NZ GREEN Grid Household Power Demand Profiles: Hot Water

Data extraction and preliminary plots

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

Last run at: 2018-06-08 16:01:15

Contents

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 and clean the project electricity power data (Grid Spy)
- select the Heat Pump circuits (via their labels)
- build exploratory demand profiles

3.2 Requirements:

- cleaned and safe grid spy 1 minute data processed via https://git.soton.ac.uk/ba1e12/nzGREENGrid/blob/master/dataProcessing/processNZGGElecCons1minData.Rmd

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 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
Powerco	rf_07	2	NA	2	NA	2

sample	hhID	Adults	Teenagers	Children	removed	nAdults
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 cleaned data files from /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/data/. If we loaded all the data at once and then filtered out what we want we might run out of memory so we filter as we load. Set the filters here:

```
circuitPattern <- params$circuitPattern
dateFrom <- params$dateFrom
dateTo <- params$dateTo

plotCaption <- paste0("Source: ", fpath,
                      "\nCircuits: ", circuitPattern, " from ", dateFrom, " to ", dateTo)
```

So we are looking for Hot Water circuits between 2015-04-01 and 2016-03-31. We do this by checking to see if the extract file has already been created. If so we load it. If not, we create it.

The file we are looking for is: Hot Water_2015-04-01_2016-03-31_observations.csv

```
## [1] "/Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/dataExtracts/Hot Wa
## [1] "# Loaded 14,496,831 rows of data"
```

The following table summarises the Hot Water data we have found.

	hhID	r_dateTime	circuit	powerW	obsHourMin
	Length:14496831	Length:14496831	Length:14496831	Min. :-1110.0	Length:14496831
	Class :character	Class :character	Class :character	1st Qu.: 0.0	Class :character
	Mode :character	Mode :character	Mode :character	Median : 0.0	Mode :character
	NA	NA	NA	Mean : 283.8	NA
	NA	NA	NA	3rd Qu.: 0.0	NA
	NA	NA	NA	Max. : 4076.0	NA

This table will have a large number (14,496,831) of obserations caused by the number of different circuit labels as shown by the following table.

Var1	Freq
Hot Water (2 elements)$4247	495806
Hot Water - Controlled (HEMS)$2081	519906
Hot Water - Controlled$2094	521843
Hot Water - Controlled$2102	520556
Hot Water - Controlled$2110	102188
Hot Water - Controlled$2129	102202
Hot Water - Controlled$2150	520028
Hot Water - Controlled$2208	526929
Hot Water - Controlled$2236	526097
Hot Water - Controlled$2248	523312
Hot Water - Controlled$2719	525470

