

RWorkSheet_Malayasy#2

Andrew Miguel Malayas

2024-09-23

Using Vectors #1. Create a vector using : operator a. Sequence from -5 to 5. Write the R code and its output. Describe its output.

```
num <- -5:5  
num
```

```
## [1] -5 -4 -3 -2 -1 0 1 2 3 4 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. it displays the number by 0.2

```
seq(1, 3, by=0.2)
```

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

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

```
Workers <- 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, 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)
```

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

```
Workers <- 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, 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)  
Worka <- Workers [3]  
Worka
```

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

b. Access 2nd and 4th element, what are the values?

```
Workers <- 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, 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)
Workb <- Workers [c(2,4)]
Workb
```

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

c. Access all but the 4th and 12th element is not included. Write the R script and its output.

```
Workers <- 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, 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)
Workc <- Workers [-c(4,12)]
Workc
```

```
## [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 35 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)`.

a. Print the results. Then access `x[c("first", "third")]`. Describe the output. It displays the first and third number.

b. Write the code and its output.

```
#a & b
names <- c("first"=3, "second"=0, "third"=9)
ngalan <- names[c("first", "third")]
ngalan
```

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

#5. Create a sequence `x` from `-3:2`. a. Modify 2nd element and change it to 0; `x[2] <- 0` `x`

Describe the output. It change the value -2 to 0

b. Write the code and its output.

```
# a & b
x <- c(-3:2)
x[2] <- 0
x
```

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

#6. *The following data shows the diesel fuel purchased by Mr. Cruz. Month Jan Feb March Apr May June
Price per liter (PhP) 52.50 57.25 60.00 65.00 74.25 54.00 Purchase-quantity(Liters) 25 30 40 50 10 45

- Create a data frame for month, price per liter (php) and purchase-quantity (liter). Write the R scripts and its output.
- What is the average fuel expenditure of Mr. Cruz from Jan to June? Note: Use 'weighted.mean(liter, purchase)'. Write the R scripts and its output.

```
#a
Month <- c("Jan", "Feb", "March", "Apr", "May", "June")
Php <- c( 52.50, 57.25, 60.00, 65.00, 74.25, 54.00)
Liters <- c( 25, 30, 40, 50, 10, 45)
Data <- data.frame(Month, Php, Liters)
Data
```

```
##   Month   Php 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
Average <- weighted.mean(Php, Liters)
Average
```

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

#7. R has actually lots of built-in datasets. For example, the rivers data “gives the lengths (in miles) of 141 “major” rivers in North America, as compiled by the US Geological Survey”. a. Type “rivers” in your R console. Create a vector data with 7 elements, containing the number of elements (length) in rivers, their sum (sum), mean (mean), median(median), variance(var), standard deviation(sd), minimum (min) and maximum (max).

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

```
rivers
```

```
##   [1] 735 320 325 392 524 450 1459 135 465 600 330 336 280 315 870
##  [16] 906 202 329 290 1000 600 505 1450 840 1243 890 350 407 286 280
##  [31] 525 720 390 250 327 230 265 850 210 630 260 230 360 730 600
##  [46] 306 390 420 291 710 340 217 281 352 259 250 470 680 570 350
##  [61] 300 560 900 625 332 2348 1171 3710 2315 2533 780 280 410 460 260
##  [76] 255 431 350 760 618 338 981 1306 500 696 605 250 411 1054 735
##  [91] 233 435 490 310 460 383 375 1270 545 445 1885 380 300 380 377
## [106] 425 276 210 800 420 350 360 538 1100 1205 314 237 610 360 540
## [121] 1038 424 310 300 444 301 268 620 215 652 900 525 246 360 529
## [136] 500 720 270 430 671 1770
```

- b. What are the results? [1] 735 320 325 392 524 450 1459 135 465 600 330 336 280 315 [15] 870 906 202 329 290 1000 600 505 1450 840 1243 890 350 407 [29] 286 280 525 720 390 250 327 230 265 850 210 630 260 230 [43] 360 730 600 306 390 420 291 710 340 217 281 352 259 250 [57] 470 680 570 350 300 560 900 625 332 2348 1171 3710 2315 2533 [71] 780 280 410 460 260 255 431 350 760 618 338 981 1306 500 [85] 696 605 250 411 1054 735 233 435 490 310 460 383 375 1270 [99] 545 445 1885 380 300 380 377 425 276 210 800 420 350 360 [113] 538 1100 1205 314 237 610 360 540 1038 424 310 300 444 301 [127] 268 620 215 652 900 525 246 360 529 500 720 270 430 671 [141] 1770
- c. Write the R scripts and its outputs.

```
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
```

#8. The table below gives the 25 most powerful celebrities and their annual pay as ranked by the editions of Forbes magazine and as listed on the Forbes.com website.

