Crabs

Andrea Huerfano

September 4, 2019

The function summary will be used to have a first idea of the data structure.

```
library(MASS)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.5.3
```

```
data(crabs)
colnames(crabs)
```

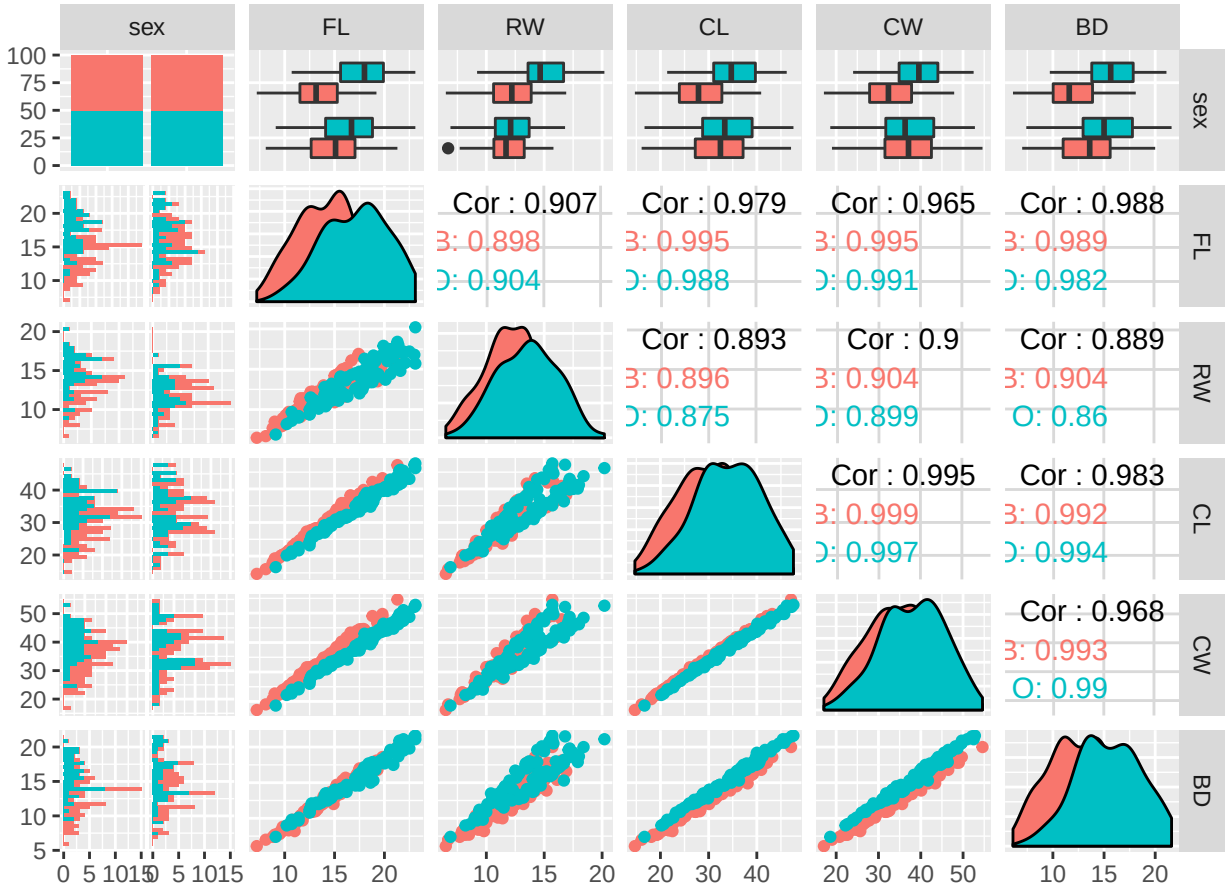
```
## [1] "sp"    "sex"   "index" "FL"    "RW"    "CL"    "CW"    "BD"
```

```
crabs<-crabs[,-3]
summary(crabs)
```

```
##  sp      sex           FL              RW              CL
##  B:100   F:100   Min.   : 7.20   Min.   : 6.50   Min.   :14.70
##  O:100   M:100   1st Qu.:12.90   1st Qu.:11.00   1st Qu.:27.27
##                  Median :15.55   Median :12.80   Median :32.10
##                  Mean   :15.58   Mean   :12.74   Mean   :32.11
##                  3rd Qu.:18.05   3rd Qu.:14.30   3rd Qu.:37.23
##                  Max.   :23.10   Max.   :20.20   Max.   :47.60
##        CW              BD
##  Min.   :17.10   Min.   : 6.10
##  1st Qu.:31.50   1st Qu.:11.40
##  Median :36.80   Median :13.90
##  Mean   :36.41   Mean   :14.03
##  3rd Qu.:42.00   3rd Qu.:16.60
##  Max.   :54.60   Max.   :21.60
```

In the data set we have two category variables: sp and sex, the others are quantitatives, that is the reason why in sex appears barplots and the other variables have density distribution plots and scatter plots. The two colors are associeted with the break up of sp wich has two levels.

```
library(GGally)
ggpairs(crabs, columns = 2:ncol(crabs), title = "",
  axisLabels = "show", mapping = aes(colour=sp))
```



