Putting data science to work – a case study

Alex Breeze & Martin Tynan
Octo Telematics

5 November 2018
Putting data science to work

Setting the scene

Version control using git

Managing dependencies in R and Python

Final thoughts
Setting the scene
Data science workflow picture from: https://ismayc.github.io/poRtland-bootcamp17/
Data science team?

Differences
- Collaboration
- Review
- Documentation
- Reproducibility

Results
- Reliability of solution
- **Sustainably** adding value
- Regulatory approval
# What’s new? Open source!

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Risks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cutting edge</td>
<td>Is it trustworthy?</td>
</tr>
<tr>
<td>Rapid improvements</td>
<td>Rapidly outdated</td>
</tr>
<tr>
<td>Free / open licence</td>
<td>Use at own risk</td>
</tr>
<tr>
<td>Leverage existing solution</td>
<td>Learning curve</td>
</tr>
<tr>
<td>New tools to <em>embed</em> good practice</td>
<td>New concepts</td>
</tr>
</tbody>
</table>

Capture *balance* Mitigate

5 November 2018
This presentation

1. Version control using Git
2. Managing dependencies in R and Python

Caveat:
- This is not (and will never be) the final version of these slides!
- Links to resources at end.
Version control using git
Version Chaos

Why?
- Experimenting
- Feedback
- Reproducibility

How do you choose?
- File name
- Date modified
- Look at each file
- Documentation
Version Chaos

Why?
- Experimenting
- Feedback
- Reproducibility

How do you choose?
- File name
- Date modified
- Look at each file
- Documentation

Needs Version Control
Git

Tree of commits

Snapshot of project & comment

Project beginning
Git in a Team

Need to *share* the *complete* project code folders

You are in **control** of when the syncing occurs
‘Feature Branch’ Workflow

master branch

Feature branch

Point of review

Approved
Example

Data prep

master

Elastic nets

xgboost

Approved

Point of review
Example

Merging is where you combine two commit trees into one unified history.

This can lead to conflicts!
Considerations

- **Tracks** projects
- **Built-in**
- **GUI & Command line**
- **Merge conflicts**
- **Script languages**
- **Extremely flexible**
- **Location of share**
- **Packrat and Conda envs**

- Easy setup
- **£0**
- RStudio
- PyCharm
- Learning
- mergetools
- Differencing
- Branching
- Best practice
- Security
- Online repos
- Compatible
Managing dependencies in R and Python
Team member

Project A
- Code
- Packages
- Run
- Thumbs up

Project B
- Code
- Packages
- Run
- Thumbs up

Project C
- Code
- Packages
- Run
- Results

Internet

Personal Packages
R + packrat approach

- Specification + source files
- Packages
- Code
- Scan
- Results
- Track
- Restore
- Isolate
Our findings: packrat

- Well **documented**
- **Integrates** with RStudio
- *Long* installation time ⇒ Use the R GUI (not RStudio)
- Implement from the **start** of project
- Doesn’t recognise all dependencies
- Only tracks packages, **not R itself**
Python + conda approach

- Specification
- Isolate
- Packages + others
- Track
- Restore
- Code
- Results
Not managing dependencies is not an option
Final thoughts
### Further resources

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Git</strong></td>
<td><strong>Main website</strong></td>
<td><a href="https://git-scm.com/">https://git-scm.com/</a></td>
</tr>
<tr>
<td></td>
<td><strong>Coursera course (free without certificate)</strong></td>
<td><a href="https://www.coursera.org/learn/version-control-with-git">https://www.coursera.org/learn/version-control-with-git</a></td>
</tr>
<tr>
<td></td>
<td><strong>Git in Practice book (free)</strong></td>
<td><a href="https://github.com/GitInPractice/GitInPractice#readme">https://github.com/GitInPractice/GitInPractice#readme</a></td>
</tr>
<tr>
<td></td>
<td><strong>DataCamp course: Intro to Git for Data Science (free)</strong></td>
<td><a href="https://www.datacamp.com/courses/introduction-to-git-for-data-science">https://www.datacamp.com/courses/introduction-to-git-for-data-science</a></td>
</tr>
<tr>
<td><strong>packrat</strong></td>
<td><strong>Main website</strong></td>
<td><a href="https://rstudio.github.io/packrat/">https://rstudio.github.io/packrat/</a></td>
</tr>
<tr>
<td><strong>conda</strong></td>
<td><strong>Main website</strong></td>
<td><a href="https://conda.io/docs/index.html">https://conda.io/docs/index.html</a></td>
</tr>
<tr>
<td></td>
<td><strong>Useful blog post “Conda: Myths and Misconceptions”</strong></td>
<td><a href="https://jakevdp.github.io/blog/2016/08/25/conda-myths-and-misconceptions/">https://jakevdp.github.io/blog/2016/08/25/conda-myths-and-misconceptions/</a></td>
</tr>
</tbody>
</table>

Links checked at time of making this presentation in October 2018
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].