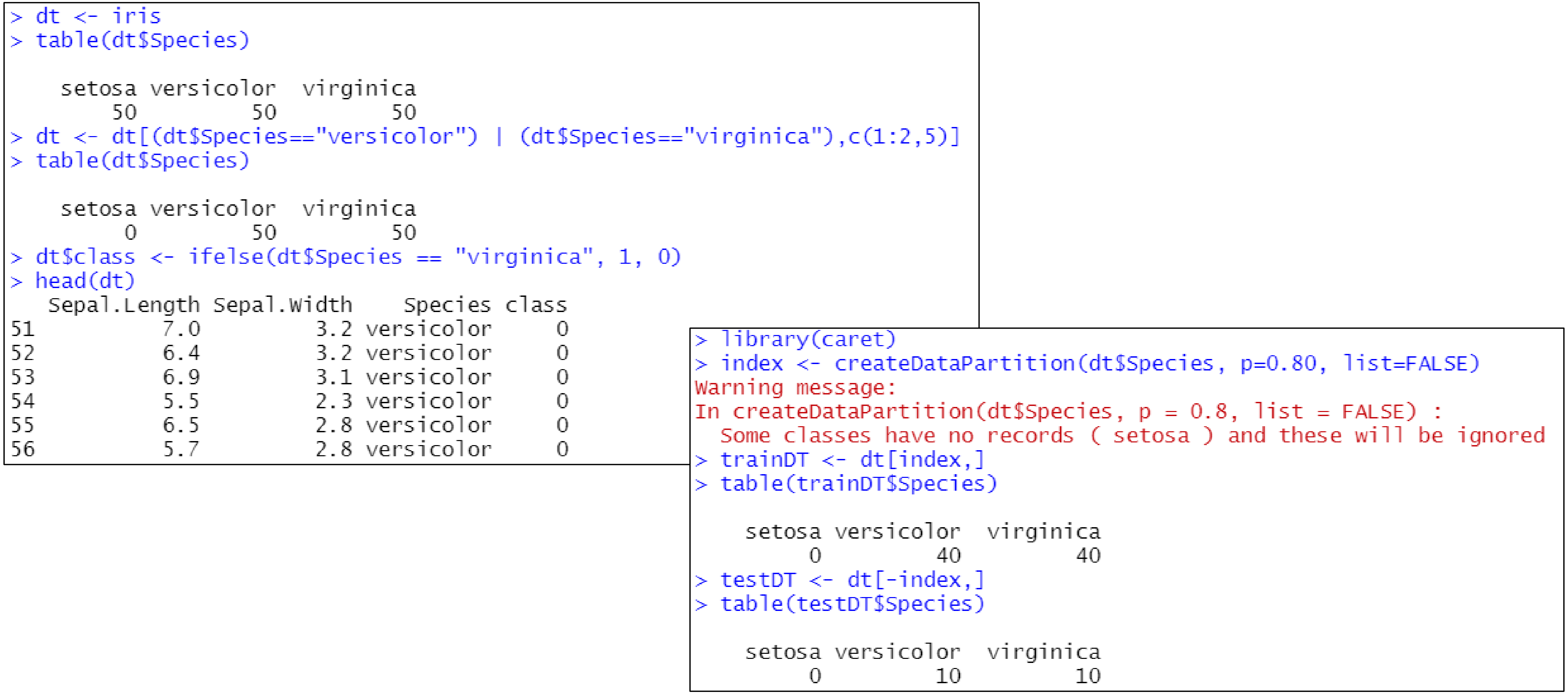
ROC curve, PR curve, and AUCs

May 28, 2022

Thien

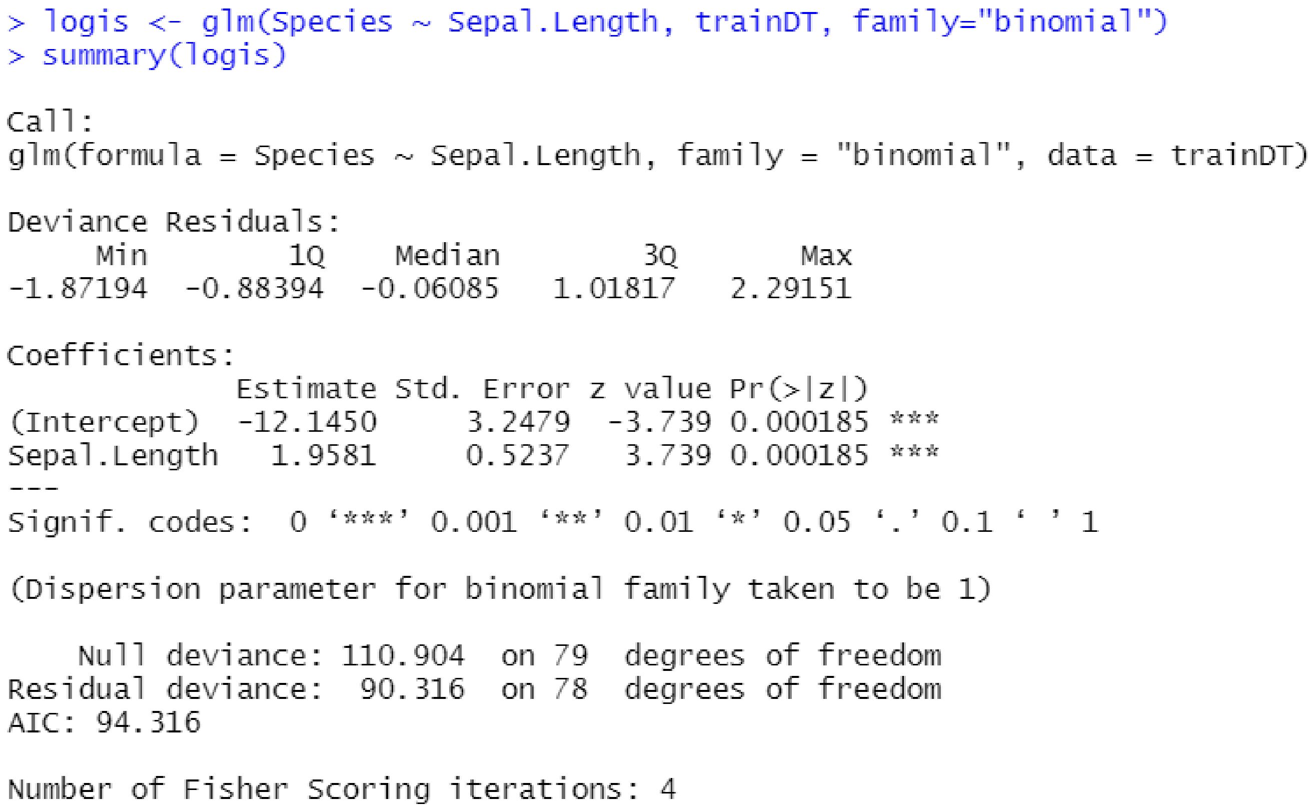
Data



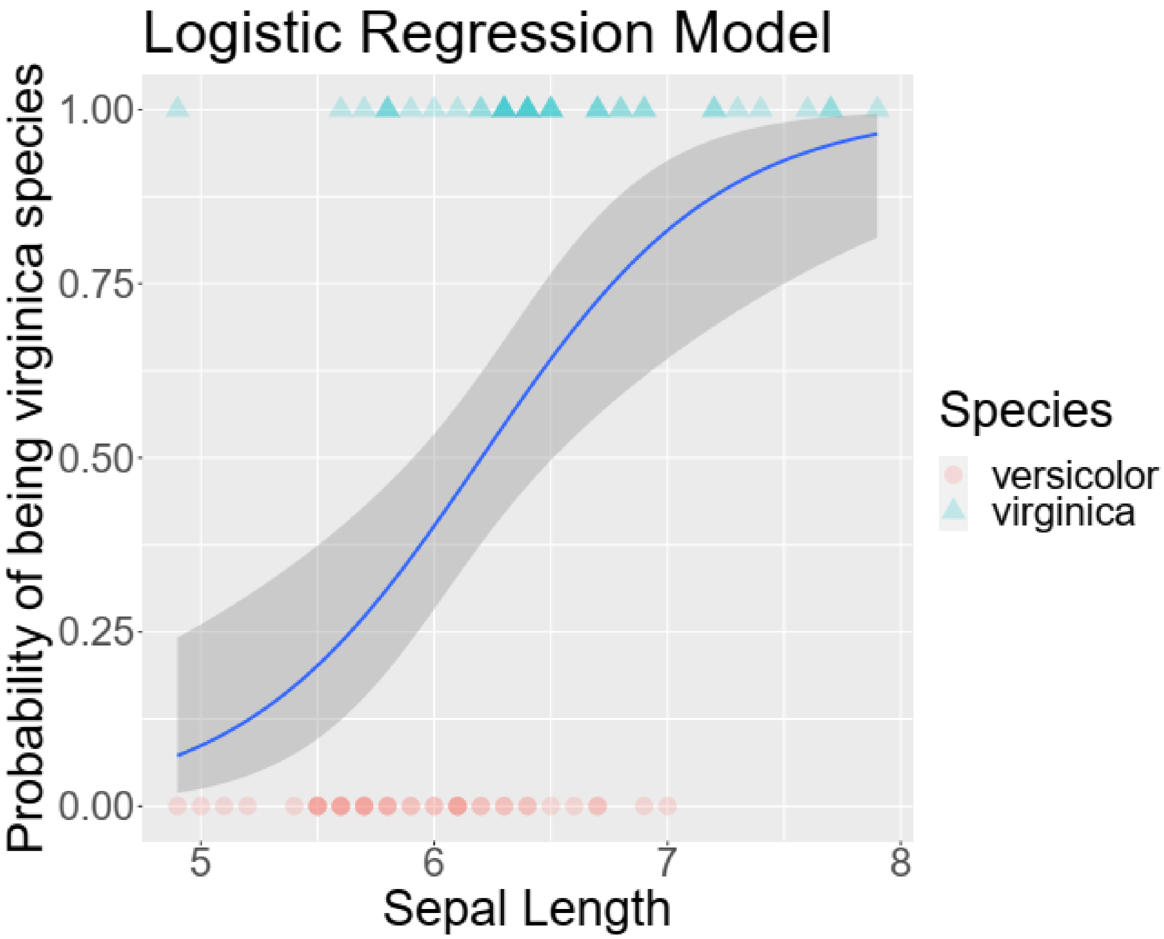
```
> dt <- iris
> table(dt$Species)

    setosa versicolor  virginica 
        50         50         50 
> dt <- dt[(dt$Species=="versicolor") | (dt$Species=="virginica"),c(1:2,5)]
> table(dt$Species)

    setosa versicolor  virginica 
         0         50         50 
> dt$class <- ifelse(dt$Species == "virginica", 1, 0)
> head(dt)
   Sepal.Length Sepal.Width    Species class
51          7.0         3.2 versicolor     0
52          6.4         3.2 versicolor     0
53          6.9         3.1 versicolor     0
54          5.5         2.3 versicolor     0
55          6.5         2.8 versicolor     0
56          5.7         2.8 versicolor     0
```

```
> library(caret)
> index <- createDataPartition(dt$Species, p=0.80, list=FALSE)
Warning message:
In createDataPartition(dt$Species, p = 0.8, list = FALSE) :
  Some classes have no records ( setosa ) and these will be ignored
> trainDT <- dt[index,]
> table(trainDT$Species)

    setosa versicolor  virginica 
         0         40         40 
> testDT <- dt[-index,]
> table(testDT$Species)

    setosa versicolor  virginica 
         0         10         10 
```

