R Workshop Handout

Melissa Ko

Useful Libraries

```
library(ggplot2)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2019a.
## 1.0/zoneinfo/America/New_York'
```

Useful Functions

Basic Numeric Operations

```
x <- 10
y <- 3
x == y
```

```
## [1] FALSE
```

```
x != y
```

```
## [1] TRUE
```

```
identical(x, y)
```

```
## [1] FALSE
```

```
x > y
```

```
## [1] TRUE
```

```
x < y
```

```
## [1] FALSE
```

```
x ^ y
```

```
## [1] 1000
```

```
x %% y
```

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

Numeric Operations

```
num <- 1.49184
floor(num)
```

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

```
ceiling(num)
```

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

```
round(num)
```

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

```
signif(num)
```

```
## [1] 1.49184
```

```
signif(num, digits = 4)
```

```
## [1] 1.492
```

```
signif(num, digits = 2)
```

```
## [1] 1.5
```

```
signif(num, digits = 1)
```

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

Building Numeric Vectors

```
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
2:8
```

```
## [1] 2 3 4 5 6 7 8
```

```
seq(1, 10, by = 2)
```

```
## [1] 1 3 5 7 9
```

```
rev(1:10)
```

```
##  [1] 10  9  8  7  6  5  4  3  2  1
```

```
rep(1:5, times = 2)
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5
```

```
rep(1:5, each = 2)
```

```
##  [1] 1 1 2 2 3 3 4 4 5 5
```

```
runif(10)
```

```
##  [1] 0.5899524 0.3530244 0.9566667 0.7195624 0.8589717 0.5806758 0.9343715
##  [8] 0.8745518 0.2149528 0.9421794
```

```
runif(5, min = 1, max = 100)
```

```
## [1] 86.69128 30.65478 76.98846 85.37404 11.57310
```

```
many_num <- runif(100)
head(many_num)
```

```
## [1] 0.4052027 0.9179402 0.7516361 0.7437117 0.9340249 0.7317192
```

```
head(many_num, n = 10)
```

```
##  [1] 0.40520270 0.91794021 0.75163606 0.74371166 0.93402489 0.73171916
##  [7] 0.72710303 0.53943432 0.06841647 0.60328586
```

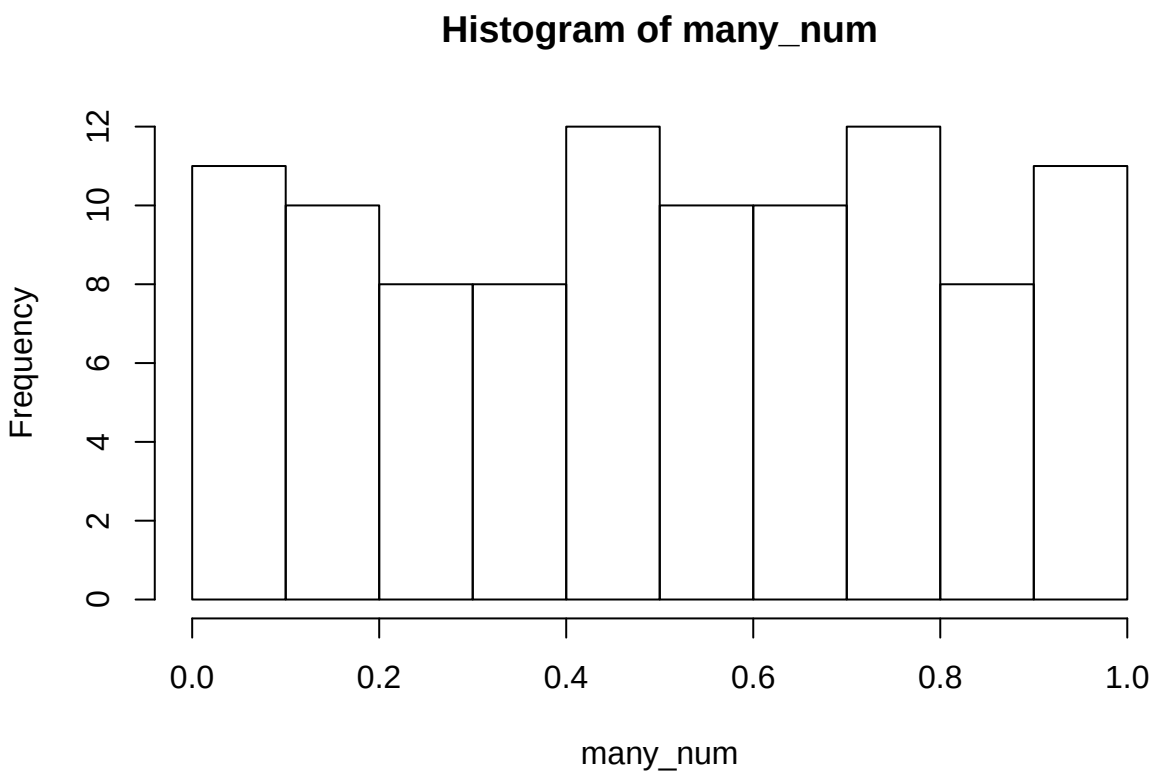
```
tail(many_num)
```

```
## [1] 0.2930864 0.3568447 0.8656073 0.4907471 0.2907348 0.2253714
```

```
tail(many_num, n = 10)
```

```
##  [1] 0.8253191 0.4461598 0.4081952 0.2210148 0.2930864 0.3568447 0.8656073
##  [8] 0.4907471 0.2907348 0.2253714
```

```
hist(many_num)
```



