

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

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

Last run at: 2018-05-05 17:38:39

Contents

1	Citation	2
2	Introduction	3
2.1	Purpose	3
2.2	Requirements:	3
2.3	History	3
2.4	Support	3
3	Obtain listing of files	3
3.1	Data file quality checks	8
4	Load data files	9
5	Data quality analysis	18
6	Runtime	21
7	R environment	21

1 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 consumption data, University of Otago: Dunedin, NZ.

2 Introduction

Report circulation:

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

2.1 Purpose

This report is intended to:

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

2.2 Requirements:

- grid spy 1 minute data downloads

2.3 History

Generally tracked via our git.soton repo.

2.4 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).

This work uis (c) 2018 the University of Southampton.

3 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:

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

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

```
print(paste0("Looking for 1 minute data using pattern = ", pattern1Min, " in ", fpath, " - could take a  
## [1] \"Looking for 1 minute data using pattern = *at1.csv$ in /Volumes/hum-csafe/Research Projects/GRE  
system.time(fListCompletedT <- as.data.table(list.files(path = fpath, pattern = pattern1Min, # use the  
recursive = TRUE)))
```

```

##      user      system elapsed
##    0.827      6.436 5883.518

nFiles <- nrow(fListCompleteDT)
print(paste0("Found ", tidyNum(nFiles), " files"))

## [1] "Found 21,264 files"

if(nrow(fListCompleteDT) == 0){
  stop(paste0("No matching data files found, please check your path (", fpath, ") or your search pattern"))
} else {
  print(paste0("Processing file list and getting file meta-data (please be patient)"))
  fListCompleteDT <- fListCompleteDT[, c("hhID", "fileName") := tstrsplit(V1, "/")]
  fListCompleteDT <- fListCompleteDT[, fullPath := paste0(fpath, hhID, "/", fileName)]
  loopCount <- 1
  # now loop over the files and collect metadata
  for(f in fListCompleteDT[,fullPath]){
    rf <- path.expand(f) # just in case of ~ etc
    fsize <- file.size(rf)
    fmtime <- ymd_hms(file.mtime(rf), tz = "Pacific/Auckland") # requires lubridate
    fListCompleteDT <- fListCompleteDT[fullPath == f, fSize := fsize]
    fListCompleteDT <- fListCompleteDT[fullPath == f, fMTime := fmtime]
    fListCompleteDT <- fListCompleteDT[fullPath == f, fMDate := as.Date(fmtime)]
    fListCompleteDT <- fListCompleteDT[fullPath == f, dateColName := paste0("unknown - file not loaded
    # only try to read files where we think there might be data
    skipThis <- ifelse(fsize > dataThreshold, "Loading (fsize > threshold)", "Skipping (fsize < threshold)
    if(fullFb){print(paste0("Checking file ", loopCount, " of ", nFiles,
                          " (", round(100*(loopCount/nFiles),2), "% checked): ", skipThis))}
    if(fsize > dataThreshold){
      if(fullFb){print(paste0("fSize (", fsize, ") > threshold (", dataThreshold, ") -> loading ", f))}
      row1DT <- fread(f, nrow = 1)
      # what is the date column called?
      fListCompleteDT <- fListCompleteDT[fullPath == f, dateColName := "unknown - can't tell"]
      if(nrow(select(row1DT, contains("NZ"))) > 0){ # requires dplyr
        setnames(row1DT, 'date NZ', "dateTime_char")
        row1DT <- row1DT[, dateColName := "date NZ"]
        fListCompleteDT <- fListCompleteDT[fullPath == f, dateColName := "date NZ"]
      }
      if(nrow(select(row1DT, contains("UTC"))) > 0){ # requires dplyr
        setnames(row1DT, 'date UTC', "dateTime_char")
        row1DT <- row1DT[, dateColName := "date UTC"]
        fListCompleteDT <- fListCompleteDT[fullPath == f, dateColName := "date UTC"]
      }
      # split dateTime
      row1DT <- row1DT[, c("date_char", "time_char") := tstrsplit(dateTime_char, " ")]
      # add example of date to metadata - presumably they are the same in each file?!
      fListCompleteDT <- fListCompleteDT[fullPath == f, dateExample := row1DT[1, date_char]]

      if(fullFb){print(paste0("Checking date formats in ", f))}
      dt <- gs_checkDates(row1DT)
      fListCompleteDT <- fListCompleteDT[fullPath == f, dateFormat := dt[1, dateFormat]]
      if(fullFb){print(paste0("Done ", f))}
    }
    loopCount <- loopCount + 1
  }
}

