

RWorksheet_Buenvenida#2

me

2024-09-25

1. Create a vector using : operator

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

```
seq(-5, 5)
```

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

-is a numeric vector containing 11 elements from -5 to 5.

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

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

```
x
```

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

-The value of x are integers ranging from 1 to 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.

```
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
```

-The sequence starts from 1 and increments by 0.2 until it reaches 3.

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,

4.

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

```
ages <- 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)
ages[3]
```

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

-The value is 22

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

```
ages[c(2,4)]
```

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

-The values are 28 and 36

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

```
ages[-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 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.

b. Write the code and its output.

```
x <- c("first"=3, "second"=0, "third"=9)
names(x)
```

```
## [1] "first" "second" "third"
```

```
x[c("first", "third")]
```

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

-The first element is named “first” and has a value of 3. The second element is named “third” and has a value of 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.

b. Write the code and its output.

```
x <- -3:2
x[2] <- 0
x
```

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

-A numeric vector `x` with elements -3, 0, 1, and 2, where the 2nd element is modified to 0.

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

- a. Create a data frame for month, price per liter (php) and purchase-quantity (liter). Write the R scripts and its output.

```
month <- c("Jan", "Feb", "March", "Apr", "May", "June")
price_per_liter <- c(52.50, 57.25, 60.00, 65.00, 74.25, 54.00)
purchase_quantity <- c(25, 30, 40, 50, 10, 45)
```

```
fuel_purchases <- data.frame(
  Month = month,
  Price_per_liter = price_per_liter,
  Purchase_quantity = purchase_quantity
)
```

```
print(fuel_purchases)
```

```
##   Month Price_per_liter Purchase_quantity
## 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, purchase)'. Write the R scripts and its output.

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

```
print(fuel_data)
```

```
##   Month Price_per_liter Purchase_quantity
## 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
```

```
average_expenditure <- weighted.mean(fuel_data$Price_per_liter, fuel_data$Purchase_quantity)
cat("The average fuel expenditure from January to June is:", average_expenditure, "PhP\n")
```

```
## The average fuel expenditure from January to June is: 59.2625 PhP
```

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”.
- 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))`
 - What are the results?
 - Write the R scripts and its outputs.

Results:

```
data(rivers)
```

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

```
print(vector_data)
```

##	length	sum	mean	median	variance	sd
##	141.0000	83357.0000	591.1844	425.0000	243908.4086	493.8708
##	min	max				
##	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. a. Create vectors according to the above table. Write the R scripts and its output.

```
celebrity_names <- 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", "Bradd Pitt", "Peter Jackson",
                    "Dr. Phil McGraw", "Jay Lenon", "Celine Dion", "Kobe Bryant")

power_ranking <- c(1:13, 14:25)

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)
```

celebrity_names

## [1]	"Tom Cruise"	"Rolling Stones"	"Oprah Winfrey"
## [4]	"U2"	"Tiger Woods"	"Steven Spielberg"
## [7]	"Howard Stern"	"50 Cent"	"Cast of the Sopranos"
## [10]	"Dan Brown"	"Bruce Springsteen"	"Donald Trump"
## [13]	"Muhammad Ali"	"Paul McCartney"	"George Lucas"
## [16]	"Elton John"	"David Letterman"	"Phil Mickelson"
## [19]	"J.K Rowling"	"Bradd Pitt"	"Peter Jackson"
## [22]	"Dr. Phil McGraw"	"Jay Lenon"	"Celine Dion"
## [25]	"Kobe Bryant"		

power_ranking

```
## [1] 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
```

pay

```
## [1] 67 90 225 110 90 332 302 41 52 88 55 44 55 40 233 34 40 47 75
## [20] 25 39 45 32 40 31
```

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.

```
power_ranking[power_ranking == 19] <- 15
pay[celebrity_names == "J.K Rowling"] <- 90
```

celebrity_names

## [1]	"Tom Cruise"	"Rolling Stones"	"Oprah Winfrey"
## [4]	"U2"	"Tiger Woods"	"Steven Spielberg"
## [7]	"Howard Stern"	"50 Cent"	"Cast of the Sopranos"
## [10]	"Dan Brown"	"Bruce Springsteen"	"Donald Trump"
## [13]	"Muhammad Ali"	"Paul McCartney"	"George Lucas"
## [16]	"Elton John"	"David Letterman"	"Phil Mickelson"
## [19]	"J.K Rowling"	"Bradd Pitt"	"Peter Jackson"
## [22]	"Dr. Phil McGraw"	"Jay Lenon"	"Celine Dion"
## [25]	"Kobe Bryant"		

```
power_ranking
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 15 20 21 22 23 24 25
pay
```

```
## [1] 67 90 225 110 90 332 302 41 52 88 55 44 55 40 233 34 40 47 90
```

```
## [20] 25 39 45 32 40 31
```

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?

```
PowerRanking <- 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",
                    "Donald Trump", "Muhammad Ali", "Paul McCartney", "George Lucas", "Elton John", "David Letterman",
                    "Phil Mickelson", "J.K Rowling", "Bradd Pitt", "Peter Jackson", "Dr. Phil McGraw", "Jay Lenon", "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)
)
```

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

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

```
print(PowerRanking)
```

