Data analysis report - Root Insurance

Issac Lee[a]

[a]Department of Statistics & Actuarial Science, 241 Schaeffer Hall, Iowa City, Iowa 52242-1409

This version was compiled on October 30, 2019

This report illustrates how I approach the project and the decision making the process for Root insurance's data analysis project. The goal of this project is to find the best-matched OBDII trip data, which corresponds to a trip data from a user's smartphone. The data analysis process appears in the first chapter, and the instructional manual follows in the later chapter. The `R` code implementations follow Google's `R` code style guide.

Data Preparation. There are two telematics data set from independent sensors: GPS in smartphone, OBDII. The two data sets are provided as two separate file. Here we assume that the two `json` files are extracted in the working directory, which means you have the following two files in the working directory:

mobile_trips.json, obd2_trips.json

Data load and structure. Since the two files are recorded as `json` format, `jsonlite` package will be used to load the data.

```
library(jsonlite)

# read mobile trip & obd2 trip
mobile_data <- fromJSON("./mobile_trips.json")
obd2_data <- fromJSON("./obd2_trips.json")
```

Smartphone data set consisists of 44 trips and OBDII data set consists of 41 trips. The colum names of the each data set and the data types are as follows:

- OBDII data
  - trip_id (`char`), timestamp (`dbl`), speed (`int`)
- Mobile data
  - trip_id (`char`), created_at (`dbl`), timestamp (`dbl`), speed (`dbl`), accuracy (`int`)

Visualization of sample trips. Fig 1 shows the sample speed graph of one of the trips from OBDII data. As we can see, the trip was recorded for around 1,200 seconds and we do not know the unit for the recorded speed. By examing the data set littel bit, it can be easily realized that the first trip from each sources corresponds to each other as in Fig. 1 and Fig 2.

This *pinp is not PNAS* template started when the introduction to Rcpp by Eddelbuettel and Balamuta (2017) was converted into this updated Rcpp Introduction vignette. It is based on the pnas_article template of the wonderful rticles package by Allaire *et al.* (2017b). The conversion from markdown to latex is facilitated by rmarkdown (Allaire *et al.*, 2017a) and knitr (Xie, 2017). The underlying LaTeX macros are from pnas.org.

The remainder of the document carries over from the corresponding pnas_article template document. but has been edited and updated to our use case. A few specific tips follow. In general, for fine-tuning some knowledge of LaTeX is helpful.

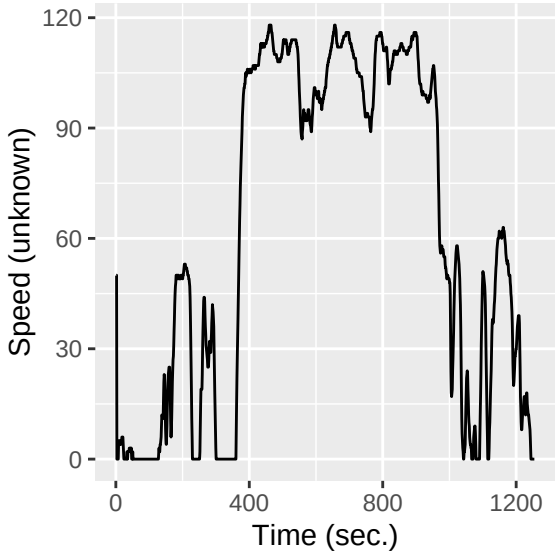

Fig. 1. A sample of speed graph of OBDII (the 1st trip).