Numeric Vector Operations

```
values <- c(1, 5, 2, 6, 8, 9)
values + 1
```

```
## [1]  2  6  3  7  9 10
```

```
values * 2
```

```
## [1]  2 10  4 12 16 18
```

```
sample(values, size = 1)
```

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

```
sample(values, size = 3)
```

```
## [1] 9 8 1
```

```r
temp <- sample(values, size = 50, replace = TRUE)
temp
```

```
##  [1] 6 8 9 1 6 6 5 9 5 8 5 1 1 1 9 6 5 8 6 6 1 2 8 1 2 8 2 2 5 2 9 6 8 1 1
## [36] 9 2 5 9 8 1 5 2 2 9 6 2 8 8 2
```

```r
table(temp)
```

```
## temp
##  1  2  5  6  8  9
##  9 10  7  8  9  7
```

```r
min(values)
```

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

```r
max(values)
```

```
## [1] 9
```

```r
sort(values)
```

```
## [1] 1 2 5 6 8 9
```

```r
length(values)
```

```
## [1] 6
```

```r
order(values)
```

```
## [1] 1 3 2 4 5 6
```

```r
sum(values)
```

```
## [1] 31
```

```r
mean(values)
```

```
## [1] 5.166667
```

Indexing into a Numeric Vector

```r
values[3]
```

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

```r
values[c(1, 3, 4)]
```

```
## [1] 1 2 6
```

```r
values[order(values)]
```

```
## [1] 1 2 5 6 8 9
```

```r
which(values > 3)
```

```
## [1] 2 4 5 6
```

```r
values[which(values > 3)]
```

```
## [1] 5 6 8 9
```

Building and Indexing into Character Vectors

```
vec1 <- c("a", "b", "d", "x")
vec2 <- c("c", "b", "f", "x", "y", "z")

rev(vec1)
```

```
## [1] "x" "d" "b" "a"
```

```
vec1[1:3]
```

```
## [1] "a" "b" "d"
```

```
vec1[seq(2, 4, by = 2)]
```

```
## [1] "b" "x"
```

```
rep(vec1, times = 2)
```

```
## [1] "a" "b" "d" "x" "a" "b" "d" "x"
```

```
rep(vec1, each = 2)
```

```
## [1] "a" "a" "b" "b" "d" "d" "x" "x"
```

```
which(vec1 == c("a", "b"))
```

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

```
vec1[which(vec1 == c("a", "b"))]
```

```
## [1] "a" "b"
```

```
vec3 <- c(vec1, "a", "x", "b", "c")
vec4 <- c(vec1, vec2)

vec3
```

```
## [1] "a" "b" "d" "x" "a" "x" "b" "c"
```

```
vec4
```

```
##  [1] "a" "b" "d" "x" "c" "b" "f" "x" "y" "z"
```

```
unique(vec3)
```

```
## [1] "a" "b" "d" "x" "c"
```

```
unique(vec4)
```

```
## [1] "a" "b" "d" "x" "c" "f" "y" "z"
```

Set Operations

```
union(vec1, vec2)
```

```
## [1] "a" "b" "d" "x" "c" "f" "y" "z"
```

```
intersect(vec1, vec2)
```

```
## [1] "b" "x"
```

```
setdiff(vec1, vec2)
```

```
## [1] "a" "d"
```

```
setdiff(vec2, vec1)
```

```
## [1] "c" "f" "y" "z"
```

Building and Indexing into Matrices

```
data <- matrix(values, nr = 2, byrow = TRUE)
data
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    2
## [2,]    6    8    9
```

```
data[1, ]
```

```
## [1] 1 5 2
```

```
data[, 2]
```

```
## [1] 5 8
```

```
data[2, 3]
```

```
## [1] 9
```

```
nrow(data)
```

