Hands-On Data Science with R
Using CKAN Data Resources

Graham.Williams@togaware.com

19th December 2015

Visit http://HandsOnDataScience.com/ for more Chapters.

The data that sits behind reports that have been produced to guide policy development in government is an important resource that many governments are beginning to make available through various initiatives like https://data.gov, https://data.gov.uk, https://data.gov.au, and https://data.gov.sg to name just a few. In many cases the data is available through the Comprehensive Knowledge Archive Network (**CKAN**) application programming interface (**API**).

In this chapter we explore how to make access this public data accessible via the CKAN API utilizing ckanr (Chamberlain, 2015) and then perform a simple analysis of the dataset.

The required packages for this chapter include:

```
library(ckanr)     # Access data from CKAN.
library(magrittr) # Use pipelines for data processing.
library(readr)     # Modern data reader: read_csv().
library(dplyr)     # Data wrangling: select().
library(jsonlite) # Manage JSON objects.
library(readxl)    # Read Excel spreadsheets.
```

As we work through this chapter, new R commands will be introduced. Be sure to review the command's documentation and understand what the command does. You can ask for help using the `?` command as in:

```
?read.csv
```

We can obtain documentation on a particular package using the *help=* option of `library()`:

```
library(help=rattle)
```

This chapter is intended to be hands on. To learn effectively, you are encouraged to have R running (e.g., RStudio) and to run all the commands as they appear here. Check that you get the same output, and you understand the output. Try some variations. Explore.

1 Introduction

Public and government datasets available through CKAN are a great source of open source data. Some of the available datasets contain data that underlies policy modelling. Having them available allows us to review the actual data that sits behind the policy decision making.

Here we get started using CKAN. After loading the ckanr package we can review the list of `ckanr::servers()` known to be offering up a CKAN API to their data holdings.

```
servers()

##   [1] "http://catalog.data.gov"
##   [2] "http://africaopendata.org"
##   [3] "http://annuario.comune.fi.it"
##   [4] "http://bermuda.io"
##   [5] "http://catalogue.data.gov.bc.ca"
##   [6] "http://catalogue.datalocale.fr"
##   [7] "http://ckan.gsi.go.jp"
##   [8] "http://dados.al.gov.br"
##   [9] "http://dados.gov.br"
##  [10] "http://dados.recife.pe.gov.br"
##  [11] "http://dados.rs.gov.br"
##  [12] "http://dadosabertos.senado.gov.br"
##  [13] "http://dartportal.leeds.ac.uk"
##  [14] "http://data.bris.ac.uk/data"
##  [15] "http://data.buenosaires.gob.ar"
##  [16] "http://data.cityofsantacruz.com"
##  [17] "http://data.edostate.gov.ng"
##  [18] "http://data.glasgow.gov.uk"
##  [19] "http://data.go.id"
##  [20] "http://data.gov.au"
##  [21] "http://data.gov.hr"
##  [22] "http://data.gov.ie"
##  [23] "http://data.gov.ro"
##  [24] "http://data.gov.sk"
##  [25] "http://data.gov.uk"
....
```

For our purposes we will access a particular server. We can check which server is the default with `ckanr::get_default_url()`. To change to another server we set the environment variable CKANR_DEFAULT_URL using `base::Sys.setenv()`.

```
get_default_url()

## [1] "http://data.techno-science.ca/"

Sys.setenv(CKANR_DEFAULT_URL="http://data.gov.au")
get_default_url()

## [1] "http://data.gov.au"
```

The Australian server contains the public dataset that we will use to demonstrate access through the CKAN API.

