The Data Scientist's Toolbox

What is Data Science?

- Data science is using data to answer questions
- It can involve statistics, computer science, mathematics, data cleaning formatting and visualisation
- Big data involves large data sets (volume) which are growing quickly (velocity) and which contain many different types/formats of data (variety)

What is Data?

- "*Information, especially facts or numbers collected to be examined and considered and used to help decision making*" – Cambridge Dictionary
- "*A set of values if qualitative or quantitative variables"* – Wikipedia
- Set: the population you are trying to discover something about
- Variable: Measurements of characteristics of an item
- Qualitative Variable: Measurements or information about qualities, usually recorded with a word (e.g. hair colour, birthplace or sex)
- Quantitative Variable: Measurements or information about quantities usually recorded with a number (e.g. height, weight or temperature)
- Data is generally messy and requires some processing and cleaning to transform it into a form which can be analysed
- A good data scientist asks a question first and seeks data second rather than letting the data drive the research (though in reality this is rarely the case)

Getting Help

Try the following (in this order):

1. Check for typos in code
2. Read error messages to find the source of the issue
3. Read the manuals or help files for R (type ? in R)
4. Try forums (e.g. Stack Overflow or Cross Validated) to see if your issues or a similar issue has already been resolved
5. Post to a forum ensuring you follow the below etiquette:
    - Be polite and courteous
    - Be highly specific about the issue you're having
    - Explain what you tried and what you expected to happen
    - Provide example data that illustrates the issue (enough to explain the problem but not so much so as to swamp any potential helpers)

The Data Science Process

1. Form the question you are trying to answer
2. Find or generate the data to answer the question
3. Explore and clean the data
4. Analyse the data using statistical and machine learning techniques
5. Draw conclusions

6. Visualise and communicate the results

Installing R

- R is a free, open-source extremely powerful coding language primarily used for statistical analysis
- Install from CRAN website

Installing RStudio

- RStudio is a graphical user interface for R which improves the general functionality and usability of R

RStudio Tour

- RStudio has 4 quadrants for the source, the console, the environment and files/plots/packages/help as well as a menu bar
- The console is where commands are inputted and the results executed
- Ensure you set the working directory to the desired folder

R Packages

- A package is a set of data and functions which improves upon the (somewhat limited) functionality of the base R programme
- Anyone can make and share a package and there are packages for a variety of tasks
- CRAN and GitHub are sources of R packages
- Packages are installed via the command install.packages(c("package_1", …, "package_n"))
- Once installed, packages must be loaded via the command library(packagename)
- Use update.packages() to ensure you have the latest version of all packages
- Use sessionInfo() to find what version of R you are running and list all your installed packages
- To find what functions are contained within a packages use help(package = "packagename")

Projects in R

- A built-in functionality of RStudio that keeps related file together
- Creating a projects creates a folder where files are kept; upon reopening a projects will restore all those files to the environment
- It makes it easy to organise and share your work as well as making it easy to start back on a project after a break
- Done via File>New Project (you can also create a project from an existing folder)
- In general, it's best practice to have separate folders for data, scripts and output

Version Control

- A system that records changes to a set of files over time
- It helps to avoid keeping multiple similar copies of a file which could lead to using the wrong version
- Who made the change, what the change was and why the change was made should all be recorded
- It helps to integrate changes made by multiple people
- Git is a free and open source version control system and is the most common version control software
- Git keeps a version of your document that you can edit offline and then share with others once you have finished making your changes; this allows multiple people to work in parallel since you are not waiting for someone to finish their part before you can start work
- Git is software used locally on your PC to record changes while GitHub is an online host for your files which also records the changes made
- Repository: The equivalent of a project directory where all your versions and recorded changes are held
- Commit: To save your changes made (typically a note about what is changed and why is included by the user)
- Push: To update the repository with your (committed) edits for everyone to access
- Pull: To update your local version of the repository to the current version with the changes others have made
- Staging: Preparing files to be committed (best to commit just one file at a time)
- Branch: When a file has two simultaneous copies
- Merge: When independent edits of a single file are brought together intro one single file. This is potentially problematic if there are contradictory edits
- Conflict: When multiple people make conflicting edits to a file; Git will recognise this and ask for user assistance to either manually decide which edits to keep or to decide which version to keep
- Clone: To make a clone of an existing Git repository
- Fork: A personal copy of a repository taken from another person; edits are recorded on your repository not theirs
- You should make purposeful single issue commits which have informative commit messages and you should try to push and pull often

GitHub and Git

- The bell icon will take you to notifications where you will find messages and notifications for all the repositories, teams, and conversations you are a part of
- Set username in Git
- Set commit email address for Git and Github (note these can differ)
- GitHub provides a no-reply email address if you don’t want to use your personal one

- Create a repository
- When committing edits, to the default branch, choose to create a new branch for your commit and then create a pull request to propose your changes; doing so means the default branch only contains finished and approved work.

