MATHS 7107 Data Taming
Assignment 3

Trimester 3 2024

1 Background

Source: Bing Copilot.[1]

The software company was happy with your work on their previous problem about the amount of debugging time required for their software programs. Their new software project is almost finished, and (as always) things are more complicated than expected. The length of the new project is now expected to be about 100,000 lines of code, and they are looking at putting together a debugging team to work on it before it is released to the customers. Since this is such a large project, they'd like some estimate of whether the program will have any fatal errors (a very serious bug) before they release it.

The chief of product development has put together the debugging team for this project, with Bob as the team leader. Bob is quite new to the company — he only completed his computer science degree last year and was immediately employed at the company, just in time to collect his yearly bonus. (This was somewhat controversial, but since Bob is the chief's son, nothing was done about it.) In this short time he has become very popular, as he has managed to attend every work party and function, and also become the office's champion foosball player. He has promised to take his team along with him to all of these events, in the interests of "team building".

The team members that Bob will be leading all have at least a moderate amount of experience, and they all met their KPIs last year, so they all received their pay bonuses. Yet, the CEO is a little concerned about Bob being in charge of such an important project, so they'd like you to do some analysis on the probability of Bob's team missing any fatal errors. They have provided you with 4 data sets, one from each of the company's worldwide divisions. Using this data, try to help the CEO determine if this new team will be any good. As with the last report, the CIO uses `R` and `R Markdown`, and even completed Data Taming in the past. So make sure you only use commands from the course, so that the CIO can easily see what analysis you've done. In your `R Markdown` code chunks: make sure that you **do not** set `echo = FALSE` so that they can see what `R` code you used to generate your output. But of course, they don't want to see irrelevant warnings or messages.

But remember that your report is for the CEO, who is not really a technical person, and who certainly doesn't know `R`. So make sure you include descriptions allowing the average person to understand what you are doing and what the output means.

[1]Prompt: "A realistic drawing of a computer suffering from a "fatal error" with a computer tower sitting on the desk, with smoke and sparks coming out of the computer tower, and the screen saying "error"."

1.1 Number of digits

When writing your own text, or **USING** the output from R:

- For integer results, report the whole integer.
- For non-integers with absolute value > 1: use 2 decimal places
- For non-integers with absolute value < 1: use 3 significant figures.

  For example:

  - $135.5681 \approx 135.57$
  - $-0.0004586 \approx -0.000459$

Exceptions:

- If you're just **PRINTING** the output from R, then just keep the output as it is.
  - But if you have R do the rounding for you then you need to conform to these two conventions listed above.
- If your data has fewer digits of precision than specified above (eg. because of the way it was stored in the original data, or because of the way it was calculated) then only report that level of precision.

2 The data

The company has four datasets labelled `programs_0.csv`, `programs_1.csv`, `programs_2.csv` and `programs_3.csv` (one from each of the divisions). Each dataset contains 6 columns:

- `FError`: if the program was found to have a fatal error.
- `LOC`: the number of lines of code in the program.
- `XP`: the experience level of the debugging team leader.
- `foosball`: if the office with the debugging team has a foosball table this is `yes`, otherwise `no`.
- `Parties`: the number of office parties that the program's debugging team attended during the debugging project, `none`, `some` or `many`.
- `Bonus`: `yes` or `no` indicating whether or not the team members received their bonus in the previous year.

Each dataset has data on 36,000 software programs. Luckily, the data itself has already been cleaned and so there should not be any missing or erroneous rows in the data. However, the data cleaner is a typical Gen-Y who has trouble spelling, and a keen interest in deep cuts of Coolio, so be on the lookout for some required data taming.

IMPORTANT!

If you remove any data, then make sure you only remove data that you MUST remove. Do not just delete data because it is inconvenient. You must have specific instructions from the client, or it must be an impossible value, before you remove any data from your analysis. Even then, you need to describe why it was removed.

3 Your job

To help the company, we will analyse the data of fatal errors, and then make a prediction about how Bob's team will go.

Note

Make sure you write text to explain what you are doing at each point and why you are doing it. You need to justify all the things you do or claim. Also describe the results.

1. Load the correct dataset and save it as a tibble. Output the first 10 lines of the dataset and the dimensions of the data set.
2. Using dot points, identify what types of variables we now have in our data set, i.e., "Quantitative Discrete", "Quantitative Continuous", "Categorical Nominal", "Categorical Ordinal". (Don't just describe what data type they are in the data set — you need to think about the type of variable in the context of the meaning of the data.) Make sure you provide some justification for your choice of variable types.
   - Don't just provide vague statements, but be very concrete about describing this particular set of data.
3. Now it's time to tame our data. But since we are going to fit a logistic regression model, we need to modify our requirements a little bit.
   - Fix and tame all column names.
   - Convert the fatal error status to a `<fct>` data type, with `yes` for `1` and `no` for `0`.
   - Treat the number of lines of code as a **quantitative continuous** variable. This is because we want to fit a series of lines, which assumes that the predictor is continuous.
   - If you have identified any Categorical Ordinal variables, store them as a `<fct>`.
   - Make the remaining variables conform to the Tame Data conventions in Module 2 (page 3).

