Logistic Regression

Andrea Huerfano

September 1, 2019

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(cangrejos), 0.8 * nrow(cangrejos))
test_index <- setdiff(1:nrow(cangrejos), train_index)

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

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.

```
fit <- glm(y ~ 1 + w + c + s + c*s + c*w + s*w, family=binomial, data=X_train)
stepAIC(fit)
```

```
## Start:  AIC=159.97
## y ~ 1 + w + c + s + c * s + c * w + s * w
##
##        Df Deviance    AIC
## - c:s   1   146.48 158.48
## - w:s   1   146.84 158.84
## - w:c   1   147.95 159.95
## <none>      145.97 159.97
##
## Step:  AIC=158.48
## y ~ w + c + s + w:c + w:s
##
##        Df Deviance    AIC
## - w:s   1   147.25 157.25
## - w:c   1   148.40 158.40
## <none>      146.48 158.48
##
## Step:  AIC=157.25
## y ~ w + c + s + w:c
##
##        Df Deviance    AIC
## - w:c   1   148.43 156.43
## - s     1   148.52 156.52
## <none>      147.25 157.25
##
## Step:  AIC=156.43
## y ~ w + c + s
```

```
##
##        Df Deviance    AIC
## - s     1   149.48 155.48
## <none>      148.43 156.43
## - c     1   152.35 158.35
## - w     1   169.16 175.16
##
## Step:  AIC=155.48
## y ~ w + c
##
##        Df Deviance    AIC
## <none>      149.48 155.48
## - c     1   152.45 156.45
## - w     1   170.94 174.94
```

```
##
## Call:  glm(formula = y ~ w + c, family = binomial, data = X_train)
##
## Coefficients:
## (Intercept)            w           c1
##    -11.9728       0.4675       0.6923
##
## Degrees of Freedom: 137 Total (i.e. Null);  135 Residual
## Null Deviance:       179.5
## Residual Deviance: 149.5     AIC: 155.5
```

After running the aic step the best model is described below

```
########### Estimación del modelo ########
fit2 <- glm(y ~ 1 + w, family=binomial, data=cangrejos)
summary(fit2)
```

```
##
## Call:
## glm(formula = y ~ 1 + w, family = binomial, data = cangrejos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0281  -1.0458   0.5480   0.9066   1.6942
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.3508     2.6287  -4.698 2.62e-06 ***
## w             0.4972     0.1017   4.887 1.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 194.45  on 171  degrees of freedom
## AIC: 198.45
##
## Number of Fisher Scoring iterations: 4
```

The second one model has small AIC however not all the coefficient are significat at 5% level

```
###################### Prueba de hipótesis ##################
fit3 <- glm(y ~ 1 + w + c, family=binomial, data=cangrejos)
summary(fit3)
```

```
##
## Call:
## glm(formula = y ~ 1 + w + c, family = binomial, data = cangrejos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1080  -0.9708   0.5346   0.8958   1.8188
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.9501     2.6643  -4.485 7.28e-06 ***
## w             0.4670     0.1037   4.506 6.61e-06 ***
## c1            0.6531     0.3571   1.829   0.0675 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 191.12  on 170  degrees of freedom
## AIC: 197.12
##
## Number of Fisher Scoring iterations: 4
```

```
anova(fit2, fit3 , test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ 1 + w
## Model 2: y ~ 1 + w + c
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       171     194.45
## 2       170     191.12  1   3.3344  0.06785 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When x=26, we would like to know the probability that the event occurs.

```
###################### Estimación de una probabilidad ###################
x <- c(1,26)
eta <- sum(x*coef(fit2))
prob <- exp(eta)/(1+exp(eta))
prob
```

```
## [1] 0.6404177
```

Now, using the function predict

```
newdata <- data.frame(w = c(20,  26))
probabilities <- fit2 %>% predict(newdata, type = "response")
probabilities
```

```
##         1         2
```

```
## 0.08270068 0.64041770
```

Conidence interval for the coeficent

```
alpha <- 0.05
li <- coef(fit2)[-1] - qnorm(1-alpha/2)*diag(vcov(fit2))[-1]
ls <- coef(fit2)[-1] + qnorm(1-alpha/2)*diag(vcov(fit2))[-1]
li <- exp(li)
ls <- exp(ls)
c(li,ls)
```

```
##        w        w
## 1.611144 1.677855
```

Predicting the value for in the testing set, for this case the threshold is 0.7, that means that if the probability is greater than 0.7 the observation will be associated with 1 and in another case will be mark with 0.

