Assignment4

Dipesh Poudel

10/4/2021

# Assignment 4: Packages in R & Tidying and Transforming Data in R

**Instructions** Solve/do all the examples/exercises of Unit 2 Session 1 (S6) and Session 2 (S7) slides, copy them to the attached word file with discussion and/or interpretation for review, feedback and grading for your own understanding.

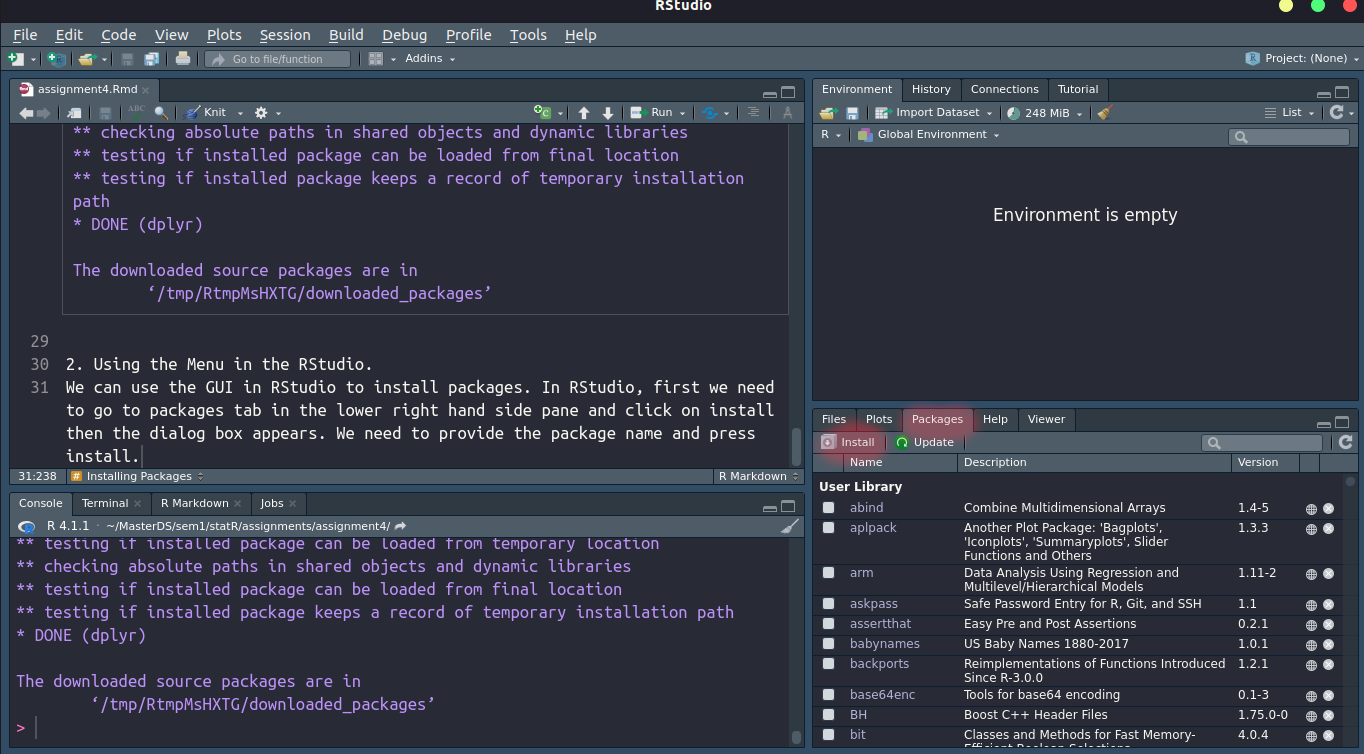
## Installing Packages

There are multiple ways of installing packages in R and R studio.

