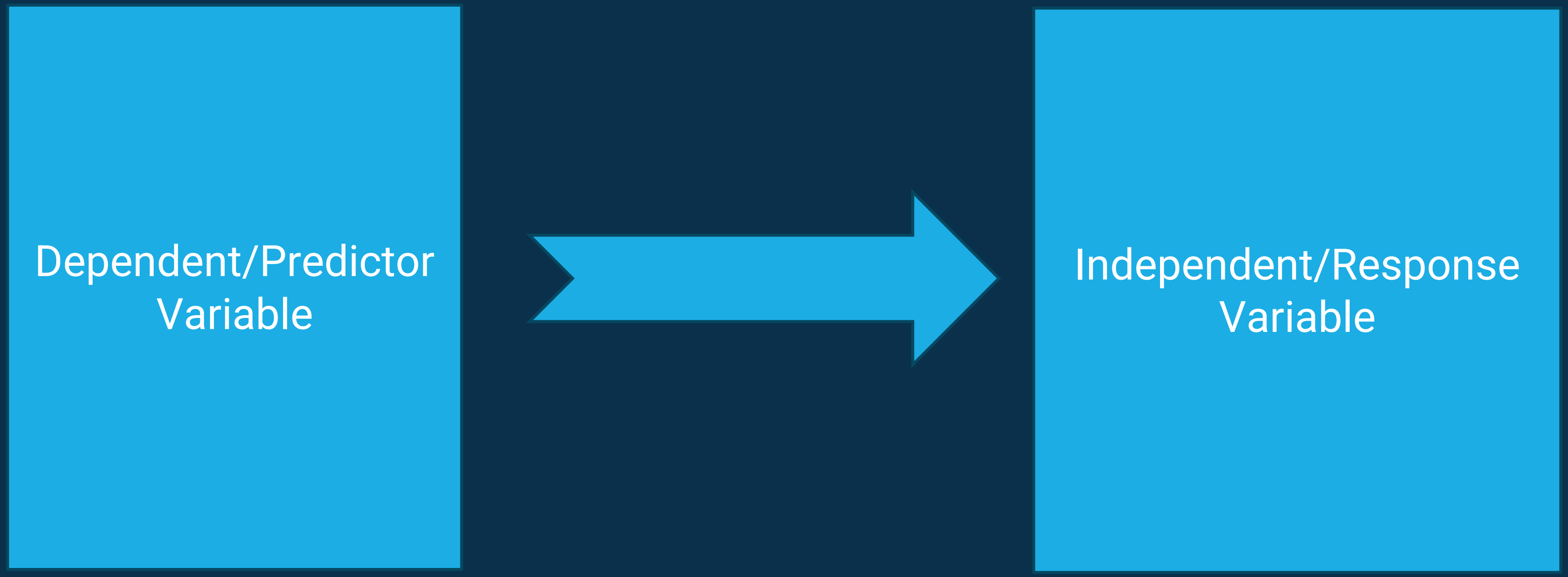
UNIVERSITY OF NEBRASKA AT OMAHA
DATA AND DECISION SCIENCES


Statistical Learning
Supervised Learning
Unsupervised Learning
Regression
Classification

Supervised Learning

Supervised Learning

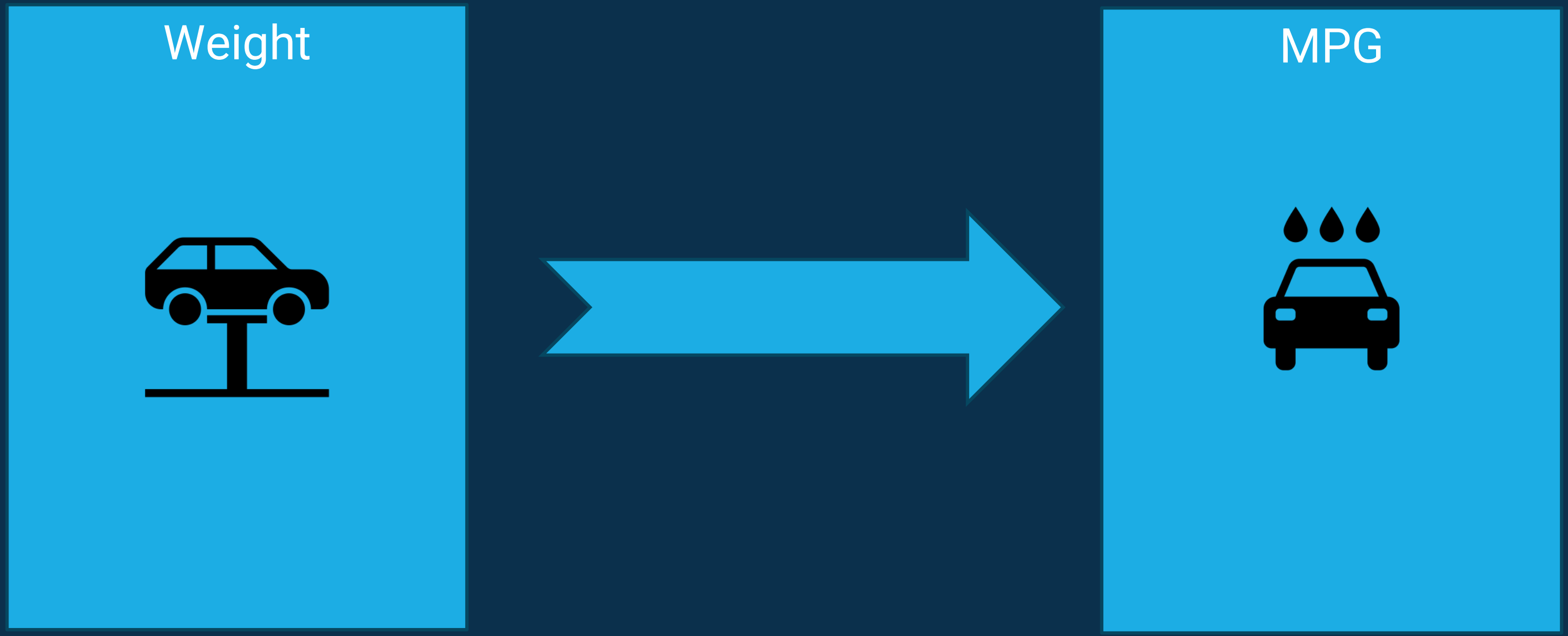
Learning techniques to find a function to **<u>predict a response</u>** variable from set of dependent variables



$$Y = \hat{f}(x)$$

Supervised Learning: Regression

Learning techniques to find a function to predict a **continuous response variable**
from a set of dependent variables

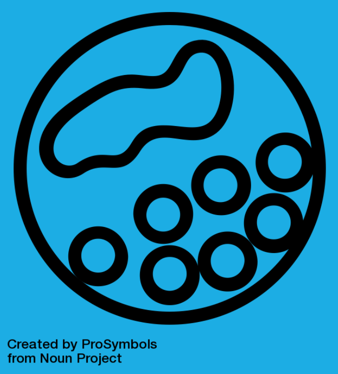


$$MPG = f(Weight)$$

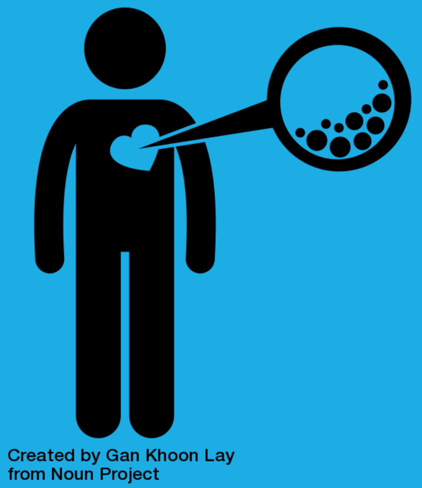
Supervised Learning: Classification

Learning techniques to find a function to predict a **categorical response variable** from a set of dependent variables

