Stat 340: Intro. to Classification and Logistic Regression

Example: Crab species identification

We will work with a data set about Leptograpsus crabs originally presented in

Campbell, N.A. and Mahon, R.J. (1974) A multivariate study of variation in two species of rock crab of genus Leptograpsus. Australian Journal of Zoology 22, 417–425.

They have also been discussed previously in

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

There are two species of this crab; we will attempt to predict the species of a crab based on measurements of its physical dimensions. The data we are working with contains 5 morphological measurements on 200 crabs, 100 each of two species of Leptograpsus crabs collected at Fremantle, W. Australia.

The variables in this data set are as follows:

- `sp`: species - “B” or “O” for blue or orange.
- `sex`: the crab’s sex
- `FL`: frontal lobe size (mm).
- `RW`: rear width (mm).
- `CL`: carapace length (mm).
- `CW`: carapace width (mm).
- `BD`: body depth (mm).

Binary encoding of response variable

It is traditional in logistic regression to use an indicator variable for the response variable:

$$Y_i = \begin{cases} 1 & \text{if crab number } i \text{ is an orange crab} \\ 0 & \text{otherwise (if a blue crab)} \end{cases}$$

```r
crabs <- crabs %>%
  mutate(
    sp_01 = ifelse(sp == "O", 1, 0)
  )

head(crabs)
```

```
##   sp sex   FL   RW   CL   CW   BD sp_01
## 1  O   F 21.4 18.0 41.2 46.2 18.7     1
## 2  O   M 15.1 11.4 30.2 33.3 14.0     1
## 3  O   M 18.8 13.4 37.2 41.1 17.5     1
## 4  O   F 22.5 17.2 43.0 48.7 19.8     1
## 5  O   M 14.2 10.7 27.8 30.9 12.7     1
## 6  B   M 17.9 14.1 39.7 44.6 16.8     0
```

```r
dim(crabs)
```

```
## [1] 200   8
```

Train/test split

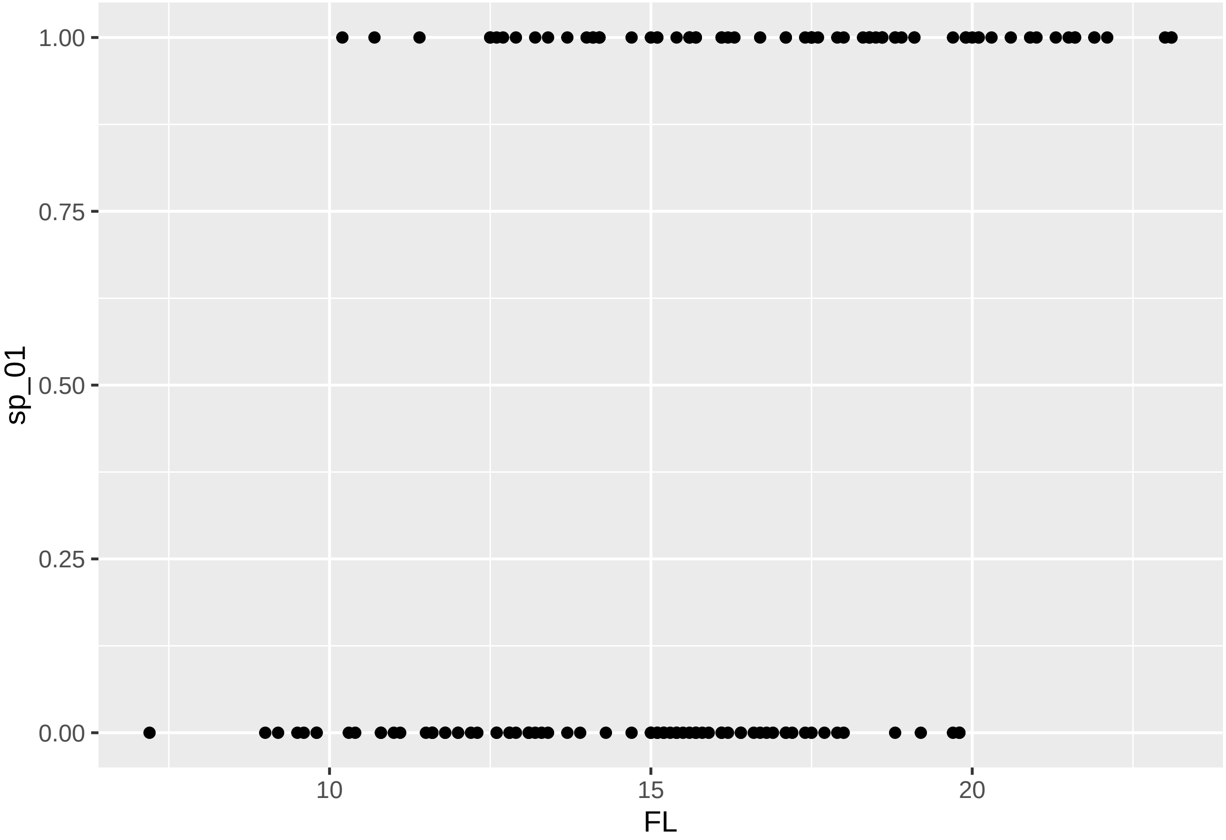
```
set.seed(64781) # seed generated at random.org

train_inds <- caret::createDataPartition(crabs$sp, p = 0.75)

train_crabs <- crabs %>% dplyr::slice(train_inds[[1]])
test_crabs <- crabs %>% dplyr::slice(-train_inds[[1]])
```