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

```
ncol(data)
```

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

```
data2 <- rbind(values, values + 1)
data2
```

```
##        [,1] [,2] [,3] [,4] [,5] [,6]
## values    1    5    2    6    8    9
##           2    6    3    7    9   10
```

```
data3 <- cbind(values, values * 2)
data3
```

```
##      values
## [1,]      1  2
## [2,]      5 10
## [3,]      2  4
## [4,]      6 12
## [5,]      8 16
## [6,]      9 18
```

Building and Indexing into Dataframes

```r
data4 <- as.data.frame(data2)
data4
```

```
##        V1 V2 V3 V4 V5 V6
## values  1  5  2  6  8  9
##         2  6  3  7  9 10
```

```r
class(data4)
```

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

```r
colnames(data4)
```

```
## [1] "V1" "V2" "V3" "V4" "V5" "V6"
```

```r
rownames(data4)
```

```
## [1] "values" ""
```

```r
colnames(data4) <- c("A", "B", "C", "D", "E", "F")
rownames(data4) <- c("Alice", "Bob")
data4
```

```
##       A B C D E  F
## Alice 1 5 2 6 8  9
## Bob   2 6 3 7 9 10
```

```r
data4[, 2]
```

```
## [1] 5 6
```

```r
data4[2, ]
```

```
##     A B C D E  F
## Bob 2 6 3 7 9 10
```

```r
data4[2, 4]
```

```
## [1] 7
```

```r
data4$A
```

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

```r
data4$E
```

```
## [1] 8 9
```

Building and Modifying Strings

```r
vec1_string <- paste(vec1, collapse = "")
vec2_string <- paste(vec2, collapse = "")
vec1_string
```

```
## [1] "abdx"
```

```r
vec2_string
```

```
## [1] "cbfxyz"
```

```r
paste("hello", "friend")
```

```
## [1] "hello friend"
```

```r
paste("hello", "friend", sep = "")
```

```
## [1] "hellofriend"
```

```r
paste(vec1, vec2, sep = "")
```

```
## [1] "ac" "bb" "df" "xx" "ay" "bz"
```

```r
paste(vec1_string, vec2_string, sep = "")
```

```
## [1] "abdxcbfxyz"
```

```r
vec5 <- toupper(vec4)
vec5
```

```
##  [1] "A" "B" "D" "X" "C" "B" "F" "X" "Y" "Z"
```

```r
tolower(vec5)
```

```
##  [1] "a" "b" "d" "x" "c" "b" "f" "x" "y" "z"
```

```r
nchar("hello")
```

```
## [1] 5
```

```r
toupper("hello")
```

```
## [1] "HELLO"
```

```r
tolower("HeLlO")
```

```
## [1] "hello"
```

Breaking Apart a Single String

```r
substring("hello", first = 1, last = 2)
```

```
## [1] "he"
```

```r
substring("hello", first = c(1, 3), last = c(2, 4))
```

```
## [1] "he" "ll"
```

```r
grep("o", "hello")
```

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

```r
grep("h", "hello")
```

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

```r
length("hello")
```

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

```r
grep("x", "hello")
```

```
## integer(0)
```

```r
unlist(strsplit("hello", split = ""))
```

```
## [1] "h" "e" "l" "l" "o"
```

```r
unlist(strsplit("hello friend", split = " "))
```

```
## [1] "hello"  "friend"
```

```r
unlist(strsplit("hello friend", split = "e"))
```

```
## [1] "h"       "llo fri" "nd"
```

Searching and Modifying Multiple Strings

```r
save_string <- c("h", "e", "l", "l", "o")
grep("o", save_string)
```

```
## [1] 5
```

```r
grep("x", save_string)
```

```
## integer(0)
```

```r
grep("l", save_string)
```

```
## [1] 3 4
```

```r
"o" %in% save_string
```

```
## [1] TRUE
```

```r
"x" %in% save_string
```

```
## [1] FALSE
```

```r
save_words <- c("cat", "dog", "bird", "fish")
grep("i", save_words)
```

```
## [1] 3 4
```

```r
save_words[grep("i", save_words)]
```

```
## [1] "bird" "fish"
```

```r
gsub("e", "a", "hello")
```

```
## [1] "hallo"
```

```r
gsub("i", "e", save_words)
```

```
## [1] "cat"  "dog"  "berd" "fesh"
```

```r
chartr(old = "id", new = "er", x = save_words)
```

```
## [1] "cat"  "rog"  "berr" "fesh"
```