Cholesterol Level



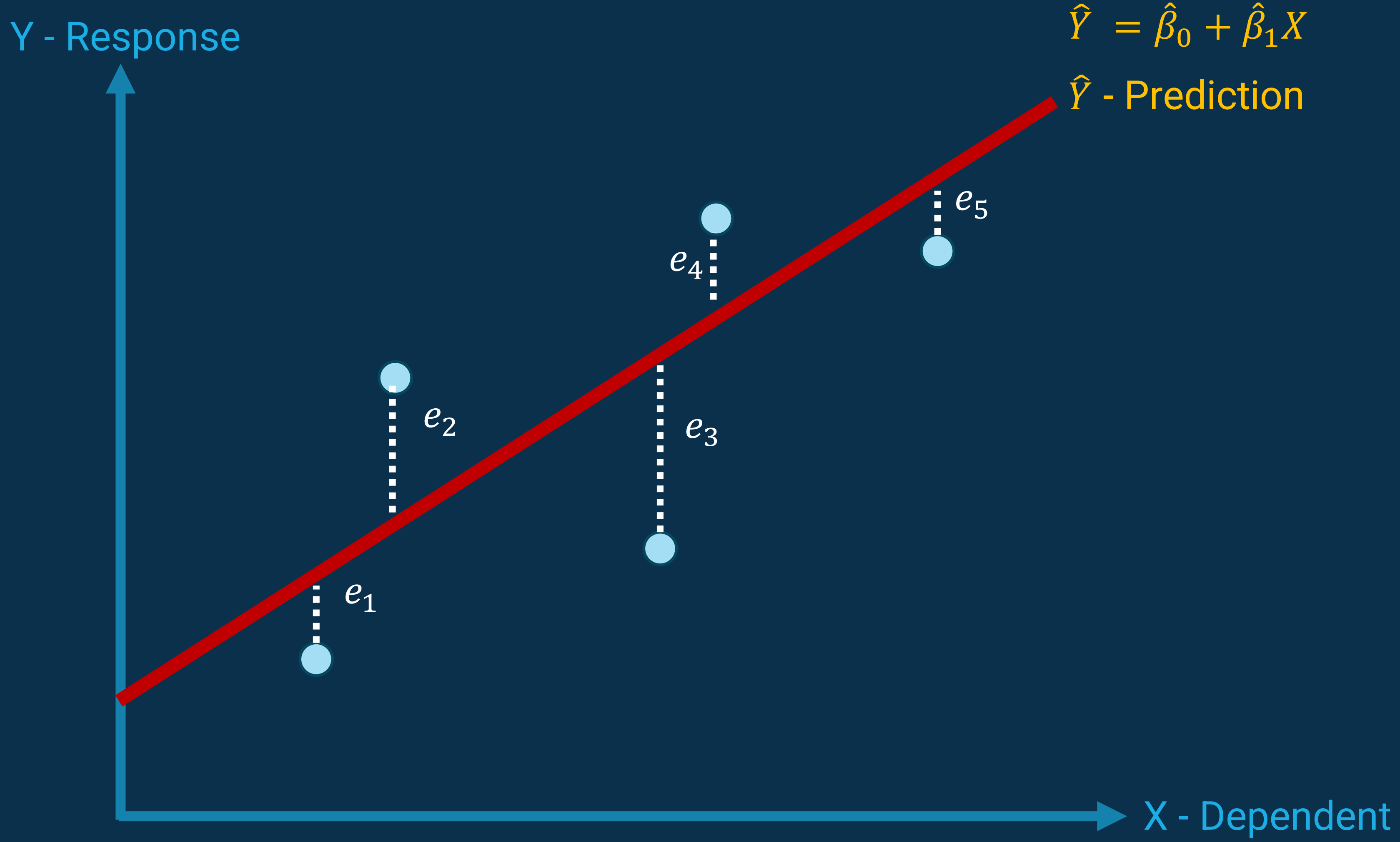
Get a Heart Disease or Not



$$\text{Heart Disease}_{(Yes\ or\ No)} = f(\text{Cholesterol Level})$$

Regression

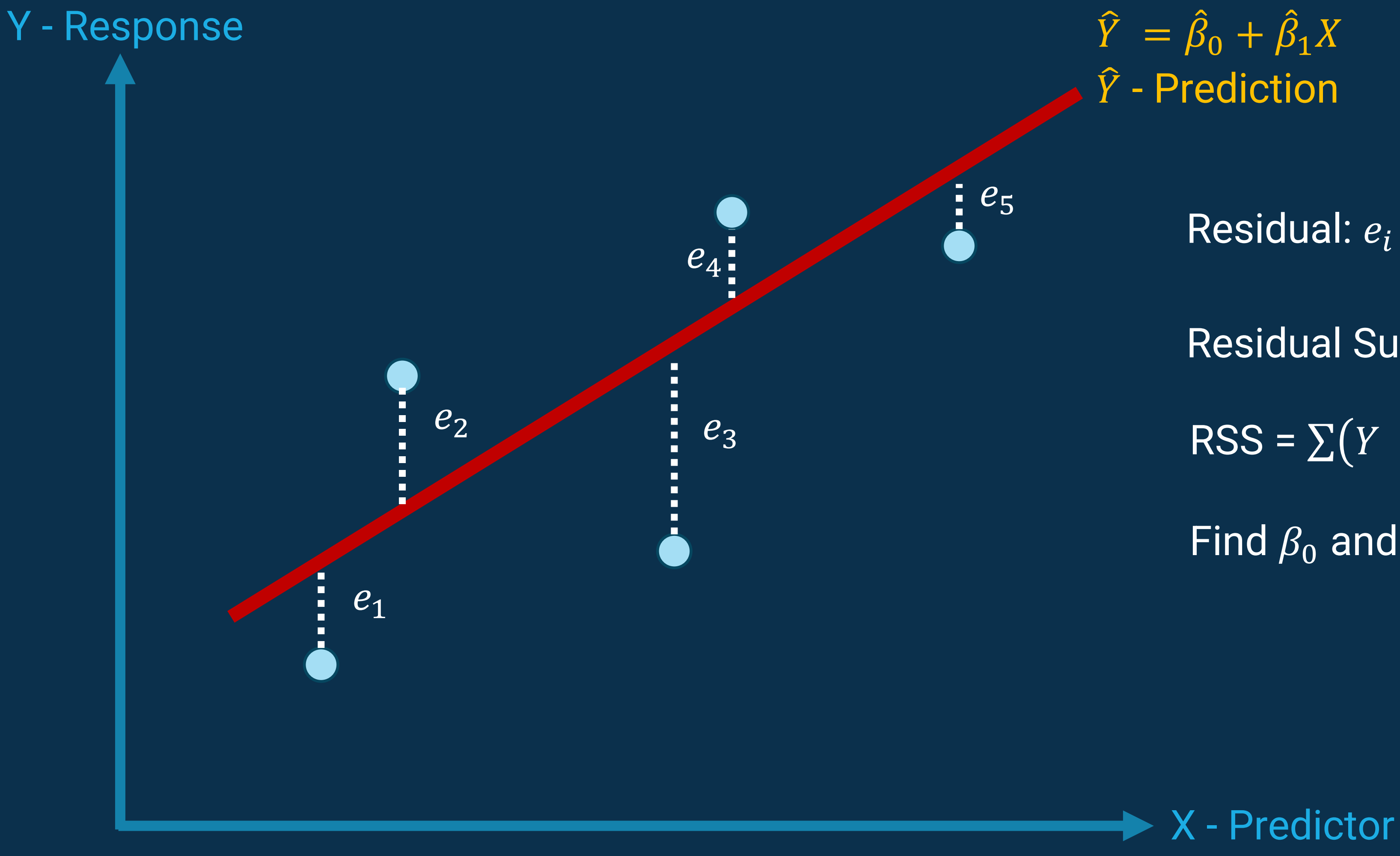
Linear Regression



X	Y
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4
x_5	y_5

Simple Linear Regression Model: $Y = \beta_0 + \beta_1 X + \text{Error}$

β_0 - Response when X = 0
β_1 - Increase of Response when X increase by 1-unit

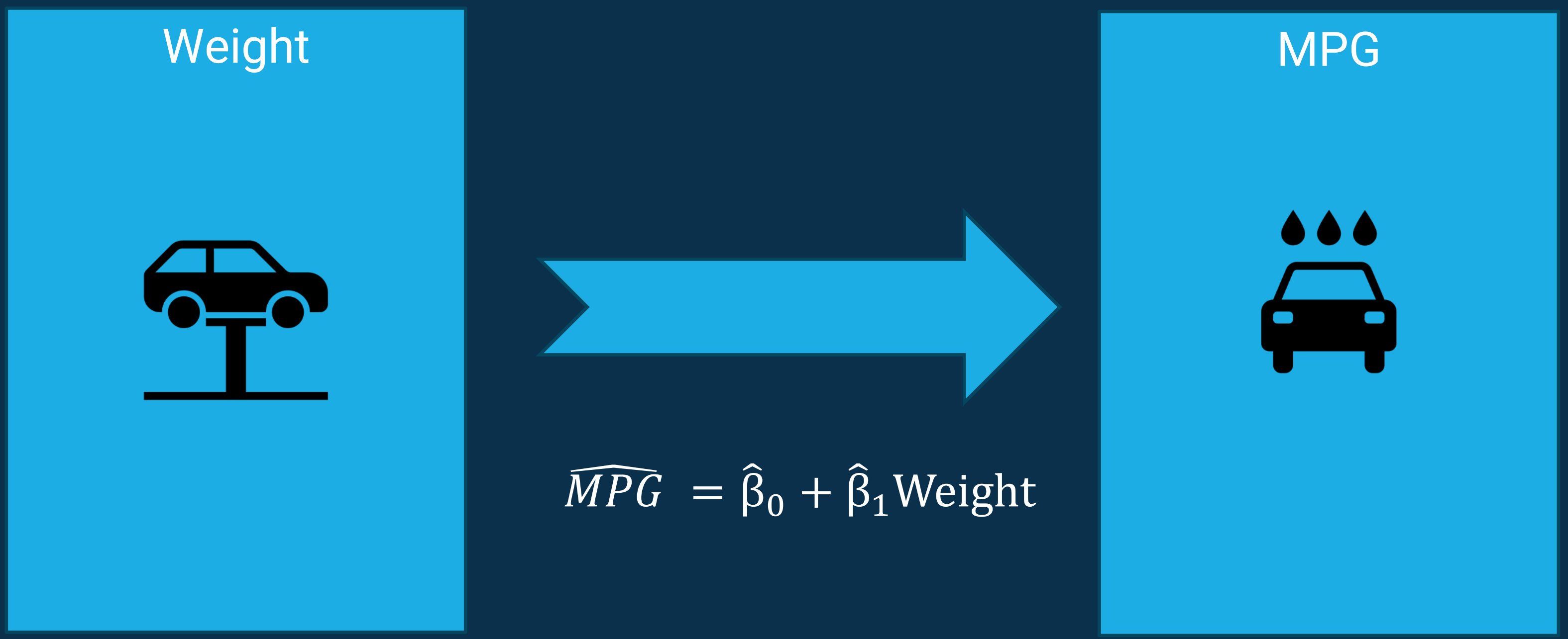


Residual: $e_i = y_i - \hat{y_i}$

Residual Sum of Square: RSS = $e_1^2 + e_2^2 + e_3^2 + e_4^2 + e_5^2$

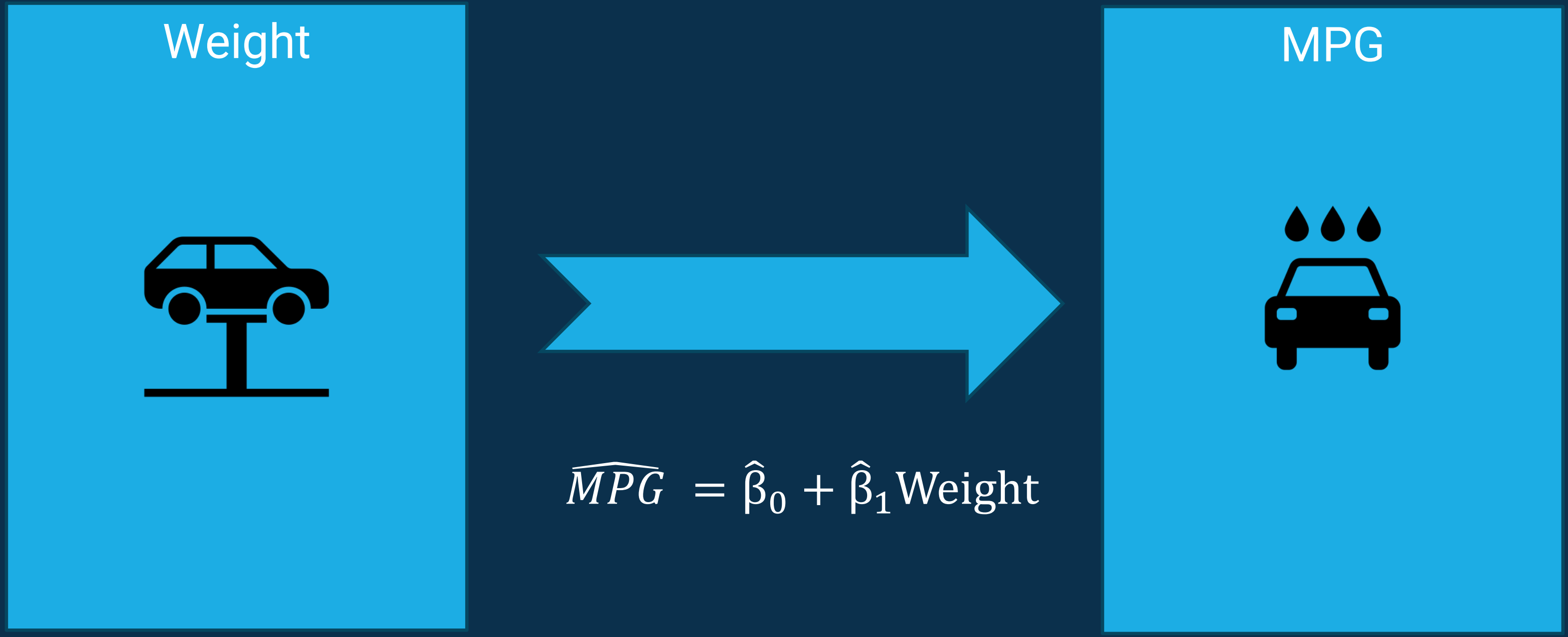
RSS = $\sum(Y - \hat{Y})^2$

Find β_0 and β_1 which minimize the RSS



$$\widehat{MPG} = \hat{\beta}_0 + \hat{\beta}_1 \text{Weight}$$

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4



$$\widehat{MPG} = \hat{\beta}_0 + \hat{\beta}_1 \text{Weight}$$

```
lm(formula = mpg ~ wt ,data = mtcars)
```

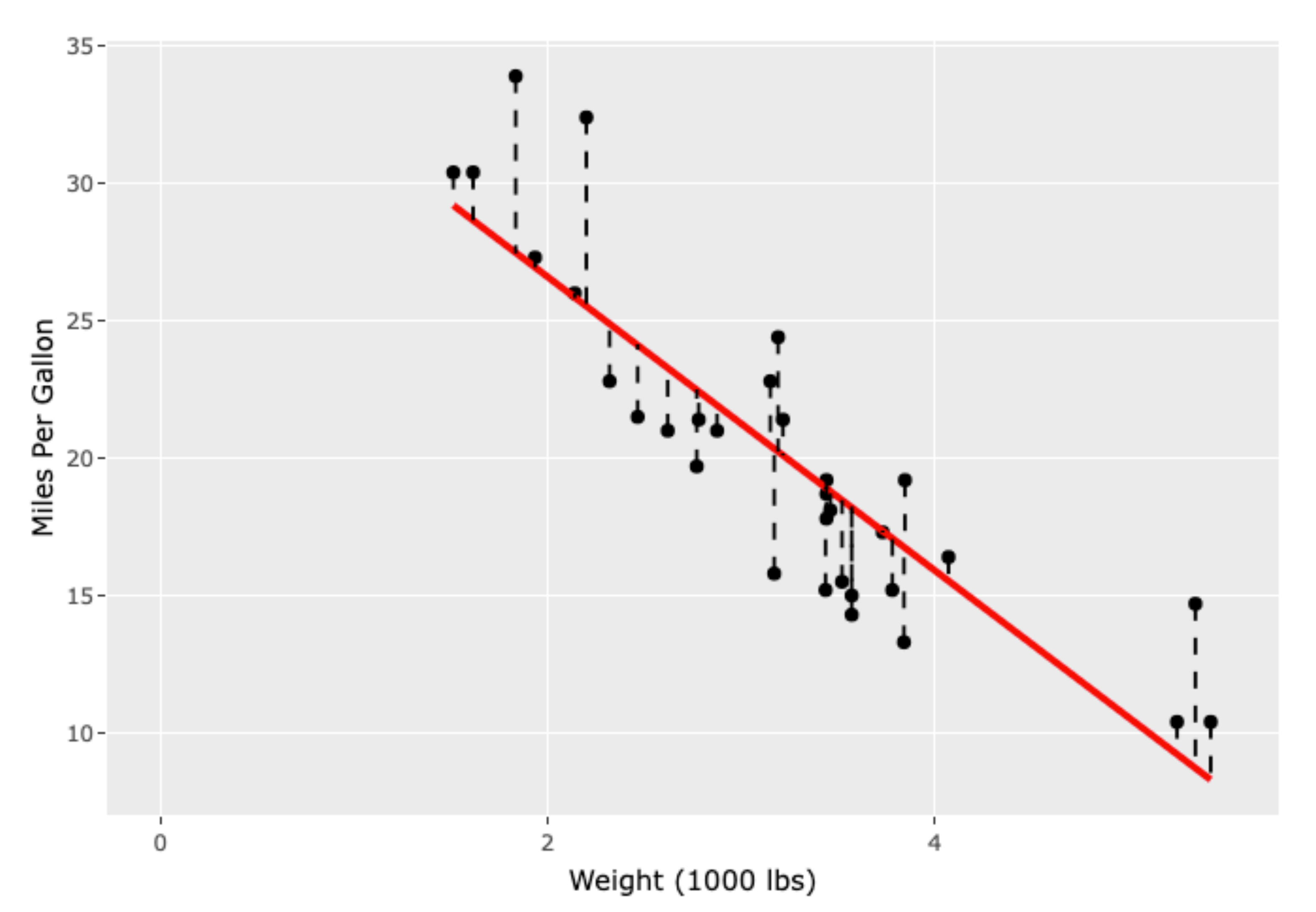
```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
wt           -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$\hat{\beta}_0 = 37.28$
$\hat{\beta}_1 = -5.34$

$$\widehat{mpg} = 37.28 - 5.34\, Weight$$

p-value will provide statistical significance
β_1 – Weight is a significant variable in estimating MPG

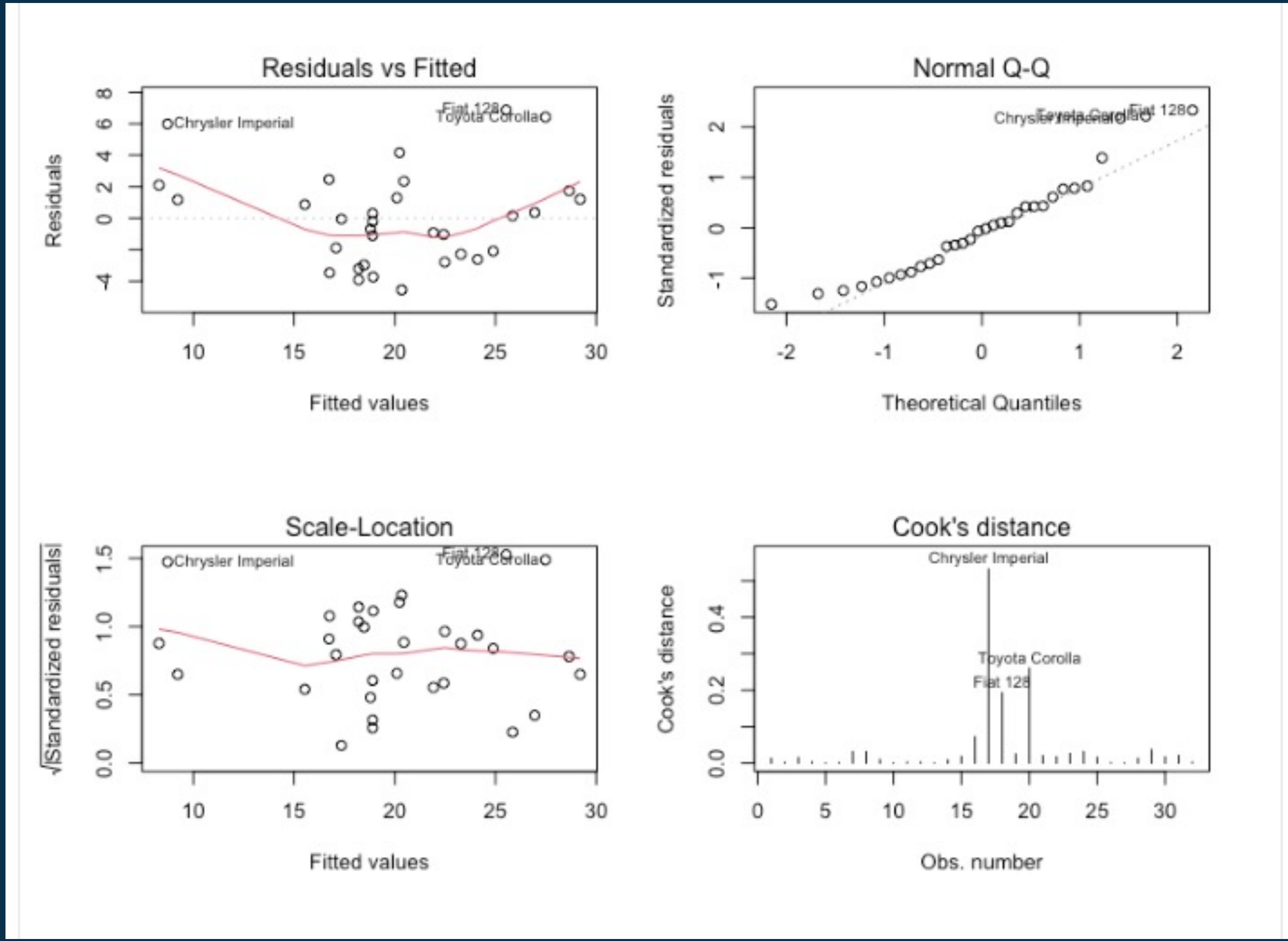
$$\widehat{mpg} = 37.28 - 5.34\,Weight$$



Linear Regression Assumptions

Assumptions

- Linearity of Data
- Constant Variance of residuals
- Normality of Residuals
- No Outliers



Linear Regression: Evaluation Model Performance

R^2 - Coefficient of Determination = 0.75

- How much variability of response is explained by the input variables
- $\mathbf{0 \leq R^2 \leq 1}$
- Higher the better

Adjusted R^2 = 0.74

- Panelize for additional input variables
- $\mathbf{R^2_{adj} \leq 1}$
- Higher the better

```
Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7446
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

RMSE – Root mean squared error = 3.04

- Lower the Better

Multiple Linear Regression

$$MPG = \beta_0 + \beta_1\, Weight + \beta_2\, Cylinders + \beta_3\, Rear\_axle\_Ratio + Error$$

```
multiple_linear_regression = lm(formula = mpg ~ wt + cyl + drat ,data = mtcars)
summary(multiple_linear_regression)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  39.7677     6.8729   5.786 3.26e-06 ***
wt           -3.1947     0.8293  -3.852 0.000624 ***
cyl          -1.5096     0.4464  -3.382 0.002142 **
drat         -0.0162     1.3231  -0.012 0.990317
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.613 on 28 degrees of freedom
Multiple R-squared:  0.8302,	Adjusted R-squared:  0.812
F-statistic: 45.64 on 3 and 28 DF,  p-value: 6.569e-11
```

$$R^2 = 0.83$$

$$R^2_{adj} = 0.81$$

$$\widehat{MPG} = 39.76 - 3.19\, Weight - 1.51\, Cylinders - 0.016\, Rear\_axle\_Ratio$$

Model Selection: Multiple Linear Regression

Backward Elimination

This method starts with the full model and eliminates unimportant predictor variables from the model.

```
library(MASS)
full_model_linear_regression = lm(formula = mpg ~ . ,data = mtcars)
backward_model = stepAIC(full_model_linear_regression, direction = "backward", trace = FALSE)
summary(backward_model)
```

```
##
## Call:
## lm(formula = mpg ~ wt + qsec + am, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4811 -1.5555 -0.7257  1.4110  4.6610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.6178     6.9596   1.382 0.177915
## wt           -3.9165     0.7112  -5.507 6.95e-06 ***
## qsec          1.2259     0.2887   4.247 0.000216 ***
## am            2.9358     1.4109   2.081 0.046716 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.459 on 28 degrees of freedom
## Multiple R-squared:  0.8497, Adjusted R-squared:  0.8336
## F-statistic: 52.75 on 3 and 28 DF,  p-value: 1.21e-11
```

$$\widehat{MPG} = 9.61 \; - 3.91\, Weight + 1.22\; qsec \; - 2.93\, Transmission$$

Model Selection: Multiple Linear Regression

Stepwise Method

Systematically add and remove predictor variables from the model.

