Exercise 7

2025902616

2025-05-22

# Q1. Logistic regression on binary outcome

library(Ecdat)

## Loading required package: Ecfun

##   
## Attaching package: 'Ecfun'

## The following object is masked from 'package:base':  
##   
## sign

##   
## Attaching package: 'Ecdat'

## The following object is masked from 'package:datasets':  
##   
## Orange

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(e1071)  
library(lattice)  
library(AER)

## Loading required package: car  
## Loading required package: carData  
##   
## Attaching package: 'carData'  
##   
## The following object is masked from 'package:Ecdat':  
##   
## Mroz  
##   
##   
## Attaching package: 'car'  
##   
## The following object is masked from 'package:dplyr':  
##   
## recode  
##   
## The following object is masked from 'package:purrr':  
##   
## some  
##   
## Loading required package: lmtest  
## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric  
##   
## Loading required package: sandwich  
## Loading required package: survival

library(neuralnet)

##   
## Attaching package: 'neuralnet'  
##   
## The following object is masked from 'package:dplyr':  
##   
## compute

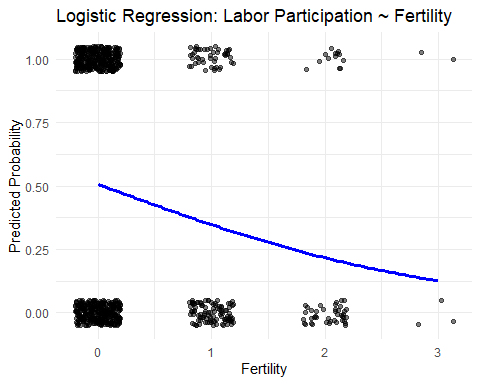
library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select  
##   
## The following object is masked from 'package:Ecdat':  
##   
## SP500

data("SwissLabor")  
str(SwissLabor)

## 'data.frame': 872 obs. of 7 variables:  
## $ participation: Factor w/ 2 levels "no","yes": 1 2 1 1 1 2 1 2 1 1 ...  
## $ income : num 10.8 10.5 11 11.1 11.1 ...  
## $ age : num 3 4.5 4.6 3.1 4.4 4.2 5.1 3.2 3.9 4.3 ...  
## $ education : num 8 8 9 11 12 12 8 8 12 11 ...  
## $ youngkids : num 1 0 0 2 0 0 0 0 0 0 ...  
## $ oldkids : num 1 1 0 0 2 1 0 2 0 2 ...  
## $ foreign : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

#Fit logistic regression model  
  
model1 <- glm(participation ~ youngkids + income, data = SwissLabor, family = binomial)  
  
newdata1 <- data.frame(youngkids = seq(min(SwissLabor$youngkids), max(SwissLabor$youngkids), length.out = 100), income = mean(SwissLabor$income))  
  
newdata1$predicted\_prob <- predict(model1, newdata = newdata1, type = "response")  
  
ggplot(SwissLabor, aes(x = youngkids, y = as.numeric(participation) -1)) +  
 geom\_jitter(height = 0.05, width = 0.2, alpha = 0.5) +  
 geom\_line(data = newdata1, aes(x = youngkids, y = predicted\_prob), color = "blue", linewidth = 1.2) +  
 labs(title = "Logistic Regression: Labor Participation ~ Fertility", x = "Fertility", y = "Predicted Probability") +  
 theme\_minimal()



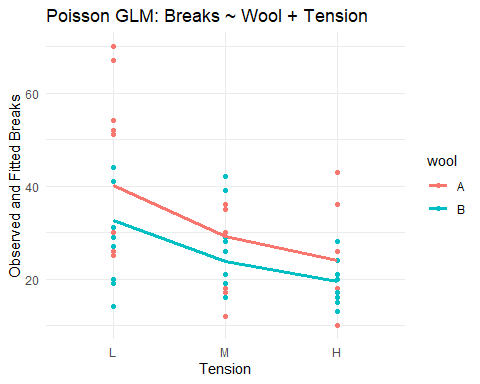
# Q2. Poisson regression for count data

data("warpbreaks")  
  
model2 <- glm(breaks ~ wool + tension, data = warpbreaks, family = poisson)  
  
#Predict fitted values so use $fit  
warpbreaks$fit <- predict(model2, type = "response")  
warpbreaks$fit

## [1] 40.12354 40.12354 40.12354 40.12354 40.12354 40.12354 40.12354 40.12354  
## [9] 40.12354 29.09722 29.09722 29.09722 29.09722 29.09722 29.09722 29.09722  
## [17] 29.09722 29.09722 23.89035 23.89035 23.89035 23.89035 23.89035 23.89035  
## [25] 23.89035 23.89035 23.89035 32.65424 32.65424 32.65424 32.65424 32.65424  
## [33] 32.65424 32.65424 32.65424 32.65424 23.68056 23.68056 23.68056 23.68056  
## [41] 23.68056 23.68056 23.68056 23.68056 23.68056 19.44298 19.44298 19.44298  
## [49] 19.44298 19.44298 19.44298 19.44298 19.44298 19.44298

