STATS 769
Data Science Workflow

Paul Murrell

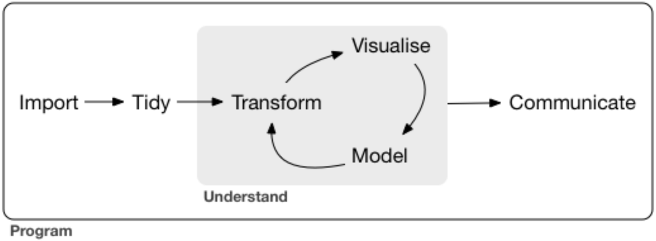
The University of Auckland

July 22, 2019

Overview

- This section of the course provides an overview of the activities involved in Data Science.
- Our primary computing environment will be R.

Data Science Workflow



Source: `https://r4ds.had.co.nz/` CC-BY-NC-ND

Data Science Workflow

- This course will focus on computational concepts and tools that impact on each of the stages in the Data Science Workflow.
- Some topics (Code Efficiency and Parallel Code) have an impact at all stages (because we are always writing code).
- This course will not focus on Visualisation or Modelling or Communication because those topics are covered in other Statistics courses.
- This topic will provide some basics on Visualisation and Modelling and Communication so that everyone has at least one tool to perform those stages of the Data Science Workflow.

Data Import

- The important thing here is to be able to deal with a multitude of data formats.
- We will review some basics in the next topic (Data Tech Review).
- We will look at some more advanced import scenarios in later topics (Data Formats and Web Scraping).

Tidying and Transforming Data

- The important thing here is to be comfortable with data structures in R and converting between different data structures.
- We will review some basics in the next topic (Data Tech Review).
- We will look at some tools outside of R in the Linux topics.
- We will consider some issues with Large Data in the second half of the course.

Visualisation

- Base graphics in R.
- `plot()`, `barplot()`, `boxplot()`, `plot(density())`
- `lines()`, `abline()`
- `par(mfrow)`

Modelling

- We will provide a basic common framework in this topic.
- We may introduce some additional analysis techniques in later sections.
- We will consider issues with Large Data, Code Efficiency, and Parallel Computing in the second half of the course.

Modelling

- We will consider a simplified modelling framework where we are only interested in prediction (no inference).
- We have an outcome variable, Y, and one or more predictor variables, X_1, X_2, etc.
- We assume that $Y = f(X) + \epsilon$.
- Our problem is to find $\hat{f}(X)$, an estimate of $f(X)$.
- *epsilon* is "irreducible" error (assumed to have mean zero).

Modelling

- We will distinguish between **training** data, which we use to find $\hat{f}$, and **test** data, which we use to evaluate $\hat{f}$.
- We will distinguish between when Y is continuous (or quantitative) and when Y is categorical (or discrete or qualitative).
- Linear regression for continuous Y.
- (Multinomial) Logistic regression for categorical Y.

Modelling

- Linear regression assumes ...

$$f(X) = \beta_0 + \beta_1 * X_1 + ... + \beta_p * X_p$$

- We can fit a linear regression model in R with `lm()`.
- We can obtain predictions from the model with `predict()`.
- We will evaluate models using Root Mean Square Error (RMSE) for linear regression.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2}$$

Modelling

- The "worst" RMSE is the (population) standard deviation.

$$\mathrm{RMSE}_{\min} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_y)^2}$$

- The "best" RMSE is zero, but this will almost certainly require a more flexible model than linear regression and would almost certainly correspond to an overfitted model (which is why we evaluate on a test set).

- We can also use visualisations to explore where the model performs better or worse.

Modelling

- Logistic regression does not model Y directly.
- Instead we model the probability that Y takes one of the possible values.
- Logistic regression assumes ...

$$P(X) = \frac{e^{\beta_0 + \beta_1 * X}}{1 + e^{\beta_0 + \beta_1 * X}}$$

$$\log\left(\frac{P(X)}{1 - P(X)}\right) = \beta_0 + \beta_1 * X$$

- Multinomial logistic regression extends this to more than two possible outcomes for Y.

Modelling

- We can fit a logistic model with `glm(..., family="binomial")`.
- We can obtain predictions from the model with `predict(..., type="response")`.
- These predictions are probabilities that must be converted to values of Y.

Modelling

- We can fit a multinomial logistic model with `nnet::multinom()`.
- We can obtain predictions from the model with `predict(..., type="prob")`.
- These predictions are probabilities that must be converted to values of Y.
- We can obtain predictions from the model with `predict()`.
- These are values of Y.

Modelling

- We can evaluate models by how many predictions are "correct."
- Accuracy is the proportion of correct predictions. It can be misleading if the proportion of Y values in each category are not even.
- When there are only two possible Y values, sensitivity measures the proportion of category 1 that are correct.
- When there are only two possible Y values, specificity measures the proportion of category 2 that are correct.
- A confusion matrix is a table of counts for all combinations of actual Y values and predicted Y values.

Modelling

- The "worst" accuracy is the overall (population) proportion, which leads to always predicting the most common category.
- The "best" accuracy is 1, but this will almost certainly require a more flexible model than logistic regression and would almost certainly correspond to an overfitted model (which is why we evaluate on a test set).
- We can also use visualisations to explore where the model performs better or worse.

Communication

- Literate documents.
- R Markdown documents.

Template for 769 Lab submissions.

- Collect data (import and tidy - often the largest part in this course).
- EDA (distributions, outliers, missing values, correlations), especially if first use of data set(s) or variable(s).
- Model fit.
- Model assessment.
- Reflect (briefly) on the course topic that the lab is built around; what have we used, why, and how successful were we?

- No more than 10 pages in total.

Reading

- Introduction to Statistical Learning (Chapter 2)
  http://www-bcf.usc.edu/~gareth/ISL/
- R Markdown Cheat Sheet
  https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf