

R assignment 2

Joanna Galgano

14/11/2018

Task 1

1

```
lawyers <- read.csv('Lawyers.csv', head = T)
lit <- nrow(lawyers[lawyers$Practice == 'Litigation',])
other <- nrow(lawyers[lawyers$Practice,])
round(lit/other, 2)
```

```
## [1] 0.58
```

The proportion of lawyers practicing litigation law is 0.58.

2

I was unsure about the intention of this question so done it two ways:

1) proportions of corporate lawyers against the whole firm per office:

```
lawyers2 <- data.frame(lawyers$Office, lawyers$Practice)
t <- table(lawyers2)
proptable <- t / sum(t)
proptable # table of proportions per practice, per office for the whole firm
```

```
##               lawyers.Practice
## lawyers.Office Corporate Litigation
##   Boston    0.26760563 0.40845070
##   Harvard    0.11267606 0.15492958
##   Providence 0.04225352 0.01408451
```

```
proptable['Boston', 'Corporate']
```

```
## [1] 0.2676056
```

```
round(proptable['Boston', 'Corporate'], 2)
```

```
## [1] 0.27
```

```
proptable['Providence', 'Corporate']
```

```
## [1] 0.04225352
```

```
round(proptable['Providence', 'Corporate'], 2)
```

```
## [1] 0.04
```

The proportion of all lawyers in Corporate practice in Boston is 0.27 (0.2676056), and the proportion of all lawyers in Corporate practice in Providence is 0.04 (0.04225352).

2) proportions of corporate lawyers against office totals:

```
office_totals <- table(lawyers$Office)
B.corp <- nrow(lawyers[lawyers$Practice == 'Corporate' &
  lawyers$Office == 'Boston',])
H.corp <- nrow(lawyers[lawyers$Practice == 'Corporate' &
  lawyers$Office == 'Harvard',])
P.corp <- nrow(lawyers[lawyers$Practice == 'Corporate' &
  lawyers$Office == 'Providence',])
pp <- lawyers[lawyers$Practice == 'Corporate' &
  lawyers$Office == 'Providence',]
corpprop <- data.frame(B.corp, H.corp, P.corp)
final <- corpprop / office_totals
colnames(final) <- c('Boston_corp', 'Harvard_corp', 'Providence_corp')
rownames(final) <- 'Proportion'
final

##           Boston_corp Harvard_corp Providence_corp
## Proportion   0.3958333   0.4210526             0.75
```

The proportion of Corporate lawyers in Boston office is 0.40 (0.3958333), and the proportion of Corporate lawyers in Providence office is 0.75.

3

```
lawyer_age <- aggregate(lawyers$Age, list(lawyers$Seniority, lawyers$Practice), mean)
colnames(lawyer_age) <- c('Seniority', 'Practice', 'Mean_age')
lawyer_age

##   Seniority Practice Mean_age
## 1 Associate Corporate 36.71429
## 2 Partner Corporate 48.50000
## 3 Associate Litigation 34.61905
## 4 Partner Litigation 47.70000
```

4

```
m.age <- aggregate(lawyers$Age, list(lawyers$Office), median)
colnames(m.age) <- c('Office', 'Median_age')
m.age[which.min(m.age[,2]),1]

## [1] Harvard
## Levels: Boston Harvard Providence
```

The office with the youngest median age is Havard.

Task 2

1

```
banana <- function(vec){  
  z <- vector(mode = 'double') # creates empty vector z of mode 'double'  
  for(i in 1:length(vec)-1) # initiates a for loop to work with input vector  
    # which will end on the last but one element  
  {x1 <- vec[i] # defines object x1 which equals the current value of iteration  
   x2 <- vec[i+1] # defines object x2 which equals the next value of iteration  
   z <- c(z, (100*(x2-x1*x1)^2+(1-x1)^2))} # appends results of formula  
    # calculations for n-1 vector elements to the z vector  
  return(sum(z)) # returns sum of the appended vector  
}  
  
banana(c(0.2, 0.5))  
  
## [1] 21.8  
banana(c(0.2, 0.5, 0.1, 0.6))  
  
## [1] 59.92
```

2

```
noloopba <- function(vec){  
  x <- vec[-length(vec)]  
  y <- vec[-1]  
  z <- 100*(y-x*x)^2+(1-x)^2  
  return(sum(z))  
}  
  
noloopba(c(0.2, 0.5))  
  
## [1] 21.8  
noloopba(c(0.2, 0.5, 0.1, 0.6))  
  
## [1] 59.92
```

3

```
system.time(replicate(1000, banana(c(0.2, 0.5, 0.1, 0.6))))  
  
##      user      system elapsed  
## 0.012    0.000    0.012
```

```
system.time(replicate(1000, noloopba(c(0.2, 0.5, 0.1, 0.6))))

##    user  system elapsed
##  0.009   0.000   0.008
# increased repetitions to 1000 because at 100 there was no difference
```

Function without a loop is faster.

