Assignment 1

Yan Zichu, 300476924

30 July 2020

Q1 (3 Marks): Classify the following variables as categorical/numerical and further classify as logical,nominal, ordinal, binary, continous or discrete. Give the R data type that corresponds to each.

ANS: a.Earthquake magnitude : Numeric variable - Continous variable - Numeric (decimal) in R

b.Presence of infection: Categorical variable - Nominal variable - Logical (Boolean) in R

c.Length of time to symptoms following infection : Numeric variable - Continous variable - Numeric (decimal) in R

d.Patient reported level of pain (scale 0-5) : Categorical variable - Ordinal variable - Integer in R

e.Aircraft speed : Numeric variable - Continous variable - Numeric (decimal) in R

f.Brand of noodles : Categorical variable - Nominal variable - Logical (Boolean) in R

g.Number of patients seen in a day : Numeric variable - Discrete variable - Integer in R

h.Mean number of patients seen in a day : Numeric variable - Discrete variable - Integer in R

i.Proportion of ground covered with vegetation :Numeric variable - Continous variable - Numeric (decimal) in R

Q2 (2 Marks): Find out and explain what happens when the following code is evaluated

ANS:

```
x <- y <- 5
x
```

```
## [1] 5
```

```
y
```

```
## [1] 5
```

As we can see, x and y are equal 5. Because we assign 5 to y first and then assign value of y (5) to x, they have same value.

Q3. (2 Marks) When measuring angles values less than 0 or greater than 360 can be converted into the range [0,360]. Write R code that takes the vector x of angles c(45, 345, 600, 23, 705) and converts them into the correct angles in the range [0,360].

ANS:

```
x <- c(45, 345, 600, 23, 705)
x%%360
```

```
## [1]  45 345 240  23 345
```

Q4. (7 Marks)

a.Write R code to define the following objects

ANS:

```
x <- matrix(c(2,3,4,2),ncol = 1,byrow=FALSE)
x
```

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

```
y <- matrix(c(2,3),ncol = 1,byrow=FALSE)
y
```

```
##      [,1]
## [1,]    2
## [2,]    3
```

```
z <- matrix(c(12,3,13,-1,11,6),ncol = 3,byrow=FALSE)
z
```

```
##      [,1] [,2] [,3]
## [1,]   12   13   11
## [2,]    3   -1    6
```

b.Write R code that will extract the second column of z and save it to a vector s

ANS:

```
z <- matrix(c(12,3,13,-1,11,6),ncol = 3,byrow=FALSE)
s <- z[,2]
s
```

```
## [1] 13 -1
```

c.Write R code that will append y onto the end of x - showing your output to verify the result

ANS:

```
x <- matrix(c(2,3,4,2),ncol = 1,byrow=FALSE)
y <- matrix(c(2,3),ncol = 1,byrow=FALSE)
rbind(x,y)
```

```
##      [,1]
## [1,]    2
## [2,]    3
## [3,]    4
## [4,]    2
## [5,]    2
## [6,]    3
```

d.Write R code that will extract the values of the elements of x that are less than 3.

ANS:

```
x <- matrix(c(2,3,4,2),ncol = 1,byrow=FALSE)
x[x<3]
```

```
## [1] 2 2
```

e.Write R code that will extract the positions of the elements of x that are less than 3.

ANS:

```r
x <- matrix(c(2,3,4,2),ncol = 1,byrow=FALSE)
x
```

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

```r
which(x < 3 , TRUE)
```

```
##      row col
## [1,]   1   1
## [2,]   4   1
```

Q5. (22 Marks) Read the dataset ypd.csv from the course website: it contains a subset of survey responses from young people in Slovakia, and is availble from https://www.kaggle.com/miroslavsabo/young-people-survey.

Note: You can assume that the file ypd.csv is in the working directory of R when you submit your solution. Then, giving and executing all relevant R code:

