

Lab 1

Lin Cheng

7/2/2019

1 Installing R & RStudio

Proof of Installation

```
system("who", intern=TRUE)
```

```
## [1] "Caroline.C console Jul 1 13:27 "
```

The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
1 ---
2 title: "Lab 1"
3 author: "Lin Cheng"
4 date: "7/2/2019"
5 output: html_document
6 ---
7 #1
8 ##problem 1
9 ![Screen Capture](lab1proof.png)
```

Below the script editor, a message indicates: (No image at path lab1proof.png).

The console at the bottom shows the R startup output:

```
R version 3.6.0 (2019-04-26) -- "Planting of a Tree"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]
```

The environment pane on the right shows the Global Environment with the following variables:

Variable	Class	Values
m	num [1:4, 1:2]	3.14 4.14 5.14 6.1...
a	int [1:4, 1:3, 1:2]	1 2 3 4 5 6 7 8...
even5	num [1:5]	0 2 4 6 8
even500	num [1:500]	0 2 4 6 8 10 12 14 16 1...
ones	num [1:20]	1 1 1 1 1 1 1 1 1 ...
pi		3.14159265358979
v	int [1:24]	1 2 3 4 5 6 7 8 9 10 ...

The Files pane on the right shows the R documentation for the `rep` function, titled "Replicate Elements of Vectors and Lists".

Screen Capture

2 Installing Packages and Loading them using library()

Proof of Installation

```
find.package("tidyverse")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.6/Resources/library/tidyverse"
```

```
find.package("readr")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.6/Resources/library/readr"
```

```
find.package("foreign")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.6/Resources/library/foreign"
```

3 Application of Basic R Rules

3.1 Working with Vectors and some vector math

```
(79-32)*(5/9)
```

```
## [1] 26.11111
```

```
nhtemp
```

```
## Time Series:  
## Start = 1912  
## End = 1971  
## Frequency = 1  
## [1] 49.9 52.3 49.4 51.1 49.4 47.9 49.8 50.9 49.3 51.9 50.8 49.6 49.3 50.6  
## [15] 48.4 50.7 50.9 50.6 51.5 52.8 51.8 51.1 49.8 50.2 50.4 51.6 51.8 50.9  
## [29] 48.8 51.7 51.0 50.6 51.7 51.5 52.1 51.3 51.0 54.0 51.4 52.7 53.1 54.6  
## [43] 52.0 52.0 50.9 52.6 50.2 52.6 51.6 51.9 50.5 50.9 51.7 51.4 51.7 50.8  
## [57] 51.9 51.8 51.9 53.0
```

```
m<-as.numeric(nhtemp)  
nhtemp.celsius<-(m-32)*(5/9)  
nhtemp.celsius
```

```
## [1] 9.944444 11.277778 9.666667 10.611111 9.666667 8.833333 9.888889
## [8] 10.500000 9.611111 11.055556 10.444444 9.777778 9.611111 10.333333
## [15] 9.111111 10.388889 10.500000 10.333333 10.833333 11.555556 11.000000
## [22] 10.611111 9.888889 10.111111 10.222222 10.888889 11.000000 10.500000
## [29] 9.333333 10.944444 10.555556 10.333333 10.944444 10.833333 11.166667
## [36] 10.722222 10.555556 12.222222 10.777778 11.500000 11.722222 12.555556
## [43] 11.111111 11.111111 10.500000 11.444444 10.111111 11.444444 10.888889
## [50] 11.055556 10.277778 10.500000 10.944444 10.777778 10.944444 10.444444
## [57] 11.055556 11.000000 11.055556 11.666667
```

3.2 Character (String) and Logical Vectors

```
a<-c("wrath","avarice","sloth","pride","lust","envy","gluttony")
a
```

```
## [1] "wrath"      "avarice"    "sloth"      "pride"      "lust"       "envy"
## [7] "gluttony"
```

```
a[6:7]
```

```
## [1] "envy"      "gluttony"
```

Note: `b+3` does not result in an error because `TRUE` and `FALSE` were perceived as numeric value 1 and 0, respectively, thus the binary operator “+” can be used. `d`, on the other hand, puts “`TRUE`” and “`FALSE`” in a matrix in the form of strings (non-numeric) and therefore “+” cannot be used.

3.3 Matrices

3 different ways to create the same matrix

Method 1

```
b<-t(matrix(49:20,nrow=5))
b
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  49  48  47  46  45
## [2,]  44  43  42  41  40
## [3,]  39  38  37  36  35
## [4,]  34  33  32  31  30
## [5,]  29  28  27  26  25
## [6,]  24  23  22  21  20
```

Method 2

```
a1<-49:45
a2<-44:40
a3<-39:35
a4<-34:30
a5<-29:25
a6<-24:20
b<-rbind(a1,a2,a3,a4,a5,a6)
b
```

```
##      [,1] [,2] [,3] [,4] [,5]
## a1  49  48  47  46  45
## a2  44  43  42  41  40
## a3  39  38  37  36  35
## a4  34  33  32  31  30
## a5  29  28  27  26  25
## a6  24  23  22  21  20
```

Method 3

