FlipTheFleet Black Box Data Tests

Testing Time of Charging

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

Last run at: 2018-09-17 12:53:42

Contents

1 Citation

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

- Anderson, B. (2018) *FlipTheFleet Black Box Data Tests: Testing Time of Charging*, Centre for Sustainability, University of Otago: Dunedin.

Figure 1: The Black Box (Source: FlipTheFleet)

2 About

2.1 Circulation

Report circulation:

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

2.2 Purpose

This report is intended to:

- load and test preliminary 'black box' (see Figure 1) EV monitoring data provided for assessment purposes by FlipTheFleet
- preliminary analysis of time of charging to understand impact on 'peak' electricity demand

2.3 Requirements:

- test FlipTheFleet black box dataset stored on the University of Otago's High-Capacity Central File Storage HCS at: /Volumes/hum-csafe/Research Projects/GREEN Grid/

2.4 Code and report history

Generally tracked via our git.soton repo:

- history
- issues

Specific history of this code:

- https://git.soton.ac.uk/ba1e12/nzGREENGrid/tree/master/analysis/ev

2.5 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 :-)

2.6 Notes

This document was created using knitr in RStudio with R version 3.5.1 (2018-07-02) running on x86_64-apple-darwin15.6.0. Some of the R code has been included where used for information and reference purposes. Full code is available as noted above.

3 Load and check data

3.1 Load data

In this section we load and describe the data from . First we load the data.

```
ftfDT <- data.table::as.data.table(readr::read_csv(fileToLoad))
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   `Reg No` = col_character(),
##   `Date (GPS)` = col_character(),
##   `Time (GPS)` = col_time(format = ""),
##   Latitude = col_double(),
##   Longitude = col_double(),
##   Altitude = col_double(),
##   `Speed (GPS)` = col_double(),
##   `Speed (Speedometer)` = col_double(),
##   `Course (deg)` = col_double(),
##   SOC = col_double(),
##   AHr = col_double(),
##   `Pack volts` = col_double(),
##   `Pack amps` = col_double(),
##   `Pack 1 temp (C)` = col_double(),
##   `Pack 2 temp (C)` = col_double(),
##   `Pack 3 temp (C)` = col_double(),
##   `Pack 4 temp (C)` = col_double(),
##   `12V battery (amps)` = col_double(),
```

Table 1: (#tab:process ftf data)Number of obs and mean pack volts where date cannot be set by original GPS Date and Time

Date (GPS)	Time (GPS)	nObs	meanPackVolts
NA	NA	1160	372.2431

Table 2: (#tab:process ftf data)Number of obs and mean pack volts where time cannot be set by original GPS Date and Time

Date (GPS)	Time (GPS)	nObs	meanPackVolts
NA	NA	1160	372.2431

```
##   Hx = col_double(),
##   VIN = col_character()
##   # ... with 16 more columns
## )
```

```
## See spec(...) for full column specifications.
```

That loaded 12,487 observations.

3.2 Create derived variables

Next we create some useful derived variables. Errors may be 'odd' dates and times or even not set in the original data as we see below.

```
# create derived variables
ftfDT <- createDerivedFtF(ftfDT)
```

```
## Warning: 1160 failed to parse.
```

```
t <- ftfDT[is.na(rDate), .(nObs = .N,
                           meanPackVolts = mean(`Pack volts`)),
           keyby = .(`Date (GPS)`, `Time (GPS)`)]

knitr::kable(caption = "Number of obs and mean pack volts where date cannot be set by original GPS Date
```

```
t <- ftfDT[is.na(rTime), .(nObs = .N,
                           meanPackVolts = mean(`Pack volts`)),
           keyby = .(`Date (GPS)`, `Time (GPS)`)]

knitr::kable(caption = "Number of obs and mean pack volts where time cannot be set by original GPS Date
```

We remove these NAs from the data

> *XX (should we - what does GPS NA mean? no signal?) XX*

Doing so removed 1160 (9.29 %) observations.

