EARL
ENTERPRISE APPLICATIONS OF THE R LANGUAGE

NAO
National Audit Office

Using R and Shiny to Automate Audit

Rachel Kirkham

Enterprise applications of R – London, September 2017

About the NAO

The National Audit Office (NAO) scrutinises public spending for Parliament. Our public audit perspective helps Parliament hold government to account and improve public services.

We audit the financial statements of all central government departments, agencies and other public bodies, and report the results to Parliament. Our other work comprises value for money studies, local audit, investigations, support to Parliament and international activities.

Our recommendations and reports help government improve public services, and our work led to audited savings of £1.21 billion in 2015. This is the highest level of financial savings to date for the taxpayer, and is equivalent to £19 saved for every pound spent.

About Me

- Head of Data Analytics Research Team at the NAO
- Statistics Lead for NAO
- NAO representative on INTOSAI Big Data Working Group
- ACA-qualified chartered accountant

(Most important bit!)

- Taught myself to program in R in 2013
- My favourite/most used package is data.table!

What is Financial Audit at the NAO?

- Purpose is the get sufficient assurance that the financial statements are free from material misstatement due to fraud/error.
- We also provide an opinion on the regularity of transactions summarised by the financial statements (for many of our audits). This considers whether funds have been applied to the purposes intended by Parliament.

Challenges in Government Auditing

- Wide range of clients, some very large (70,000+ employees).
- High Volume of Sensitive Data
- Data Quality mixed (although not exclusive to public sector)
- High variability in systems used for payroll and accounting

The Issue for staff costs testing

- We have ~370 financial statements audits.
- Team using different methods for estimating staff costs.
- **Lots** of spreadsheets – easy to get something wrong, inconsistent formulae, painful to review, some people spending a number of days completing this testing.

Solution

- Build an app to complete end to end process –
  - Data loading
  - Data standardisation
  - Analysis and recalculation
  - Complete Audit tests
  - Identify areas for follow up
  - Document work completed

Why use R?

- Shiny apps relatively portable; just need client to install R
- Good for handling data of this size
- Easy to build reproducible analysis
- Lots of functions that can be exposed to users that they don't have access to in Excel or traditional audit software

Benefits

- Can deploy application Client-side – therefore eliminating need to transfer large volume of sensitive data
- Saves auditor time – upload data, out comes result and working paper for audit file!
- Improves quality, as single method applied across audits and standardised working papers, easy to review.
- Better level of granularity of audit; move away from sample testing to 100% analysis and consideration of outliers instead. Gives us greater comfort of assurance.
- Not a black box – we've written and QA'ed application, so understand all of the calculations under the hood
- Greater client insight – able to provide interesting findings to client.
- Effectively, free (dev time only) rather than us spending £££ for audit software – better value for taxpayers

Use of R and Shiny

- Shiny is great for building an interface to R functionality + pre-defined analysis
- Lightweight app development that doesn't require heavy investment to develop – really important in getting widespread implementation across different types of analysis
- Allows us to expose R functionality to non-programmers
- Auditing is made up of a number of standard procedures, so is easy to break into pre-written scripts, assuming data is standardised (which is isn't)
- By automating as much of the analysis as possible, we free up valuable time for auditing judgements, talking to clients about findings, adding value as well as completing the audit.

Key Packages used

- shiny
- shinyjs
- shinyBS
- DT
- ggplot2 + plotly
- data.table
- dplyr
- rmarkdown
- lubridate
- packrat