   Output the first 10 rows of your data and the dimensions of the data set.
4. Setting the correct seed, split your data into a training set (with 25,000 rows) and a testing set, with the remaining rows. Output the first 10 lines of each dataset and the dimensions of each data set.
5. Fit a logistic regression model to your training data, with the fatal error status as the response and all other variables as the predictors. (Just use them individually, don't include any interaction terms.) Output the summary of the model.
6. Since we are using general linear models, the model summary in Question 5 describes linear geometric objects, where the dimension of the geometric object is determined by the number of quantitative continuous predictors. We have only a single quantitative continuous predictor so our model describes a set of lines. How many lines are described by the model in Question 5? Make sure you give some justification for your answer.
   - *(Hint: see the Week 8 seminar and pages 12 – 16 of Module 7. The model summary output should help.)*
7. Now it is time to get serious with our data. There may be some interactions between the variables in the data set, so fit a new model to your training set using all the individual variables and all the second-order interaction terms. Use `Anova()` to find the p-values for each of the variables. Identify all interaction terms that meet the 90% significance level.
   - *(Hint: if you have three predictors x_1, x_2, x_3, then the second order interaction terms are x_1x_2, x_1x_3, x_2x_3. There is an easy way and a hard way to do this — see the Reminder sheet for the easy way.)*
8. We'll now apply **backwards stepwise regression**. As we learned in Module 7, best practice is to only remove terms one-by-one starting with the least significant. However, the CEO has said that we are not to consider the foosball table (since the staff are likely to strike if there is any suggestion that we might recommend removing it).
   (a) So ignore anything to do with the foosball table, and fit a new model with all the remaining individual variables and interactions. Show the `Anova()` output.
   (b) Then ONLY looking at the interaction terms, continue with step-by-step **backwards stepwise regression** to find a model where all interaction terms meet the 90% significance level. At each step, identify the interaction term that you will remove, and why you will choose that one. Then show the resulting `Anova()` after you fit each model.

- The **"principle of marginality"** tells us that a variable shouldn't appear in an interaction term if we don't have the variable appear by itself.

(c) Finally, now focus on the individual terms and finish the **backwards stepwise regression** so that all terms (individual terms and interaction terms) meet the 90% significance level. At each step, identify the variable that you will remove, and why you will choose that one. Then show the resulting `Anova()` after you fit each model.

- *(Note: since everybody's training set is different, you may find that there are no non-significant individual terms at this point. In which case, just explain that in text and show the* `Anova()` *again.)*

9. (a) Which interaction terms are significant (at the 90% level) in your final model?

   (b) Thinking about the context of the data, provide some reasonable hypotheses for why those interaction terms might represent real effects (and are not just statistical noise).

10. So we have now fit a logistic regression model for the **log-odds**, which has the general form:

$$\log\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = \hat{b}_i$$

where $\hat{b}_i$ is an estimated function of the predictors. Write down the general form of $\hat{b}_i$ for your final model in Question 8. Keep the coefficients as pronumerals for now, so it should look like:

$$\hat{b}_i = \hat{\beta}_0 + \ldots$$

Be sure to define all variables in your equation.

11. Looking at Question 10, the geometric situation is slightly more complicated now than in Question 6, although our model should still produce a set of lines.

   (a) How many lines does your final model describe? Make sure you provide some justification for your answer.

   (b) Are the lines all parallel? If not, explain why not.

12. Now output the summary of your final model showing the estimated coefficients, and use that to write $\hat{b}_i$ with all the estimated coefficients replacing the $\hat{\beta}_j$ pronumerals.

13. What is our estimate for the **log-odds** of a program still containing a fatal error if:

   (a) the office has a foosball table, and the team is lead by a highly experienced staff member, who doesn't allow his team to attend any work functions, but they all received their bonuses last year?

   (b) the office has a foosball table, the team leader has a moderate amount of experience, who lets his staff attend some functions, but the staff did not get their bonus last year?

14. Now apply your final model to the testing data. Produce a new tibble containing the true classes, the predicted classes and the prediction probabilities. Output the first 10 lines of this tibble and the dimensions of the data set.

15. Now we need to evaluate our model.

   (a) Find the confusion matrix and the accuracy of the model.

   (b) If "having a fatal bug" is classified as a success, find the sensitivity and specificity of our model. (*Hint: make sure you calculate the values yourself, as R may not choose the right level.)*

   (c) Plot the ROC curve. You might want to add the following code to your `autoplot()`

```
+ geom_vline(xintercept=...)  + geom_hline(yintercept=...)
```

   to help you identify which is the correct ROC curve.

   (d) What is the AUC of this ROC curve?

16. Finally, let's try to answer the CEO's question. Based on your model, do you predict that Bob's team will result in a fatal error in the final program? Write some text to interpret your results for the CEO, and make sure you give the probabilities of your predicted class.

4 Submission

You must submit your assignment via MyUni. Do not email it to the teaching staff. Detailed instructions are on the assignment submission page in MyUni. Make sure that all your output is relevant to the questions being asked.

5 Deliverable Specifications (DS)

Before you submit your assignment, make sure you have met all the criteria in the **Deliverable Specifications (DS)**. The client will not be happy if you do not deliver your results in the format that they've asked for.