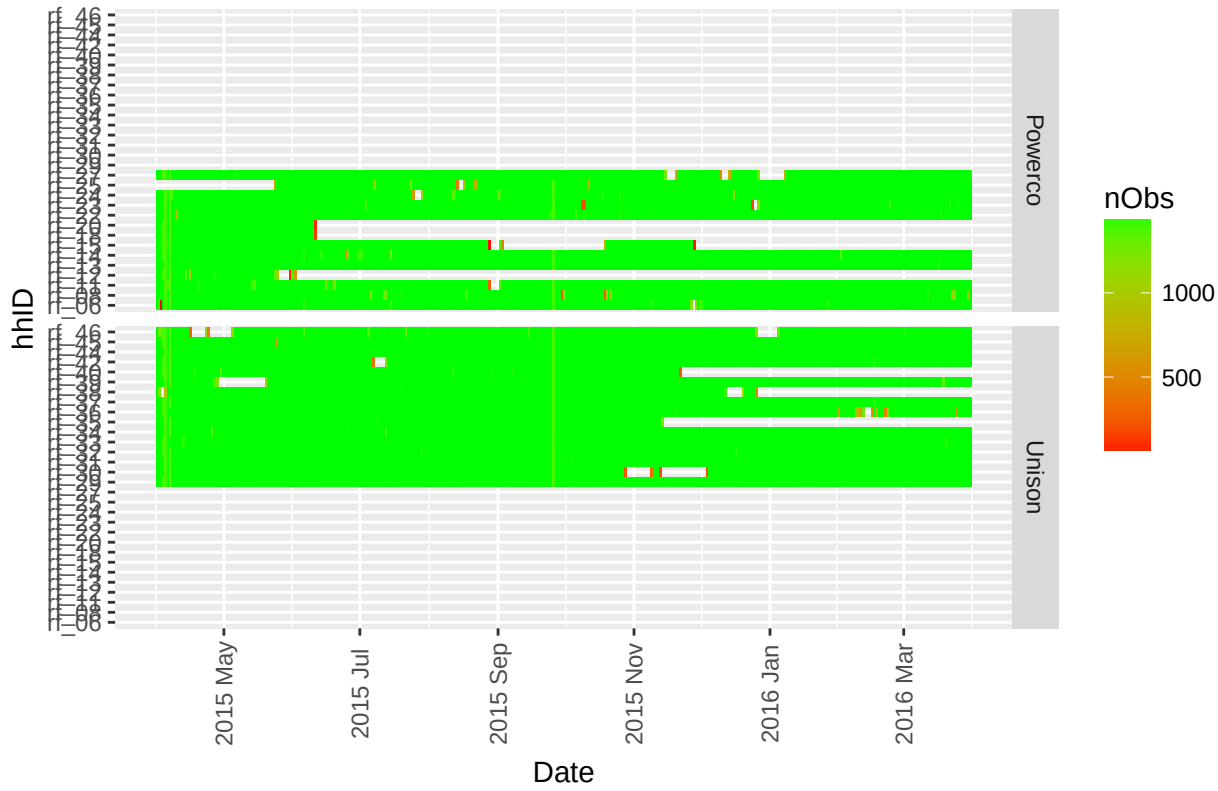
Var1	Freq
Hot Water - Controlled$2761	443936
Hot Water - Controlled$2825	497806
Hot Water - Controlled$4135	526771
Hot Water - Controlled$4144	526863
Hot Water - Controlled$4155	526850
Hot Water - Controlled$4158	526110
Hot Water - Controlled$4167	338289
Hot Water - Controlled$4178	373722
Hot Water - Controlled$4184	526780
Hot Water - Controlled$4198	526785
Hot Water - Controlled$4200	526878
Hot Water - Controlled$4231	486982
Hot Water - Controlled$4238	477491
Hot Water - Controlled$4400	463994
Hot Water - Uncontrolled$4125	327974
Hot Water - Uncontrolled$4131	518179
Hot Water - Uncontrolled$4147	516242
Hot Water - Uncontrolled$4224	526677
Hot Water Cpbd Heater- Cont$2586	519185
Hot Water$1574	81647
Hot Water$3952	273252
Incomer 1 - Hot Water - Cont$2626	80081

Note that some households may have more than one Hot Water circuit.

4.3 Test Hot Water data

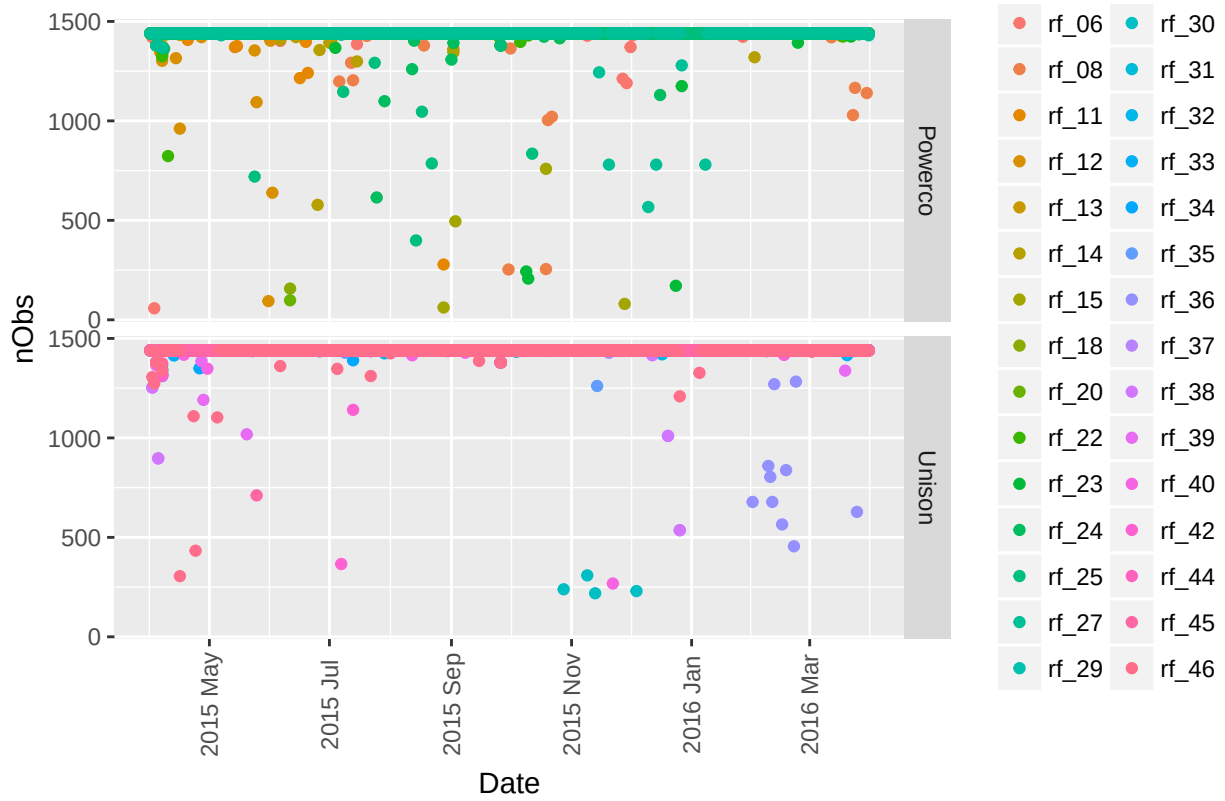
This section tests the availability of Hot Water data by replicating the standard data quality checks used for the whole gridSpy dataset.