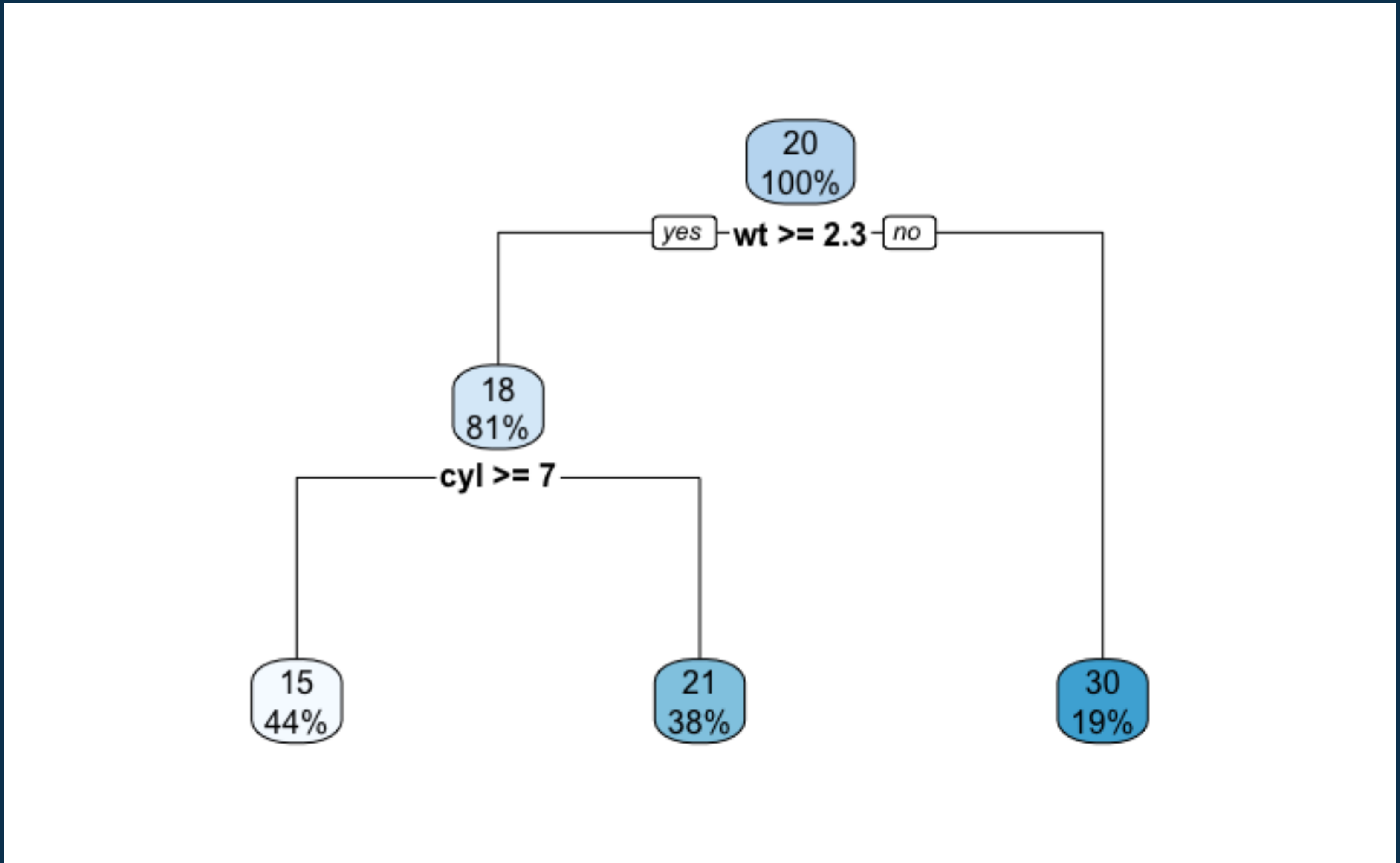
```
library(MASS)
full_model_linear_regression = lm(formula = mpg ~ . ,data = mtcars)
stepwise_model = stepAIC(full_model_linear_regression, direction = "both", trace = FALSE)
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = mpg ~ wt + qsec + am, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4811 -1.5555 -0.7257  1.4110  4.6610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.6178     6.9596   1.382 0.177915
## wt           -3.9165     0.7112  -5.507 6.95e-06 ***
## qsec          1.2259     0.2887   4.247 0.000216 ***
## am            2.9358     1.4109   2.081 0.046716 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.459 on 28 degrees of freedom
## Multiple R-squared:  0.8497, Adjusted R-squared:  0.8336
## F-statistic: 52.75 on 3 and 28 DF,  p-value: 1.21e-11
```

$$\widehat{MPG} = 9.61 \; - 3.91\, Weight + 1.22\; qsec \; - 2.93\, Transmission$$

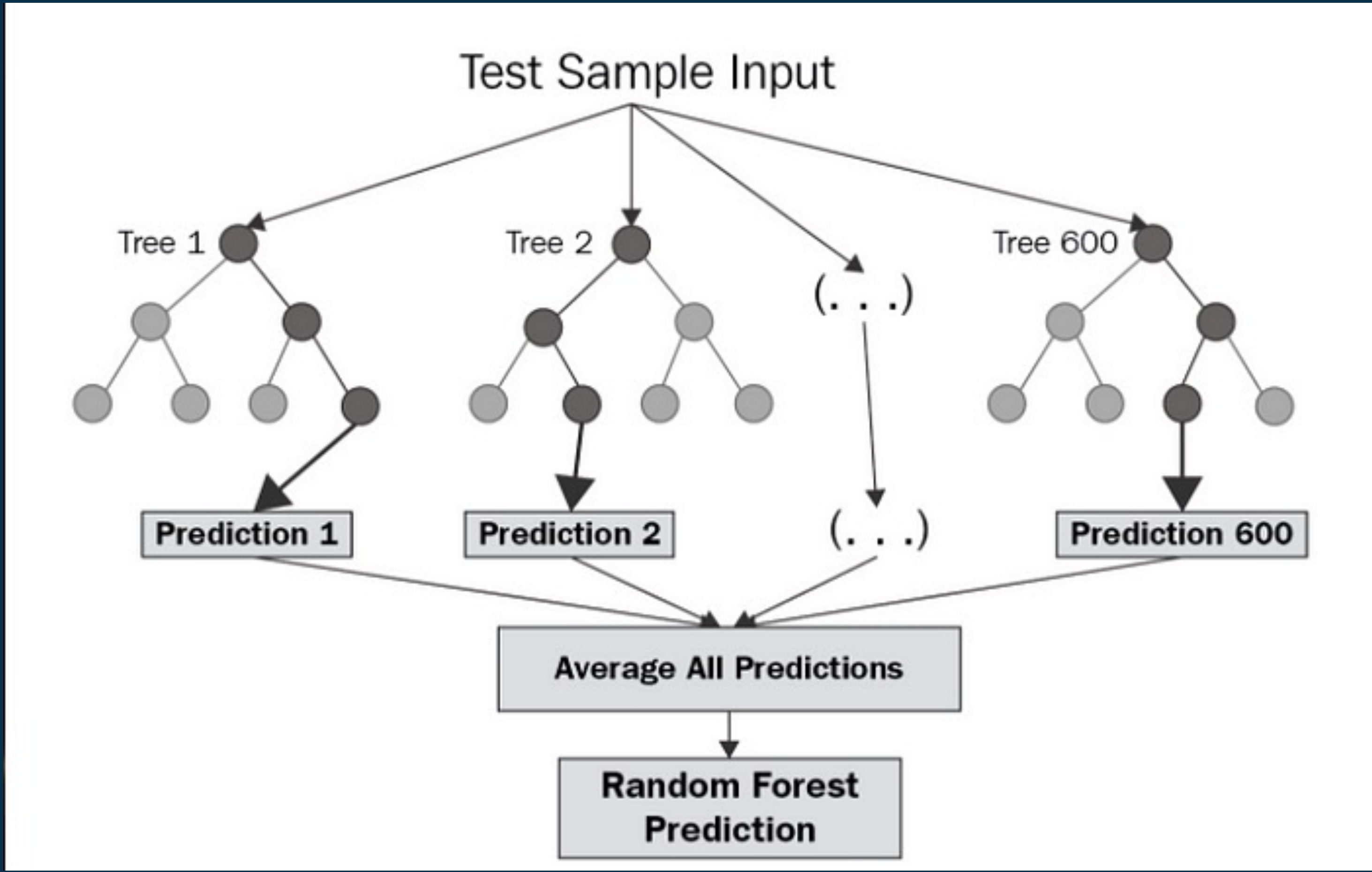
Random Forest: Regression

Regression Trees (Decision Trees)



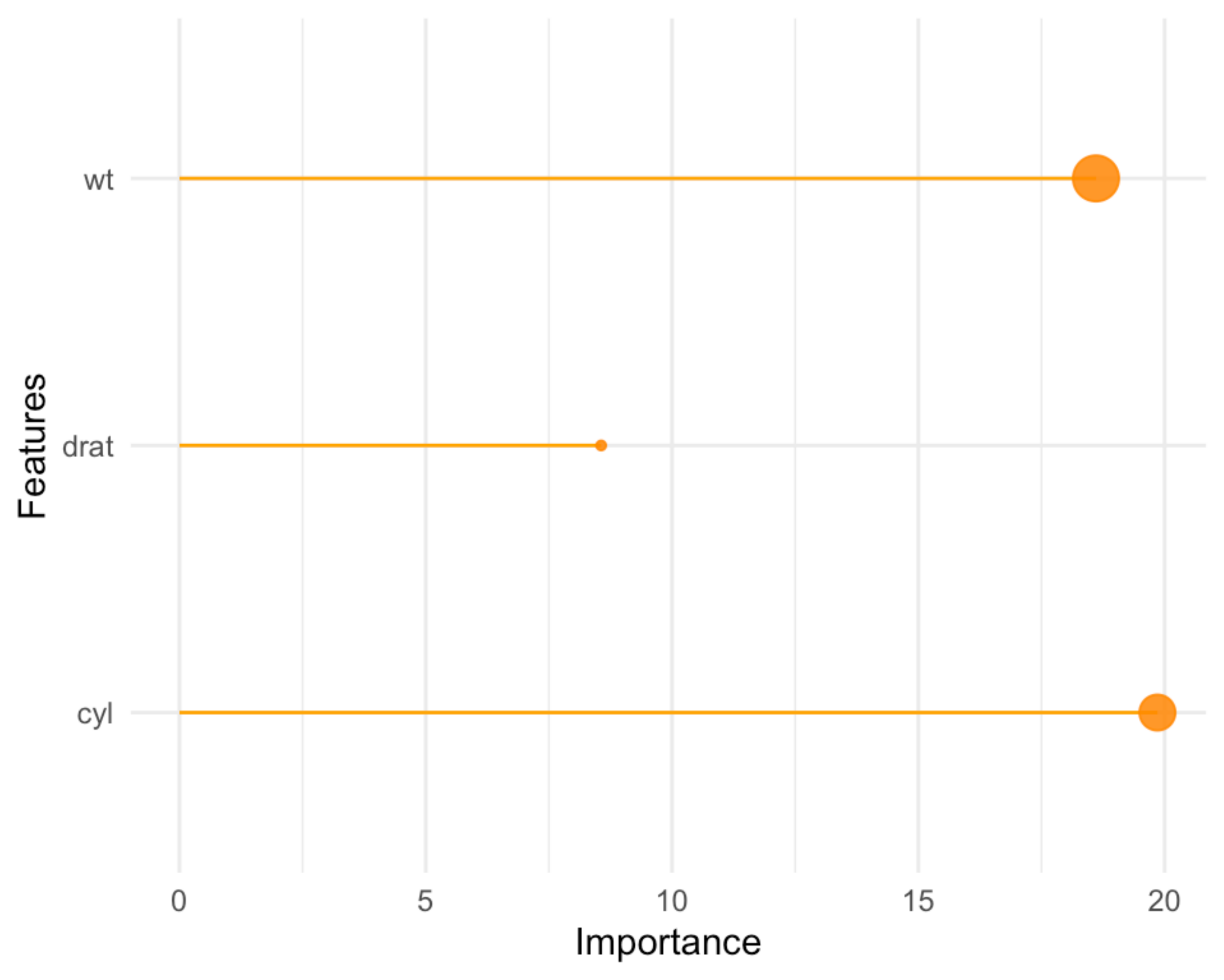
Partition the data set into decision boundaries.

Random Forest Regression use the "Bagging" Technique



Source: https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

```
library(randomForest)
Random_Forest_model = randomForest(formula = mpg ~ wt + cyl + drat,
                                   data = mtcars,
                                   ntree=500, importance=TRUE)
```

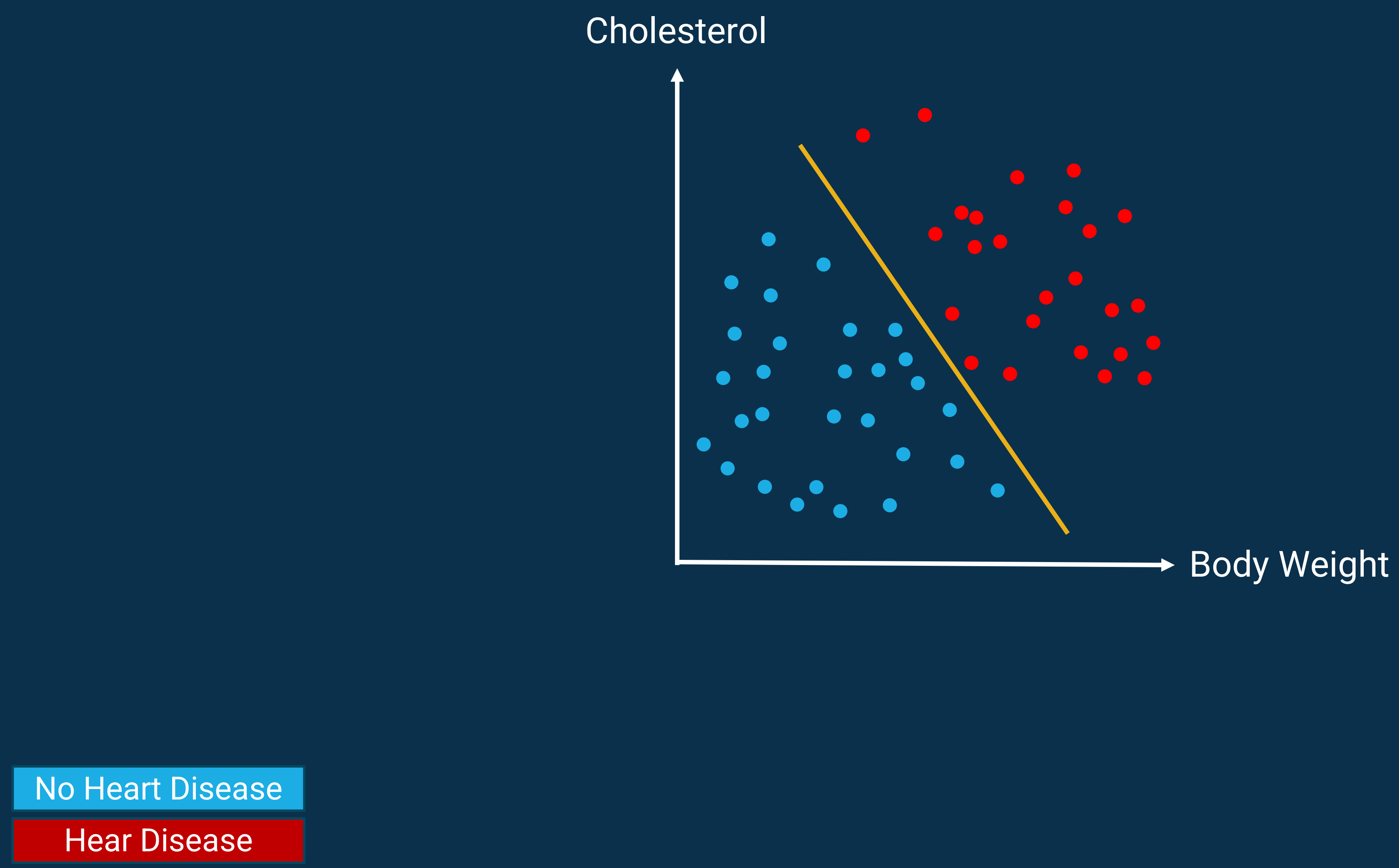


Comparison of Linear Regression and Random Forest

Linear Regression	Random Forest
Linear Model Works well when data is linear	**Non-linear Model** Work well with both linear and nonlinear data **Ensemble Technique** Take the average prediction of multiple models
Multicollinearity exists Model parameter estimates will not be accurate when there is a significant relationship between input variables (features)	**Multicollinearity is not an issue**
Model Interpretation A better interpretation of variables	**Model Interpretation is moderate** Not best as linear regression

