Stat291_Project-1

Dumbledore's army:

- 2429066 Mert Göksel
- 2290658 Rümeysa Durdağ
- 2361160 Merve Çakır

1-)

The data is about daily coronavirus readings, it has 5 columns. This columns are firstly date, then comes the province of the country, the country, latitude/longitude, the type of the case (cases/deaths/recovered) and finally the number of those cases.

```r
library(coronavirus) #CORONAVIRUS DATA

data <- coronavirus #assign the dataset to a variable

head(data) #first 6 rows of the data
```

```
##         date province     country      lat     long      type cases
## 1 2020-01-22          Afghanistan 33.93911 67.70995 confirmed     0
## 2 2020-01-23          Afghanistan 33.93911 67.70995 confirmed     0
## 3 2020-01-24          Afghanistan 33.93911 67.70995 confirmed     0
## 4 2020-01-25          Afghanistan 33.93911 67.70995 confirmed     0
## 5 2020-01-26          Afghanistan 33.93911 67.70995 confirmed     0
## 6 2020-01-27          Afghanistan 33.93911 67.70995 confirmed     0
```

2-)

```r
sapply(data, class)
```

```
##        date    province     country         lat        long        type
##      "Date" "character" "character"   "numeric"   "numeric" "character"
##       cases
##   "integer"
```

3-) Created questions about the dataset:

3.1-) What is the index of the max case number for any type?

```r
which.max(data$cases)
```

```
## [1] 106531
```

```r
data[106531,]
```

```
##              date province country     lat     long      type  cases
## 106531 2020-07-02           Brazil -14.235 -51.9253 recovered 140050
```

3.2-) What rows are for United states?

```r
c(min(which(data$country == "US")), max(which(data$country == "US")))
```

```
## [1]  33793 136320
```

4.1-) Is there any NA value?

```
any(is.na(data))
```

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

Our dataframe doesnt contain any NA values but if it did then we would have shown it as

```
which(is.na(data))
```

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

4.2-) Print the col names

```
names(data)
```

```
## [1] "date"     "province" "country"  "lat"      "long"     "type"     "cases"
```

If there was no names assigned we would have assigned them with

colnames(data) <- c(names) #c(names) = names we gave in order

4.3-) Show first 20 rows of our data:

```
head(data, 20)
```

```
##          date province     country      lat     long      type cases
## 1  2020-01-22          Afghanistan 33.93911 67.70995 confirmed     0
## 2  2020-01-23          Afghanistan 33.93911 67.70995 confirmed     0
## 3  2020-01-24          Afghanistan 33.93911 67.70995 confirmed     0
## 4  2020-01-25          Afghanistan 33.93911 67.70995 confirmed     0
## 5  2020-01-26          Afghanistan 33.93911 67.70995 confirmed     0
## 6  2020-01-27          Afghanistan 33.93911 67.70995 confirmed     0
## 7  2020-01-28          Afghanistan 33.93911 67.70995 confirmed     0
## 8  2020-01-29          Afghanistan 33.93911 67.70995 confirmed     0
## 9  2020-01-30          Afghanistan 33.93911 67.70995 confirmed     0
## 10 2020-01-31          Afghanistan 33.93911 67.70995 confirmed     0
## 11 2020-02-01          Afghanistan 33.93911 67.70995 confirmed     0
## 12 2020-02-02          Afghanistan 33.93911 67.70995 confirmed     0
## 13 2020-02-03          Afghanistan 33.93911 67.70995 confirmed     0
## 14 2020-02-04          Afghanistan 33.93911 67.70995 confirmed     0
## 15 2020-02-05          Afghanistan 33.93911 67.70995 confirmed     0
## 16 2020-02-06          Afghanistan 33.93911 67.70995 confirmed     0
## 17 2020-02-07          Afghanistan 33.93911 67.70995 confirmed     0
## 18 2020-02-08          Afghanistan 33.93911 67.70995 confirmed     0
## 19 2020-02-09          Afghanistan 33.93911 67.70995 confirmed     0
## 20 2020-02-10          Afghanistan 33.93911 67.70995 confirmed     0
```

4.4-) The numeric values are latitude/longitude and cases

```
c(mean(data$lat), mean(data$long), mean(data$cases))
```

```
## [1]  20.67650  24.49493 189.77153
```

The table of country&cases can be made with the code below but as there are too many countries in the dataset we wont run it on our pdf

```
table(data$country, data$cases)
```