```

```

print("All files checked")

ofile <- paste0(outPath, indexFile)
print(paste0("Saving 1 minute data files metadata to ", ofile))
write.csv(fListCompleteDT, ofile)
print("Done")

# any date formats are still ambiguous need a deeper inspection using the full file - could be slow
fAmbig <- fListCompleteDT[dateFormat == "ambiguous", fullPath]

for(fa in fAmbig){
  if(baTest | fullFb){print(paste0("Checking ambiguous date formats in ", fa))}
  ambDT <- fread(fa)
  if(nrow(select(ambDT, contains("NZ"))) > 0){ # requires dplyr
    setnames(ambDT, 'date NZ', "dateTime_char")
  }
  if(nrow(select(ambDT, contains("UTC"))) > 0){ # requires dplyr
    setnames(ambDT, 'date UTC', "dateTime_char")
  }
  ambDT <- ambDT[, c("date_char", "time_char") := tstrsplit(dateTime_char, " ")]
  ambDT <- gs_checkDates(ambDT)
  # set what we now know (or guess!)
  fListCompleteDT <- fListCompleteDT[fullPath == fa, dateFormat := ambDT[1,dateFormat]]
}

ofile <- paste0(outPath, indexFile)
print(paste0("Saving final 1 minute data files metadata to ", ofile))
write.csv(fListCompleteDT, ofile)
print("Done")
}

```

```

## [1] "Processing file list and getting file meta-data (please be patient)"
## [1] "All files checked"
## [1] "Saving 1 minute data files metadata to /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data"
## [1] "Done"
## [1] "Saving final 1 minute data files metadata to /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data"
## [1] "Done"

```

```

print(paste0("Overall we have ", nrow(fListCompleteDT), " files from ", uniqueN(fListCompleteDT$hhID), " households."))

```

```

## [1] "Overall we have 21264 files from 44 households."

```

```

# for use below
nFiles <- nrow(fListCompleteDT)
nFilesNotLoaded <- nrow(fListCompleteDT[dateColName %like% "unknown"])

```

Overall we have 21,264 files from 44 households. Of the 21,264, 12,362 (58.14%) were *not* loaded/checked as their file sizes indicated that they contained no data.

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

```

# short cut if file list already saved
ifile <- paste0(outPath, indexFile)
#print(paste0("Loading 1 minute data files metadata to ", ifile))
fListCompleteDT <- fread(ifile)

```

```
t <- fListCompleteDT[, .(nFiles = .N), keyby = .(dateColName, dateFormat)]

kable(caption = "Number of files with given date column names by inferred date format", t)
```

Table 1: Number of files with given date column names by inferred date format

dateColName	dateFormat	nFiles
date NZ	dmy - definite	1
date NZ	mdy - definite	2
date NZ	ymd - default (but day/month value <= 12)	12
date NZ	ymd - definite	67
date UTC	ambiguous	28
date UTC	ymd - default (but day/month value <= 12)	3445
date UTC	ymd - definite	5347
unknown - file not loaded (fsize = 2751)	NA	1812
unknown - file not loaded (fsize = 43)	NA	10550

Results to note:

- There are 28 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 are ambiguous.

```
# list ambiguous files
aList <- fListCompleteDT[dateFormat == "ambiguous", .(file = V1, dateColName, dateExample, dateFormat)]

kable(caption = "Files with ambiguous date formats", aList)
```

Table 2: Files with ambiguous date formats

file	dateColName	dateExample	dateFormat
rf_06/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_07/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_08/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_10/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_11/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_13/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_19/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_21/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_22/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_23/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_24/15Jul2014-25May2016at1.csv	date UTC	27/07/14	ambiguous
rf_25/12Oct2016-20Nov2017at1.csv	date UTC	11-10-16	ambiguous
rf_26/15Jul2014-25May2016at1.csv	date UTC	14/07/14	ambiguous
rf_27/15Jul2014-25May2016at1.csv	date UTC	27/07/14	ambiguous
rf_29/24Mar2015-25May2016at1.csv	date UTC	25/03/15	ambiguous
rf_30/15Feb2016-25May2016at1.csv	date UTC	14/02/16	ambiguous