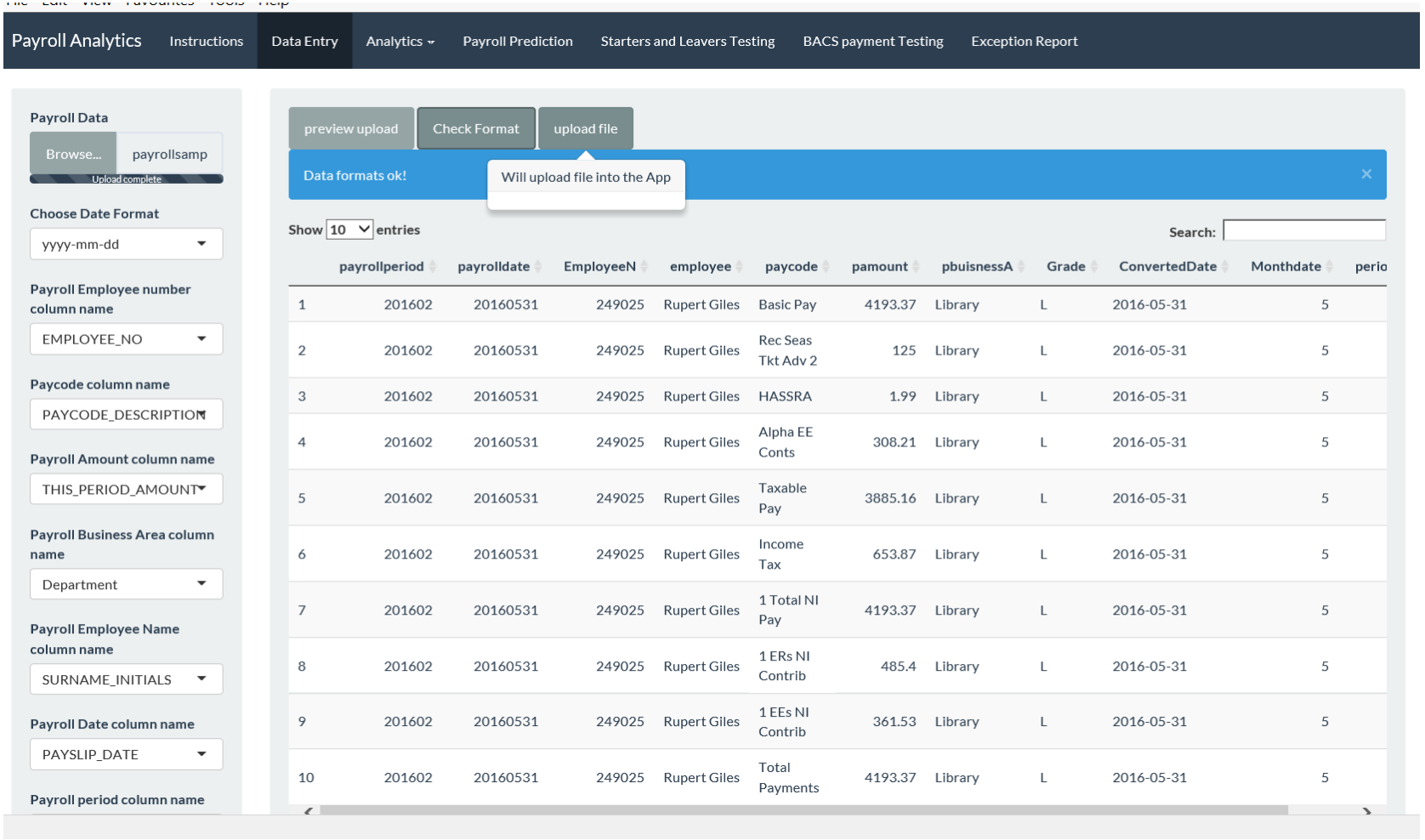
Really important to get UX right for this sort of app – we want our auditors to be able to interact with app with little frustration.

shinyjs and shinyBS are really helpful here.

shinyBS allows us to include pop-ups, alerts and tooltips – great for including usage instructions in the app itself.

shinyjs is really helpful for hiding elements – for example hiding the upload button until data formats have been checked by user.