ANS: ## a.Name and classify the type of each variable in the dataset.

```r
ypd <- read.csv("ypd.csv",stringsAsFactors = FALSE)
names(ypd)
```

```
## [1] "Age"                          "Height"
## [3] "Weight"                       "Number.of.siblings"
## [5] "Gender"                       "Education"
## [7] "Have.difficulty.getting.up" "Prefer.fast.songs"
## [9] "Degree.belief.in.God"
```

```r
str(ypd)
```

```
## 'data.frame':    1010 obs. of  9 variables:
##  $ Age                        : int  20 19 20 22 20 20 20 19 18 19 ...
##  $ Height                     : int  163 163 176 172 170 186 177 184 166 174 ...
##  $ Weight                     : int  48 58 67 59 59 77 50 90 55 60 ...
##  $ Number.of.siblings         : int  1 2 2 1 1 1 1 1 1 3 ...
##  $ Gender                     : chr  "female" "female" "female" "female" ...
##  $ Education                  : chr  "college/bachelor degree" "college/bachelor degree" "secondary s
##  $ Have.difficulty.getting.up: int  2 5 4 1 4 3 2 5 5 4 ...
##  $ Prefer.fast.songs          : int  3 4 5 3 3 3 5 3 3 3 ...
##  $ Degree.belief.in.God       : int  1 1 5 4 5 3 5 4 5 5 ...
```

b.Give the dimensions of the data in the dataset (number of rows and columns).

ANS:

```
nrow(ypd)
```

```
## [1] 1010
```

```
ncol(ypd)
```

```
## [1] 9
```

c.Remove all rows from the data frame which have missing height, saving the result in a new data frame. Use this new data frame for the rest of this question.

ANS:

```
newypd <- ypd[complete.cases(ypd[,"Height"]),]
```

d.Give the number of rows of data in the new data frame.

ANS:

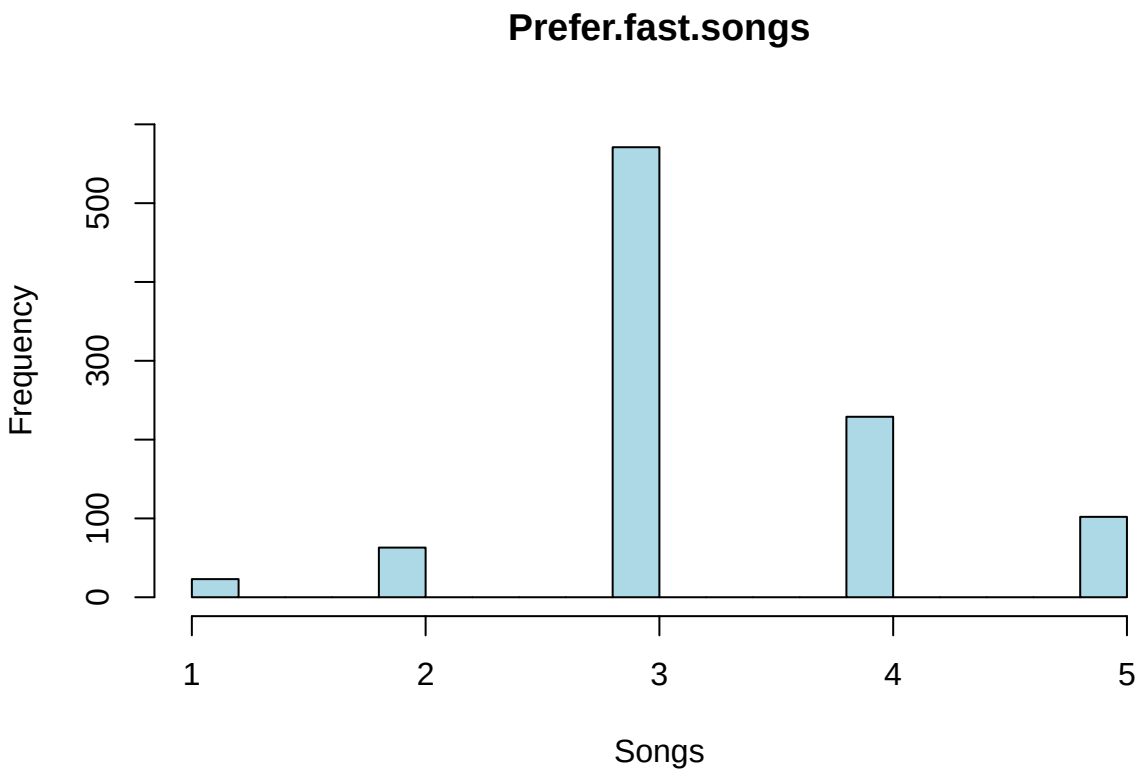
```
nrow(newypd)
```

```
## [1] 990
```