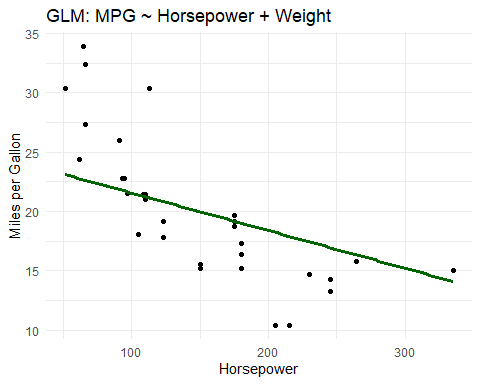
ggplot(warpbreaks, aes( x= tension, y = breaks, color = wool )) +  
 geom\_point() +  
 geom\_line(aes(y = fit, group = wool), size = 1.2) +  
 labs(title = "Poisson GLM: Breaks ~ Wool + Tension", x = "Tension", y = "Observed and Fitted Breaks") +  
 theme\_minimal()

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



# Q3. Ordinary Linear Regression

data("mtcars")  
  
model3 <- glm(mpg ~ hp + wt, data = mtcars)  
  
#predict over horsepower range at fixed weight  
newdata3 <- data.frame(hp = seq(min(mtcars$hp), max(mtcars$hp), length.out = 100),  
 wt = mean(mtcars$wt))  
newdata3$predicted <- predict(model3, newdata = newdata3)  
  
ggplot(mtcars, aes(x = hp, y = mpg)) +  
 geom\_point() +  
 geom\_line(data = newdata3, aes(x = hp, y = predicted), color = "darkgreen", size = 1.2) +  
 labs(title = "GLM: MPG ~ Horsepower + Weight", x = "Horsepower", y = "Miles per Gallon") +  
 theme\_minimal()

 predict function(model, newdata)

# Q4. Applying SVM on Wages data to model the income status based on the education, experience, marital status and sex

data(Wages)  
str(Wages)

## 'data.frame': 4165 obs. of 12 variables:  
## $ exp : int 3 4 5 6 7 8 9 30 31 32 ...  
## $ wks : int 32 43 40 39 42 35 32 34 27 33 ...  
## $ bluecol: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 2 2 ...  
## $ ind : int 0 0 0 0 1 1 1 0 0 1 ...  
## $ south : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 1 1 1 ...  
## $ smsa : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ married: Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...  
## $ sex : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 2 2 2 ...  
## $ union : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 2 ...  
## $ ed : int 9 9 9 9 9 9 9 11 11 11 ...  
## $ black : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ lwage : num 5.56 5.72 6 6 6.06 ...

any(is.na(Wages))

## [1] FALSE

#Create binary target: high wage  
mean\_wage <- mean(Wages$lwage, na.rm = TRUE)  
Wages\_clean <- na.omit(Wages)  
Wages\_clean$high\_wage <- as.factor(Wages\_clean$lwage > mean\_wage)  
  
#Split into training and test sets  
set.seed(123)  
n <- nrow(Wages\_clean)  
train\_index <- sample(1:n, size = 0.7 \* n)  
train\_data <- Wages\_clean[train\_index, ]  
test\_data <- Wages\_clean[ - train\_index, ]  
  
#the training dataset selects only the rows whose numbers are in train\_index which is 70%  
#the test dataset selects all rows excepot those in training set, which is 30%  
  
model2 <- svm(high\_wage ~ exp + ed + married + sex, data = train\_data, kernel = "radial", probability = TRUE)  
#we are trying to predict high\_wage by using the predictor variables of exp, ed, married, gender. It uses 70% of the data  
  
#Train accuracy  
pred2 <- predict(model2, test\_data)  
acc2 <- mean(pred2 == test\_data$high\_wage)  
cat("Training Accuracy(Wages):", round(acc2, 4), "\n")

## Training Accuracy(Wages): 0.7152

#acc2 calculates the accuracy by comparing predictions to actual values  
  
#Dummy input  
dummy2 <- data.frame(exp = 10, ed = 14, married = factor("yes", levels = levels(Wages\_clean$married)), sex = factor("male", levels = levels(Wages\_clean$sex)))  
pred\_dummy2 <- predict(model2, dummy2, probability = TRUE)  
cat("Dummy Prediction (Wages):", as.character(pred\_dummy2), "\n")

## Dummy Prediction (Wages): FALSE

# the factor function ensures the categorical variables have the same factor levels as the original training data. Then pred\_dummy2 uses the trained SVM model

# Q5. Housing Price Categorization using Boston data