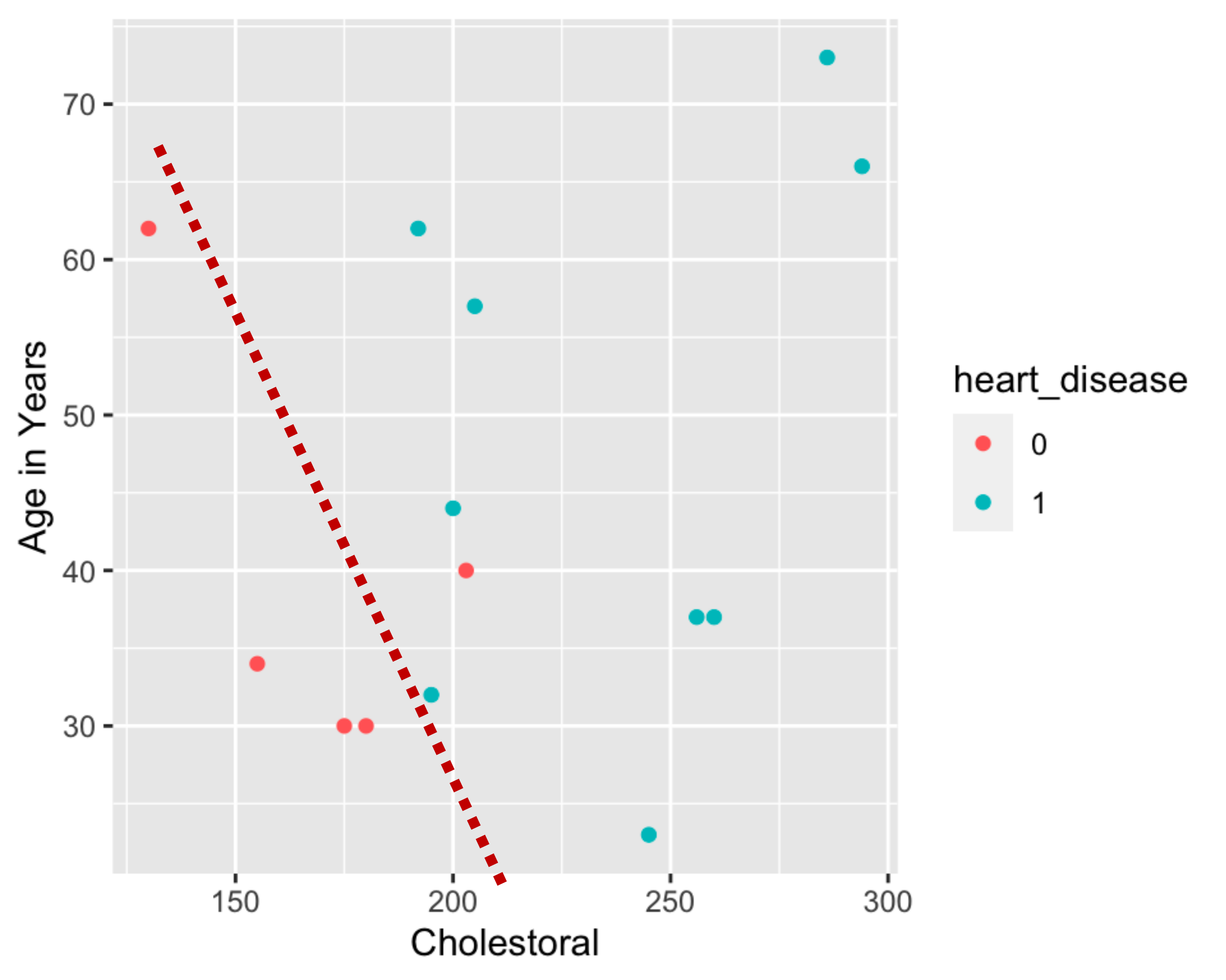
Multicollinearity – Input (predictor) variables are significantly correlated

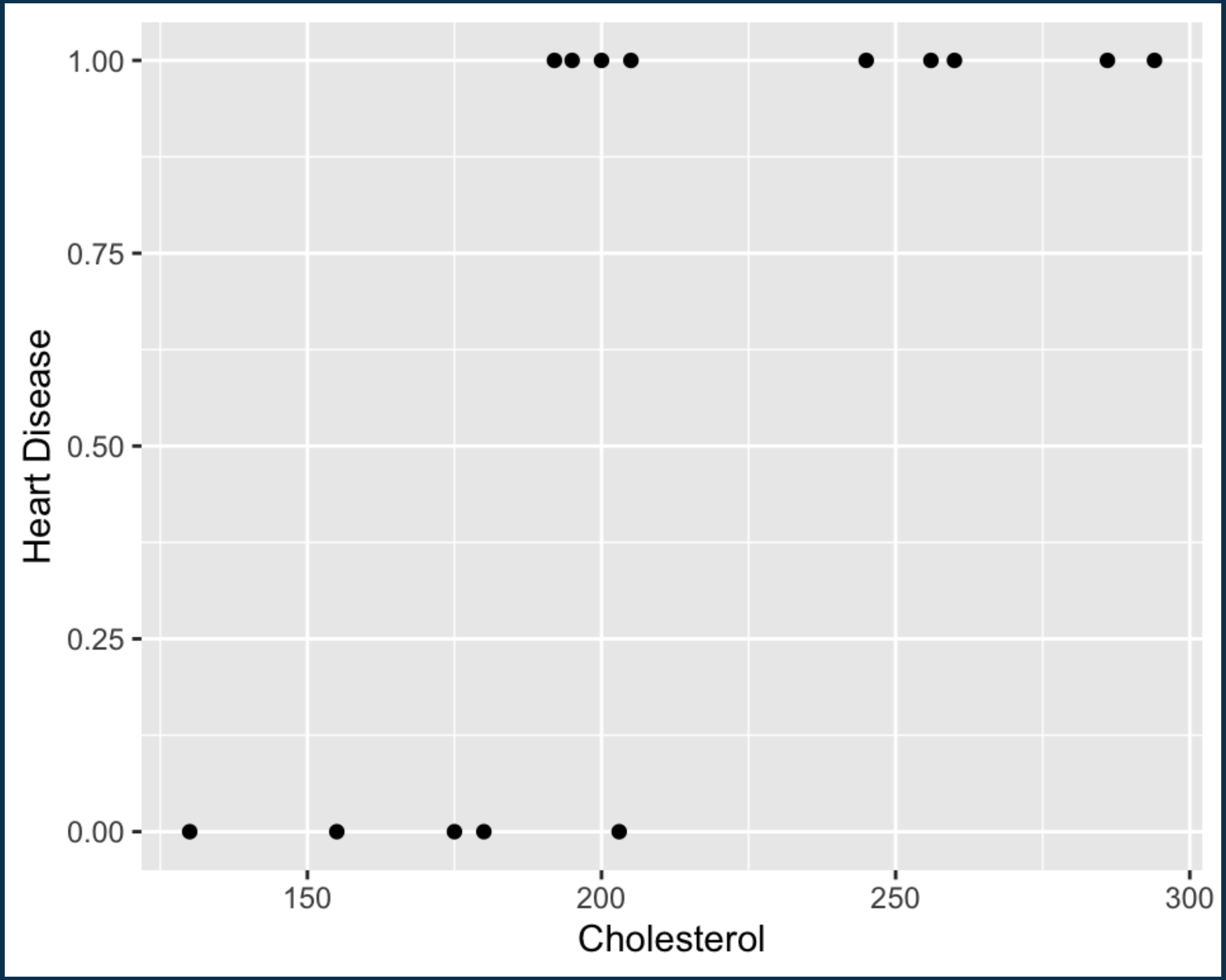
Classification


Cholesterol
Body Weight
No Heart Disease
Hear Disease

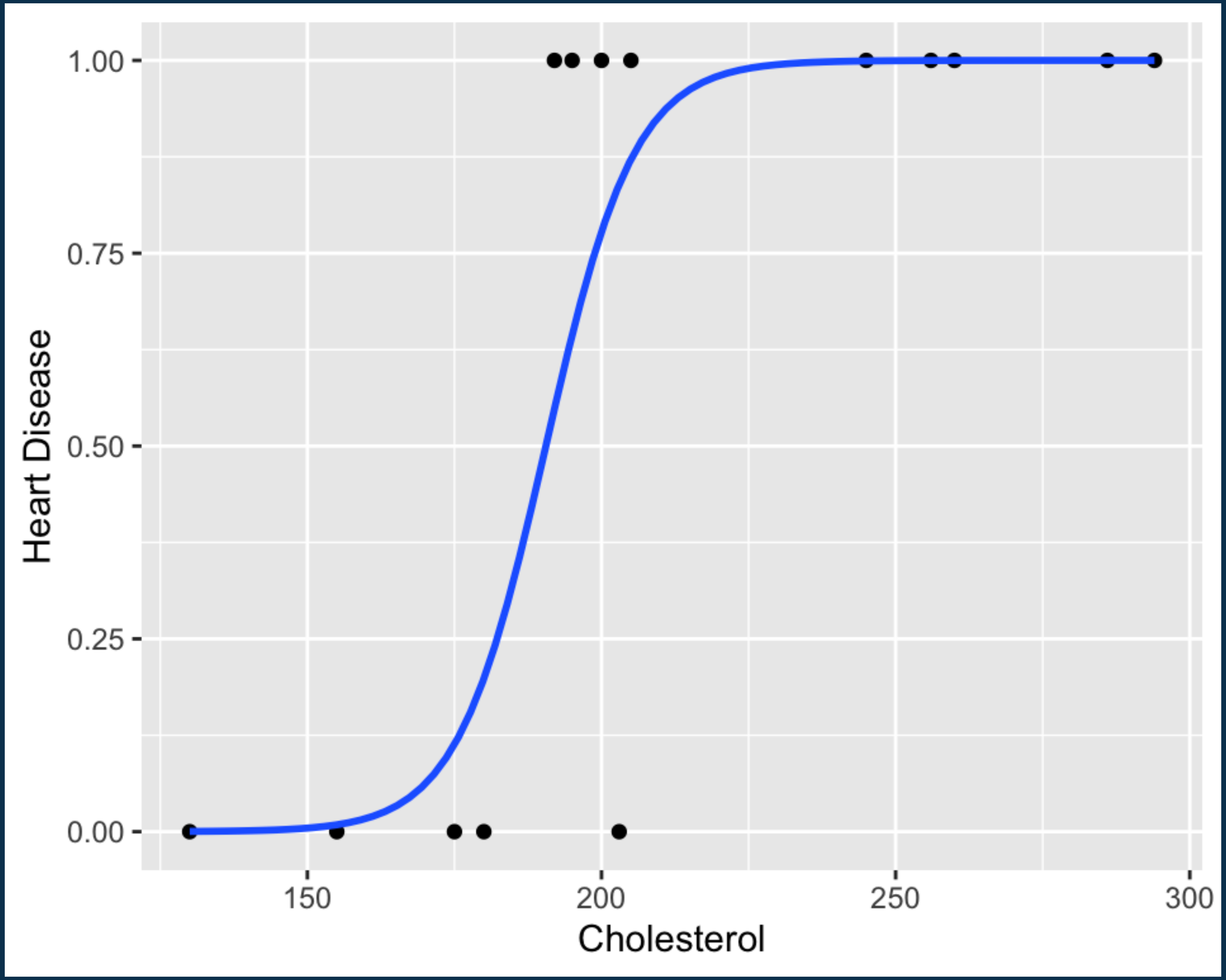
Logistic Regression

Cholesterol	Age	Heart Disease
180	30	0
200	44	1
195	32	1
245	23	1
205	57	1
130	62	0
155	34	0
260	37	1
175	30	0
286	73	1
180	30	0
200	44	1
195	32	1
245	23	1





1.00
0.75
0.50
0.25
0.00
Heart Disease
150
200
250
300
Cholesterol



Logit Function: $f(x) = \frac{\exp(x)}{1+\exp(x)}$

Y – **Probability of Getting a Heart Disease**

$$\mathbf{Logit}\ \{Y\} = \beta_0 + \beta_1 Cholesterol$$

$$\mathbf{Logit}\ Y = Log\left[\frac{Y}{1-Y}\right] = \beta_0 + \beta_1 Cholesterol$$

This is also equivalent to:

$$Y = \frac{\exp(\beta_0 + \beta_1 Cholesterol)}{1 + \exp(\beta_0 + \beta_1 Cholesterol)}$$

```
logistic_regression = glm(formula = heart_disease ~ cholesterol,
                          data = loan_df,family = 'binomial')

summary(logistic_regression)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -25.24143   17.94588  -1.407    0.160
cholesterol   0.13247    0.09289   1.426    0.154
```

$$logit\left(\hat{Y}\right) = -25.24 + 0.13\ Cholesterol$$

$$\hat{Y} = \frac{\exp(-25.24 + 0.13\ Cholesterol)}{1 + \exp(-25.24 + 0.13\ Cholesterol)}$$

$\hat{Y}$ - Probability of getting a heart disease given the Cholesterol Level

If $Y > 0.5$ $\rightarrow$ Heart Disease

We can change 0.5 to a different probability threshold.

```r
logistic_regression = glm(formula = heart_disease ~ cholesterol + age ,
                          data = loan_df,family = 'binomial')

summary(logistic_regression)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -24.31146   14.88515  -1.633    0.102
cholesterol   0.10649    0.07342   1.450    0.147
age           0.09857    0.10119   0.974    0.330
```

$$logit\left(\hat{Y}\right) = -24.3 + 0.10\ Cholesterol + 0.09\ Age$$

$$\hat{Y} = \frac{\exp(-24.3 + 0.10\ Cholesterol + 0.09\ Age)}{1 + \exp(-24.3 + 0.10\ Cholesterol + 0.09\ Age)}$$

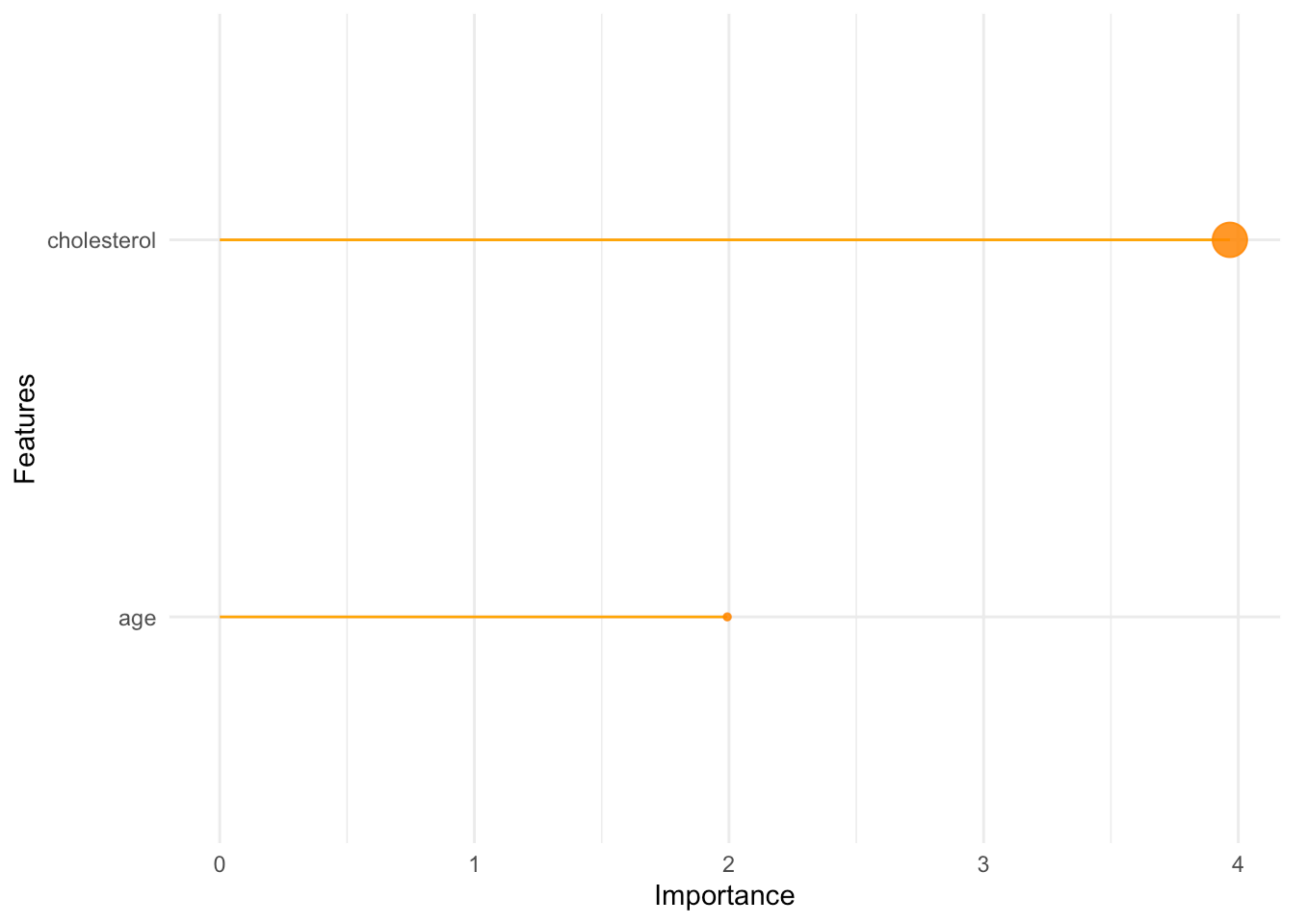
$\hat{Y}$ - Probability of approving the loan when income and age is known

- The odds of getting a heart disease will increase by 11% (i.e., $\exp(0.10) = 1.11$) when the Cholesterol level increases by 1 unit.