2 Server Information

Now we can obtain meta data about the server itself and the extensions is supports.

```
ckan_info()

## $ckan_version
## [1] "2.3.1b"
##
## $site_url
## [1] "http://data.gov.au"
##
## $site_description
## [1] "Opening up government data"
##
## $site_title
## [1] "data.gov.au"
##
## $error_emails_to
## [1] "alex.sadleir@linkdigital.com.au"
##
## $locale_default
## [1] "en_GB"
##
## $extensions
##  [1] "disqus"                       "dga_stats"
##  [3] "text_view"                    "webpage_view"
##  [5] "image_view"                   "recline_view"
##  [7] "datastore"                    "datapusher"
##  [9] "agls"                         "datagovau"
## [11] "datagovau_hierarchy"          "googleanalytics"
## [13] "resource_proxy"               "spatial_metadata"
## [15] "spatial_query"                "harvest"
## [17] "ckan_harvester"               "csw_harvester"
## [19] "waf_harvester"                "spatial_harvest_metadata_api"
## [21] "ga-report"                    "sitemap"
## [23] "sentry"                       "cesium_viewer"
## [25] "wms_view"                     "kml_view"
## [27] "geojson_view"                 "officedocs_view"
## [29] "datajson_harvest"             "viewhelpers"
## [31] "dashboard_preview"            "linechart"
## [33] "barchart"                     "piechart"
## [35] "basicgrid"                    "recline_grid_view"
## [37] "recline_graph_view"           "recline_map_view"
## [39] "pdf_view"                     "odata"
## [41] "zip_view"
```

3 Organizations

Many organizations may contribute resources (such as datasets) to a repository. Resources are organized into packages and packages are provided by an organization.

To check how many organizations are represented on the server we might first query for the ckanr::organization_list(). Here we request it be provided in the raw JSON format direct from the API call (as="json". We transform the JSON data structure into an R data structure (a list in this case) and then magrittr::extract2() the result of the CKAN query. We save the result locally to avoid having to query the server again if we wanted the same data (as we do below).

```
organization_list(as="json") %>%
    fromJSON(flatten=TRUE) %>%
    extract2('result') ->
    orgs
```

We can then report the number of organizations represented and the names of the columns contained in the organization table.

```
nrow(orgs)

## [1] 171

names(orgs)

##  [1] "image_display_url" "display_name"      "description"
##  [4] "title"             "package_count"     "created"
##  [7] "approval_status"   "is_organization"   "state"
## [10] "image_url"         "revision_id"       "packages"
## [13] "type"              "id"                "name"
```

4 Prettify Organizations

Using jsonlite (Ooms *et al.*, 2015) we can display formatted JSON objects to make it considerably easier to review using jsonlite::prettify().

```
orgs %>% toJSON() %>% prettify()
```

```
## [
##     {
##         "image_display_url": "https://www.acnc.gov.au/images/ACNC_Logo.png",
##         "display_name": "Australian Charities and Not-for-profits Commissi...
##         "description": "The independent national regulator of charities",
##         "title": "Australian Charities and Not-for-profits Commission",
##         "package_count": 5,
##         "created": "2013-09-03T01:20:25.828188",
##         "approval_status": "approved",
##         "is_organization": true,
##         "state": "active",
##         "image_url": "https://www.acnc.gov.au/images/ACNC_Logo.png",
##         "revision_id": "945f810c-e173-4a90-ba8f-70c9de6f69af",
##         "packages": 5,
##         "type": "organization",
##         "id": "4d907503-b2aa-4c38-ac13-1273e791cb08",
##         "name": "acnc"
##     },
##     {
##         "image_display_url": "",
##         "display_name": "ACT Government",
##         "description": "",
##         "title": "ACT Government",
##         "package_count": 110,
##         "created": "2015-09-16T03:52:10.428251",
##         "approval_status": "approved",
##         "is_organization": true,
##         "state": "active",
##         "image_url": "",
##         "revision_id": "86c71d67-38e7-4569-b772-013b6f616d78",
##         "packages": 110,
##         "type": "organization",
##         "id": "9ad7ef22-735a-45c1-9e83-0f702d56984b",
##         "name": "act-government"
##     },
##     {
##         "image_display_url": "https://data.gov.au/logos/aihw.gif",
##         "display_name": "Australian Institute of Health and Welfare",
##         "description": "The Australian Institute of Health and Welfare (AI...
##         "title": "Australian Institute of Health and Welfare",
##         "package_count": 7,
##         "created": "2013-05-12T09:15:15.624886",
....
```

