Processing, cleaning and saving NZ GREEN Grid project 1 minute electricity power data

Ben Anderson (b.anderson@soton.ac.uk, @dataknut)

Last run at: 2018-05-15 19:27:04

Contents

1 Status

Test run using reduced data from ~/Data/NZGreenGrid/gridspy/1min_orig/

2 Citation

If you wish to use any of the material from this report please cite as:

- Anderson, B. (2018) Processing, cleaning and saving NZ GREEN Grid project 1 minute electricity power data, University of Otago: Dunedin, NZ.

3 Introduction

Report circulation:

- Restricted to: NZ GREEN Grid project partners and contractors.

3.1 Purpose

This report is intended to:

- load and clean the project electricity power data (Grid Spy)
- save the cleaned data out as a single file per household
- produce summary data quality statistics

The resulting cleaned data has *no* identifying information such as names, addresses, email addresses, telephone numbers and is therefore safe to share across all partners.

The data contains a unique household id which can be used to link it to the NZ GREEN Grid time use diaries and dwelling/appliance surveys. With some additional non-disclosure checks it should also be safe to archive all of these linkable datasets for re-use via the UK reshare service.

3.2 Requirements:

- grid spy 1 minute data downloads

3.3 History

Generally tracked via our git.soton repo:

- history
- issues

3.4 Support

This work was supported by:

- The University of Otago
- The New Zealand Ministry of Business, Innovation and Employment (MBIE)
- SPATIALEC - a Marie Skłodowska-Curie Global Fellowship based at the University of Otago's Centre for Sustainability (2017-2019) & the University of Southampton's Sustainable Energy Research Group (2019-202).

We do not 'support' the code but if you have a problem check the issues on our repo and if it doesn't already exist, open one. We might be able to fix it :-)

4 Obtain listing of files

In this section we generate a listing of all 1 minute data files that we have received. If we are running over the complete dataset then we will be using data from:

- /hum-csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/

In this run we are using data from:

- ~/Data/NZGreenGrid/gridspy/1min_orig/

If these do not match then this may be a test run.

```
## Loading required package: data.table

## Loading required package: lubridate

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year

## The following object is masked from 'package:base':
##
##     date

## Loading required package: readr

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following object is masked from 'package:ggplot2':
##
##     vars

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: progress

## [1] "Looking for 1 minute data using pattern = *at1.csv$ in ~/Data/NZGreenGrid/gridspy/1min_orig/ - c
## [1] "Looking for data using pattern = *at1.csv$ in ~/Data/NZGreenGrid/gridspy/1min_orig/ - could take
## [1] "Found 958 files"
## [1] "Processing file list and getting file meta-data (please be patient)"
## [1] "All files checked"
## [1] "Checking ambiguous date formats in ~/Data/NZGreenGrid/gridspy/1min_orig/rf_46/12Oct2016-20Nov20
## [1] "Saving 1 minute data files interim metadata to ~/Data/NZGreenGrid/gridspy/consolidated/1min/fLis
## [1] "Done"

## [1] "Overall we have 958 files from 2 households."
```

Overall we have 958 files from 2 households. Of the 958, 544 (56.78%) were *not* loaded/checked as their file sizes indicated that they contained no data.

4.1 Date format checks

We now need to check how many of the loaded files have an ambiguous or default date - these could introduce errors.

Table 1: Number of files and min/max date (as char) with given date column names by inferred date format

dateColName	dateFormat	nFiles	minDate	maxDate
date NZ	dmy - definite	1	27/03/2015	27/03/2015
date NZ	mdy - definite	1	5/26/2016	5/26/2016
date NZ	ymd - default (but day/month value <= 12)	1	2014-01-06	2014-01-06
date NZ	ymd - definite	2	2014-05-24	2015-05-25
date UTC	ambiguous	1	11-10-16	11-10-16
date UTC	ymd - default (but day/month value <= 12)	161	2017-01-08	2018-02-12
date UTC	ymd - definite	247	2015-05-24	2018-02-19
unknown - do not load (fsize = 2751)	NA	302	NA	NA
unknown - do not load (fsize = 43)	NA	242	NA	NA

Results to note:

- There are 1 ambiguous files
- The non-loaded files only have 2 distinct file sizes, confirming that they are unlikely to contain useful data.

We now inspect the ambiguous and (some of) the default files.

To help with data cleaning the following table lists files that have ambiguous dates.

```
# list ambigious files
aList <- fListCompleteDT[dateFormat == "ambiguous",
                         .(file, dateColName, dateExample, dateFormat)]

cap <- paste0("All ", nrow(aList),
              " files with an ambiguous dateFormat")

knitr::kable(caption = cap, aList)
```

Table 2: All 1 files with an ambiguous dateFormat

file	dateColName	dateExample	dateFormat
rf_46/12Oct2016-20Nov2017at1.csv	date UTC	11-10-16	ambiguous

Check against file names to see what is reasonable and then fix them.

```
# Setting to dmy seems OK
fListCompleteDT <- fListCompleteDT[dateFormat == "ambiguous",
                                   dateFormat := "dmy - inferred"]

paste0("Fixed ", nrow(aList), " files with an ambiguous dateFormat")
```

```
## [1] "Fixed 1 files with an ambiguous dateFormat"
```

The following table lists up to 10 of the 'date NZ' files which are set by default - do they look OK to assume the default dateFormat? Compare the file names with the dateExample...

```
# list default files with NZ time
aList <- fListCompleteDT[dateColName == "date NZ" & dateFormat %like% "default",
                         .(file, fSize, dateColName, dateExample, dateFormat)]

cap <- paste0("First 10 (max) of ", nrow(aList),
              " files with dateColName = 'date NZ' and default dateFormat")

knitr::kable(caption = cap, head(aList))
```

Table 3: First 10 (max) of 1 files with dateColName = 'date NZ' and default dateFormat

file	fSize	dateColName	dateExample	dateFormat
rf_01/1Jan2014-24May2014at1.csv	6255737	date NZ	2014-01-06	ymd - default (but day/month value <= 12

The following table lists up to 10 of the 'date UTC' files which are set by default - do they look OK to assume the default dateFormat? Compare the file names with the dateExample...

```
# list default files with UTC time
aList <- fListCompleteDT[dateColName == "date UTC" & dateFormat %like% "default",
                         .(file, fSize, dateColName, dateExample, dateFormat)]

cap <- paste0("First 10 (max) of ", nrow(aList),
              " files with dateColName = 'date UTC' and default dateFormat")

knitr::kable(caption = cap, head(aList, 10))
```

Table 4: First 10 (max) of 161 files with dateColName = 'date UTC' and default dateFormat

file	fSize	dateColName	dateExample	dateFormat
rf_46/10Apr2017-11Apr2017at1.csv	292721	date UTC	2017-04-09	ymd - default (but day/month value <= 12
rf_46/10Aug2017-11Aug2017at1.csv	292888	date UTC	2017-08-09	ymd - default (but day/month value <= 12
rf_46/10Dec2017-11Dec2017at1.csv	292823	date UTC	2017-12-09	ymd - default (but day/month value <= 12
rf_46/10Feb2017-11Feb2017at1.csv	286736	date UTC	2017-02-09	ymd - default (but day/month value <= 12
rf_46/10Feb2018-11Feb2018at1.csv	299084	date UTC	2018-02-09	ymd - default (but day/month value <= 12
rf_46/10Jan2017-11Jan2017at1.csv	297659	date UTC	2017-01-09	ymd - default (but day/month value <= 12
rf_46/10Jan2018-11Jan2018at1.csv	294418	date UTC	2018-01-09	ymd - default (but day/month value <= 12
rf_46/10Jul2017-11Jul2017at1.csv	291082	date UTC	2017-07-09	ymd - default (but day/month value <= 12
rf_46/10Jun2017-11Jun2017at1.csv	295979	date UTC	2017-06-09	ymd - default (but day/month value <= 12
rf_46/10Mar2017-11Mar2017at1.csv	290244	date UTC	2017-03-09	ymd - default (but day/month value <= 12

