

NZ GREEN Grid Household Power Demand Profiles: Heat Pump (2015-04-01 to 2016-03-31)

Data extraction and preliminary plots

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

Last run at: 2018-07-30 11:15:19

Contents

1 Status	2
2 Citation	2
3 Introduction	3
3.1 Purpose	3
3.2 Requirements:	3
3.3 History	3
3.4 Support	3
4 Load data files	3
4.1 Grid Spy metadata	3
4.2 Grid Spy data	4
4.3 Test Heat Pump data	4
5 Heat Pump profiles	9
5.1 Profile plots: means per household	9
5.2 Profile plots: overall household mean	11
6 Runtime	14
7 R environment	14
References	15

1 Status

Full run using all data from /Volumes/hum-csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/

2 Citation

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

- Anderson, B. (2018) GREEN Grid Heat Pump Profiles, 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 a data file containing pre-extracted Heat Pump circuits (via string match on their circuit label) for the period 2015-04-01 to 2016-03-31 (inclusive);
- build exploratory demand profiles for Heat Pump circuits.

3.2 Requirements:

- Cleaned and safe grid spy 1 minute data processed using the nzGREENGridDataR package
- Heat Pump circuit data pre or newly extracted from the cleaned safe data using <https://git.soton.ac.uk/ba1e12/nzGREENGrid/blob/master/dataProcessing/gridSpy/extractCleanGridSpy1minCircuits.R>

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 University of Southampton;
- The New Zealand Ministry of Business, Innovation and Employment (MBIE) through the NZ GREEN Grid project;
- 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 Load data files

4.1 Grid Spy metadata

In this section we load metadata from /Users/ben/Syncplicity Folders/Green Grid Project Management Folder/Gridspy/Master list of Gridspy units.xlsx to link to the power data.

sample	hhID	Adults	Teenagers	Children	removed	nAdults
Powerco	rf_06	2	NA	NA	NA	2

sample	hhID	Adults	Teenagers	Children	removed	nAdults
Powerco	rf_07	2	NA	2	NA	2
Powerco	rf_08	2	NA	NA	NA	2
Powerco	rf_09	2	NA	1	42171	2
Powerco	rf_10	2	NA	1(3yo)	NA	3
Powerco	rf_11	NA	NA	NA	NA	1

4.2 Grid Spy data

In this section we load the pre-extracted Heat Pump data from 2015-04-01 to 2016-03-31 from /Volumes/humcsafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/dataExtracts/Heat Pump_2015-04-01_2016-03-31_observations.csv.gz.

```
## Warning: The following named parsers don't match the column names:
## circuitLabel, circuitID
```

The following table summarises the Heat Pump data we have found.

hhID	r_dateTime	circuit	powerW
Length:14250284	Min. :2015-04-01 00:00:00	Length:14250284	Min. :-655.00
Class :character	1st Qu.:2015-06-22 12:39:00	Class :character	1st Qu.: 0.00
Mode :character	Median :2015-09-16 13:12:00	Mode :character	Median : 0.00
NA	Mean :2015-09-21 08:00:39	NA	Mean : 147.92
NA	3rd Qu.:2015-12-17 17:52:00	NA	3rd Qu.: 61.29
NA	Max. :2016-03-31 23:59:00	NA	Max. :27759.00

This table will have a large number (14,250,284) of observations caused by the number of different circuit labels as shown by the following table.