Plot of the data

```
ggplot(data = train_crabs, mapping = aes(x = FL, y = sp_01)) +
  geom_point()
```



Train logistic regression model

Note:

- Behind the scenes, sp is converted to 0/1 representation by the train function
- By default, assignment is in alphabetic order, so “B” goes to 0 and “O” goes to 1.

```
logistic_fit <- train(
  form = sp ~ FL,
  data = train_crabs,
  family = "binomial", # this is an argument to glm; response is 0 or 1, binomial
  method = "glm", # method for fit; "generalized linear model"
  trControl = trainControl(method = "none")
)
```

Print model summary (coefficient estimates and p-values for tests)

```
summary(logistic_fit$finalModel)
```

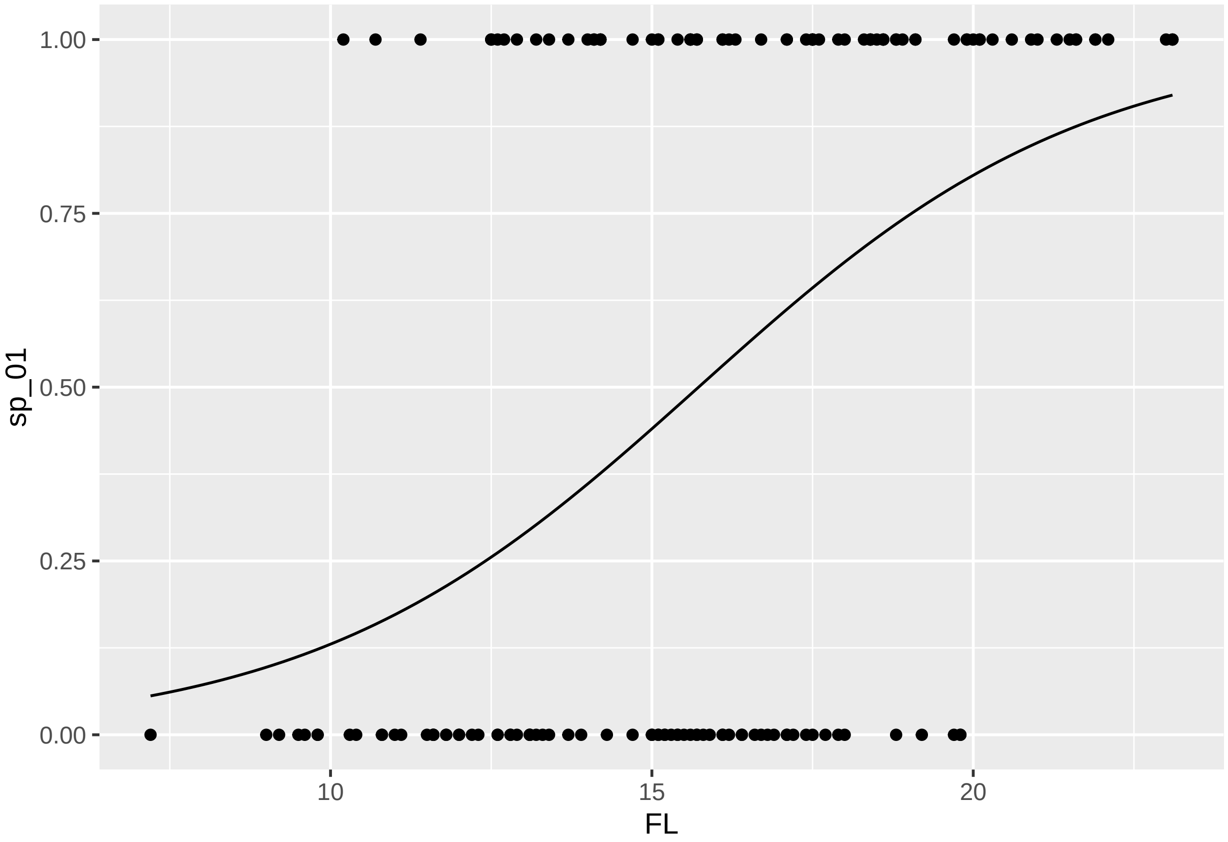
```
## 
## Call:
## NULL
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -1.77800  -1.02364   0.03439   0.94002   1.99013  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -5.21258    1.03635   -5.03 4.91e-07 ***
## FL           0.33145    0.06486    5.11 3.22e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 207.94  on 149  degrees of freedom
## Residual deviance: 173.04  on 148  degrees of freedom
## AIC: 177.04
## 
## Number of Fisher Scoring iterations: 3
```

