

Unit 4- Introduction to R
Homework 1 of 2

Due: Wednesday September 30, 2020

Last date for submission with -10 points: Friday October 2, 2020

Last date for submission for credit (-20 points): Wednesday October 7, 2020

Introduction

This homework gives you practice working with objects and some selected basics.

How to do this assignment:

- __1. Launch R Studio.
- __2. Open a new R script file. Immediately save it under the name YOURLASThw4.R
- __3. Print out a hard copy of this assignment and try your hand at replicating all the commands. In doing these, type them into your R script file and use <control> ENTER to send them to the console window for execution. Tip – I do this one line at a time, or perhaps a few. When errors occur, simply make your edits in your script file and try again.
- __4. When you're all done, save your final "perfect" R script file. EDIT/COPY/PASTE into a word document. *I encourage you to annotate your word document with any comments or questions you like!*
- __5. Print your word document to a PDF
- __6. Upload your PDF homework to the ASSIGNMENT tab in Blackboard

... Don't forget to have fun with this...

Preliminaries. User edits yellow shaded

```
setwd("/Users/cbigelow/Desktop")
getwd()
```

Load packages we will use

```
library(summarytools) # command used: descr()
```

#1. Create Some Objects

Tip for beginners: Include specification of object type (character, numeric, etc)

FYI – As you get more comfortable, no need to use as.integer, as.character, etc.

#1a. Vector of integers using as.integer() and seq()

```
v_id <- as.integer(seq(from=1, to=10, by=1))
v_id
```

#1b. Vector of characters using as.character()

```
v_name <- as.character(c("Bob", "Jean", "Peg", "John", "Carol", "Larry", "William", "Susanne", "Martha", "Ed"))
v_name
```

#1c. Vector of numbers using as.numeric()

```
v_age <- as.numeric(c(87.8,92.0,53.5,66.4,64.2,67.0,27.1,56.6,58.1,70.2))
v_age
```

#1d. Vector corresponding to a categorical variable is a factor in R, using as.factor()

```
v_happy <- as.factor(c("high", "high", "high", "medium", "high", "medium", "medium", "high", "medium", "low"))
v_happy
```

1e. Data Frame using data.frame()

```
df1 <- data.frame(v_id,v_name,v_age,v_happy)
df1
```

#1f. Table from table, row by row, using as.table() and rbind()

```
table1 <- as.table(rbind(c(59,48),c(11,462)))
dimnames(table1) <- list(
  TEST=c("Positive","Negative"),
  DISEASE=c("Diabetes","Non_diabetes"))
table1
```

#1g. List using list()

```
list1 <- list(v_name,df1,table1)
list1
```

#2. Write some R code yourself! Check your answers using R functions

#2a. SAMPLE MEAN

```
nsize <- length(v_age)          # sample size using length()
xbar <- sum(v_age)/nsize         # sample mean
```

Compare R code with function mean() and command descr() in the package summarytools

```
paste("User R code solution for mean = ", xbar)
paste("R function mean( ) solution for mean = ", mean(v_age))
descr(v_age, stats = c("n.valid","mean"))
```

#2b. SAMPLE STANDARD DEVIATION

```
sample_var <- sum((v_age-xbar)^2)/(nsize-1)    # sample variance
sample_sd <- sqrt(sample_var)                 # sample standard deviation
```

Compare R code with function sd() and command descr() in the package summarytools

```
paste("User R code solution for standard deviation = ", sample_sd)
paste("R function sd( ) solution for sample standard deviation = ", sd(v_age))
descr(v_age, stats = c("n.valid","mean","sd"))
```

#3. EXPLORING WHAT COULD GO WRONG – Factors

PROBLEM: Factors are stored as integers "under the hood", alphabetically!!!

For example, In R, ("low", "medium", "high") is stored as (2, 3, 1). This may not be what we want.

How to fix the storage of an ordinal factor variable

Example: We want "low"=1, "medium"=2, "high"=3

Solution: Use option ordered=TRUE

```
v_happynew <- factor(v_happy, levels=c("low", "medium", "high"), ordered=TRUE)
str(v_happy)
str(v_happynew)
```

```
# How to fix the storage of a nominal factor variable
# Example 2: We want 1="winter", 2="spring", 3="summer", 4="fall"
# Solution: Create a vector containing YOUR desired levels using c()
v_season <- factor(c("winter", "summer", "summer", "fall", "fall", "spring"))
paste("Factor levels are NOT what we want")
sort(v_season)
# Solution
mylevels <- c("winter", "spring", "summer", "fall")
v_seasonnew <- factor(v_season, levels=mylevels)
paste("Factor levels Fixed!")
sort(v_seasonnew)
```

4. EXPLORING WHAT COULD GO WRONG - Missing values

PROBLEM: If your data has missing values (NA) you get an error message

```
v_glitch <- c(87.8,92.0,NA,66.4,64.2,67.0,27.1,56.6,58.1,70.2)
paste("my vector has missing values and I did not implement a solution")
mean(v_glitch)
```

Solution: Use option na.rm=TRUE to tell R to remove missing values from the calculation

```
paste("Fixed!")
mean(v_glitch, na.rm=TRUE)
```