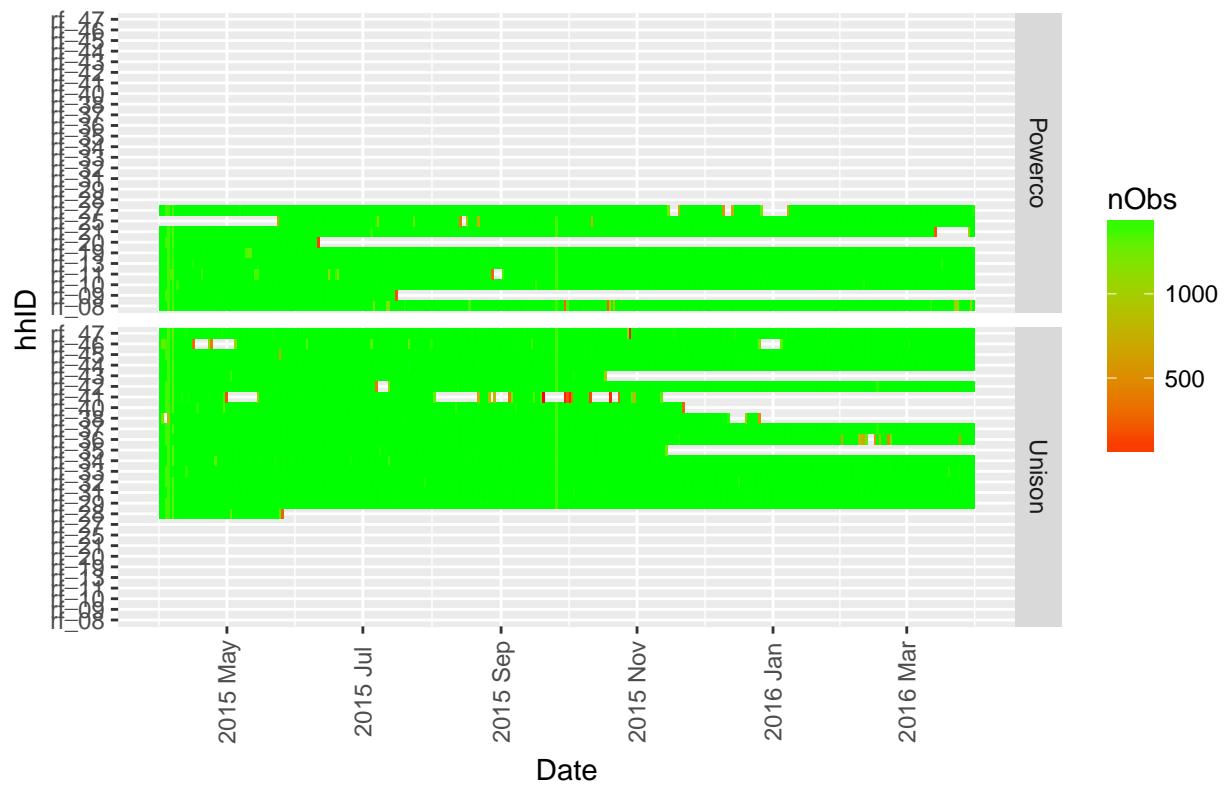
Freq

Note that some households may have more than one Heat Pump circuit.

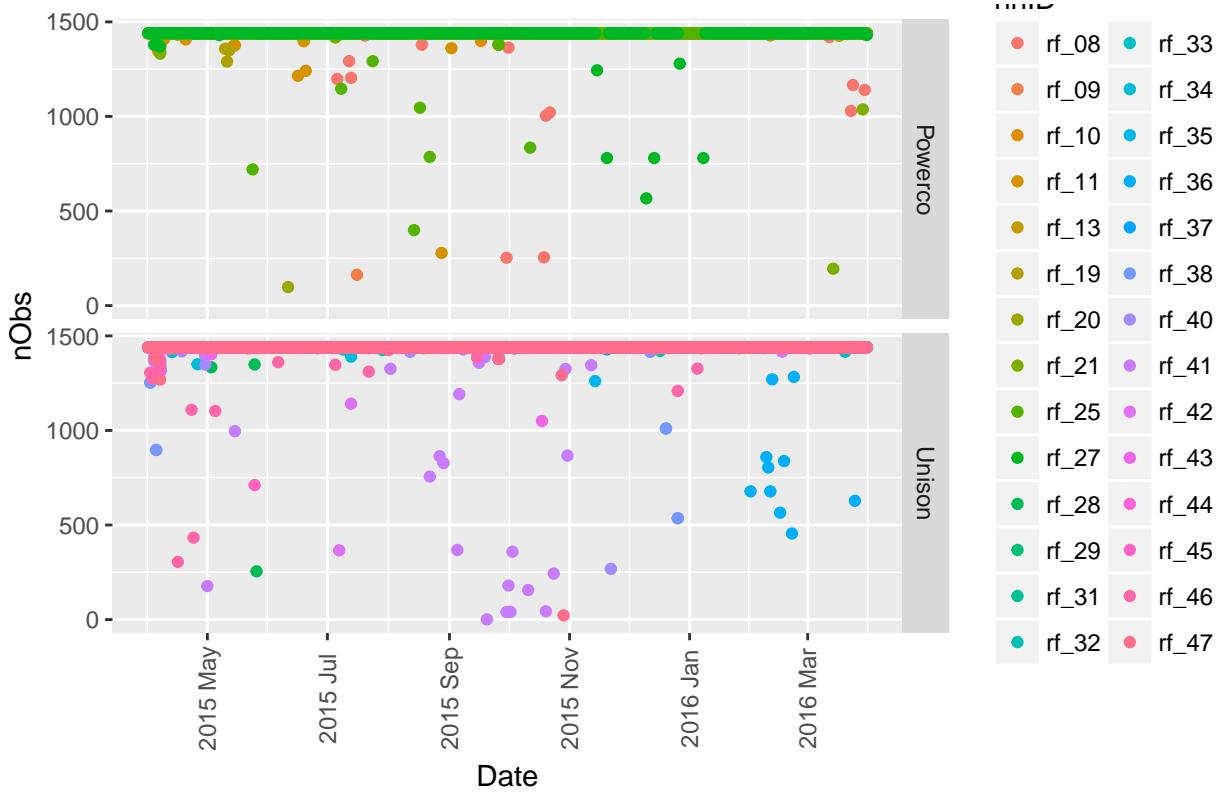
4.3 Test Heat Pump data

This section tests the availability of Heat Pump data by replicating the standard data quality checks used for the whole gridSpy dataset.

