GreenSlides

Rotem

3/13/2023

Green Slides 1

In this R Markdown document, you will record responses to the questions presented on the green slides from our class. Remember to submit both the (.rmd) file and the generated pdf file after knitting the document. Don't forget to update the "author" and date information accordingly.

1 Practicing - Importing Datasets (1)

Import file "table1.txt" to a dataframe (the file is on moodle, under week1 (###1###)). Make sure you: - skip the first descriptive line - deal with the "/" quoting the strings - have columns as strings (and not factors) - the columns will have names (and not be called V1, V2,..)

```
table1<- read.table("table1.txt", header = TRUE, skip =2, quote = "/")
table1
```

```
##       Name Age Height Weight Sex
## 1     Alex  25    177     57   F
## 2    Lilly  31    163     69   F
## 3     Mark  23    190     83   M
## 4   Oliver  52    179     75   M
## 5   Martha  76    163     70   F
## 6    Lucas  49    183     83   M
## 7 Caroline  26    164     53   F
```

```
colnames(table1)<-c('gil', 'gova' , 'mishkal' , 'migdar')
table1
```

1.2) Change the name of the columns to "gil","gova","mishkal", "migdar"

```
##        gil gova mishkal migdar NA
## 1     Alex   25     177     57  F
## 2    Lilly   31     163     69  F
## 3     Mark   23     190     83  M
## 4   Oliver   52     179     75  M
## 5   Martha   76     163     70  F
## 6    Lucas   49     183     83  M
## 7 Caroline   26     164     53  F
```

1.3) Create a column with the names (note that you can name also the rows)

(undoing the argument "row.names="Name"," used when reading the file)

```
colnames(table1)<-c('name','gil', 'gova' , 'mishkal' , 'migdar')
table1
```

```
##       name gil gova mishkal migdar
## 1     Alex  25  177      57      F
## 2    Lilly  31  163      69      F
## 3     Mark  23  190      83      M
## 4   Oliver  52  179      75      M
## 5   Martha  76  163      70      F
## 6    Lucas  49  183      83      M
## 7 Caroline  26  164      53      F
```

1.4) Display how many rows and columns it has

```
rows_number<- nrow(table1)
cols_number<- ncol(table1)
rows_number
```

```
## [1] 7
```

```
cols_number
```

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

1.5) Drop column "name"

```
del_col<-table1[,-1]
 del_col
```

```
##   gil gova mishkal migdar
## 1  25  177      57      F
## 2  31  163      69      F
## 3  23  190      83      M
## 4  52  179      75      M
## 5  76  163      70      F
## 6  49  183      83      M
## 7  26  164      53      F
```

2 Practicing - Importing Datasets (2)

Read table2.txt (also in moodle), making sure that: - all columns have names (and not be called V1, V2,..) - all missing values are set to NA - decimals points are defined with a "." and not a "," as in the file - Sex columns should be of type factor

```
table2<- read.table("table2.txt", header = TRUE, sep = ";", quote = "/",
                    na.strings = c("--", "**", ""), dec = ",",
                    colClasses = c ("character", "numeric", "numeric", "numeric", "factor"))
table2
```

```
##       Name Age Height Weight Sex
## 1     Alex  25   1.77     57   F
## 2    Lilly  31     NA     69   F
## 3     Mark  NA   1.90     83   M
## 4   Oliver  52   1.79     75   M
## 5   Martha  76     NA     70   F
## 6    Lucas  49   1.83     NA   M
## 7 Caroline  26   1.64     53   F
```

2.2) Load the Titanic dataset and perform the following:

(Q1) Display how can we know whether the dataset titanic is a dataframe (Q2) Convert it to a data frame if it is not (Q3) Find out how many rows and columns the dataframe has, their names, types, etc. (Q4) Display the first rows of the dataset (Q5) show all column names (Q6) show the values of a specific column in dataset (e.g. "Age") (Q7) What's the length of the column "Age" ? (Q8) What's the class of the column "Class" ?

```
#Q1
class (Titanic)
```

```
## [1] "table"
```

```
#Q2
Titanic<-as.data.frame(Titanic)
class(Titanic)
```