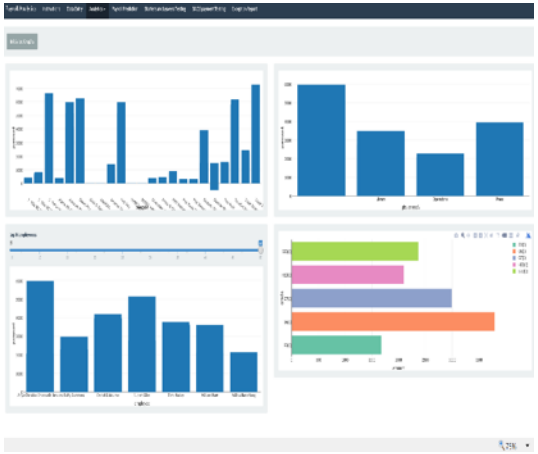
Data Loading and Standardisation

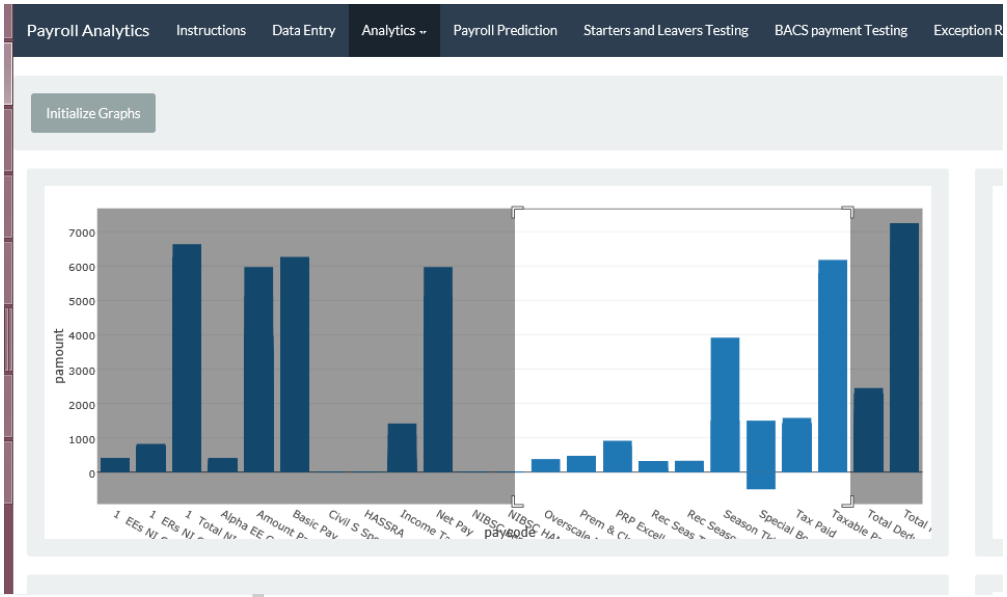


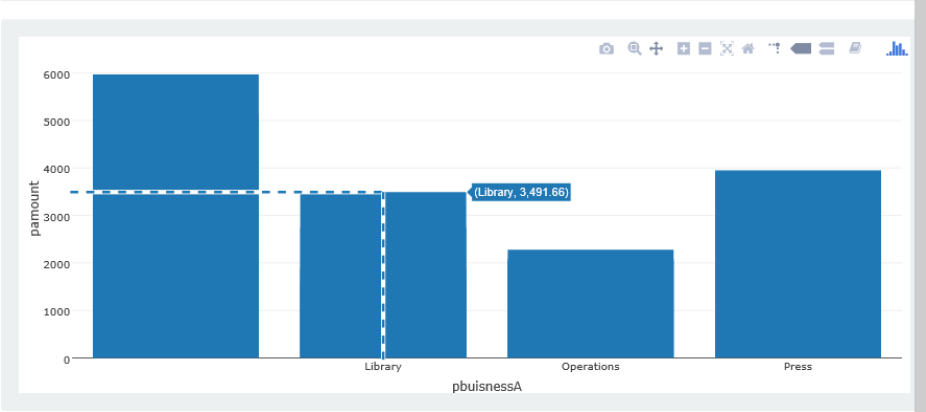
EARL

National Audit Office

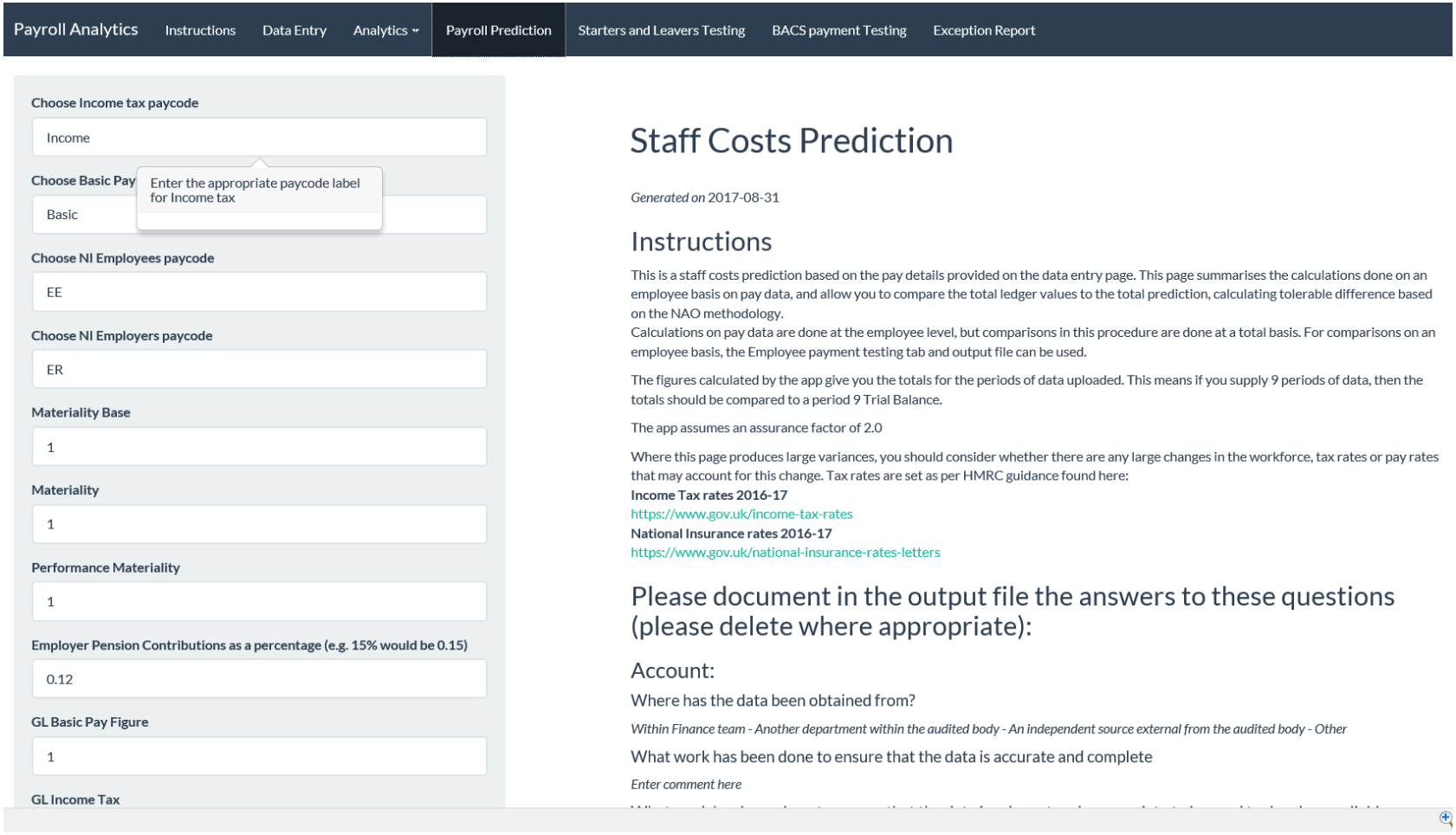