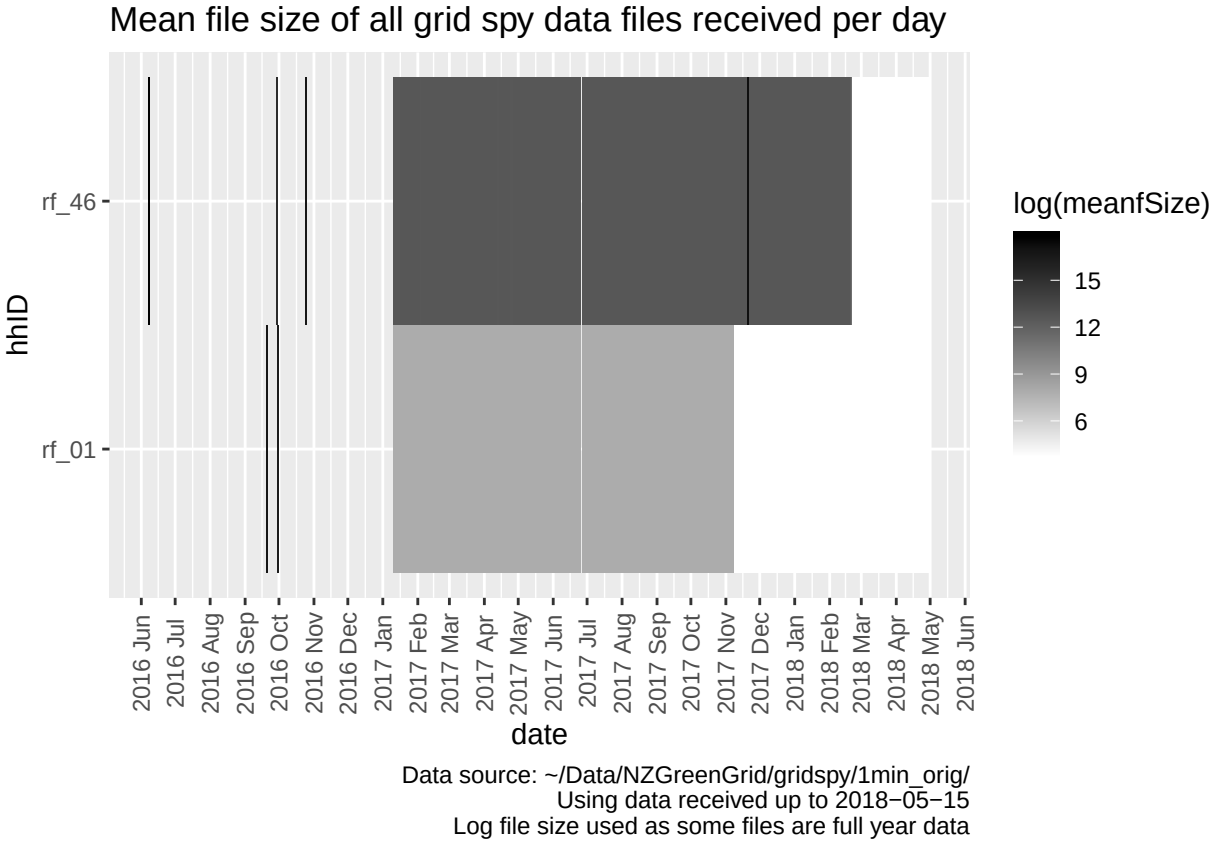
Check final date formats:

Table 5: Number of files & min/max dates (as char) with given date column names by final imputed date format

dateColName	dateFormat	nFiles	minDate	maxDate
date NZ	dmy - definite	1	27/03/2015	27/03/2015
date NZ	mdy - definite	1	5/26/2016	5/26/2016
date NZ	ymd - default (but day/month value <= 12)	1	2014-01-06	2014-01-06
date NZ	ymd - definite	2	2014-05-24	2015-05-25
date UTC	dmy - inferred	1	11-10-16	11-10-16
date UTC	ymd - default (but day/month value <= 12)	161	2017-01-08	2018-02-12
date UTC	ymd - definite	247	2015-05-24	2018-02-19
unknown - do not load (fsize = 2751)	NA	302	NA	NA
unknown - do not load (fsize = 43)	NA	242	NA	NA