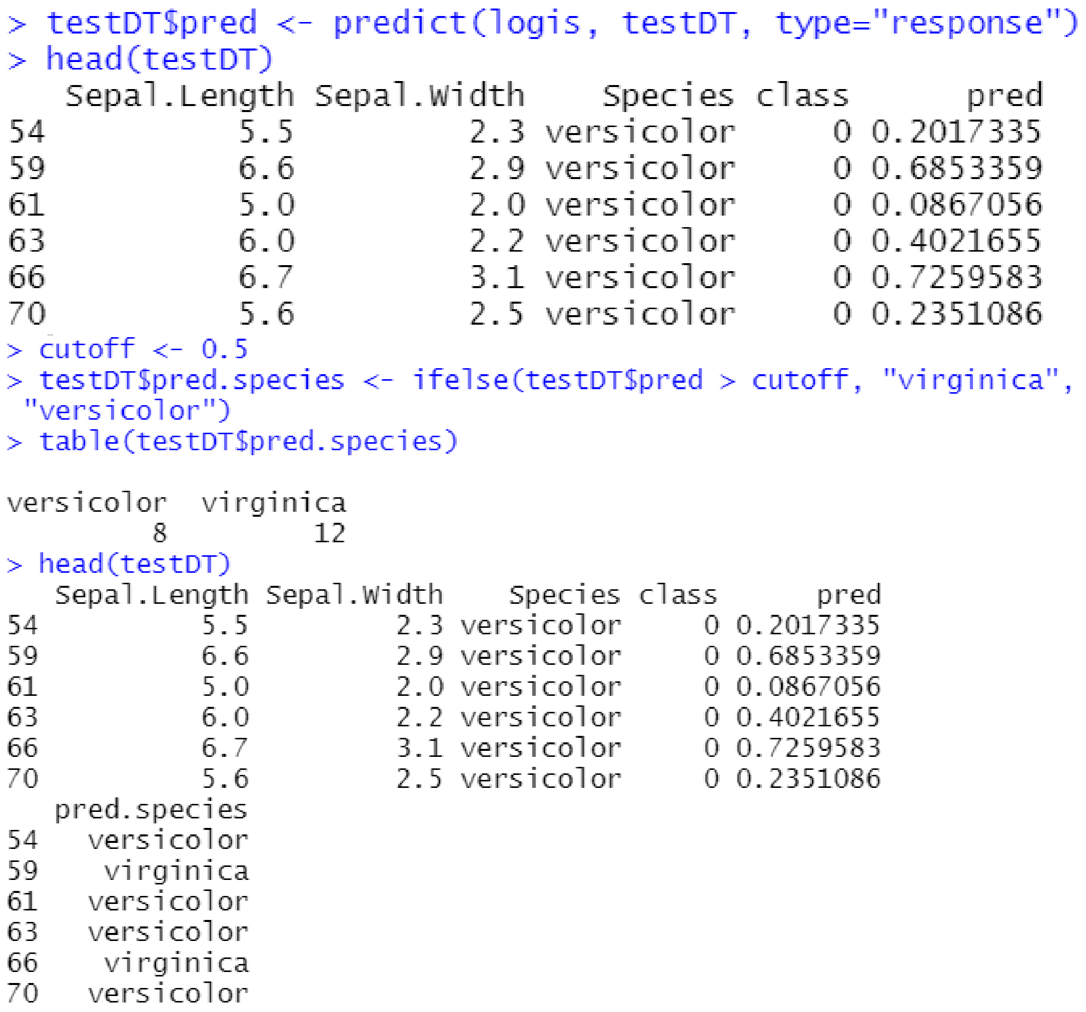
Bulid up logistic regression model



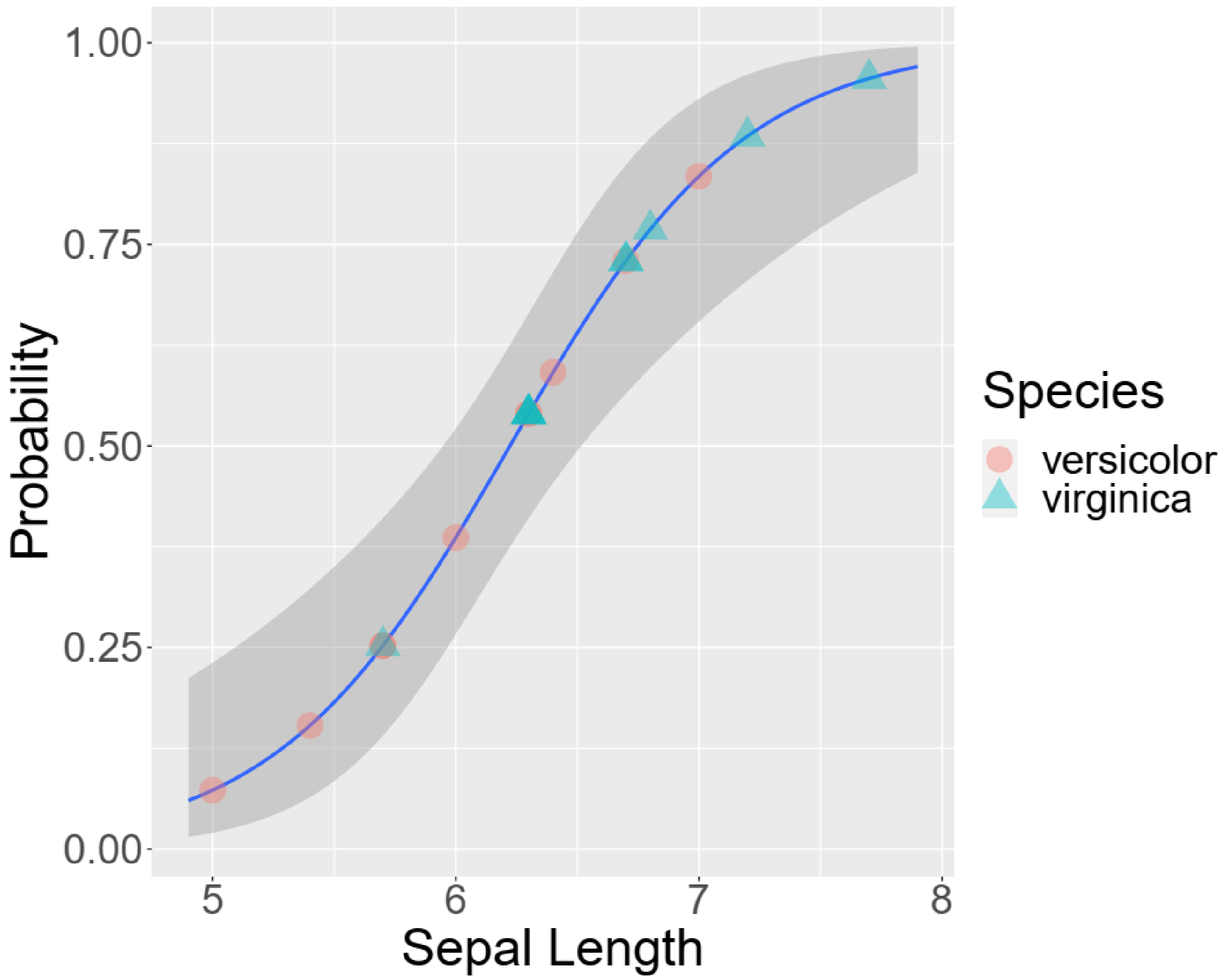
```
> logis <- glm(Species ~ Sepal.Length, trainDT, family="binomial")
> summary(logis)

Call:
glm(formula = Species ~ Sepal.Length, family = "binomial", data = trainDT)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.87194  -0.88394  -0.06085   1.01817   2.29151  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  -12.1450     3.2479  -3.739 0.000185 ***
Sepal.Length   1.9581     0.5237   3.739 0.000185 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 110.904  on 79  degrees of freedom
Residual deviance:  90.316  on 78  degrees of freedom
AIC: 94.316

Number of Fisher Scoring iterations: 4
```



Prediction

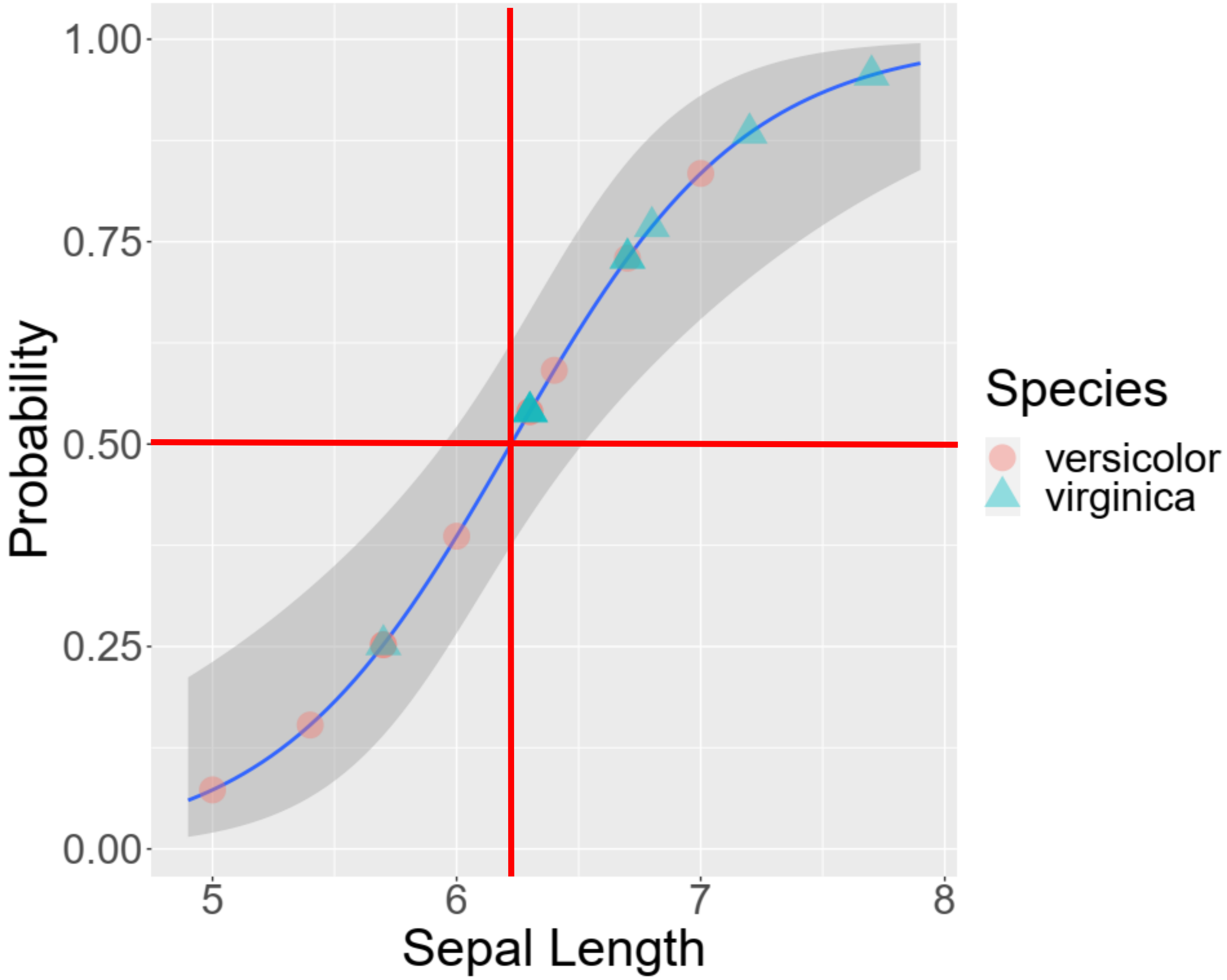


```
> testDT$pred <- predict(logis, testDT, type="response")
> head(testDT)
   Sepal.Length Sepal.Width    Species class      pred
54          5.5         2.3 versicolor     0 0.2017335
59          6.6         2.9 versicolor     0 0.6853359
61          5.0         2.0 versicolor     0 0.0867056
63          6.0         2.2 versicolor     0 0.4021655
66          6.7         3.1 versicolor     0 0.7259583
70          5.6         2.5 versicolor     0 0.2351086
> cutoff <- 0.5
> testDT$pred.species <- ifelse(testDT$pred > cutoff, "virginica",
 "versicolor")
> table(testDT$pred.species)

versicolor  virginica 
         8         12 
> head(testDT)
   Sepal.Length Sepal.Width    Species class      pred
54          5.5         2.3 versicolor     0 0.2017335
59          6.6         2.9 versicolor     0 0.6853359
61          5.0         2.0 versicolor     0 0.0867056
63          6.0         2.2 versicolor     0 0.4021655
66          6.7         3.1 versicolor     0 0.7259583
70          5.6         2.5 versicolor     0 0.2351086
   pred.species
54   versicolor
59    virginica
61   versicolor
63   versicolor
66    virginica
70   versicolor
```

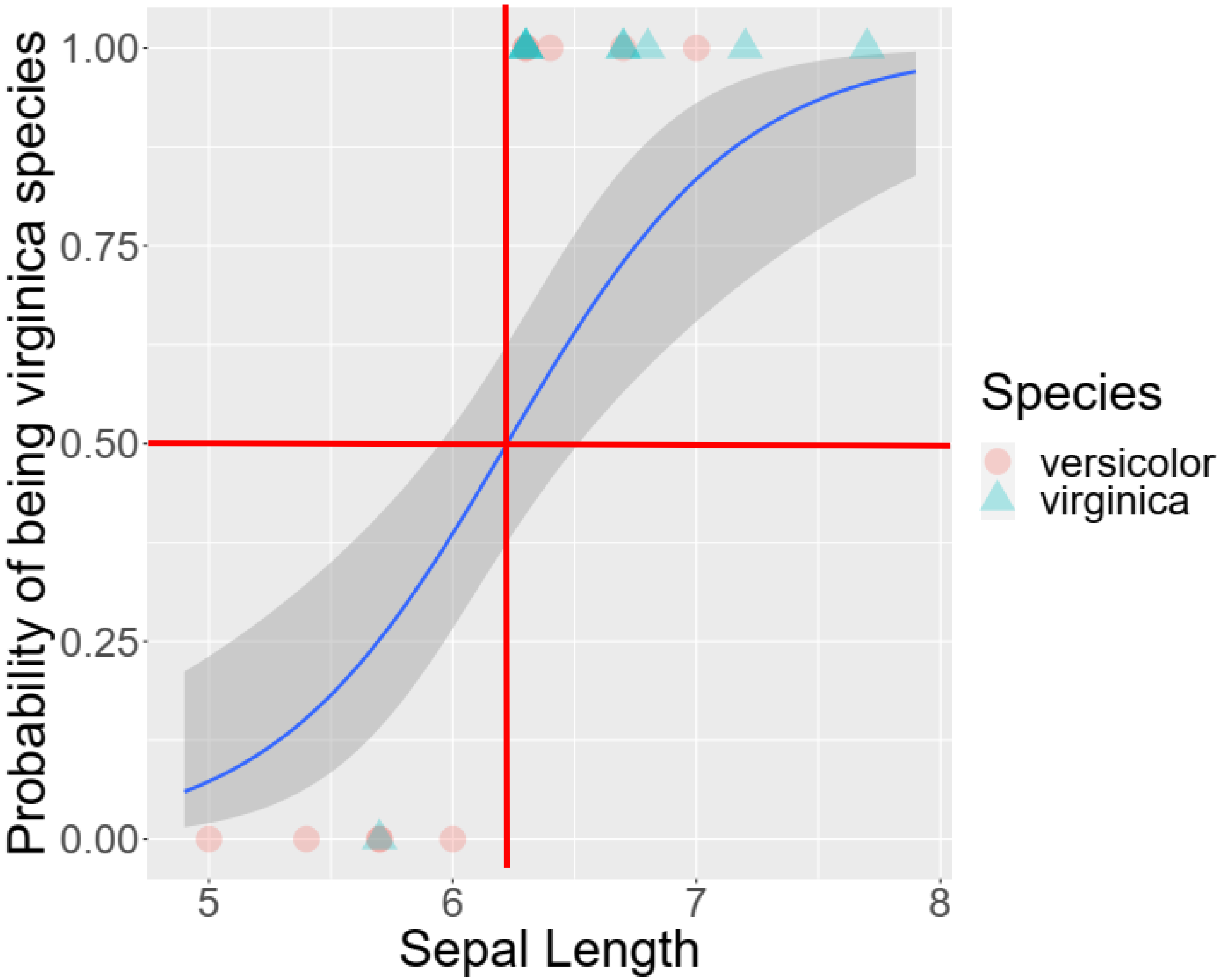


Threshold = 0.5 for classification

Original testset



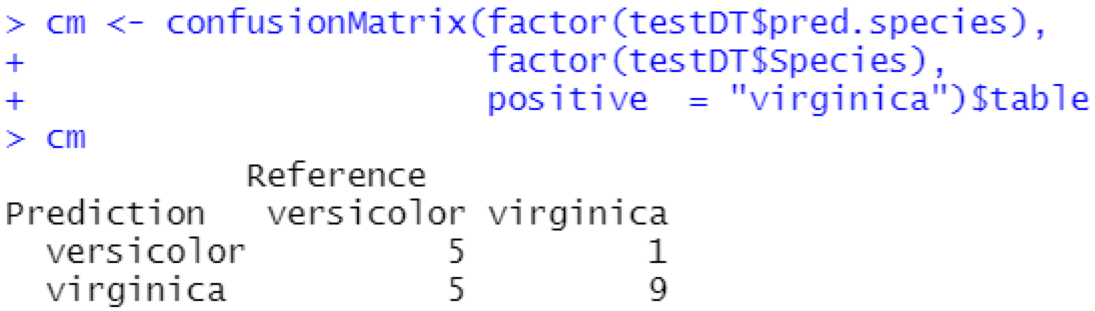
Predicted testset



Confusion Matrix

	Virginica	Versicolor
Virginica (Predicted)	9	5
Versicolor (Predicted)	1	5

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negative (FNs)	True Negative (TNs)



```
> cm <- confusionMatrix(factor(testDT$pred.species),
+                       factor(testDT$Species),
+                       positive  = "virginica")$table
> cm
            Reference
Prediction   versicolor virginica
  versicolor          5         1
  virginica           5         9
```

Sensitivity and Specificity

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negative (FNs)	True Negative (TNs)

- **Sensitivity** (true positive rate): probability of a positive test, conditioned on truly being positive. $\mathrm{TPR} = \frac{\mathrm{TPs}}{\mathrm{TPs+FNs}}$