Descriptive Stats on uploaded data







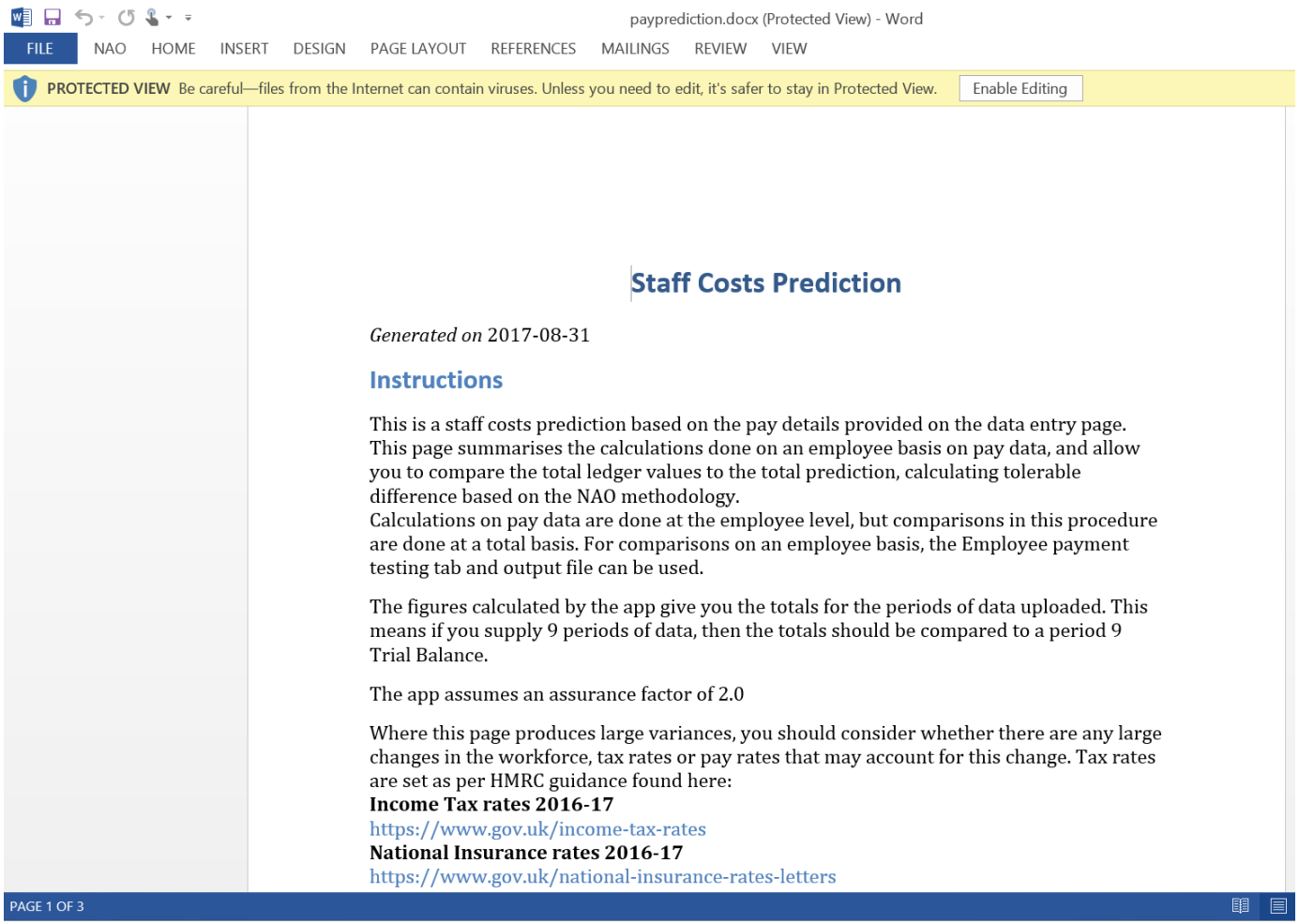
Prediction of Staff costs



EARL

National Audit Office

Generating Audit Documentation



Further Developments

- rintrojs implementation to build a walkthrough of the application
- Implement crosstalk to build tableau style dashboard
- Further analysis options, particularly around outlier detection.

Questions?

- rachel.kirkham@nao.gsi.gov.uk
- 0207 798 5348