file	dateColName	dateExample	dateFormat
rf_30/24Mar2015-25May2016at1.csv	date UTC	27/03/15	ambiguous
rf_31/24Mar2015-25May2016at1.csv	date UTC	25/03/15	ambiguous
rf_34/18Jan2016-25May2016at1.csv	date UTC	17/01/16	ambiguous
rf_34/20Jul2015-25May2016at1.csv	date UTC	19/07/15	ambiguous
rf_34/24Mar2015-25May2016at1.csv	date UTC	26/03/15	ambiguous
rf_35/24Mar2015-25May2016at1.csv	date UTC	23/03/15	ambiguous
rf_39/24Mar2015-25May2016at1.csv	date UTC	27/03/15	ambiguous
rf_43/24Mar2015-25May2016at1.csv	date UTC	26/03/15	ambiguous
rf_43/27Mar2015-18Oct2015at1.csv	date UTC	26/03/15	ambiguous
rf_44/24Mar2015-25May2016at1.csv	date UTC	24/03/15	ambiguous
rf_46/12Oct2016-20Nov2017at1.csv	date UTC	11-10-16	ambiguous
rf_47/24Mar2015-25May2016at1.csv	date UTC	24/03/15	ambiguous

Looking at the file names we will assume they are dmy.

```
fListCompletedDT <- fListCompletedDT[dateFormat == "ambiguous", dateFormat := "dmy - inferred"]
```

The following table lists ‘date NZ’ files which are set by default only - do they look OK to assume dateFormat?

```
# list default files
aList <- fListCompletedDT[dateColName == "date NZ" & dateFormat %like% "default", .(file = V1, fSize, da
kable(caption = "Files with inferred default date formats", head(aList))
```

Table 3: Files with inferred default date formats

file	fSize	dateColName	dateExample	dateFormat
rf_01/1Jan2014-24May2014at1.csv	6255737	date NZ	2014-01-06	ymd - default (but day/month value <=
rf_02/1Jan2014-24May2014at1.csv	6131625	date NZ	2014-03-03	ymd - default (but day/month value <=
rf_06/24May2014-24May2015at1.csv	19398444	date NZ	2014-06-09	ymd - default (but day/month value <=
rf_10/24May2014-24May2015at1.csv	24386048	date NZ	2014-07-09	ymd - default (but day/month value <=
rf_11/24May2014-24May2015at1.csv	23693893	date NZ	2014-07-08	ymd - default (but day/month value <=
rf_12/24May2014-24May2015at1.csv	21191785	date NZ	2014-07-09	ymd - default (but day/month value <=

These look OK if we compare the file names with the dateExample.

The following table lists ‘date NZ’ files which are set by default only - do they look OK to assume dateFormat?

```
# list default files
aList <- fListCompletedDT[dateColName == "date UTC" & dateFormat %like% "default", .(file = V1, fSize, da
kable(caption = "Files with inferred default date formats", head(aList))
```

Table 4: Files with inferred default date formats

file	fSize	dateColName	dateExample	dateFormat
rf_06/10Apr2018-11Apr2018at1.csv	156944	date UTC	2018-04-09	ymd - default (but day/month value <= 12
rf_06/10Dec2017-11Dec2017at1.csv	156601	date UTC	2017-12-09	ymd - default (but day/month value <= 12
rf_06/10Feb2018-11Feb2018at1.csv	153353	date UTC	2018-02-09	ymd - default (but day/month value <= 12
rf_06/10Jan2018-11Jan2018at1.csv	153982	date UTC	2018-01-09	ymd - default (but day/month value <= 12
rf_06/10Mar2018-11Mar2018at1.csv	156471	date UTC	2018-03-09	ymd - default (but day/month value <= 12
rf_06/10Nov2017-11Nov2017at1.csv	155639	date UTC	2017-11-09	ymd - default (but day/month value <= 12

These also look OK so we will stick with the following derived date formats:

```
t <- fListCompleteDT[, .(nFiles = .N), keyby = .(dateColName, dateFormat)]

kable(caption = "Number of files with given date column names by final imputed date format", t)
```

Table 5: Number of files with given date column names by final imputed date format

dateColName	dateFormat	nFiles
date NZ	dmy - definite	1
date NZ	mdy - definite	2
date NZ	ymd - default (but day/month value <= 12)	12
date NZ	ymd - definite	67
date UTC	dmy - inferred	28
date UTC	ymd - default (but day/month value <= 12)	3445
date UTC	ymd - definite	5347
unknown - file not loaded (fsize = 2751)	NA	1812
unknown - file not loaded (fsize = 43)	NA	10550