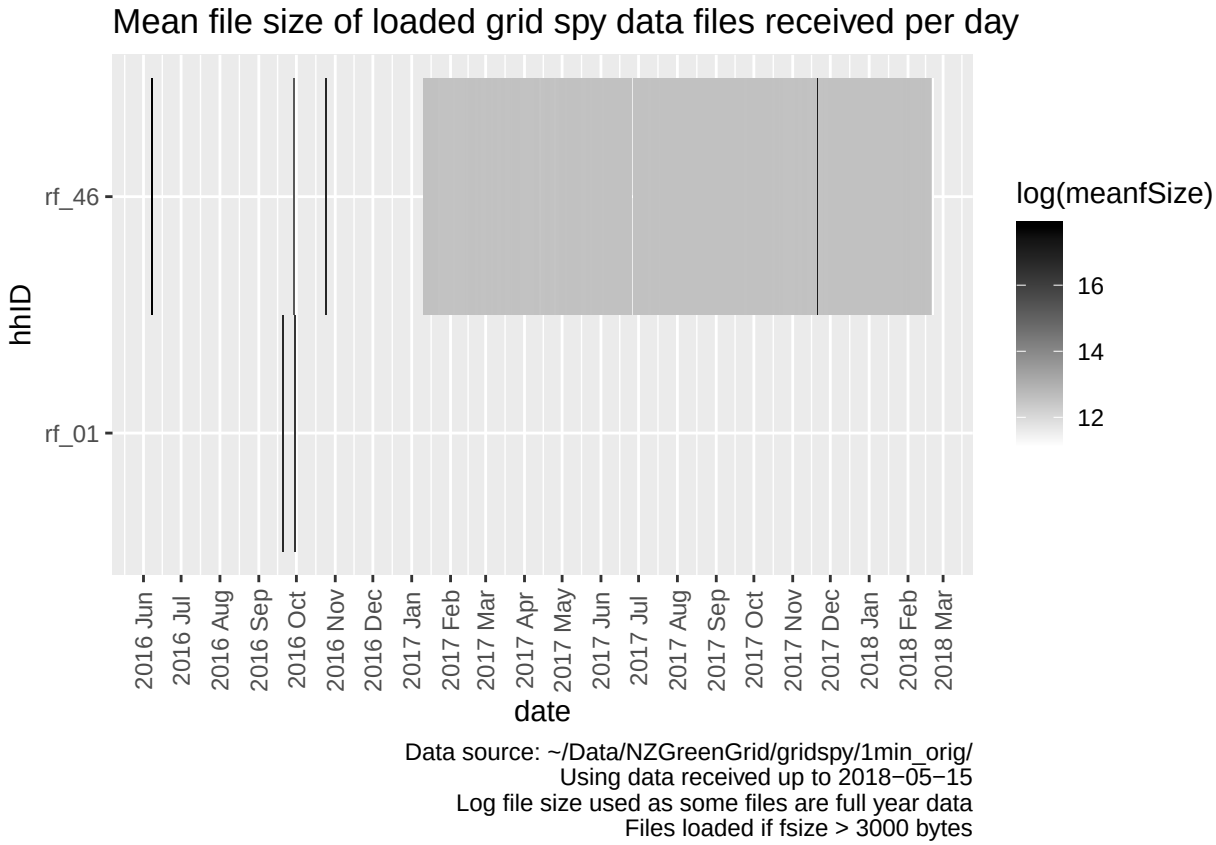
4.2 Data file quality checks

The following chart shows the distribution of these files over time using their sizes. Note that white indicates the presence of small files which may not contain observations.




```
## Saving 6.5 x 4.5 in image
```

The following chart shows the same chart but only for files which we think contain data.




```
## Saving 6.5 x 4.5 in image
```

5 Load data files

In this section we load the data files that have a file size > 3000 bytes. Things to note:

- We assume that any files smaller than this value have no observations. This is based on:
  - Manual inspection of several small files
  - The identical (small) file sizes involved
  - *But* we should probably test the first few lines to double check...
- We have to deal with quite a lot of duplication some of which has caused the different date formats. See our repo issues list.

The following table shows the number of files per household that we willl load.

```
# check files to load
t <- fListCompleteDT[dateColName %like% "do not load", .(nFiles = .N,
                        meanSize = mean(fSize),
                        minFileDate = min(fMDate),
                        maxFileDate = max(fMDate)), keyby = .(hhID)]

knitr::kable(caption = "Summary of household files to load", t)
```

Table 6: Summary of household files to load

hhID	nFiles	meanSize	minFileDate	maxFileDate
rf_01	475	1764.718	2017-01-11	2018-04-29

hhID	nFiles	meanSize	minFileDate	maxFileDate
rf_46	69	43.000	2018-02-22	2018-05-01

```
# load data using external R script
source("../scripts/process1minGridSpyData.R")
```

```
## [1] "Loading: rf_01"
## [1] "Saving ~/Data/NZGreenGrid/gridspy/consolidated/1min/data/rf_01_all_1min_data.csv..."
## [1] "Saved ~/Data/NZGreenGrid/gridspy/consolidated/1min/data/rf_01_all_1min_data.csv, gzipping..."
## [1] "Gzipped ~/Data/NZGreenGrid/gridspy/consolidated/1min/data/rf_01_all_1min_data.csv"
## [1] "Loading: rf_46"
## [1] "Saving ~/Data/NZGreenGrid/gridspy/consolidated/1min/data/rf_46_all_1min_data.csv..."
## [1] "Saved ~/Data/NZGreenGrid/gridspy/consolidated/1min/data/rf_46_all_1min_data.csv, gzipping..."
## [1] "Gzipped ~/Data/NZGreenGrid/gridspy/consolidated/1min/data/rf_46_all_1min_data.csv"
## [1] "Saving daily observations stats by hhid to ~/Data/NZGreenGrid/gridspy/consolidated/1min/hhDaily
## [1] "Done"
## [1] "Saving 1 minute data files final metadata to ~/Data/NZGreenGrid/gridspy/consolidated/1min/fList
## [1] "Done"
```

6 Data quality analysis

Now produce some data quality plots & tables.

6.1 Circuit label checks

The following table shows the number of data files with different circuit labels by household. In theory there should only be one unique list per household and it should be present in every data file.

