

Task 3.2: Thoughts and Data preparation for Dashboard

This document summarises my thoughts for the creation of the dashboard to monitor electricity service requests

Both new requests and open requests are relevant information. New requests are indicative of whether there are systemic problems in certain areas. Open requests add to this, the concept of efficiency in resolving requests.

The simplest way for an executive level person to be able to monitor requests is by looking at the number of requests by date. Line graphs showing the number of requests by date will give a clear indication on if the number requests remain fairly constant, are increasing, or decreasing.

Data Prep

To get to this point the service request data needs to be reshaped into a dataset containing the number of new and open requests by suburb, type and date.

```
library(data.table)
library(ggplot2)
```

Get and read the data

I download the files to the project folder, uncompress them and remove the compressed versions. If the uncompressed files already exist, there is no need to download them.

```
if (file.exists("sr_hex_tr.csv")==FALSE) {
  print("File does not exist. Downloading it...")
  download.file(url='https://cct-ds-code-challenge-input-data.s3.af-south-1.amazonaws.com/sr_hex_truncated.csv', destfile="sr_hex_tr.csv")
}
sr_hex_t <- fread("sr_hex_tr.csv")
```

Data Manipulation

```
#Subset Data to Electricity
elect <- sr_hex_t[department=="Electricity"]

#Format Date Variables to allow it to work the seq()
elect[,CreationDate:= as.Date(CreationDate)]
elect[,CompletionDate:= as.Date(CompletionDate)]

#Deal with missing completion dates and also unnecessary expansion of the panel. We are only interested in the first date
elect[,expandmaxdate:=as.Date(max(CreationDate) + 1 ,origin="1970-01-01")]

# Generate the Time Series of NotificationNumber by Date
noti_ts <- elect[, list(NotificationNumber, date = seq(CreationDate, expandmaxdate, by = "day")), by = .(OfficialSuburbs)]

# Merge in other data on NotificationNumbers
noti_ts <- merge(noti_ts, elect[,.(OfficialSuburbs, Code, CreationDate, CompletionDate, NotificationNumber)], by = c("OfficialSuburbs", "Code"), all = TRUE)
```

```

#Generate Variables to Indicate New and Open requests on each date
noti_ts[,open:= ifelse(date<=CompletionDate | is.na(CompletionDate),1,0)]
noti_ts[,new:= ifelse(date==CreationDate,1,0)]

#Generate the Data for the Dashboard by getting the sum of new and open requests by OfficialSuburbs, Co
dash_data <- noti_ts[, lapply(.SD,sum), by=.(OfficialSuburbs, Code, date), .SDcols = c("new","open")]

## Export Data for Use with Excel for example
write.csv(dash_data, "dash_data.csv")

```

#Example of how Dashboard could look using ggplot

Ideally dropdown Select Buttons should be added. I assume this is doable through Shiny but I did not have the time to figure it out exactly. An Excel Dashboard was created called **Dashboard_Prototype**. The one benefit of excel dashboards is that excel is well known and fairly user friendly. Obviously it also slots in well in the Microsoft Office Suite which is still widely used at executive level.

```
ggplot(dash_data[Code=="No Power" & OfficialSuburbs=="PHILIPPI",]) + aes(y=open, x=date) + geom_line()
```