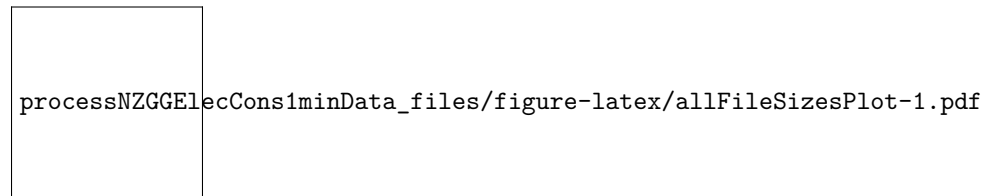
3.1 Data file quality checks

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.

```
myCaption <- paste0("Data source: ", fpath,
                    "\nUsing data received up to ", Sys.Date())

plotDT <- fListCompleteDT[, .(nFiles = .N,
                              meanfSize = mean(fSize)),
                          keyby = .(hhID, date = as.Date(fMDate))]

ggplot(plotDT, aes( x = date, y = hhID, fill = log(meanfSize))) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "black") +
  scale_x_date(date_labels = "%Y %b", date_breaks = "1 month") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.5)) +
  labs(title = "Mean file size of all grid spy data files received per day",
       caption = paste0(myCaption,
                        "\nLog file size used as some files are full year data")
  )
```



```
ggsave(paste0(outPath, "gridSpyAllFileListSizeTilePlot.png"))
```

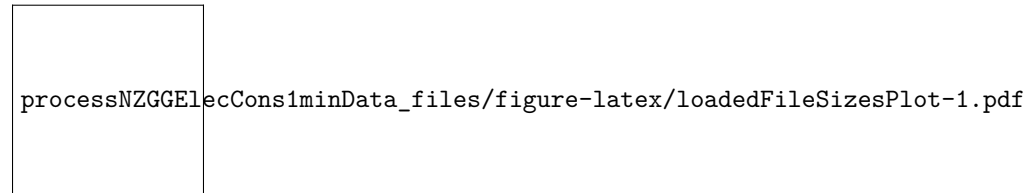
Saving 6.5 x 4.5 in image

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

```
myCaption <- paste0("Data source: ", fpath,
  "\nUsing data received up to ", Sys.Date())

plotDT <- fListCompletedDT[!is.na(dateFormat), .(nFiles = .N,
  meanfSize = mean(fSize)),
  keyby = .(hhID, date = as.Date(fMDate))]

ggplot(plotDT, aes( x = date, y = hhID, fill = log(meanfSize))) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "black") +
  scale_x_date(date_labels = "%Y %b", date_breaks = "1 month") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.5)) +
  labs(title = "Mean file size of loaded grid spy data files received per day",
    caption = paste0(myCaption,
      "\nLog file size used as some files are full year data",
      "\nFiles loaded if fsize > ", dataThreshold, " bytes")
  )
```



```
ggsave(paste0(outPath, "gridSpyLoadedFileListSizeTilePlot.png"))
```

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

4 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 will load.

```
filesToLoadDT <- fListCompletedDT[!is.na(dateFormat)]

t <- filesToLoadDT[, .(nFiles = .N,
  meanSize = mean(fSize),
  minFileDate = min(fMDate),
  maxFileDate = max(fMDate)), keyby = .(hhID)]

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

Table 6: Summary of household files to load

hhID	nFiles	meanSize	minFileDate	maxFileDate
rf_01	3	15548174.7	2016-09-20	2016-09-30
rf_02	3	10134268.3	2016-09-20	2016-09-30
rf_06	182	804031.7	2016-05-25	2018-05-04
rf_07	182	864118.9	2016-05-25	2018-05-04
rf_08	5	23989121.0	2016-05-25	2017-11-21
rf_09	2	14344605.0	2016-09-21	2016-09-21
rf_10	358	525455.0	2016-05-25	2018-03-30
rf_11	484	426639.1	2016-05-25	2018-05-04
rf_12	2	10713096.0	2016-09-21	2016-09-21
rf_13	416	493774.2	2016-05-25	2018-05-04
rf_14	329	424262.0	2016-06-08	2017-12-31
rf_15	2	10553143.0	2016-09-21	2016-09-21
rf_16	1	20037376.0	2016-09-20	2016-09-20
rf_17	202	415129.2	2016-09-21	2018-04-12
rf_18	2	14374309.5	2016-09-21	2016-09-21
rf_19	484	566448.2	2016-05-25	2018-05-04
rf_20	2	14665810.0	2016-09-21	2016-09-21
rf_21	4	23058797.8	2016-05-25	2016-10-12
rf_22	371	533704.5	2016-05-25	2018-01-16
rf_23	484	442331.7	2016-05-25	2018-05-04
rf_24	484	430746.6	2016-05-25	2018-05-04
rf_25	3	12341581.3	2016-06-08	2017-11-21
rf_26	390	411028.3	2016-05-25	2018-05-04
rf_27	3	22607698.7	2016-05-25	2016-09-21
rf_28	2	2297483.0	2016-06-08	2016-09-19
rf_29	481	342604.7	2016-05-25	2018-05-04
rf_30	5	13695336.0	2016-05-25	2016-10-13
rf_31	484	341791.1	2016-05-25	2018-05-04
rf_32	2	13934454.0	2016-06-08	2016-09-20
rf_33	483	288401.9	2016-06-08	2018-05-04
rf_34	7	14106275.3	2016-05-25	2016-10-13
rf_35	134	573648.6	2016-05-25	2017-11-21
rf_36	434	301279.9	2016-06-08	2018-05-04
rf_37	483	302296.3	2016-06-08	2018-05-04
rf_38	201	385707.5	2016-06-08	2017-11-21
rf_39	360	383944.2	2016-05-25	2018-05-04
rf_40	2	9299902.0	2016-06-08	2016-09-20
rf_41	475	265800.7	2016-06-08	2018-05-04
rf_42	45	1315953.6	2016-06-08	2017-11-21
rf_43	4	9442492.0	2016-05-25	2016-09-28
rf_44	484	343415.9	2016-05-25	2018-05-04
rf_45	4	10513812.0	2016-06-08	2017-11-21
rf_46	411	605048.1	2016-06-08	2018-02-21
rf_47	3	17544847.0	2016-05-25	2016-09-20