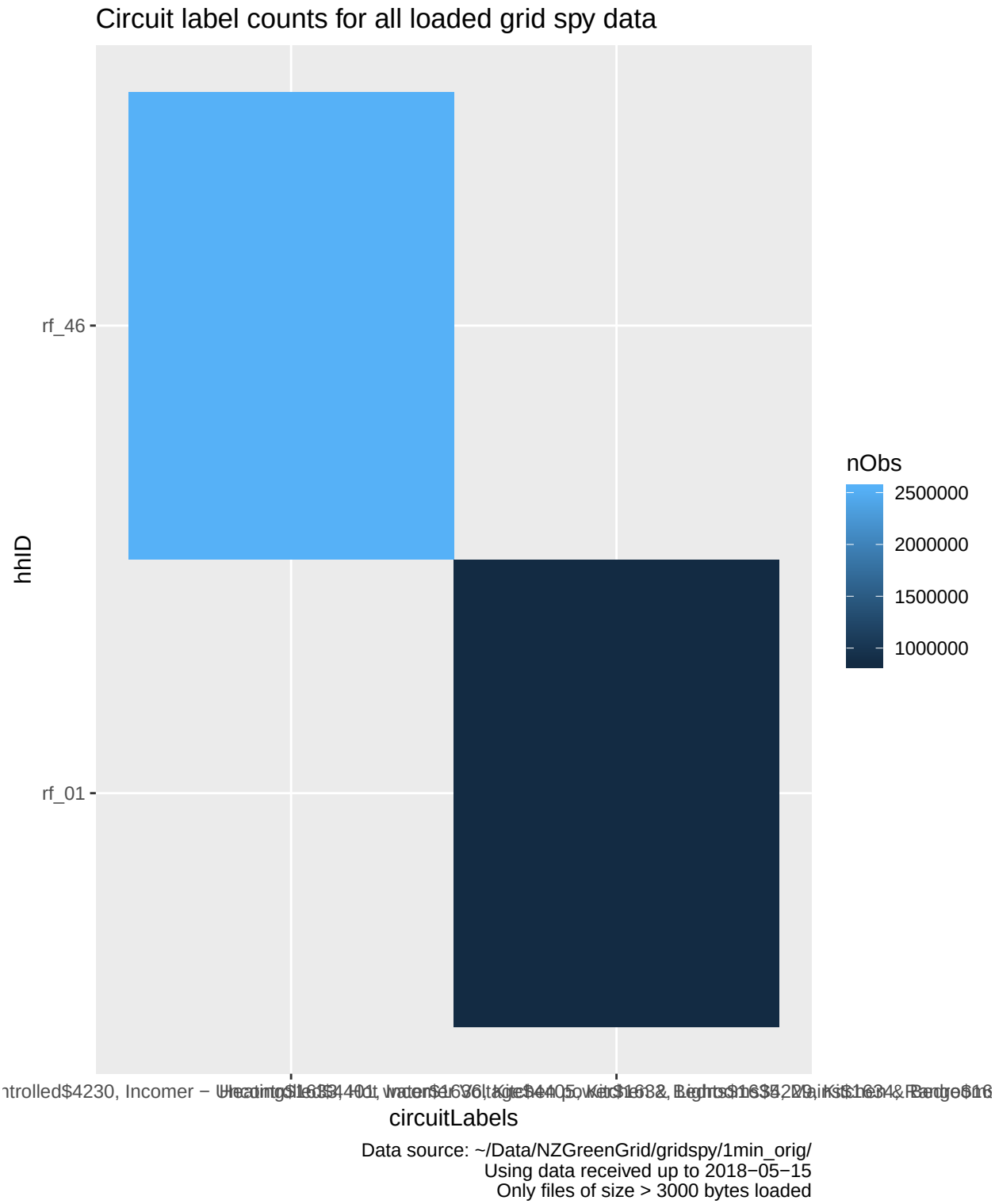
Heat Pumps (2x) & Power$4232, Heat Pumps (2x) & Power$4399, Hot Water - Controlled$4231, Hot Water - Controlled$4
Heating$1633, Hot water$1636, Kitchen power$1632, Lights$1635, Mains$1634, Range$1637

If this is not the case then this implies that:

- some of the circuit labels for these households may have been changed during the data collection process;
- some of the circuit labels may have character conversion errors which have changed the labels during the data collection process;
- at least one file from one household has been saved to a folder containing data from a different household (unfortunately the raw data files do *not* contain household IDs in the data or the file names which would enable checking/preventative filtering). This will be visible in the table if two households appear to share *exactly* the same list of circuit labels.

Some or all of these may be true at any given time!

Errors are easy to spot in the following plot where a hhID spans 2 or more circuit labels.




```
## Saving 6.5 x 8 in image
```

The following table provides more detail to aid error checking. Check for:

- 2+ adjacent rows which have exactly the same circuit labels but different hh_ids. This implies some data from one household has been saved in the wrong folder;
- 2+ adjacent rows which have different circuit labels but identical hh_ids. This could imply the same

thing but is more likely to be errors/changes to the circuit labelling.