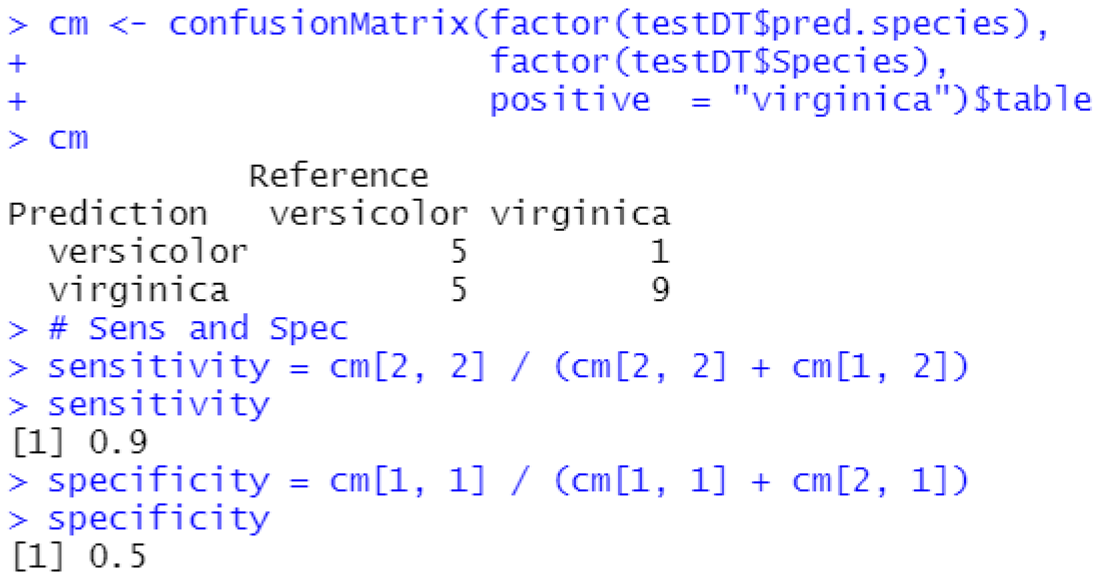
- **Specificity** (true negative rate): probability of a negative test, conditioned on truly being negative. $\mathrm{TNR} = \frac{\mathrm{TNs}}{\mathrm{FPs+TNs}}$

Sensitivity and Specificity

	Virginica	**Versicolor**
Virginica (Predicted)	**9**	**5**
Versicolor (Predicted)	**1**	**5**

- $\text{Sens} = \frac{\text{TPs}}{\text{TPs+FNs}} = \frac{9}{5+5} = 0.9$

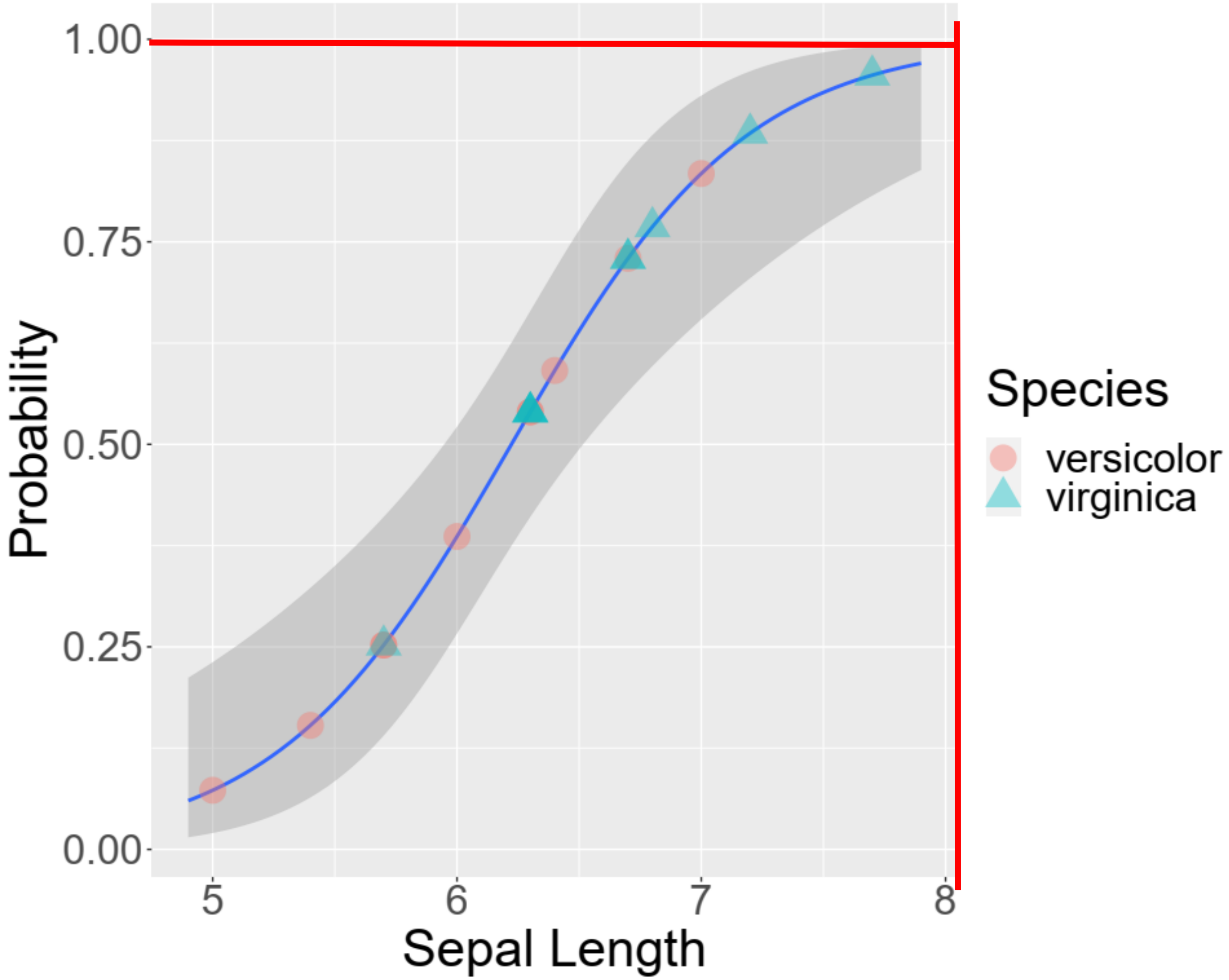
- $\text{Spec} = \frac{\text{TNs}}{\text{FPs+TNs}} = \frac{5}{1+9} = 0.5$



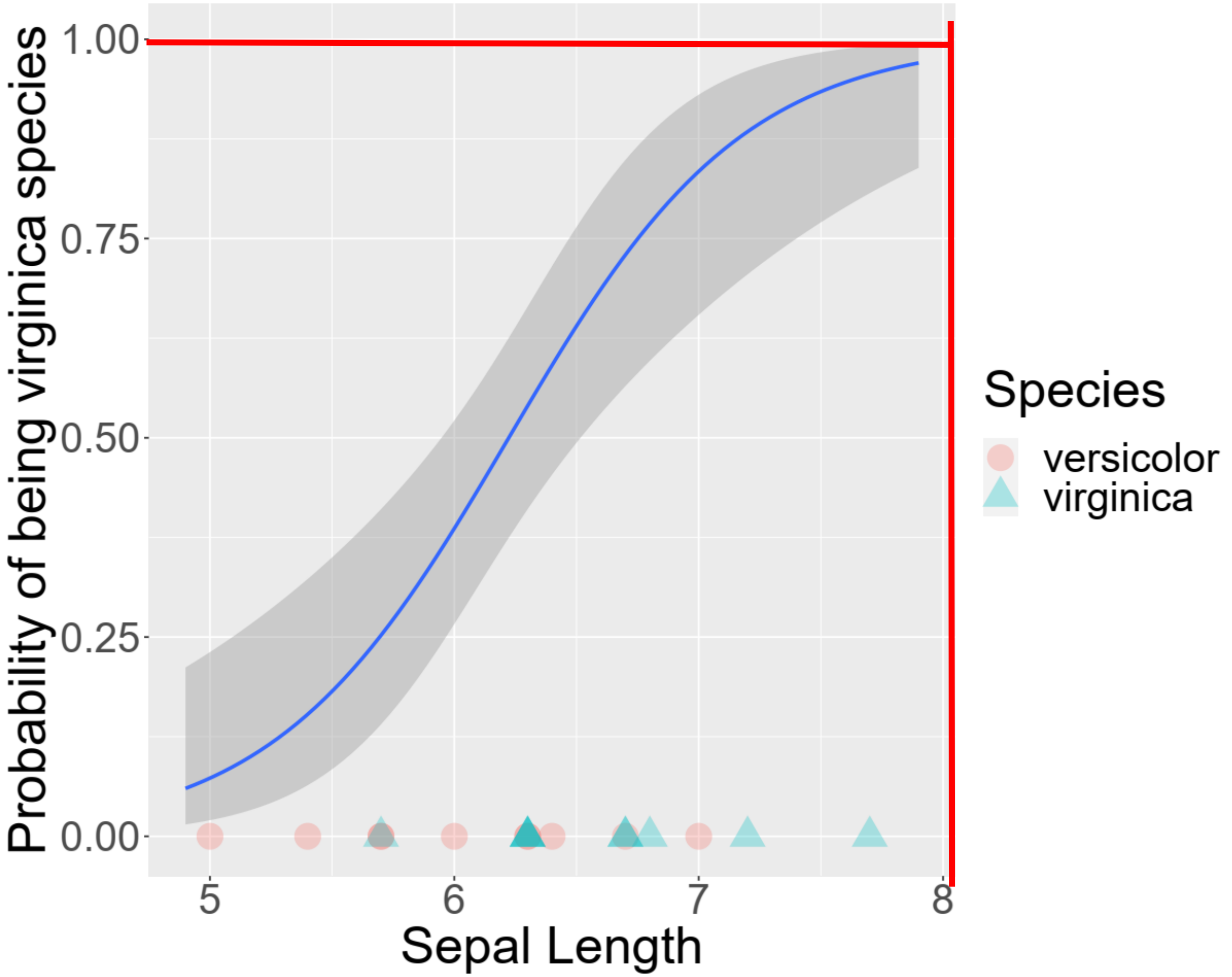
```
> cm <- confusionMatrix(factor(testDT$pred.species),
+                       factor(testDT$Species),
+                       positive  = "virginica")$table
> cm
            Reference
Prediction   versicolor virginica
  versicolor          5         1
  virginica           5         9
> # Sens and Spec
> sensitivity = cm[2, 2] / (cm[2, 2] + cm[1, 2])
> sensitivity
[1] 0.9
> specificity = cm[1, 1] / (cm[1, 1] + cm[2, 1])
> specificity
[1] 0.5
```

Threshold = 1

Original testset



Predicted testset

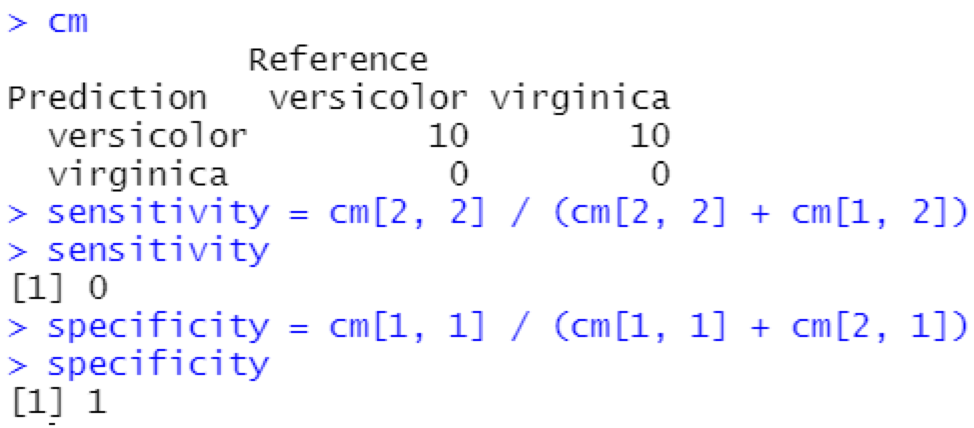


Threshold = 1

	Virginica	**Versicolor**
Virginica (Predicted)	**0**	**0**
Versicolor (Predicted)	**10**	**10**

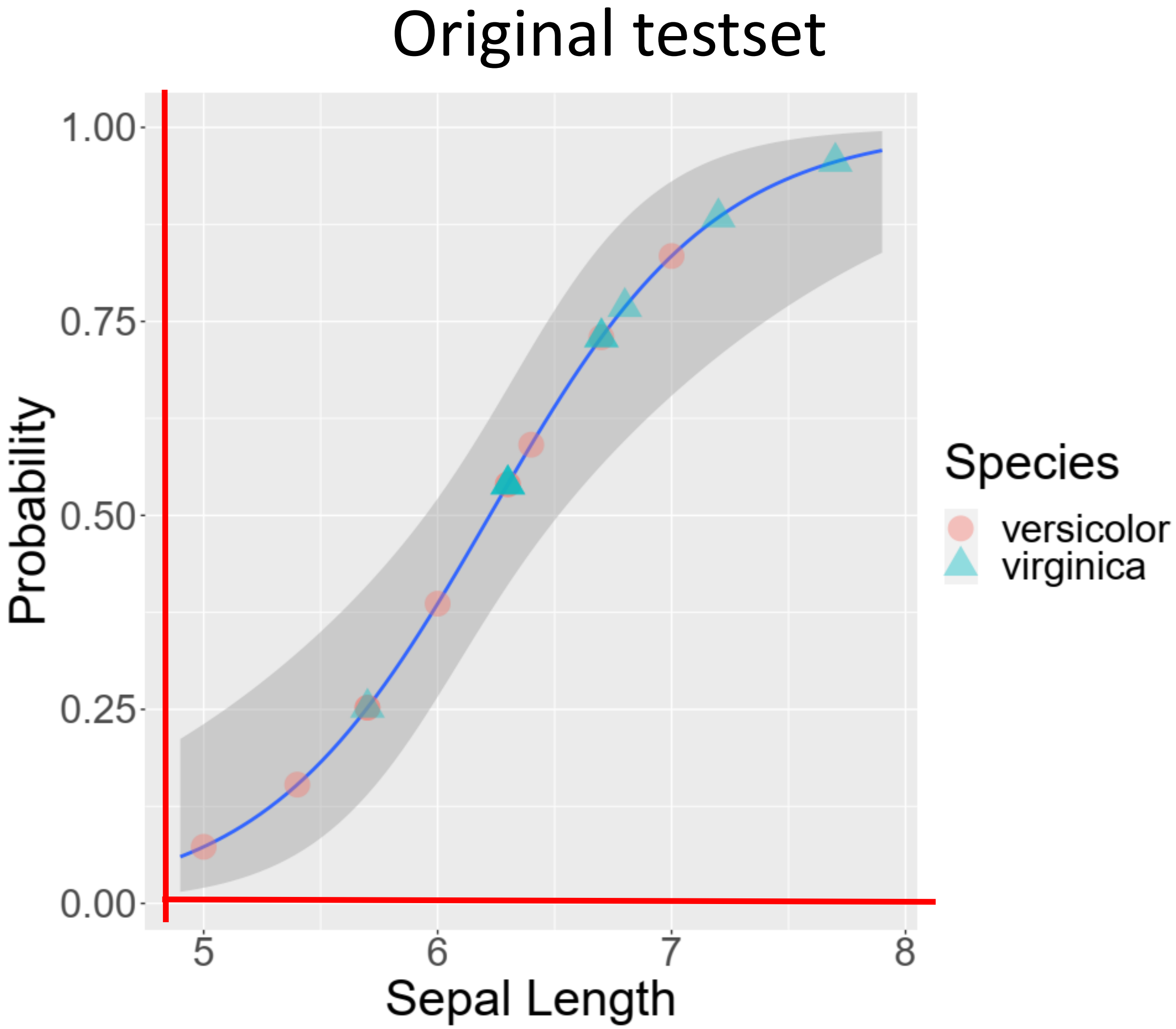
- $\text{Sens} = \frac{\text{TPs}}{\text{TPs}+\text{FNs}} = \frac{0}{10+0} = 0.0$