```
a1<-seq(49,24,-5)
a2<-seq(48,23,-5)
a3<-seq(47,22,-5)
a4<-seq(46,21,-5)
a5<-seq(45,20,-5)
b<-cbind(a1,a2,a3,a4,a5)
b
```

```
##      a1 a2 a3 a4 a5
## [1,] 49 48 47 46 45
## [2,] 44 43 42 41 40
## [3,] 39 38 37 36 35
## [4,] 34 33 32 31 30
## [5,] 29 28 27 26 25
## [6,] 24 23 22 21 20
```

Display the values in the 3rd row only

```
b[3,]
```

```
## a1 a2 a3 a4 a5
## 39 38 37 36 35
```

Display the values in the 2nd and 4th columns only

```
cbind(b[,2],b[,4])
```

```
##      [,1] [,2]
## [1,]  48  46
## [2,]  43  41
## [3,]  38  36
## [4,]  33  31
## [5,]  28  26
## [6,]  23  21
```

3.4 Data Frames

```
data(mtcars)
class(mtcars)
```

```
## [1] "data.frame"
```

```
mode(mtcars)
```

```
## [1] "list"
```

```
str(mtcars,list.len=0)
```

```
## 'data.frame':   32 obs. of  11 variables:
##  [list output truncated]
```

i The data for California

```
USArrests[5,]
```

```
##           Murder Assault UrbanPop Rape
## California      9      276      91 40.6
```

ii All values for UrbanPop

```
USArrests[,3]
```

```
## [1] 58 48 80 50 91 78 77 72 80 60 83 54 83 65 57 66 52 66 51 67 85 74 66
## [24] 44 70 53 62 81 56 89 70 86 45 44 75 68 67 72 87 48 45 59 80 80 32 63
## [47] 73 39 66 60
```

Alternative way:

```
USArrests["UrbanPop"][,1]
```

iii All variables for state with lowest murder rate

```
USArrests[34,]
```

```
##           Murder Assault UrbanPop Rape
## North Dakota    0.8      45      44  7.3
```

iv Obs. with a UrbanPop >= 85

```
rbind(USArrests[5,],USArrests[21,],USArrests[30,], USArrests[32,],USArrests[39,])
```

```
##           Murder Assault UrbanPop Rape
## California    9.0      276      91 40.6
## Massachusetts  4.4      149      85 16.3
## New Jersey     7.4      159      89 18.8
## New York      11.1      254      86 26.1
## Rhode Island   3.4      174      87  8.3
```

Sidenote about USArrests

```
head(USArrests)
```

```
##           Murder Assault UrbanPop Rape
## Alabama    13.2      236      58 21.2
## Alaska     10.0      263      48 44.5
## Arizona     8.1      294      80 31.0
## Arkansas    8.8      190      50 19.5
## California   9.0      276      91 40.6
## Colorado    7.9      204      78 38.7
```

```
head(ChickWeight)
```

```
##      weight Time Chick Diet
## 1      42     0     1     1
## 2      51     2     1     1
## 3      59     4     1     1
## 4      64     6     1     1
## 5      76     8     1     1
## 6      93    10     1     1
```

```
myUSA<-USArrests
myUSA$NAME<-row.names(USArrests)
head(myUSA)
```

```
##           Murder Assault UrbanPop Rape      NAME
## Alabama      13.2      236      58 21.2  Alabama
## Alaska       10.0      263      48 44.5   Alaska
## Arizona       8.1      294      80 31.0   Arizona
## Arkansas      8.8      190      50 19.5   Arkansas
## California    9.0      276      91 40.6 California
## Colorado      7.9      204      78 38.7   Colorado
```

```
row.names(myUSA)<-1:nrow(myUSA)
head(myUSA)
```

```
##      Murder Assault UrbanPop Rape      NAME
## 1      13.2      236      58 21.2  Alabama
## 2      10.0      263      48 44.5   Alaska
## 3       8.1      294      80 31.0   Arizona
## 4       8.8      190      50 19.5   Arkansas
## 5       9.0      276      91 40.6 California
## 6       7.9      204      78 38.7   Colorado
```

```
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
```

```
myUSA2<-USArrests%>%mutate(NAME2=row.names(USArrests))
head(myUSA2)
```

```
##      Murder Assault UrbanPop Rape      NAME2
## 1      13.2      236        58 21.2    Alabama
## 2      10.0      263        48 44.5     Alaska
## 3       8.1      294        80 31.0     Arizona
## 4       8.8      190        50 19.5     Arkansas
## 5       9.0      276        91 40.6    California
## 6       7.9      204        78 38.7     Colorado
```

4 Putting it all together: Writing Functions

A. Writing Pseudocode

|Tempconv (F) //Function name, F is the input temperatue in Fehrenheit |If F is not numeric, show error message. Else, continue. |Convert input temperature F to C (Celsius) | $C=(F-32)*(5/9)$ //Formula of conversion |Output C //C is the output temperature in Celsius

B. Implementing the pseudocode

```
Tempconv<-function(F){
  if(!is.numeric(F))
    stop("please input a numeric vector")
  (F-32)*(5/9)
}
```