Stats380 exam

Kerui Du

04/11/2021

Q1

###(a)

```
(1:10)^(1:2)
```

```
##  [1]   1   4   3  16   5  36   7  64   9 100
```

###(b)

```
seq(1,19,by=3)
```

```
## [1]  1  4  7 10 13 16 19
```

###(c)

```
(1:18)[-seq(2,17,by=3)]
```

```
##  [1]  1  3  4  6  7  9 10 12 13 15 16 18
```

###(d)

```
rep(1:9,rep(3:1,3))
```

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

###(e)

```
1:16 %/% 5
```

```
##  [1] 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3
```

Q2

(a)

X[X>=-1 & X<=1]

(b)

X[X<mean(X,na.rm=T)]

(c)

X[3:length(X)-2]

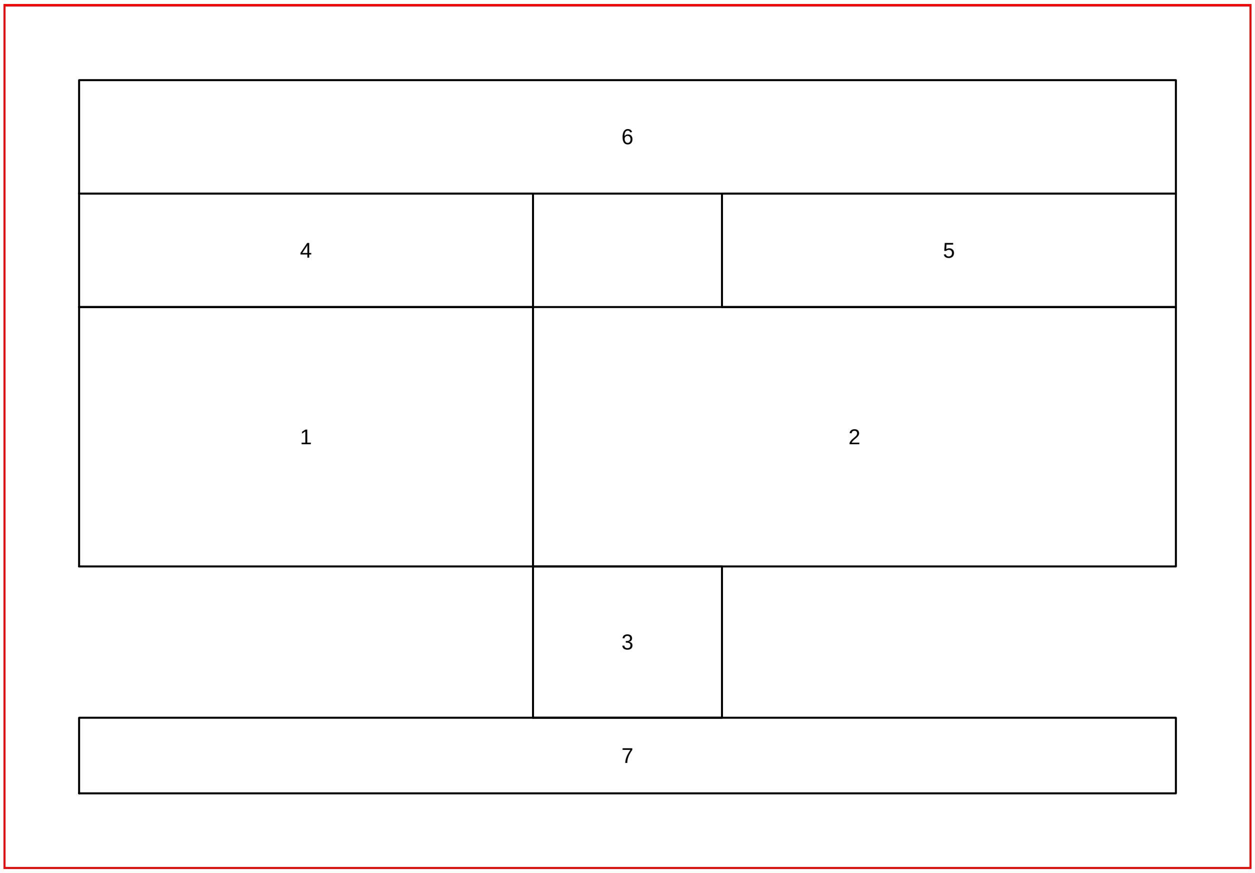
Q3

###(a)

```
# funQ3a = function{
#   max_n = apply(matrix(1:9,3),2,max)
#   min_n = apply(matrix(1:9,3),2,min)
#   return((a+b)/2)
# }
```