```

# > Load, process & save the ones which probably have data ----
fListCompleteDT <- fListCompleteDT[, fileLoaded := "No"] # set default
hhIDs <- unique(filesToLoadDT$hhID) # list of household ids
hhStatDT <- data.table() # stats collector

```

```

for(hh in hhIDs){
  tempHhDT <- data.table() # hh data collector
  print(paste0("Loading: ", hh))
  filesToLoad <- filesToLoadDT[hhID == hh, fullPath]
  for(f in filesToLoad){
    if(fullFb){print(paste0("File size (", f, ") = ",
                           filesToLoadDT[fullPath == f, fSize],
                           " so probably OK"))} # files under 3kb are probably empty
    # attempt to load the file
    tempDT <- fread(f)
    if(fullFb){print("File loaded")}
    # set some file stats
    fListCompletedDT <- fListCompletedDT[fullPath == f, fileLoaded := "Yes"]
    fListCompletedDT <- fListCompletedDT[fullPath == f, nObs := nrow(tempDT)] # could include duplicates

    # what is the date column called?
    if(nrow(select(tempDT, contains("NZ"))) > 0){ # requires dplyr
      setnames(tempDT, 'date NZ', "dateTime_char")
      tempDT <- tempDT[, dateColName := "date NZ"]
    }
    if(nrow(select(tempDT, contains("UTC"))) > 0){ # requires dplyr
      setnames(tempDT, 'date UTC', "dateTime_char")
      tempDT <- tempDT[, dateColName := "date UTC"]
    }

    # Now use the pre-inferred dateFormat
    tempDT <- tempDT[, dateFormat := filesToLoadDT[fullPath == f, dateFormat]]
    tempDT <- tempDT[dateFormat %like% "mdy" & dateColName %like% "NZ", r_dateTime := mdy_hm(dateTime_char)]
    tempDT <- tempDT[dateFormat %like% "dmy" & dateColName %like% "NZ", r_dateTime := dmy_hm(dateTime_char)]
    tempDT <- tempDT[dateFormat %like% "ydm" & dateColName %like% "NZ", r_dateTime := ymd_hm(dateTime_char)]
    tempDT <- tempDT[dateFormat %like% "ymd" & dateColName %like% "NZ", r_dateTime := ymd_hm(dateTime_char)]
    tempDT <- tempDT[dateFormat %like% "mdy" & dateColName %like% "UTC", r_dateTime := mdy_hm(dateTime_char)]
    tempDT <- tempDT[dateFormat %like% "dmy" & dateColName %like% "UTC", r_dateTime := dmy_hm(dateTime_char)]
    tempDT <- tempDT[dateFormat %like% "ydm" & dateColName %like% "UTC", r_dateTime := ymd_hm(dateTime_char)]
    tempDT <- tempDT[dateFormat %like% "ymd" & dateColName %like% "UTC", r_dateTime := ymd_hm(dateTime_char)]
    if(fullFb){
      print(head(tempDT))
      print(summary(tempDT))
      #print(table(tempDT$dateFormat))
    }

    fListCompletedDT <- fListCompletedDT[fullPath == f, obsStartDate := min(as.Date(tempDT$r_dateTime))]
    fListCompletedDT <- fListCompletedDT[fullPath == f, obsEndDate := max(as.Date(tempDT$r_dateTime))] #
    fListCompletedDT <- fListCompletedDT[fullPath == f, nCircuits := ncol(select(tempDT, contains("$")))]
    tempHhDT <- rbind(tempHhDT, tempDT, fill = TRUE) # just in case there are different numbers of columns
  }