If the above plot and this table flag a lot of errors then some re-naming of the circuit labels (column names) may be necessary.

circuitLabels
Heat Pumps (2x) & Power$4232, Heat Pumps (2x) & Power$4399, Hot Water - Controlled$4231, Hot Water - Controlled$4 Heating$1633, Hot water$1636, Kitchen power$1632, Lights$1635, Mains$1634, Range$1637

Things to note:

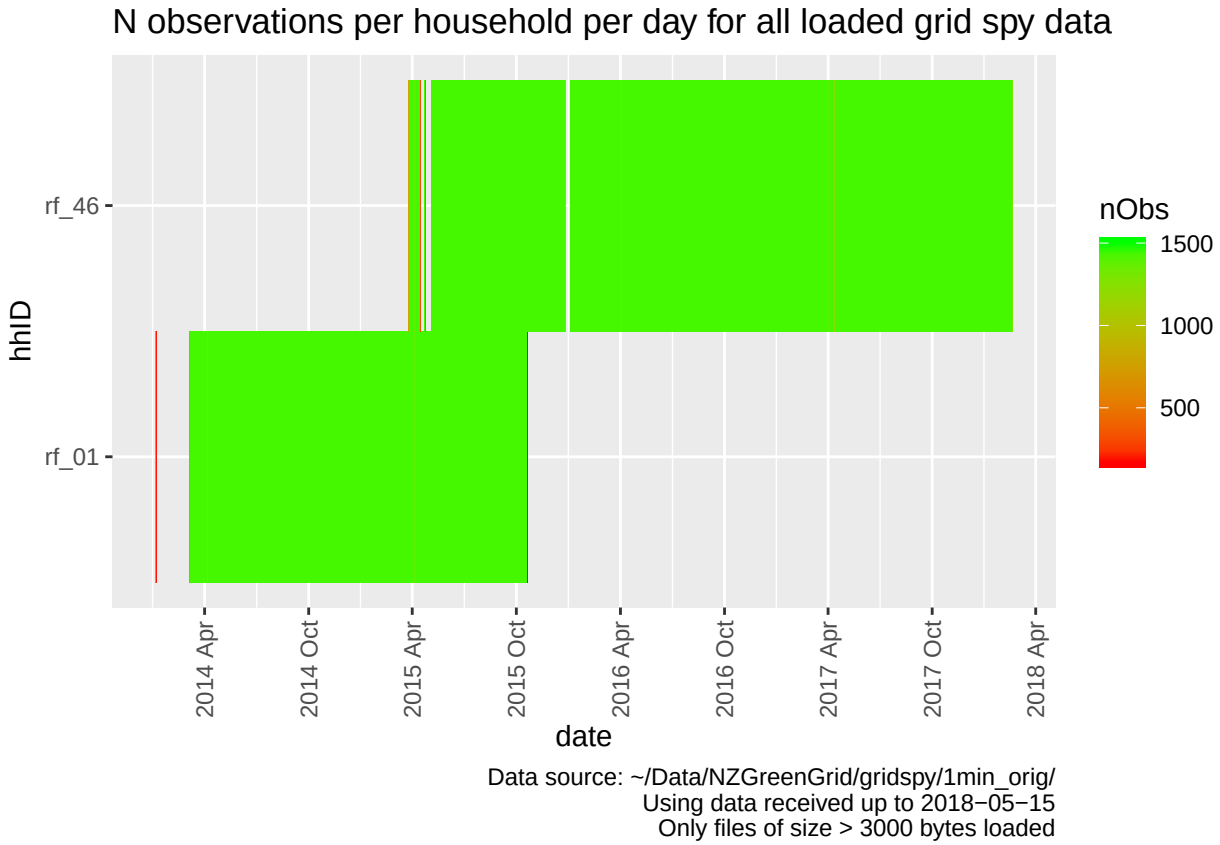
- rf_25 has an aditional unexpected "Incomer 1 - Uncontrolled$2757" circuit in some files but it's value is always NA

