INTRODUCTION TO R

Iris Classification Example Notebook

Danny Lumian, Ph.D.

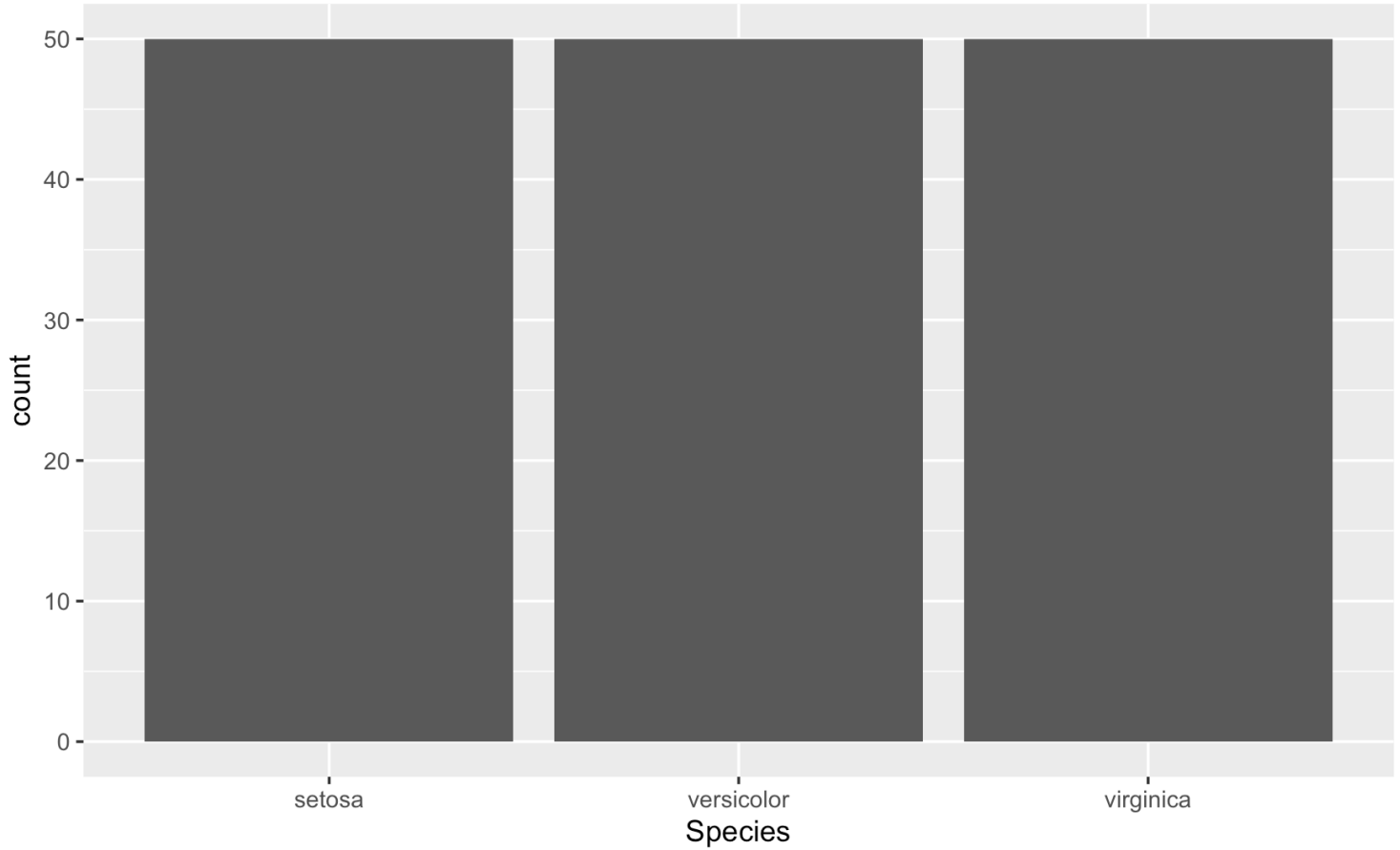
Data Science Training Specialist

NIH Office of Data Science Strategy

Iris Classification

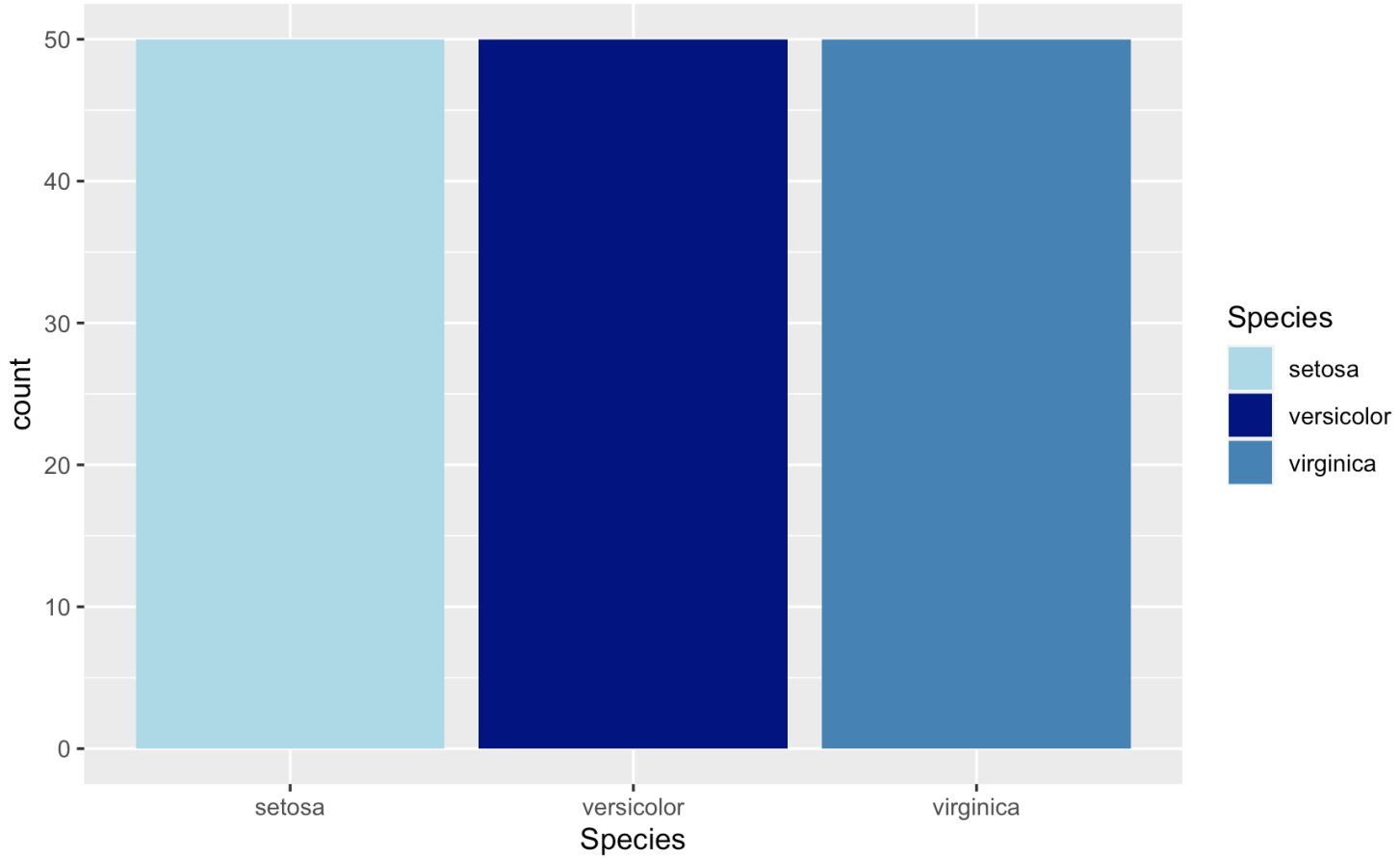
- This notebook uses the built-in Iris dataset to outline a basic approach to machine learning with a decision tree classifier
- There are 4 features and 1 target variable consisting of 3 groups
- This notebook uses several popular machine learning libraries in R

