Week 4 - Imputation

DATA201/422

2024-08-05

Overview

In our lectures we learned about imputation. While this can be a broad sub-field of statistics in its own right, it is worth raising in a Data Engineering context. It is common that as part of a data processing pipeline data needs to be cleaned and sometimes imputed. We will go over some basic imputation techniques and compare their accuracy.

Maintain good practice following everything we have learned to date, including the correct styles, projects, repositories.

Use Case

A client representing the Green Party collected data about the Cannabis referendum in 2020. They wish to know from their data if it is worth pursuing a citizen initiated referendum in the next election cycle. However, they do not wish to undertake such a massive administrative task if the support is not there. Your task is to address the following:

- What proportion of people in the sample supported legalisation ?
- Who in the sample supported legalisation ?

Deliverables

You must produce the deliverable as a reproducible report as a PDF using R Markdown or Quarto. It must have clean formatting and explanation in-text. Do not include the code in your compiled report. You must have the following elements:

- Table on missingness
- Visualisation on demographics
- Inline reporting of proportions
- Tables of each logistic regression
- A clear conclusion
- Extra for experts for those game enough

Getting started

The data is available on Learn under Week 4. Put the CSV file into the appropriate directory so it can be loaded into your R environment. It is a single CSV file containing survey responses with Age, Gender, and whether someone voted yes (1) or no (0) in the referendum. However, there was a considerable amount of missingness in the data. It is believed this is because some people, particularly younger people, were reluctant to give their voting preference.

Load the data into your environment

There are a lot of ways to load CSVs in R, use the appropriate Tidyverse function. Explore the data so you are familiar with all the variables and any issues that may be lurking in the data.

Test for missingness

Using the following function, which can be a little complicated, check for missingness in each variable. Present the findings in a table, easy to read table and include any necessary commentary. Use your second lab as a guide on our expectations for tables.

```
referendum_raw %>%
  summarise(across(everything(), ~ sum(is.na(.))))
```

Show the demographics

Produce a visualisation that clearly shows the distribution of age across genders. You may wish to use the `facet_wrap()` function if you want to try something new. Once again we have strong expectations on visualisations. Include any interpretation that is necessary to communicate to the client.

Conduct required analyses

There are two main tests you must conduct and articulate to the client. The overall proportion and a regression determining what kind of person was more likely to vote 'yes' in the referendum. However, we want to be careful that missingness did not influence our findings so we will conduct our analyses using complete cases and multiple imputation.

Complete cases

Calculate the proportion of 'yes' voters inline, not in a code chunk. So you can report a proportion such as 0.6, generated using inline code. You may have to google this as we haven't covered it previously. There are a number of ways to calculate the proportion, those who want to be fancy should use the `mean()` function (all methods gain the same marks).

Now when determining *who* is more likely to vote yes, we will use regression. The outcome is a binomial so we will use a type of regression called logistic regression. This is not a course on regression and some have not covered it so here is the necessary code. You can show how tidy results can be presented but please pipe the result into a clean table using `kableExtra`. Interpret the results for the client as best you can.

```
# Conduct a logistic regression model and assign result to ref_model
ref_model <- glm(referendum ~ age + gender, data = referendum_raw, family = binomial)

# Show tidied results using the broom package
# It would look much nicer with kableExtra...
tidy(ref_model)
```

Imputed cases

Now the fun starts and we will use the `mice` package to conduct multiple imputation. You may need to search around to get an idea on how to approach this. The complexity is that you have imputed multiple datasets and need to conduct your analysis on each one, then pool the results. Example code is included below. Once again, kableExtra would provide a nice table for this and the proportion would look much better in-text with interpretation.

```
# Run multiple imputation
referendum_imputed <- mice(referendum_raw)

# Calculate the proportion of yes voters
mean(complete(referendum_imputed, "long")$referendum)

# Fit the logistic regression model on each of the imputed datasets,
# pool them, and show summary stats
```

```r
with(referendum_imputed, glm(referendum ~ age + gender, family = binomial)) %>%
  pool() %>%
  summary()
```

Conclusion

Add a short conclusion on the findings, summarising key points for the client. It should be one paragraph maximum.

Extra for experts

This week the extra for experts is broken into two parts, this part can be graded but is tough. The tutors will prioritise students who have not completed up to this point in tutorials.

Interpret data quality

Include some interpretation on whether there is any bias worth considering. Discuss whether there was an impact on the data due to missingness.

Compare results

Compare the regression analyses for the complete cases and imputed data. Present the finding in a single visualisation that can highlight any difference between the two datasets. Include an estimate on the error around the estimates in each analysis. This visualisation is meant to challenge people and marking will be strict. A hint is that you may need to remove some genders if the scales vary substantially between effects.