5 Organizations and Packages

The dataset we are interested in has been provided by the Australian Taxation Office. Thus we want to find this organization in `org` dataset. A visual inspection of the dataset shows that the `title` column contains the appropriate strings for us to search.

```
orgs %>%
  extract("title") %>%
  '=='("Australian Taxation Office") %>%
  which() %T>%
  print() ->
  ato

## [1] 29
```

There are 12 packages available from the ATO.

```
orgs[ato,'packages']

## [1] 12
```

We will store the organization identification so as to investigate the packages further.

```
oid.ato <- orgs[ato,'id'] %T>% print()

## [1] "90d1f157-c01f-4589-93bf-600dee01996e"
```

Out of interest we might select a few organizations before and after our organization of interest and list their number of packages.

```
orgs %>%
  '['(seq(ato-5, ato+5),) %>%
  select(name, packages)

##                                                    name packages
## 24                     australianinstituteofcriminology        1
## 25                                     australianmuseum        0
## 26 australianpesticidesandveterinarymedicinesauthority        1
## 27              australianprudentialregulationauthority       15
## 28                    australianpublicservicecommission       16
## 29                             australiantaxationoffice       12
## 30                     bioregional-assessment-programme        0
## 31                                brisbane-city-council       71
## 32 bureauofinfrastructuretransportandregionaleconomics       18
## 33                                  bureauofmeteorology       40
## 34                  bureauofresourcesandenergyeconomics        0
```

6 Packages

Packages contain resources (often datasets). We can list the packages and note that by default ckanr::package_list() will limit the query to just 31 packages. We also note the number of packages available over the whole repository.

```
package_list(as="table")

##  [1] "0-5m-contours"                                                      ...
##  [2] "10m-contours"                                                       ...
##  [3] "1-1-000-000-scale-australian-geoscience-map-sheet-index-national-geo...
##  [4] "1-100-000-scale-australian-geoscience-map-sheet-index-national-geosc...
##  [5] "1-250-000-scale-australian-geoscience-map-sheet-index-national-geosc...
##  [6] "19th-century-photographs-by-captain-samuel-sweet"                   ...
##  [7] "19th-century-photographs-by-ernest-gall"                            ...
##  [8] "19th-century-photographs-by-townsend-duryea"                        ...
##  [9] "1m-contours"                                                        ...
## [10] "1-second-srtm-derived-hydrological-digital-elevation-model-dem-h-ver...
## [11] "1-second-srtm-level-2-derived-digital-elevation-model-v1-063422"    ...
## [12] "2001-02-to-2007-08-local-government-survey-victoria"                ...
## [13] "2003-bushfire-affected-areas"                                       ...
## [14] "2007-community-attitudes-to-privacy-survey-data"                    ...
## [15] "2008-assembly-election-summary-of-first-preference-votes-by-electora...
## [16] "2008-assembly-election-summary-of-first-preference-votes-by-party-an...
## [17] "2008-seismic-workstation-packages"                                  ...
## [18] "2008-tree-canopy-urbanforest-7d1bf"                                 ...
## [19] "2009-2059-act-population-projections"                               ...
## [20] "2009-2059-act-population-projections-females-by-single-year-of-age" ...
## [21] "2009-2059-act-population-projections-males-by-single-year-of-age"   ...
## [22] "2009-2059-act-population-projections-persons-by-single-year-of-age" ...
## [23] "2009-green-light-report"                                            ...
## [24] "2009-seismic-workstation-package"                                   ...
## [25] "2011-seismic-workstation-packages"                                  ...
## [26] "2011-tree-canopy-urbanforest-adec6"                                 ...
## [27] "2012-offshore-petroleum-acreage-release-areas04479"                 ...
## [28] "2013-14-austender-ict-contract-statistcs"                           ...
## [29] "2013-community-attitudes-to-privacy-survey-data"                    ...
## [30] "2013-offshore-petroleum-acreage-release-areasf2a22"                 ...
....

package_list(as="table", limit=NULL) %>% length()

## [1] 7118
```

7 Finding a Specific Package

To illustrate the use of the CKAN API we will access the controversial Corporate Tax Transparency dataset released 17 December 2015 by the Australian Taxation Office.