Plot the count of each target species:
ggplot(data = iris, aes(x=Species)) + geom_bar()



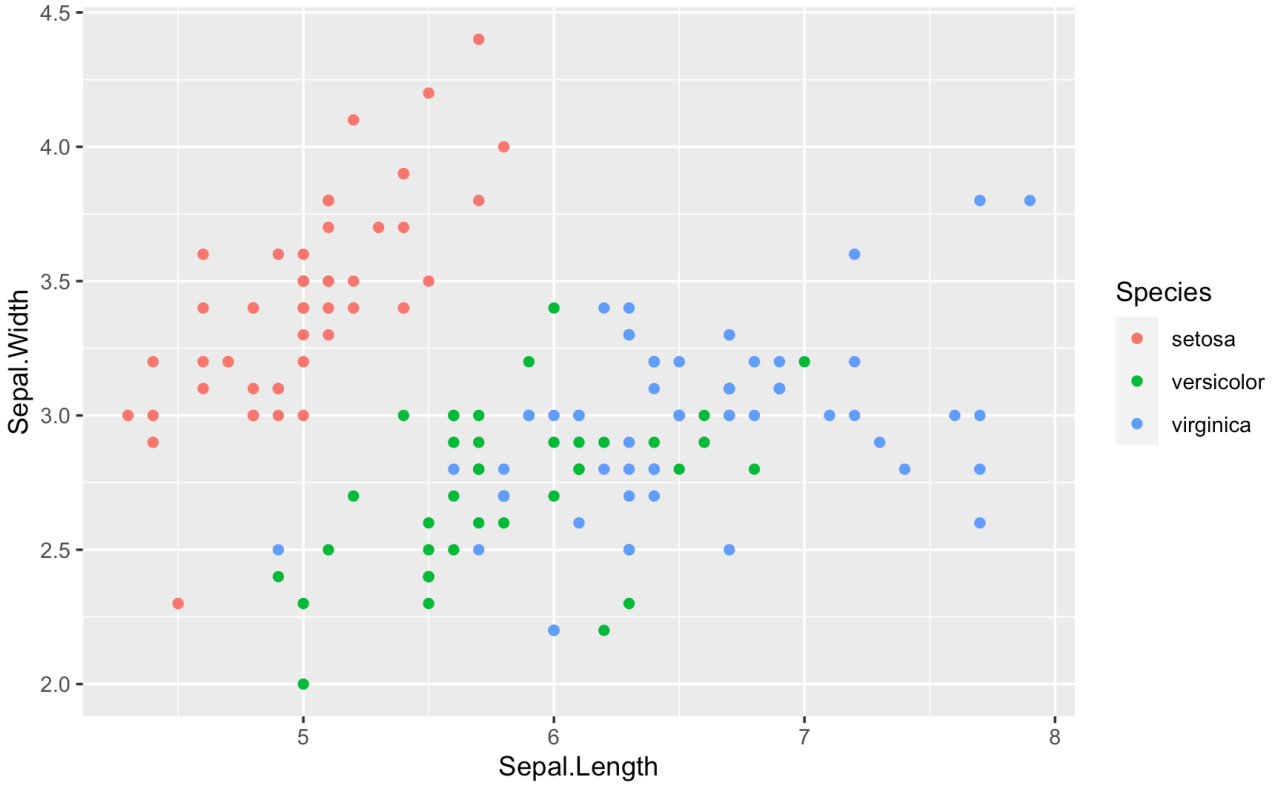
Plot the count of each target species with color:

```
ggplot(data = iris, aes(x=Species, fill=Species)) + geom_bar() +
  scale_fill_manual(values = c('lightblue', 'navy', 'steelblue'))
```



