eMDP: Business Analytics

Computation with R , SQL, Graphs ,HR Data Analysis , Regression

Indian Institute of Management Rohtak

e-MDP

https://www.jcu.edu.au/__data/assets/pdf_file/0005/1265342/Basics-of-R-Studio.pdf

https://sites.tufts.edu/datalab/files/2018/04/R_RStudio_Basics.pdf

https://www.cs.upc.edu/~robert/teaching/estadistica/rprogramming.pdf

https://web.itu.edu.tr/~tokerem/The_Book_of_R.pdf

https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf

Import Data (input.csv)

- data <- read.csv("C:/Users/admin/Desktop/input.csv")

- **data <- read.csv("input.csv")**

data <- read.csv(file.choose(), header=T)

List

Lists

A list is an R-object which can contain many different types of elements inside it like vectors, functions and even another list inside it.

```
list1 <-list(c(2,5,3),21.3,"ramanjeet")
 ..............................
> print(list1)
[[1]]
[1] 2 5 3

[[2]]
[1] 21.3

[[3]]
[1] "ramanjeet"
```

Matrices

Creating a Matrix

```
# Elements are arranged sequentially by row.
M <- matrix(c(3:14), nrow=4, byrow=TRUE)
print(M)

# Elements are arranged sequentially by column.
N <- matrix(c(3:14), nrow=4, byrow=FALSE)
print(N)

# Define the column and row names.
rownames = c("row1", "row2", "row3", "row4")
colnames = c("col1", "col2", "col3")

P <- matrix(c(3:14), nrow=4, byrow=TRUE, dimnames=list(rownames, colnames))

print(P)
```

Data Frames

Data frame is a two dimensional data structure in R.

It contains vectors of equal lengths

Creating Data Frames

```
Console ~/
> id <- 1:5
> age <- c(23,25,22,27,24)
> name <- c("Ram", "Caroline", "Gopal", "Vidushi", "Meldie")
> x <- data.frame(id,age,name)
> x
  id age     name
1  1  23      Ram
2  2  25 Caroline
3  3  22    Gopal
4  4  27  Vidushi
5  5  24   Meldie
```

Operation on data frame

```
x<-
data.frame("roll"=1:5,"name"=c("
jack","jill","jeeva","smith","bob"),
"age"=c(20,22,30,28,21))
```

Sort by name

```
newdata<-x[order(x$name),]
```

Sort by age and with in sort by name

```
newdata<-x[order(x$age,x$name),]
```

Practices

In a given csv file (input.csv)
Get the max salary from data frame.

Get the person detail having max salary.

#Get all the people working in IT department

#Get the persons in IT department whose salary is greater than 600

Write filtered data into a new file.

Import Data (input.csv)

- data <- read.csv("C:/Users/admin/Desktop/input.csv")

- **data <- read.csv("input.csv")**

data <- read.csv(file.choose(), header=T)

Practices

Paste your input.csv file in working directory

```
data <- read.csv("input.csv")
print(data)
# Get the max salary from data frame.
sal <- max(data$salary)
print(sal)
[1] 843.25
Get the details of the person with max salary
# Get the person detail having max salary.
retval <- subset(data, salary == max(salary))
print(retval)
```

Practices

```
#Get all the people working in IT department
retval <- subset( data, dept == "IT")
print(retval)

#Get the persons in IT department whose salary is greater
than 600

info <- subset(data, salary > 600 & dept == "IT")
print(info)

# Write filtered data into a new file.
write.csv(retval,"output.csv")
newdata <- read.csv("output.csv")
print(newdata)
```

SQL ?

https://dept.stat.lsa.umich.edu/~jerrick/courses/stat701/notes/sql.html

popularity of SQL in worldwide analytics / data science industry. According to an online survey conducted by Oreilly Media in 2017, it was found that among all the programming languages, SQL was used by 70% of the respondents followed by R and Python. It was also discovered that people who know Excel (Spreadsheet) tend to get significant salary boost once they learn SQL.
Also, according to a survey done by datasciencecentral, it was inferred that R users tend to get a nice salary boost once they learn SQL. In a way, SQL as a language is meant to complement your current set of skills.