Then we search for a particular package hosted on the server. We are interested in one provided by the ATO.

```
package_search(q="australiantaxationoffice", as="json") %>%
    fromJSON(flatten=TRUE) %>%
    extract2('result') %>%
    extract2('results') ->
    pkgs.ato
names(pkgs.ato)

##  [1] "license_title"                "maintainer"
##  [3] "relationships_as_object"      "jurisdiction"
##  [5] "temporal_coverage_to"         "private"
##  [7] "maintainer_email"             "num_tags"
##  [9] "geospatial_topic"             "id"
## [11] "metadata_created"             "spatial_coverage"
....

nrow(pkgs.ato)

## [1] 10
```

Let's have a look at the titles to find the one we are interested in.

```
pkgs.ato$title

##  [1] "Corporate Tax Transparency"
##  [2] "Cumulative Total Tax Returns Received"
##  [3] "Taxation Statistics 2010-11"
##  [4] "Taxation Statistics 2009-10"
##  [5] "Taxation statistics - individual sample files"
##  [6] "Taxation Statistics 1994-95 to 1998-99"
##  [7] "Ad-hoc requested data"
##  [8] "Taxation Statistics 1999-00 to 2003-04"
##  [9] "Taxation Statistics 2004-05 to 2008-09"
## [10] "ATO Web Analytics July 2013 to April 2014"

pkgs.ato$title %>%
  grep('Transparency', .) %>%
  extract(pkgs.ato$id, .) %T>%
  print() ->
  pid.ato.trans

## [1] "c2524c87-cea4-4636-acac-599a82048a26"
```

So that provides us with the package identifier for the Corporate Tax Transparency package from the ATO.

8 Resources

We can then list the contents of the package.

```
pkg.ato.trans <- package_show(pid.ato.trans, as="table") %>% print()

## $license_title
## [1] "Creative Commons Attribution 3.0 Australia"
##
## $maintainer
## [1] "allanbarger"
##
....
```

This contains quite a bit of meta data.

```
names(pkg.ato.trans)

##  [1] "license_title"             "maintainer"
##  [3] "relationships_as_object"   "jurisdiction"
##  [5] "temporal_coverage_to"      "private"
##  [7] "maintainer_email"          "num_tags"
##  [9] "geospatial_topic"          "id"
## [11] "metadata_created"          "spatial_coverage"
....

pkg.ato.trans$name

## [1] "corporate-transparency"

pkg.ato.trans$metadata_created

## [1] "2015-12-08T02:57:38.485976"

pkg.ato.trans$license_title

## [1] "Creative Commons Attribution 3.0 Australia"

pkg.ato.trans$license_id

## [1] "cc-by"

pkg.ato.trans$type

## [1] "dataset"

pkg.ato.trans$num_resources

## [1] 1
```

We see there is just 1 resource—this is a package containing a single dataset.

9 Resource Information

We can list some information about the dataset. The description reads:

> This report contains the total income, taxable income and tax payable of over 1500 public and foreign private entities for the 2013-14 income year.

```
names(pkg.ato.trans$resources)

##  [1] "cache_last_updated"    "package_id"
##  [3] "webstore_last_updated" "datastore_active"
##  [5] "id"                    "size"
##  [7] "wms_layer"             "state"
##  [9] "hash"                  "description"
## [11] "format"                "tracking_summary"
....

pkg.ato.trans$resources[1,]$name

## [1] "2013-14 Report of Entity Tax Information"

pkg.ato.trans$resources[1,]$format

## [1] "XLSX"

pkg.ato.trans$resources[1,]$url

## [1] "http://data.gov.au/dataset/c2524c87-cea4-4636-acac-599a82048a26/resou...
```

10 Download the Resource (Dataset)

From the supplied URL which points to an Excel spreadsheet we will want to extract the data of interest. Trial and error informs us that the second sheet of the spreadsheet contains the data of interest, though it contains three tables so we need to restrict the data to just the first table on that sheet.