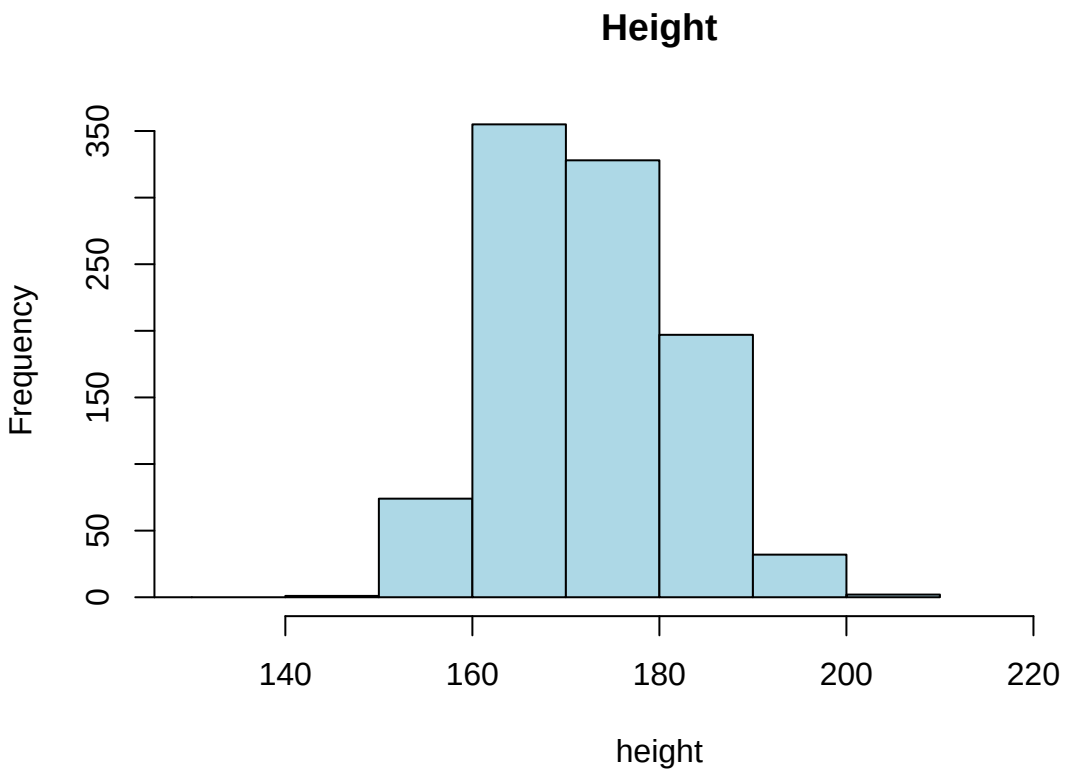
e.Showing your code, draw suitable graphical displays of Prefer.fast.songs, Height and Have.difficulty.getting.up. Label both axes, and give each graph a title.

ANS:

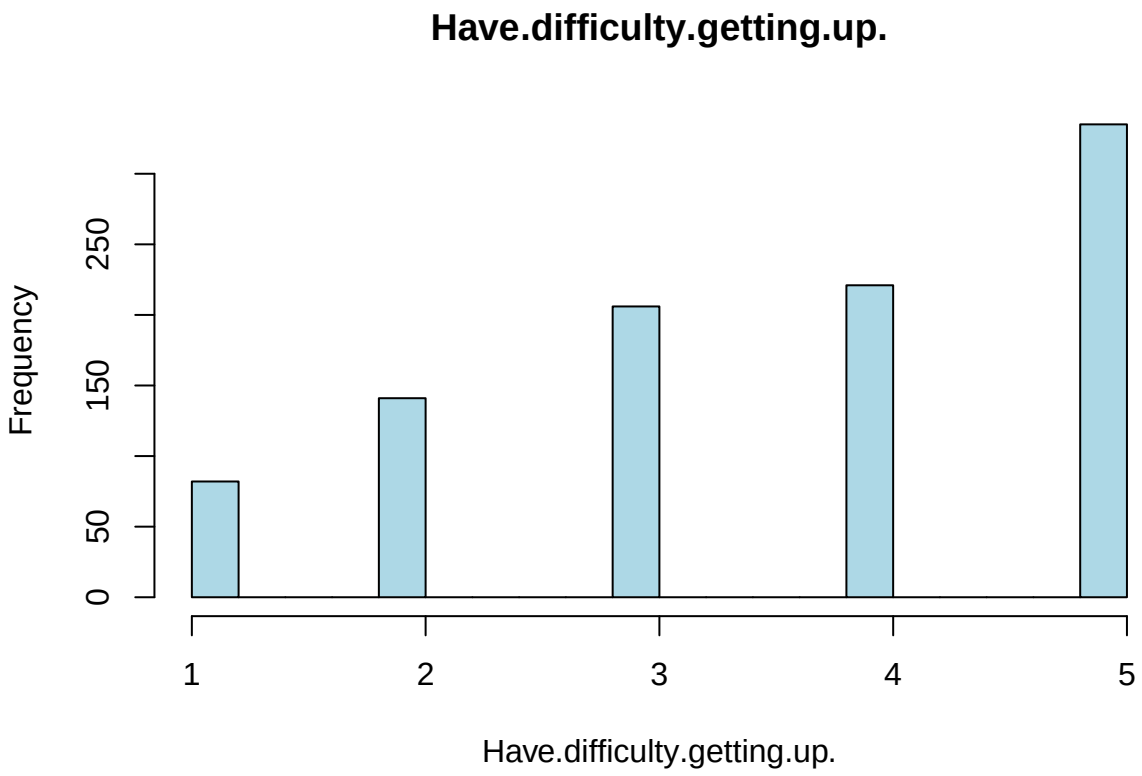
```
hist(newypd$Prefer.fast.songs, main="Prefer.fast.songs", xlab="Songs", col="lightblue",
     xlim = c(1,5),ylim=c(0,600),breaks = 17)
```




```
hist(newypd$Height, main="Height", xlab="height", col="lightblue",xlim = c(130,230))
```




```
hist(newypd$Have.difficulty.getting.up, main="Have.difficulty.getting.up.", xlab="Have.difficulty.getti
```



