

Week1_Homework_Solution

Problem 1

Get familiar with your working directory: • What is your working directory?

```
getwd()
```

- Change it to your desktop

```
setwd("/Users/Username/Desktop") # Mac OS X  
setwd("C:\\Username\\Desktop") # Windows 7+  
list.dir(path = "/Users/Username/Desktop") # list directories at path
```

Problem 2

Give the commands to do the following: • Install the knitr package

```
install.packages('knitr') # single package name is automatically made a vector  
install.packages(c('ggplot2', 'ggviz')) # multiple packages installed at the same time
```

- Load knitr into your R environment

```
library("knitr")
```

- Bonus: Unloading a library

```
detach("package:knitr", unload = TRUE)
```

Problem 3

Search for help • Bring the html help page up for function called “c” • Bring the html help page up for function called “sum”

```
help(c) # get help on a known function  
help('sum') # alternative style 1  
?sum # shorthand style  
??sum # search for all functions with 'sum' in the name
```

- Find all the functions which contain the word of “plot”

```
??plot
apropos('plot', mode='function')
```

- Run the example function for “plot”

```
example("plot")
```

Problem 4

What kind of object CANNOT be placed in a list object? • list • matrix • data.frame • function

```
list.of.all = list(mylist=list(c(1,2)), twobytwo=matrix(rep(1:2, 2)))
# start off the list
list.of.all[['research']] = data.frame(apples=1:4, bananas=5:8)
list.of.all[['square']] = function(x) { return(x * x) }
# Test
summary(list.of.all)
```

```
##           Length Class      Mode
## mylist    1      -none-    list
## twobytwo  4      -none-    numeric
## research  2      data.frame list
## square    1      -none-    function
```

Let's see if a list can include the function “square”

```
list.of.all[['square']] # reference to a function
```

```
## function(x) { return(x * x) }
```

We can even use this function as usual!

```
list.of.all [['square']](4) # and we can call it like normal
```

```
## [1] 16
```

Problem 5

- Generate a 5 x 4 data.frame that looks like this:

```
## labels X6.10 X11.15 X16.20 X21.25
## 1 one 6 11 16 21
## 2 two 7 12 17 22
## 3 three 8 13 18 23
## 4 four 9 14 19 24
## 5 five 10 15 20 25
```

There are many ways of thing this. Think of others!

```
labels <- c('one', 'two', 'three', 'four', 'five')
my_frame <- data.frame(labels, 6:10, 11:15, 16:20, 21:25)
my_frame
```

```
## labels X6.10 X11.15 X16.20 X21.25
## 1 one 6 11 16 21
## 2 two 7 12 17 22
## 3 three 8 13 18 23
## 4 four 9 14 19 24
## 5 five 10 15 20 25
```

- Do a summary, what did you find?

```
summary(my_frame)
```

```
## labels X6.10 X11.15 X16.20 X21.25
## five :1 Min. : 6 Min. :11 Min. :16 Min. :21
## four :1 1st Qu.: 7 1st Qu.:12 1st Qu.:17 1st Qu.:22
## one :1 Median : 8 Median :13 Median :18 Median :23
## three:1 Mean : 8 Mean :13 Mean :18 Mean :23
## two :1 3rd Qu.: 9 3rd Qu.:14 3rd Qu.:19 3rd Qu.:24
## Max. :10 Max. :15 Max. :20 Max. :25
```

Problem 6

Review the slides, and understand how to subset a dataframe with either \$ and/or [row_condition, column_condition]. Using the people(with temperature info) data.frame provided below, find:

- average temperature of city (using the \$ sign and [,2] interchangeably)

```
mean(people$temp, na.rm = TRUE)
```

```
## [1] 23.25
```

```
mean(people[, 4], na.rm = TRUE)
```

```
## [1] 23.25
```

- the records whose city=shanghai, and sex=F

```
people[(people$sex=='F' & people$city=='shanghai'), ]
```

```
##           city age sex temp  
## 4 shanghai  32   F   14
```

- for all the females, how many of them live in shanghai,chengdu

```
condition <- people$sex == 'F' & (people$city == 'shanghai' | people$city == 'cheng  
du')  
sum(ifelse(condition, 1, 0))
```

```
## [1] 2
```

- for all the males, how many of them is older than 20.

```
condition <- people$sex == 'M' & people$age > 20  
sum(ifelse(condition, 1, 0))
```

```
## [1] 2
```

- how many of the city has records whose temperature was between 20 and 30

```
condition <- people$temp >= 20 & people$temp <= 30  
sum(ifelse(condition, 1, 0), na.rm = TRUE)
```

```
## [1] 3
```