FORE224/STAT202 Assignment 10

Logistic regression

Due at 23.59 (1 minute to midnight!) on Sunday 24 October 2021

Note that the deadline is different to our usual assignment deadlines!

As usual, create a new project in its own folder for this assignment.

[1] Load the `tidyverse` library.

The dataset for this assignment concerns default on credit card debts. Download the file defaults.csv from Learn and save it in your project folder. The file contains data for 666 observations. Three variables are recorded:

- the categorical variable default, which is "Yes" if the case defaulted on their credit card debt and "No" otherwise.
- the categorical variable student, which is "Yes" if the case is a student and "No" otherwise.
- the continuous variable balance, which is the case's credit card balance in $.

You will use this data to create a logistic regression model for the probability that a case with a given credit card balance defaults on their debt.

Read the dataset from defaults.csv. Use `mutate` to make the variables default and student in the data into factors (the automatic factor levels that R will use work here so they do not have to be set explicitly).

Use `set.seed` to set the random seed to your student ID number and make a subset called my_defaults of 600 cases from this defaults data using `sample_n` as in previous labs. Use this subset of the data for the rest of the assignment.

[2] Can we predict whether or not a credit card holder will default using their credit card balance, and does this depend on whether or not they are a student?

Use the code below to make jittered scatter plot of the data (or if you prefer you can make your own plotting code).

```
my_defaults %>% ggplot(aes(x = balance, y = default, colour = student)) +
  geom_jitter(alpha = 0.6, size = 2, height = 0.1) +
  labs(x = "Balance", y = "Default") +
  theme_bw() +
  theme(legend.position = "bottom")
```

[3] Use the R function `glm` to create a logistic regression model for response default with predictors balance and student including the interaction between balance and student. Remember that with `glm` you need to specify the model family. The code below will create the required logistic regression model including the balance:student interaction.

```
m1 <- glm(default ~ balance + student + balance:student,
  family = binomial(logit),
  data = my_defaults
)
```

Show the code used to create the model in your assignment.

Show the summary output from the model (use R's `summary` function on the model).

[4] Create a logistic regression model for response default with predictors balance and student but NOT including interaction between balance and student. Show the code used to create the model and the model summary output in your assignment.

[5] Use R's `anova` function to test the full model m1 against the reduced model m2. Remember that for the 'analysis of deviance' test for nested logistic regression models we specify `test = "Chisq"`. The following code will do the required nested model test.

```
anova(m2, m1, test = "Chisq")
```

Is the interaction between balance and student statistically significant? Briefly (20-30 words) justify your answer including the value(s) from the `anova' output that you have used.

[6] Use model m2 (the model without the balance:student interaction) to answer the following questions. For each question show how you have used the model summary output in your calculations & interpretations. Each answer should be 30-50 words.

- m2 gives an intercept for the baseline group (non-students, student = "No"). According to the model what is the probability that a non-student with a balance of 0 defaults?
- m2 also gives an adjustment this intercept for students (student = "Yes"): According to the model what is the probability that a student with a balance of 0 defaults?
- m2 gives a coefficient for the slope in relation to balance. This is the same for both students and non-students in the model without a balance:student interaction. Interpret this coefficient in terms of the effect of an extra $ of credit card debt on the odds of default.

[7] Make a tibble of new data to use for prediction. The tibble should contain data with a balance of $1500 for both a student and a non-student.

Use R's `predict` function to predict the probabilities using model m2 for the cases represented by the `new_data` tibble. Remember that you need to specify `type = "response"` in `predict` to get the predicted probabilities.

Briefly (about 50-60 words) comment on these probabilities in the context of the model (how do the values compare and what does this indicate about the situations being modeled).

[8] Suppose that that you used a 'boundary probability' to classify cases in the sample as predicted default or predicted non-default and found that, in the sample, the number of false positives that boundary probability gave was 42 and the number of false negatives was 33. What would the Apparent Error Ratio be?

Optional: If you want to try to create a plot of the sample data overlaid with the fitted probability curves for model m2 you can use this code. Note that the code assumes that you have variables for the model coefficients. Two have been created for you but you need to add two more!

```
# save coefficient values
b0_N <- coef(m2)[[1]] # intercept, not student
b1_N <- coef(m2)[[2]] # logit slope, not student
# you need to create variables for b0_Y and b1_Y
# create b0_Y the intercept for students
# create b1_Y the logit slope for students

# plot the data and curves
# balances data for plotting
plot_balances <- tibble(balance = seq(0, 3000, 500))
ggplot(my_defaults, aes(
  x = balance,
```

```
  y = ifelse(default == "No", -0.1, 1.1), # plot observed points below 0 and above
1
  colour = student
)) +
  geom_jitter(alpha = 0.8, height = 0.1) +
  # plot curve for not students (assumes you have variables for b0_N, b1_N)
  geom_function(
    data = plot_balances, aes(x = balance),
    inherit.aes = FALSE, # does not use parent ggplot's aesthetics
    fun = function(x) plogis(b0_N + b1_N * x), # plogis does expit conversion
    size = 1, colour = "#F8766D" # emulate ggplot default colours :)
  ) +
  geom_function(
    # plot curve for students (assumes you have variables for b0_Y, b1_Y)
    data = plot_balances, aes(x = balance),
    inherit.aes = FALSE,
    fun = function(x) plogis(b0_Y + b1_Y * x),
    size = 1, colour = "#00BFC4" # emulate ggplot default colours :)
  ) +
  # tidy up R's default y-axis
  scale_y_continuous(expand = c(0, 0), limits = c(-0.2, 1.2), breaks = c(0, 1)) +
  labs(x = "balance", y = "y = 1 if default, 0 otherwise\n and modeled
probabilities") +
  theme_bw() +
  theme(legend.position = "bottom")
```

Include all the answers and code used to obtain them in a Word document and submit it through Learn. Remember to use your own words when answering the questions.