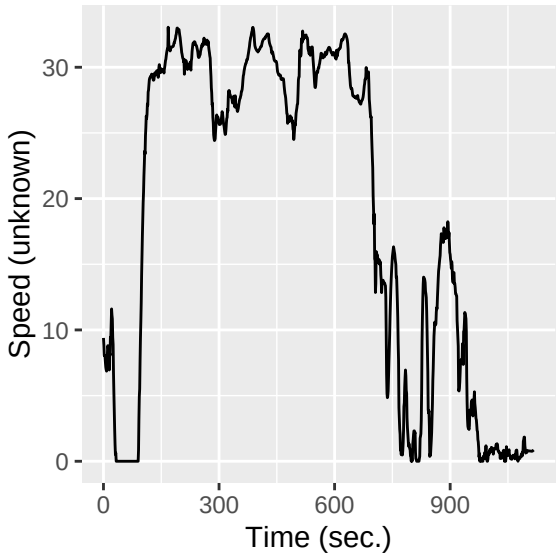

Fig. 2. A sample of speed graph of Smartphone (the 1st trip).

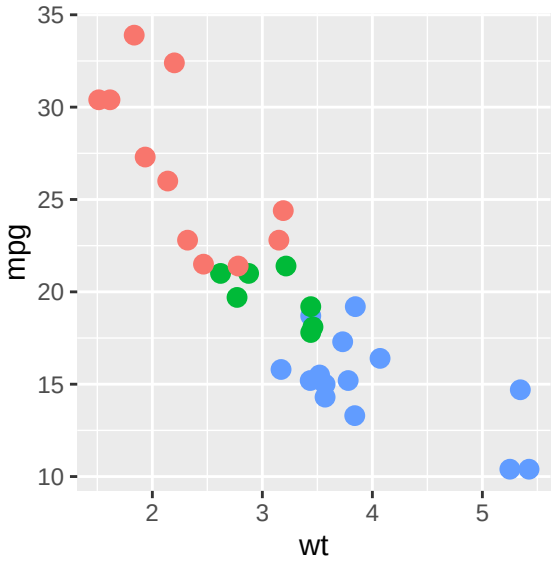

Fig. 3. Narrow ggplot2 figure

Author Affiliations. Per common academic best practice, you can include your department, institution, and complete address, with the ZIP/postal code, for each author. Use lower case letters to match authors with institutions, as shown in the example. Authors with an ORCID ID may supply this information at submission.

Document Options. We support several options via the YAML header

- Setting a DOI or URL footer, for example for the CRAN package URL, which is placed in the bottom-left footer of the title page and even pages;
- Setting a footer label, for example *YourPackage Vignette* stating your package, which is placed in the bottom-right footer on odd pages;
- Setting a free-form author field used on the inside footer;
- Optional *Draft* watermark to be added to each page;
- Line of custom text in subtitle (`date_subtitle`) suitable to give publication info of the draft, e.g. journal name in a post-print.

References. Here we differ from PNAS and suggest natbib. References will appear in author-year form. Use `\citet{}`, `\citep{}`, etc as usual.

We default to the `jss.bst` style. To switch to a different bibliography style, please use `biblio-style: style` in the YAML header.

Inline R Code. The PNAS sample included a fixed PNG image here, but this document prefers to show the results and embedding of *R* code.

```
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
    geom_point(size=3, aes(colour=factor(cyl))) +
    theme(legend.position="none")
```

Here we use a standard knitr bloc with explicit options for

- figure width and height (`fig.width`, `fig.height`), both set to three inches;
- whether the code is shown (`echo=TRUE`); and
- the caption (`fig.cap`) as shown above.

Digital Figures. Markdown, Pandoc and LaTeX support `.eps` and `.pdf` files.

Figures and Tables should be labelled and referenced in the standard way using the `\label{}` and `\ref{}` commands.

The R examples above show how to insert a column-wide figure. To insert a figure wider than one column, please use the `\begin{figure*}...\end{figure*}` environment.

One (roundabout) way of doing this is to *not* actually plot a figure, but to save it in a file as the following segment shows:

```
library(ggplot2)
p <- ggplot(data = midwest,
            mapping = aes(x = area,
                          fill = state,
                          color = state)) +
    geom_density(alpha = 0.3)
## save to file
suppressMessages(ggsave("densities.pdf", p))
```

This file is then included via standard LaTeX commands.