The following plot shows loaded data observation plots - just to confirm what Heat Pump data we have.



The next plot shows the same data but as a dot plot to highlight those households and dates where we did not receive $60 * 24 = 1440$ observations per day.



umes/hum-csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/
Heat Pump (2015-04-01 to 2016-03-31)

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

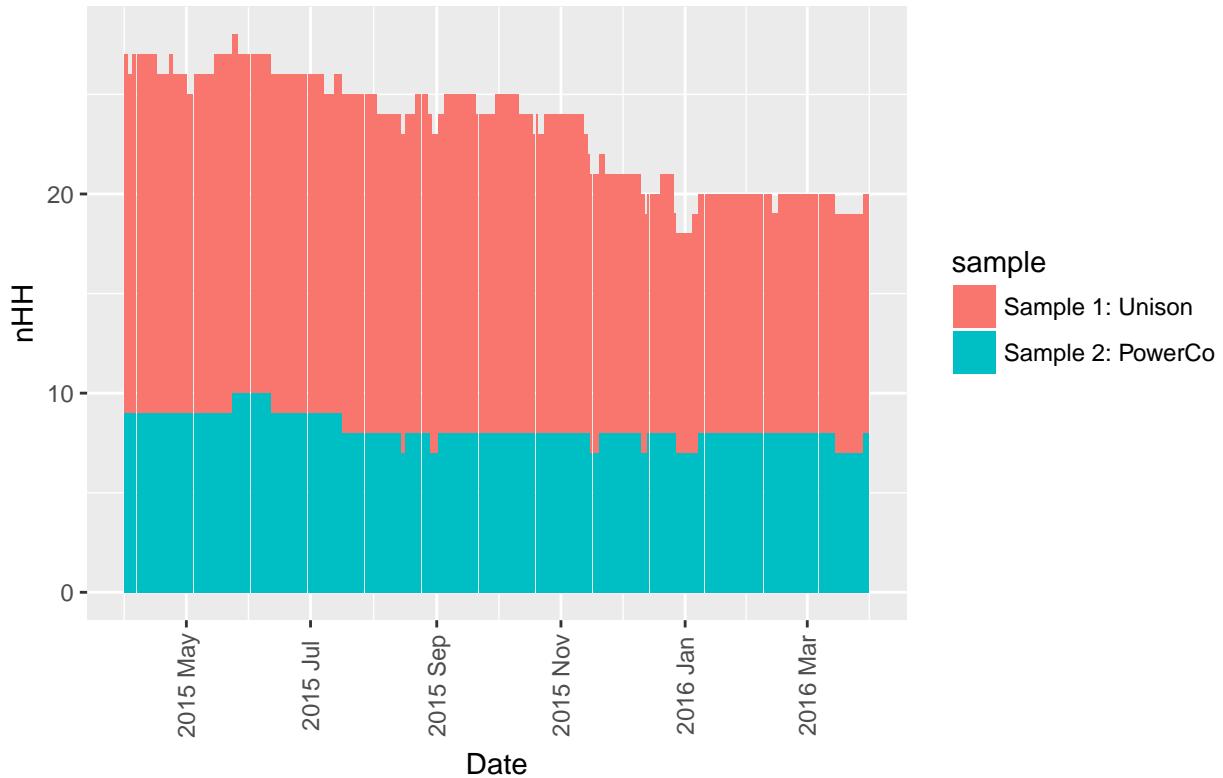
We should also not see NA in any row (indicates date conversion errors).

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

hhID	sample	nObs	minDate	maxDate
rf_28	Unison	79033	2015-04-01 00:00:00	2015-05-26 04:56:00
rf_20	Powerco	102168	2015-04-01 00:00:00	2015-06-11 01:37:00
rf_09	Powerco	152664	2015-04-01 00:00:00	2015-07-16 02:42:00
rf_43	Unison	288814	2015-04-01 00:00:00	2015-10-18 17:29:00
rf_41	Unison	223790	2015-04-01 00:00:00	2015-11-12 22:24:00
rf_35	Unison	327860	2015-04-01 00:00:00	2015-11-14 21:00:00
rf_40	Unison	338280	2015-04-01 00:00:00	2015-11-22 04:27:00
rf_38	Unison	747324	2015-04-01 00:00:00	2015-12-26 08:55:00
rf_08	Powerco	521724	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_10	Powerco	526678	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_11	Powerco	519127	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_13	Powerco	1053820	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_19	Powerco	1052976	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_21	Powerco	505028	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_25	Powerco	443888	2015-05-24 12:00:00	2016-03-31 23:59:00
rf_27	Powerco	497686	2015-04-01 00:00:00	2016-03-31 23:59:00