1. Using install.packages() function: We can use install.packages() function to install packages from CRAN repository.

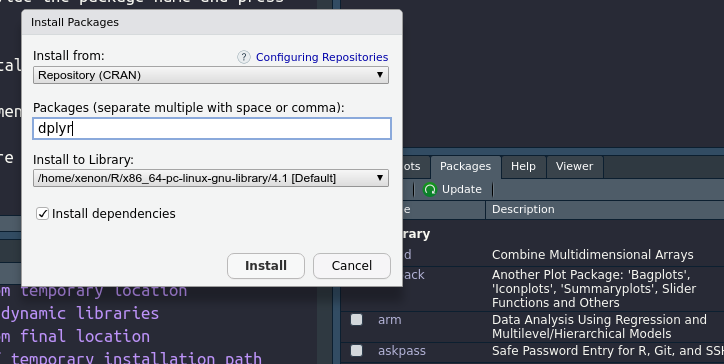
# Example for install.packages  
# Syntax: install.packages('<package\_name>')  
# install.packages('dplyr')

1. Using the Menu in the RStudio. We can use the GUI in RStudio to install packages. In RStudio, first we need to go to packages tab in the lower right hand side pane and click on install then the dialog box appears. We need to provide the package name and press install.



Installing Package from RStudio Menu

In the image above we need to click on the menu highlighted in red. Then we will see a dialog as shown in picture below



Install Package Dialog

1. Using the devtools package. This package is used for installing package from github.

# If not installed uncomment line below   
# install.packages("devtools")  
  
#Taking Instruction from https://rqda.github.io/RQDA/  
#install.packages("devtools") ## install it only if you haven't done it yet  
devtools::install\_github("RQDA/RQDA", INSTALL\_opts = "--no-multiarch")

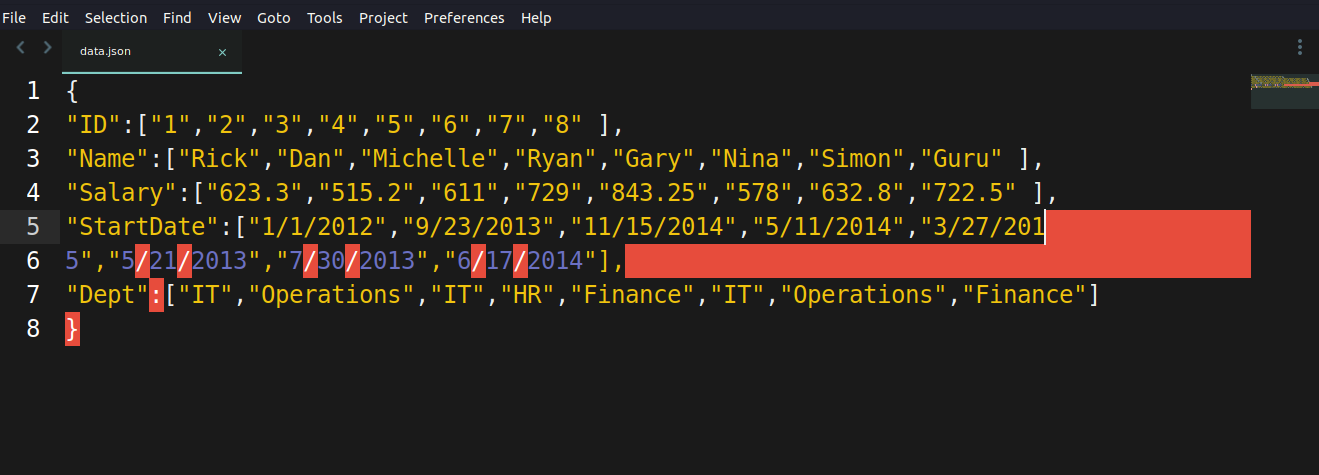
## Skipping install of 'RQDA' from a github remote, the SHA1 (6e25ebe3) has not changed since last install.  
## Use `force = TRUE` to force installation

## Warning: 1 components of `...` were not used.  
##   
## We detected these problematic arguments:  
## \* `INSTALL\_opts`  
##   
## Did you misspecify an argument?

In our case the package was already installed so no need to install for now.