6.2 Observations

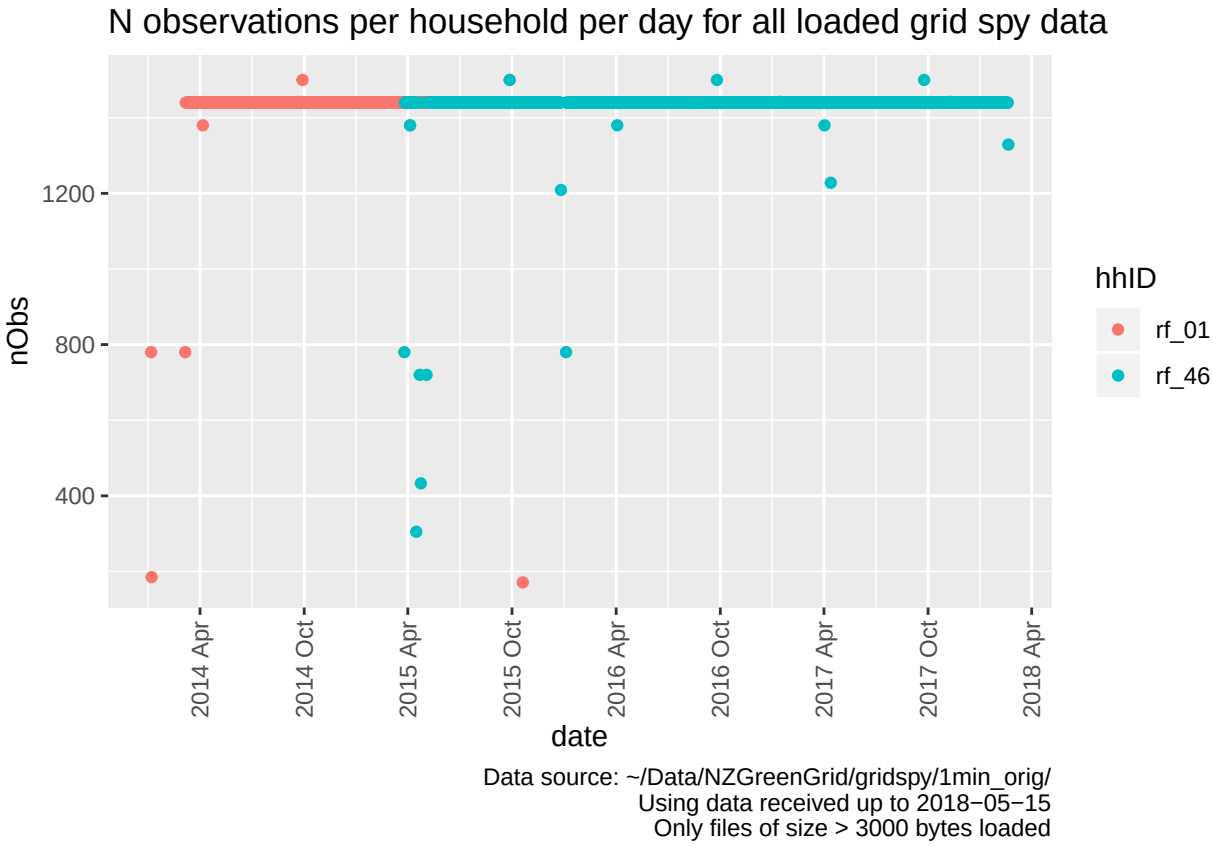
The following plots show the number of observations per day per household. In theory we should not see:

- dates before 2014 or in to the future. These may indicate:
  - date conversion errors;
- more than 1440 observations per day. These may indicate:
  - duplicate time stamps - i.e. they have the same time stamps but different power (W) values or different circuit labels;
  - observations from files that are in the 'wrong' rf_XX folder and so are included in the 'wrong' household as 'duplicate' time stamps.

If present both of the latter may have been implied by the table above and would have evaded the de-duplication filter which simply checks each complete row against all others within it's consolidated household dataset (a *within household absolute duplicate* check).




```
## Saving 6.5 x 4.5 in image
```




```
## Saving 6.5 x 4.5 in image
```

The following table shows the min/max observations per day and min/max dates for each household. As above, we should not see:

- dates before 2014 or in to the future (indicates date conversion errors)
- more than 1440 observations per day (indicates potentially duplicate observations)
- non-integer counts of circuits as it suggests some column errors

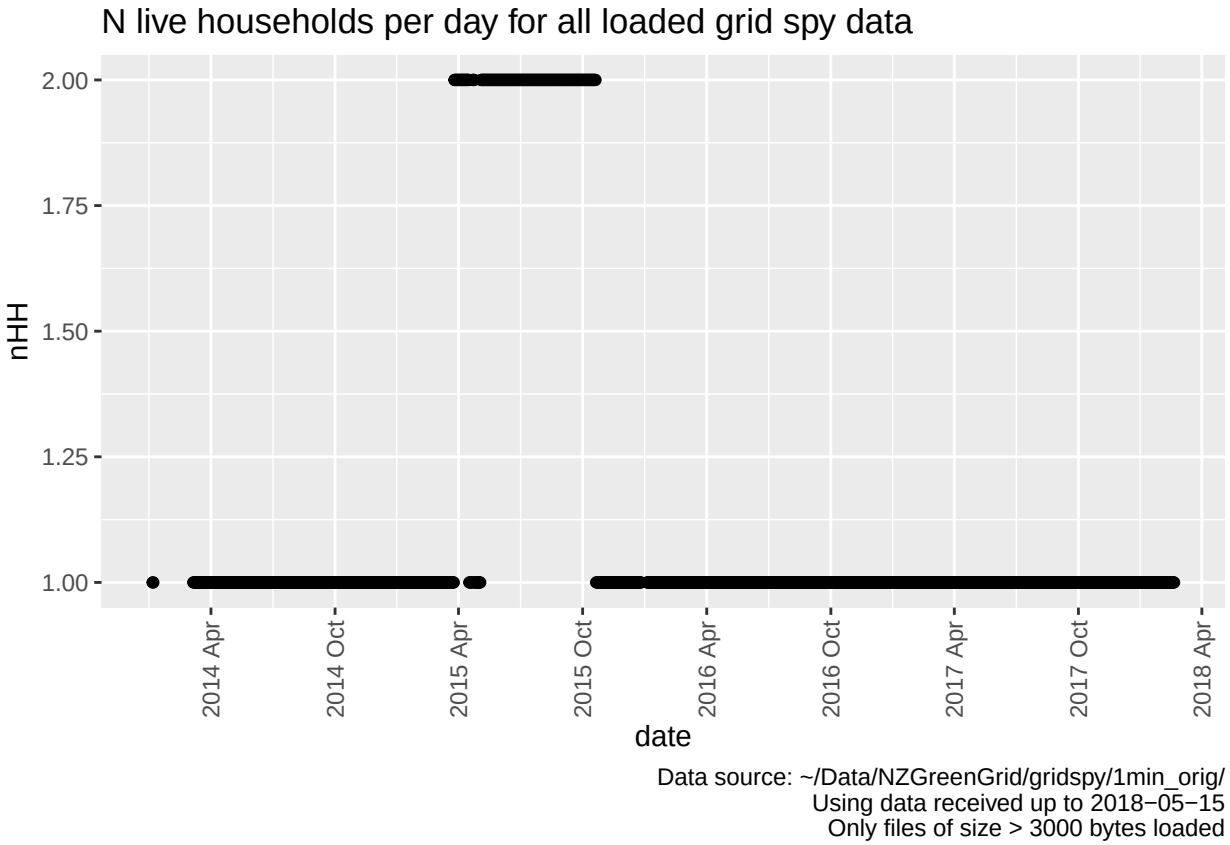
We should also not see NA in any row (indicates date conversion errors).