hhID	sample	nObs	minDate	maxDate
rf_29	Unison	526778	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_31	Unison	526802	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_32	Unison	526665	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_33	Unison	526863	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_34	Unison	526557	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_36	Unison	516127	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_37	Unison	526651	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_42	Unison	518064	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_44	Unison	526737	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_45	Unison	525994	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_46	Unison	950927	2015-04-01 00:00:00	2016-03-31 23:59:00
rf_47	Unison	525011	2015-04-01 00:00:00	2016-03-31 23:59:00

Finally we show the total number of households which we think we have Heat Pump data for.



umes/hum-csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/
Heat Pump (2015-04-01 to 2016-03-31)

The following table summarises the Heat Pump data. Any surprises?

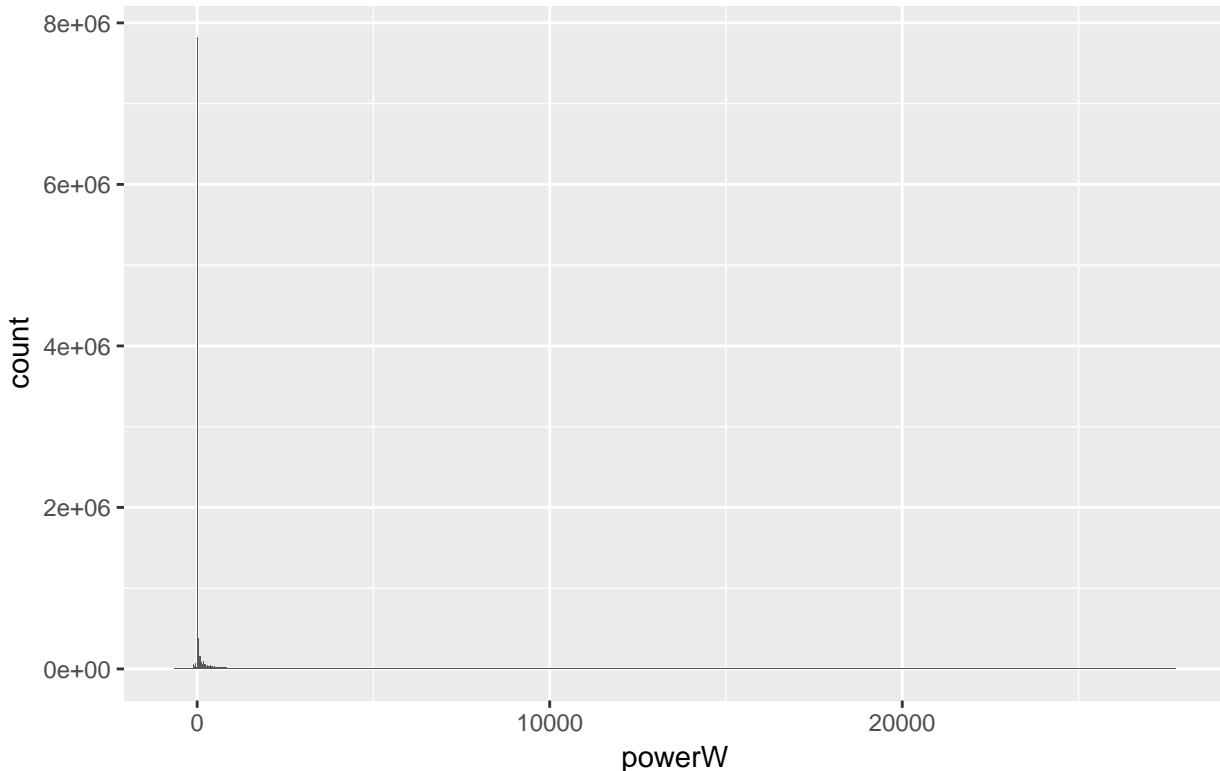
```
t <- summary(gs1MinDT)
knitr::kable(caption = paste0("Summary of ", circuitPattern, " circuits"), t)
```

hhID	r_dateTime	circuit	powerW
Length:14250284	Min. :2015-04-01 00:00:00	Length:14250284	Min. : -655.00
Class :character	1st Qu.:2015-06-22 12:39:00	Class :character	1st Qu.: 0.00
Mode :character	Median :2015-09-16 13:12:00	Mode :character	Median : 0.00
NA	Mean :2015-09-21 08:00:39	NA	Mean : 147.92