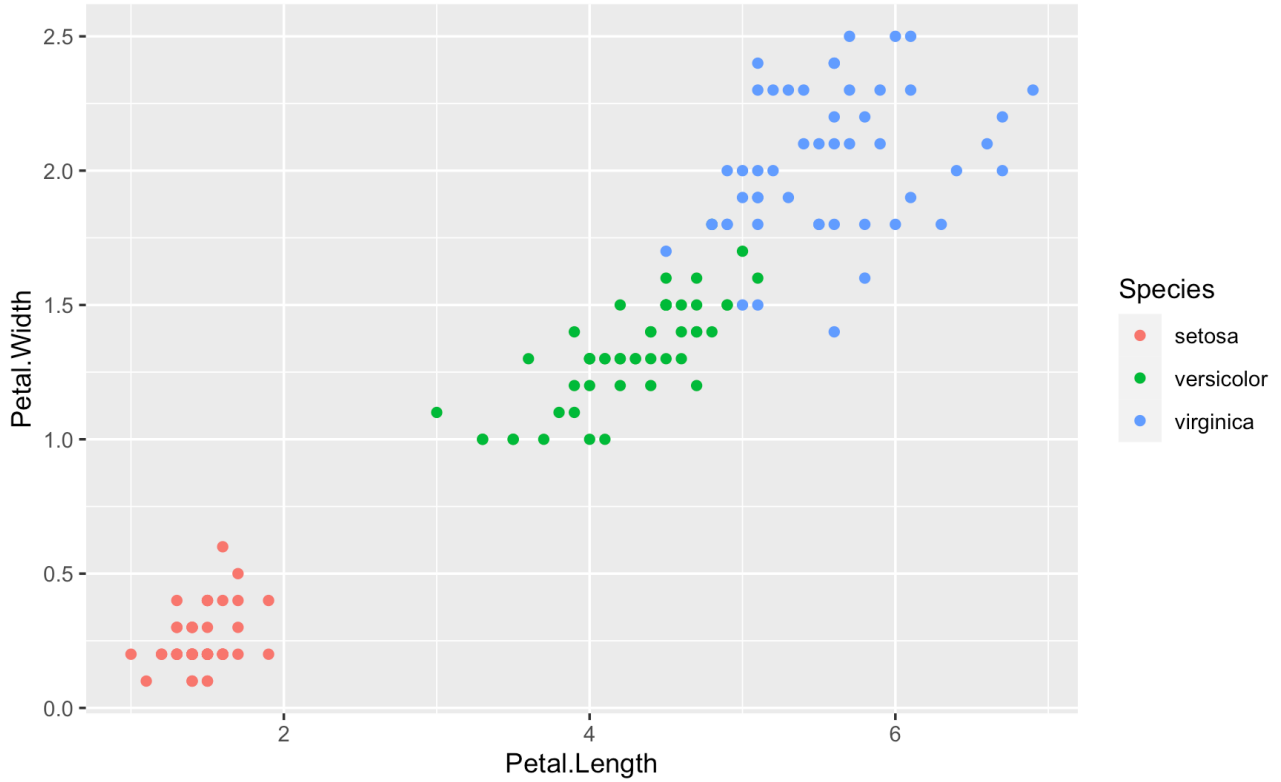
Scatter plot of Features

```
point1 = ggplot(data=iris,
aes(x=Sepal.Length,
y=Sepal.Width,
color=Species))

point1 + geom_point()
```



Scatter plot of Features

```
point1 = ggplot(data=iris,
aes(x=Petal.Length,
y=Petal.Width,
color=Species))

point1 + geom_point()
```



Sepal.Length <dbl>	Sepal.Width <dbl>	Petal.Length <dbl>	Petal.Width <dbl>	Species <fctr>
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

1–10 of 150 rows Previous 1 2 3 4 5 6 ...

Species <fctr>	variable <fctr>	value <dbl>
setosa	Sepal.Length	5.1
setosa	Sepal.Length	4.9
setosa	Sepal.Length	4.7
setosa	Sepal.Length	4.6
setosa	Sepal.Length	5.0
setosa	Sepal.Length	5.4
setosa	Sepal.Length	4.6
setosa	Sepal.Length	5.0
setosa	Sepal.Length	4.4
setosa	Sepal.Length	4.9