Task 3

1

```
DublinAiport <- read.csv('2018_09_Dublin_Airport.csv', head = T)
WeatherData <- list(DublinAiport)
class(DublinAiport) <- c('WeatherData', 'data.frame')
class(DublinAiport)

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

2

```
summary.WeatherData <- function(airp) {
  md <- data.frame(airp$rain, airp$maxtp, airp$mintp)
  colnames(md) <- c('rain', 'max_tem', 'min_temp')
  x1 <- round(apply(md, 2, mean), 2)
  x2 <- round(apply(md, 2, sd), 2)
  x3 <- apply(md, 2, min)
  x4 <- apply(md, 2, max)
  list(mean = x1, standard_deviation = x2, minimum = x3, maximum = x4)
}

summary(DublinAiport)

## $mean
##    rain max_tem min_temp
##    1.46   16.69    7.65
##
## $standard_deviation
##    rain max_tem min_temp
##    3.08    2.73    3.85
##
## $minimum
##    rain max_tem min_temp
##    0.0    11.9    0.4
##
## $maximum
##    rain max_tem min_temp
```

```
##      15.7      23.0      13.6
```

3

```
CorkAirport <- read.csv('2018_09_Cork_Airport.csv', head = T)
class(CorkAirport) <- c('WeatherData', 'data.frame')
summary(CorkAirport)
```

```
## $mean
##      rain max_tem min_temp
##      2.58  15.95   8.69
##
## $standard_deviation
##      rain max_tem min_temp
##      4.58   2.38   2.34
##
## $minimum
##      rain max_tem min_temp
##      0.0   10.3   4.7
##
## $maximum
##      rain max_tem min_temp
##      19.2  20.2  13.0
```

September day temperatures were on average warmer at Dublin Airport, than at Cork Airport as suggested by the higher average maximum temperature and higher maximum temperature. Cork Airport saw more rain on average and higher maximum rainfall. At the same time, Cork Airport had higher minimum temperature so night time temperatures didn't get as low at Cork Airport as they did at Dublin Airport. This is consistent with higher rainfall as suggests cloudier and milder nights. Higher standard deviation at Cork Airport also suggests changeable weather in terms of rainfall.

4

```
plot.WeatherData <- function(data, maxtpcol = 'red', mintpcol = 'blue',
                             mmaxr = 'red', maxtpoint = T, mintppoint = T,
                             title = 'Temperature and precipitation'){
  par(mfrow = c(2,1), mar=c(4,4,2,2), xaxs = "i", yaxs = "i", cex.axis=1.3, cex.lab=1.4)
  plot(data$maxtp, type = 'n', xlab = 'day', ylab = "Air Temperature(C)",
       ylim = c(0, 36), main = title, cex.lab= 1)
  lines(data$maxtp, col = maxtpcol)
  lines(data$minntp, col = mintpcol)
  legend("topright", legend = c("max daily temp", "min daily temp"),
       cex = 0.74, lty=c(1,1), col = c(maxtpcol, mintpcol))
  if (maxtpoint == T) points(data[data$maxtp == max(data$maxtp),],
                           col = maxtpcol, pch = 19)
  if (mintppoint == T) points(data[data$minntp == min(data$minntp),],
                           col = mintpcol, pch = 19)
```