3.3 Location inference

Figure 2 maps the location of a subset of the observations in the dataset coloured by `Speed (GPS)`. If we selected just one vehicle and zoomed our map to the location at 01:00 - 04:00 when speed is 0 then we would probably determine their home. We could also determine other places visited at other times... This indicates how disclosive GPS Lat/Long can be even if address data is not provided.

```
## Warning: bounding box given to google - spatial extent only approximate.
```

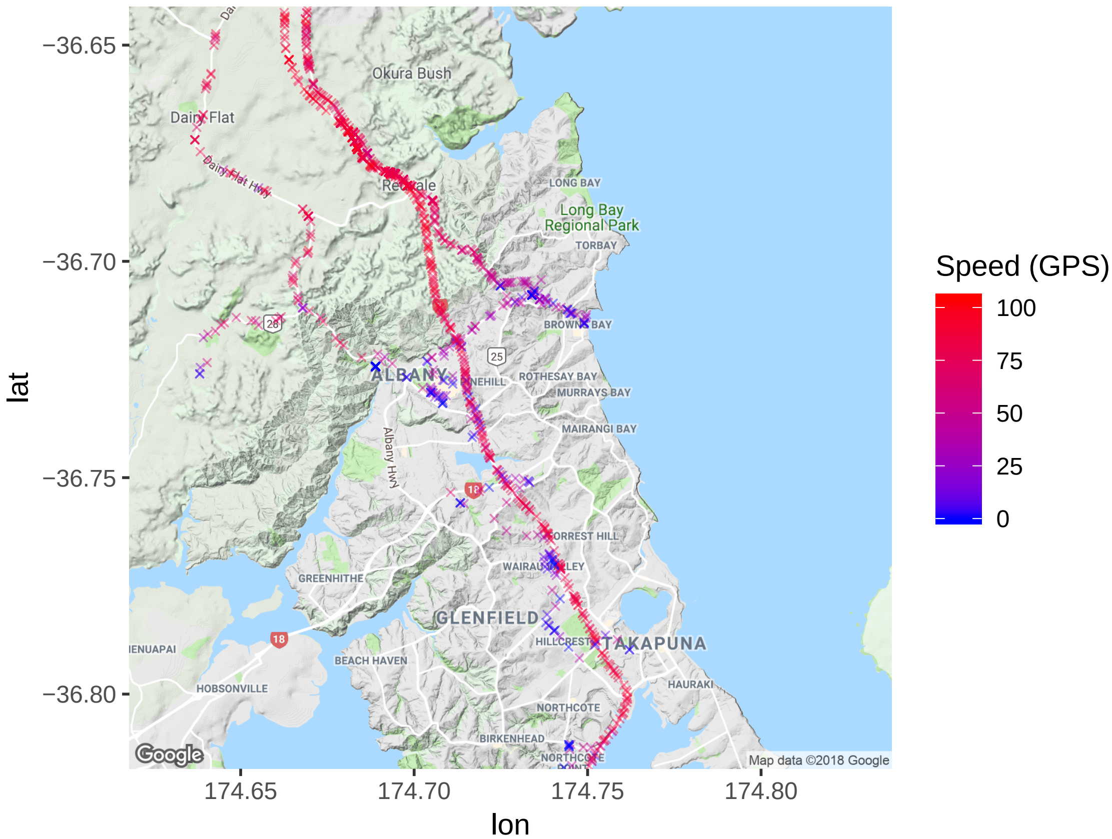

Figure 2: Map of all observations coloured by speed

```
## converting bounding box to center/zoom specification. (experimental)

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=-36.729092,174.727632&zoom=12&si

## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.

## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.

## Warning: Removed 10172 rows containing missing values (geom_point).
```

To avoid any risk of such disclosure we next infer a very coarse geo-location at each time point so that we can remove the potentially disclosive GPS data before moving on to the analysis.

Note that this does not necessarily render this dataset *safe* (anonymised) as there may well be other variables that provide sufficient information either on their own or together which would enable identification of the car and it's owner.

```
ftfDT <- inferLocationFtF(ftfDT)

plotDT <- ftfDT[, .(nObs = .N),
                keyby = .(geoLoc, obsHour, rDow)]

ggplot(plotDT, aes(x = obsHour, y = nObs)) +
  geom_col() +
  facet_grid(rDow ~ geoLoc)
```