Figure 1: Forbes Ranking a. Create vectors according to the above table. Write the R scripts and its output.

- b. Modify the power ranking and pay of J.K. Rowling. Change power ranking to 15 and pay to 90. Write the R scripts and its output.

```
# a
celebritydata <- data.frame(
  PowerRanking = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                  11, 12, 13, 14, 15, 16, 17, 18,
                  19, 20, 21, 22, 23, 24, 25),
  CelebrityName = c("Tom Cruise", "Rolling Stones", "Oprah Winfrey", "U2",
                    "Tiger Woods", "Steven Spielberg", "Howard Stern",
                    "50 Cent", "Cast of the Sopranos", "Dan Brown",
                    "Bruce Springsteen", "Donald Trump", "Muhammad Ali",
                    "Paul McCartney", "George Lucas", "Elton John",
                    "David Letterman", "Phil Mickelson", "J.K Rowling",
                    "Brad Pitt", "Peter Jackson", "Dr. Phil McGraw",
                    "Jay Leno", "Celine Dion", "Kobe Bryant"),
  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, 40, 31)
)

#b
celebritydata[celebritydata$CelebrityName == "J.K Rowling", "Power_Ranking"] <- 15
celebritydata[celebritydata$CelebrityName == "J.K Rowling", "Pay"] <- 90

celebritydata
```

```
##      PowerRanking      CelebrityName Pay Power_Ranking
## 1              1      Tom Cruise  67             NA
## 2              2    Rolling Stones  90             NA
## 3              3    Oprah Winfrey 225             NA
```

## 4	4	U2	110	NA
## 5	5	Tiger Woods	90	NA
## 6	6	Steven Spielberg	332	NA
## 7	7	Howard Stern	302	NA
## 8	8	50 Cent	41	NA
## 9	9	Cast of the Sopranos	52	NA
## 10	10	Dan Brown	88	NA
## 11	11	Bruce Springsteen	55	NA
## 12	12	Donald Trump	44	NA
## 13	13	Muhammad Ali	55	NA
## 14	14	Paul McCartney	40	NA
## 15	15	George Lucas	233	NA
## 16	16	Elton John	34	NA
## 17	17	David Letterman	40	NA
## 18	18	Phil Mickelson	47	NA
## 19	19	J.K Rowling	90	15
## 20	20	Brad Pitt	25	NA
## 21	21	Peter Jackson	39	NA
## 22	22	Dr. Phil McGraw	45	NA
## 23	23	Jay Leno	32	NA
## 24	24	Celine Dion	40	NA
## 25	25	Kobe Bryant	31	NA

- c. Create an excel file from the table above and save it as csv file(PowerRanking). Import the csv file into the RStudio. What is the R script?

```
write.csv(celebritydata, file = "PowerRanking.csv", row.names = FALSE)

celebritydata_imported <- read.csv("PowerRanking.csv")

celebritydata_imported
```

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

## 21	21	Peter Jackson	39	NA
## 22	22	Dr. Phil McGraw	45	NA
## 23	23	Jay Leno	32	NA
## 24	24	Celine Dion	40	NA
## 25	25	Kobe Bryant	31	NA

d. Access the rows 10 to 20 and save it as Ranks.RData. Write the R script and its output.

```
ranked_subset <- celebritydata[10:20, ]
save(ranked_subset, file = "Ranks.RData")
ranked_subset
```

##	PowerRanking	CelebrityName	Pay	Power_Ranking
## 10	10	Dan Brown	88	NA
## 11	11	Bruce Springsteen	55	NA
## 12	12	Donald Trump	44	NA
## 13	13	Muhammad Ali	55	NA
## 14	14	Paul McCartney	40	NA
## 15	15	George Lucas	233	NA
## 16	16	Elton John	34	NA
## 17	17	David Letterman	40	NA
## 18	18	Phil Mickelson	47	NA
## 19	19	J.K Rowling	90	15
## 20	20	Brad Pitt	25	NA

e. Describe its output. It shows the data from 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)
hotels_data <- read_excel("hotels-vienna.xlsx")
hotels_data
```

```
## # 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?

```
dataset_dimensions <- dim(hotels_data)
dataset_dimensions
```

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

c. Select columns country, neighbourhood, price, stars, accommodation_type, and ratings. Write the R script.