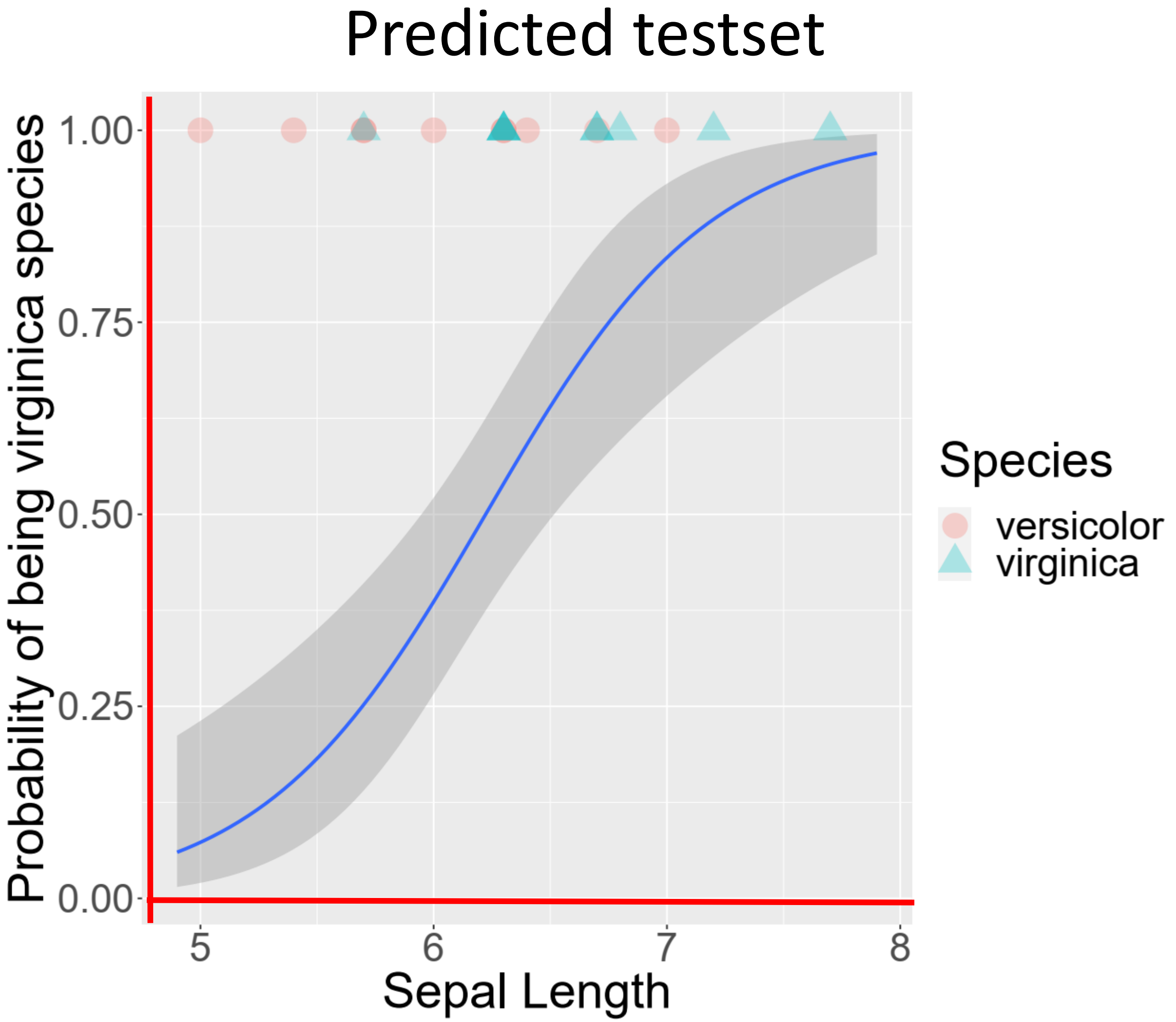
- $\text{Spec} = \frac{\text{TNs}}{\text{FPs}+\text{TNs}} = \frac{10}{10+0} = 1.0$



```
> cm
            Reference
Prediction   versicolor virginica
  versicolor         10        10
  virginica           0         0
> sensitivity = cm[2, 2] / (cm[2, 2] + cm[1, 2])
> sensitivity
[1] 0
> specificity = cm[1, 1] / (cm[1, 1] + cm[2, 1])
> specificity
[1] 1
```

Threshold = 0


Original testset
Probability
1.00
0.75
0.50
0.25
0.00
5
6
7
8
Sepal Length
Species
versicolor
virginica

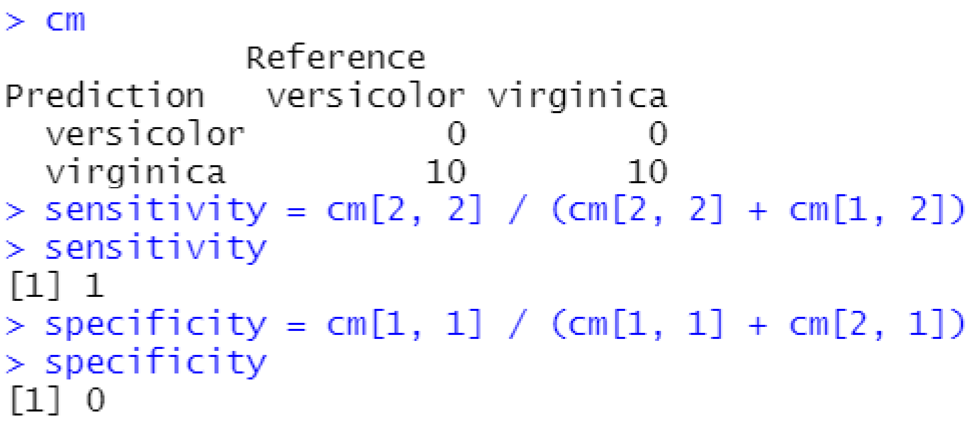

Predicted testset
Probability of being virginica species
1.00
0.75
0.50
0.25
0.00
5
6
7
8
Sepal Length
Species
versicolor
virginica

Threshold = 0

	Virginica	Versicolor
Virginica (Predicted)	**10**	**10**
Versicolor (Predicted)	**0**	**0**

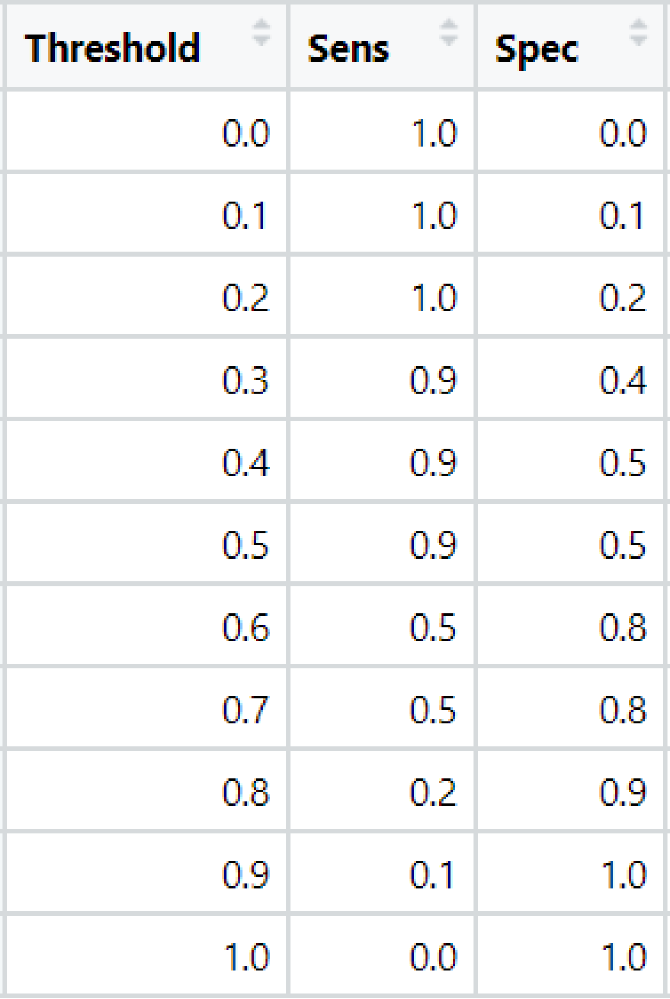
- $\text{Sens} = \frac{\text{TPs}}{\text{TPs+FNs}} = \frac{10}{0+10} = 1.0$

- $\text{Spec} = \frac{\text{TNs}}{\text{FPs+TNs}} = \frac{0}{0+10} = 0.0$



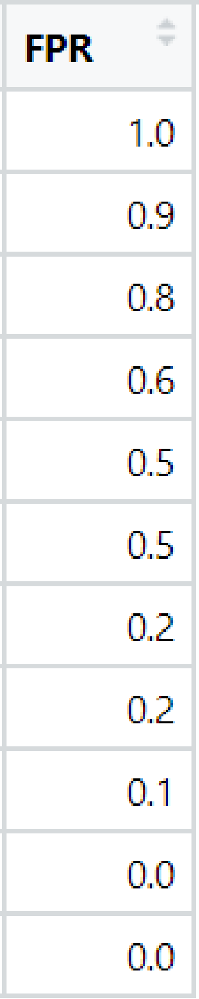
```
> cm
            Reference
Prediction   versicolor virginica
  versicolor          0         0
  virginica          10        10
> sensitivity = cm[2, 2] / (cm[2, 2] + cm[1, 2])
> sensitivity
[1] 1
> specificity = cm[1, 1] / (cm[1, 1] + cm[2, 1])
> specificity
[1] 0
```

False Positive Rate: probability of a positive test, conditioned on truly being negative.



Threshold	Sens	Spec
0.0	1.0	0.0
0.1	1.0	0.1
0.2	1.0	0.2
0.3	0.9	0.4
0.4	0.9	0.5
0.5	0.9	0.5
0.6	0.5	0.8
0.7	0.5	0.8
0.8	0.2	0.9
0.9	0.1	1.0
1.0	0.0	1.0

$$\mathrm{FPR} = 1 - \mathrm{TNR} = \frac{\mathrm{FPs}}{\mathrm{FPs} + \mathrm{TNs}}$$

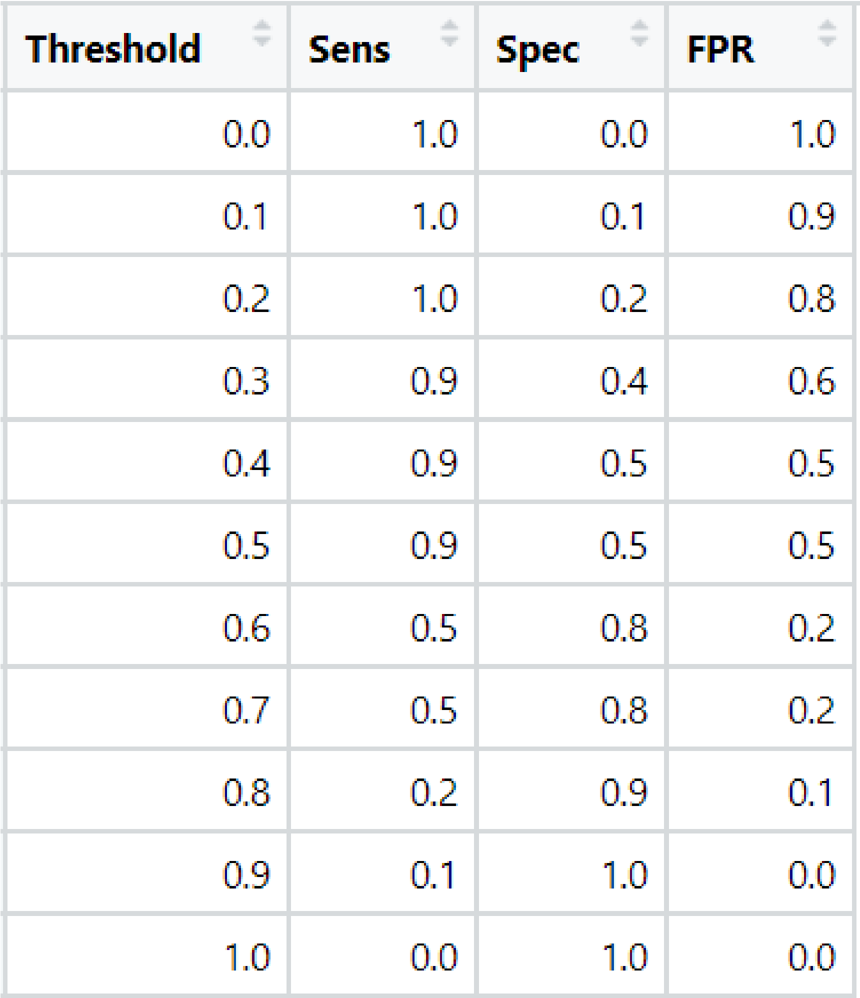


FPR
1.0
0.9
0.8
0.6
0.5
0.5
0.2
0.2
0.1
0.0
0.0

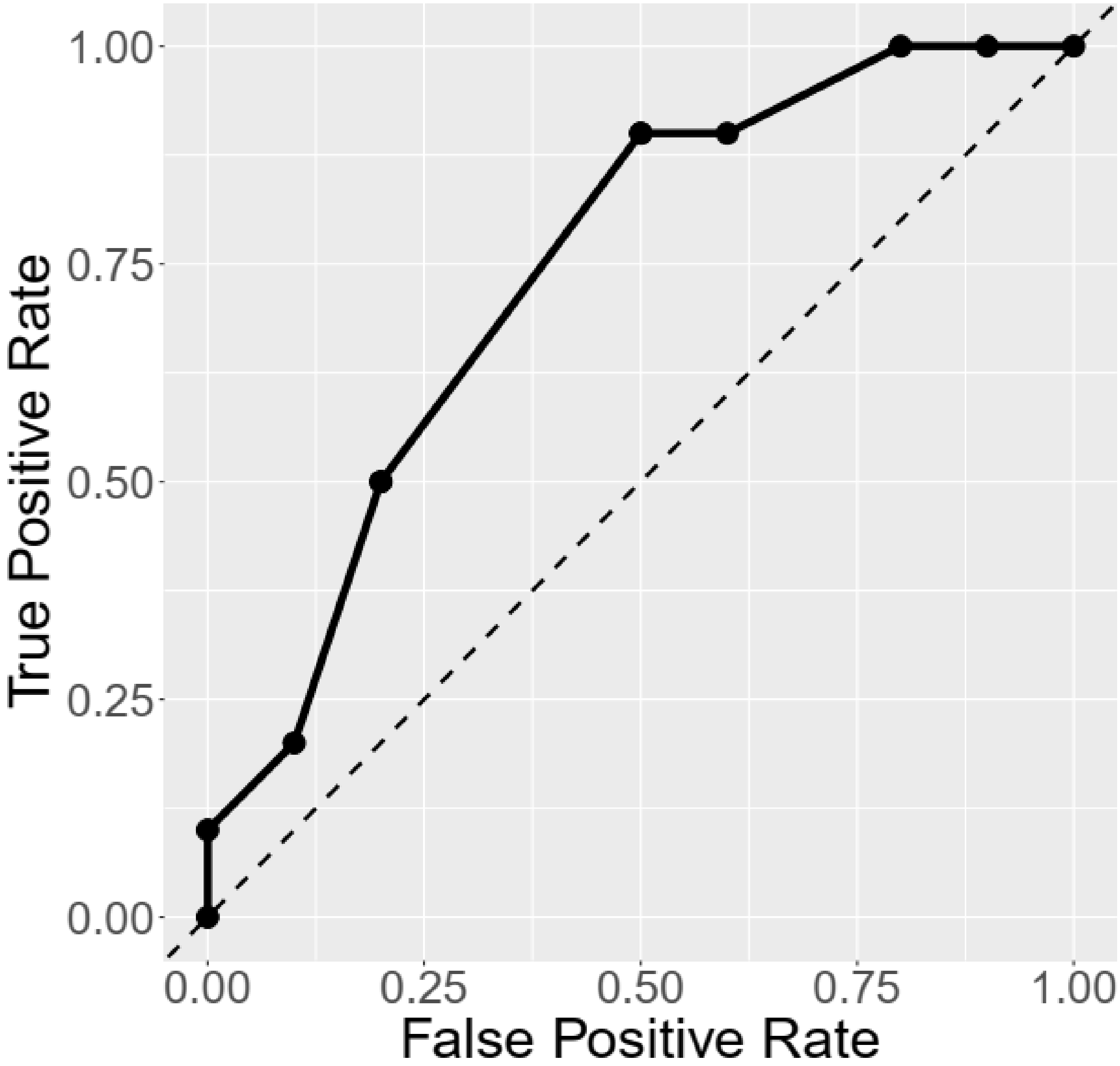
ROC curve
(Receiver Operating Characteristic curve)