Practising SQL in R

For writing SQL queries, we'll use **sqldf package**. It is one of the most versatile package packages available these days which activate SQL in R.

```
library(sqldf)
sqldf("select count(*) from data")
```

```
[1]  8
```

```
sqldf("select * from data")
```

id	name	salary	start_date	dept
1	Rick	623.30	1/1/2012	IT
2	Dan	515.20	9/23/2013	Operations
3	Michelle	611.00	11/15/2014	IT
4	Ryan	729.00	5/11/2014	HR
5	Gary	843.25	3/27/2015	Finance
6	Nina	578.00	5/21/2013	IT
7	Simon	632.80	7/30/2013	Operations
8	Guru	722.50	6/17/2014	Finance

Name ,dept where sal is less then 623

```
sqldf('SELECT name,dept FROM data WHERE
salary < 623')
```

or

```
 sqldf("select name,dept FROM data WHERE
salary < 623")
```

Or

```
sqldf("select name,Dept From data WHERE
Salary < 623")
```

	name	dept
1	Dan	Operations
2	Michelle	IT
3	Nina	IT

```
sqldf("SELECT name FROM data WHERE
salary < 623 and dept=='IT'")
1Michelle
2     Nina
```

```
sqldf("select * from data where dept!='HR'")
```

```
 id     name salary start_date       dept
1  1     Rick 623.30   1/1/2012         IT
2  2      Dan 515.20  9/23/2013 Operations
3  3 Michelle 611.00 11/15/2014         IT
4  5     Gary 843.25  3/27/2015    Finance
5  6     Nina 578.00  5/21/2013         IT
6  7    Simon 632.80  7/30/2013 Operations
7  8     Guru 722.50  6/17/2014    Finance
```

```
sqldf("select sum(salary) as 'Total_Count' from data")
  or
sqldf("select sum(salary) from data")
> Total_Count          sum(salary)
     5255.05                5255.05
> sqldf("select min(salary), max(salary) from data")
  min(salary) max(salary)
1       515.2      843.25
```

sqldf("select * from data order by name ")

	id	name	salary	start_date	dept
1	2	Dan	515.20	9/23/2013	Operations
2	5	Gary	843.25	3/27/2015	Finance
3	8	Guru	722.50	6/17/2014	Finance
4	3	Michelle	611.00	11/15/2014	IT
5	6	Nina	578.00	5/21/2013	IT
6	1	Rick	623.30	1/1/2012	IT
7	4	Ryan	729.00	5/11/2014	HR
8	7	Simon	632.80	7/30/2013	Operations

sqldf("select * from data where name like 'R%' ")

	id	name	salary	start_date	dept
1	1	Rick	623.3	1/1/2012	IT
2	4	Ryan	729.0	5/11/2014	HR

```
sqldf("select * from data where salary in (515.20,722.50,611.00)")
```

```
  id     name salary start_date       dept
1  2      Dan  515.2  9/23/2013 Operations
2  3 Michelle  611.0 11/15/2014         IT
3  8     Guru  722.5  6/17/2014    Finance
```

```
sqldf("SELECT * FROM data WHERE (salary > 600 AND name like 'R%')")
```

```
  id name salary start_date dept
1  1 Rick  623.3   1/1/2012   IT
2  4 Ryan  729.0  5/11/2014   HR
```

```
sqldf("select * from data where salary > 600 order by name desc ")
```

```
  id     name salary start_date       dept
1  7    Simon 632.80  7/30/2013 Operations
2  4     Ryan 729.00  5/11/2014         HR
3  1     Rick 623.30   1/1/2012         IT
4  3 Michelle 611.00 11/15/2014         IT
5  8     Guru 722.50  6/17/2014    Finance
6  5     Gary 843.25  3/27/2015    Finance
```

Test

- UCBAdmissions

Return Female student admission result

```
sqldf("select * from ucb where Gender = 'Female'")
```

1	Admitted	Female	A 89
2	Rejected	Female	A 19
3	Admitted	Female	B 17
4	Rejected	Female	B 8
5	Admitted	Female	C 202
6	Rejected	Female	C 391
7	Admitted	Female	D 131
8	Rejected	Female	D 244
9	Admitted	Female	E 94
10	Rejected	Female	E 299
11	Admitted	Female	F 24
12	Rejected	Female	F 317

total admitted student's

```
sqldf("select sum(Freq) from
ucb where Admit =
'Admitted'")
##   SUM("Freq")
## 1        1755
```

return total reject females

```
sqldf("select sum(Freq) as total_ladies from ucb where Admit = 'Rejected' AND Gender = 'Female'")
```

```
##   total_ladies
## 1         1278
```

Department wise total admitted students

Hint : Group by dept

```
sqldf("select Dept, sum(Freq) as sum_admitted from ucb where Admit = 'Admitted' group by Dept")
```

	Dept	sum_admitted
1	A	601
2	B	370
3	C	322
4	D	269
5	E	147
6	F	46

Group by admitted and rejected

```
sqldf("select Admit, sum(Freq) as sum_admitted from ucb  group by admit")
```

```
  Admit sum_admitted
1 Admitted        1755
2 Rejected        2771
```

Thank you !!!