RWorksheet_Alpanghe#2

Jersey Gabriel E. Alpanghe

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.

```
vec<- -5:5

print(vec)
```

```
##  [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 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.

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

print (vec)
```

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

This output represents a vector starting at 1 and ending at 3, with increments of 0.2, showcasing evenly spaced decimal values.

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.

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

third <-ages[3]
print(third)
```

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

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

```
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

secondandfourth <-ages[c(2, 4)]
print(secondandfourth)
```

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

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

```
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

exclude_4th_12th <-ages[-c(4, 12)]
print(exclude_4th_12th)
```

```
##  [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.

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

print(num)
```

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

b. Write the code and its output.

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

print(num)
```

```
## 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.

The values of the first element is 3, the second element now has the value of zero after changing it to 0 while the third element is 9.

b. Write the code and its output.

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

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

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

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

```
months <- c("Jan", "Feb", "March", "Apr", "May", "June")
price <- c(53.50, 57.25, 60.00, 65.00, 74.25, 54.00)
purchases <- c(24, 30, 40, 50, 10, 45)

results <- data.frame(Month = months, Price_per_Liter = price, Purchase_Quantity = purchases)

results
```

```
##   Month Price_per_Liter Purchase_Quantity
## 1   Jan           53.50                 24
## 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.

```
months <- c("Jan", "Feb", "March", "Apr", "May", "June")
price <- c(53.50, 57.25, 60.00, 65.00, 74.25, 54.00)
purchases <- c(24, 30, 40, 50, 10, 45)



fuel_per_month <-(price * purchases)

average <- weighted.mean(price, purchases)

print(average)
```

```
## [1] 59.41709
```

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))

b. What are the results?

Length: 141 (the number of rivers) Sum: 5834 (total length of all rivers) Mean: 41.2922 (average length) Median: 25.0000 (middle value) Variance: 542.7324 (measure of spread) Standard Deviation: 23.2866 (average distance from the mean) Minimum: 3 (shortest river) Maximum: 130 (longest river)

c. Write the R scripts and its outputs.

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

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.

```
power_ranking <- 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)

celebrity <- c("Tom Cruise","Rolling Stones", "Oprah Winfrey", "U2","Tiger Woods", "Steven Spielberg",

pay <- c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 40, 47, 75, 25, 39, 45, 3

celebrity_data <- data.frame(PowerRanking = power_ranking,
                             Celebrity = celebrity,
                             Pay = pay)
```

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 <- 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)

celebrity <- c("Tom Cruise","Rolling Stones", "Oprah Winfrey", "U2","Tiger Woods", "Steven Spielberg",

pay <- c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 40, 47, 75, 25, 39, 45, 3

celebrity_data <- data.frame(PowerRanking = power_ranking,
                             Celebrity = celebrity,
                             Pay = pay)

celebrity_data[celebrity_data$Celebrity == "J.K. Rowling", c("PowerRanking", "Pay")] <- c(15, 90)

print(celebrity_data)
```

```
##    PowerRanking            Celebrity 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           15         J.K. Rowling  90
## 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
```

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?

```
power_ranking <- 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)

celebrity <- c("Tom Cruise","Rolling Stones", "Oprah Winfrey", "U2","Tiger Woods", "Steven Spielberg",

pay <- c(67, 90, 225, 110, 90, 332, 302, 41, 52, 88, 55, 44, 55, 40, 233, 34, 40, 47, 75, 25, 39, 45, 3
```

```r
celebrity_data <- data.frame(Power_Ranking = power_ranking,
                             Celebrity = celebrity,
                             Pay = pay)

celebrity_data[celebrity_data$Celebrity == "J.K. Rowling", c("PowerRanking", "Pay")] <- c(15, 90)


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

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

```
##    Power_Ranking            Celebrity Pay PowerRanking
## 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           Bradd Pitt  25           NA
## 21            21        Peter Jackson  39           NA
## 22            22      Dr. Phil McGraw  45           NA
## 23            23            Jay Lenon  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.

