

Using R and Shiny to Automate Audit

Rachel Kirkham

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 to 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

File Edit View Favorites Tools Help

Payroll Analytics Instructions Data Entry Analytics Payroll Prediction Starters and Leavers Testing BACS payment Testing Exception Report

Payroll Data

Browse... payrollsamp
Upload complete

Choose Date Format
yyyy-mm-dd

Payroll Employee number column name
EMPLOYEE_NO

Paycode column name
PAYCODE_DESCRIPTION

Payroll Amount column name
THIS_PERIOD_AMOUNT

Payroll Business Area column name
Department

Payroll Employee Name column name
SURNAME_INITIALS

Payroll Date column name
PAYSリップ_DATE

Payroll period column name

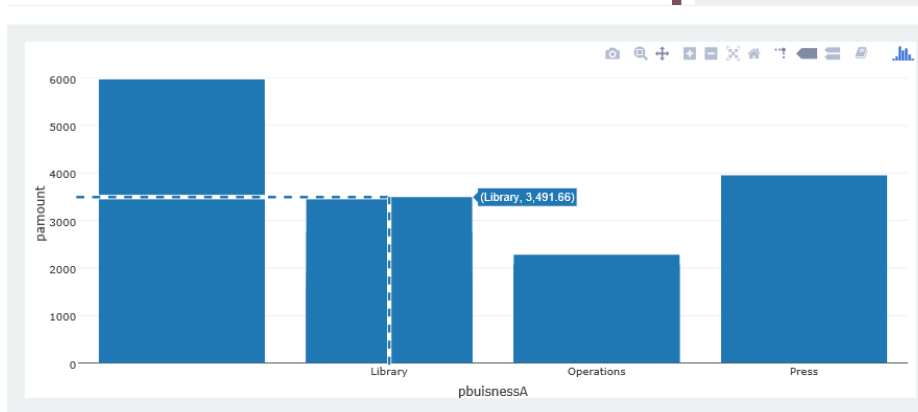
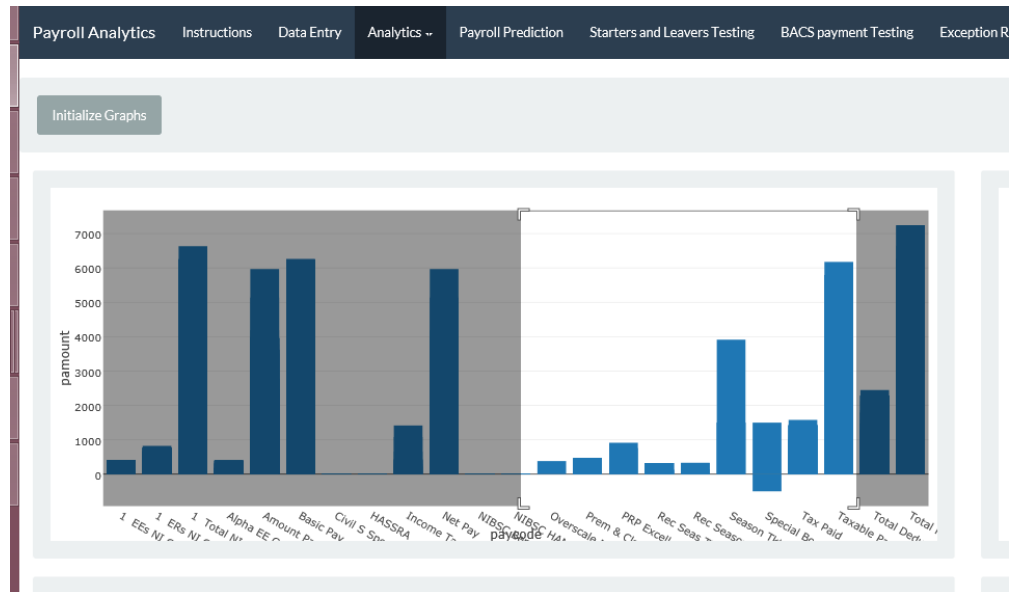
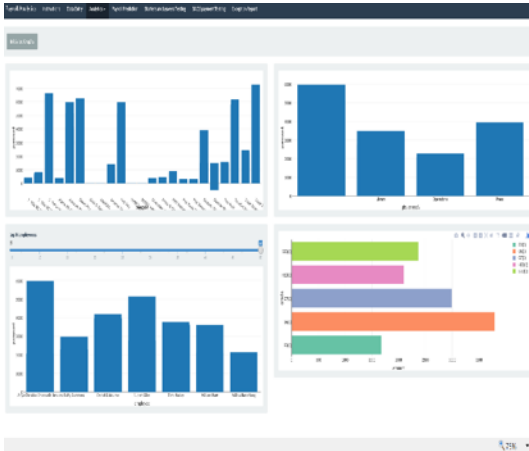
preview upload Check Format upload file

