ABS DataLab Manual

Elyse Dwyer

2022-05-12

Contents

Chapter 1

Preface

1.1 Why make a manual on this data?

Like many other large Australian microdata sets, e.g. HILDA, there is a broad scope of data items that are included in the ABS DataLab's suite of products. The way data is organised in DataLab is complex, and datasets are very large in terms of observations. As such, it can be useful to have a guide to get started.

This manual is intended to build on what already has been compiled elsewhere in terms of metadata and best practices, as well as to provide some handy utilities in R for common operations used on the data, e.g. merging data files.

1.2 General Tips for using DataLab

1.2.1 Efficiency is Key

As with most microdata, there is usually a large number of observations, however this is especially true for datasets such as BLADE, where nearly the entire population of Australian firms is observed. It can be helpful to speed up your code by avoiding for loops and/or running operations in parallel rather than series. Packages such as `data.table` in R already provide some parallel functionality, thus it is worth familiarising yourself with this if you have not already. If you are familiar with the R packages in `tidyverse` such as `dplyr`, there are many suitable analogues in `data.table`.

1.2.2 Packages

If you are an R user, DataLab uses R package manager for downloading packages securely. Most packages in CRAN are available in the package manager, however, there are some uncommon/non-CRAN packages which are not included. Where possible, try to avoid using obscure R packages.

1.3 Third-Party DataLab Resources

1.3.1 BLADE

Productivity Commission's discussion of data item coverage in BLADE and its uses for productivity research.

https://www.pc.gov.au/research/supporting/blade

Chapter 2

BLADE

2.1 General Structure of BLADE

BLADE is stored in the Product network drive of your virtual machine in DataLab. There are separate folders for different file types, I recommend working with csv files if you are an R or Python user. There is a separate csv file for each financial year and source, e.g. one file will represent BAS returns in FY18-19. Alongside general data cleaning, you will need to merge across different sources, as well as appending data for each year, if you are interested in panel data.

The 'core' of BLADE consists of four different data sources containing administrative data which most businesses will complete during the financial year:

- Indicative Data Items: This contains the 'spine' of ABNs from which all other datasets can be linked.
- BAS: These are filled out by all firms that remit GST and include information on sales, operating expenses, purchases and capital expenditures
- BIT: These files contain income tax data (including balance sheet data) that is submitted to the ATO.
- PAYG: This contains information on employee headcount and wages expense when payment summaries are submitted to the ATO.

There are other datasets within the extended version of BLADE which cover R & D expenses and customs data, however these tend to cover a smaller proportion of firms within the population.

2.1.1 TAU or ABN levels

There are two levels of observation inside BLADE, the ABN level, which encompasses a legal entity and the TAU level, which attempts to identify the 'economic

firm'. A TAU may consist of multiple ABNs, for example, all of the legal entities that Kmart Australia operates would be aggregated into one TAU. TAUs can be aggregated further into an enterprise id, which in the case of Kmart, would identify its parent company Wesfarmers Ltd.

2.2 BLADE Utils in R

2.2.1 Append multiple time periods together

```
library(matrixStats)
library(data.table)

blade_path <- " " # Directory containing BLADE source files in CSV format

destination_path <-"  " # Directory where merged files will go

file_patterns <- c("frame", "BAS", "BIT", "PAYG") # File path patterns

dest_patterns <- c("ind_items.csv", "BAS.csv", "BIT.csv", "PAYG.csv") # Destination fi

for (i in 1:4) {

    files <- list.files(path, pattern=file_patterns[i], full.names=TRUE)
    files <- lapply(files, fread)
    for (j in 1:length(files)) {
        files[[j]] <- files[[j]][,source:=match(j,files)+10]
    }
    files <- rbindlist(files)
    fwrite(files,file = paste(destination_path, dest_patterns[i], sep="/"))

}
```

2.2.2 Merging BLADE core datasets

```
# This is assuming you have completed the appending step

# Read in the files you saved down

objects <- c("ind_items", "BAS", "BIT", "PAYG")

for (i in 1:4){
```

```
    assign(objects[i],fread(paste(destination_path, dest_patterns[i], sep="/")))
}

# Merge datatable objects

for (i in list(BAS,BIT,PAYG)){

    ind_items <- i[ind_items, on = .(id,tsid)]
}
```

Chapter 3

MADIP

Watch this space for further info