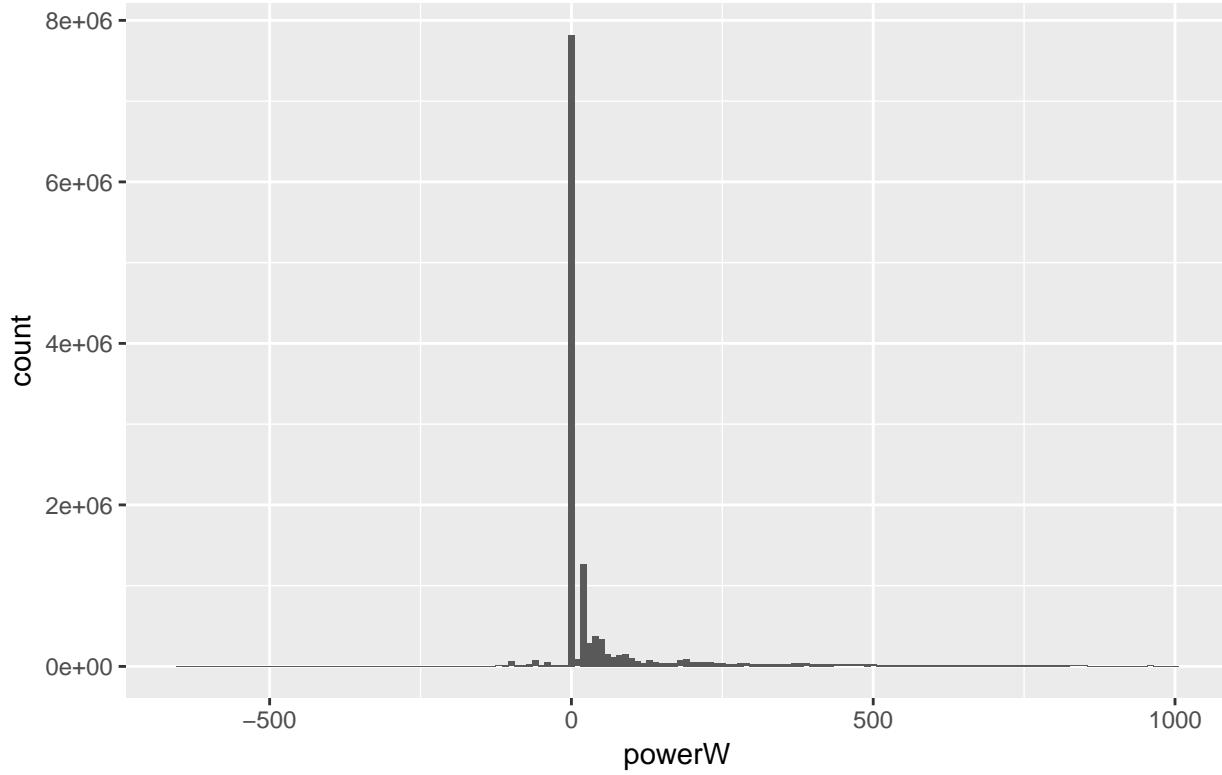
hhID	r_dateTime	circuit	powerW
NA	3rd Qu.:2015-12-17 17:52:00	NA	3rd Qu.: 61.29
NA	Max. :2016-03-31 23:59:00	NA	Max. :27759.00

We seem to have some negative powerW values and at least one very large power value.

Nasty surprises often lurk in histograms... The following histogram shows all observations.



The next shows the histogram for powerW < 1000W...



There are a lot of zeros (as we'd expect) but why are there negative values?