- The odds of getting heart disease will increase by 10% (i.e.,$\exp(0.09) = 1.10$) when the age increases by 1 year.

Random Forest: Classification

```r
library(randomForest)

Random_Forest_Classifier = randomForest(as.factor(heart_disease) ~ cholesterol + age,
                                        data=heart_df, importance = TRUE, ntree=500)
```



Comparison of Logistic Regression and Random Forest

Logistic Regression	Random Forest Classification
Linear Model Works well when data is linear	**Non-linear Model** Work well with both linear and nonlinear data **Ensemble Technique** Take the average prediction of multiple models
Works with two classes - Multinomial logistic regression shall be used to for than 2 classes	**Can extend for multiple classes**
Model Interpretation A better interpretation of variables	**Model Interpretation is moderate** Not best as the logistic regression

Evaluate Model Performance in Classification

Confusion Matrix

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

$$\text{Accuracy} = \frac{TP + TN}{N}$$

$$\text{Sensitivity} = \text{True Positive Rate} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Specificity} = 1 - \text{False Positive Rate} = \frac{TN}{TN + FP}$$

N – Total Number of Observations

Evaluate Model Performance in Classification

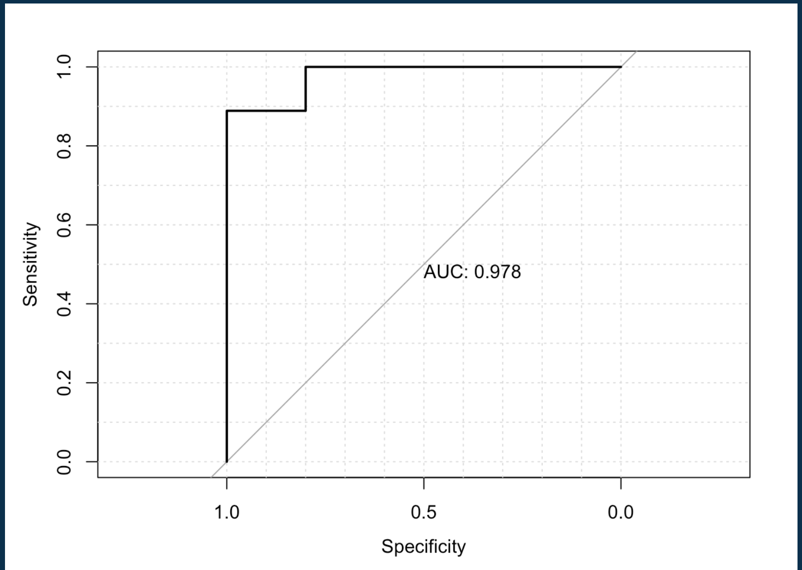
```
library(caret)

random_forest_prediction = predict(Random_Forest_Classifier)

observed_data = as.factor(heart_df$heart_disease)

confusionMatrix( random_forest_prediction, observed_data, positive = '1')
```

```
Confusion Matrix and Statistics

          Reference
Prediction 0 1
         0 3 2
         1 2 7
```

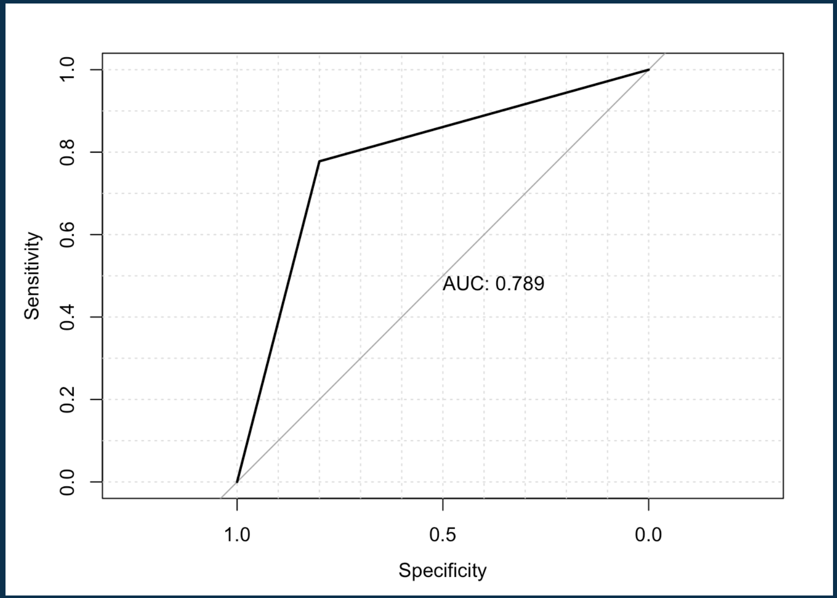
```
               Accuracy : 0.7143
                 95% CI : (0.419, 0.9161)
    No Information Rate : 0.6429
    P-Value [Acc > NIR] : 0.4007

                  Kappa : 0.3778

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.7778
            Specificity : 0.6000
         Pos Pred Value : 0.7778
         Neg Pred Value : 0.6000
             Prevalence : 0.6429
         Detection Rate : 0.5000
   Detection Prevalence : 0.6429
      Balanced Accuracy : 0.6889
```

Evaluate Model Performance in Classification

Receiver Operating Characteristic (ROC) Curve



Logistic Regression



Random Forest Classifier

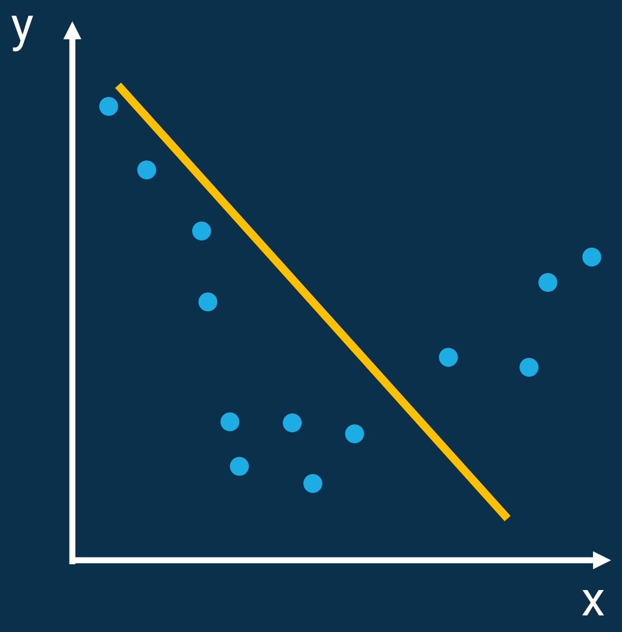
- ROC curve is a plot between the Sensitivity and Specificity of a classification model
- If the area under the curve is 0.5, we are making random guesses
- If the area under the curve is close to 1, we are making accurate predictions

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

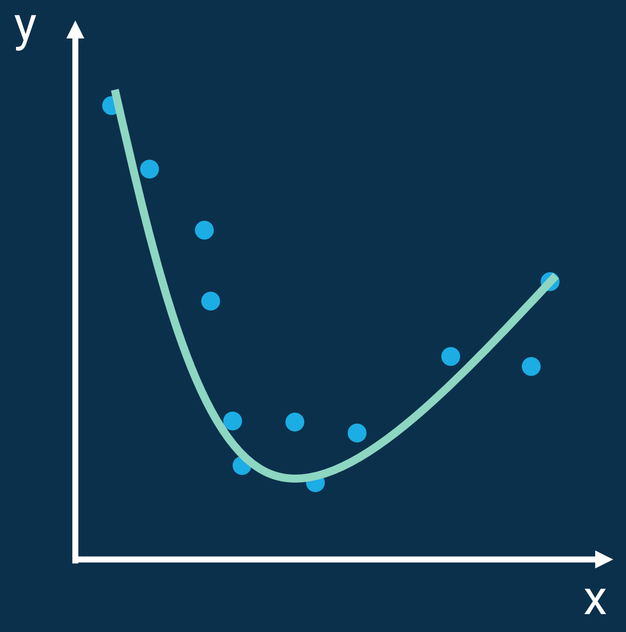
$$\text{Specificity} = \frac{TN}{TN + FP}$$