Linking GitHub and RStudio

- To link GitHub and RStudio do the following: Tools>Global Options>Git/SVN>Ensure git.exe is located in the correct folder>Create RSA key>View Public Key>Copy Key>Github.com>Settings>SSH and GPG Keys>New SSH key>Paste key>Confirm
- To link a repository with RStudio go to File>New Project>version Control>Git>Paste Repository URL>Name and choose location for project>Create Project
- Any files created in R can now be linked with Github by saving the file (by default in the new project)>checking the staged box next to the file name> Commit>Enter a commit message>Commit>Check over the changes and when happy click Push

Projects Under Version Control

- To link an existing R project to version control do the following:
- Open Gitbash>Navigate to the location of your projects via the command cd filepath>git init(initialises repository)>git add . (adds all files in the current location to the repository)>git commit -m “Commit Comment” (commits files to the repository with a comment)>Github.com New Repository>Ensure name is identical to R project>uncheck README>Copy code under “…or push an existing repository from the command line” into Gitbash
- After doing the above you should see the Git tab in the environment quadrant in RStudio and should now be able to push to GitHub from within RStudio

R Markdown

- A way of creating fully reproducible documents that combine text, code and images/graphs
- Outputted as HTML, pdf or word documents without changing the syntax
- Found in the RMarkdown Package
- Enclose code chunks by ```
- RMarkdown cheat sheet

Types of Data Science Questions

In order of increasing complexity:

1. Descriptive: Summarise a set of data
2. Exploratory: Find new relationships between variables and suggest hypotheses for future studies and data collection (but remember correlation does not imply causation)

3. Inferential: Use a sample of data to make generalisations about the wider population (including a measure of uncertainty in your estimate)
4. Predictive: Use current data to make predictions about future data (but remember just because one variable may predict another does not mean one causes the other)
5. Casual: To see the effect of manipulating one variable on another variable (gold standard in data science)
6. Mechanistic: Understand the exact changes in variables that lead to changes in other variables (more common I physical or engineering sciences where often the only noise in the data is measurement error)

<u>Experimental Design</u>

- Ensuring you have the correct data (and enough of it) to properly answer your data science question
- The steps are
    1. Formulate your question
    2. Design your experiment
    3. Identify issues and sources of error
    4. Collect the data
- Independent Variable: What the experimenter manipulates and had is unaffected by other variables (e.g. treatment group)
- Dependent variable: Expected to change as a result of changes in the independent variable (e.g. change in tumour size)
- Hypothesis: An educated guess as to the relationship between your variables
- Sample Size: The number of individuals from which to take measurements
- Confounder: An extraneous variable which may affect the relationship between the independent and dependent variables (e.g. age is a confounder for lung size and frequency of smoking in children)
- Blinding: When a subject does not know whether they have received the treatment or not; this accounts for the placebo and Hawthorne effect
- Double blinding: When the subject and researcher do not know whether the subject has received the treatment or not; this accounts for experimenter bias
- If you can’t control for something, randomise it
- Replication is repeating an experiment with different subjects; if can reach the same conclusion your study is much stronger and also allows you to better measure variation and error
- P-hacking is where you search data sets for statistically significant relationships until you find one and declare it to be true when in reality it probably occurred by chance

<u>Big Data</u>

- Technology has allowed many forms of data that could not previously be recorded to now be able to be recorded in a way that a computer can analyse (e.g. audio, video, GPS)
- Increasingly, unstructured data is becoming the norm and the challenge of how to analyse messy data is becoming more salient

- On the other hand, big data does have its perks:
    - Volume: A large data set means the effects small errors are lessened
    - Velocity: Large amounts of new data can allow for highly informed real-time decision making
    - Variety: Unconventional data sources allow you to answer unconventional questions
    - Unstructured: Hidden correlations can be resolved