SESSION 2: Introduction to working with R

Assignment 2

Submitted by: Munmun Ghosal
Login Id: munmun55@gmail.com
(M):+91-8007178659

Table of Contents

1. Problem Statement

1. Read multiple json files into a working directory for further converting into a dataset.

   I have files text1, text2, text3 in the directory json.

2. Parse the following JSON into a data frame

```
js<-'{
"name": null, "release_date_local": null, "title": "3 (2011)",
"opening_weekend_take": 1234, "year": 2011, "release_date_wide": "2011-09-16",
"gross": 59954}'
```

3. Write a script for variable binning using R.

2. Solution

1. **Read multiple json files into a working directory for further converting into a dataset.**

The sample json files text1.json, text2.json and text3.json are present in the folder "json" in E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3 ASSIGNMENT\\json

Jsonlite and dplyr packages are installed and then following commands are executed using R-studio:

```
# Reading multiple files using for loop and convert into a dataset
library(jsonlite)
library(dplyr)

ls <- list("E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3
ASSIGNMENT\\json\\text1.json",
        "E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3
ASSIGNMENT\\json\\text2.json",
        "E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3
ASSIGNMENT\\json\\text3.json")
for (i in ls){
  a[i] <- read_json(i, simplifyVector = TRUE)
  z[i] <- data.frame( i,row.names = NULL, check.rows = FALSE,
                check.names = TRUE, fix.empty.names = TRUE,
                stringsAsFactors = default.stringsAsFactors())

  z[i] <- cbind(z[i],a[i])
}
View(a)
View(z)
```

Hence multiple json files are read into the working directory and are then converted into datasets.

The current working directory may be obtained by using getwd()

 # Reading multiple files one by one and convert into a dataset

```
library(jsonlite)
library(dplyr)
y1<-read_json("E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3 ASSIGNMENT\\json\\text1.json")
View(y1)
result1<- as.data.frame(do.call("rbind",y1))
result1

y2<-read_json("E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3 ASSIGNMENT\\json\\text2.json")
View(y2)
result2<- as.data.frame(do.call("rbind",y2))
result2

y3<-read_json("E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3 ASSIGNMENT\\json\\text3.json")
View(y3)
result3<- as.data.frame(do.call("rbind",y3))
result3
```

The output is shown as follows:

```
Console   Terminal
~/
> library(jsonlite)
> library(dplyr)
> y1<-read_json("E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3 ASSIGNMENT\\json\\text1.json")
> View(y1)
> result1<- as.data.frame(do.call("rbind",y1))
> result1
  name species                      foods
1  abx     cat tuna, catnip, ham, zucchini
2  Bcd     dog        bones, carrots, tuna
3  egg     cat           mice, nn, cookies
>
> y2<-read_json("E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3 ASSIGNMENT\\json\\text2.json")
> View(y2)
> result2<- as.data.frame(do.call("rbind",y2))
> result2
   name species                      foods
1   Meo     cat tuna, catnip, ham, zucchini
2 Barky     dog        bones, carrots, tuna
3 Purws     cat                   a, nn, s
>
> y3<-read_json("E:\\munmun_acadgild\\acadgild data analytics\\b3\\ASSIGNMENT\\B3 ASSIGNMENT\\json\\text3.json")
> View(y3)
> result3<- as.data.frame(do.call("rbind",y3))
> result3
      name species                      foods
1   Meowsy     cat tuna, catnip, ham, zucchini
2    Barky     dog        bones, carrots, tuna
3 Purrpaws     cat           mice, nn, cookies
>
```

Show Attributes

Name	Type	Value
y1	list [3]	List of length 3
[[1]]	list [3]	List of length 3
name	character [1]	'abx'
species	character [1]	'cat'
foods	list [2]	List of length 2
likes	list [2]	List of length 2
[[1]]	character [1]	'tuna'
[[2]]	character [1]	'catnip'
dislikes	list [2]	List of length 2
[[1]]	character [1]	'ham'
[[2]]	character [1]	'zucchini'
[[2]]	list [3]	List of length 3
[[3]]	list [3]	List of length 3