Overfitting & Cross-Validation

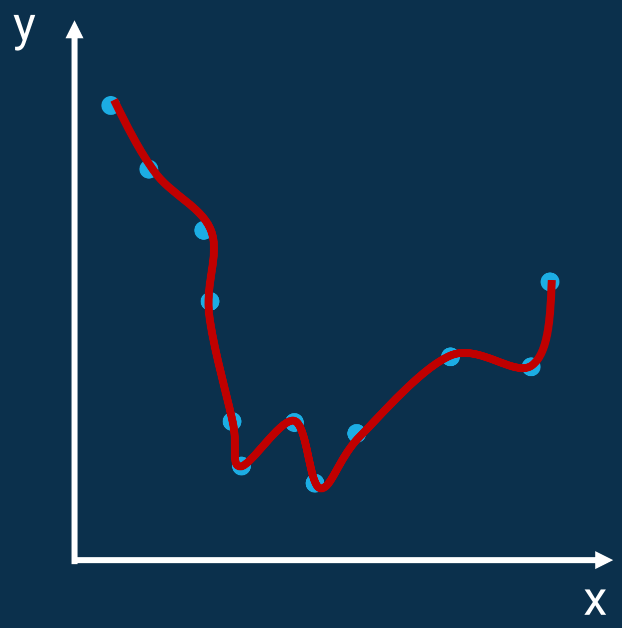
Overfitting in Regression



Underfit
– less complex model

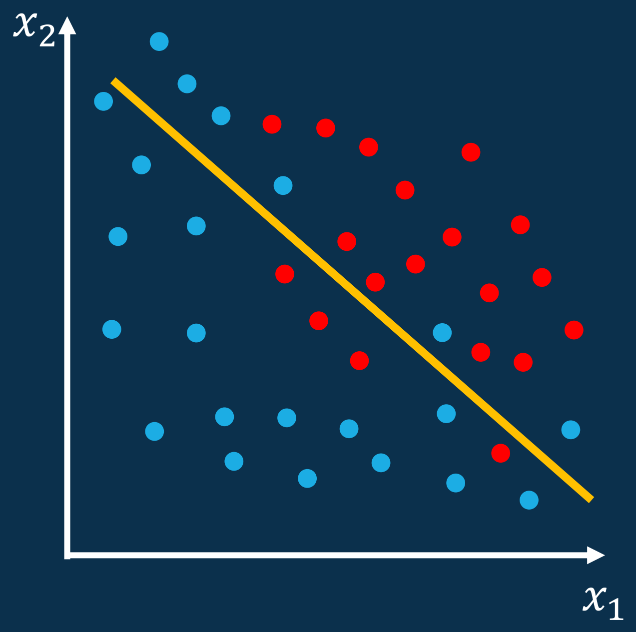


Better Fit
- moderately complex model

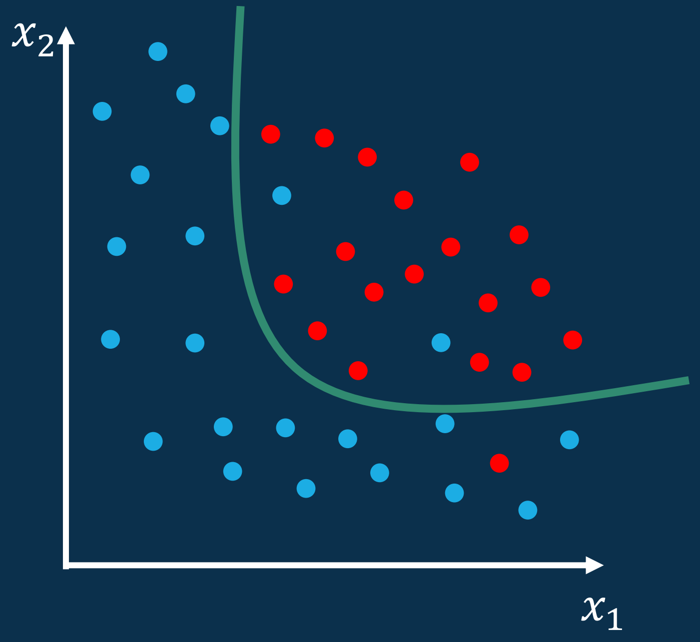


Overfit
– high complex model

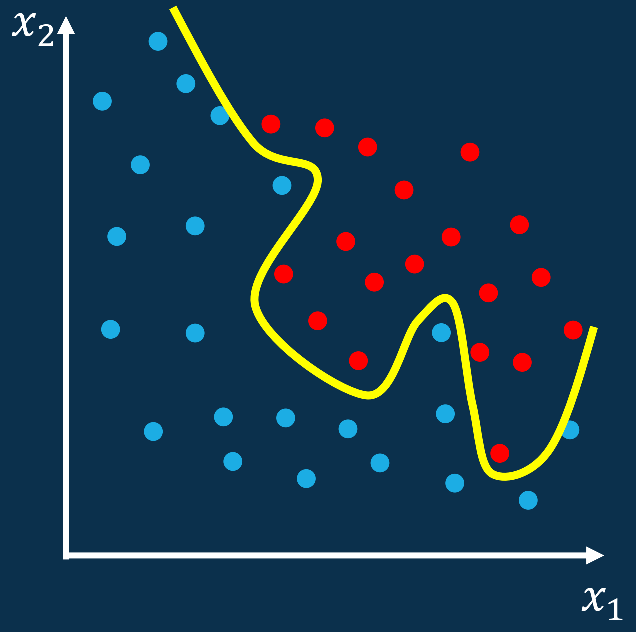
Overfitting in Classification



Underfit
– less complex model



Better Fit
- moderately complex model

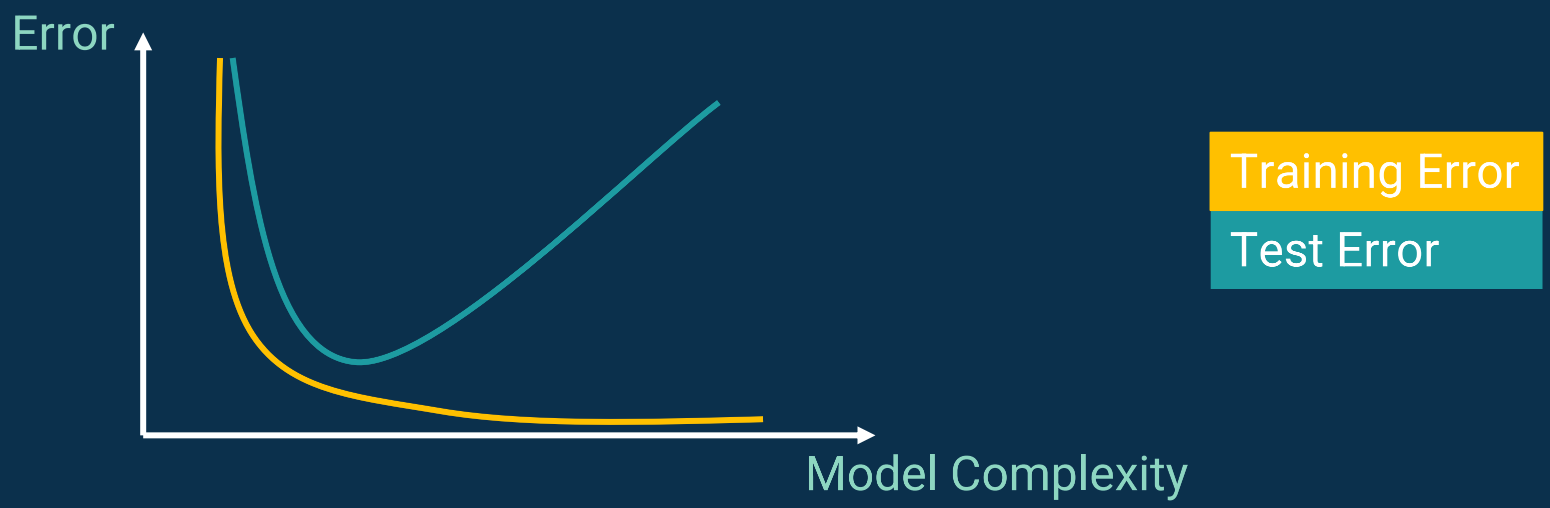


Overfit
– high complex model

How to Overcome Overfitting

Train/Test Data Split

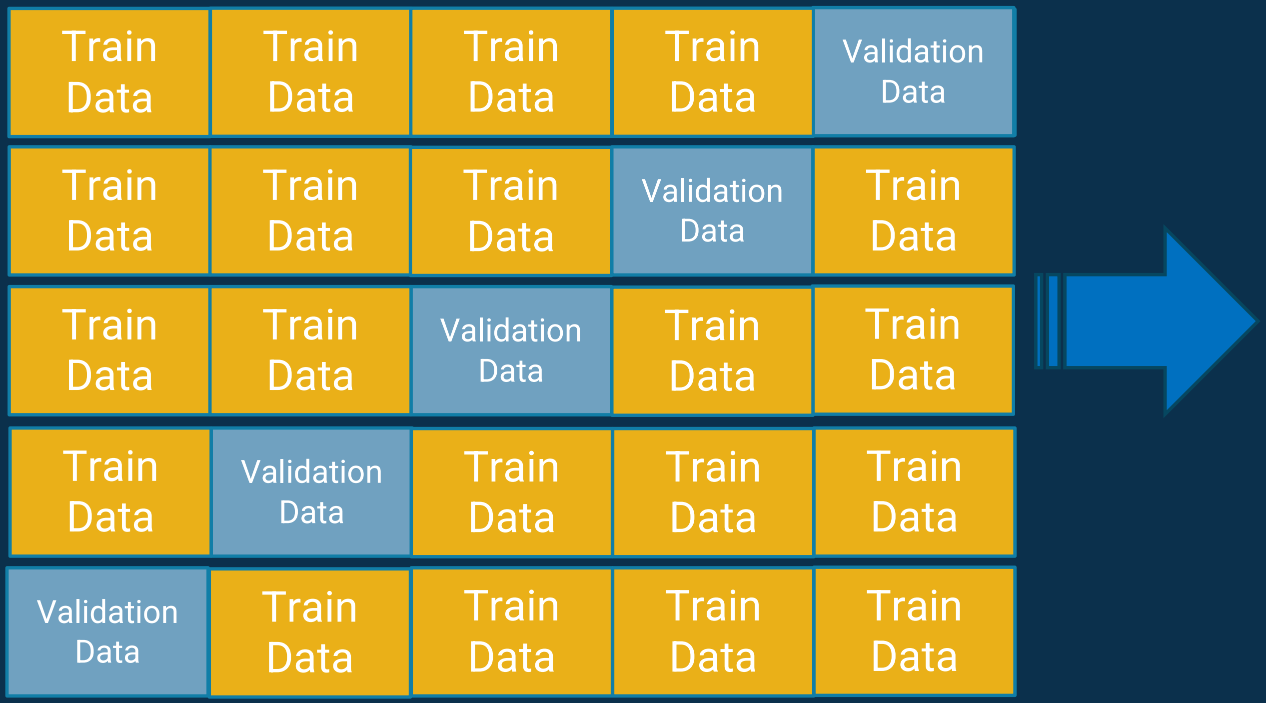
Training Data | Test Data



Error = Actual - Predicted

How to Overcome Overfitting

K-Fold Cross-Validation

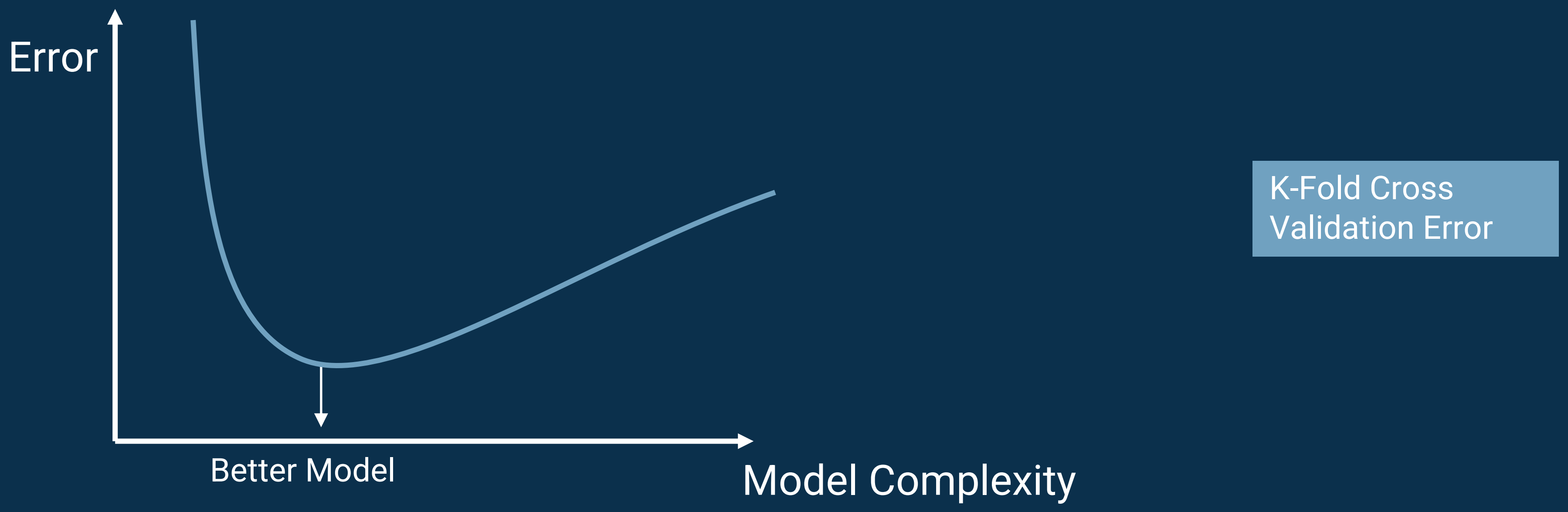


K-Fold Cross Validation Error

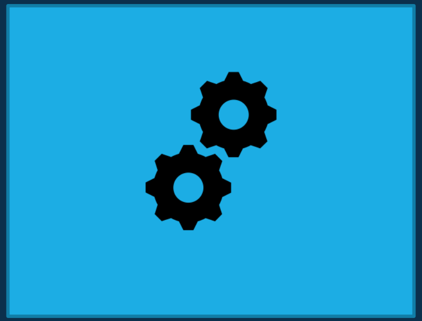
Take average of all Validation Errors

How to Overcome Overfitting

K-Fold Cross-Validation



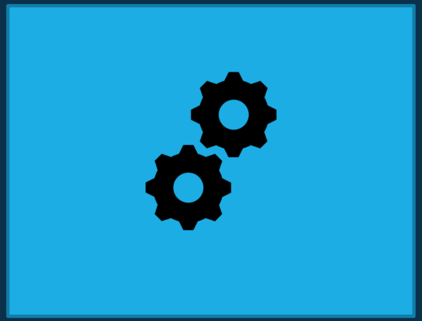
Hyperparameter Tuning



Equipment 1 Settings

Equipment 2 Settings

Hyperparameter Tuning



Model 1 Settings

Model 2 Settings

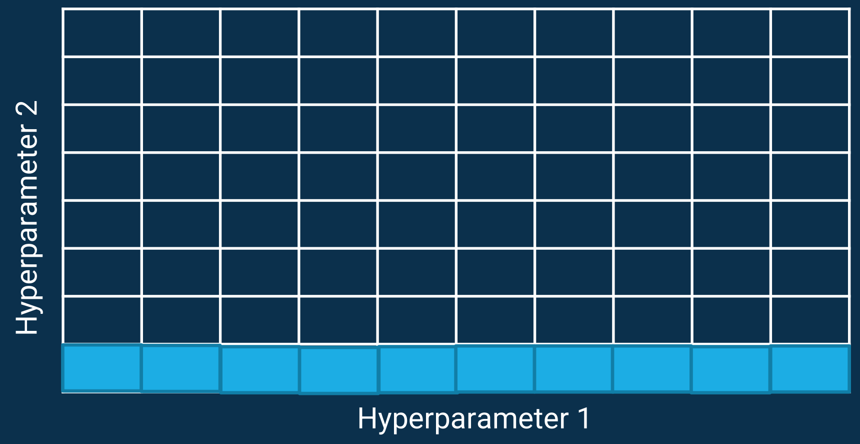
Hyperparameter Tuning

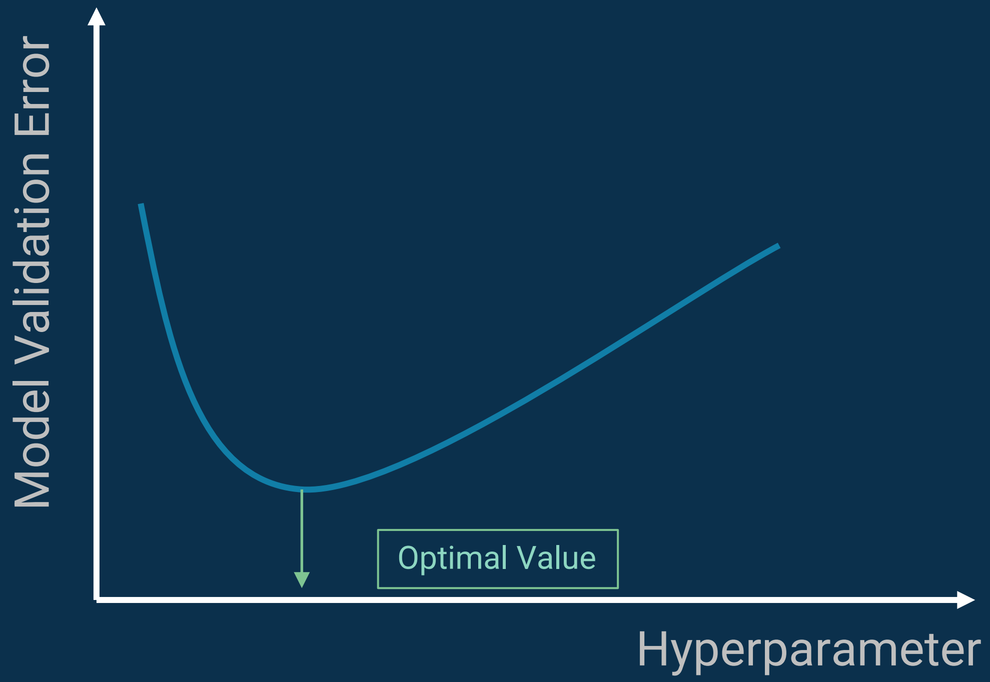
Hyperparameters

- Model specific parameters which can not be estimated with data
- Example: ntrees and mtry in Random Forest

Grid Search

- Require to find optimal values with a search





Training Data	Use to Train Model
Validation Data	• Use to control overfit • Use to tune model parameters
Test Data	Test model with unseen data