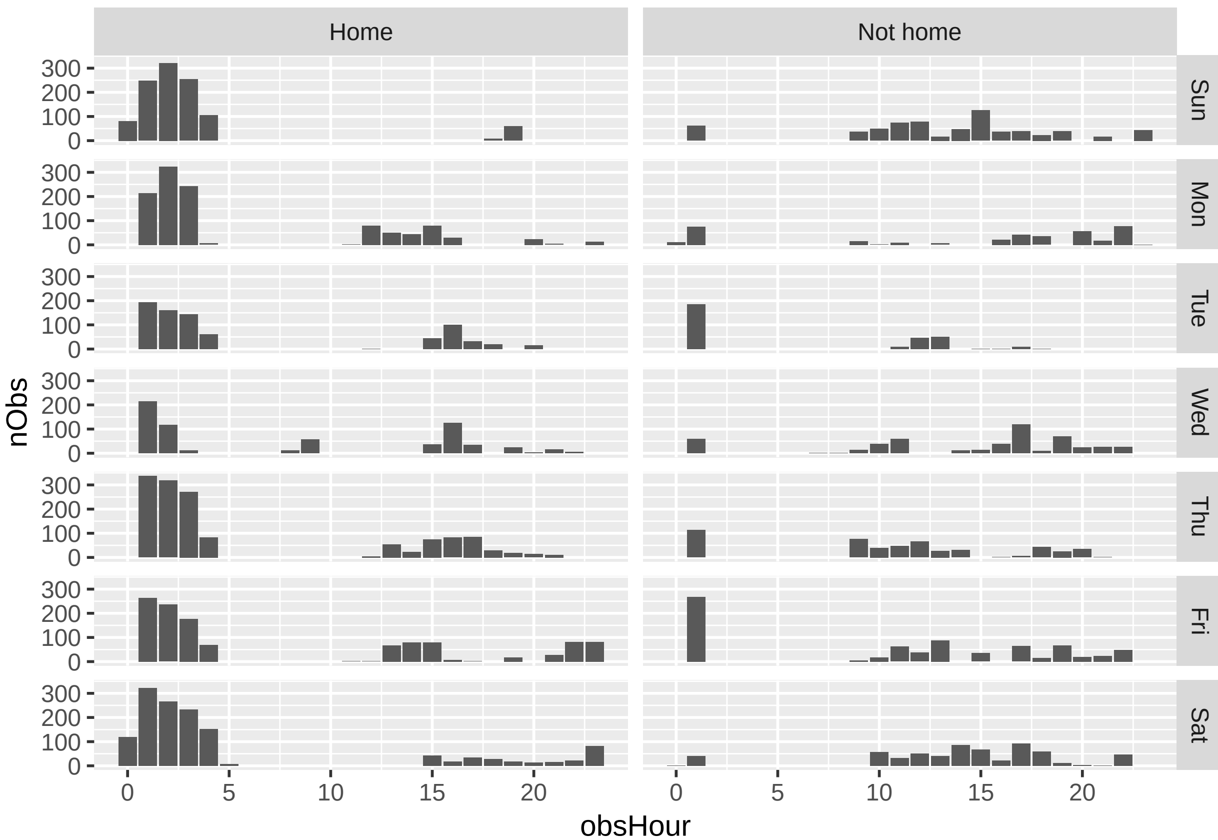

Figure 3: Check inferred location

Table 3: Summary of Charger (amps)

	Charger (amps)	Charger (V)	powerW
	Min. : 0.00	Min. : 0.000	Min. : 0.00
	1st Qu.:15.56	1st Qu.: 5.711	1st Qu.: 89.23
	Median :15.62	Median :239.094	Median :3733.15
	Mean :12.05	Mean :155.725	Mean :2448.08
	3rd Qu.:15.62	3rd Qu.:241.336	3rd Qu.:3768.19
	Max. :33.44	Max. :249.445	Max. :7977.19

Figure 3 shows the results of this inference. Does it look like a reasonable guestimate of location?

3.4 Make safe dataset

Next we remove the following variables as they are potentially disclosive:

- `Reg No` - this is converted to `evID` using an encryption method
- `Latitude`
- `Longitude`
- `Course (deg)`

```
## Deleting old .gz file...
```

```
## DONE
```

```
## gzipping file to: /Volumes/hum-csafe/Research Projects/GREEN Grid/cleanData/flipTheFleet/EVBlackBox 
```

```
## Done
```

The remaining variables are described in detail in ftFBlackBoxTestDataCodebook.docx.

3.5 Check variables of interest for this analysis

Check charger related variables. We think these are:

- `Charger (amps)`
- `Charger (V)`

Multiplying these two will give power in W. Figures 4 to 6 examine the distribution of amps, volts and the derived W.

