B1: Approaches to Building Models in R

Speaker: Simon Tam, EY
Agenda

- Introduction to R
- Key tools
- Simplified modelling process demonstration
- Benefits of using R for building models
- R limitations
- Resources
Introduction to R

• Open source statistical programming language based upon “S”
• R is one of the most popular data science tools (along with Python)
• The base functionality can be expanded using “packages”
• The usage of R has dramatically increased over recent years:
  – Popular with educational and research communities (e.g. LondonR)
  – Known to be used at many of the leading tech firms (Airbnb, Facebook, Google, Twitter, Uber, etc.)
  – R Consortium support from Google, IBM, Microsoft, Oracle, etc.
  – Microsoft have invested significantly in R after their purchase of Revolutions Analytics (R Open, R Server, SQL Server, AzureML)
  – Insurance applications (e.g. library(actuar), library(ChainLadder), R in Insurance)
Key tools (1/2) – RStudio, R Markdown, R Notebooks

- RStudio is a popular Integrated Development Environment (IDE) for R
- R Markdown documents can be “knit” into HTML, PDF, Word documents or even PowerPoint slides
  - “R Markdown documents are fully reproducible. Use a productive notebook interface to weave together narrative text and code to produce elegantly formatted output. Use multiple languages including R, Python, and SQL.” ([http://rmarkdown.rstudio.com/](http://rmarkdown.rstudio.com/))
- R Notebooks are an extension of R Markdown documents that allow outputs to be inline with code (similar concept to Jupyter Notebooks)
The successful execution of an R project relies on numerous packages. However, the packages installed by one user may differ from another. To ensure reproducibility, the checkpoint package downloads the packages as of a particular date for use within an R project.

```r
library(checkpoint)
checkpoint("2017-06-06")
```
Simplified modelling process


4. Build models → 5. Validate and score

6. Compile results
1. Acquire data

- Common file formats are easily read into R
  - `library(data.table), fread(...)` for CSV (as an alternative to `read.csv(...)`)
  - `library(readxl)` for Excel
  - `library(haven)` for SAS datasets

- Access and submit SQL queries using ODBC and `library(dplyr)`

- Other packages can be used to access online APIs or scrape information from the internet

- For testing purposes `library(insuranceData)` may be useful

- Use `readRDS(...)` and `saveRDS(...)` to read and save R objects
2. Process data

- Data is usually stored in a `data.frame` object
  - Consider using `tibble` which makes some improvements in the default `data.frame` behaviour

- Two main packages are used for processing data in R
  - `library(dplyr)` uses action verbs to act upon data frames
  - `library(data.table)` is faster and more powerful however the syntax is more challenging to learn

- `library(feather)` can be used to datasets in a faster binary format
  - `feather` was co-created by the authors of `dplyr` and Python package `pandas`
  - This package was meant to be cross compatible between R and Python
3. Explore and visualise

- `library(ggplot2)` is a very popular graphics package for R
  - “The Grammar of Graphics” is written by Leland Wilkinson and defines a structure to the way data is presented and visualised
  - Graphs are built up by defining aesthetics (assign data to x, y, fill, etc.), geoms (types of plots), and other components such as labels, axes, titles, themes etc. `facet_grid` can create panels by dimension.
  - Commonly used in professional publications (newspapers, data journalism, etc.)
  - Functionality can also be expanded using extensions
    - `library(ggvis)` is the next iteration of “gg” graphics with interactive components on the Shiny platform

- Alternatives include the Base R graphics and `library(lattice)` which focuses on displaying multivariate relationships

- Note: two axes charts are not easy to implement in R
4. Build models

- \texttt{glm(...)} is already included within the included \texttt{library(stats)} package
  - Use \texttt{step(...)} to execute stepwise regression
  - Use \texttt{drop1(glmfit, test="Chisq")} to test factor significance
  - \texttt{library(broom)} makes it easier to process \texttt{glm(...)} output using the verbs \texttt{tidy(...), glance(...), augment(...)}
  - Note: \texttt{glm} objects should be pared down to save memory (http://www.win-vector.com/blog/2014/05/trimming-the-fat-from-glm-models-in-r/)

- A selection of popular machine learning packages is listed below:

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>earth</td>
<td>Multivariate Adaptive Regression Splines</td>
</tr>
<tr>
<td>gam</td>
<td>Generalized additive models with smoothness estimation</td>
</tr>
<tr>
<td>gbm</td>
<td>Gradient Boosted Regression models</td>
</tr>
<tr>
<td>glmnet</td>
<td>Lasso and Elastic-Net Regularized GLMs</td>
</tr>
<tr>
<td>lme4</td>
<td>Linear Mixed-Effects Models using ‘Eigen’ and S4</td>
</tr>
<tr>
<td>xgboost</td>
<td>Extreme Gradient Boosting</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>caret</td>
<td>(C)lassification And (RE)gression (T)raining</td>
</tr>
<tr>
<td>h2o</td>
<td>R scripting functionality for H2O, open source math engine</td>
</tr>
<tr>
<td>sparklyr</td>
<td>R Interface to Apache Spark</td>
</tr>
</tbody>
</table>
5. Validate and score (1/2)