Session 2: Part 2

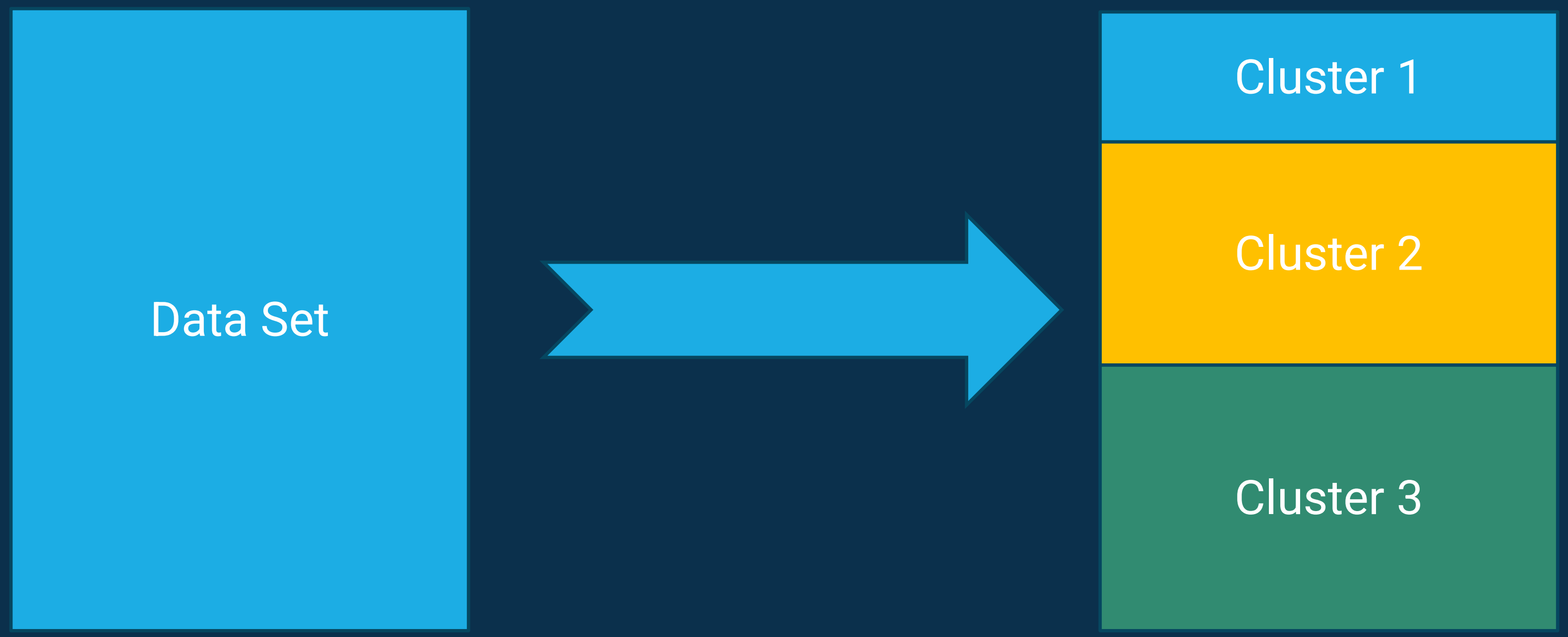
Unsupervised Learning

Workshop on Quantitative Literacy and Statistics

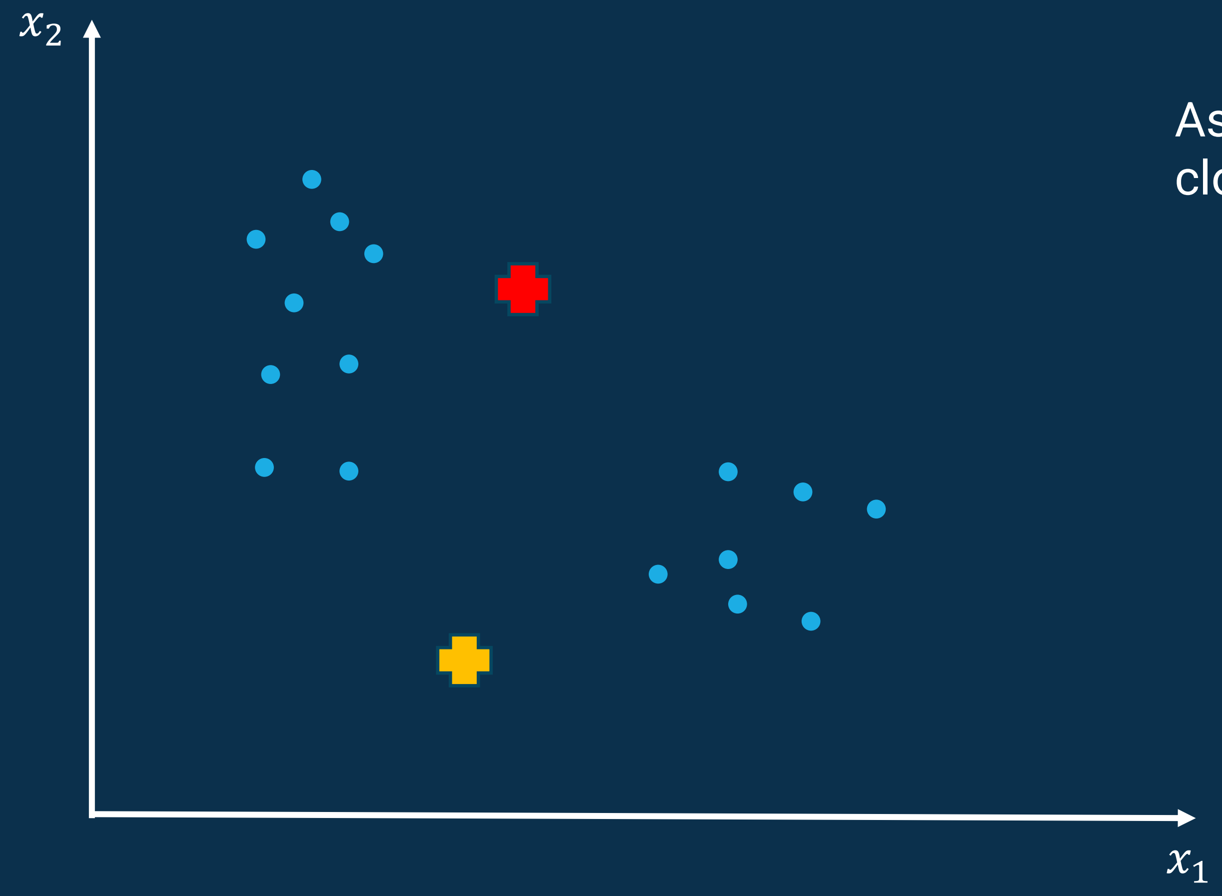
UNIVERSITY OF NEBRASKA AT OMAHA
DATA AND DECISION SCIENCES

Unsupervised Learning

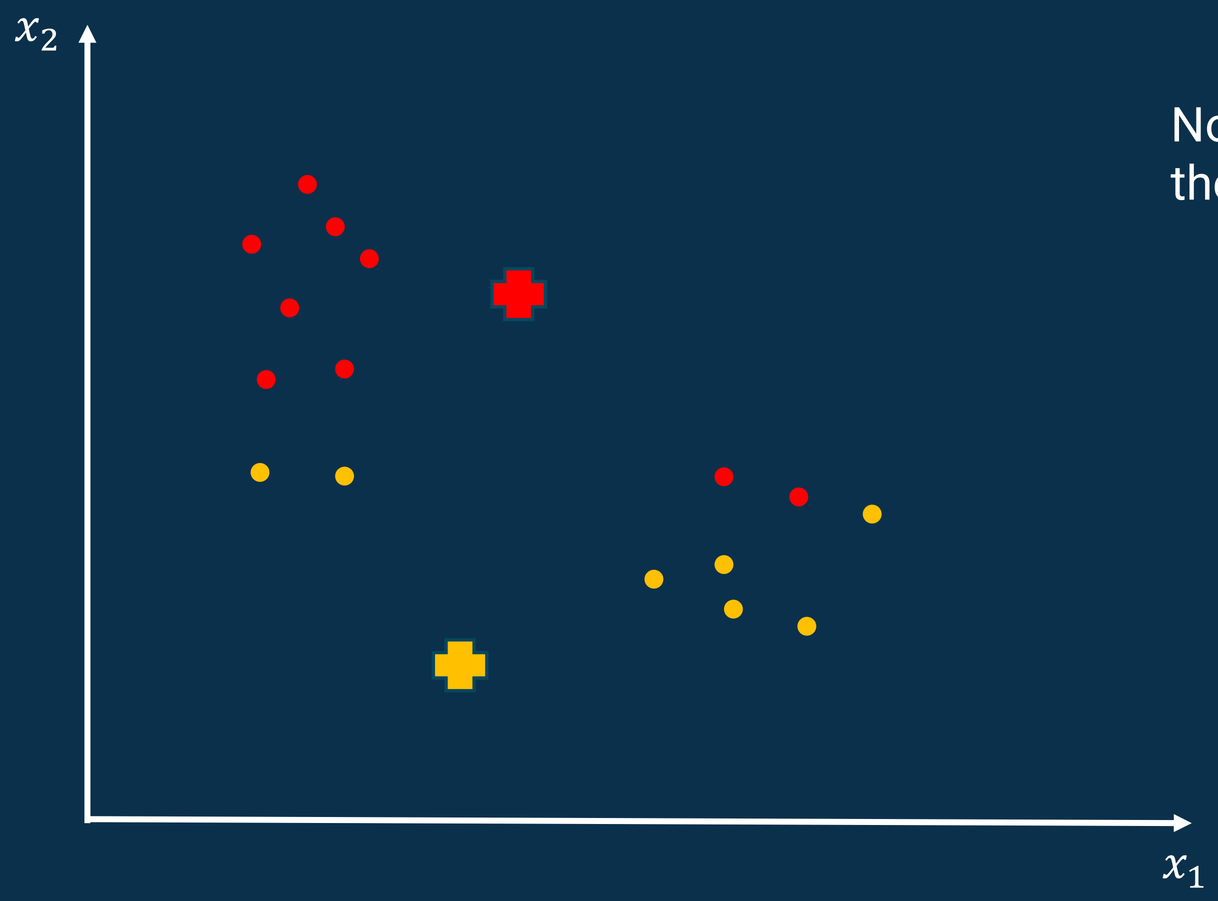
Learning techniques to find a patterns and cluster unlabeled data



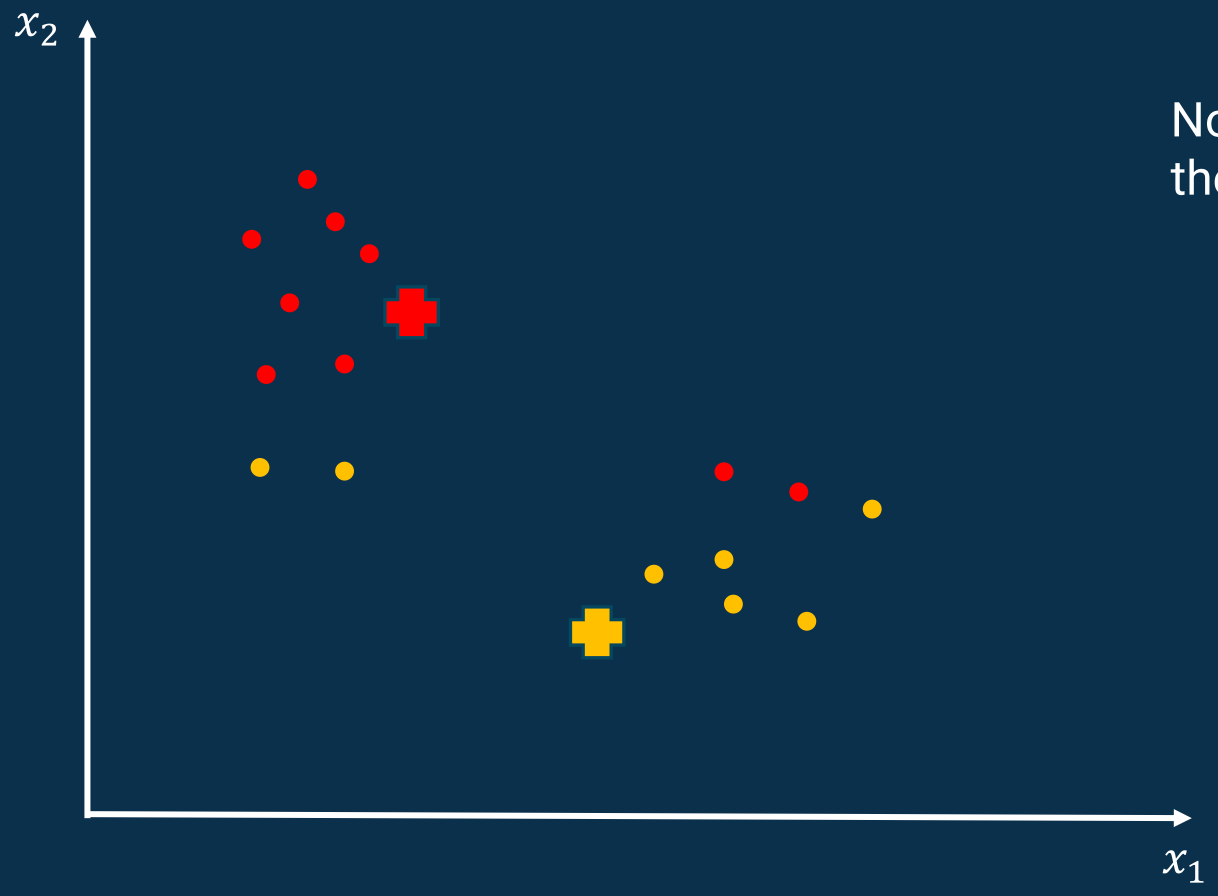
K-Means Clustering



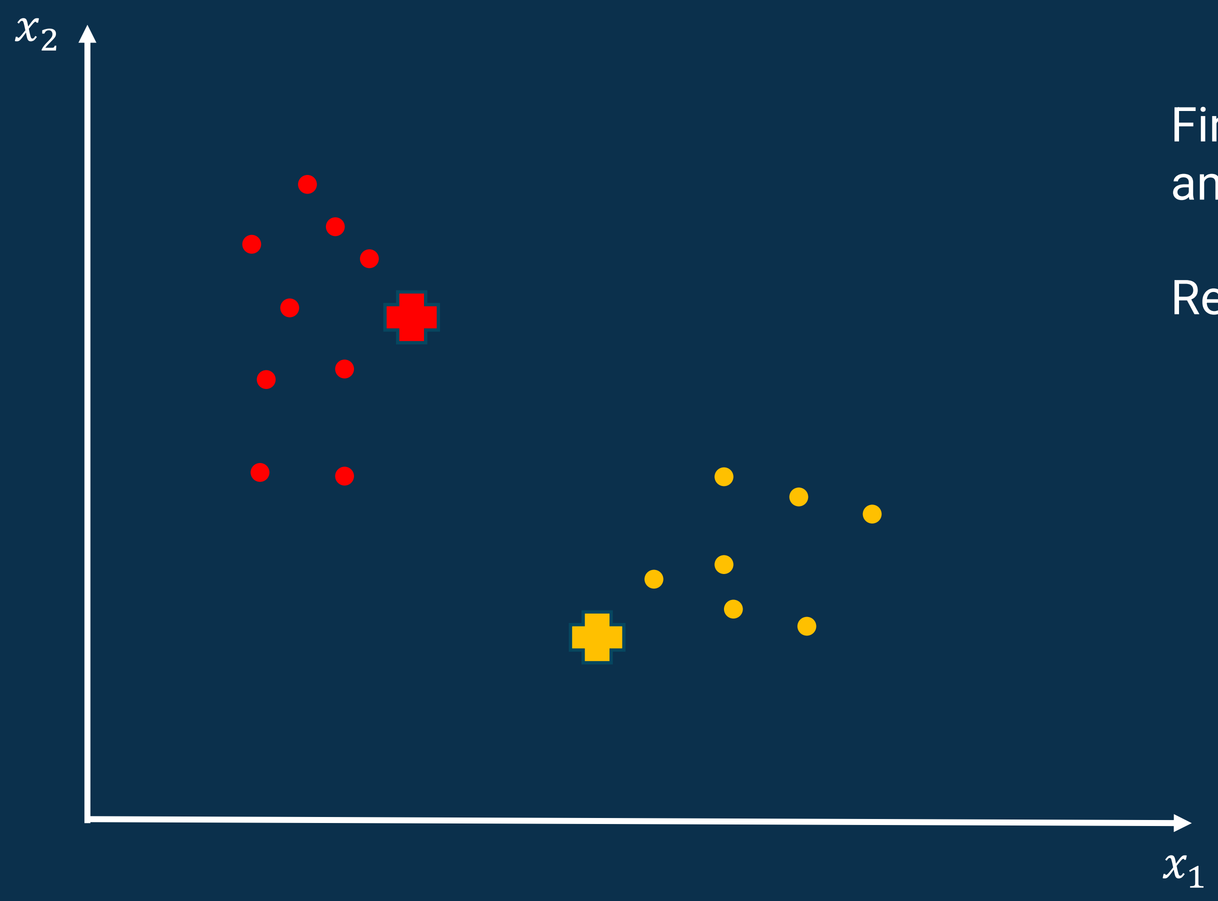
Assign data points for the closest centroid