#4.5-) Order your data depending on one of the numeric variables.

```
data_ordered <- data[order(data$cases),]
head(data_ordered)
```

```
##                 date province    country      lat       long      type  cases
## 123444 2020-07-19              Mexico 23.63450 -102.55280 recovered -16298
## 130549 2020-07-20              Serbia 44.01650   21.00590 recovered -15564
## 29854  2020-04-24               Spain 40.46367   -3.74922 confirmed -10034
## 119019 2020-07-10          Kazakhstan 48.01960   66.92370 recovered  -2637
## 11427  2020-04-29              France 46.22760    2.21370 confirmed  -2512
## 136240 2020-05-12                  US 40.00000 -100.00000 recovered  -2446
```

```
#this is too long as well so we printed the head of the dataset.
```

5.1-) Create a new variable which is a linear combination Our numeric variables will be: v1 = latitude, v2 = longitude Our formula for linear combination will be = 5v1-4v2

```
v1 <- data$lat
v2 <- data$long
dumbeldors_army <- 5*v1-4*v2
data$dumbeldors_army <- dumbeldors_army
head(data)
```

```
##         date province     country      lat     long      type cases
## 1 2020-01-22          Afghanistan 33.93911 67.70995 confirmed     0
## 2 2020-01-23          Afghanistan 33.93911 67.70995 confirmed     0
## 3 2020-01-24          Afghanistan 33.93911 67.70995 confirmed     0
## 4 2020-01-25          Afghanistan 33.93911 67.70995 confirmed     0
## 5 2020-01-26          Afghanistan 33.93911 67.70995 confirmed     0
## 6 2020-01-27          Afghanistan 33.93911 67.70995 confirmed     0
##   dumbeldors_army
## 1       -101.1443
## 2       -101.1443
## 3       -101.1443
## 4       -101.1443
## 5       -101.1443
## 6       -101.1443
```

5.2-) By using for loops, multiply one of your numeric variables with another one We took the lat, long var for v1&v2 but they are too repetitive as each country has many lines of data but their lat&long doesnt change. So we will take only the unique values

```
v1_unique = unique(v1)
v2_unique = unique(v2)

multiply_list <- list()
for(i in 1:length(v1_unique)){
  multiply_list[i] <- v1_unique[i]*v2_unique[i]
}
mutliply_matrix <- matrix(multiply_list, nrow = length(v1_unique), ncol = length(v2_unique))
#mutliply_matrix This matrix has too many columns for a pdf so we wont print it.
```

6-)By using one of your numeric variables, generate a new vector including the values greater than the median of this vector and name it as mynewvector.

```
mynewvector <- data$cases[data$cases>median(data$cases)]
```

7-) Convert your data to a list, show the class of each member of your list

```
data_list <- as.list(data)
lapply(data_list, FUN = typeof)
```

```
## $date
## [1] "double"
##
## $province
## [1] "character"
##
## $country
## [1] "character"
##
## $lat
## [1] "double"
##
## $long
## [1] "double"
##
## $type
## [1] "character"
##
## $cases
## [1] "integer"
##
## $dumbeldors_army
## [1] "double"
```

8-) Select your list elements by using their names.

```
data_list$date
data_list$province
data_list$country
data_list$lat
data_list$long
data_list$cases
data_list$dumbeldors_army
#Because dataset is too long we wont print these.
```

9-)

```
# Because that we have non numeric columns the max function returns the last value as the maximum.
# Unlisting date variables is not good but for this questions sake we did it anyway...
for (i in 1:length(data_list)){
  list_names <- names(data_list)
  number <- i
  name <- names(data_list)[i]
  maxim <- max(unlist(data_list[list_names[i]]))
  cat("My", number,"variable name is" ,name , "and the greatest value for my vector is", maxim, "\n")
}
```

```
## My 1 variable name is date and the greatest value for my vector is 18474
## My 2 variable name is province and the greatest value for my vector is Zhejiang
## My 3 variable name is country and the greatest value for my vector is Zimbabwe
## My 4 variable name is lat and the greatest value for my vector is 71.7069
## My 5 variable name is long and the greatest value for my vector is 178.065
## My 6 variable name is type and the greatest value for my vector is recovered
## My 7 variable name is cases and the greatest value for my vector is 140050
## My 8 variable name is dumbeldors_army and the greatest value for my vector is 861.4115
```

10-) Change one of your list element to NULL. This means deleting the list so it wont show up in the head()

```
data_list[["date"]] <- NULL
head(as.data.frame(data_list))
```

```
##   province     country      lat     long      type cases dumbeldors_army
## 1          Afghanistan 33.93911 67.70995 confirmed     0       -101.1443
## 2          Afghanistan 33.93911 67.70995 confirmed     0       -101.1443
## 3          Afghanistan 33.93911 67.70995 confirmed     0       -101.1443
## 4          Afghanistan 33.93911 67.70995 confirmed     0       -101.1443
## 5          Afghanistan 33.93911 67.70995 confirmed     0       -101.1443
## 6          Afghanistan 33.93911 67.70995 confirmed     0       -101.1443
```

As seen above the "date" list is no longer there

11-)

```
rm(list = ls(all.names = TRUE))
```