- Showing the performance of a classification model at all classification thresholds
- Provide a simple way to summarize information alternative to tons of confusion matrix
- ROC curve is a compromise of
  - False positive rate (FPR): X-axis
  - True positive rate (TPR): Y-axis

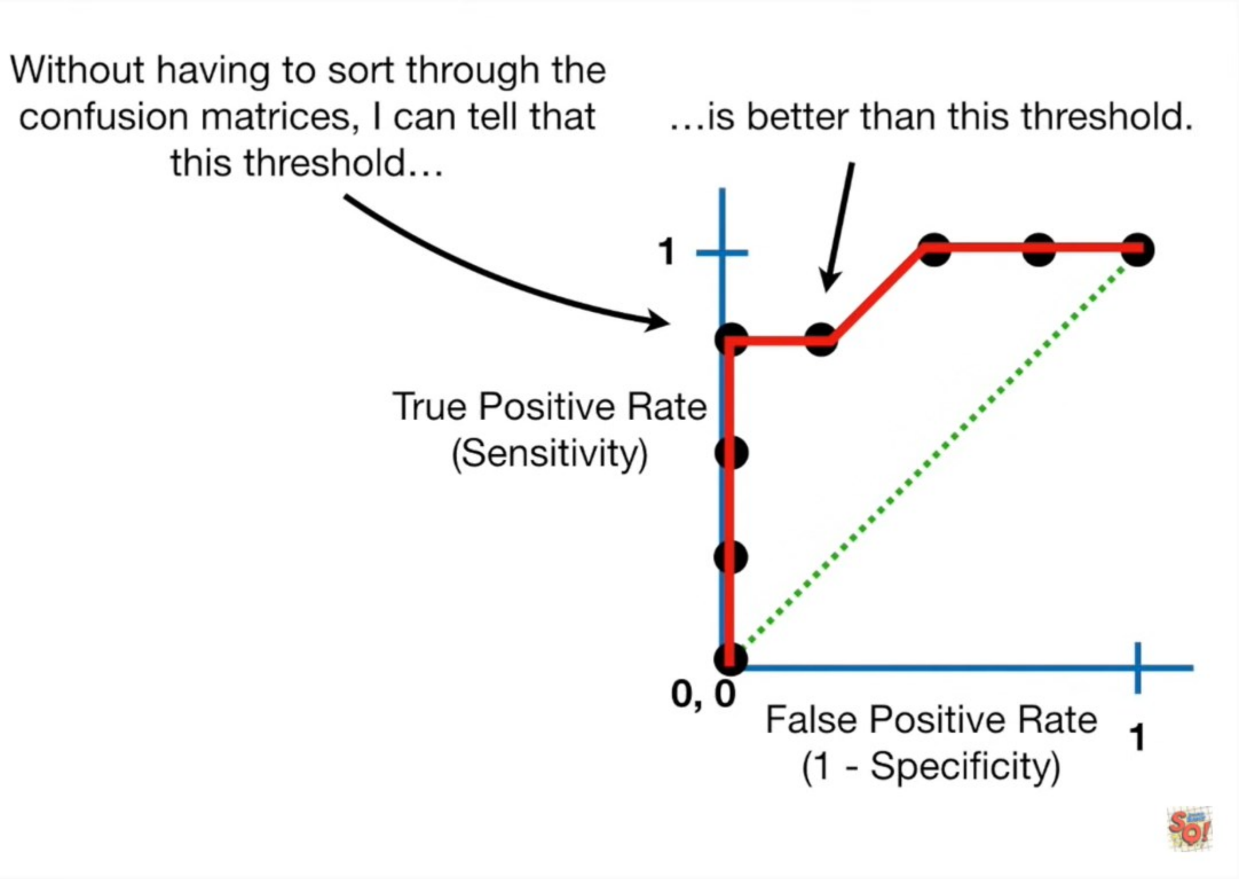
ROC curve



Threshold	Sens	Spec	FPR
0.0	1.0	0.0	1.0
0.1	1.0	0.1	0.9
0.2	1.0	0.2	0.8
0.3	0.9	0.4	0.6
0.4	0.9	0.5	0.5
0.5	0.9	0.5	0.5
0.6	0.5	0.8	0.2
0.7	0.5	0.8	0.2
0.8	0.2	0.9	0.1
0.9	0.1	1.0	0.0
1.0	0.0	1.0	0.0

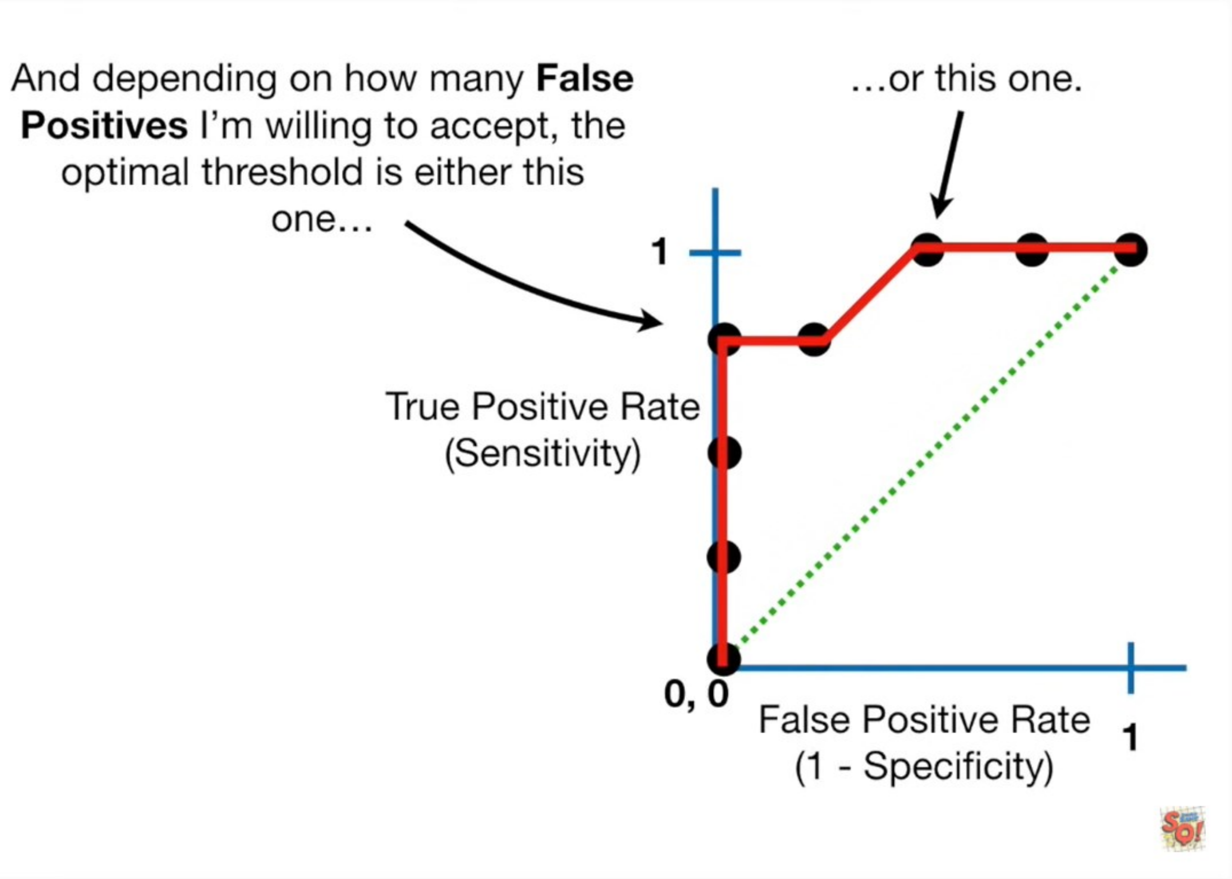


ROC curve



Without having to sort through the confusion matrices, I can tell that this threshold…

…is better than this threshold.

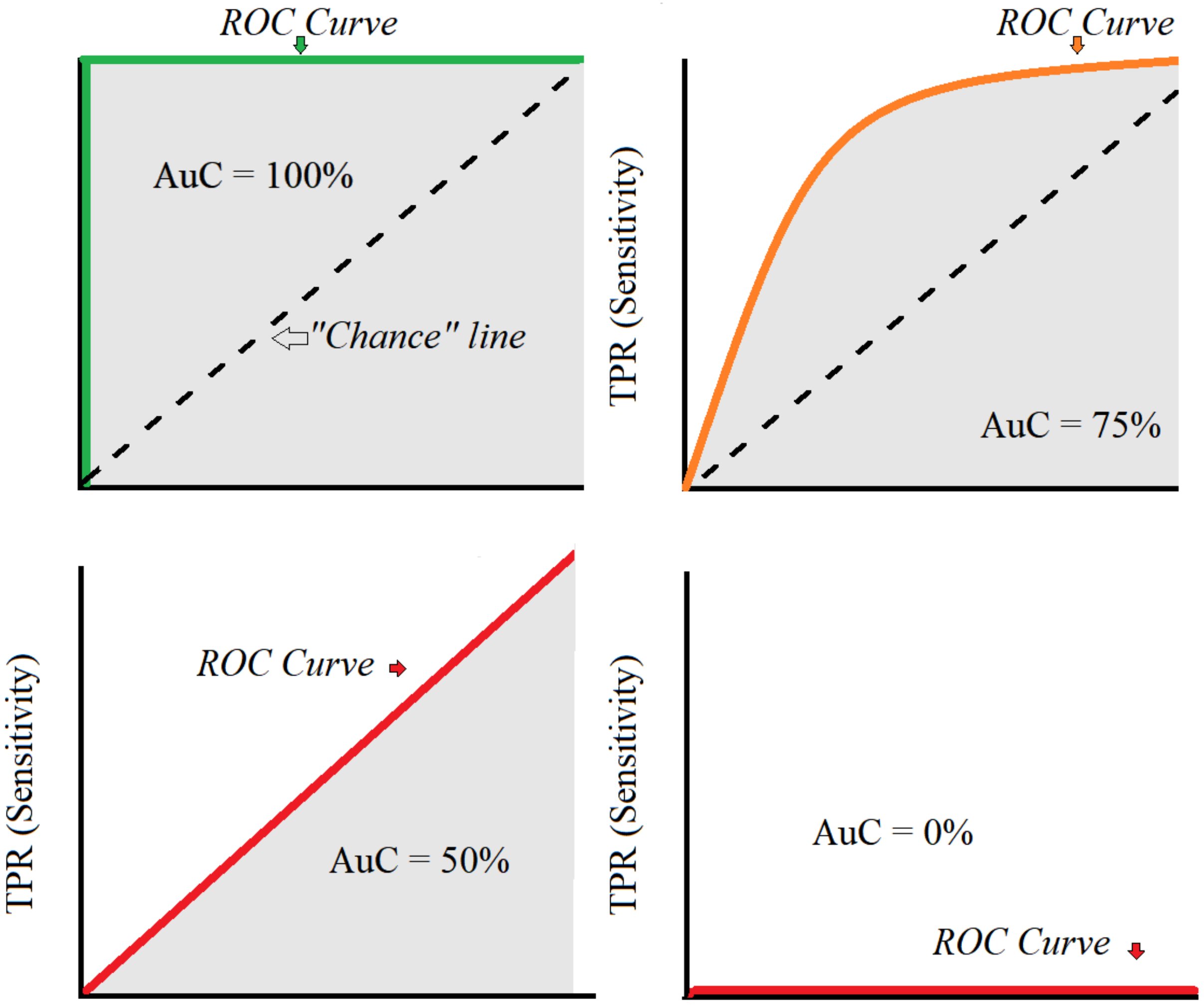


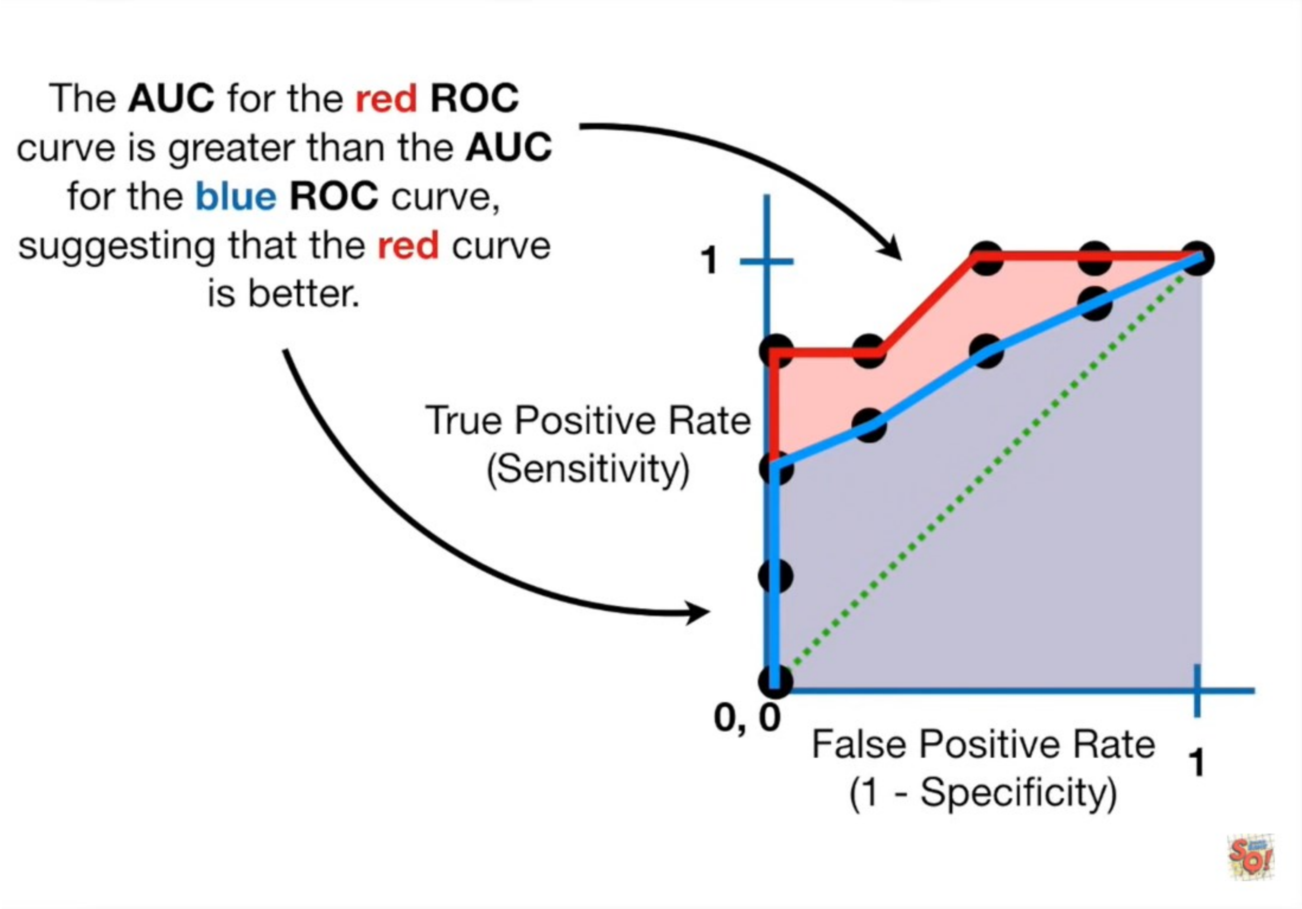
And depending on how many **False Positives** I'm willing to accept, the optimal threshold is either this one…