Now update the center of the centroids

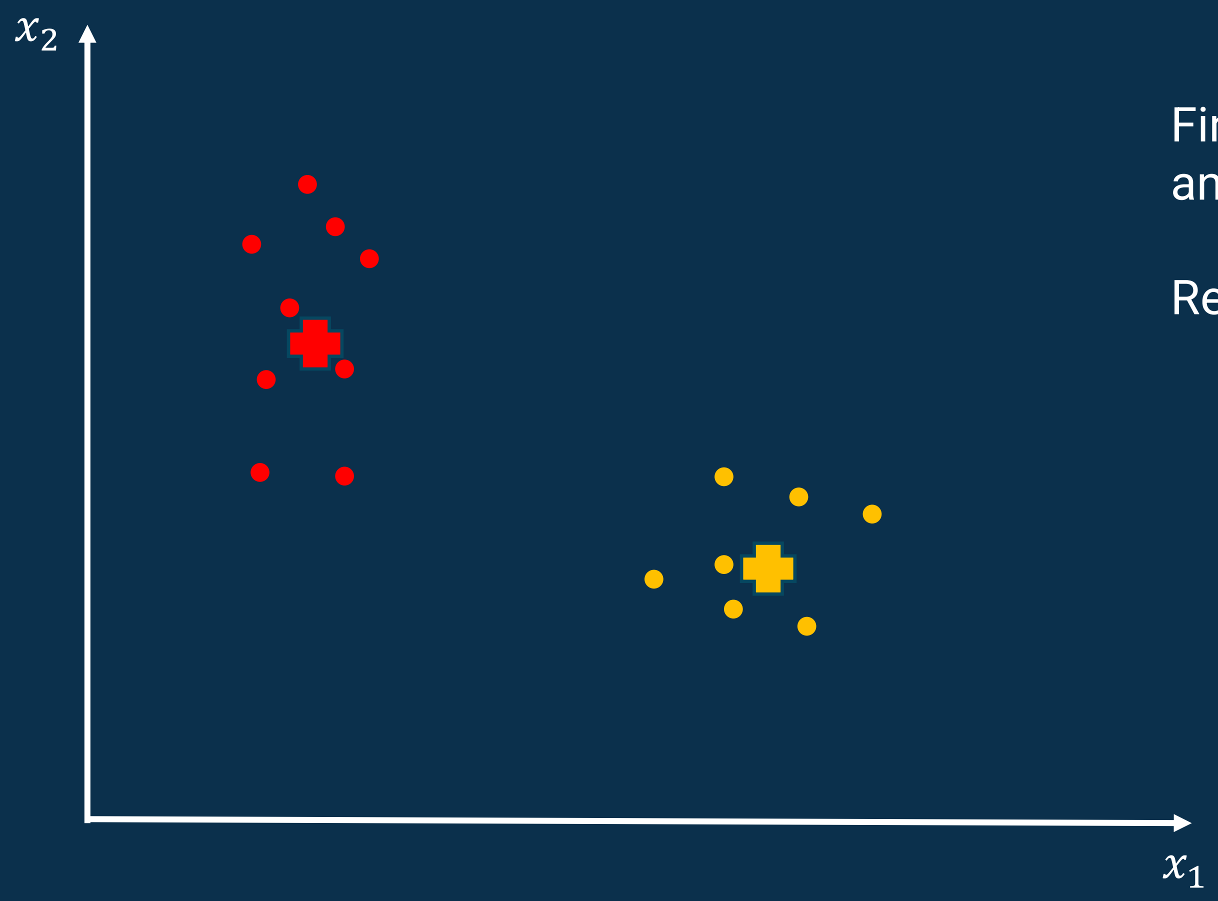


Now update the center of the centroids



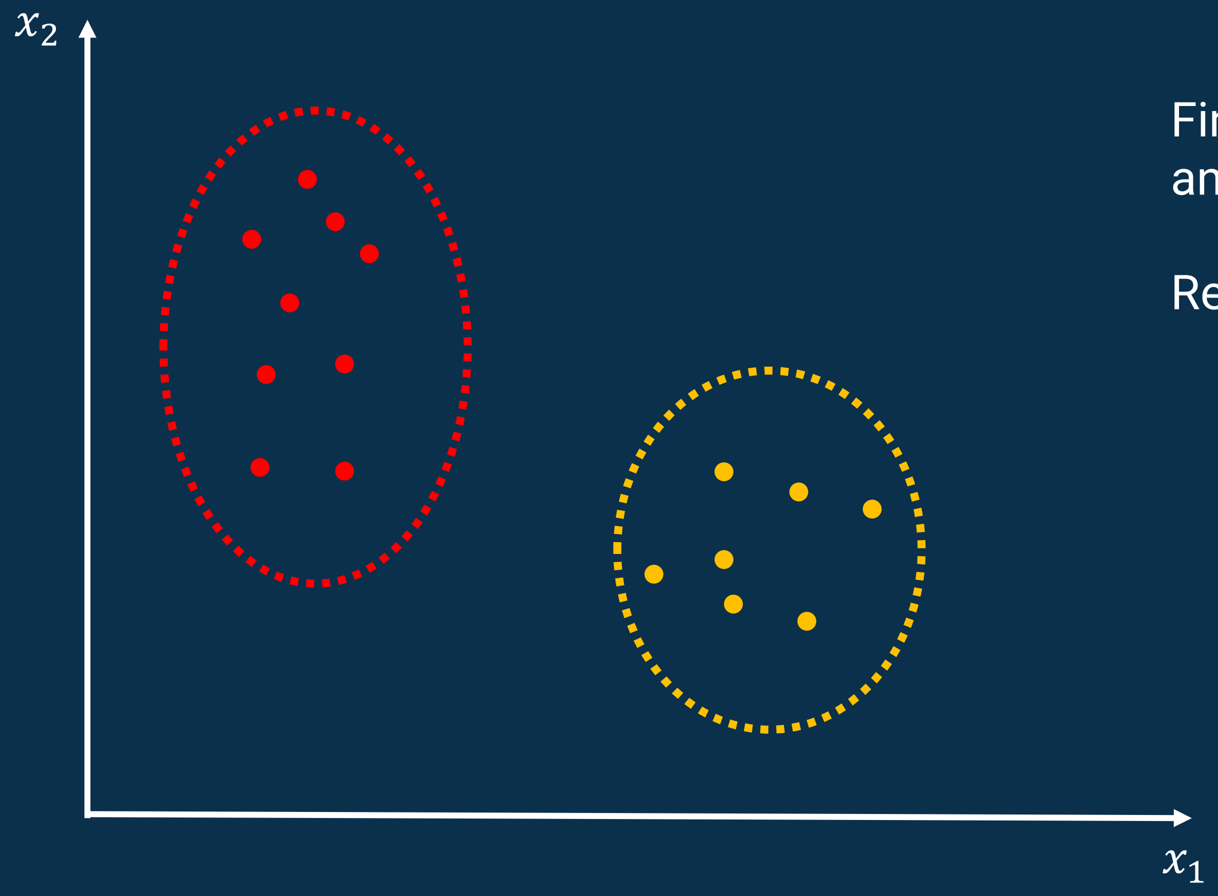
Find the the closest points
and and update the centroid

Repeat the process



Find the the closest points
and and update the centroid

Repeat the process



Find the the closest points
and and update the centroid

Repeat the process

K-Means Clustering

```
set.seed(123)

k_means_clustering <- kmeans(mtcars, centers = 2)

print(k_means_clustering$cluster)
```

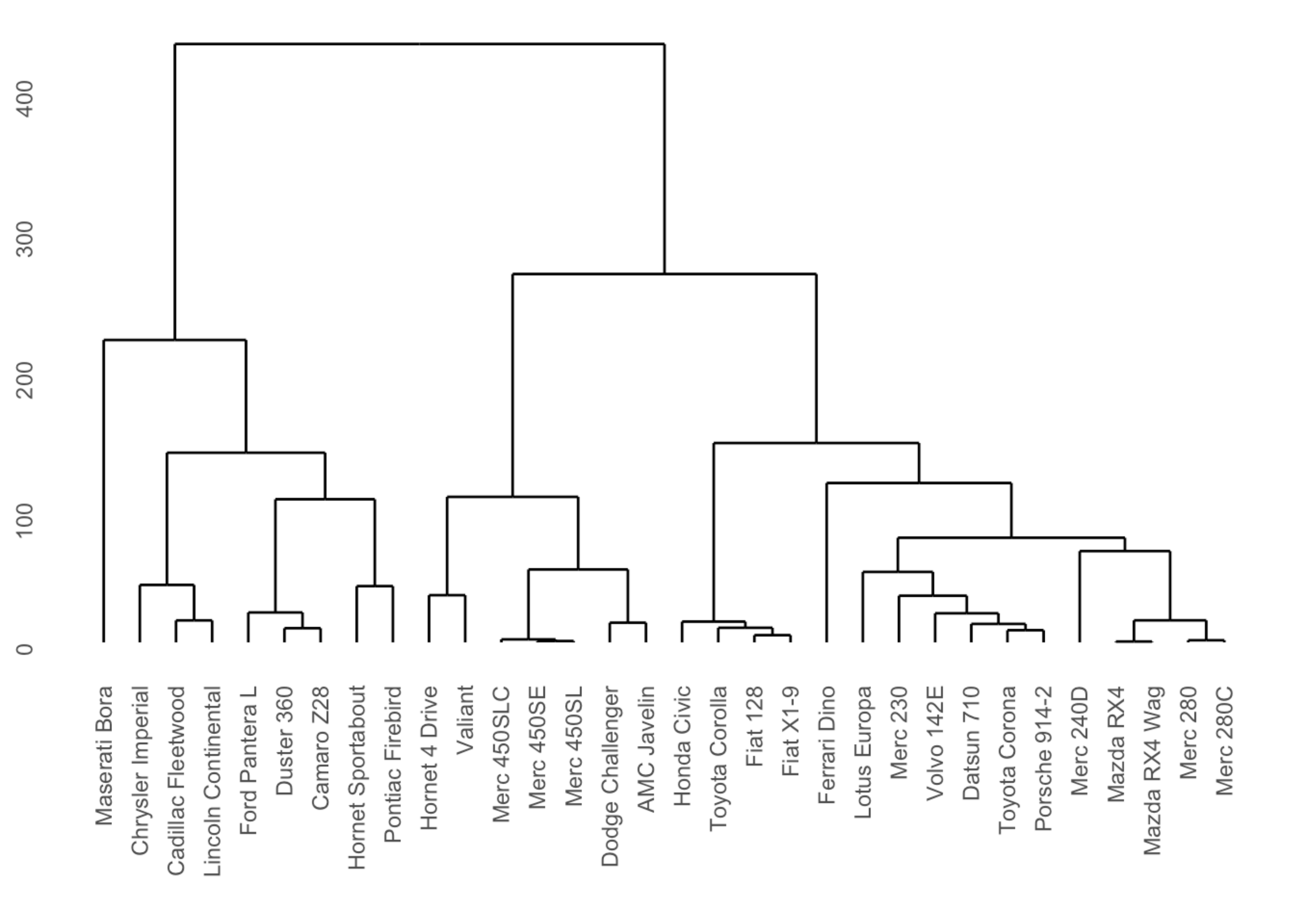
```
          Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive   Hornet Sportabout             Valiant          Duster 360
                  1                   1                   1                   1                   2                   1                   2
          Merc 240D            Merc 230            Merc 280           Merc 280C          Merc 450SE          Merc 450SL         Merc 450SLC
                  1                   1                   1                   1                   2                   2                   2
 Cadillac Fleetwood Lincoln Continental   Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla       Toyota Corona
                  2                   2                   2                   1                   1                   1                   1
   Dodge Challenger         AMC Javelin          Camaro Z28    Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa
                  2                   2                   2                   2                   1                   1                   1
     Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E
                  2                   1                   2                   1
```

Plot K-Means Clustering

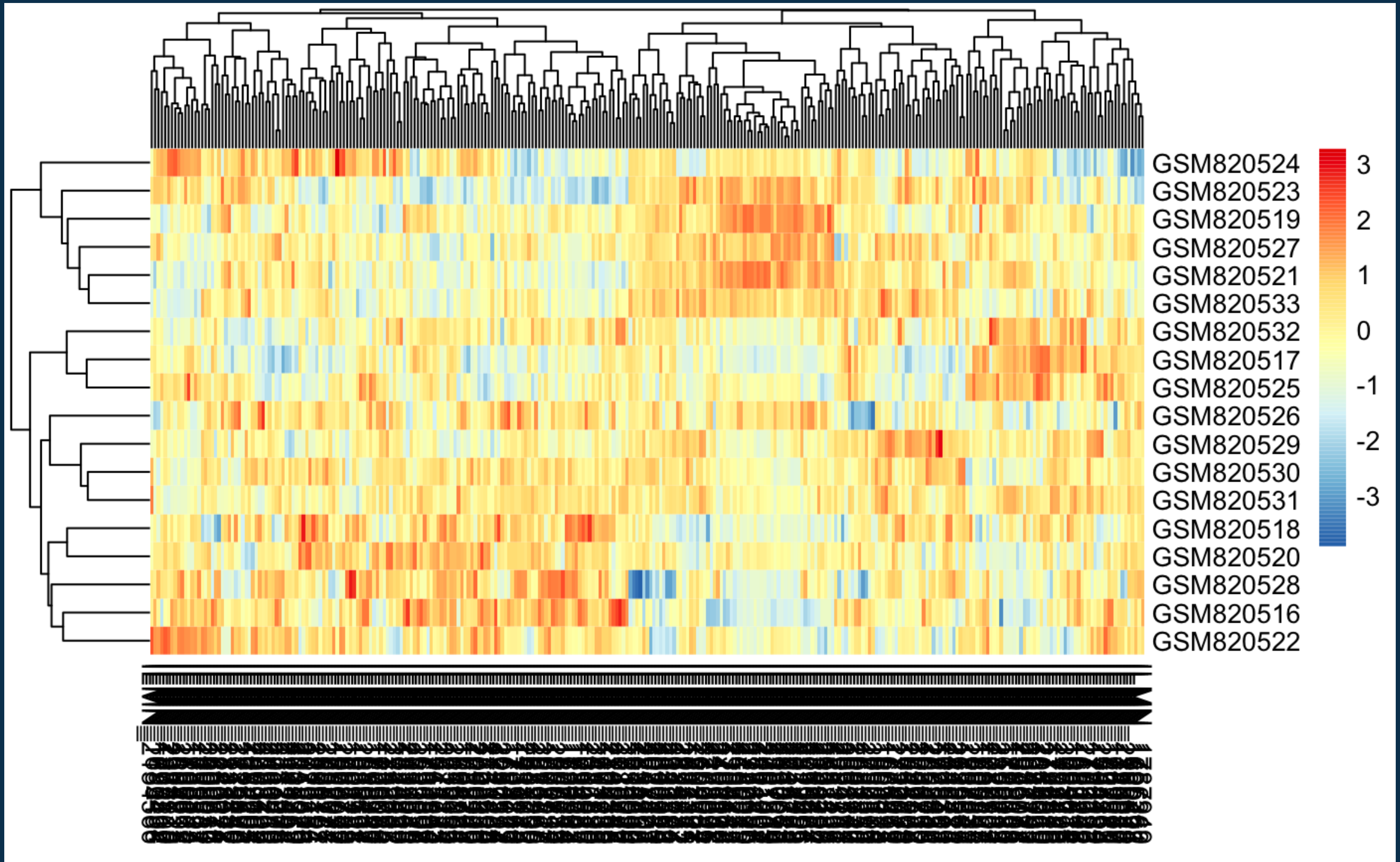
```
library(factoextra)

fviz_cluster(k_means_clustering,
data = mtcars)
```

Hierarchical Clustering

```
library(ggdendro)
set.seed(123)
mtcars_distance = dist(mtcars, method = 'euclidean')
h_clustering = hclust(mtcars_distance)
ggdendrogram(h_clustering)
```

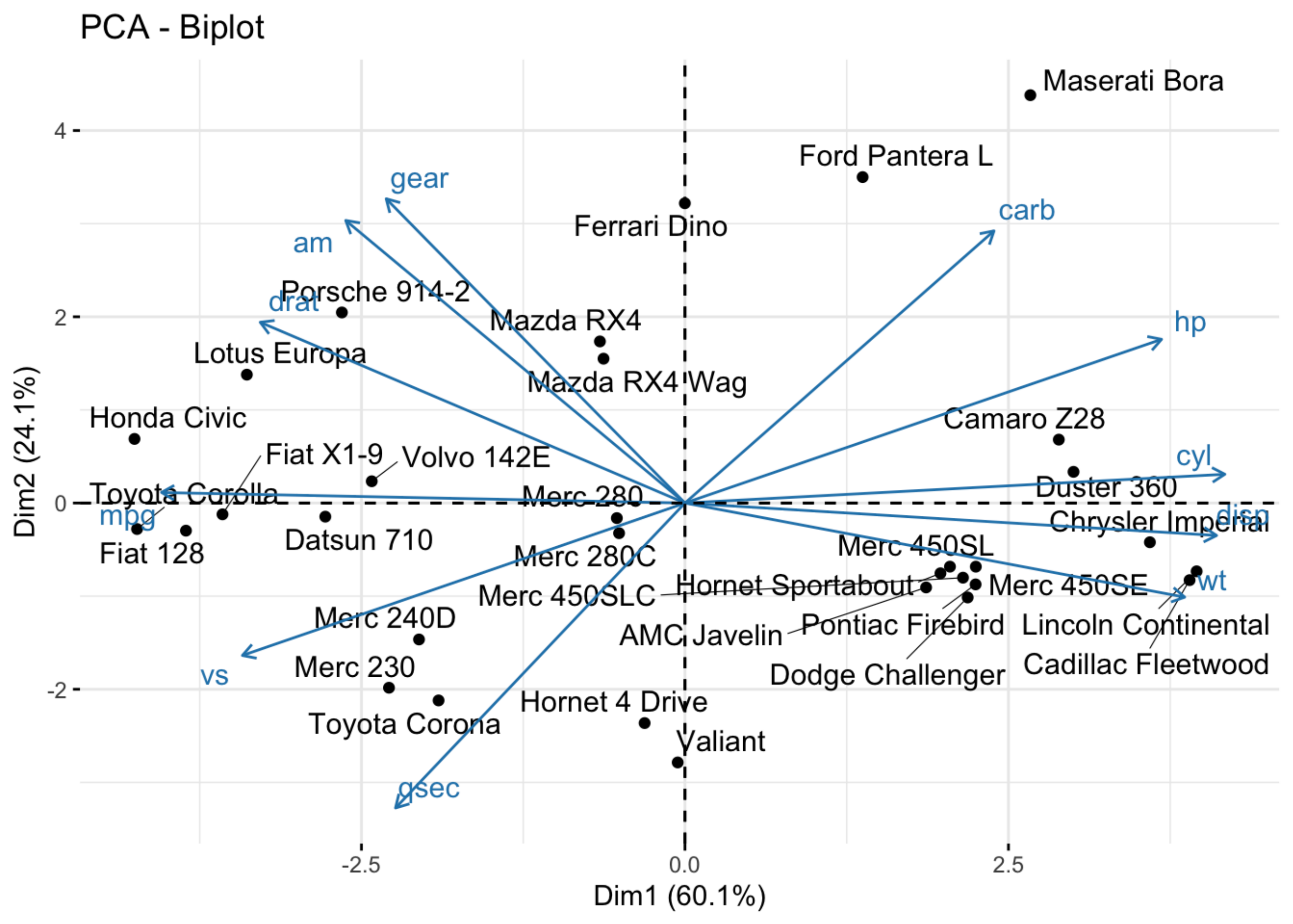


Hierarchical-Means Clustering based heatmaps are popular in analyzing gene expression data



Principal Component Analysis Biplot

```
library(FactoMineR)

pca_mtcars <- PCA(mtcars,  graph = FALSE)

fviz_pca_biplot(pca_mtcars, repel = TRUE)
```



PCA Biplot can be used to understand the correlation structure of the variables

- Statistical Learning
  - Supervised Learning
    - Regression
    - Classification
  - Unsupervised Learning

Thank you!