For this model the first step is preparing the split to create two sets from the original dataset: training and testing sets, in this case we are going to use 80 percent of the sample for training the model and the other 20 percent will be used to validate de model's quality. The distribution of the observation in the two set is made over a random simple sampling applied over the index.

```
# Random sample indexes
train_index <- sample(1:nrow(crabs), 0.8 * nrow(crabs))
test_index <- setdiff(1:nrow(crabs), train_index)

# Build X_train
X_train <- crabs[train_index,]
X_test <- crabs[test_index,]
```

Correlation

We can see that the variables have a strong correlation almost all of them, that is a problem if I would try to put in the model all of these variables. The just the varaibles sex,FL,RW and CL will keep to start the model.

```
cor(X_train[,3:ncol(crabs)])
```

```
##           FL        RW        CL        CW        BD
## FL 1.0000000 0.9014291 0.9789911 0.9654567 0.9870798
## RW 0.9014291 1.0000000 0.8890831 0.8972548 0.8856202
## CL 0.9789911 0.8890831 1.0000000 0.9951528 0.9828619
## CW 0.9654567 0.8972548 0.9951528 1.0000000 0.9676702
## BD 0.9870798 0.8856202 0.9828619 0.9676702 1.0000000
```

For the model selection we are going to use the Step aic and the R function glm will be used to compute de logistic regression, specifying the option family = binomial, that means the response variable is binary. AIC penalizes increasing the number of parameters into de model, and the best option will be the model with the smallest. Not all the variables are in the inicial model because of the high correlation between them as I said before.

```
fit <- glm(sp ~ 1 + sex+FL+RW+CL+sex*FL+sex*RW, family=binomial, data=X_train)
stepAIC(fit)
```

```
## Start:  AIC=38.99
## sp ~ 1 + sex + FL + RW + CL + sex * FL + sex * RW
##
##          Df Deviance     AIC
## - sex:RW  1   25.175  37.175
## - sex:FL  1   25.546  37.546
## <none>        24.987  38.987
## - CL      1  127.226 139.226
##
## Step:  AIC=37.17
## sp ~ sex + FL + RW + CL + sex:FL
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##          Df Deviance     AIC
## - sex:FL  1   26.396  36.396
## <none>        25.175  37.175
## - RW      1   30.684  40.684
## - CL      1  127.693 137.693
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
##
## Step:  AIC=36.4
## sp ~ sex + FL + RW + CL
##
##        Df Deviance     AIC
## <none>      26.396  36.396
## - RW    1   30.703  38.703
## - sex   1   35.544  43.544
## - CL    1  166.082 174.082
## - FL    1  196.112 204.112
```

```
##
## Call:  glm(formula = sp ~ sex + FL + RW + CL, family = binomial, data = X_train)
##
```

```
## Coefficients:
## (Intercept)         sexM           FL           RW           CL
##     -24.212        6.126       14.217        2.057       -6.995
##
## Degrees of Freedom: 159 Total (i.e. Null);  155 Residual
## Null Deviance:       221.7
## Residual Deviance: 26.4  AIC: 36.4
```

Well, let see the best model in detail:

```
fit<-glm(formula = sp ~ sex + FL + CL, family = binomial, data = X_train)
summary(fit)
```

```
##
## Call:
## glm(formula = sp ~ sex + FL + CL, family = binomial, data = X_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.55888  -0.06070   0.00001   0.01202   2.06593
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -14.618      4.015  -3.641 0.000272 ***
## sexM           2.318      1.050   2.206 0.027355 *
## FL            12.967      3.291   3.940 8.14e-05 ***
## CL            -5.826      1.483  -3.928 8.58e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 221.707  on 159  degrees of freedom
## Residual deviance:  30.703  on 156  degrees of freedom
## AIC: 38.703
##
## Number of Fisher Scoring iterations: 9
```

We would like to know the probability that the event occurs when we have a male, with FL=11.1 and CL=23.8.

```
###################### Estimación de una probabilidad ###################
x <- c(1,1,11.1,23.8)
eta <- sum(x*coef(fit))
prob <- exp(eta)/(1+exp(eta))
prob
```

```
## [1] 0.0008931251
```

Now, using the function predict

```
newdata <- data.frame( sex='M',FL=11.1,RW=9.9,CL=23.8,CW=27.1,BD=9.8)
probabilities <- fit %>% predict(newdata, type = "response")
probabilities
```

```
##            1
## 0.0008931251
```

To predict the value in the testing set the threshold will be 0.4, that means that if the probability is greater than 0.4 the observation will be associated with O and in another case will be mark with B.

```
rev<-predict(fit,X_test,type = "response")
 X_test$predicted.classes<- ifelse(rev > 0.4, "O", "B")
head(X_test)
```

```
##    sp sex   FL   RW   CL   CW   BD predicted.classes
## 2   B   M  8.8  7.7 18.1 20.8  7.4                 B
## 4   B   M  9.6  7.9 20.1 23.1  8.2                 B
## 8   B   M 11.6  9.1 24.5 28.4 10.4                 B
## 14  B   M 12.8 10.2 27.2 31.8 10.9                 B
## 16  B   M 12.9 11.0 26.8 30.9 11.4                 B
## 20  B   M 13.9 11.1 29.2 33.3 12.1                 B
```