  # > Remove duplicates caused by over-lapping files and dates etc ----
  # Need to remove all unnecessary vars for this
  try(tempHhDT$dateColName <- NULL)
  try(tempHhDT$dateFormat <- NULL)
  try(tempHhDT$nCircuits <- NULL)
  try(tempHhDT$dateTime_char <- NULL) # if we leave this one in then we get duplicates where we have data

```

```

nObs <- nrow(tempHhDT)
if(fullFb){print(paste0("N rows before removal of duplicates: ", nObs))}
tempHhDT <- unique(tempHhDT)
nObs <- nrow(tempHhDT)
if(fullFb){print(paste0("N rows after removal of duplicates: ", nObs))}

hhStatTempDT <- tempHhDT[, .(nObs = .N,
                             nDataColumns = ncol(select(tempDT, contains("$"))), # the actual number of
                             keyby = (date = as.Date(r_dateTime))) # can't do sensible summary stats on W

# add hhID
hhStatTempDT <- hhStatTempDT[, hhID := hh]

hhStatDT <- rbind(hhStatDT, hhStatTempDT) # add to the collector

# > Save hh file ----

ofile <- paste0(outPath, "data/", hh, "_all_1min_data.csv")
print(paste0("Saving ", ofile, "..."))
write_csv(tempHhDT, ofile)
print(paste0("Saved ", ofile, ", gzipping..."))

cmd <- paste0("gzip -f ", "", path.expand(ofile), "") # gzip it - use quotes in case of spaces in f
try(system(cmd)) # in case it fails - if it does there will just be .csv files (not gzipped) - e.g. u
print(paste0("Gzipped ", ofile))

  if(fullFb){
    print("Col names: ")
    print(names(tempHhDT))
  }

tempHhDT <- NULL # just in case
}

```

```

## [1] "Loading: rf_01"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_01_all_1m
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_01_all_1m
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_01_all_1m
## [1] "Loading: rf_02"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_02_all_1m
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_02_all_1m
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_02_all_1m
## [1] "Loading: rf_06"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_06_all_1m
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_06_all_1m

```

```

## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_06_all_1m"
## [1] "Loading: rf_07"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_07_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_07_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_07_all_1m"
## [1] "Loading: rf_08"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_08_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_08_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_08_all_1m"
## [1] "Loading: rf_09"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_09_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_09_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_09_all_1m"
## [1] "Loading: rf_10"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_10_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_10_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_10_all_1m"
## [1] "Loading: rf_11"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_11_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_11_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_11_all_1m"
## [1] "Loading: rf_12"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_12_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_12_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_12_all_1m"
## [1] "Loading: rf_13"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_13_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_13_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_13_all_1m"
## [1] "Loading: rf_14"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

```

```

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_14_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_14_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_14_all_1m"
## [1] "Loading: rf_15"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_15_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_15_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_15_all_1m"
## [1] "Loading: rf_16"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_16_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_16_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_16_all_1m"
## [1] "Loading: rf_17"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_17_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_17_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_17_all_1m"
## [1] "Loading: rf_18"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_18_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_18_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_18_all_1m"
## [1] "Loading: rf_19"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_19_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_19_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_19_all_1m"
## [1] "Loading: rf_20"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_20_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_20_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_20_all_1m"
## [1] "Loading: rf_21"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_21_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_21_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_21_all_1m"
## [1] "Loading: rf_22"

```

[illegible]

```

## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_29_all_1m"
## [1] "Loading: rf_30"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_30_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_30_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_30_all_1m"
## [1] "Loading: rf_31"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_31_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_31_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_31_all_1m"
## [1] "Loading: rf_32"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_32_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_32_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_32_all_1m"
## [1] "Loading: rf_33"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_33_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_33_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_33_all_1m"
## [1] "Loading: rf_34"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_34_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_34_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_34_all_1m"
## [1] "Loading: rf_35"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_35_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_35_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_35_all_1m"
## [1] "Loading: rf_36"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_36_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_36_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_36_all_1m"
## [1] "Loading: rf_37"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

```



