

RWorksheet2Final

Nicole Duero

2023-10-04

```
knitr::opts_chunk$set(echo = TRUE)
```

```
#USING VECTORS
```

```
#1. Create a vector using : operator
```

```
#a. Sequence from -5 to 5. Write the R code and its output. Describe its output.
```

```
seq1<-seq(-5,5)
```

```
seq1
```

```
## [1] -5 -4 -3 -2 -1 0 1 2 3 4 5
```

```
#Describe its output:-5 -4 -3 -2 -1 0 1 2 3 4 5. It ranges from the negative 5(-5) to positive 5.
```

```
#b. x <- 1:7. What will be the value of x?
```

```
x<- 1:7
```

```
x
```

```
## [1] 1 2 3 4 5 6 7
```

```
#2.* Create a vector using seq() function
```

```
#a. seq(1, 3, by=0.2) # specify step size. Write the R script and its output. Describe the output.
```

```
seq2<-seq(1,3, by=0.2)
```

```
seq2
```

```
## [1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0
```

```
#Answer: 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0. The sequence is start from 1 to 3 by adding each :
```

```
#3. A factory has a census of its workers. There are 50 workers in total. The following list shows their
```

```
#22, 37, 34, 19, 20, 57, 49, 50, 37, 46, 25, 17, 37, 43, 53, 41, 51, 35,
```

```
#24,33, 41, 53, 40, 18, 44, 38, 41, 48, 27, 39, 19, 30, 61, 54, 58, 26,18
```

```
ageWorker<-c(34, 28, 22, 36, 27, 18, 52, 39, 42, 29, 35, 31, 27, 22, 37, 34, 19, 20, 57, 49, 50, 37, 46, 25, 17,
```

```
ageWorker
```

```
## [1] 34 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19 20 57 49 50 37 46 25 17
```

```
## [26] 37 43 53 41 51 3 24 33 41 53 40 18 44 38 41 48 27 39 19 30 61 54 58 26 18
```

```
#a. Access 3rd element, what is the value?
```

```
ageWorker[3]
```

```
## [1] 22
```

```
#b. Access 2nd and 4th element, what are the values?  
ageWorker[c(2,4)]
```

```
## [1] 28 36
```

```
#c. Access all but the 4th and 12th element is not included. Write the R script and its output.  
ageWorker[-c(4,12)]
```

```
## [1] 34 28 22 27 18 52 39 42 29 35 27 22 37 34 19 20 57 49 50 37 46 25 17 37 43  
## [26] 53 41 51 3 24 33 41 53 40 18 44 38 41 48 27 39 19 30 61 54 58 26 18
```

```
ageWorker[c(-4,-12)]
```

```
## [1] 34 28 22 27 18 52 39 42 29 35 27 22 37 34 19 20 57 49 50 37 46 25 17 37 43  
## [26] 53 41 51 3 24 33 41 53 40 18 44 38 41 48 27 39 19 30 61 54 58 26 18
```

```
#4. *Create a vector x <- c("first"=3, "second"=0, "third"=9). Then named the vector, names(x).  
names<-c("first"=3, "second"=0, "third"=9)  
names
```

```
## first second third  
##      3      0      9
```

```
#a. Print the results. Then access x[c("first", "third")]. Describe the output.  
names[c("first", "third")]
```

```
## first third  
##      3      9
```

```
#b. Write the code and its output.  
#names<-c("first"=3, "second"=0, "third"=9)  
#names  
#first second third  
#3      0      9  
#Print the results. Then access x[c("first", "third")]. Describe the output.  
#names[c("first", "third")]  
#first third  
#3      9
```

```
#5. Create a sequence x from -3:2.  
x<-c(-3:2)  
x
```

```
## [1] -3 -2 -1 0 1 2
```

```
#a. Modify 2nd element and change it to 0; Describe the output.  
x[2]<-0  
x
```

```
## [1] -3 0 -1 0 1 2
```

```
#b. Write the code and its output.
#x<-c(-3:2)= -3 -2 -1 0 1 2
#x[2]= -3 0 -1 0 1 2
```

#6. *The following data shows the diesel fuel purchased by Mr. Cruz.

```
Month= c("Jan", "Feb", "March", "Apr", "May", "June")
Price_per_liter_Php= c(52.50,57.25,60.00,65.00,74.25,54.00)
Purchase_quantity_Liters= c(25, 30, 40, 50, 10, 45)
```

```
#a. Create a data frame for month, price per liter (php) and purchase-quantity (liter). Write the R script.
data.frame<- data.frame(Month, Price_per_liter_Php, Purchase_quantity_Liters)
data.frame
```

```
##      Month Price_per_liter_Php Purchase_quantity_Liters
## 1   Jan           52.50                25
## 2   Feb           57.25                30
## 3 March           60.00                40
## 4   Apr           65.00                50
## 5   May           74.25                10
## 6   June          54.00                45
```

```
Cruz<-data.frame(
  Month= c("Jan", "Feb", "March", "Apr", "May", "June"),
  Price_per_liter_Php= c(52.50,57.25,60.00,65.00,74.25,54.00),
  Purchase_quantity_Liters= c(25, 30, 40, 50, 10, 45)
)
Cruz
```

```
##      Month Price_per_liter_Php Purchase_quantity_Liters
## 1   Jan           52.50                25
## 2   Feb           57.25                30
## 3 March           60.00                40
## 4   Apr           65.00                50
## 5   May           74.25                10
## 6   June          54.00                45
```

#Output

```
#Month Price_per_liter_Php Purchase_quantity_Liters
```

```
#1   Jan           52.50                25
#2   Feb           57.25                30
#3 March           60.00                40
#4   Apr           65.00                50
#5   May           74.25                10
#6   June          54.00                45
```

```
#b.What is the average fuel expenditure of Mr. Cruz from Jan to June? Note: Use 'weighted.mean(liter,price)'.
weighted.mean(Price_per_liter_Php, Purchase_quantity_Liters)
```

```
## [1] 59.2625
```

#Output: 59.2625

#7. R has actually lots of built-in datasets. For example, the rivers data "gives the lengths (in miles) of the longest rivers in each state." #a.

```
data <- c(length(rivers), sum(rivers), mean(rivers), median(rivers), var(rivers), sd(rivers), min(rivers))
data
```

```
## [1] 141.0000 83357.0000 591.1844 425.0000 243908.4086 493.8708
## [7] 135.0000 3710.0000
```

#b. What are the results?

```
#141.0000 83357.0000 591.1844 425.0000 243908.4086 493.8708 135.0000 3710.0000
```

#8. The table below gives the 25 most powerful celebrities and their annual pay as ranked by the edition

#a. Create vectors according to the above table. Write the R scripts and its output.

```
Power_ranking<- c(1:25)
```

```
Celebrity_Name<- c("Tom Cruise", "Rolling Stones", "Oprah Winfrey", "U2", "Tiger Woods", "Steven Speilb
```

```
Pay<-c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 40, 47, 75, 25, 39, 45, 32,
```

```
Forbes<-data.frame(Power_ranking, Celebrity_Name, Pay)
```

```
Forbes
```

```
## Power_ranking Celebrity_Name Pay
## 1 1 Tom Cruise 67
## 2 2 Rolling Stones 90
## 3 3 Oprah Winfrey 225
## 4 4 U2 110
## 5 5 Tiger Woods 90
## 6 6 Steven Speilberg 332
## 7 7 Howard Stern 302
## 8 8 50 Cent 41
## 9 9 Cast of Sopranos 52
## 10 10 Dan Brown 88
## 11 11 Bruce Springsteen 55
## 12 12 Donald Trump 44
## 13 13 Muhammad Ali 55
## 14 14 Paul Mccartney 40
## 15 15 George Lucas 233
## 16 16 Elton John 34
## 17 17 David Letterman 40
## 18 18 Phil Mickelson 47
## 19 19 J.K Rowling 75
## 20 20 Bradd Pitt 25
## 21 21 Peter Jackson 39
## 22 22 Dr. Phi McGraw 45
## 23 23 Jay Lennon 32
## 24 24 Celine Dion 40
## 25 25 Kobe Bryant 31
```

#b. Modify the power ranking and pay of J.K. Rowling. Change power ranking to 15 and pay to 90. Write t

```
Forbes$Power_ranking[Forbes$Power_ranking==19]<-15
```

```
Forbes$Pay[Forbes$Pay==75]<-90
```

```
Forbes
```

```
## Power_ranking Celebrity_Name Pay
```

```
## 1      1      Tom Cruise  67
## 2      2      Rolling Stones  90
## 3      3      Oprah Winfrey 225
## 4      4      U2 110
## 5      5      Tiger Woods  90
## 6      6      Steven Speilberg 332
## 7      7      Howard Stern 302
## 8      8      50 Cent  41
## 9      9      Cast of Sopranos 52
## 10     10     Dan Brown  88
## 11     11     Bruce Springsteen 55
## 12     12     Donald Trump  44
## 13     13     Muhammad Ali  55
## 14     14     Paul Mccartney 40
## 15     15     George Lucas 233
## 16     16     Elton John  34
## 17     17     David Letterman 40
## 18     18     Phil Mickelson 47
## 19     15     J.K Rowling  90
## 20     20     Bradd Pitt  25
## 21     21     Peter Jackson 39
## 22     22     Dr. Phi McGraw 45
## 23     23     Jay Lennon  32
## 24     24     Celine Dion  40
## 25     25     Kobe Bryant  31
```

```
#c. Create an excel file from the table above and save it as csv file(PowerRanking). Import the csv file
library(readr)
PowerRankingList<-read.csv("PowerRanking.csv")
PowerRankingList
```

```
##      Power_Ranking  Celebrity_Name Pay
## 1      1      Tom Cruise  67
## 2      2      Rolling Stones  90
## 3      3      Oprha Winfrey 225
## 4      4      U2 110
## 5      5      Tiger Woods  90
## 6      6      Steven Spielberg 332
## 7      7      Howard Stern 302
## 8      8      50 Cent  41
## 9      9      Cast of Sopranos 52
## 10     10     Dan Brown  88
## 11     11     Bruce Springsteen 55
## 12     12     Donald Trump  44
## 13     13     Muhammad Ali  55
## 14     14     Paul McCartney 40
## 15     15     George Lucas 233
## 16     16     Elton John  34
## 17     17     David Letterman 40
## 18     18     Phil Mickelson 47
## 19     19     J.K Rowling  75
## 20     20     Bradd Pitt  25
## 21     21     Peter Jackson 39
## 22     22     Dr. Phil McGraw 45
```

```
## 23          23          Jay Lennon 32
## 24          24          Celine Dion 40
## 25          25          Kobe Bryant 31
```

```
#d. Access the rows 10 to 20 and save it as Ranks.RData. Write the R script and its output.
Ranks.RData<-PowerRankingList [10:20,]
Ranks.RData
```

```
##      Power_Ranking  Celebrity_Name Pay
## 10              10      Dan Brown  88
## 11              11 Bruce Springsteen 55
## 12              12      Donald Trump 44
## 13              13      Muhammad Ali 55
## 14              14      Paul McCartney 40
## 15              15      George Lucas 233
## 16              16      Elton John  34
## 17              17      David Letterman 40
## 18              18      Phil Mickelson 47
## 19              19      J.K Rowling  75
## 20              20      Bradd Pitt   25
```

```
#e. Describe its output. It states all of the list from the selected 10 to 20.
```

```
#9. Download the Hotels-Vienna https://tinyurl.com/Hotels-Vienna
#a. Import the excel file into your RStudio. What is the R script?
library(readxl)
hotellist<-read_excel("hotels-vienna.xlsx")
hotellist
```

```