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

```
#Q3
str(Titanic)
```

```
## 'data.frame':    32 obs. of  5 variables:
##  $ Class   : Factor w/ 4 levels "1st","2nd","3rd",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ Sex     : Factor w/ 2 levels "Male","Female": 1 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : Factor w/ 2 levels "Child","Adult": 1 1 1 1 1 1 1 1 2 2 ...
##  $ Survived: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Freq    : num  0 0 35 0 0 0 17 0 118 154 ...
```

```
#Q4
Titanic[1:4,]
```

```
##   Class  Sex   Age Survived Freq
## 1   1st Male Child       No    0
## 2   2nd Male Child       No    0
## 3   3rd Male Child       No   35
## 4  Crew Male Child       No    0
```

```
#Q5
colnames(Titanic)
```

```
## [1] "Class"    "Sex"      "Age"      "Survived" "Freq"
```

```
#Q6
Titanic$Age
```

```
##  [1] Child Child Child Child Child Child Child Child Adult Adult Adult Adult
## [13] Adult Adult Adult Adult Child Child Child Child Child Child Child Child
## [25] Adult Adult Adult Adult Adult Adult Adult Adult
## Levels: Child Adult
```

```
#Q7
length(Titanic$Age)
```

```
## [1] 32
```

```
#Q8
class(Titanic$Class)
```

```
## [1] "factor"
```

3 Practicing vectors

3.1) Create a vector of all the letters in the alphabet, in reverse order.

```
rev(letters)
```

```
##  [1] "z" "y" "x" "w" "v" "u" "t" "s" "r" "q" "p" "o" "n" "m" "l" "k" "j" "i" "h"
## [20] "g" "f" "e" "d" "c" "b" "a"
```

3.2) Use the subset() function to create a vector of all the elements of the vector c(1, 2, 3, 4, 5, 6) that are greater than 2:

```
m_vec<-c(1,2,3,4,5,6)
subset(m_vec,m_vec>2)
```

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

3.3) Using again the subset() function, create a vector of the first 20 elements of the Fibonacci sequence that are divisible by 3. Note you can use also the fibonacci() function.

```
library(numbers)
fibo_vec<-fibonacci(80,TRUE)
```

```
## Warning in fibonacci(80, TRUE): For 'n > 78' not exactly representable in R as
## integer.
```

```
subset(fibo_vec,fibo_vec%%3==0)
```

```
##  [1] 3.000000e+00 2.100000e+01 1.440000e+02 9.870000e+02 6.765000e+03
##  [6] 4.636800e+04 3.178110e+05 2.178309e+06 1.493035e+07 1.023342e+08
## [11] 7.014087e+08 4.807527e+09 3.295128e+10 2.258514e+11 1.548009e+12
## [16] 1.061021e+13 7.272346e+13 4.984540e+14 3.416455e+15 1.447233e+16
```

3.4) Using one line of code, create a vector of the first 100 multiples of 7. Hint: use the seq() function

```
seq(0,by=7,length.out=100)
```

```
##   [1]   0   7  14  21  28  35  42  49  56  63  70  77  84  91  98 105 112 119
##  [19] 126 133 140 147 154 161 168 175 182 189 196 203 210 217 224 231 238 245
##  [37] 252 259 266 273 280 287 294 301 308 315 322 329 336 343 350 357 364 371
##  [55] 378 385 392 399 406 413 420 427 434 441 448 455 462 469 476 483 490 497
##  [73] 504 511 518 525 532 539 546 553 560 567 574 581 588 595 602 609 616 623
##  [91] 630 637 644 651 658 665 672 679 686 693
```

3.5) Create a vector of the first 20 elements of the sequence of powers of 2

```
vec<-0:19
2**vec
```

```
##  [1]      1      2      4      8     16     32     64    128    256    512
## [11]   1024   2048   4096   8192  16384  32768  65536 131072 262144 524288
```

Good luck !