```
rev<-predict(fit2,X_test,type = "response")
 X_test$predicted.classes<- ifelse(rev > 0.7, "1", "0")
X_test
```

```
##     c s    w y predicted.classes
## 2   0 0 26.0 1                 0
## 3   0 0 25.6 0                 0
## 11  1 1 26.1 1                 0
## 16  0 0 24.5 1                 0
## 18  1 0 26.2 1                 0
## 20  1 0 25.4 1                 0
## 33  1 1 24.9 1                 0
## 38  1 0 30.0 1                 1
## 40  1 0 23.9 1                 0
## 41  1 0 26.0 1                 0
## 46  1 0 23.8 0                 0
## 56  1 0 28.2 1                 1
## 69  0 0 24.5 1                 0
## 77  1 1 24.5 1                 0
## 79  1 1 25.0 1                 0
## 86  0 0 24.1 0                 0
## 89  0 1 24.7 0                 0
## 90  1 0 25.8 0                 0
## 97  1 0 27.9 1                 1
## 104 1 0 26.2 0                 0
## 107 1 0 25.1 1                 0
## 110 1 1 26.8 0                 1
## 114 1 0 29.0 1                 1
## 121 1 0 24.9 0                 0
## 123 1 1 24.3 0                 0
## 127 0 1 29.8 1                 1
## 142 1 0 28.5 1                 1
## 145 0 0 27.1 0                 1
## 147 1 1 26.5 0                 0
## 148 0 0 23.0 0                 0
## 149 0 1 26.0 1                 0
## 155 1 0 28.2 1                 1
## 156 1 1 25.2 1                 0
## 166 1 0 25.8 0                 0
## 171 1 0 26.5 1                 0
```

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

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

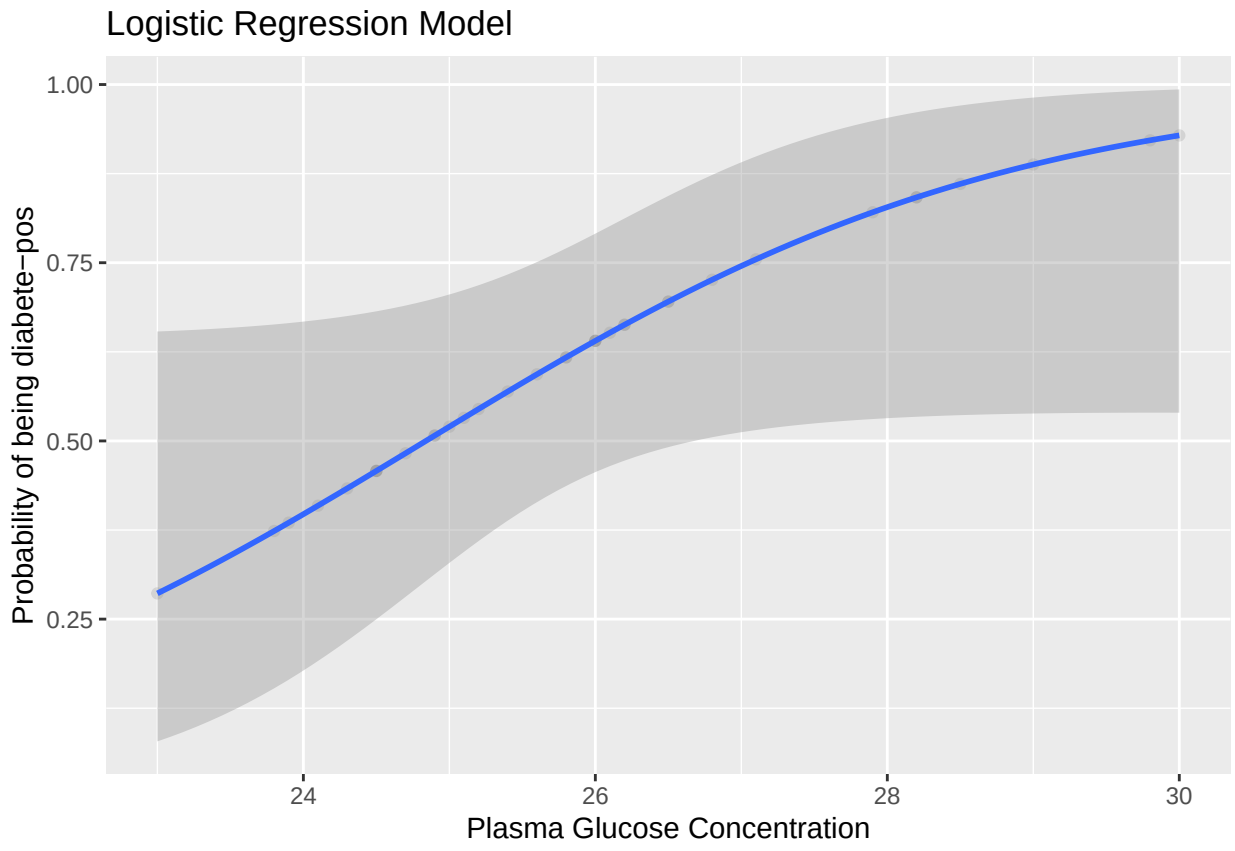
```
## [1] 0.5142857
```

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(fit2,X_test,type = "response")


X_test %>%
  ggplot(aes(w, 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!
```




```
table(X_test$y,X_test$predicted.classes)
```

```
##
```

```
##      0  1
##   0 11  2
##   1 15  7
```

References

- R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
- Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2019). dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr

http://www.sthda.com/english/articles/36-classification-methods-essentials/151-logistic-regression-essentials-in-r/