Show Attributes

Name	Type	Value
y2	list [3]	List of length 3
[[1]]	list [3]	List of length 3
name	character [1]	'Meo'
species	character [1]	'cat'
foods	list [2]	List of length 2
likes	list [2]	List of length 2
dislikes	list [2]	List of length 2
[[2]]	list [3]	List of length 3
name	character [1]	'Barky'
species	character [1]	'dog'
foods	list [2]	List of length 2
[[3]]	list [3]	List of length 3
name	character [1]	'Purws'
species	character [1]	'cat'
foods	list [2]	List of length 2

Name	Type	Value
y3	list [3]	List of length 3
[[1]]	list [3]	List of length 3
name	character [1]	'Meowsy'
species	character [1]	'cat'
foods	list [2]	List of length 2
likes	list [2]	List of length 2
dislikes	list [2]	List of length 2
[[2]]	list [3]	List of length 3
name	character [1]	'Barky'
species	character [1]	'dog'
foods	list [2]	List of length 2
[[3]]	list [3]	List of length 3
name	character [1]	'Purrpaws'
species	character [1]	'cat'
foods	list [2]	List of length 2

2. Parse the given JSON into a data frame.

The script for parsing the given Jason into a dataframe is shown below:

```
js<-'{
"name": null, "release_date_local": null, "title": "3 (2011)",
"opening_weekend_take": 1234, "year": 2011, "release_date_wide": "2011-09-16", "gross":
59954 }'

mydf <- fromJSON(js)
mydf
```

Here the given Jason is stored in variable named js.
fromJSON() function is used for the parsing the data into dataframe.
The resultant data frame is stored in mydf.

Following output is obtained after executing the script in console:

```
Console   Terminal ×
~/
> js<-'{
+ "name": null, "release_date_local": null, "title": "3 (2011)",
+
+ "opening_weekend_take": 1234, "year": 2011, "release_date_wide": "2011-09-16", "gross"
: 59954 }'
>
> mydf <- fromJSON(js)
> mydf
$`name`
NULL

$release_date_local
NULL

$title
[1] "3 (2011)"

$opening_weekend_take
[1] 1234

$year
[1] 2011

$release_date_wide
[1] "2011-09-16"

$gross
[1] 59954

> View(mydf)
>
```

Show Attributes

Name	Type	Value
mydf	list [7]	List of length 7
name	NULL	Pairlist of length 0
release_date_local	NULL	Pairlist of length 0
title	character [1]	'3 (2011)'
opening_weekend_take	integer [1]	1234
year	integer [1]	2011
release_date_wide	character [1]	'2011-09-16'
gross	integer [1]	59954

3. Write a script for Variable Binning using R.

Binning is the process of transforming numerical variables into categorical counterparts.

Writing binning() function for dividing the variable named age into 4 bins named as "group1-(1 to 25)","group2-(26 to 50)","group3-(51 to 75)","group4-(76 to 90)"

```
age <- c(1:90)
age

binning <- function(x)
{
  for(i in c(1:90))
  {
    ifelse(i <= 25, paste(i,"group1"),
          ifelse(i <= 50, paste(i,"group2"),
                ifelse(i <= 75, paste(i,"group3"),
                      paste(i,"group4"))))
    break
  }
}

binning(age)
```

 # Example 1:Let us consider a vector consisting of values from 1 to 90 and we need to create 4 bins named "group1","group2","group3","group4".