The following table sumarises these distributions.

XX Is the Amp reading of 33 an outlier? If so should we ignore the ~= 8kW values? XX

4 Analysis: Timing of charging

We assume `powerW` is a good indicator of charging. For now we do not exclude the apparent outlier caused by the few very high Amp values (`Charger (amps)` > 30) but we flag them as potential problems.

Figure 7 shows mean kW by time of day over the entire test datset of 39 days from 2018-05-01 to 2018-06-11.

Figure 7 seems to show that the very high Amp values occured on one Sunday and we also appear to see some night charging 'not at home' which may indicate our location inference is imperfect. We re-draw this plot (Figure 8) excluding the Amp outliers and also excluding values of exactly 0 for clarity.

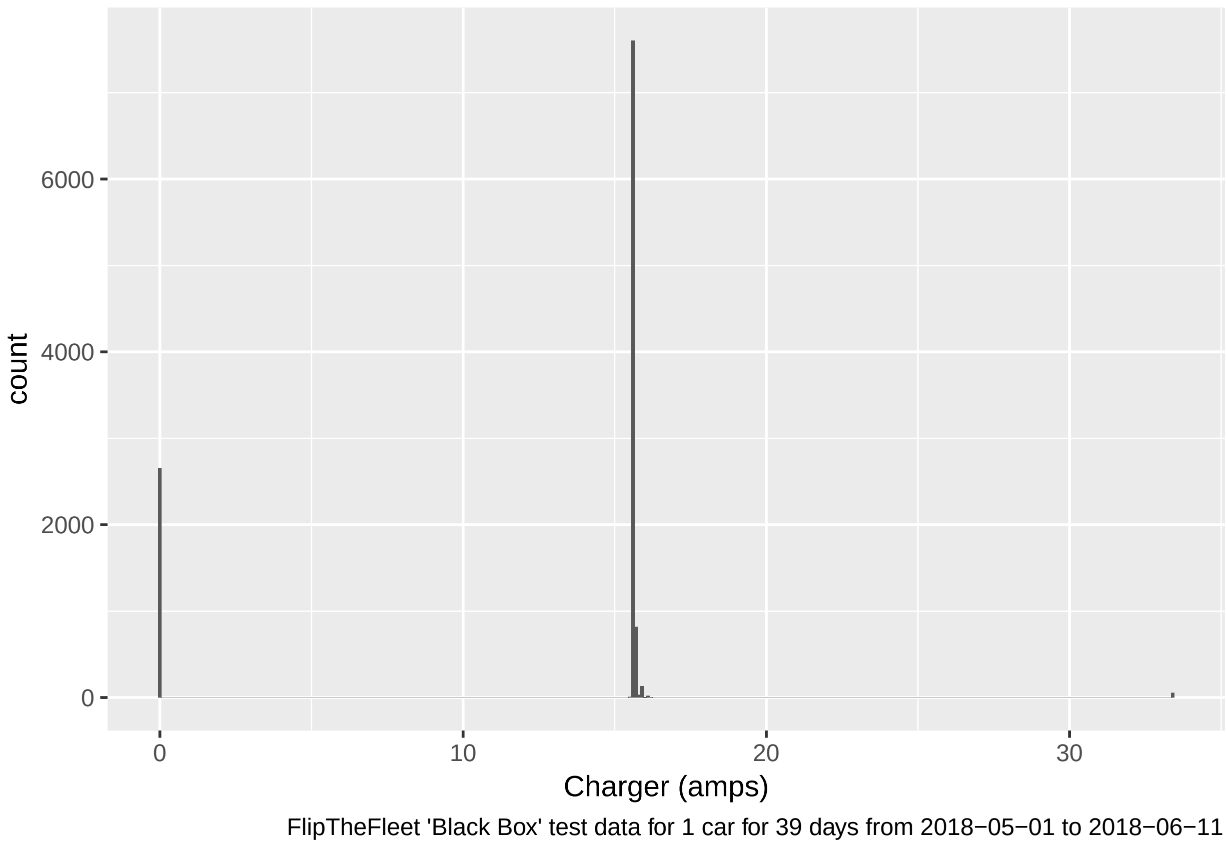

Figure 4: Distribution of charger amp readings

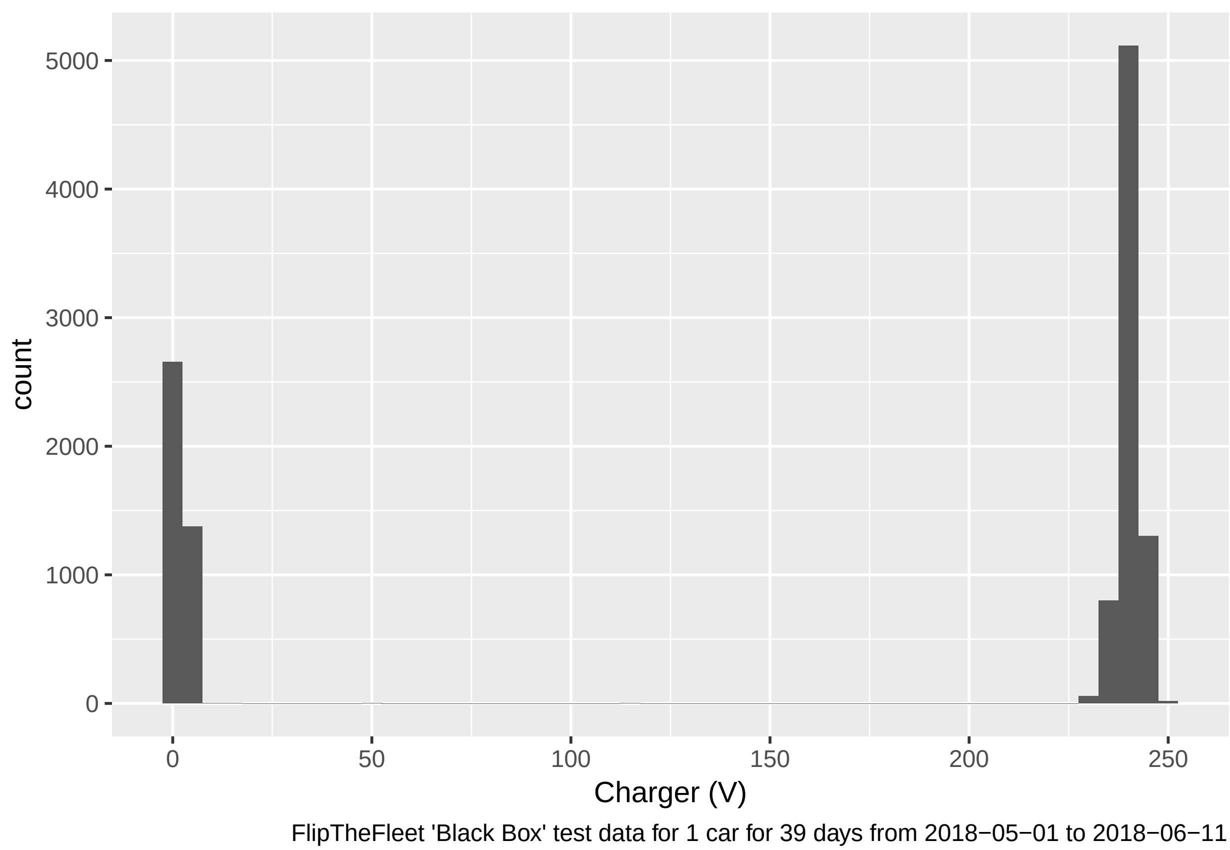

Figure 5: Distribution of charger volt readings

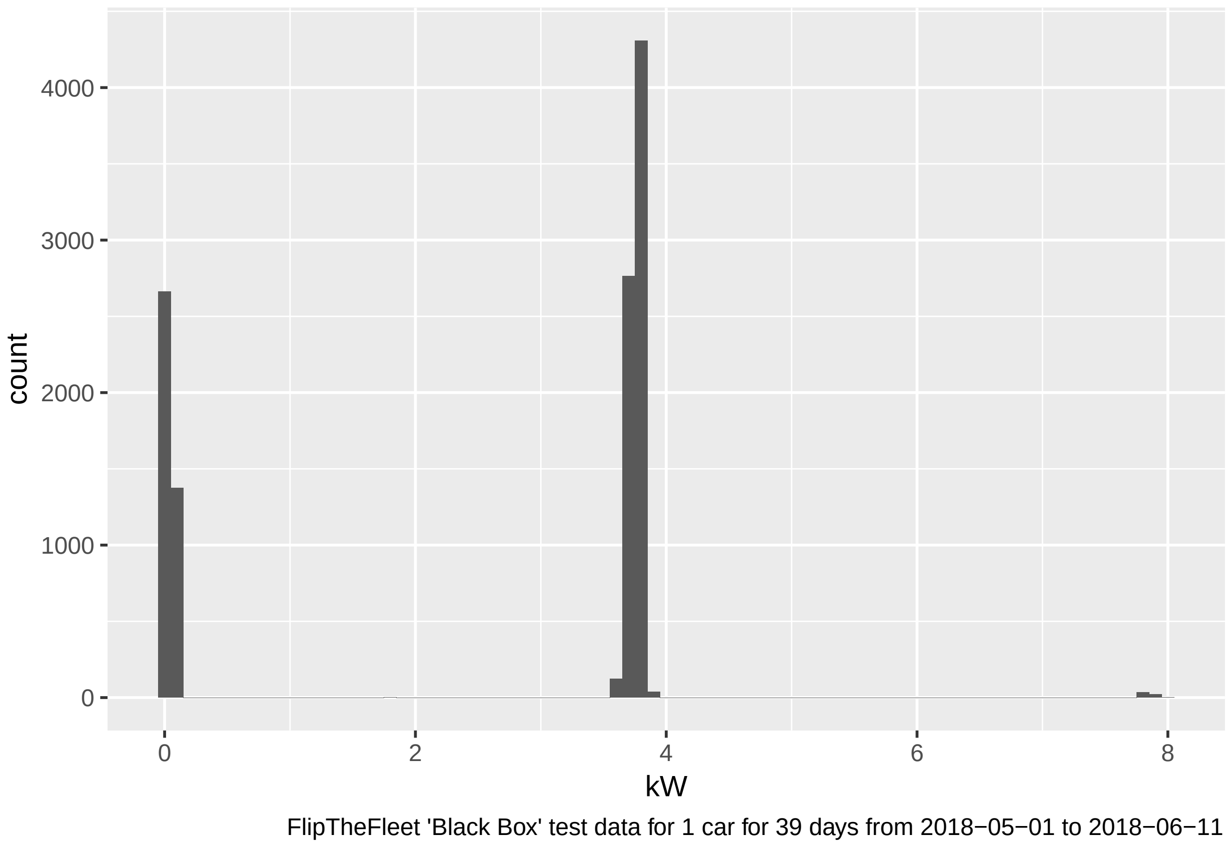

Figure 6: Distribution of derived power demand

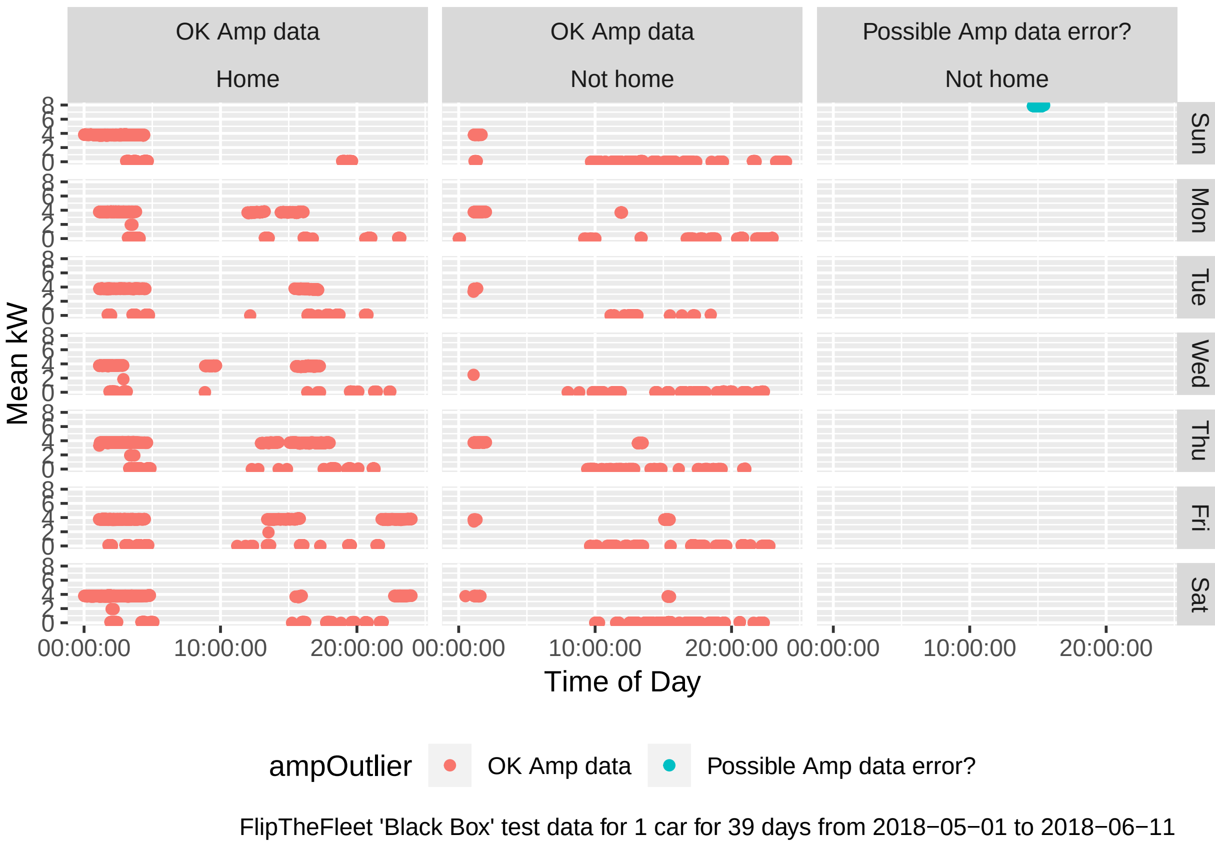

Figure 7: Inferred charging times and power draw (all data)