```

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_37_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_37_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_37_all_1m"
## [1] "Loading: rf_38"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_38_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_38_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_38_all_1m"
## [1] "Loading: rf_39"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_39_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_39_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_39_all_1m"
## [1] "Loading: rf_40"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_40_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_40_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_40_all_1m"
## [1] "Loading: rf_41"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_41_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_41_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_41_all_1m"
## [1] "Loading: rf_42"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_42_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_42_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_42_all_1m"
## [1] "Loading: rf_43"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_43_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_43_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_43_all_1m"
## [1] "Loading: rf_44"

## Warning in `[<-data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_44_all_1m"
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_44_all_1m"
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_44_all_1m"
## [1] "Loading: rf_45"

```

```
## Warning in `[<-.data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_45_all_1m
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_45_all_1m
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_45_all_1m
## [1] "Loading: rf_46"

## Warning in `[<-.data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_46_all_1m
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_46_all_1m
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_46_all_1m
## [1] "Loading: rf_47"

## Warning in `[<-.data.table`(x, j = name, value = value): Adding new column
## 'nCircuits' then assigning NULL (deleting it).

## [1] "Saving /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_47_all_1m
## [1] "Saved /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_47_all_1m
## [1] "Gzipped /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean_data/gridSpy/1min/data/rf_47_all_1m
#> Save observed data stats for all files loaded ----
ofile <- paste0(outPath, "hhDailyObservationsStats.csv")
print(paste0("Saving daily observations stats by hhid to ", ofile)) # write out version with file stats

## [1] "Saving daily observations stats by hhid to /Volumes/hum-csafe/Research Projects/GREEN Grid/Clean
write.csv(hhStatDT, ofile)
print("Done")

## [1] "Done"
```

5 Data quality analysis


Now produce some data quality plots & tables.

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 (they indicate data conversion errors)
- more than 1440 observations per day (they indicate potentially duplicate data)

```
# short cut if already generated
# hhStatDT <- as.data.table(read_csv(ofile)) # parses dates


# tile plot ----
ggplot(hhStatDT, aes( x = date, y = hhID, fill = nObs)) +
  geom_tile() +
  scale_fill_gradient(low = "red", high = "green") +
  scale_x_date(date_labels = "%Y %b", date_breaks = "6 months") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.5)) +
  labs(title = "N observations per household per day for all loaded grid spy data",
       caption = paste0(myCaption,
                        "\nOnly files of size > ", dataThreshold, " bytes loaded"))
)
```



processNZGGElecCons1minData_files/figure-latex/loadedFilesObsPlots-1.pdf

```
ggsave(paste0(outPath, "gridSpyLoadedFileNobsTilePlot.png"))

## Saving 6.5 x 4.5 in image
# point plot ----
ggplot(hhStatDT, aes( x = date, y = nObs, colour = hhID)) +
  geom_point() +
  scale_x_date(date_labels = "%Y %b", date_breaks = "6 months") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.5)) +
  labs(title = "N observations per household per day for all loaded grid spy data",
       caption = paste0(myCaption,
                        "\nOnly files of size > ", dataThreshold, " bytes loaded")
  )
```



processNZGGElecCons1minData_files/figure-latex/loadedFilesObsPlots-2.pdf

```
ggsave(paste0(outPath, "gridSpyLoadedFileNobsPointPlot.png"))
```

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!

```
# Stats table (so we can pick out the dateTime errors)
t <- hhStatDT[, .(minObs = min(nObs),
                  maxObs = max(nObs), # should not be more than 1440, if so suggests duplicates
                  meanNDataColumns = mean(nDataColumns), #i.e. n circuits
                  minDate = min(date),
                  maxDate = max(date)),
               keyby = .(hhID)]