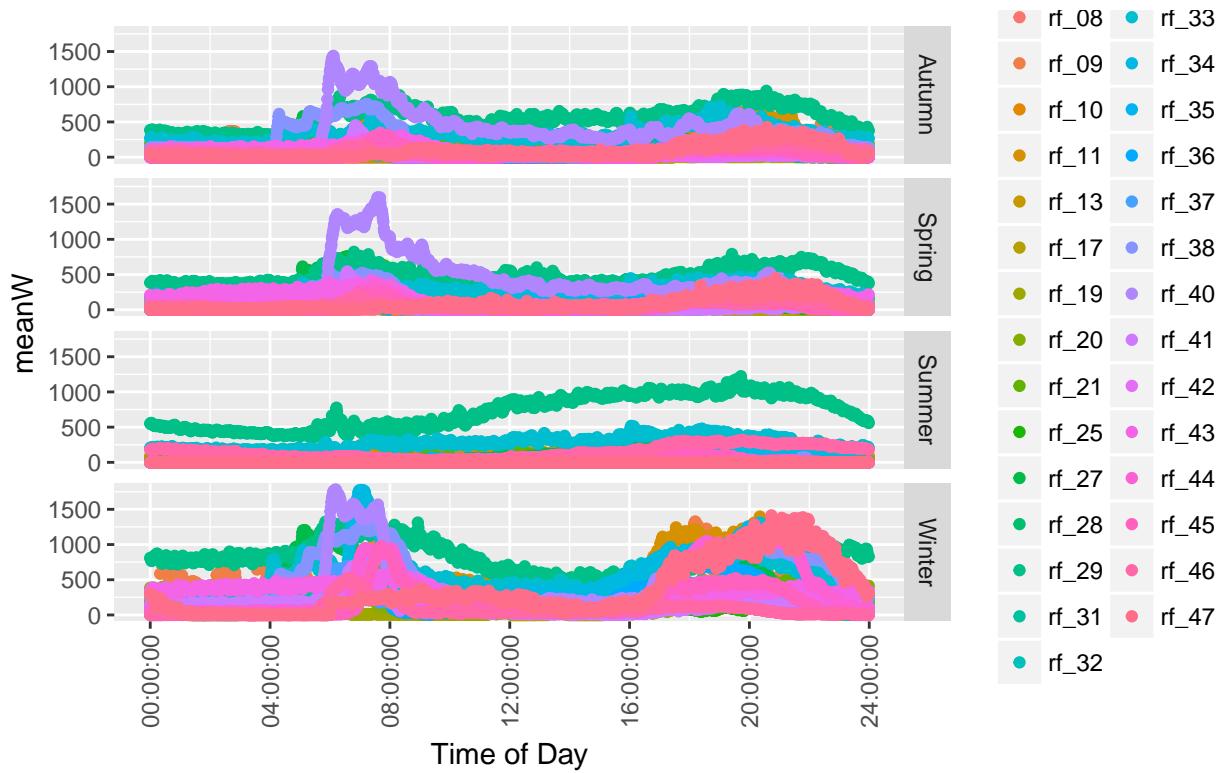
5 Heat Pump profiles

This section produces the profiles as one for each HH but averaged over each season. Data is kept at 1 minute intervals. Note definition of season below...

```
# add season
gs1MinDT <- gs1MinDT[, month := lubridate::month(r_dateTime, label = TRUE)]
gs1MinDT <- gs1MinDT[, season := "Summer"]
gs1MinDT <- gs1MinDT[, season := ifelse(month == "Mar" |
                                         month == "Apr" |
                                         month == "May", "Autumn", season)]
gs1MinDT <- gs1MinDT[, season := ifelse(month == "Jun" |
                                         month == "Jul" |
                                         month == "Aug", "Winter", season)]
gs1MinDT <- gs1MinDT[, season := ifelse(month == "Sep" |
                                         month == "Oct" |
                                         month == "Nov", "Spring", season)]
```

5.1 Profile plots: means per household

This section shows a plot of mean profiles for each household by season.



umes/hum-csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/
Heat Pump (2015-04-01 to 2016-03-31)
n households = 29

```

## [1] "Saving plot to /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/prof...
## [1] "Saving profile data used to build this plot to: /Volumes/hum-csafe/Research Projects/GREEN Grid...
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/profiles/H...

```

hhID	obsHourMin	season	meanW
Length:151200	Length:151200	Length:151200	Min. : -2.422
Class :character	Class1:hms	Class :character	1st Qu.: 12.137
Mode :character	Class2:difftime	Mode :character	Median : 57.750
NA	Mode :numeric	NA	Mean : 149.360
NA	NA	NA	3rd Qu.: 195.743
NA	NA	NA	Max. :1778.669

As we can see there is considerable variation between households in both the level and timing of heat pump demand.

