R Programing Workshop

Ebude Yolande

This worksheet is for data science and AI beginners and made for a 1:30 mins session. To begin install Rstudio *(download)* or update version.

```
#install.packages("installr")
#library(installr)
#updateR()
```

The following datasets would be used throughout this worksheet:

- IoT sensor dataset *(download)*
- Grid bikeshare data *(download)*

Content

- Dealing with columns in a dataset
- Selecting subsets and merging of datasets.
- Visualization
- Functions and Reusable code

1.Dealing with columns in a dataset

Manipulating data columns, making them easy to interprete and use in analysis.

```
iot<-read.csv('iot_telemetry_data.csv')
#View(iot) #to view the entire dataset
#head(iot,3)
```

Get more information on the dataset/function

```
#summary(iot)
#help("sapply")
```

POSIXct function would convert the object time class to timestamp, *see more*

```
iot$ts<-as.POSIXct(iot$ts,origin = '1970-01-01')
#head(iot,3)
```

The decimal points on the columns with type double.

```
#install.packages(purrr)
library(purrr)
iot<-modify_if(iot, ~is.numeric(.), ~round(., 4))
#tail(iot,3)
```

Exercise I:

We would look at the free bike dataset.

1. Import json file

```
#install.packages("jsonlite")
library("jsonlite")
```

```
##
## Attaching package: 'jsonlite'
```

```
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
BShare<-as.data.frame(fromJSON('free-bike-status-1.json'))
#tail(BShare,4)
```

2. Modify the header

```
names(BShare)<-c('last_updated','ttl','id','name','lon','lat','reserved','disabled')
#head(BShare,3)
```

3. Removing unecessary info

```
#install.packages("stringr")
library(stringr)
new<-list()
for (i in 1:dim(BShare)[1]){  #nrow(BShare)

  val<-strtoi(str_split(BShare$id[i],'_')[[1]][2])

  new<-append(new,val)

}
BShare$new_id<-unlist(new)
#head(BShare,3)
```

4. Save in excel

```
#install.packages("writexl")
library("writexl")
BShare$rec_update<-iot[1:21,'ts']
write_xlsx(BShare, 'bike-share.xlsx')

write_xlsx(iot[1:100,], 'iot.xlsx')
```

Challenge I:

1. Import bike-share excel file, then put the last_updated column to a readable datetime value.
2. Using the iot excel dataset, modify the device column by removing the colon that separate each term (for example 1c:bf:ce:15:ec:4d becomes 1cbfce15ec4d).

2. Selecting subsets and merging of datasets.

Selecting the appropraite subset of a data to use as well as merging different datasets are very important.

```
#iot[1:30,c(1,3,6)]
#head(iot[,c('ts','device', 'smoke')], 30)
```

From the iot_telemery csv dataset, we would select the data of device b8:27:eb:bf:9d:51.

```
iot_d1<-iot[which(iot$device=='b8:27:eb:bf:9d:51'),]
#head(iot_d1,3)
```

We can add more conditions

```
iot_d1<-iot[which(iot$device=='b8:27:eb:bf:9d:51' & iot$co >0.005), c('ts','device','co','temp')]
#head(iot_d1,3)
```

If just device b8:27:eb:bf:9d:51 is known and we dont want its information in the dataset

```
#install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
iot_d2<-filter(iot[ , c('ts','device','co','temp')],iot$device!='b8:27:eb:bf:9d:51')
#tail(iot_d2,3)
```

We would merge the two new datasets to reconstruct the old dataset.

```
iot_d<-rbind(iot_d1,iot_d2)
iot_d<-bind_rows(iot_d1,iot_d2)
```

Exercise II:

1. Creating a vector from taking the last 30 elements in a column ts of iot_d2 and assign to first 30 elements in iot_d1, for example

$$x = [2, 6, 4, 8, 1, 5, 8, 9], new_x = [9, 8, 5, 1]$$

```
iot_d1[1:30,'ts']<-rev(tail(iot_d2$ts,n=30))
#head(iot_d1,3)
```

2.Create a new dataset of iot data merging iot_d1 and iot_d2 based on column ts.

```
iot_d<-merge(iot_d1,iot_d2, by='ts')
#head(iot_d,3)
```

3. Select the iot data which was recorded between 12/07/2020 to 15/07/2020.

```
iot_3d<-iot[which(iot$ts > '2020-07-12 00:00:00' & iot$ts < '2020-07-16 00:00:00'),]
#head(iot_3d,3)
```