- Models can be easily scored on different datasets using the associated `predict(...)` functions
- Deviance can be extracted from `glance(...)`
- `library(Hmisc), rcorr.cens(...)` used to determine the Gini coefficient
- `library(ggplot2)` can be used to create coefficient and standard error graphs using `geom_pointrange(...)`

- Other techniques to consider
  - k-fold cross-validation can be implemented using `cv.glm(...)` although difficult to interpret and perhaps better done using a `library(purrr)` approach
  - `library(broom)` has functionality to bootstrap models
5. Validate and score (2/2)

- *data.frames* are not constrained to holding “data”; they can also hold lists of other objects.
- For example, the structure below could be useful in examining different model types:

```
<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
<th>Model_Object</th>
<th>Gini_Train</th>
<th>Gini_Val</th>
<th>Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gradient Boosted Tree</td>
<td>model_gbm</td>
<td>0.342342</td>
<td>0.27373</td>
<td>list...</td>
</tr>
<tr>
<td>2</td>
<td>Lasso</td>
<td>model_glmnet</td>
<td>0.340994</td>
<td>0.2699</td>
<td>list...</td>
</tr>
<tr>
<td>3</td>
<td>Mixed Model</td>
<td>model_mm</td>
<td>0.330509</td>
<td>0.2238</td>
<td>list...</td>
</tr>
<tr>
<td>4</td>
<td>XGBoost</td>
<td>model_xgboost</td>
<td>0.350329</td>
<td>0.28882</td>
<td>list...</td>
</tr>
</tbody>
</table>
```

- *library(purrr)* facilitates computing metrics and diagnostics across multiple models.
- This structure also makes it easy to extract all relevant information for one model iteration.
6. Compile results

- Previously discussed R Markdown and R Notebook
- R Markdown also supports Python code chunks
- Government Digital Services (GDS) is using R Markdown for a project
  - Working on a project with the Department for Culture, Media, and Sport (DCMS) on the production of “Economic Estimates for DCMS Sectors Statistical First Release (SFR)”
  - “At any point in the future we should be able to look back at this work and be able to reproduce everything that we have done today - something that is difficult with manual/semi-manual processes.” (GDS)
  - https://gdsdata.blog.gov.uk/2017/03/27/reproducible-analytical-pipeline/
Benefits of using R for building models

- In “Assessing data analysis and programming” (Hadley Wickham, Garret Grolemund), three properties of good data analysis are noted:
  - [Reproducibility] Rather than using multiple pieces of software which require user intervention, the modelling process can be completed more efficiently from end to end exclusively in R
  - [Automation] With a robust process in R, changes to data and models are easily managed within the processing framework, e.g. updating models with new data and creating comparisons against existing models
  - [Communication] Description of the modelling process is improved with Markdown and Notebook documents
- Packages can be used to enhance and support every phase of the modelling process
- New techniques can be quickly tested within the modelling framework
- Very easy to get started
- Community support and established knowledge base online
R limitations

• All objects are stored in memory

• For larger scale projects, professionally supported versions or cloud-based solutions should be considered

• Python offers functionality that in many cases exceeds R and is perhaps preferred by data scientists and computer programmers https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis

• Newer packages and developments may not be as thoroughly vetted as commercial software options
Resources

• R help function, e.g. `help(glm)`

• Package vignettes which provide examples

• Hadley Wickham is a well known author/contributor of many of the packages discussed today and Chief Scientist at RStudio [http://www.tidyverse.org/](http://www.tidyverse.org/)

• RStudio Cheat Sheets (dplyr, ggplot2, rmarkdown, etc.) [https://www.rstudio.com/resources/cheatsheets/](https://www.rstudio.com/resources/cheatsheets/)

• Cross Validated (Stack Overflow) [https://stats.stackexchange.com/](https://stats.stackexchange.com/)

• YouTube
The views expressed in this [publication/presentation] are those of invited contributors and not necessarily those of the IFoA. The IFoA do not endorse any of the views stated, nor any claims or representations made in this [publication/presentation] and accept no responsibility or liability to any person for loss or damage suffered as a consequence of their placing reliance upon any view, claim or representation made in this [publication/presentation].

The information and expressions of opinion contained in this publication are not intended to be a comprehensive study, nor to provide actuarial advice or advice of any nature and should not be treated as a substitute for specific advice concerning individual situations. On no account may any part of this [publication/presentation] be reproduced without the written permission of the IFoA [or authors, in the case of non-IFoA research].
Contact

Presenter:
Simon Tam  STam@uk.ey.com

Alternate contacts:
Andy Saunders  ASaunders1@uk.ey.com
Ben Wilson  BWilson2@uk.ey.com
Theeban Kuganesan  TKuganesan@uk.ey.com