…or this one.

https://www.youtube.com/watch?v=4jRBRDbJemM&t=821s

AUC (Area Under the Curve)



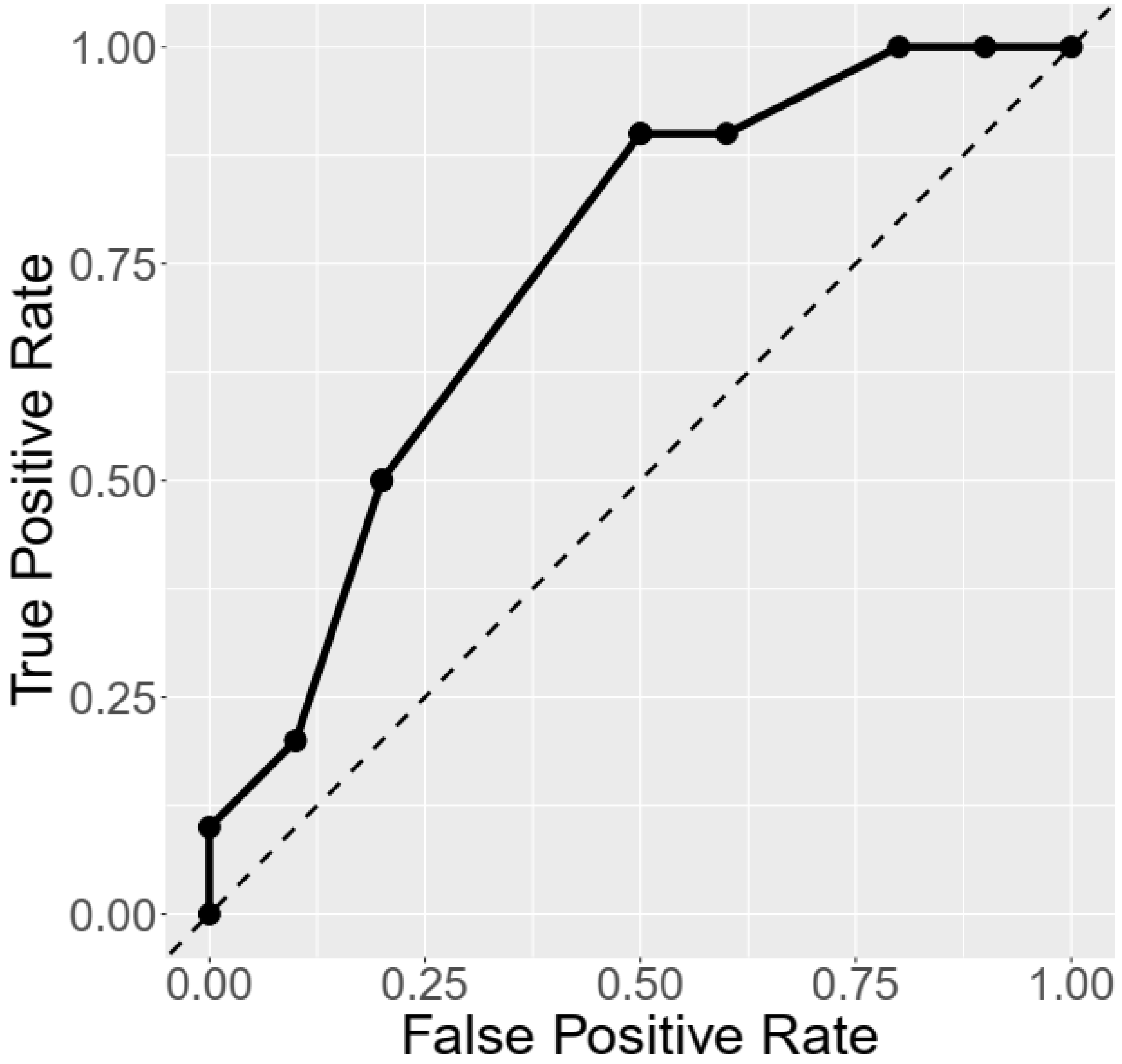


The **AUC** for the **red ROC** curve is greater than the **AUC** for the **blue ROC** curve, suggesting that the **red** curve is better.

https://www.youtube.com/watch?v=4jRBRDbJemM&t=821s

AUC (Area Under the Curve)

```
> ROC_height = (table_sens_spec$Sens[-1]+table_sens_spec$Sens[-length(table_sens_spec$Sens)])/2
> ROC_width = -diff(table_sens_spec$FPR)
> AUC <- sum(ROC_height*ROC_width)
> AUC
[1] 0.74
```



Accuracy: ratio of number of correct predictions to the total number of observations.

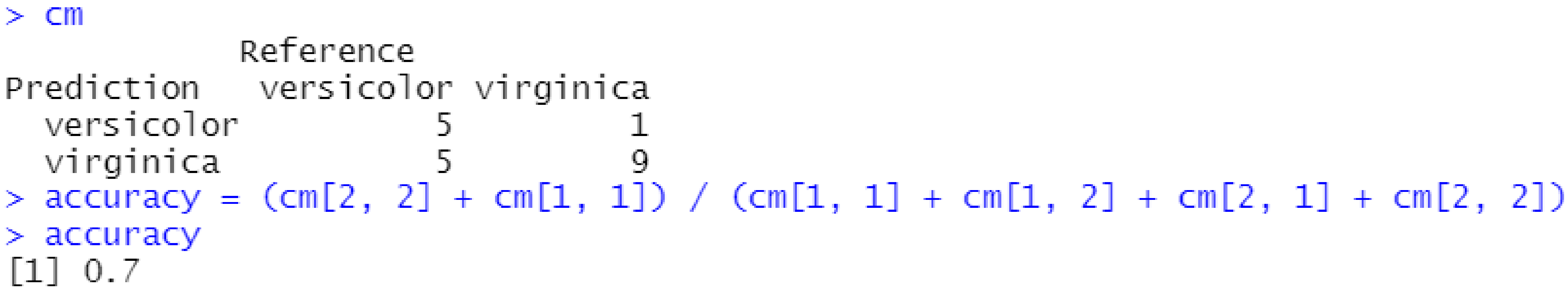
	Actually Positive (1)	**Actually Negative (0)**
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negative (FNs)	True Negative (TNs)

- Accuracy $= \frac{\mathrm{TPs}+\mathrm{TNs}}{\mathrm{TPs}+\mathrm{TNs}+\mathrm{FPs}+\mathrm{FNs}}$

Accuracy

	Virginica	**Versicolor**
Virginica (Predicted)	**9**	**5**
Versicolor (Predicted)	**1**	**5**

- Accuracy $= \dfrac{\text{TPs+TNs}}{\text{TPs+TNs+FPs+FNs}} = \dfrac{9+5}{9+5+5+1} = 0.7$



```
> cm
            Reference
Prediction   versicolor virginica
  versicolor          5         1
  virginica           5         9
> accuracy = (cm[2, 2] + cm[1, 1]) / (cm[1, 1] + cm[1, 2] + cm[2, 1] + cm[2, 2])
> accuracy
[1] 0.7
```

Accuracy

- What if the test data have 990 Virginica and 10 Versicolor?

	Virginica	**Versicolor**
Virginica (Predicted)	**990**	**10**
Versicolor (Predicted)	**0**	**0**

- Accuracy $= \frac{\text{TPs+TNs}}{\text{TPs+TNs+FPs+FNs}} = \frac{990+0}{990+0+10+0} = 0.99$

- The accuracy now is 0.99 but completely miss classified all Versicolor species → Imbalance

Precision and Recall

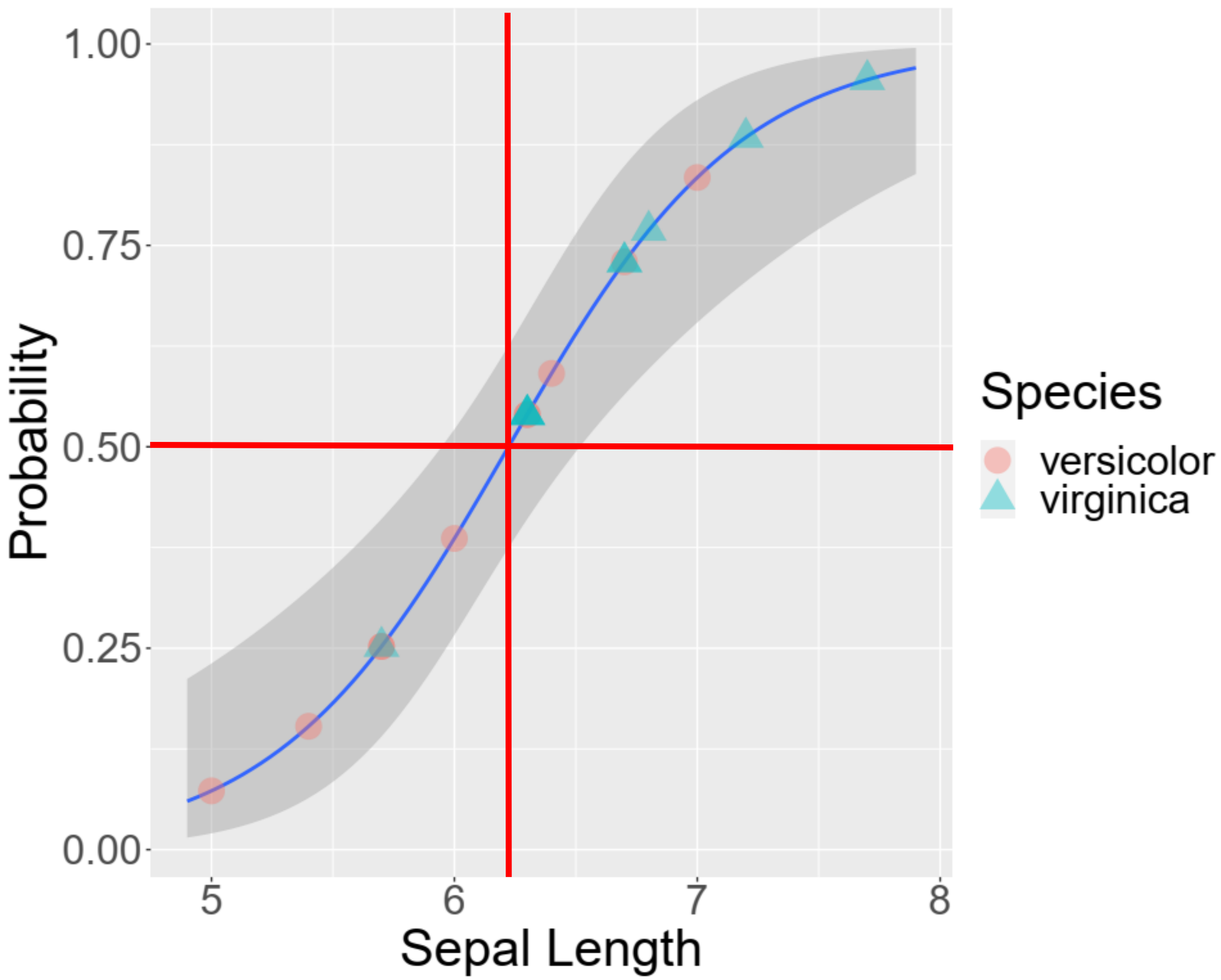
	Actually Positive (1)	**Actually Negative (0)**
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negative (FNs)	True Negative (TNs)

- Precision (positive predictive value): The proportion of positive tests that were correctly classified. $\mathrm{PPV} = \frac{\mathrm{TPs}}{\mathrm{TPs}+\mathrm{FPs}}$

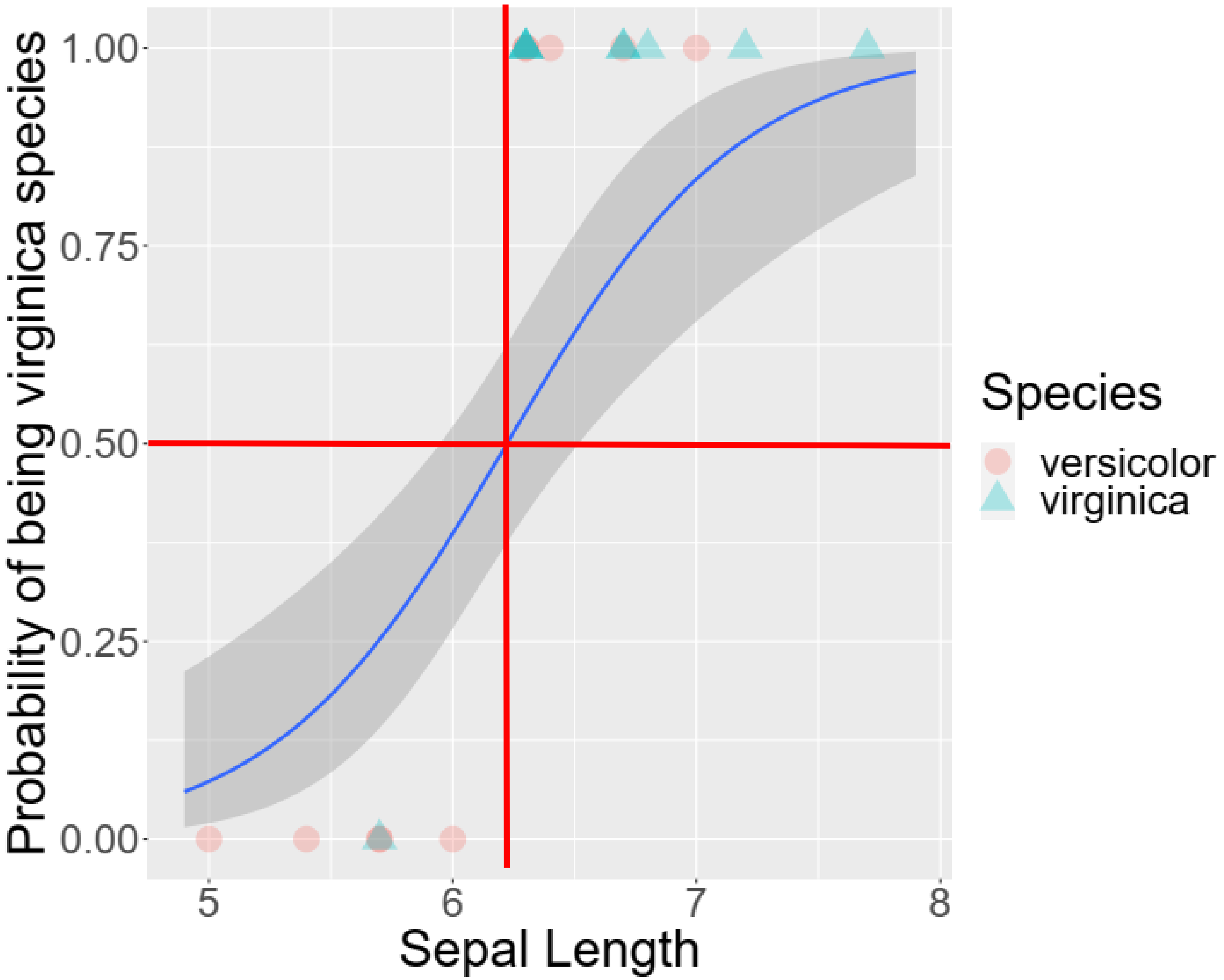
- Recall = Sensitivity: $\mathrm{TPR} = \frac{\mathrm{TPs}}{\mathrm{TPs}+\mathrm{FNs}}$