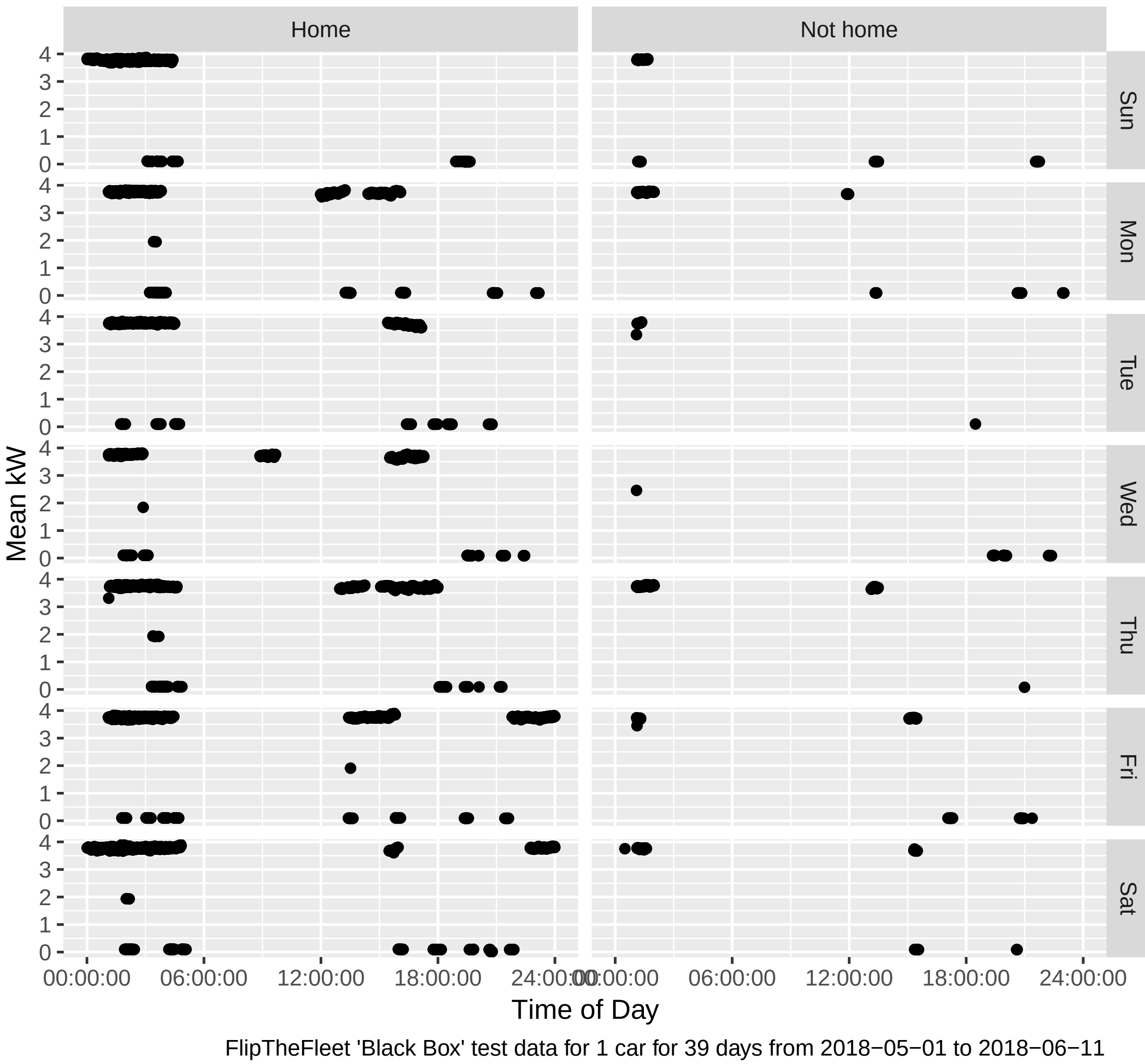

Figure 8: Inferred charging times and power draw (outliers and exactly zero values excluded)

We can see that this car appears to be on an overnight charging timer at home although we do still see charging scattered through the rest of the day.

So where is the non-home charging happening? We need a way to infer other locations without being disclosive.

5 Conclusions

Questions to be asked

- Data:
  - Do the Amp & Volt distributions look right?
  - Cause of Amp outliers?
  - Date/Time NA (9.29 % of observations) are due to a lack of GPS date/time (and lat/long). Does this mean that there is no date/time in the data when GPS has no signal? Our tests suggest that other data (e.g. power etc) is logged even though there is no date/time. Do we need another source of date/time? Could we infer time from seconds since powered on (assumes GPS OK at start-up?)?
- Research:
  - Do all FlipTheFleet EV owners charge like this?
  - Where are the EVs being charged when not at home and how can we tell?
  - Does it vary by car/tariff/commute pattern/main use?
  - What other patterns exist and how much within-vehicle and between-vehicle variation is there?

6 Runtime

Analysis completed in 29.11 seconds (0.49 minutes) using knitr in RStudio with R version 3.5.1 (2018-07-02) 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)
- openssl - for hashing `Reg No` (**???**)
- knitr - to create this document & neat tables (Xie 2016)