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



ɹrce: /Volumes/hum−csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/
Circuits: Hot Water from 2015−04−01 to 2016−03−31

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/
Circuits: Hot Water from 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

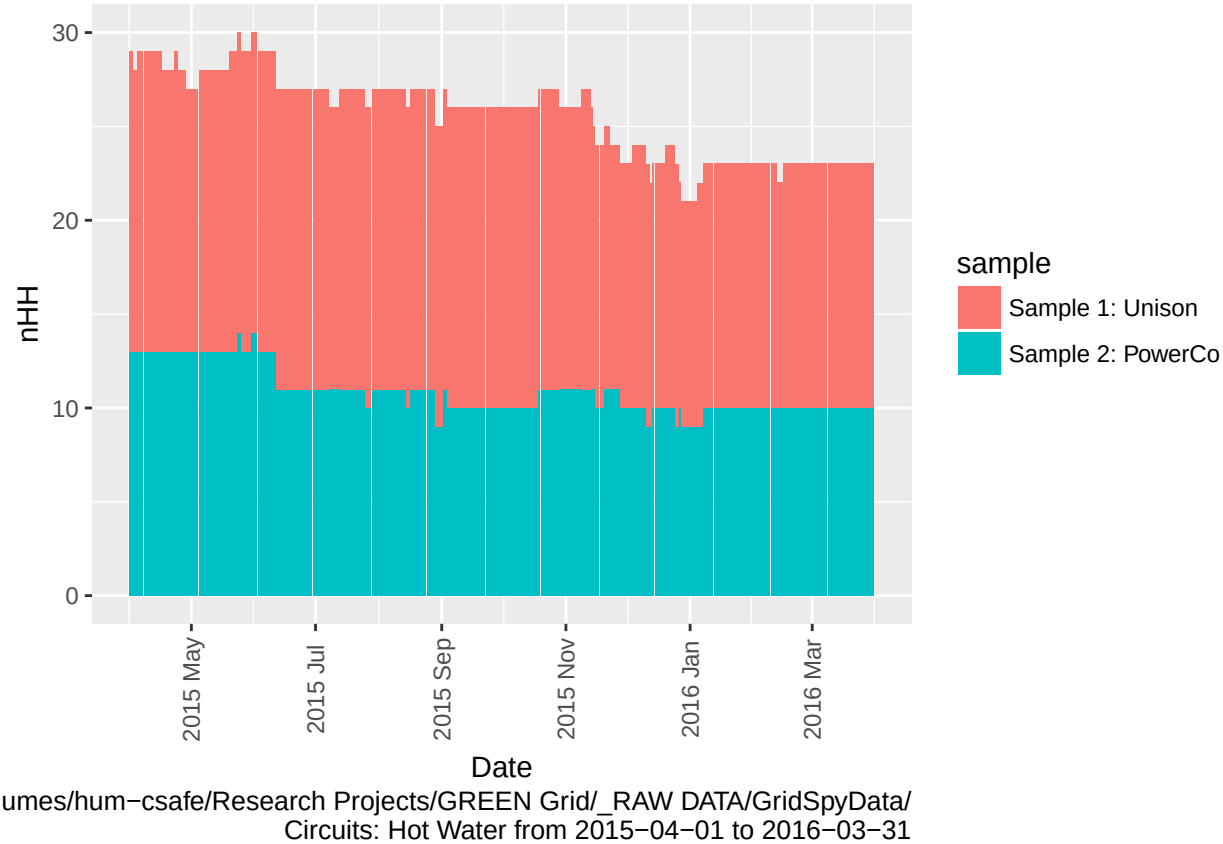
We should also not see NA in any row (indicates date conversion errors).