1–10 of 600 rows Previous

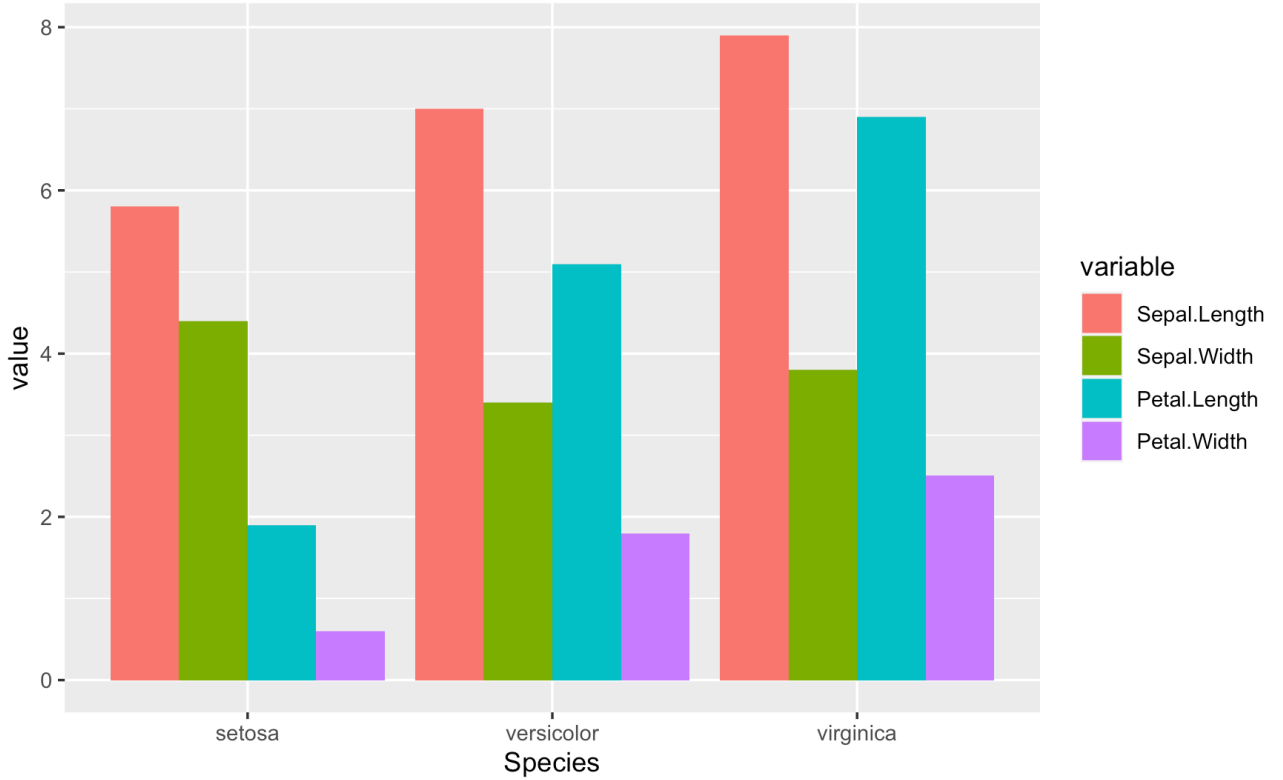
Wide Form vs Long Form

Wide form has a column for each feature. Long form has a column indicating feature and a column indicating the value of that feature.
Some plots in R require a long form of the data.

```
iris_melted <- melt(iris, id.vars="Species")
iris_melted
```