To check the model accuracy we are going to see the percent associated with values that were classified right.

```
# Model accuracy
mean(X_test$predicted.classes == X_test$sp)
```

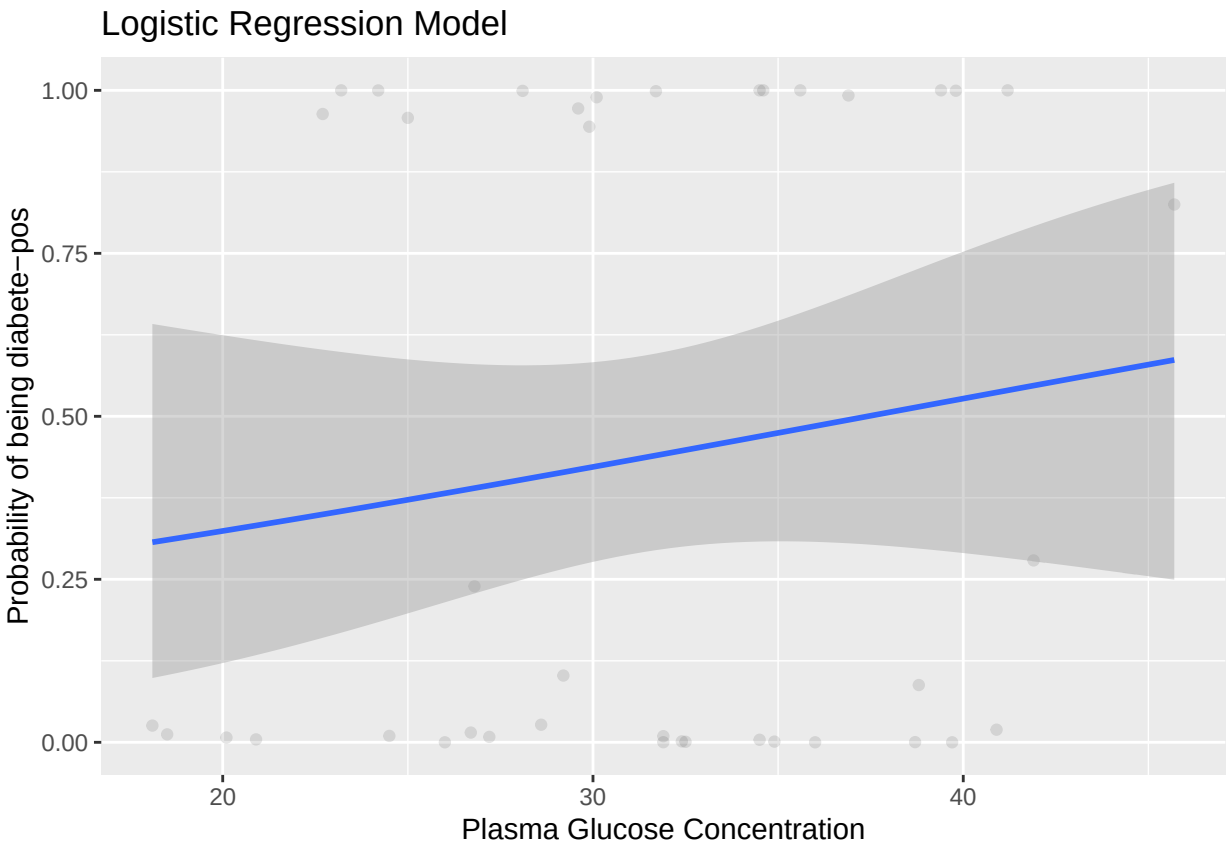
```
## [1] 0.975
```

Finally to check the well know s-shape for the logistic regreesion we are going to use the ggplot library

```
library(ggplot2)
X_test$rev<-predict(fit,X_test,type = "response")


X_test %>%
  ggplot(aes(CL, rev)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  labs(
    title = "Logistic Regression Model",
    x = "Plasma Glucose Concentration",
    y = "Probability of being diabete-pos"
    )
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial
## glm!
```



#Pseudo R2 McFadden measure is 0.88 that is a really good value because this metric ranges from 0 to just under 1, with values closer to zero indicating that the model has no predictive power.

```
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 3.5.3
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
pR2(fit)
```

```
##          llh      llhNull           G2     McFadden         r2ML
##  -15.3514333 -110.8535437  191.0042207    0.8615161    0.6969254
##         r2CU
##    0.9294276
```

Finally, we exmaine the ROC curve which shows the trade off between the rate at which you can correctly predict something with the rate of incorrectly predicting something.

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.5.3
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.5.3
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
# Compute AUC for predicting Class with the model
prob <- predict(fit, newdata=X_test, type="response")
pred <- prediction(prob, X_test$sp)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf)
```