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

hhID	sample	nObs	minDate	maxDate
rf_12	Powerco	80081	2015-04-01T00:00:00Z	2015-06-02T20:07:00Z
rf_20	Powerco	102188	2015-04-01T00:00:00Z	2015-06-11T01:37:00Z
rf_18	Powerco	102202	2015-04-01T00:00:00Z	2015-06-11T02:36:00Z
rf_35	Unison	327974	2015-04-01T00:00:00Z	2015-11-14T21:00:00Z
rf_40	Unison	338289	2015-04-01T00:00:00Z	2015-11-22T04:27:00Z
rf_15	Powerco	273252	2015-04-01T00:00:00Z	2015-11-28T01:19:00Z
rf_38	Unison	747444	2015-04-01T00:00:00Z	2015-12-26T08:55:00Z
rf_06	Powerco	523312	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_08	Powerco	521843	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_11	Powerco	519185	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_13	Powerco	526929	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_14	Powerco	525470	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_22	Powerco	526097	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_23	Powerco	519906	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_24	Powerco	520556	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z

hhID	sample	nObs	minDate	maxDate
rf_25	Powerco	443936	2015-05-24T12:00:00Z	2016-03-31T23:59:00Z
rf_27	Powerco	497806	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_29	Unison	526780	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_30	Unison	477491	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_31	Unison	526878	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_32	Unison	526785	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_33	Unison	526863	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_34	Unison	526677	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_36	Unison	516242	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_37	Unison	526771	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_39	Unison	495806	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_42	Unison	518179	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_44	Unison	526850	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_45	Unison	526110	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z
rf_46	Unison	950976	2015-04-01T00:00:00Z	2016-03-31T23:59:00Z

Finally we show the total number of households which we think we have Hot Water data for.



The following table summarises the Hot Water data. Any surprises?

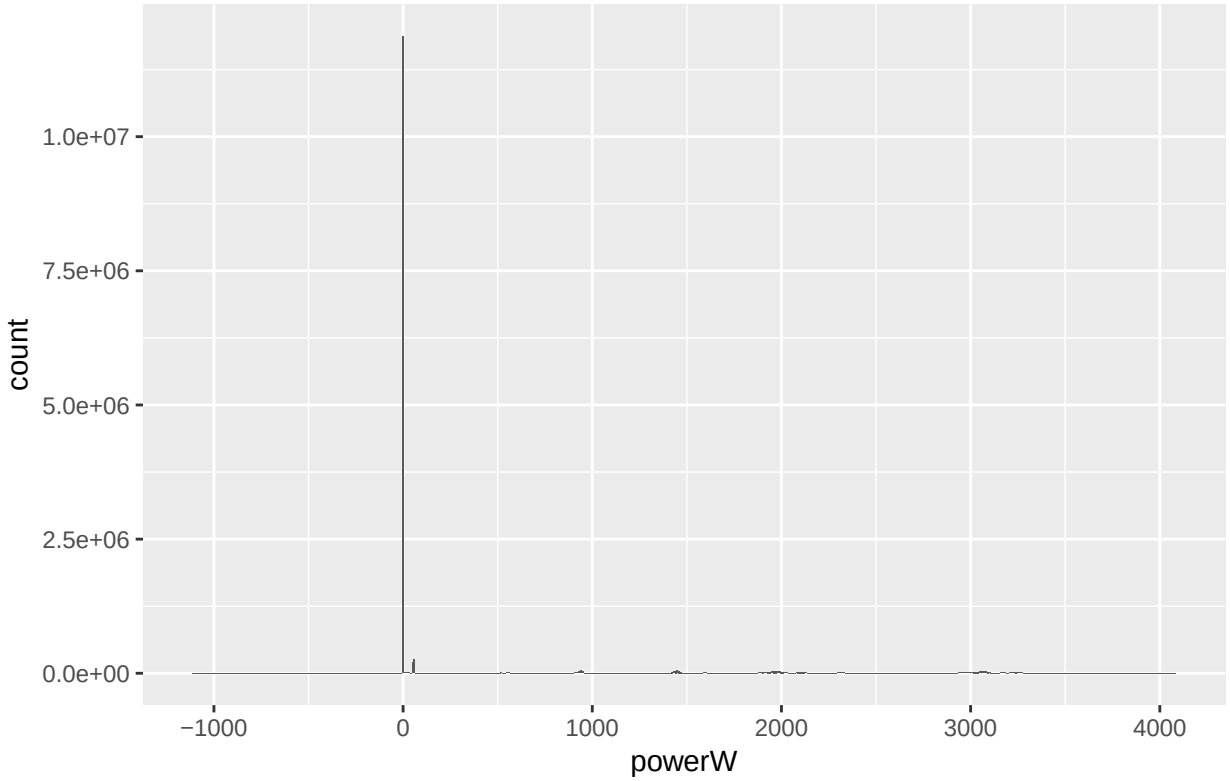
```
t <- summary(gs1MinDT)
knitr::kable(caption = paste0("Summary of ", circuitPattern, " circuits"), t)
```