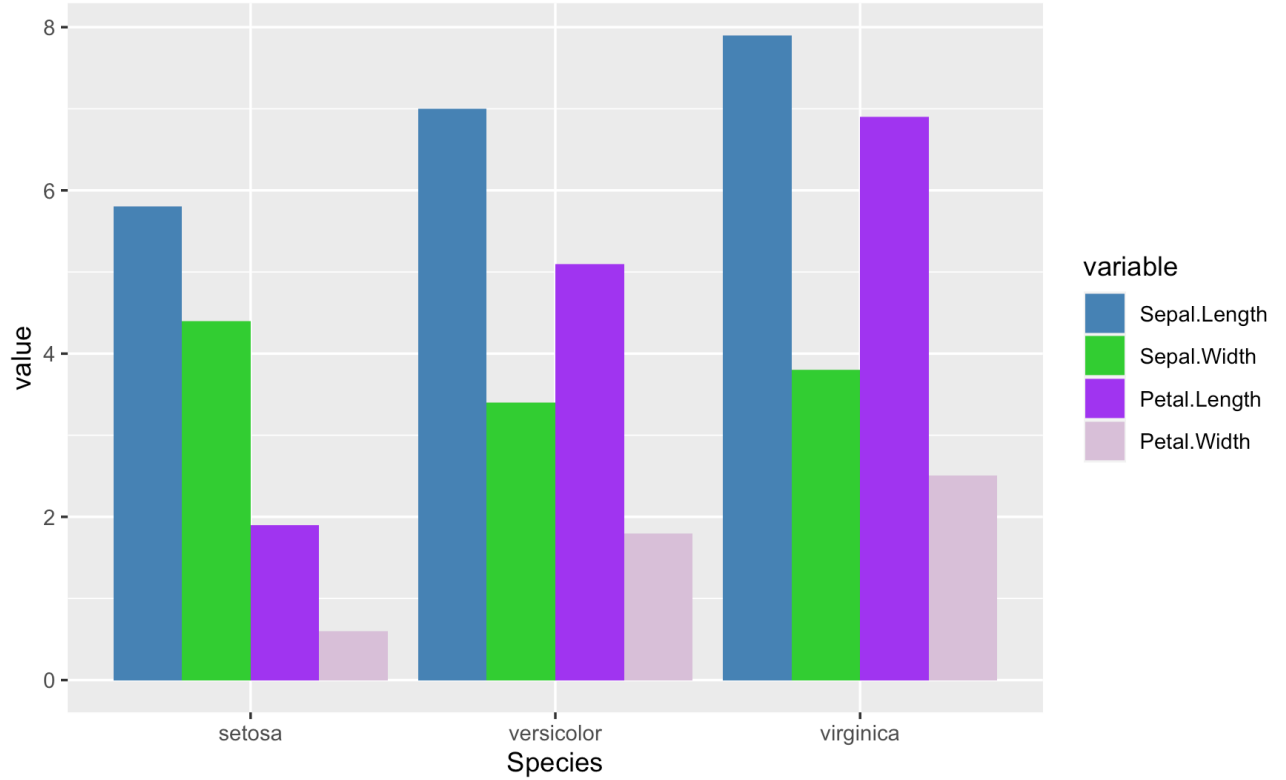
Bar Plot of all Features

```
bar1 =
ggplot(data=iris_melted,
aes(x=Species, y=value,
fill=variable))

bar1 +
geom_bar(stat="identity"
, position="dodge")
```



Bar Plot of all Features with Custom Colors

```
bar1 = ggplot(data=iris_melted, aes(x=Species, y=value, fill=variable))

bar1 + geom_bar(stat="identity", position="dodge") +

  scale_fill_manual(values = c("steelblue", "limegreen", "purple", "thistle"))
```



Train-Test Split

```
set.seed(1)
train.index = createDataPartition(iris$Species, p=.7, list=FALSE)
train <- iris[train.index, ]
test <- iris[-train.index, ]
```

- For machine learning models, it is important to do a train-test split
- The training data is used to train the model
- The test data is used to validate the model on unseen data (that is, data not used in trainig)
- It can be important to stratify the data so that each group of the target is equally represented

Training the model

```
model <- rpart(Species ~ ., data = train, method = "class")
model
```