VARIABLE BINNING USING cut() function

```
x<-c(1:90)
cut(x,4,labels=c("group1","group2","group3","group4"))
```

```
Console   Terminal
~/
> x<-c(1:90)
> cut(x,4,labels=c("group1","group2","group3","group4"))
 [1] group1 group1 group1 group1 group1 group1 group1 group1 group1 group1 group1 group1 group1 group1
[15] group1 group1 group1 group1 group1 group1 group1 group1 group1 group2 group2 group2 group2 group2
[29] group2 group2 group2 group2 group2 group2 group2 group2 group2 group2 group2 group2 group2 group2
[43] group2 group2 group2 group3 group3 group3 group3 group3 group3 group3 group3 group3 group3 group3
[57] group3 group3 group3 group3 group3 group3 group3 group3 group3 group3 group3 group4 group4 group4
[71] group4 group4 group4 group4 group4 group4 group4 group4 group4 group4 group4 group4 group4 group4
[85] group4 group4 group4 group4 group4 group4
Levels: group1 group2 group3 group4
>
```

 # Example 2: Import a mtcars.csv file into R-Studio and divide the variable named mpg into 5 bins named "FIRST","SECOND","THIRD","FOURTH" and "FIFTH"

```
library(readr)
mtcars <- read_csv("E:/munmun_acadgild/acadgild data analytics/supporting files/mtcars.csv")
mpg<- mtcars$mpg
mpg
cut(mpg,5)
cut(mpg,5,labels=c("FIRST","SECOND","THIRD","FOURTH","FIFTH"))
```

```
Console   Terminal ×
~/
> library(readr)
> mtcars <- read_csv("E:/munmun_acadgild/acadgild data analytics/supporting files/mtcars.csv")
Parsed with column specification:
cols(
  model = col_character(),
  mpg = col_double(),
  cyl = col_double(),
  disp = col_double(),
  hp = col_double(),
  drat = col_double(),
  wt = col_double(),
  qsec = col_double(),
  vs = col_double(),
  am = col_double(),
  gear = col_double(),
  carb = col_double()
)
> mpg<- mtcars$mpg
> mpg
 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9
[21] 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7 15.0 21.4
> cut(mpg,5)
 [1] (19.8,24.5] (19.8,24.5] (19.8,24.5] (19.8,24.5] (15.1,19.8] (15.1,19.8] (10.4,15.1] (19.8,24.5]
 [9] (19.8,24.5] (15.1,19.8] (15.1,19.8] (15.1,19.8] (15.1,19.8] (15.1,19.8] (10.4,15.1] (10.4,15.1]
[17] (10.4,15.1] (29.2,33.9] (29.2,33.9] (29.2,33.9] (19.8,24.5] (15.1,19.8] (15.1,19.8] (10.4,15.1]
[25] (15.1,19.8] (24.5,29.2] (24.5,29.2] (29.2,33.9] (15.1,19.8] (15.1,19.8] (10.4,15.1] (19.8,24.5]
Levels: (10.4,15.1] (15.1,19.8] (19.8,24.5] (24.5,29.2] (29.2,33.9]
> cut(mpg,5,labels=c("FIRST","SECOND","THIRD","FOURTH","FIFTH"))
 [1] THIRD  THIRD  THIRD  THIRD  SECOND SECOND FIRST  THIRD  THIRD  SECOND SECOND SECOND SECOND SECOND
[15] FIRST  FIRST  FIRST  FIFTH  FIFTH  FIFTH  THIRD  SECOND SECOND FIRST  SECOND FOURTH FOURTH FIFTH
[29] SECOND SECOND FIRST  THIRD
Levels: FIRST SECOND THIRD FOURTH FIFTH
```

In the above example,the value of mtcars$mpg ranges from 10.4 to 33.9.This range is divided into 5 bins with
Levels: (10.4,15.1] (15.1,19.8] (19.8,24.5] (24.5,29.2] (29.2,33.9]
Named as Levels: FIRST SECOND THIRD FOURTH FIFTH

Since the first value of mpg=21.0 lies in the THIRD bin ranging from 19.8 to 24.5; the output for the same is shown as (19.8,24.5] or THIRD.