	hhID	r_dateTime	circuit	powerW	obsHourMin
	Length:14496831	Length:14496831	Length:14496831	Min. :-1110.0	Length:14496831

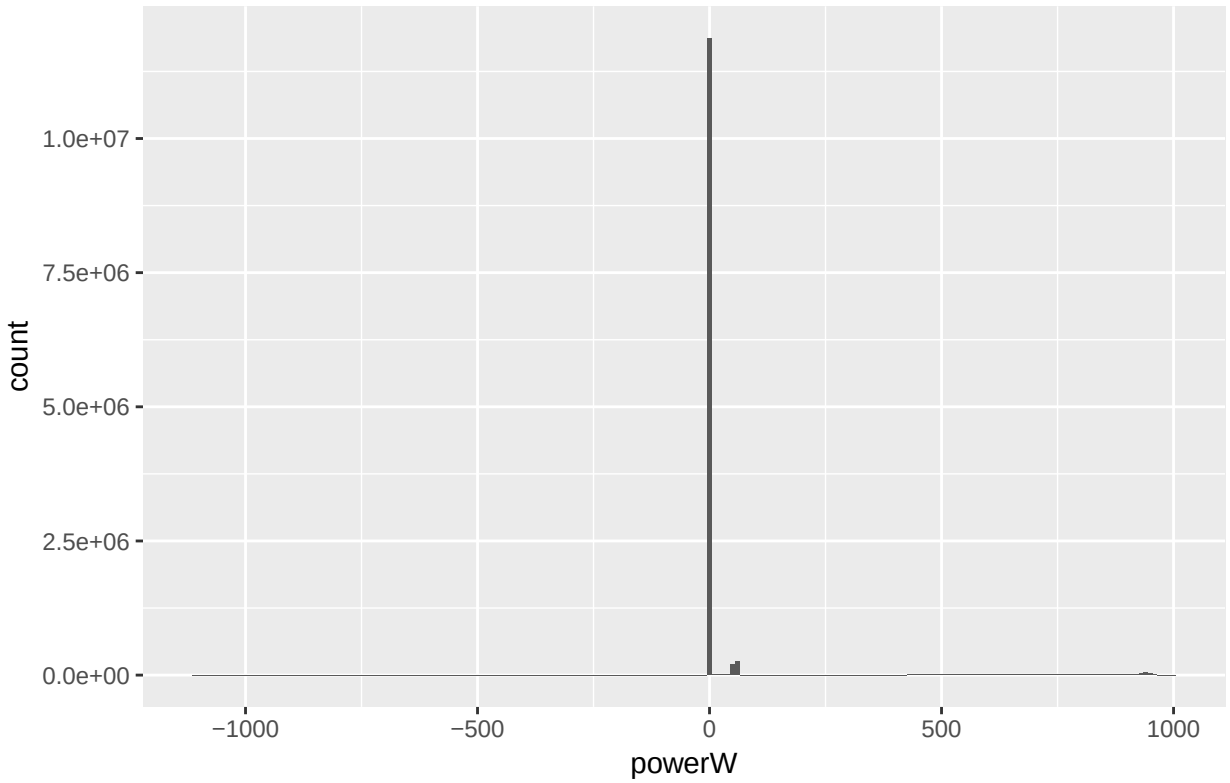
hhID	r_dateTime	circuit	powerW	obsHourMin
Class :character	Class :character	Class :character	1st Qu.: 0.0	Class :character
Mode :character	Mode :character	Mode :character	Median : 0.0	Mode :character
NA	NA	NA	Mean : 283.8	NA
NA	NA	NA	3rd Qu.: 0.0	NA
NA	NA	NA	Max. : 4076.0	NA

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?