Typeset Code (But Do Not Run It). We can also just show code.

```
xx <- faithful[,"eruptions"]
fit <- density(xx)
plot(fit)
```

This simply used a pandoc bloc started and ended by three backticks, with `r` as the language choice. Similarly, *many* other languages can be typeset directly simply by relying on pandoc.

Single column equations. Authors may use 1- or 2-column equations in their article, according to their preference.

To allow an equation to span both columns, options are to use the `\begin{figure*}...\end{figure*}` environment mentioned above for figures. The `\begin{widetext}...\end{widetext}` environment as shown in equation 1 below is deprecated, but LaTeX commands `\onecolumn` and `\twocolumn` work fine.

Please note that this option may run into problems with floats and footnotes, as mentioned in the cuted package documentation. In the case of problems with footnotes, it may be possible to correct the situation using commands `\footnotemark` and `\footnotetext`.

$$\begin{aligned}(x+y)^3 &= (x+y)(x+y)^2 \\ &= (x+y)(x^2+2xy+y^2) \\ &= x^3+3x^2y+3xy^3+x^3.\end{aligned} \tag{1}$$

References

Allaire J, Cheng J, Xie Y, McPherson J, Chang W, Allen J, Wickham H, Atkins A, Hyndman R, Arslan R (2017a). *rmarkdown: Dynamic Documents for R*. R package version 1.6, URL https://CRAN.R-project.org/package=rmarkdown.

Allaire J, R Foundation, Wickham H, Journal of Statistical Software, Xie Y, Vaidyanathan R, Association for Computing Machinery, Boettiger C, Elsevier, Broman K, Mueller K, Quast B, Pruim R, Marwick B, Wickham C, Keyes O, Yu M (2017b). *rticles: Article Formats for R Markdown*. R package version 0.4.1, URL https://CRAN.R-project.org/package=rticles.

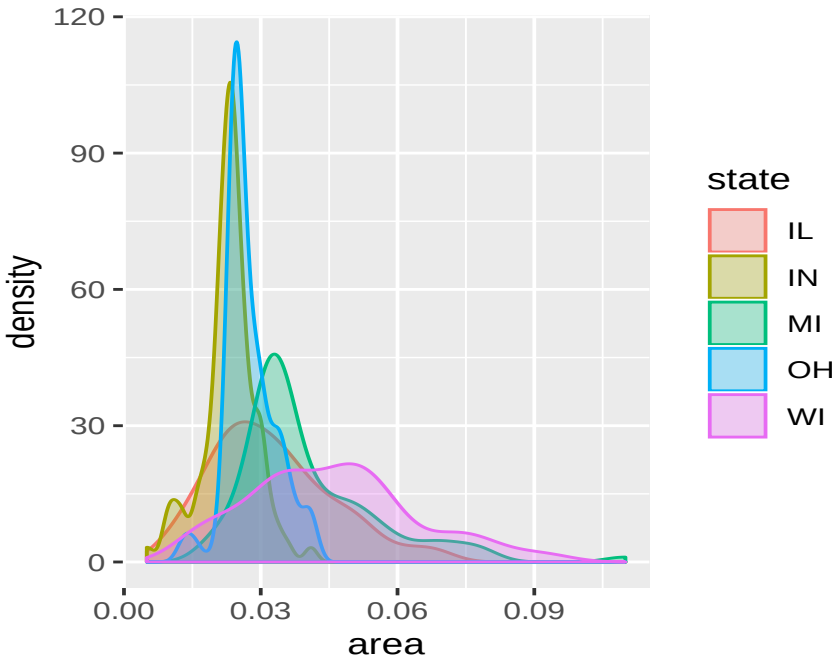

Fig. 4. Wide ggplot2 figure

Eddelbuettel D, Balamuta JJ (2017). "Extending *R* with *C++*: A Brief Introduction to *Rcpp*." *PeerJ Preprints*, **5**, e3188v1. ISSN 2167-9843. doi:10.7287/peerj.preprints.3188v1. URL https://doi.org/10.7287/peerj.preprints.3188v1.

Xie Y (2017). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.17, URL https://yihui.name/knitr/.