Q4

```
 mat = matrix(c(0, 0, 0, 0, 0,
                0, 6, 6, 6, 0,
                0, 4, 0, 5, 0,
                0, 1, 2, 2, 0,
                0, 0, 3, 0, 0,
                0, 7, 7, 7, 0,
                0, 0, 0, 0, 0), nc=5, byrow=T);
layout(mat,widths = c(lcm(1), 1, lcm(2.5), 1, lcm(1)),
       heights = c(lcm(1), lcm(1.5), lcm(1.5), 1, lcm(2), lcm(1),lcm(1)))
layout.show(7)
box('o', lty= 'solid', col='red', lwd = 2)
```



Q5

```
# plot(dat$flush.dist, cex = 0.75, xlim = c(0, 650), ylim = c(0, 200),
#      xlab = 'Individuals', ylab = 'Distance (m)', type= 'b', col='blue')
# points(dat$flush.dist, cex = 0.75, pch = 16, col = 'red')
# lines(dat$flush.dist, cex = 0.75, pch =16, col = 'red')
# legend('topright', c('Flushing distance', 'Landing distance'),lty = c(1,1),
#        pch = c(1, 16), col = c('blue','red'), bty = 'n',cex = 0.75)
# text(x = 170, y = 200, labels = 'Oystercatchers')
# lines(x = c(-20, 670), y = c(max(dat$flush.dist),max(dat$flush.dist)), lty = 2)
```

Q6

(a) grepl(“red”, text):

The grepl function determines whether if the ‘red’ exist in each element of text,

if it is, return TRUE, otherwise return FALSE

(b) gsub("<.*?>“,”", text):

if any element in the text contains the pattern that start from ‘<’ end with ‘>’ and each character replace by ’ ’

(c) strsplit(text, “[,|.]”)

each element in the text split by ‘,’ or ‘.’

(d) regmatches(text, gregexpr(“[1]+”, text)) = ""

find 1 or more length of number from start of each element in the text and these numbers replace by ’ ’. The data structure of text wouldn’t chanage

(e) sapply(strsplit(text, “\.”), length)

split each element of text at ‘\ and a letter’, and count the number of elements after splitting and create a list which contains these counts.

Q7

```
# HTMLTable <- function(text) {
#   readin = readLines(text)
#   index1 = grep('<table',readin)
#   index2 = grep('/table>',readin)
#   paste(readin[index1:index2],collapse = '')
# }
```

Q8

```
row1=' <tr><th> Country </th><th> Population </th> <th> Yearly Change </th><th>Density </th> </tr> '
row2=' <td> 2021 </td><td> New Zealand </td> <td> Auckland </td> '
processRow <- function(row) {
  mod1 = strsplit(row,'<th>|<td>')[[1]][-1]
  mod2 = sub(' <.+$','',mod1)
  mod3 = sub('^ ','',mod2)
  mod3
}
processRow(row1)
```

```
## [1] "Country"       "Population"    "Yearly Change" "Density"
```

[1] 0-9

```
processRow(row2)
```

```
## [1] "2021"        "New Zealand" "Auckland"
```

Q9

```
# group.stats <- function(data) {
#   factors = as.factor(data$group)
#   means = aggregate(cbind(x1,x2,x3)~group,mean,data=df)
#   cov = lapply(levels(factors), function(x) cov(subset(data,group==x)[-1]))
#   list(means=means,cov=cov)
# }
```

Q10

```
# df = sapply(oceania, readTable.country)
# mod_df = lapply(1:length(oceania), function(i) cbind(Country=oceania[i],df[[i]][1:2]))
# f_df = do.call(rbind,mod_df)
```

Q11

```
# final_df=(with(f_df,tapply(Population, list(Country,as.factor(Year)), mean)))
# final_df = as.data.frame(final_df)
# final_df=cbind(Country=rownames(final_df),final_df)
# rownames(final_df)=1:nrow(final_df);final_df
```