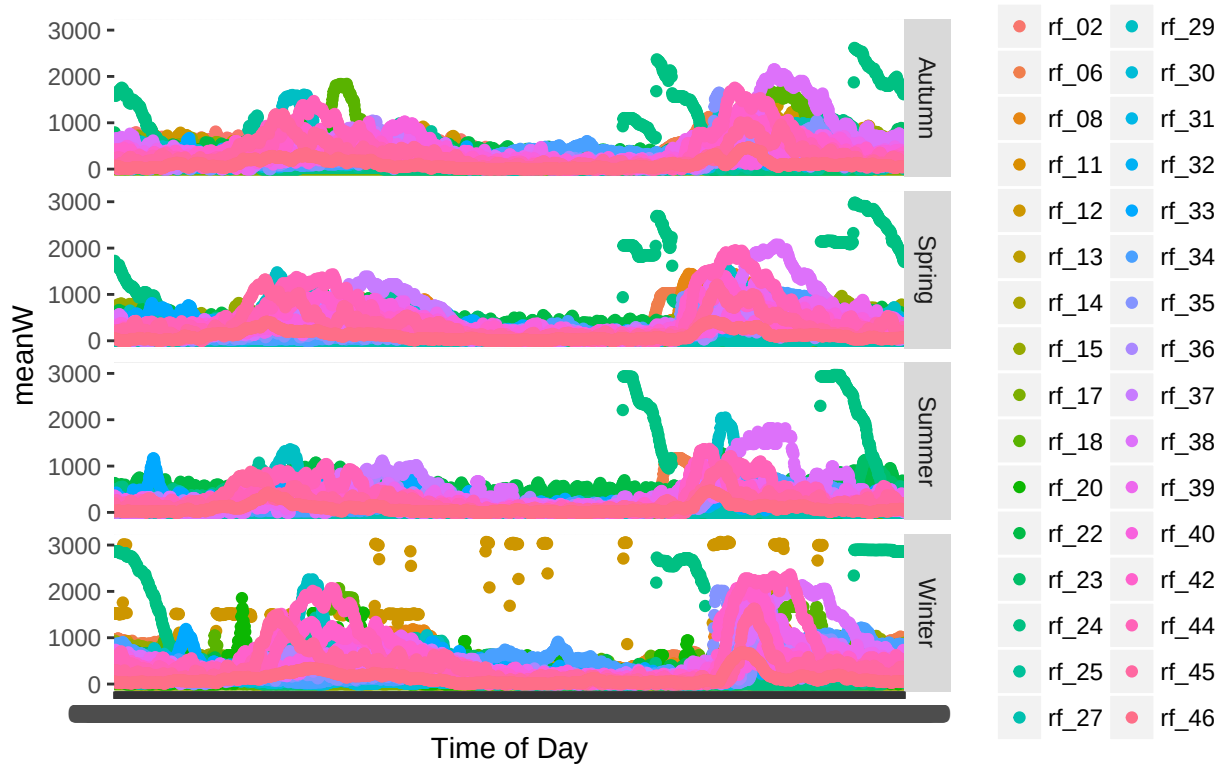
5 Hot Water profiles

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/
Circuits: Hot Water from 2015−04−01 to 2016−03−31
n households = 32

```
## [1] "Saving plot to /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/pro
## [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:167011	Length:167011	Length:167011	Min. : -8.417
Class :character	Class :character	Class :character	1st Qu.: 70.890
Mode :character	Mode :character	Mode :character	Median : 197.851
NA	NA	NA	Mean : 293.928
NA	NA	NA	3rd Qu.: 396.325
NA	NA	NA	Max. :3072.650

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

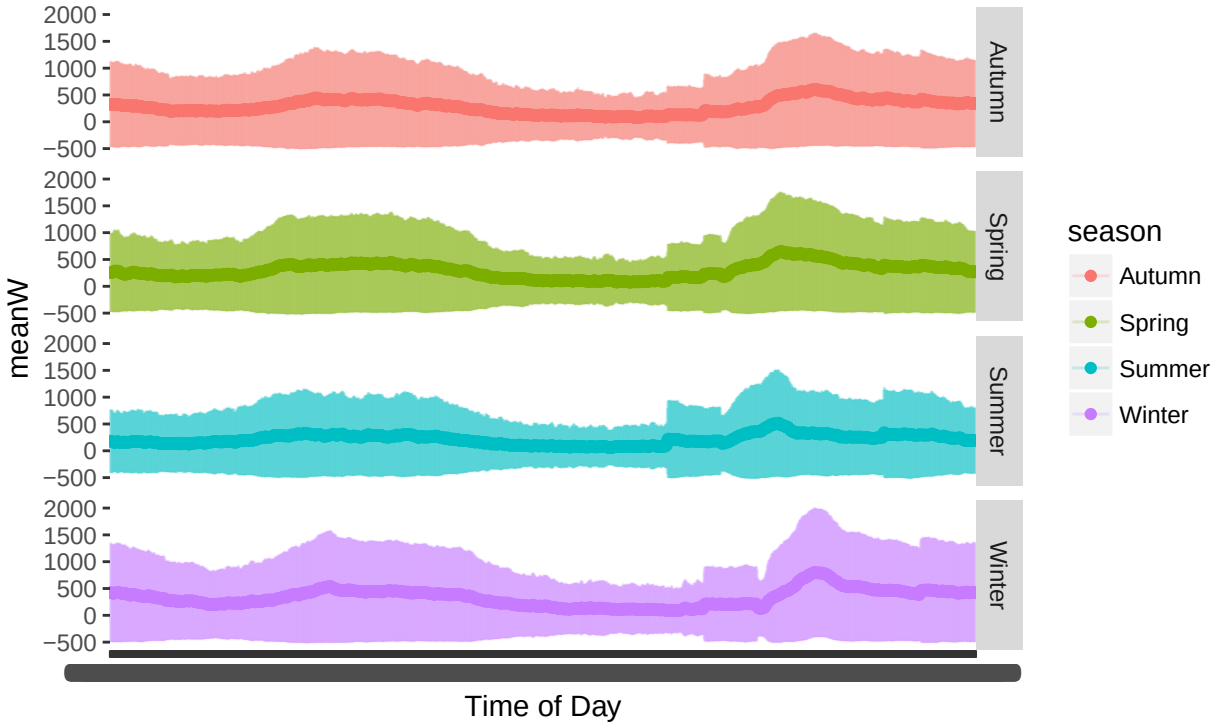