Data formats ok! Will upload file into the App

Show 10 entries Search:

	payrollperiod	payrolldate	EmployeeN	employee	paycode	pamount	pbusinessA	Grade	ConvertedDate	Monthdate	perio
1	201602	20160531	249025	Rupert Giles	Basic Pay	4193.37	Library	L	2016-05-31	5	
2	201602	20160531	249025	Rupert Giles	Rec Seas Tkt Adv 2	125	Library	L	2016-05-31	5	
3	201602	20160531	249025	Rupert Giles	HASSRA	1.99	Library	L	2016-05-31	5	
4	201602	20160531	249025	Rupert Giles	Alpha EE Conts	308.21	Library	L	2016-05-31	5	
5	201602	20160531	249025	Rupert Giles	Taxable Pay	3885.16	Library	L	2016-05-31	5	
6	201602	20160531	249025	Rupert Giles	Income Tax	653.87	Library	L	2016-05-31	5	
7	201602	20160531	249025	Rupert Giles	1 Total NI Pay	4193.37	Library	L	2016-05-31	5	
8	201602	20160531	249025	Rupert Giles	1 ERs NI Contrib	485.4	Library	L	2016-05-31	5	
9	201602	20160531	249025	Rupert Giles	1 EEs NI Contrib	361.53	Library	L	2016-05-31	5	
10	201602	20160531	249025	Rupert Giles	Total Payments	4193.37	Library	L	2016-05-31	5	

Descriptive Stats on uploaded data



Prediction of Staff costs

Payroll Analytics Instructions Data Entry Analytics Payroll Prediction Starters and Leavers Testing BACS payment Testing Exception Report

Choose Income tax paycode
Income

Choose Basic Pay Enter the appropriate paycode label for Income tax

Choose NI Employees paycode
EE

Choose NI Employers paycode
ER

Materiality Base
1

Materiality
1

Performance Materiality
1

Employer Pension Contributions as a percentage (e.g. 15% would be 0.15)
0.12

GL Basic Pay Figure
1

GL Income Tax

Staff Costs Prediction

Generated on 2017-08-31

Instructions

This is a staff costs prediction based on the pay details provided on the data entry page. This page summarises the calculations done on an employee basis on pay data, and allow you to compare the total ledger values to the total prediction, calculating tolerable difference based on the NAO methodology.

Calculations on pay data are done at the employee level, but comparisons in this procedure are done at a total basis. For comparisons on an employee basis, the Employee payment testing tab and output file can be used.

The figures calculated by the app give you the totals for the periods of data uploaded. This means if you supply 9 periods of data, then the totals should be compared to a period 9 Trial Balance.

The app assumes an assurance factor of 2.0

Where this page produces large variances, you should consider whether there are any large changes in the workforce, tax rates or pay rates that may account for this change. Tax rates are set as per HMRC guidance found here:

Income Tax rates 2016-17
<https://www.gov.uk/income-tax-rates>

National Insurance rates 2016-17
<https://www.gov.uk/national-insurance-rates-letters>

Please document in the output file the answers to these questions (please delete where appropriate):

Account:
Where has the data been obtained from?
Within Finance team - Another department within the audited body - An independent source external from the audited body - Other

What work has been done to ensure that the data is accurate and complete
Enter comment here

Generating Audit Documentation

payprediction.docx (Protected View) - Word

FILE NAO HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. [Enable Editing](#)

Staff Costs Prediction

Generated on 2017-08-31

Instructions

This is a staff costs prediction based on the pay details provided on the data entry page. This page summarises the calculations done on an employee basis on pay data, and allow you to compare the total ledger values to the total prediction, calculating tolerable difference based on the NAO methodology.

Calculations on pay data are done at the employee level, but comparisons in this procedure are done at a total basis. For comparisons on an employee basis, the Employee payment testing tab and output file can be used.

The figures calculated by the app give you the totals for the periods of data uploaded. This means if you supply 9 periods of data, then the totals should be compared to a period 9 Trial Balance.

The app assumes an assurance factor of 2.0

Where this page produces large variances, you should consider whether there are any large changes in the workforce, tax rates or pay rates that may account for this change. Tax rates are set as per HMRC guidance found here:

Income Tax rates 2016-17
<https://www.gov.uk/income-tax-rates>

National Insurance rates 2016-17
<https://www.gov.uk/national-insurance-rates-letters>

PAGE 1 OF 3

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