STATS380: Statistical Computing

2016 Exam

AceNighJohn

Question 1

Question 1a

```
1:4 * 2
```

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

Question 1b

```
sum(100:1 > 80)
```

```
## [1] 20
```

Question 1c

```
(1:10)[2:1>1]
```

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

Question 1d

```
diff(cumsum(1:5))
```

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

Question 1e

```
pmax(1:10, 10:1)
```

```
##  [1] 10  9  8  7  6  6  7  8  9 10
```

Question 1f

```
{s = 0; for(i in 1:5) s = s - i; s}
```

```
## [1] -15
```

Question 1g

```
paste(substring("380", 1, 3:1), collapse = "")
```

```
## [1] "380383"
```

Question 1h

```
levels(factor(c("green","blue", "green", "brown")))[2]
```

```
## [1] "brown"
```

Question 2

Question 2a

```
seq(1, 28, by = 3)
```

Question 2b

```
rep(1:5, 5:1)
```

Question 2c

```
rep(1:5, each = 2) * 10 + 1:2
```

Question 3

```
maxjump = function(x){
  D = diff(x)
  pos = which.max(D)
  x[c(pos, pos+1)]
}
```

Question 4

```
nn = function(x, y){
  D = rowSums(abs(sweep(x, 2, y)))
  row = which.min(D)
  x[row, ]
}
```

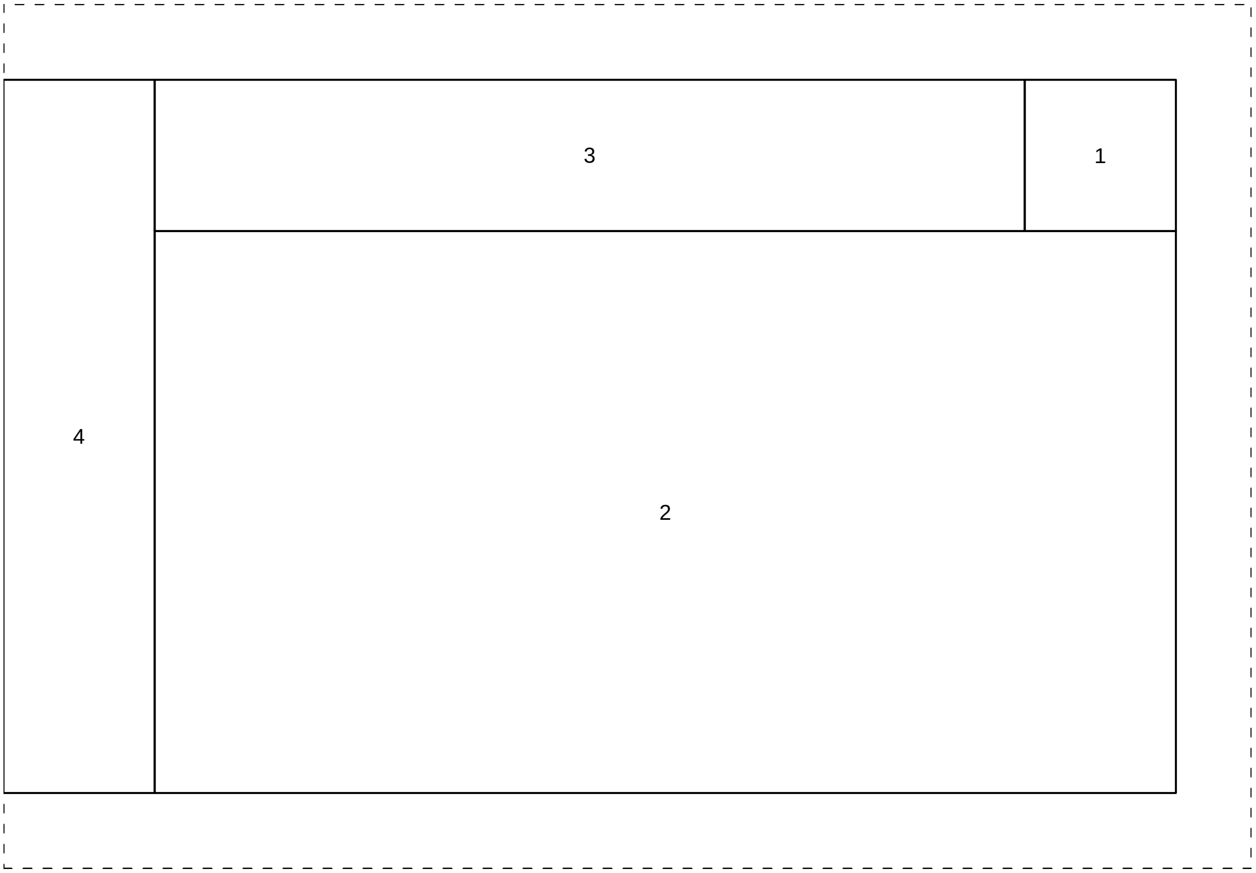
Question 5

Question 5a

```
cols = c("red", "green", "blue")
i = as.numeric(Iris$Species)
with(Iris, plot(Petal.Length, Petal.Width, pch=i, col=cols[i]))
legend(1, 2.5, levels(Iris$Species), pch=1:3, col=cols)
```

Question 5b

```
layout(rbind(c(0, 0, 0, 0),
             c(4, 3, 1, 0),
             c(4, 2, 2, 0),
             c(0, 0, 0, 0)),
height = c(lcm(1), lcm(2), 1, lcm(1)),
width = c(lcm(2), 1, lcm(2), lcm(1)))
layout.show(4)
box("outer", lty = "55")
```



Question 6

#4: We define our own function called processLine, its input argument is a character vector and output is a data frame.

#5: CountryIndex is a list with additional attribute information.

#6: countryName is a list of that contains a country's name that is extracted using gregexpr() and regmatches() in lines 5 and 6.

#7: subpatt is a character vector containing the country name followed by ,\\1 ,\\2.
Eg: "Afghanistan, \\1, \\2"

#8: datapatt is a character vector which will be used in regular expressions, it has a pattern of: E[Division, {(a word)}] -> Q[(a sequence of numbers), People]. Notice where the parentheses are located, we will extract these using \\1 and \\2.

#9: dataVals is a list that contains two consecutive <| and |> and all the characters in between (note the ? implies non greedy matching).

#10: dataValstr is the same as dataVals except its first 2 characters and its last 2 characters removed (which are the <| and |>). dataValstr is a character vector due to the substr function.

#11: dataValsub is a character vector with country name, division name, population size (since we used \\1 and \\2 in the gsub). Eg: "Afghanistan, Badakhshan, 805500" is the start.

#12: dataValsub is split at the occurrence of ", ". This results in just the country name, division and population. The result is assigned to the symbol dataValsplit. dataValsplit is a character vector like in line 2.

#13: The result from line 12 is stored in a matrix which has 3 columns, and is filled row wise. This matrix is assigned to dataValmat.

#14: The columns of dataValmat are given the names “country”, “divsion” and “population”. No names are specified for the rows.

#15: dataValmat is transformed into a data frame, where each column is a character vector. This dataframe is assigned to the symbol dataValframe.

#16: The population column of dataValframe is converted to a numeric vector.

#17: Our processLine function returns dataValframe.

Question 7

```
popdata = do.call("rbind", lapply(splitText[dataLines], processLine))
```

Question 8

```
popBig = subset(popdata, population > 10000000)
```

Question 9

```
countryPopulation = aggregate(population ~ country, data = popdata, FUN = sum)
```

Question 10

```
Largest.row = which.max(countryPopulation$population)
popLargest = countryPopulation[Largest.row, ]
```