- GREENGrid - for local NZ GREEN Grid project utilities

Session info:

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.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] knitr_1.20.13     readr_1.1.1       ggplot2_3.0.0     data.table_1.11.4
## [5] GREENGrid_0.1.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.4 xfun_0.3          purrr_0.2.5
##  [4] reshape2_1.4.3   lattice_0.20-35   colorspace_1.3-2
##  [7] htmltools_0.3.6  yaml_2.2.0        rlang_0.2.2
## [10] pillar_1.3.0     glue_1.3.0        withr_2.1.2
## [13] sp_1.3-1         bindrcpp_0.2.2    jpeg_0.1-8
## [16] plyr_1.8.4       bindr_0.1.1       stringr_1.3.1
## [19] munsell_0.5.0    gtable_0.2.0      RgoogleMaps_1.4.2
## [22] mapproj_1.2.6    evaluate_0.11     labeling_0.3
## [25] highr_0.7        proto_1.0.0       Rcpp_0.12.18
## [28] geosphere_1.5-7  openssl_1.0.2     scales_1.0.0
## [31] backports_1.1.2  rjson_0.2.20      hms_0.4.2
## [34] png_0.1-7        digest_0.6.15     stringi_1.2.4
## [37] bookdown_0.7     dplyr_0.7.6       grid_3.5.1
## [40] rprojroot_1.3-2  tools_3.5.1       magrittr_1.5
## [43] maps_3.3.0       lazyeval_0.2.1    tibble_1.4.2
## [46] crayon_1.3.4     pkgconfig_2.0.2   lubridate_1.7.4
## [49] assertthat_0.2.0 rmarkdown_1.10    R6_2.2.2
## [52] ggmap_2.6.1      compiler_3.5.1
```

References

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, 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.