Precision and Recall

Original testset



Predicted testset

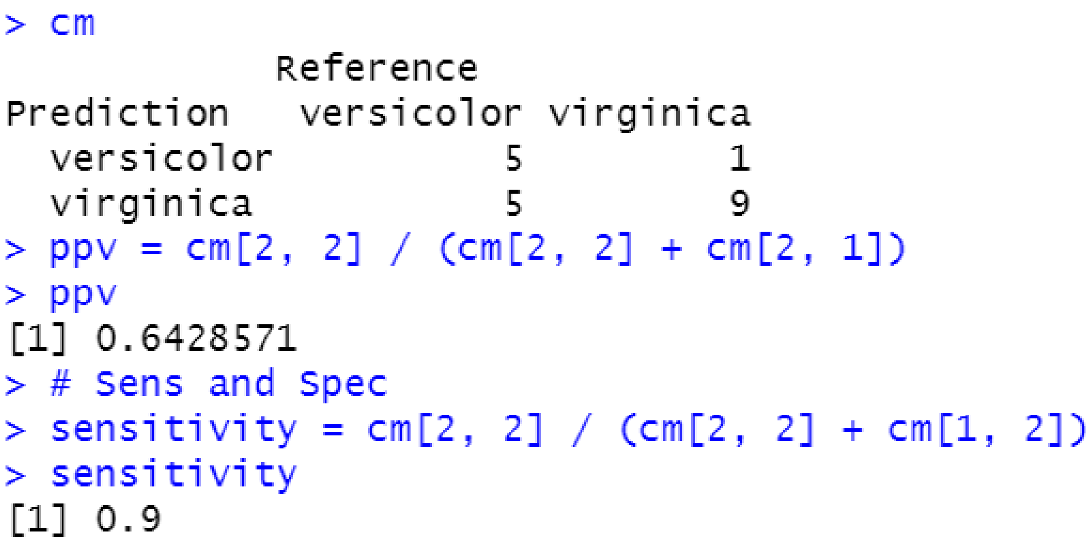


Precision and Recall

	Virginica	Versicolor
Virginica (Predicted)	**9**	**5**
Versicolor (Predicted)	**1**	**5**

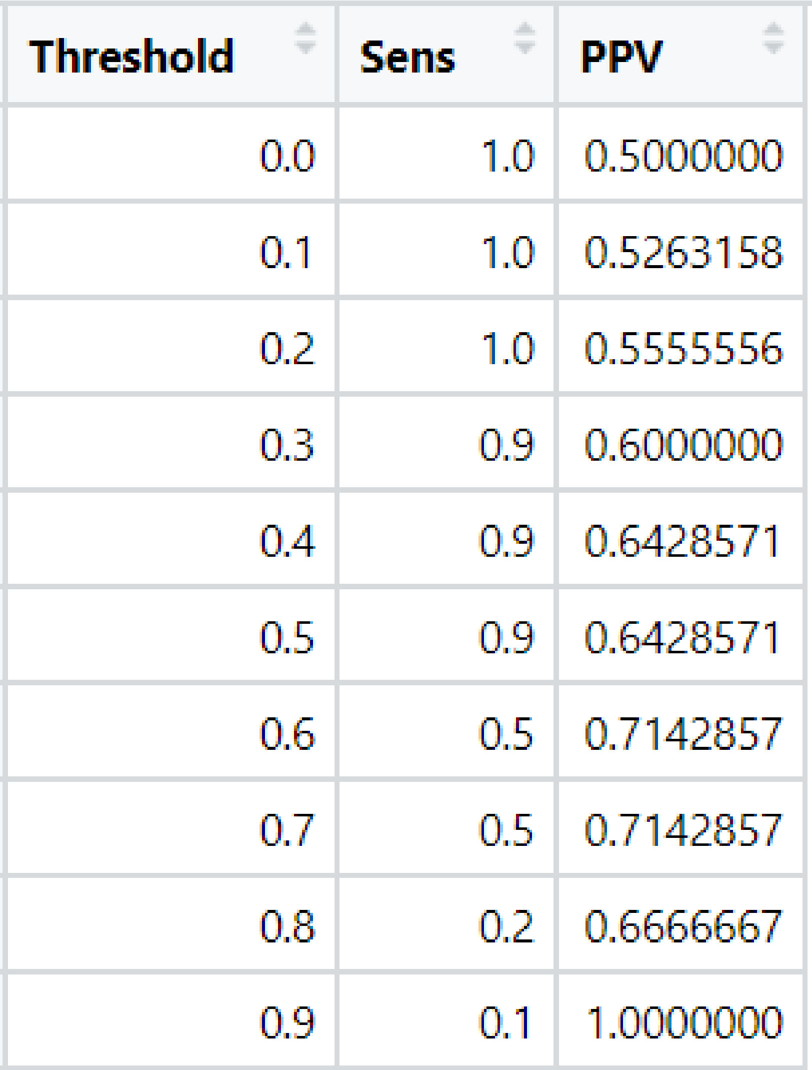
- $\text{PPV} = \frac{\text{TPs}}{\text{TPs+FPs}} = \frac{9}{9+5} = 0.64$

- $\text{TPR} = \text{Sens} = \frac{\text{TPs}}{\text{TPs+FNs}} = \frac{9}{9+1} = 0.9$

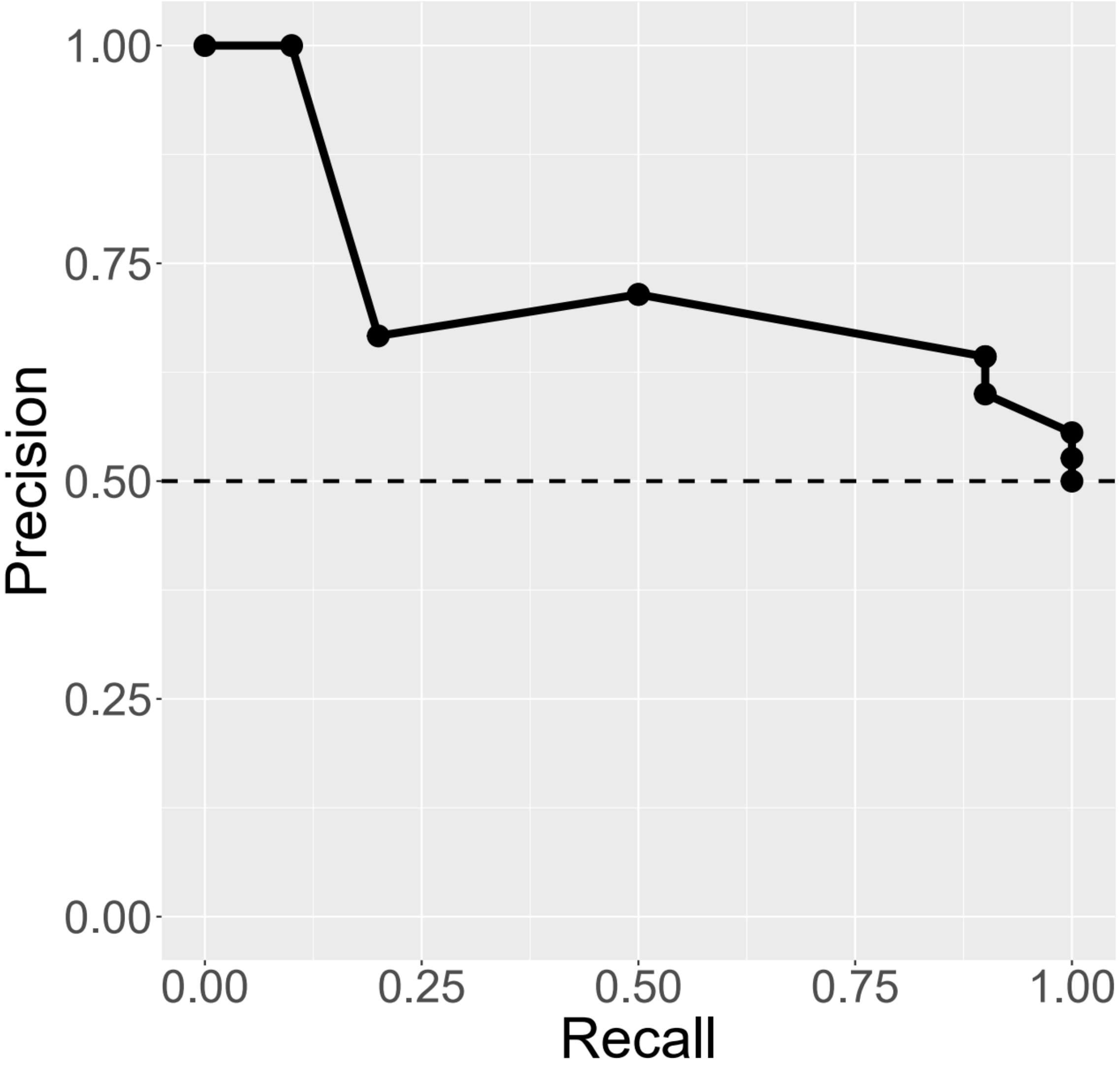


```
> cm
             Reference
Prediction   versicolor virginica
  versicolor          5         1
  virginica           5         9
> ppv = cm[2, 2] / (cm[2, 2] + cm[2, 1])
> ppv
[1] 0.6428571
> # Sens and Spec
> sensitivity = cm[2, 2] / (cm[2, 2] + cm[1, 2])
> sensitivity
[1] 0.9
```

PR curve

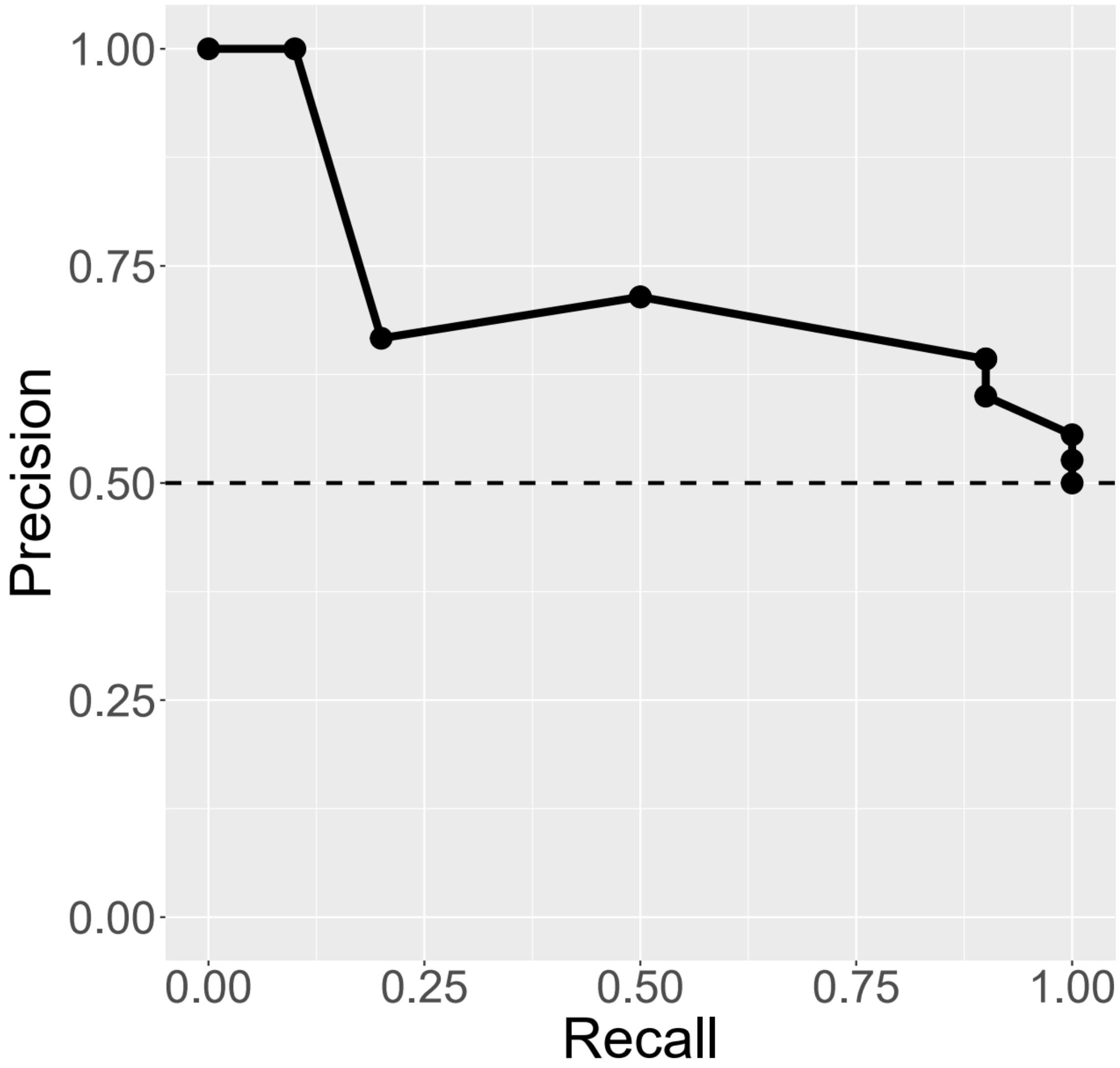


Threshold	Sens	PPV
0.0	1.0	0.5000000
0.1	1.0	0.5263158
0.2	1.0	0.5555556
0.3	0.9	0.6000000
0.4	0.9	0.6428571
0.5	0.9	0.6428571
0.6	0.5	0.7142857
0.7	0.5	0.7142857
0.8	0.2	0.6666667
0.9	0.1	1.0000000



AUPRC (Area Under the Precision-RecallCurve)

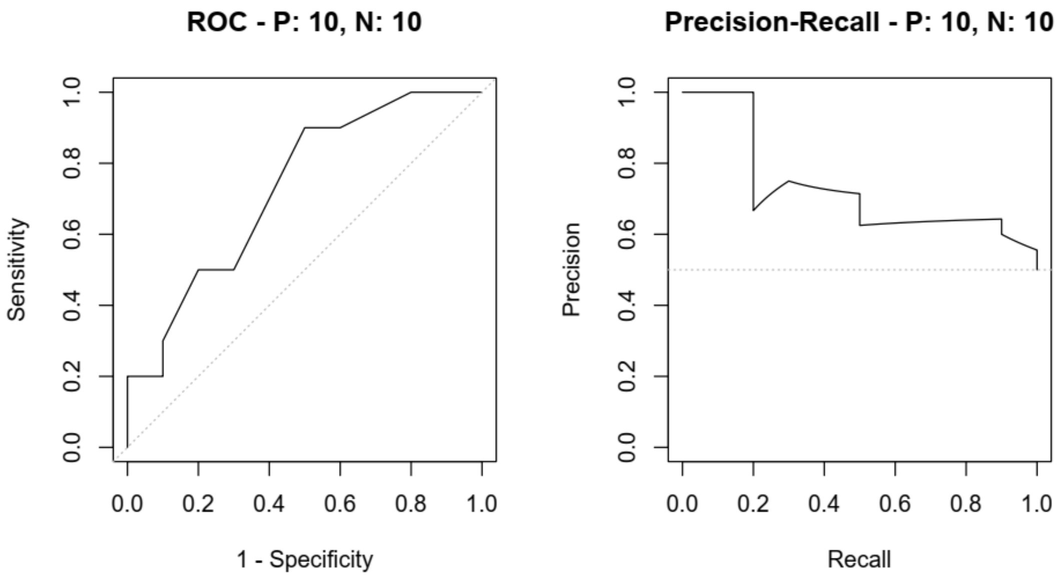
```
> PR_height = (table_sens_ppv$PPV[-1]+table_sens_ppv$PPV[-length(table_sens_ppv$PPV)])/2
> PR_width = -diff(table_sens_ppv$Sens)
> AUPRC <- sum(PR_height*PR_width)
> AUPRC
[1] 0.7196825
```