If we do see any of these then we still have data cleaning work to do!

Table 9: Summary observation stats by hhID

hhID	minObs	maxObs	meanNDataColumns	minDate	maxDate
rf_01	171	1500	6	2014-01-05	2015-10-20
rf_46	305	1500	13	2015-03-26	2018-02-19

Finally we show the total number of households which we think are still sending data.




```
## Saving 6.5 x 4.5 in image
```

7 Runtime

Analysis completed in 262.452 seconds (4.37 minutes) using knitr in RStudio with R version 3.4.4 (2018-03-15) running on x86_64-apple-darwin15.6.0.

8 R environment

R packages used:

- base R - for the basics (R Core Team 2016)
- data.table - for fast (big) data handling (Dowle et al. 2015)
- lubridate - date manipulation (Grolemund and Wickham 2011)
- ggplot2 - for slick graphics (Wickham 2009)
- readr - for csv reading/writing (Wickham, Hester, and Francois 2016)
- dplyr - for select and contains (Wickham and Francois 2016)
- progress - for progress bars (Csárdi and FitzJohn 2016)
- knitr - to create this document & neat tables (Xie 2016)
- kableExtra - for extra neat tables (Zhu 2018)
- nzGREENGrid - for local NZ GREEN Grid project utilities

Session info:

```
## R version 3.4.4 (2018-03-15)
```

```
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.4
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] progress_1.1.2      dplyr_0.7.4         readr_1.1.1
## [4] lubridate_1.7.4     data.table_1.10.4-3 kableExtra_0.8.0
## [7] knitr_1.20          ggplot2_2.2.1.9000  nzGREENGrid_0.1.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.16      highr_0.6         bindr_0.1.1
##  [4] pillar_1.2.2      compiler_3.4.4    plyr_1.8.4
##  [7] prettyunits_1.0.2 tools_3.4.4       digest_0.6.15
## [10] evaluate_0.10.1   tibble_1.4.2      gtable_0.2.0
## [13] viridisLite_0.3.0 pkgconfig_2.0.1   rlang_0.2.0.9001
## [16] rstudioapi_0.7    yaml_2.1.18       bindrcpp_0.2.2
## [19] withr_2.1.2       stringr_1.3.0     httr_1.3.1
## [22] xml2_1.2.0        hms_0.4.2         rprojroot_1.3-2
## [25] grid_3.4.4        glue_1.2.0        R6_2.2.2
## [28] rmarkdown_1.9     magrittr_1.5      backports_1.1.2
## [31] scales_0.5.0.9000 htmltools_0.3.6   assertthat_0.2.0
## [34] rvest_0.3.2       colorspace_1.3-2  labeling_0.3
## [37] stringi_1.1.7     lazyeval_0.2.1    munsell_0.4.3
```

References

Csárdi, Gábor, and Rich FitzJohn. 2016. *Progress: Terminal Progress Bars*. https://CRAN.R-project.org/package=progress.

Dowle, M, A Srinivasan, T Short, S Lianoglou with contributions from R Saporta, and E Antonyan. 2015. *Data.table: Extension of Data.frame*. https://CRAN.R-project.org/package=data.table.

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. http://www.jstatsoft.org/v40/i03/.

R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. http://ggplot2.org.

Wickham, Hadley, and Romain Francois. 2016. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, Jim Hester, and Romain Francois. 2016. *Readr: Read Tabular Data*. https://CRAN.

R-project.org/package=readr.

Xie, Yihui. 2016. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://CRAN.R-project.org/package=knitr.

Zhu, Hao. 2018. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.