kable(caption = "Summary observation stats by hhID", t)
```

Table 7: Summary observation stats by hhID

hhID	minObs	maxObs	meanNDataColumns	minDate	maxDate
rf_01	171	1500	6	2014-01-05	2015-10-20

hhID	minObs	maxObs	meanNDataColumns	minDate	maxDate
rf_02	215	1440	6	2014-03-02	2015-05-28
rf_06	243	1500	6	2014-06-08	2018-05-04
rf_07	105	1500	6	2014-07-13	2018-05-04
rf_08	123	1500	6	2014-05-28	2017-05-15
rf_09	163	1500	6	2014-07-13	2015-07-16
rf_10	389	1500	6	2014-07-08	2018-03-29
rf_11	278	1500	6	2014-07-07	2018-05-04
rf_12	85	1500	6	2014-07-08	2015-06-02
rf_13	456	1500	6	2014-06-05	2018-05-04
rf_14	120	1500	6	2014-07-13	2017-12-30
rf_15	62	1440	6	2015-01-14	2016-04-18
rf_16	720	1500	6	2014-07-09	2015-03-25
rf_17	22	1500	6	2014-05-29	2018-04-11
rf_18	157	1500	6	2014-05-29	2015-06-11
rf_19	387	1500	9	2014-07-14	2018-05-04
rf_20	98	1500	6	2014-05-28	2015-06-11
rf_21	195	1500	6	2014-07-14	2016-07-01
rf_22	6	1500	6	2014-06-05	2018-01-14
rf_23	171	1500	6	2014-05-25	2018-05-04
rf_24	571	1500	6	2014-05-28	2018-05-04
rf_25	45	1500	6	2015-05-24	2016-10-22
rf_26	362	2231	6	2014-07-10	2018-05-04
rf_27	567	1560	6	2014-07-27	2016-05-13
rf_28	297	1440	6	2015-03-26	2015-05-26
rf_29	720	1500	6	2015-03-25	2018-05-04
rf_30	205	1500	6	2015-03-27	2016-09-29
rf_31	720	1500	6	2015-03-25	2018-05-04
rf_32	325	1500	6	2015-03-25	2016-04-05
rf_33	369	1500	6	2015-03-23	2018-05-04
rf_34	317	1500	6	2014-11-03	2016-08-24
rf_35	50	1500	6	2015-03-22	2017-05-17
rf_36	29	1500	6	2015-03-23	2018-05-04
rf_37	720	1500	6	2015-03-23	2018-05-04
rf_38	398	1500	6	2015-03-24	2017-08-22
rf_39	163	1823	5	2015-03-27	2018-05-04
rf_40	268	1500	6	2015-03-24	2015-11-22
rf_41	1	1573	6	2015-03-25	2018-05-04
rf_42	79	1500	6	2015-03-23	2017-02-18
rf_43	780	1495	6	2015-03-26	2015-10-18
rf_44	720	1500	6	2015-03-24	2018-05-04
rf_45	69	1499	6	2015-03-24	2016-10-15
rf_46	305	3000	13	2015-03-26	2018-02-19
rf_47	159	1500	6	2015-03-24	2016-05-08

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

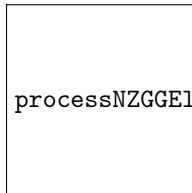
```
plotDT <- hhStatDT[, .(nHH = uniqueN(hhID)), keyby = .(date)]

# point plot ----
ggplot(plotDT, aes( x = date, y = nHH)) +
  geom_point() +
  scale_x_date(date_labels = "%Y %b", date_breaks = "6 months") +
```

```

theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.5)) +
labs(title = "N live households per day for all loaded grid spy data",
      caption = paste0(myCaption,
                       "\nOnly files of size > ", dataThreshold, " bytes loaded")
)

```



```
ggsave(paste0(outPath, "gridSpyLiveHouseholdsToDate.png"))
```

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

6 Runtime

```

t <- proc.time() - startTime
elapsed <- t[[3]]

```

Analysis completed in 3.9801353×10^4 seconds (663.36 minutes) using knitr in RStudio with R version 3.4.4 (2018-03-15) running on macOS High Sierra 10.13.4

7 R environment

R packages used: data.table, lubridate, ggplot2, readr, dplyr, knitr

- 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)
- knitr - to create this document (Xie 2016)
- greenGridr - for local NZ GREEN Grid utilities

```
sessionInfo()
```

```

## 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] knitr_1.20      dplyr_0.7.4      readr_1.1.1
## [4] ggplot2_2.2.1   lubridate_1.7.4   data.table_1.10.4-3
## [7] greenGridr_0.1.0
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.16     bindr_0.1.1      magrittr_1.5
## [4] hms_0.4.2        munsell_0.4.3    colorspace_1.3-2
## [7] R6_2.2.2         rlang_0.2.0.9001 highr_0.6
## [10] stringr_1.3.0    plyr_1.8.4       tools_3.4.4
## [13] grid_3.4.4       gtable_0.2.0     htmltools_0.3.6
## [16] assertthat_0.2.0 yaml_2.1.18       lazyeval_0.2.1
## [19] rprojroot_1.3-2  digest_0.6.15    tibble_1.4.2
## [22] bindrcpp_0.2.2   glue_1.2.0       evaluate_0.10.1
## [25] rmarkdown_1.9    labeling_0.3      stringi_1.1.7
## [28] compiler_3.4.4   pillar_1.2.2     scales_0.5.0.9000
## [31] backports_1.1.2  pkgconfig_2.0.1
```

Dowle, M, A Srinivasan, T Short, S Lianoglou with contributions from R Saporita, 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>.