Plot estimated class probability function

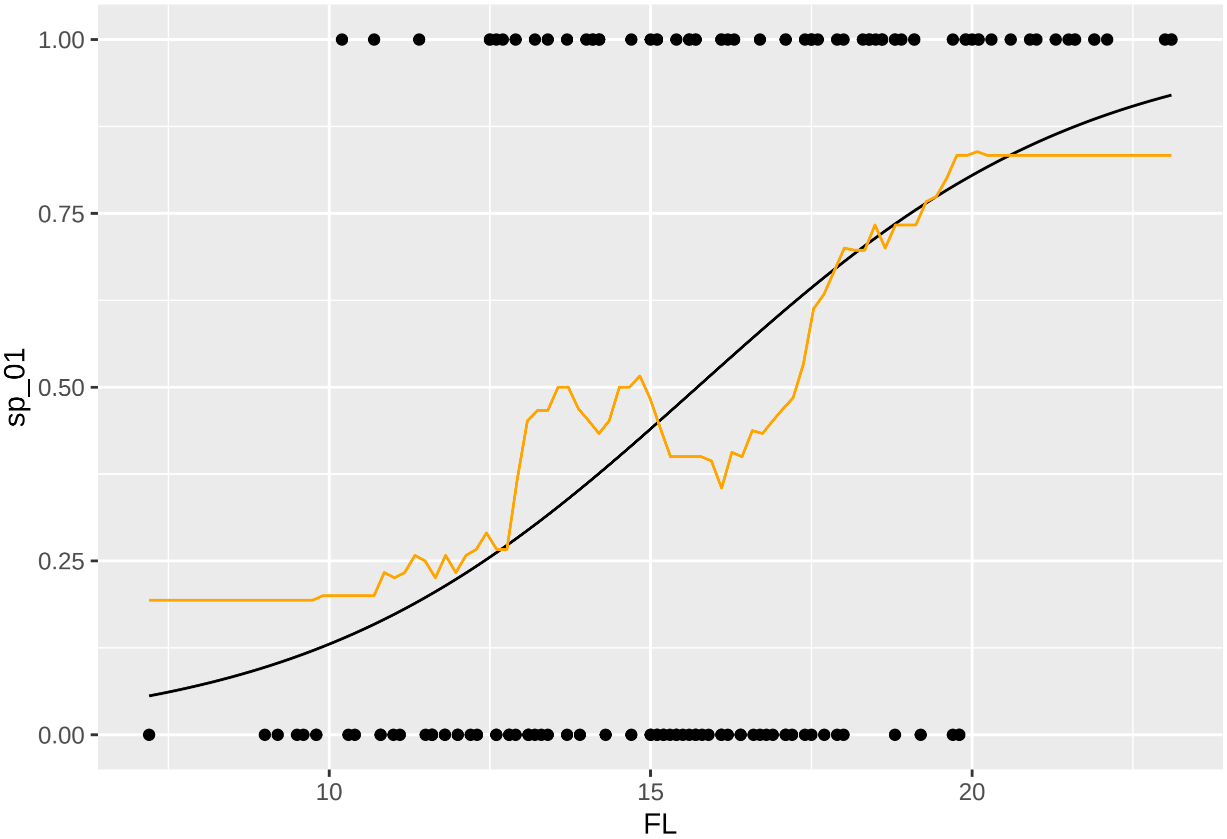
Note: code is essentially identical to what we used for KNN.

```
predict_logistic_probs <- function(x, sp) {
  f_hats <- predict(logistic_fit, newdata = data.frame(FL = x), type = "prob")
  f_hats[[sp]]
}

ggplot(data = train_crabs, mapping = aes(x = FL, y = sp_01)) +
  geom_point() +
  stat_function(fun = predict_logistic_probs,
    args = list(sp = "O")) +
  ylim(0, 1)
```



Code suppressed, but here's a comparison to a KNN fit with K = 30 neighbors:



What's the estimated probability that a crab with a frontal lobe size of 20 mm is an orange crab?

```
predict(logistic_fit, newdata = data.frame(FL = 20), type = "prob")
```

```
##          B        O
## 1 0.195218 0.804782
```

What's our decision boundary?

What's the interpretation of $\hat{\beta}_1$?

Get test set predictions and error rate

```
test_sp_hat <- predict(logistic_fit, newdata = test_crabs, type = "raw")
test_sp_hat
```

```
##  [1] O O O O B B B O O B O O O B O B B O B O B O B O B B B O O B B B B B B
## [36] B B B B O B B O B B B O B B B
## Levels: B O
```

```
mean(test_crabs$sp != test_sp_hat)
```

```
## [1] 0.32
```