```

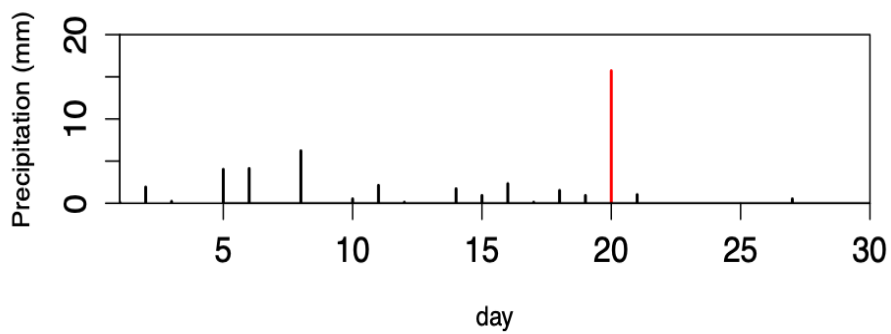
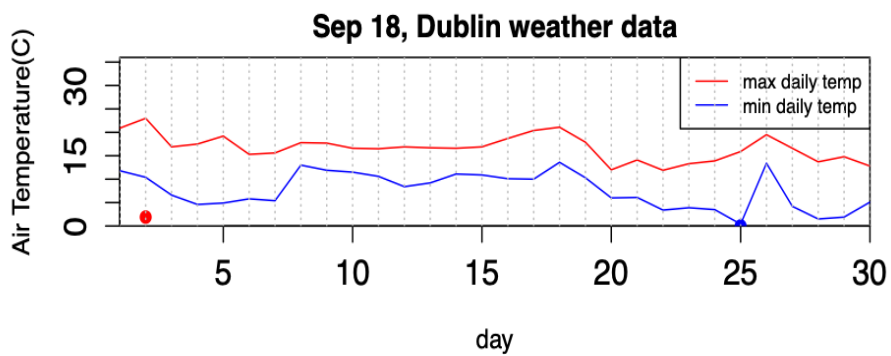
abline(v = seq(1,30), lty = 3, col = 'grey')
cols <- ifelse(data$rain == max(data$rain), mmaxr, 'black')
plot(data$rain, type = 'h', xlab = 'day', cex.lab= 1,
      ylab="Precipitation (mm)", ylim = c(0, 20), col = cols, lwd = 2)
}

```

```

plot(DublinAiport, title = 'Sep 18, Dublin weather data')

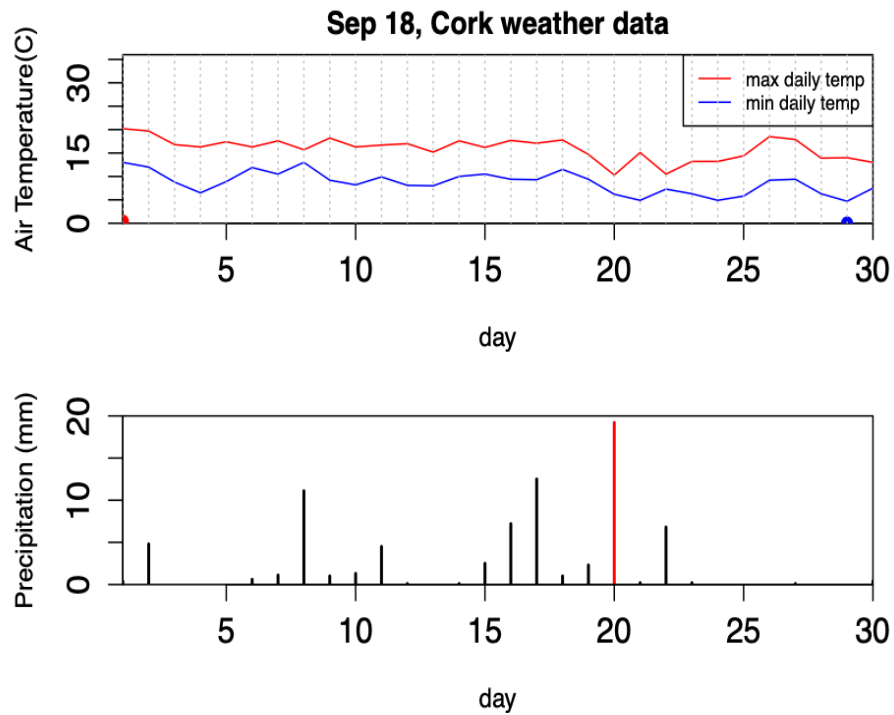
```



```

plot(CorkAiport, title = 'Sep 18, Cork weather data')

```



```
plot(DublinAiport, title = 'Sep 18, Dublin weather data (user options)',
      maxtppoint = F, mintppoint = F, maxtpcol = 'yellow',
      mintpcol = 'black', mmaxr = 'green')
```