## # A tibble: 428 x 24
##   country city_actual rating_count center1label center2label neighbourhood
##   <chr>    <chr>      <chr>      <chr>      <chr>      <chr>
## 1 Austria Vienna      36      City centre Donauturm  17. Hernals
## 2 Austria Vienna     189      City centre Donauturm  17. Hernals
## 3 Austria Vienna      53      City centre Donauturm  Alsergrund
## 4 Austria Vienna      55      City centre Donauturm  Alsergrund
## 5 Austria Vienna      33      City centre Donauturm  Alsergrund
## 6 Austria Vienna      25      City centre Donauturm  Alsergrund
## 7 Austria Vienna      57      City centre Donauturm  Alsergrund
## 8 Austria Vienna     161      City centre Donauturm  Alsergrund
## 9 Austria Vienna      50      City centre Donauturm  Alsergrund
## 10 Austria Vienna     NA      City centre Donauturm  Alsergrund
## # i 418 more rows
## # i 18 more variables: price <dbl>, city <chr>, stars <dbl>, ratingta <chr>,
## #   ratingta_count <chr>, scarce_room <dbl>, hotel_id <dbl>, offer <dbl>,
## #   offer_cat <chr>, year <dbl>, month <dbl>, weekend <dbl>, holiday <dbl>,
## #   distance <dbl>, distance_alter <dbl>, accommodation_type <chr>,
## #   nnights <dbl>, rating <chr>
```

```
#b. How many dimensions does the dataset have? What is the R script? What is its output?
dim(hotellist)
```

```
## [1] 428 24
```

```
#Answer: 428 24
```

```
#c. Select columns country, neighbourhood, price, stars, accomodation_type, and ratings. Write the R script  
hotellList[c("country", "neighbourhood", "price", "stars", "accommodation_type", "rating")]
```

```
## # A tibble: 428 x 6  
##   country neighbourhood price stars accommodation_type rating  
##   <chr>    <chr>      <dbl> <dbl> <chr>          <chr>  
## 1 Austria 17. Hernals      81     4 Apartment    4.4000000000000004  
## 2 Austria 17. Hernals      81     4 Hotel        3.9  
## 3 Austria Alsergrund      85     4 Hotel        3.7  
## 4 Austria Alsergrund      83     3 Hotel        4  
## 5 Austria Alsergrund      82     4 Hotel        3.9  
## 6 Austria Alsergrund     229     5 Apartment    4.8  
## 7 Austria Alsergrund     103     4 Hotel        3.9  
## 8 Austria Alsergrund     150     4 Hotel    4.5999999999999996  
## 9 Austria Alsergrund      80     2 Hotel        3.5  
## 10 Austria Alsergrund     153     3 Apartment    NA  
## # i 418 more rows
```

```
#d. Save the data as **new.RData to your RStudio. Write the R script.
```

```
new.RData<-hotellList  
new.RData
```

```
## # A tibble: 428 x 24  
##   country city_actual rating_count center1label center2label neighbourhood  
##   <chr>    <chr>      <chr>      <chr>      <chr>      <chr>  
## 1 Austria Vienna      36          City centre Donauturm    17. Hernals  
## 2 Austria Vienna     189          City centre Donauturm    17. Hernals  
## 3 Austria Vienna      53          City centre Donauturm    Alsergrund  
## 4 Austria Vienna      55          City centre Donauturm    Alsergrund  
## 5 Austria Vienna      33          City centre Donauturm    Alsergrund  
## 6 Austria Vienna      25          City centre Donauturm    Alsergrund  
## 7 Austria Vienna      57          City centre Donauturm    Alsergrund  
## 8 Austria Vienna     161          City centre Donauturm    Alsergrund  
## 9 Austria Vienna      50          City centre Donauturm    Alsergrund  
## 10 Austria Vienna     NA          City centre Donauturm    Alsergrund  
## # i 418 more rows  
## # i 18 more variables: price <dbl>, city <chr>, stars <dbl>, ratingta <chr>,  
## #   ratingta_count <chr>, scarce_room <dbl>, hotel_id <dbl>, offer <dbl>,  
## #   offer_cat <chr>, year <dbl>, month <dbl>, weekend <dbl>, holiday <dbl>,  
## #   distance <dbl>, distance_alter <dbl>, accommodation_type <chr>,  
## #   nnights <dbl>, rating <chr>
```

```
#e. Display the first six rows and last six rows of the new.RData. What is the R script?
```

```
head(new.RData)[[1]]
```

```
## [1] "Austria" "Austria" "Austria" "Austria" "Austria" "Austria"
```

```
tail(new.RData)[[1]]
```

```
## [1] "Austria" "Austria" "Austria" "Austria" "Austria" "Austria"
```

