#Spring2019-597 HW1 Solution

#1.weights 60, 72, 34, 56, 87, 80, 89, 95, 76, 28, 48, 59

weights <- c(60, 72, 34, 56, 87, 80, 89, 95, 76, 28, 48, 59)

# vector "weights"

weights\_hat <- mean(weights)

# weights\_hat = 65.33333

weights\_hat2 <- mean(weights^2)

# weights\_hat2 = 4694.667

n <- length(weights)

# n = 12

#How many weights are larger than 55.

subn <- length(weights[weights > 55])

# subn = 9

#(e). each weight is larger than 55 and smaller than 85

subn2 <- length(weights[weights > 55 & weights < 85])

# subn2 = 6

#2.tmp <- matrix(rnorm(12), 3, 4)

tmp <- matrix(rnorm(12), 3, 4)

tmp[,1] + tmp[,3]

#-0.1344367 -0.3216501 -0.7461074

tmp[1,] \* tmp[3,]

#-0.4679875 0.5165831 1.1667667 0.5026109

dim(tmp)

#3 4

tmp1 <- tmp[1,]

cat(tmp1[tmp1 > 0.5],"\n")

#another way

cat(tmp[1,tmp[1,] > 0.5],"\n")

# outprint = 0.6140815 0.7695796

#3.Check whether two vectors are the same if they may contain missing (NA)

x <- c(7,9,NA,NA,13)

y <- c(7,9,NA,NA,13)

all(is.na(x) == is.na(y)) && all((x == y)[! is.na(x)])

#another method

(x==y)&(!is.na(x) & !is.na(y))|(is.na(x) & is.na(y))

## length(x) == length(y) & all(is.na(x) == is.na(y)) & all(x[!is.na(x)] == y[!is.na(y)]) # my answer

#4. If x is a factor with n levels and y is a length n vector, what happens if you compute y[x]?

x <- factor(c("c","d","d","a","b"))

x

y <- c(1, 2, 3, 4)

y

y[x]

#y[x] = 3 4 4 1 2

#comment: we use this example to show that y[x] = c(y[i])

n = 100

x <- factor(1:n, levels = 1:n)

y <- c(n:1)

y[x]

#5

f <- function(a){

l = (nchar(a) - nchar(gsub("cg","",a))) / nchar("cg")

return(list("counts the number of cg" = l, "replace all cg with XY"= gsub("cg","XY",a)))

}

mydna <- paste(sample(c('a','t','c','g'),1000,replace=T),collapse='')

f(mydna)

#6. write a function and output rows containing valid phone numbers from the ﬁle.

#complete function

mydat <- read.table('http://www.ams.sunysb.edu/~pfkuan/Teaching/AMS597/Data/PhoneNumber.txt',sep='\n')

subsetPhone <- function(dat){

p1 <- "[0-9]{3}[:.:]|-|[[:space:]][0-9]{3}[:.:]|-|[[:space:]][0-9]{4}"

p2 <- "([0-9]{3})[[:space:]][0-9]{3}-[0-9]{4}"

myfunc <- function(x){

tmp1 <- grepl(p1,x)

tmp2 <- grepl(p2,x)

return(max(tmp1,tmp2))

}

return(dat[apply(mydat,1,myfunc)==1,])

}

subsetPhone(mydat)

file <- file("http://www.ams.sunysb.edu/~pfkuan/Teaching/AMS597/Data/PhoneNumber.txt", open="r")

lines <- readLines(file)

pattern.1 = '\\d\\d\\d-\\d\\d\\d-\\d\\d\\d\\d'

pattern.2 = '\\(\\d\\d\\d\\)\\s\\d\\d\\d-\\d\\d\\d\\d'

pattern.3 = '\\d\\d\\d\\s\\d\\d\\d\\s\\d\\d\\d\\d'

pattern.4 = '\\d\\d\\d\\.\\d\\d\\d\\.\\d\\d\\d\\d'

for (i in 1:length(lines)){

if (grepl(pattern.1,lines[i])) print(lines[i])

if (grepl(pattern.2,lines[i])) print(lines[i])

if (grepl(pattern.3,lines[i])) print(lines[i])

if (grepl(pattern.4,lines[i])) print(lines[i])

}

}