## Reading JSON Data

### Reading JSON data created using a text editor.

The screenshot below shows creation of JSON file in text editor(sublime text) 

library(rjson)  
data<-fromJSON(file="data.json")  
print(data)

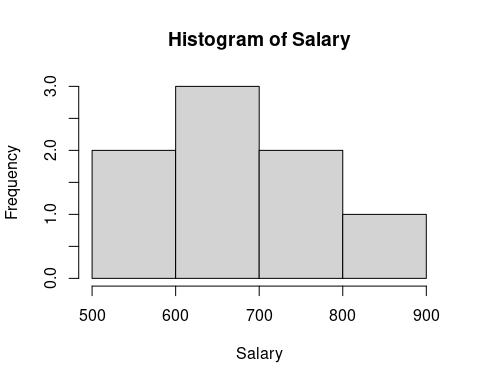
## $ID  
## [1] "1" "2" "3" "4" "5" "6" "7" "8"  
##   
## $Name  
## [1] "Rick" "Dan" "Michelle" "Ryan" "Gary" "Nina" "Simon"   
## [8] "Guru"   
##   
## $Salary  
## [1] "623.3" "515.2" "611" "729" "843.25" "578" "632.8" "722.5"   
##   
## $StartDate  
## [1] "1/1/2012" "9/23/2013" "11/15/2014" "5/11/2014" "3/27/2015"   
## [6] "5/21/2013" "7/30/2013" "6/17/2014"   
##   
## $Dept  
## [1] "IT" "Operations" "IT" "HR" "Finance"   
## [6] "IT" "Operations" "Finance"

# Converting the JSON data into a dataframe  
json\_data\_frame<-as.data.frame(data)  
print(json\_data\_frame)

## ID Name Salary StartDate Dept  
## 1 1 Rick 623.3 1/1/2012 IT  
## 2 2 Dan 515.2 9/23/2013 Operations  
## 3 3 Michelle 611 11/15/2014 IT  
## 4 4 Ryan 729 5/11/2014 HR  
## 5 5 Gary 843.25 3/27/2015 Finance  
## 6 6 Nina 578 5/21/2013 IT  
## 7 7 Simon 632.8 7/30/2013 Operations  
## 8 8 Guru 722.5 6/17/2014 Finance

#### Histogram of Salary

# Converting the salary into Numeric data type  
json\_data\_frame$Salary<-as.numeric(json\_data\_frame$Salary)  
hist(json\_data\_frame$Salary,main="Histogram of Salary",xlab = "Salary")

 #### Average salary by department

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

json\_data\_frame%>%group\_by(Dept)%>%summarise(avg\_salary=mean(Salary))

## # A tibble: 4 × 2  
## Dept avg\_salary  
## <chr> <dbl>  
## 1 Finance 783.  
## 2 HR 729   
## 3 IT 604.  
## 4 Operations 574

#### Frequency distribution of all variables

# install.packages('plyr')  
library(plyr)

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

col\_list<-names(json\_data\_frame[,-1])  
for (i in 1:length(col\_list)){  
cat("Frequency and Pecentage For",col\_list[i],"\n")  
df\_count<-count(json\_data\_frame[col\_list[i]])  
print(df\_count)  
}

## Frequency and Pecentage For Name   
## Name freq  
## 1 Dan 1  
## 2 Gary 1  
## 3 Guru 1  
## 4 Michelle 1  
## 5 Nina 1  
## 6 Rick 1  
## 7 Ryan 1  
## 8 Simon 1  
## Frequency and Pecentage For Salary   
## Salary freq  
## 1 515.20 1  
## 2 578.00 1  
## 3 611.00 1  
## 4 623.30 1  
## 5 632.80 1  
## 6 722.50 1  
## 7 729.00 1  
## 8 843.25 1  
## Frequency and Pecentage For StartDate   
## StartDate freq  
## 1 1/1/2012 1  
## 2 11/15/2014 1  
## 3 3/27/2015 1  
## 4 5/11/2014 1  
## 5 5/21/2013 1  
## 6 6/17/2014 1  
## 7 7/30/2013 1  
## 8 9/23/2013 1  
## Frequency and Pecentage For Dept   
## Dept freq  
## 1 Finance 2  
## 2 HR 1  
## 3 IT 3  
## 4 Operations 2