##	PowerRanking	CelebrityName	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 Spielberg	332
## 7	7	Howard Stern	302
## 8	8	50 Cent	41
## 9	9	Cast of the 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 Lenon	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.

```
power_ranking <- c(1:13, 14:25)
celebrity_name <- 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",
  "Bradd Pitt", "Peter Jackson", "Dr. Phil McGraw", "Jay Lenon",
  "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)
Forbes_Ranking <- data.frame(power_ranking, celebrity_name, pay)
Forbes_Ranking
```

##	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 Spielberg	332
## 7	7	Howard Stern	302
## 8	8	50 Cent	41
## 9	9	Cast of the 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 Lenon	32
## 24	24	Celine Dion	40
## 25	25	Kobe Bryant	31

```
Ranks <- Forbes_Ranking[10:20, ]
Ranks
```

##	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
```

```
save(Ranks, file = "Ranks.RData")
```

e. Describe its output.

- The output is given on the rankings and the payings of each celebrities worldwide.

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_vienna <- read_excel("/cloud/project/Buenvenida/hotels-vienna.xlsx")
```

b. How many dimensions does the dataset have? What is the R script? What is its output?

```
dim(hotels_vienna)
```

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

c. Select columns country, neighbourhood, price, stars, accomodation_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
```

```
hotels_vienna_selected <- hotels_vienna %>%
```

```
  select(country, neighbourhood, price, stars, accommodation_type, rating)
```

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

```
save(hotels_vienna_selected, file = "vienna-hotel-AnabolicRepo.Rdata")
```

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

```
load("vienna-hotel-AnabolicRepo.Rdata")
```

```
head(hotels_vienna_selected, 6)
```

```
## # 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
```

```
tail(hotels_vienna_selected, 6)
```



```
## # 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 <- list("broccoli", "Mushroom", "Lettuce", "Potato", "Okra", "Squash", "Beans", "Mushroom", "Onions")
print("List of Vegetables:")
```

```
## [1] "List of Vegetables:"
```

```
print(vegetables)
```

```
## [[1]]
## [1] "broccoli"
##
## [[2]]
## [1] "Mushroom"
##
## [[3]]
## [1] "Lettuce"
##
## [[4]]
## [1] "Potato"
##
## [[5]]
## [1] "Okra"
##
## [[6]]
## [1] "Squash"
##
## [[7]]
## [1] "Beans"
##
## [[8]]
## [1] "Mushroom"
##
## [[9]]
## [1] "Onions"
##
## [[10]]
## [1] "Cabbage"
```

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

```
vegetables <- c(vegetables, list("Carrot", "Peas"))
print("Updated List of Vegetables:")
```

```
## [1] "Updated List of Vegetables:"
```

```
print(vegetables)
```

```
## [[1]]  
## [1] "broccoli"  
##  
## [[2]]  
## [1] "Mushroom"  
##  
## [[3]]  
## [1] "Lettuce"  
##  
## [[4]]  
## [1] "Potato"  
##  
## [[5]]  
## [1] "Okra"  
##  
## [[6]]  
## [1] "Squash"  
##  
## [[7]]  
## [1] "Beans"  
##  
## [[8]]  
## [1] "Mushroom"  
##  
## [[9]]  
## [1] "Onions"  
##  
## [[10]]  
## [1] "Cabbage"  
##  
## [[11]]  
## [1] "Carrot"  
##  
## [[12]]  
## [1] "Peas"
```

- 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(vegetables[1:5], list("Eggplant", "Spinach", "Tomato", "Cucumber"), vegetables[6:length  
print("Updated List of Vegetables:")
```

```
## [1] "Updated List of Vegetables:"
```

```
print(vegetables)
```

```
## [[1]]  
## [1] "broccoli"  
##  
## [[2]]  
## [1] "Mushroom"  
##  
## [[3]]  
## [1] "Lettuce"
```

```
##
## [[4]]
## [1] "Potato"
##
## [[5]]
## [1] "Okra"
##
## [[6]]
## [1] "Eggplant"
##
## [[7]]
## [1] "Spinach"
##
## [[8]]
## [1] "Tomato"
##
## [[9]]
## [1] "Cucumber"
##
## [[10]]
## [1] "Squash"
##
## [[11]]
## [1] "Beans"
##
## [[12]]
## [1] "Mushroom"
##
## [[13]]
## [1] "Onions"
##
## [[14]]
## [1] "Cabbage"
##
## [[15]]
## [1] "Carrot"
##
## [[16]]
## [1] "Peas"
```

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

```
vegetables <- vegetables[-c(5, 10, 15)]
print("Updated List of Vegetables:")
```

```
## [1] "Updated List of Vegetables:"
```

```
print(vegetables)
```

```
## [[1]]
## [1] "broccoli"
##
## [[2]]
## [1] "Mushroom"
##
```

```
## [[3]]
## [1] "Lettuce"
##
## [[4]]
## [1] "Potato"
##
## [[5]]
## [1] "Eggplant"
##
## [[6]]
## [1] "Spinach"
##
## [[7]]
## [1] "Tomato"
##
## [[8]]
## [1] "Cucumber"
##
## [[9]]
## [1] "Beans"
##
## [[10]]
## [1] "Mushroom"
##
## [[11]]
## [1] "Onions"
##
## [[12]]
## [1] "Cabbage"
##
## [[13]]
## [1] "Peas"
```