Note that the code saves a high definition version of the plot and the profiles for future re-use.

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/1mi Water_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/Hot Water_2015-04-01_2016-03-31_byHouseholdSeasonalProfi` if you prefer `data.table`

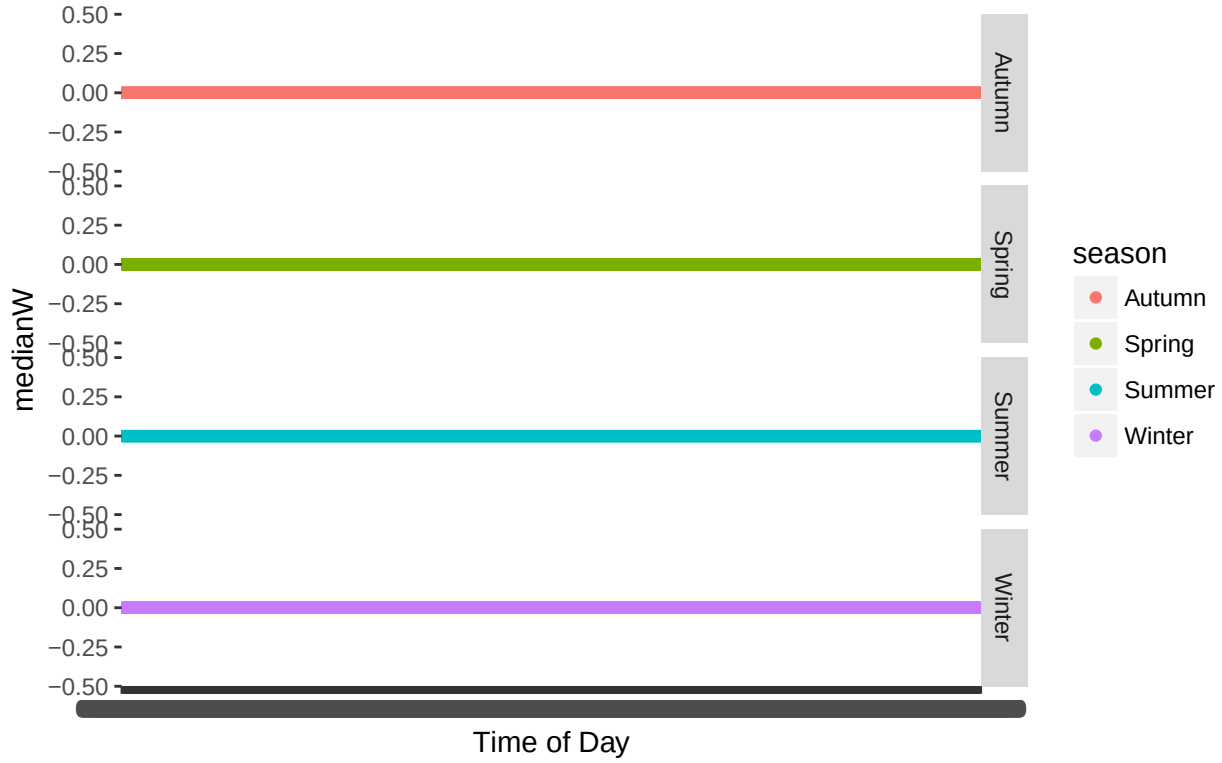
5.2 Profile plots: overall household mean

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/
Circuits: Hot Water from 2015−04−01 to 2016−03−31
n households = 32
Error bars = +/− 1 s.d.

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



e: /Volumes/hum−csafe/Research Projects/GREEN Grid/_RAW DATA/GridSpyData/
Circuits: Hot Water from 2015−04−01 to 2016−03−31
n households = 32