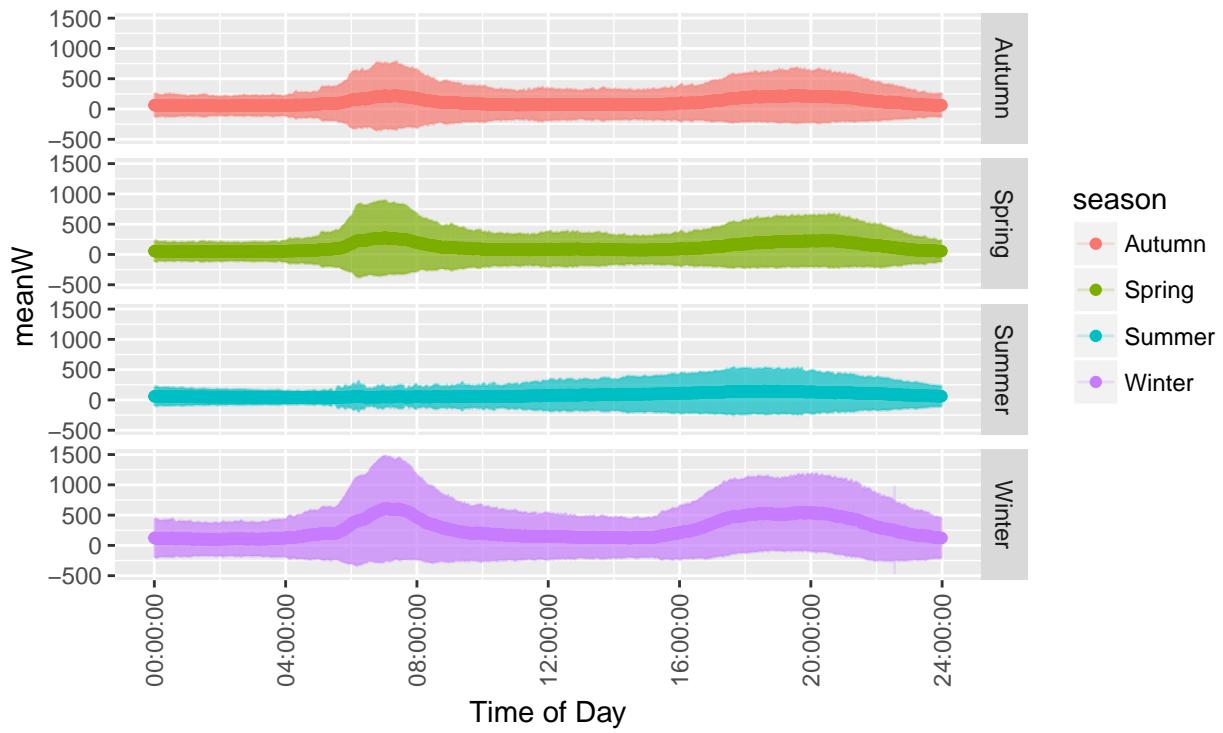
Note that the code saves a high definition version of the plot and the profiles for future re-use.

The .csv.gz file can be loaded using the following code:

- df <- readr::read_csv("/Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/Heat Pump_2015-04-01_2016-03-31_byHouseholdSeasonalProfiles.csv.gz") or
- dt <- data.table::as.data.table(readr::read_csv("/Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/profiles/Heat Pump_2015-04-01_2016-03-31_byHouseholdSeasonalProfiles.csv.gz")) if you prefer data.table

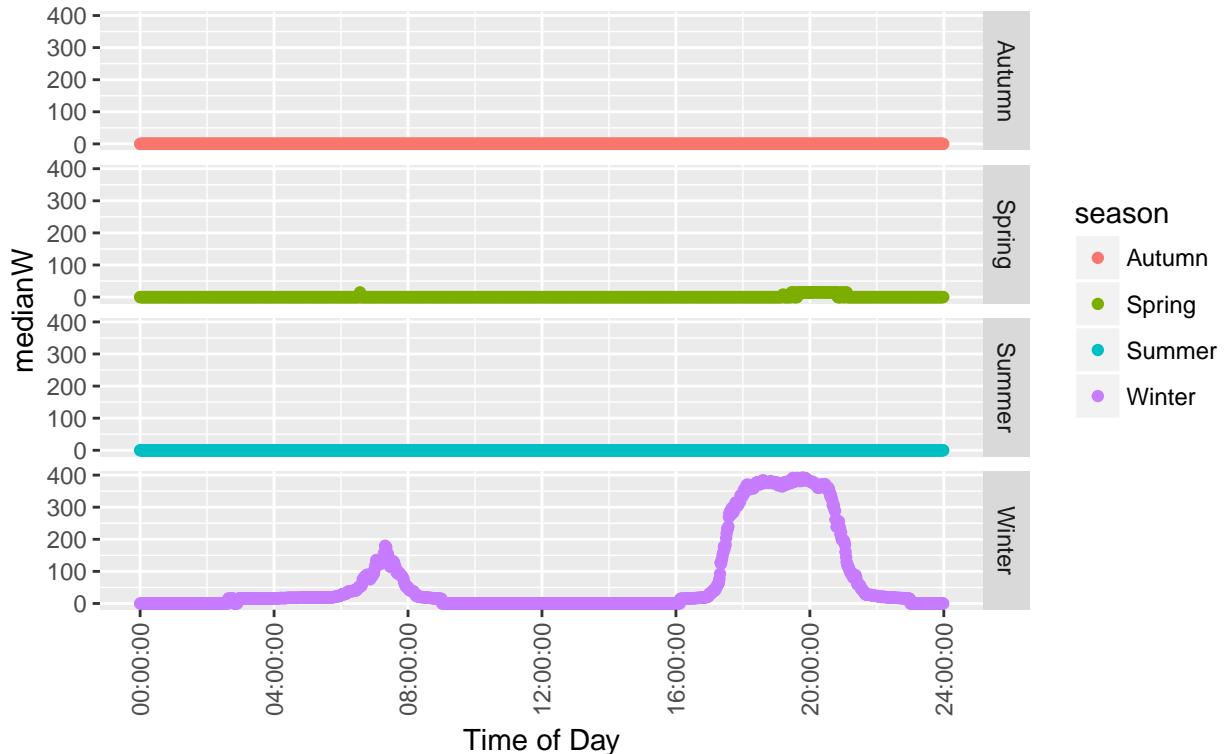
5.2 Profile plots: overall household mean