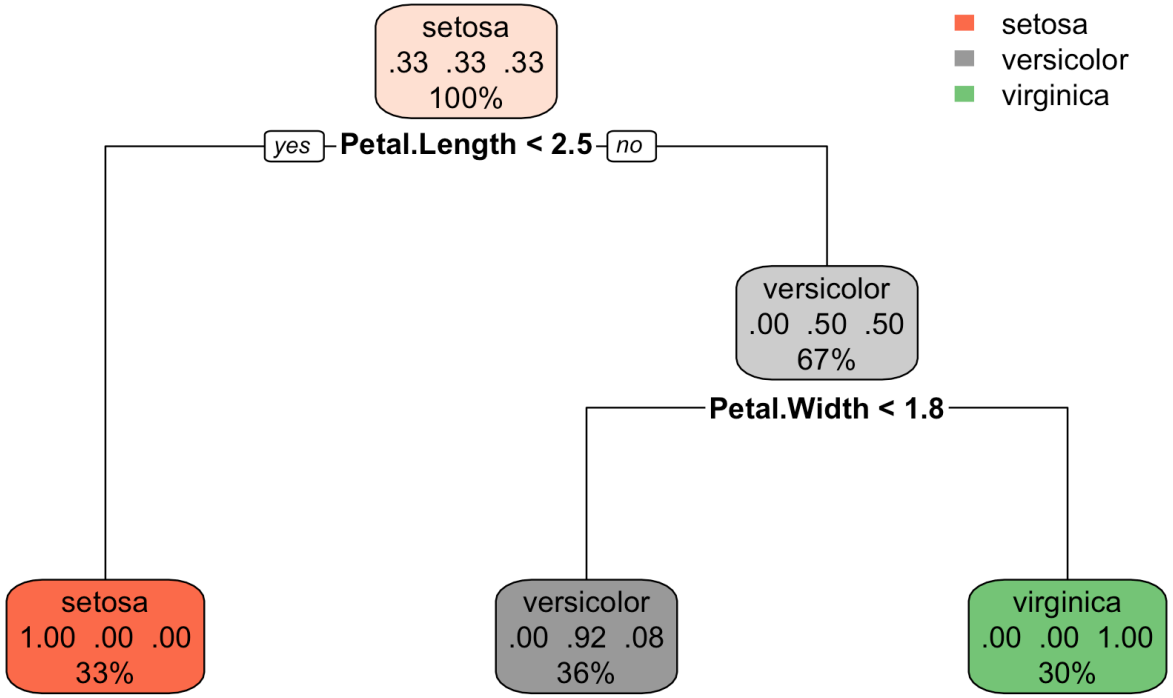
- The model is trained on the training portion of the data
- Output as shown below is generated
- Next we will plot this model

```
n= 105

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 105 70 setosa (0.33333333 0.33333333 0.33333333)
  2) Petal.Length< 2.45 35  0 setosa (1.00000000 0.00000000 0.00000000) *
  3) Petal.Length>=2.45 70 35 versicolor (0.00000000 0.50000000 0.50000000)
    6) Petal.Width< 1.75 38  3 versicolor (0.00000000 0.92105263 0.07894737) *
    7) Petal.Width>=1.75 32  0 virginica (0.00000000 0.00000000 1.00000000) *
```

Examining the model

rpart.plot(model)

- Using the rpart package we can plot the model
- This visually shows how decisions are made



Predicting with the model

Note, we predict on the **test** dataset

```
preds <- predict(model, newdata = test, type = "class")
preds
```

Confusion Matrix to Evaluate Results

confusionMatrix(test$Species, preds)