```
## [1] "Saving plot to /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/safe/gridSpy/1min/pro
## [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. : 58.62	Min. :0	Min. :2230	Min. : 320.4
Class :character	Class :character	1st Qu.:168.06	1st Qu.:0	1st Qu.:2432	1st Qu.: 578.2
Mode :character	Mode :character	Median :264.90	Median :0	Median :2570	Median : 731.1
NA	NA	Mean :281.18	Mean :0	Mean :2517	Mean : 723.0
NA	NA	3rd Qu.:390.05	3rd Qu.:0	3rd Qu.:2661	3rd Qu.: 879.5
NA	NA	Max. :805.88	Max. :0	Max. :2671	Max. :1208.5

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-facted 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 472.52 seconds (7.88 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 (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)
- 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 10.13.5
##
## Matrix products: default
## BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions
## 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] gdtools_0.1.7     rmarkdown_1.9     kableExtra_0.9.0  knitr_1.20
## [5] readr_1.1.1       ggplot2_2.2.1     dplyr_0.7.5       data.table_1.11.2
## [9] nzGREENGrid_0.1.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.17        svglite_1.2.1       lubridate_1.7.4
##  [4] lattice_0.20-35     prettyunits_1.0.2   assertthat_0.2.0
##  [7] rprojroot_1.3-2     digest_0.6.15       R6_2.2.2
## [10] cellranger_1.1.0    plyr_1.8.4          backports_1.1.2
## [13] acepack_1.4.1       evaluate_0.10.1     httr_1.3.1
## [16] highr_0.6           pillar_1.2.2        rlang_0.2.0
## [19] progress_1.1.2      lazyeval_0.2.1      readxl_1.1.0
## [22] rstudioapi_0.7      rpart_4.1-13        Matrix_1.2-14
## [25] checkmate_1.8.5     labeling_0.3        splines_3.5.0
## [28] stringr_1.3.1       foreign_0.8-70      htmlwidgets_1.2
## [31] munsell_0.4.3       tinytex_0.5         compiler_3.5.0
## [34] pkgconfig_2.0.1     base64enc_0.1-3     htmltools_0.3.6
## [37] nnet_7.3-12         tidyselect_0.2.4    tibble_1.4.2
```

```
## [40] gridExtra_2.3       htmlTable_1.11.2    Hmisc_4.1-1
## [43] viridisLite_0.3.0   grid_3.5.0          gtable_0.2.0
## [46] magrittr_1.5        scales_0.5.0        stringi_1.2.2
## [49] reshape2_1.4.3      bindrcpp_0.2.2      latticeExtra_0.6-28
## [52] xml2_1.2.0          Formula_1.2-3       RColorBrewer_1.1-2
## [55] tools_3.5.0         glue_1.2.0          purrr_0.2.4
## [58] hms_0.4.2           survival_2.42-3     yaml_2.1.19
## [61] colorspace_1.3-2    cluster_2.0.7-1     rvest_0.3.2
## [64] 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 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.