In the following code block we record the url and the XLSX file name to create a temporary file into which we store the downloaded dataset. We then `utils::download.file()` and use `readxl::read_excel()` to save it into `transparency`. The `base::unlink()` does the house keeping to remove the downloaded file.

```
url <- pkg.ato.trans$resources[1,]$url
temp  <- tempfile(fileext=".xlsx")
download.file(url, temp)
transparency <- read_excel(temp, sheet=2, skip=1)[1:1540, 2:6]
unlink(temp)
```

We now have a dataset supplied by a CKAN server available to us in R in a usable form as a data frame.

```
dim(transparency)

## [1] 1540    5

names(transparency)

## [1] "Name"             "ABN"              "Total Income $"
## [4] "Taxable Income $" "Tax Payable $"

head(transparency)

## Source: local data frame [6 x 5]
##
##                                     Name         ABN Total Income $
##                                    (chr)       (dbl)          (dbl)
## 1                     3M AUSTRALIA PTY LTD 90000100096      452892659
## 2                       A P EAGERS LIMITED 87009680013     2561960532
## 3       A2 AUSTRALIAN INVESTMENTS PTY LTD 93126014275      110173384
## 4                             AAPC LIMITED 87009175820      570890761
## 5               ABACUS GROUP HOLDINGS LIMITED 31080604619      198340272
## 6 ABB GROUP INVESTMENT MANAGEMENT PTY LTD 47082803852     1173330692
....
```

We now have a dataset supplied by a CKAN server available to us in R in a usable form as a data frame.

11 Data Wrangling

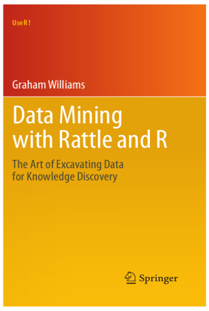
```
ds <- transparency
names(ds) <- c("name", "abn", "total", "taxable", "payable")
ds %<>% mutate(per=100*payable/total, rate=100*payable/taxable)
```

12 Command Summary

This chapter has introduced, demonstrated and described the following R packages, functions, commands, operators, and datasets:

13 Exercises

14 Further Reading


Use R!
Graham Williams
Data Mining with Rattle and R
The Art of Excavating Data for Knowledge Discovery
Springer

The Rattle Book, published by Springer, provides a comprehensive introduction to data mining and analytics using Rattle and R. It is available from Amazon. Other documentation on a broader selection of R topics of relevance to the data scientist is freely available from http://datamining.togaware.com, including the Datamining Desktop Survival Guide.

This chapter is one of many chapters available from http://HandsOnDataScience.com. In particular follow the links on the website with a * which indicates the generally more developed chapters.

We identify below other resources that augment the material we have presented in this chapter.

-

15 References

Chamberlain S (2015). *ckanr: Client for the Comprehensive Knowledge Archive Network ('CKAN') 'API'.* R package version 0.1.0, URL https://CRAN.R-project.org/package=ckanr.

Ooms J, Temple Lang D, Hilaiel L (2015). *jsonlite: A Robust, High Performance JSON Parser and Generator for R.* R package version 0.9.19, URL https://CRAN.R-project.org/package=jsonlite.

R Core Team (2015). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Williams GJ (2009). "Rattle: A Data Mining GUI for R." *The R Journal*, **1**(2), 45–55. URL http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf.

Williams GJ (2011). *Data Mining with Rattle and R: The art of excavating data for knowledge discovery.* Use R! Springer, New York.

This document, sourced from CkanO.Rnw bitbucket revision 17, was processed by KnitR version 1.9 of 2015-01-20 and took 9.6 seconds to process. It was generated by gjw on nyx running Ubuntu 14.04.3 LTS with Intel(R) Xeon(R) CPU W3520 @ 2.67GHz having 8 cores and 12.3GB of RAM. It completed the processing 2015-12-19 14:22:24.