```r
selected_ranks <- celebrity_data[10:20, ]
save(selected_ranks, file = "Ranks.RData")

load("Ranks.RData")
print(selected_ranks)
```

```
##    Power_Ranking         Celebrity Pay PowerRanking
## 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        Bradd Pitt  25          NA
```

e. Describe its output. The celebrities are all ranked accordingly along with their pay in the power rankings.

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/hotels-vienna.xlsx")
```

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

```
library(readxl)
hotels_vienna <- read_excel("/cloud/project/hotels-vienna.xlsx")

dimensions <- dim(hotels_vienna)
print(dimensions)
```

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

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

```
library(readxl)
hotels_vienna <- read_excel("/cloud/project/hotels-vienna.xlsx")

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_vienna %>%
  select(country, neighbourhood, price, stars, accommodation_type, rating)
print(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)
hotels_vienna <- read_excel("/cloud/project/hotels-vienna.xlsx")

save(hotels_vienna, file = "new.RData")
```

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

```
load("new.RData")
print(head(hotels_vienna))
```

```
## # A tibble: 6 x 24
##   country city_actual rating_count center1label center2label neighbourhood price
##   <chr>   <chr>       <chr>        <chr>        <chr>        <chr>         <dbl>
## 1 Austria Vienna      36           City centre  Donauturm    17. Hernals      81
## 2 Austria Vienna      189          City centre  Donauturm    17. Hernals      81
## 3 Austria Vienna      53           City centre  Donauturm    Alsergrund       85
## 4 Austria Vienna      55           City centre  Donauturm    Alsergrund       83
## 5 Austria Vienna      33           City centre  Donauturm    Alsergrund       82
## 6 Austria Vienna      25           City centre  Donauturm    Alsergrund      229
## # i 17 more variables: 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>
```

```
print(tail(hotels_vienna))
```

```
## # A tibble: 6 x 24
##   country city_actual rating_count center1label center2label neighbourhood price
##   <chr>   <chr>       <chr>        <chr>        <chr>        <chr>         <dbl>
## 1 Austria Vienna      53           City centre  Donauturm    Wieden           73
## 2 Austria Vienna      2            City centre  Donauturm    Wieden          109
## 3 Austria Vienna      145          City centre  Donauturm    Wieden          185
## 4 Austria Vienna      112          City centre  Donauturm    Wieden          100
## 5 Austria Vienna      169          City centre  Donauturm    Wieden           58
## 6 Austria Vienna      80           City centre  Donauturm    Wieden          110
## # i 17 more variables: 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>
```

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", "Peppers", "Lettuce", "Zucchini"

vegetables
```

```
##  [1] "Carrot"   "Broccoli" "Spinach"  "Tomato"   "Cucumber" "Peppers"
##  [7] "Lettuce"  "Zucchini" "Onion"    "Eggplant"
```

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", "Peppers", "Lettuce", "Zucchini"

vegetables <- c(vegetables, "Radish", "Kale")

vegetables
```

```
##  [1] "Carrot"   "Broccoli" "Spinach"  "Tomato"   "Cucumber" "Peppers"
##  [7] "Lettuce"  "Zucchini" "Onion"    "Eggplant" "Radish"   "Kale"
```

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", "Peppers", "Lettuce", "Zucchini"

vegetables <- append(vegetables, c("Asparagus", "Cauliflower", "Beetroot", "Pumpkin"), after = 5)

num_datapoints <- length(vegetables)

print(vegetables)
```

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

```
print(num_datapoints)
```

```
## [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", "Peppers", "Lettuce", "Zucchini"

vegetables <- append(vegetables, c("Asparagus", "Cauliflower", "Beetroot", "Pumpkin"), after = 5)

num_datapoints <- length(vegetables)

vegetables <- vegetables[-c(5, 10, 15)]

num_left <- length(vegetables)

print(vegetables)
```

```
##  [1] "Carrot"      "Broccoli"    "Spinach"     "Tomato"      "Asparagus"
##  [6] "Cauliflower" "Beetroot"    "Pumpkin"     "Lettuce"     "Zucchini"
## [11] "Onion"       "Eggplant"    "Kale"
```

```
print(num_left)
```

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