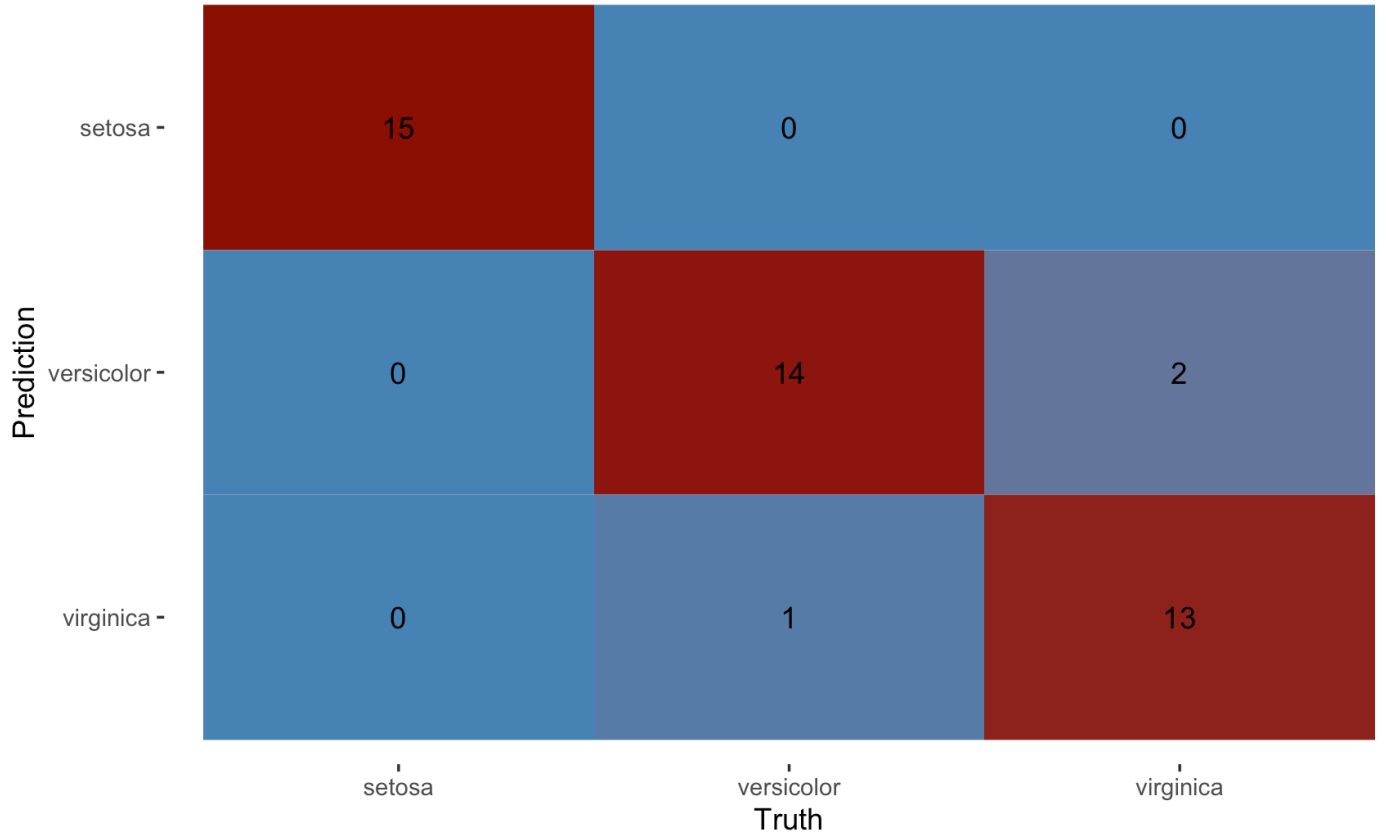
Note that results include accuracy of the model as well as other metrics for each class in the model.

Next we will plot the confusion matrix shown at the top of this results output.

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         14         1
  virginica       0          2        13

Overall Statistics

               Accuracy : 0.9333
                 95% CI : (0.8173, 0.986)
    No Information Rate : 0.3556
    P-Value [Acc > NIR] : 5.426e-16

                  Kappa : 0.9

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.8750           0.9286
Specificity                 1.0000            0.9655           0.9355
Pos Pred Value              1.0000            0.9333           0.8667
Neg Pred Value              1.0000            0.9333           0.9667
Prevalence                  0.3333            0.3556           0.3111
Detection Rate              0.3333            0.3111           0.2889
Detection Prevalence        0.3333            0.3333           0.3333
Balanced Accuracy           1.0000            0.9203           0.9320
```

Plotting Confusion Matrix

```
cm <- conf_mat(test, Species, preds)

autoplot(cm, type = "heatmap") +
  scale_fill_gradient(low = "steelblue", high = "darkred")
```



Model Overview and Results

- We took 150 samples of 3 iris species and created a decision tree model to help classify them
- Our model had good accuracy, 93%, which might have been expected given our plots of the features
- From our results, we can Setosa was the most easily classified with no mistakes
- There was some confusion when classifying Versicolor vs Virginica
- This notebook shows the process for conducting a machine learning project in R