f.Give values for the minimum, lower quartile, median, mean, upper quartile and maximum value of Age and also give the number of observations with missing values of Age.

ANS:

```
print(paste0("Sum.Na: ",sum(is.na(newypd$Age))))
```

```
## [1] "Sum.Na: 2"
```

```
print(paste0("Min: ",min(newypd$Age)))
```

```
## [1] "Min: NA"
```

```
print(paste0("Max: ",max(newypd$Age)))
```

```
## [1] "Max: NA"
```

```
print(paste0("Median: ",median(newypd$Age)))
```

```
## [1] "Median: NA"
```

```r
print(paste0("Mean: ",mean(newypd$Age)))
```

```
## [1] "Mean: NA"
```

```r
print(paste0("Lower quantile: ",quantile(newypd$Age,.25,na.rm = TRUE)))
```

```
## [1] "Lower quantile: 19"
```

```r
print(paste0("Upper quantile: ",quantile(newypd$Age,.75,na.rm = TRUE)))
```

```
## [1] "Upper quantile: 22"
```

g.Give a table of totals and percentages (to 1 decimal place) for the 5 levels of Have.difficulty.getting.up, showing the missing (NA) category as a 6th option. What percentage of people score on the lowest two levels of the scale?

ANS:

```r
table <- data.frame(level=c("1","2","3","4","5","6"),
                    percentage=round(table(factor(newypd$Have.difficulty.getting.up, exclude = NULL))
                                     / nrow(newypd)*100,1),
                    total = table(factor(newypd$Have.difficulty.getting.up, exclude = NULL)),
                    stringsAsFactors = FALSE)
table
```

```
##   level percentage.Var1 percentage.Freq total.Var1 total.Freq
## 1     1               1             8.3          1         82
## 2     2               2            14.2          2        141
## 3     3               3            20.8          3        206
## 4     4               4            22.3          4        221
## 5     5               5            33.8          5        335
## 6     6            <NA>             0.5       <NA>          5
```

Except NA, Level1 (8.3%) and Level2(14.5%) are lowest

h.Find the level of difficulty getting up experienced by the shortest and tallest people in the data set.

ANS:

```r
newypd$Have.difficulty.getting.up[which.min(newypd$Height)]
```

```
## [1] 2
```

```r
newypd$Have.difficulty.getting.up[which.max(newypd$Height)]
```

```
## [1] 4
```

i.Make a table of the number of people in the data set at each level of education. Make sure the table is ordered by increasing level of education.

ANS:

```
table <- data.frame(sort(table(newypd$Education)))
table
```

```
##                                Var1 Freq
## 1                                      1
## 2                  doctorate degree    5
## 3 currently a primary school pupil    9
## 4                    primary school   77
## 5                    masters degree   79
## 6           college/bachelor degree  208
## 7                  secondary school  611
```

Q6 (10 Marks) The Fibonacci sequence is an infinite sequence of numbers that begins with the pair of numbers 0 and 1. Each successive number is then the sum of the two preceding elements. So the third element is 0+1=1, the fourth is 1+1=2 etc.