#10. Create a list of ten (10) vegetables you ate during your lifetime. If none, just list down. a. Write

```
gulayList<-list("spinach", "alogbati", "eggplant", "squash", "papaya", "kangkong", "hantak", "monggo", "kamote", "potato")
gulayList
```

```
## [[1]]
## [1] "spinach"
##
## [[2]]
## [1] "alogbati"
##
## [[3]]
## [1] "eggplant"
##
## [[4]]
## [1] "squash"
##
## [[5]]
## [1] "papaya"
##
## [[6]]
## [1] "kangkong"
##
## [[7]]
## [1] "hantak"
##
## [[8]]
## [1] "monggo"
##
## [[9]]
## [1] "potato"
##
## [[10]]
## [1] "kamote"
```

#b. Add 2 additional vegetables after the last vegetables in the list. What is the R script and its output?

```
gulayListNew<-append(gulayList,c("radish", "malunggay"))
gulayListNew
```

```
## [[1]]
## [1] "spinach"
##
## [[2]]
## [1] "alogbati"
##
## [[3]]
## [1] "eggplant"
##
## [[4]]
## [1] "squash"
##
```



```
## [[5]]
## [1] "papaya"
##
## [[6]]
## [1] "kangkong"
##
## [[7]]
## [1] "hantak"
##
## [[8]]
## [1] "monggo"
##
## [[9]]
## [1] "potato"
##
## [[10]]
## [1] "kamote"
##
## [[11]]
## [1] "radish"
##
## [[12]]
## [1] "malunggay"
```

```
#c. Add 4 additional vegetables after index 5. How many datapoints does your vegetable list have? What
gulayListNew<-append(gulayList,c("vege", "table", "namit", "yummy"), after=5)
gulayListNew
```

```
## [[1]]
## [1] "spinach"
##
## [[2]]
## [1] "alogbati"
##
## [[3]]
## [1] "eggplant"
##
## [[4]]
## [1] "squash"
##
## [[5]]
## [1] "papaya"
##
## [[6]]
## [1] "vege"
##
## [[7]]
## [1] "table"
##
## [[8]]
## [1] "namit"
##
## [[9]]
## [1] "yummy"
```

```
##
## [[10]]
## [1] "kangkong"
##
## [[11]]
## [1] "hantak"
##
## [[12]]
## [1] "monggo"
##
## [[13]]
## [1] "potato"
##
## [[14]]
## [1] "kamote"
```

```
length(gulayListNew)
```

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

```
#d. Remove the vegetables in index 5, 10, and 15. How many vegetables were left? Write the codes and it
gul4<-gulayListNew[c(-5, -10,-15)]
gul4
```

```
## [[1]]
## [1] "spinach"
##
## [[2]]
## [1] "alogbati"
##
## [[3]]
## [1] "eggplant"
##
## [[4]]
## [1] "squash"
##
## [[5]]
## [1] "vege"
##
## [[6]]
## [1] "table"
##
## [[7]]
## [1] "namit"
##
## [[8]]
## [1] "yummy"
##
## [[9]]
## [1] "hantak"
##
## [[10]]
## [1] "monggo"
```

```
##  
## [[11]]  
## [1] "potato"  
##  
## [[12]]  
## [1] "kamote"
```

```
length(gul4)
```

```
## [1] 12
```