Writing Loops and Conditional Logic

```r
num <- 10
if (num %% 2 == 0) {
  print("even")
} else {
  print("odd")
}
```

```
## [1] "even"
```

```r
for (x in 1:10) {
  print(x)
  if (x %% 2 == 0) {
    print("even")
  } else {
    print("odd")
  }
}
```

```
## [1] 1
## [1] "odd"
## [1] 2
## [1] "even"
## [1] 3
## [1] "odd"
## [1] 4
## [1] "even"
## [1] 5
## [1] "odd"
## [1] 6
## [1] "even"
## [1] 7
## [1] "odd"
## [1] 8
## [1] "even"
## [1] 9
## [1] "odd"
## [1] 10
## [1] "even"
```

```r
all_x <- 1:10
all_x[which(all_x %% 2 == 0)]
```

```
## [1]  2  4  6  8 10
```

```r
i <- 1
while (i < 10) {
  print(i)
  i <- i + 1
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
```

```
## [1] 7
## [1] 8
## [1] 9
```

Writing Functions

```
add_one <- function(num) {
  new_num <- num + 1
  return(new_num)
}

add_one(10)
```

```
## [1] 11
```

```
add_one(-38)
```

```
## [1] -37
```

```
add_one(3849723)
```

```
## [1] 3849724
```

```
values
```

```
## [1] 1 5 2 6 8 9
```

```
add_one(values)
```

```
## [1]  2  6  3  7  9 10
```

```
data
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    2
## [2,]    6    8    9
```

```
add_one(data)
```

```
##      [,1] [,2] [,3]
## [1,]    2    6    3
## [2,]    7    9   10
```

```
apply(data, 1, sum)
```

```
## [1]  8 23
```

```
apply(data, 2, sum)
```

```
## [1]  7 13 11
```

Generating a Dataset

```
full_names <- c("Cecilia Zahia", "Youko Jordon",
                "Lara Burkhard", "Sasha Nsonowa",
                "Ziv Kumbukani", "Ling Jaci")
age <- c(21, 28, 41, 34, 23, 19)
height <- sample(140:180, size = length(age))
```

```r
color_options <- c("blue", "red", "orange", "yellow",
                   "green", "purple", "pink", "black",
                   "white", "grey")
favcolor <- sample(color_options, size = length(age), replace = TRUE)
major <- c("Aero", "Anthro", "Physics",
           "Math", "Dance", "PoliSci")
state <- sample(state.abb, size = length(age))

dataset <- cbind(age, height, favcolor, major, state)
dataset <- as.data.frame(dataset)
rownames(dataset) <- full_names
dataset
```

```
##                age height favcolor   major state
## Cecilia Zahia   21    168    green    Aero    IA
## Youko Jordon    28    155    green  Anthro    ND
## Lara Burkhard   41    144   orange Physics    CO
## Sasha Nsonowa   34    179     grey    Math    NY
## Ziv Kumbukani   23    147      red   Dance    DE
## Ling Jaci       19    176     blue PoliSci    WI
```

```r
summary(dataset)
```

```
##  age    height    favcolor     major   state
##  19:1   144:1   blue  :1   Aero   :1   CO:1
##  21:1   147:1   green :2   Anthro :1   DE:1
##  23:1   155:1   grey  :1   Dance  :1   IA:1
##  28:1   168:1   orange:1   Math   :1   ND:1
##  34:1   176:1   red   :1   Physics:1   NY:1
##  41:1   179:1               PoliSci:1   WI:1
```

Exercises

Pull out the ages and heights of people in `dataset` and find which people have an age or height that is an even number.

```r
# WRITE CODE BELOW
```

List the names of the people in `dataset` in order of increasing age.

```r
# WRITE CODE BELOW
```

Add additional rows to the `dataset` dataframe that consist of information about you and your partner.

```r
# WRITE CODE BELOW
```

Using `full_names`, create a matrix where the first column contains the first name and the second column contains the last name.

```
# WRITE CODE BELOW
```

Write a function that takes in a text string like the elements of `full_names` and returns the string broken up into blocks of three letters.

```
# WRITE CODE BELOW
```

Create a new matrix containing only the `height` and `age` columns from `dataset` and then calculate the average of both columns.

```
# WRITE CODE BELOW
```

Scramble the values of each column in `dataset` and create a variable called `dataset2` that contains this scrambled dataframe.

```
# WRITE CODE BELOW
```

Pull out the names of people in `dataset` that have the same major, favorite color, or home state as you.

```
# WRITE CODE BELOW
```

Design your own task!

```
# WRITE CODE BELOW
```