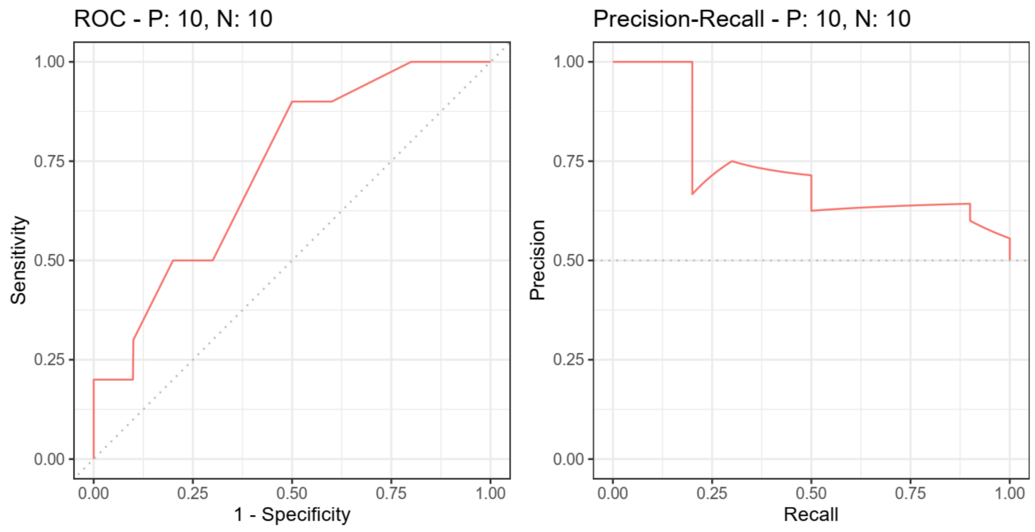


Package for ROC, AUC, PR and AUPRC

```
library(precrec)
ROCnPR <- evalmod(scores = testDT$pred, labels = testDT$Species)
plot(ROCnPR)
autoplot(ROCnPR)
autoplot(ROCnPR, "ROC")
autoplot(ROCnPR, "PRC")
aucs <- auc(ROCnPR)
knitr::kable(aucs)
```

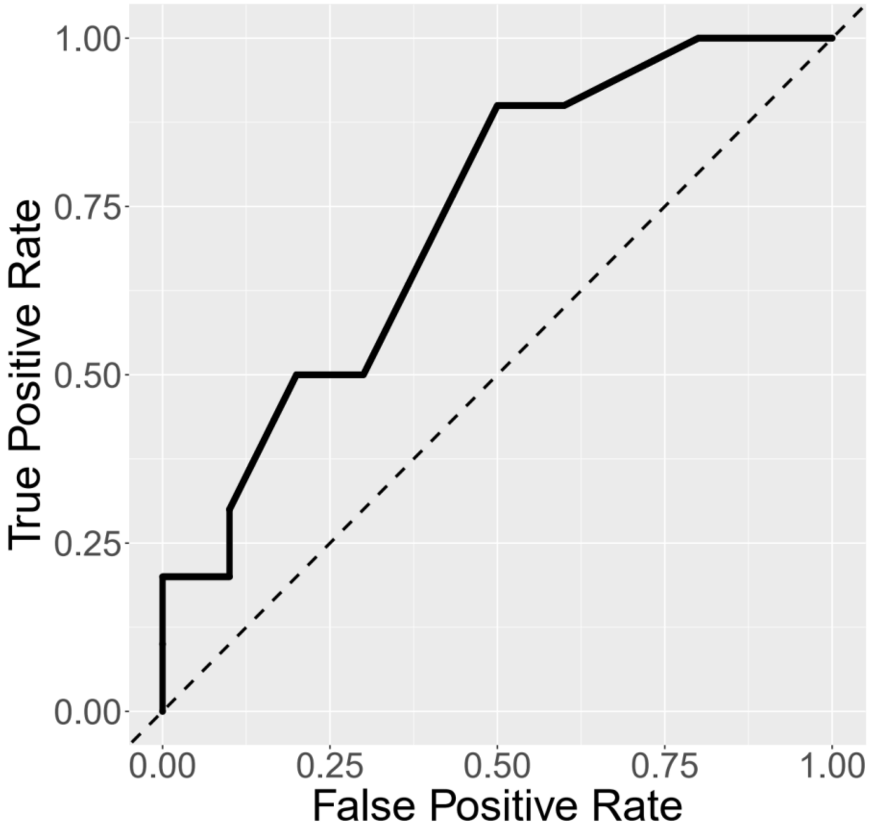
```
|modnames | dsids|curvetypes |      aucs|
|:--------|-----:|:----------|---------:|
|m1       |     1|ROC        | 0.7300000|
|m1       |     1|PRC        | 0.7288747|
```



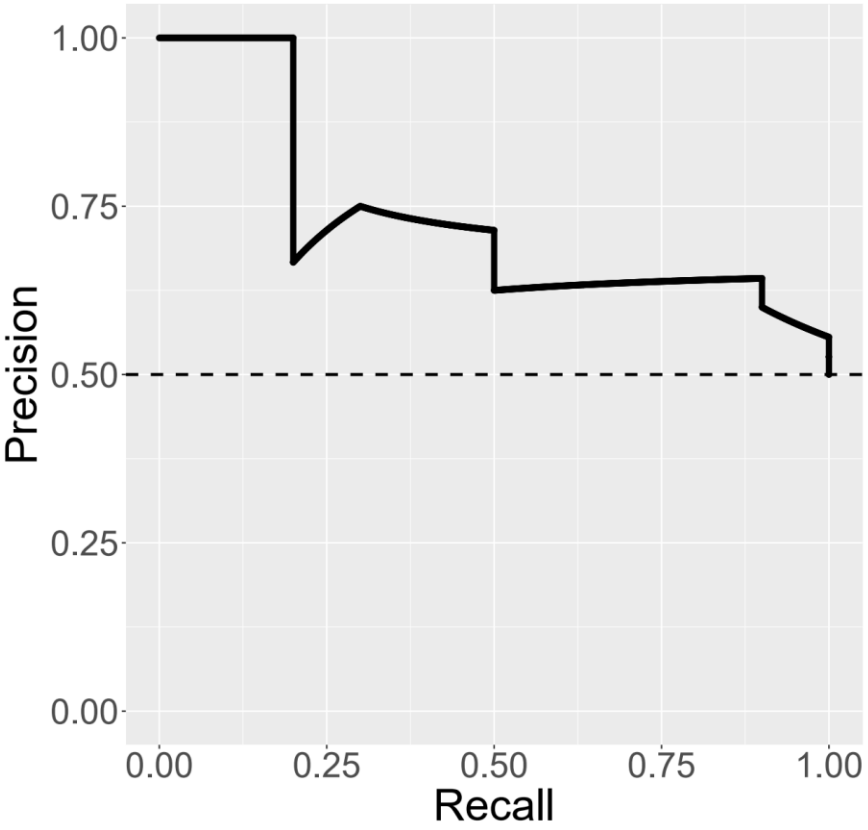


Package for ROC, AUC, PR and AUPRC

```
dt.roc <- as.data.frame(ROCnPR)[as.data.frame(ROCnPR)$type == "ROC",]
ggplot(dt.roc, aes(x=x, y=y)) + geom_path(size = 2) + geom_point() +
  labs(x = "False Positive Rate", y = "True Positive Rate") + xlim(0,1) + ylim(0,1) +
  geom_abline(intercept= 0, slope = 1, size =1, linetype=2) +
  theme(text = element_text(size = 30))
```

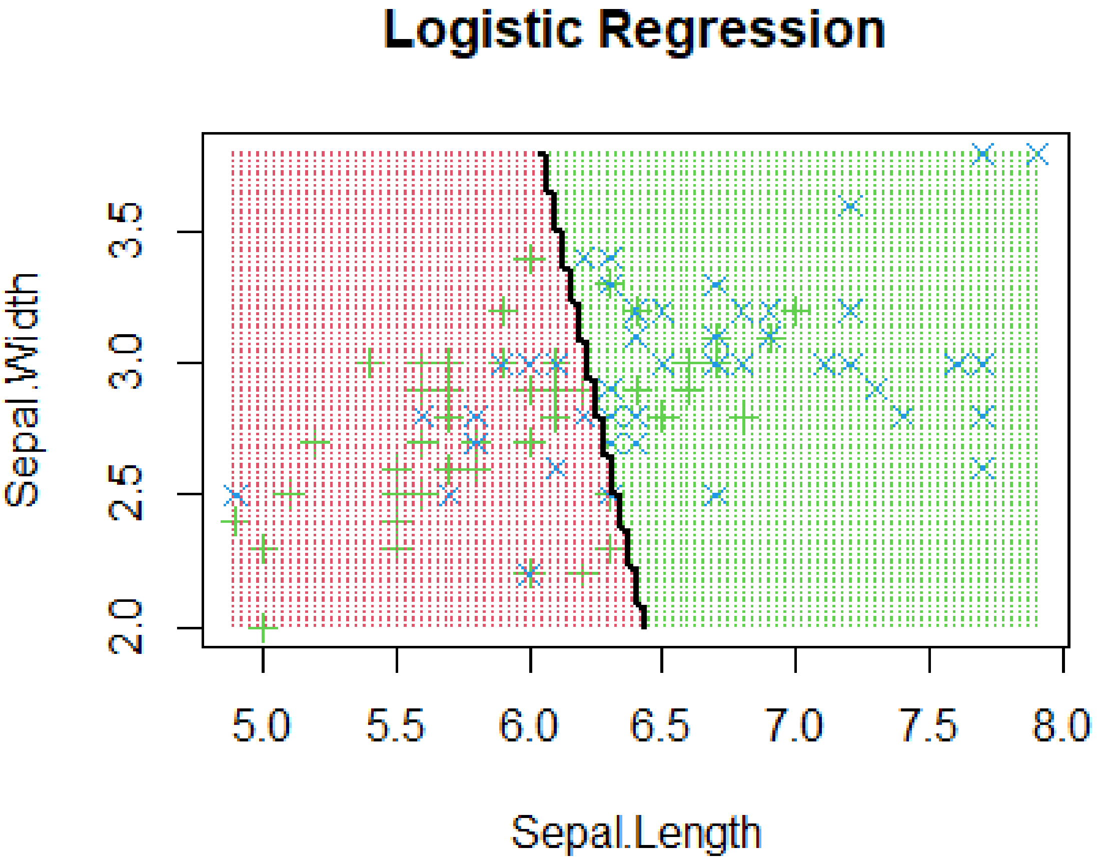


```
dt.pr <- as.data.frame(ROCnPR)[as.data.frame(ROCnPR)$type == "PRC",]
ggplot(dt.pr, aes(x=x, y=y)) + geom_path(size = 2) + geom_point() +
  labs(x = "Recall", y = "Precision") + xlim(0,1) + ylim(0,1) +
  geom_hline(yintercept = 0.5, size =1, linetype=2) +
  theme(text = element_text(size = 30))
```

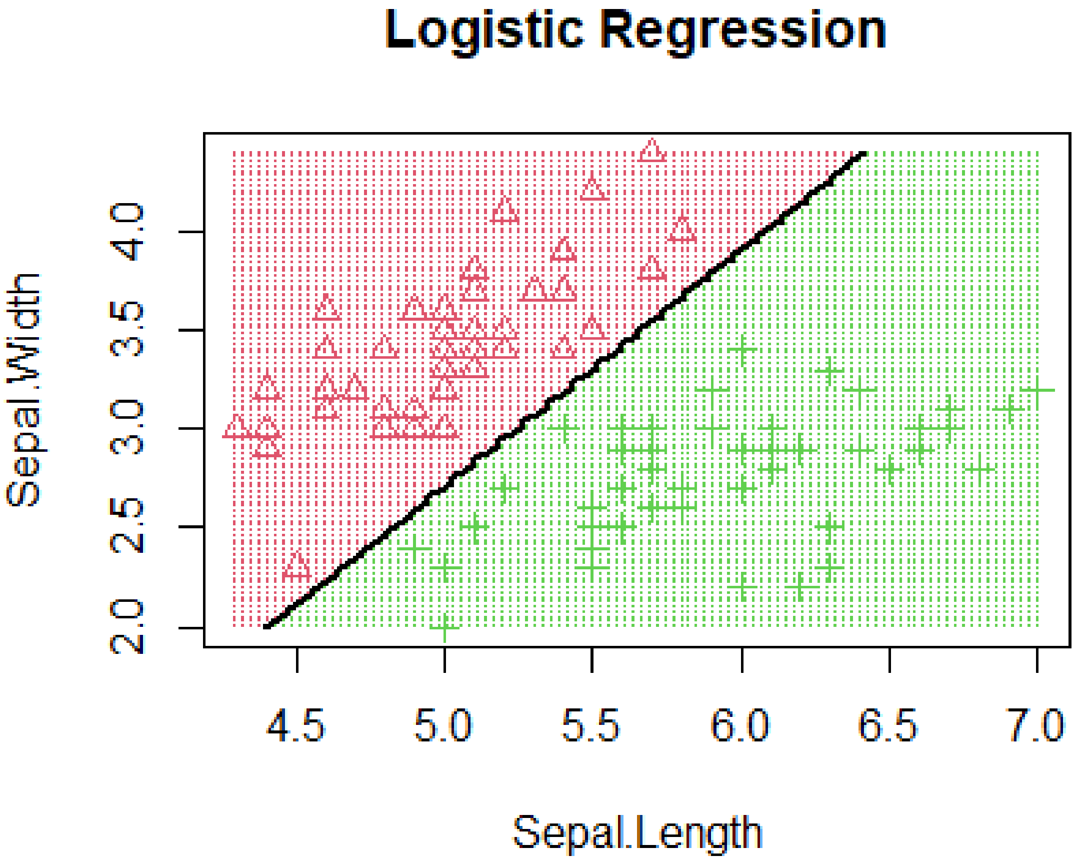


Decision Boundary

versicolor / virginica



setosa / versicolor



https://rpubs.com/ZheWangDataAnalytics/DecisionBoundary

Exercises

1. Using Sepal.Length for Species (virginica/versicolor) prediction on test data on multiple thresholds with step=0.01:
   1.1. Generate Sensitivity, Specificity, False positive rate and Precision.
   1.2. Perform ROC cuvre, AUC, PR curve and AUPRC.

2. Using Sepal.Length + Sepal.Witdh for Species (virginica/versicolor) prediction on test data:
   2.1. Perform ROC cuvre, AUC, PR curve and AUPRC.
   2.2. Perform Decision Boundary

3. Using Sepal.Length + Sepal.Witdh for Species (setosa/versicolor) prediction on test data on multiple thresholds with step=0.01:
   3.1. Perform ROC cuvre, AUC, PR curve and AUPRC.
   3.2. Perform Decision Boundary