This section shows a plot of mean and median profiles across all household by season. The mean profile also shows the level of variance by plotting error bars at +/- 1 s.d.



```
e: /Volumes/hum-csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/
Heat Pump (2015-04-01 to 2016-03-31)
n households = 29
Error bars = +/- 1 s.d.
```

```
## [1] "Saving plot to /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/proj...
```



```
#e: /Volumes/hum-csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/
Heat Pump (2015-04-01 to 2016-03-31)
n households = 29
```

```
## [1] "Saving plot to /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/prof...
## [1] "Saving profile data used to build this plot to: /Volumes/hum-csafe/Research Projects/GREEN Grid...
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/profiles/H...
```

obsHourMin	season	meanW	medianW	nObs	sdW
Length:5760	Length:5760	Min. : 34.99	Min. : 0.00	Min. :2150	Min. :101.0
Class1:hms	Class :character	1st Qu.: 69.26	1st Qu.: 0.00	1st	1st Qu.:224.7
				Qu.:2401	
Class2:difftime	Mode :character	Median :104.83	Median : 0.00	Median :2517	Median :296.2
Mode :numeric	NA	Mean :143.53	Mean : 17.05	Mean :2474	Mean :329.1
NA	NA	3rd Qu.:174.66	3rd Qu.: 0.00	3rd	3rd Qu.:404.7
NA	NA	Max. :613.89	Max. :392.55	Max. :2688	Max. :879.1

The difference between the mean and median plots is instructive - it suggests that the mean plots for summer are skewed by a few higher heat pump-using households.

The plots could be repeated or re-factored e.g. by household size.

As before, the code saves a high definition version of the plot and the profiles for future re-use.

6 Runtime

Analysis completed in 128.14 seconds (2.14 minutes) using knitr in RStudio with R version 3.5.0 (2018-04-23) running on x86_64-apple-darwin15.6.0.

7 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 (Gromelund 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)
- nzGREENGrid - for local NZ GREEN Grid project utilities

Session info:

```
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
## Matrix products: default
## BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/lib/libBLAS.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/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] kableExtra_0.9.0      dplyr_0.7.5          rmarkdown_1.10
## [4] plotly_4.7.1         nzGREENGrid_0.1.0    knitr_1.20
## [7] readr_1.1.1          ggplot2_2.2.1        lubridate_1.7.4
## [10] data.table_1.11.4     nzGREENGridDataR_0.1.0
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.17      lattice_0.20-35    tidyverse_1.2.1
## [4] prettyunits_1.0.2  png_0.1-7       assertthat_0.2.0
## [7] rprojroot_1.3-2    digest_0.6.15     mime_0.5
## [10] R6_2.2.2         cellranger_1.1.0   plyr_1.8.4
## [13] backports_1.1.2    evaluate_0.10.1   httr_1.3.1
## [16] highr_0.7        pillar_1.2.3     RgoogleMaps_1.4.2
## [19] rlang_0.2.1       progress_1.2.0    lazyeval_0.2.1
## [22] readxl_1.1.0      rstudioapi_0.7    geosphere_1.5-7
## [25] proto_1.0.0       labeling_0.3     stringr_1.3.1
## [28] htmlwidgets_1.2    munspell_0.5.0   shiny_1.1.0
## [31] compiler_3.5.0     httpuv_1.4.3     pkgconfig_2.0.1
## [34] htmltools_0.3.6    openssl_1.0.1    tidyselect_0.2.4
```

```

## [37] tibble_1.4.2      viridisLite_0.3.0 crayon_1.3.4
## [40] later_0.7.3       grid_3.5.0        jsonlite_1.5
## [43] xtable_1.8-2      gtable_0.2.0      magrittr_1.5
## [46] scales_0.5.0      stringi_1.2.3    mapproj_1.2.6
## [49] reshape2_1.4.3     promises_1.0.1   bindrcpp_0.2.2
## [52] sp_1.3-1          xml2_1.2.0       rjson_0.2.20
## [55] tools_3.5.0        ggmap_2.6.1       glue_1.2.0
## [58] purrrr_0.2.5      maps_3.3.0       hms_0.4.2
## [61] crosstalk_1.0.0    jpeg_0.1-8       yaml_2.1.19
## [64] colorspace_1.3-2   rvest_0.3.2      bindr_0.1.1

```

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 Antoneyan. 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>.