The following function computes the first n elements of the Fibonacci sequence.

a.Verify that the function works by using it to calcuate the first 10 elements of the Fibonacci sequence. Show the R call and the output.

ANS:

```
# Given code
fibonacci <- function(n) {
  x <- c(0,1)
  for(i in 3:n) {
    x[i] <- x[i-1] + x[i-2]
  }
  return(x)
}

fibonacci(10)
```

```
##  [1]  0  1  1  2  3  5  8 13 21 34
```

b.Explain briefly how the function works. The body of the function has five lines. Explain what happens in the three sections: line 1, lines 2-4, and line 5.

ANS: line 1: Creating a function which contains one parameter and function name is fibonacci

line 2-4: “x <- c(0,1)” : Initialize the x, assign a vector (0,1) to x. First line in forloop : Give a range to the forloop, traverse i from 3(because 0 and 1 are first 2 elements) to n(the parameter in function) Second line in forloop : This is the main body of the loop, this line aim to calculate and assign the next value based on sum of previous 2 values. e.g. when i =3 then “(x[i-1] = 1) + (x[i-2] = 0) = (x[i] = 1)”

line 5 : This function tend to give us a vector variable which contains a finite Fibonacci sequence we defined

c.The function goes wrong if we try to call it for n<=2. Try it! Explain what goes wrong and why.

Because of this step “x[i] <- x[i-1] + x[i-2]”, when n <= 2, we will not get any value from x[0],x[-1] or other x[negative value] as in those location we did not store any value and negative location value is invalid.

d.Fix the function by modifying to (1) terminate with an error (and a helpful error message) if n<=0, and (2) to return only the first or the first+second elements if n=1 or n=2 respectively. Give the R code for the function that fixes the problem.

ANS:

```
fibonacci <- function(n) {

  if(n>=3){
    x <- c(0,1)
    for(i in 3:n) {
      x[i] <- x[i-1] + x[i-2]
    }
    return(x)
  }else {
    if(n<=0){
      return("ERROR: 0 is invalid, Please enter the number >=3 :D")
    }
    if(n==1){
      return(0)
    }
    if(n==2){
      return(0+1)
    }
  }
}

fibonacci(1)
```

```
## [1] 0
```

```
fibonacci(2)
```

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

```
fibonacci(0)
```

```
## [1] "ERROR: 0 is invalid, Please enter the number >=3 :D"
```

Q7 (8 Marks)

a.Explain the difference between the action of the statements install.packages("gdata") and library(gdata). What is gdata here?

ANS: install.packages("gdata"): install a package called "gdata". library(gdata): import a package called "gdata". "gdata" is a package which contains a lot of functions.

b.When the package gdata is loaded using the command library(gdata) a number of warning messages are printed to the screen by R. One of them is the following:

```
    The following object is masked from 'package:utils':  object.size
```

Explain what this warning message means.

ANS: When there are two functions with the same name, the search will stop and call when the first one is found. So when we want to call second one the warning will come out and say second one is masked.

c.Explain what happens when the following code is executed

```
ff <- function(x, y) {
  #print(x)# 2
  #print(y)# 3
  #print(a)# 4
  z <- x + y + a
  a <- 2
  #print(a)# 5
  return(z)# 6
}
x <- 10
y <- 15
a <- 5
#print(a) # 1
ff(1, 3) + a
```

```
## [1] 14
```

```
#print(a)# 7
```

ANS: 1. We should know 'x <- 10;y <- 15' these two lines are useless which just simply assign value to two variables.

2. ff(1, 3) + a: This is the main part of this code, a(Global variable) is 5, although variable a will be changed in the function but the change will happen after this line ("ff(1, 3) + a") which means a will not change here, it is still 5.
3. So now we can focus on "ff(1, 3)" this part, ff have 2 parameters x and y which are 1 and 3 respectively, so "z <- x + y + a" : z = 1 + 3 + 5 -> z = 9
4. Based on step 2 and 3, all values are known in ff(1, 3) + a, so it is 9 + 5 = 14

d.Explain what happens when the following code is executed

```
data(cars)
```

ANS: Import a dataset contains 50 cars‘ information (speed and dist)

e.What is the meaning of the text utils:: in the following R code?

```
utils::str(cars)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
##  $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

ANS: call a function "str" in the utils package.(Compactly Display the Structure of an Arbitrary R Object)