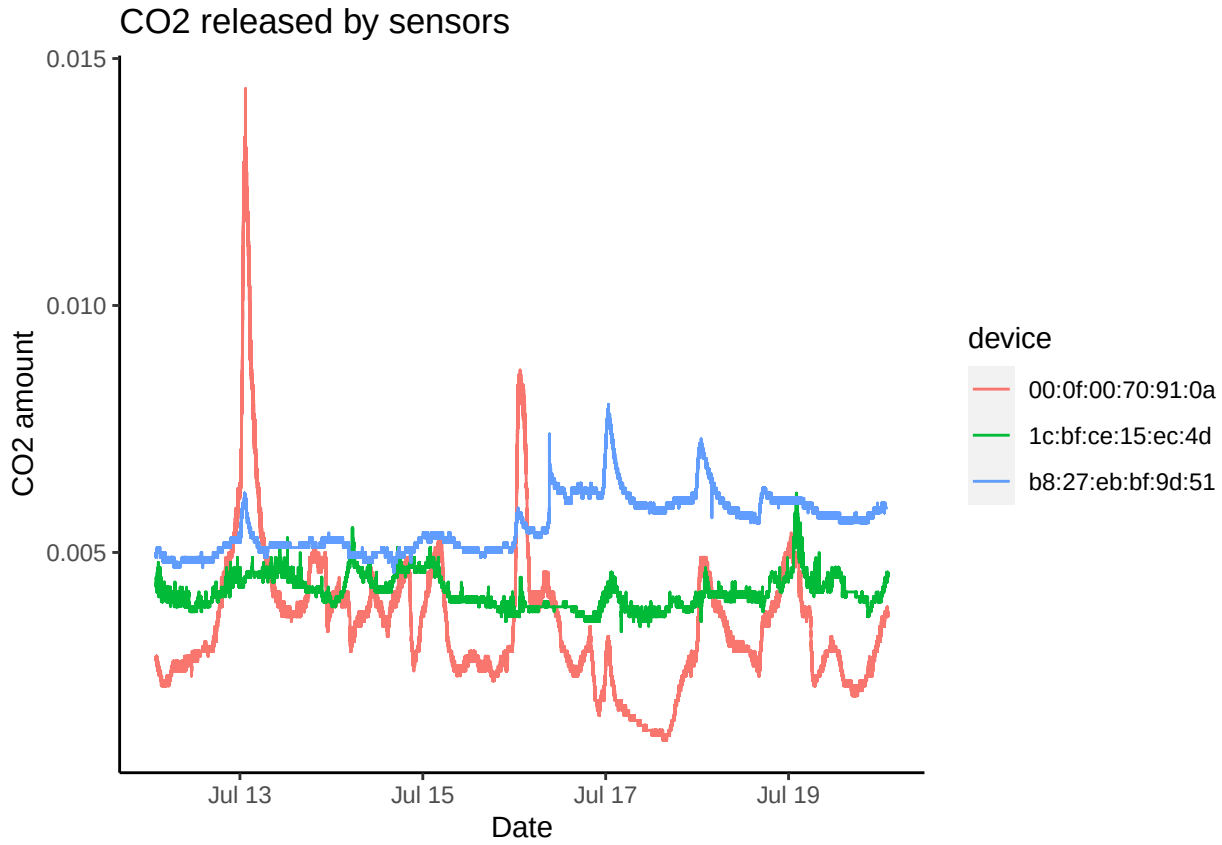
Challenge II

1. Remove the columns last_updated, id, and ttl from the bike share excel dataset.
2. Merge the new bike-share.xlsx and iot.xlsx depending on time
3. Create a new dataset which is the subset of the one in 2. where the co value is greater than the mean value of the iot.xlsx dataset.

3. Visualization

We will start to visualize iot_telemetry, we start with one variable.

```
#install.packages("ggplot2")
library(ggplot2)
qplot(ts,co,data=iot,geom='line',fill=device,color=device,main = "CO2 released by sensors") +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black")) +
xlab("Date")+ ylab("CO2 amount")
```



Exercise III

1. Visualize the relationship between humidity and temperature

```
png('image.png')
qplot(temp, humidity, data=iot,fill=device, color=device,main = "Humidity at various temperatures")
```

2. Distribution/Correlation plots

```
par(mfrow=c(1,2))

dist_p<-ggplot(iot,aes(device,smoke))+geom_boxplot() + theme_bw() + xlab("Device")+ ylab("Smoke") +
  ggtitle('Boxplot of smoke emission of devices')

corr_p<- ggplot(iot, aes(smoke, co))+geom_line()+
  theme(panel.grid.minor = element_blank(), panel.background = element_blank(),
        axis.line = element_line(colour = "black")) + xlab("Smoke")+ ylab("CO2 amount") +
```

```
  ggtitle('Correlation of smoke and Co2 emission')

ggsave('image2.png',dist_p)
```

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

Challenge III:

1. Using iot_telemetry dataset, visualize the distribution of smoke of each device.
2. Visualize the co emission of the devices at different temperature for the days of the week.

4. Functions and Reusable code

Function

$$x = \frac{-b \pm \sqrt{(b^2 - 4ac)}}{2a}$$

```
Quad <- function(a,b,c){
  if (b^2-4*a*c >=0){
    return(c((-b-sqrt(b^2-4*a*c))/(2*a),(-b+sqrt(b^2-4*a*c))/(2*a)))
  }
  else{
    A=-b/(2*a)
    B= sqrt(-(b^2-4*a*c))/ (2*a)
    return(c('complex', A, B))
  }
}
Quad(2,1,0)
```

```
## [1] -0.5  0.0
```

Process

```
con_json_xlsx<- function(js){
  library(jsonlite)
  library("writexl")
  dtf<-as.data.frame(fromJSON(js))
  write_xlsx(dtf,'new_name.xlsx')
}
con_json_xlsx('free-bike-status-1.json')
```

Importing another R code and calling a function from it

```
source('doc.R')
Quad2(2,2,0)
```

```
## [1] -1  0
```