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
selected_columns <- hotels_data %>% select(country, neighbourhood, price, stars, accommodation_type, rating)
selected_columns
```

```
## # 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.

```
library(readxl)
```

```
library(dplyr)
```

```
hotels_data <- read_excel("hotels-vienna.xlsx")
colnames(hotels_data)
```

```
## [1] "country"          "city_actual"      "rating_count"
## [4] "center1label"      "center2label"     "neighbourhood"
## [7] "price"             "city"             "stars"
## [10] "ratingta"          "ratingta_count"   "scarce_room"
## [13] "hotel_id"          "offer"            "offer_cat"
## [16] "year"              "month"            "weekend"
## [19] "holiday"           "distance"          "distance_alter"
## [22] "accommodation_type" "nnights"           "rating"
```

```
selected_columns <- hotels_data %>% select(country, neighbourhood, price, stars, accommodation_type, ra
save(selected_columns, file = "new.RData")
```

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

```
first_six_rows <- head(selected_columns)
print(first_six_rows)
```

```
## # A tibble: 6 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
```

```
last_six_rows <- tail(selected_columns)
print(last_six_rows)
```

```
## # A tibble: 6 x 6
##   country neighbourhood price stars accommodation_type rating
##   <chr>    <chr>          <dbl> <dbl> <chr>              <chr>
## 1 Austria Wieden          73    3 Hotel              3.4
## 2 Austria Wieden         109    3 Apartment          5
## 3 Austria Wieden         185    5 Hotel              4.3
## 4 Austria Wieden         100    4 Hotel              4.4000000000000004
## 5 Austria Wieden          58    3 Hotel              3.2
## 6 Austria Wieden         110   3.5 Apartment         4
```

#10. Create a list of ten (10) vegetables you ate during your lifetime. If none, just list down. a. Write the R scripts and its output.

```
vegetables <- c("Carrot", "Broccoli", "Spinach", "Tomato", "Cucumber", "Lettuce", "Pepper", "Zucchini",
vegetables
```

```
## [1] "Carrot"          "Broccoli"         "Spinach"          "Tomato"           "Cucumber"
## [6] "Lettuce"         "Pepper"           "Zucchini"         "Cauliflower"     "Onion"
```

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


```
vegetables <- c("Carrot", "Broccoli", "Spinach", "Tomato", "Cucumber", "Lettuce", "Pepper", "Zucchini",
vegetables <- c(vegetables, "Eggplant", "Mushroom")
```

- c. Add 4 additional vegetables after index 5. How many datapoints does your vegetable list have? What is the R script and its output?

```
vegetables <- c("Carrot", "Broccoli", "Spinach", "Tomato", "Cucumber", "Lettuce", "Pepper", "Zucchini",
vegetables <- c(vegetables, "Eggplant", "Mushroom")
vegetables <- append(vegetables, c("Pumpkin", "Radish", "Kale", "Beetroot"), after = 5)
vegetables
```

```
## [1] "Carrot"      "Broccoli"    "Spinach"     "Tomato"      "Cucumber"
## [6] "Pumpkin"     "Radish"      "Kale"        "Beetroot"    "Lettuce"
## [11] "Pepper"      "Zucchini"    "Cauliflower" "Onion"       "Eggplant"
## [16] "Mushroom"
```

```
length(vegetables)
```

```
## [1] 16
```

- d. Remove the vegetables in index 5, 10, and 15. How many vegetables were left? Write the codes and its output.

```
vegetables <- c("Carrot", "Broccoli", "Spinach", "Tomato", "Cucumber", "Lettuce", "Pepper", "Zucchini",
vegetables <- c(vegetables, "Eggplant", "Mushroom")
vegetables <- append(vegetables, c("Pumpkin", "Radish", "Kale", "Beetroot"), after = 5)
vegetables <- vegetables[-c(5, 10, 15)]
length(vegetables)
```

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

Prepared by: JOYCE F. JAMILE Instructor Without ethical considerations, AI becomes a tool of chaos and harm.