### Reading JSON file from URL: Web API

#install.packages(“jsonlite”)  
library(jsonlite)

##   
## Attaching package: 'jsonlite'

## The following objects are masked from 'package:rjson':  
##   
## fromJSON, toJSON

Raw <- fromJSON("https://data.ny.gov/api/views/9a8c-vfzj/rows.json?accessType=DOWNLOAD")  
food\_market <- Raw[['data']]  
Names <- food\_market[,14]  
head(Names)

## [1] "STEWARTS SHOP 389" "MAN IN THE MOON" "J M L DELI"   
## [4] "EZ WHITE STREET" "QUICKLEES 012" "180 LA DELI GROCERY"

# table(Names)

#table(food\_market$V19)

#### What is atomic vector?

An atomic vector is a one dimensional data object in R. It is created using the c or vector function

#### Converting the atomic vector to dataframe for analysis

team\_seed <- data.frame()  
team\_seed <- cbind(food\_market)   
head(food\_market)

## [,1] [,2] [,3]  
## [1,] "row-bmva-tkb5-2dx7" "00000000-0000-0000-90F2-26A64FA6E755" "0"   
## [2,] "row-2cqz\_dfmj~m5gb" "00000000-0000-0000-4C32-933D3D828E69" "0"   
## [3,] "row-yf74-k3xm~pajw" "00000000-0000-0000-F593-7EC8D32A16E1" "0"   
## [4,] "row-y39z-m2eu-6z2z" "00000000-0000-0000-9530-45D17AE44F42" "0"   
## [5,] "row-4zcr-722r-y9fi" "00000000-0000-0000-C192-5C6A048F84FB" "0"   
## [6,] "row-84xv.rwmz\_s6g2" "00000000-0000-0000-F06C-FA33AEBA3A64" "0"   
## [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]   
## [1,] "1630695187" NA "1630695201" NA "{ }" "Greene" "190060" "Store"  
## [2,] "1630695187" NA "1630695201" NA "{ }" "Oswego" "735781" "Store"  
## [3,] "1630695187" NA "1630695201" NA "{ }" "Kings" "740047" "Store"  
## [4,] "1630695187" NA "1630695201" NA "{ }" "Dutchess" "741000" "Store"  
## [5,] "1630695187" NA "1630695201" NA "{ }" "Steuben" "706669" "Store"  
## [6,] "1630695187" NA "1630695201" NA "{ }" "Bronx" "705401" "Store"  
## [,12] [,13] [,14] [,15]   
## [1,] "JAC" "STEWARTS SHOPS CORP" "STEWARTS SHOP 389" "4834"   
## [2,] "JAC" "LEAR AMY MARIE T" "MAN IN THE MOON" "192"   
## [3,] "JAC" "J M L DELI CORP" "J M L DELI" "297"   
## [4,] "JAC" "EZ WHITE STREET CORP" "EZ WHITE STREET" "31"   
## [5,] "JAC" "PEMM LLC" "QUICKLEES 012" "349-353"  
## [6,] "JAC" "180 LA DELI GROCERY CORP" "180 LA DELI GROCERY" "180"   
## [,16] [,17] [,18] [,19] [,20] [,21] [,22]   
## [1,] "RT 81 PO BOX 240" NA NA "GREENVILLE" "NY" "12083" "2900"  
## [2,] "W 1ST ST" NA NA "OSWEGO" "NY" "13126" "6000"  
## [3,] "T S BOYLAND ST." NA NA "BROOKLYN" "NY" "11233" "0"   
## [4,] "SOUTHWHITE STREET" NA NA "POUGHKEEPSIE" "NY" "12601" "1000"  
## [5,] "W MORRIS ST" NA NA "BATH" "NY" "14810" "1500"  
## [6,] "MCLELLAN ST" NA NA "BRONX" "NY" "10456" "1100"  
## [,23] [,24]  
## [1,] NA NA   
## [2,] "POINT (-76.511143 43.456378)" "240"  
## [3,] NA NA   
## [4,] NA NA   
## [5,] "POINT (-77.333015 42.338397)" "46"   
## [6,] NA NA