```r
#1. Create a vector using : operator
#a. Sequence from -5 to 5. Write the R code and its output.
#Describe its output.
seq_neg5_to_5 <- -5 : 5
seq_neg5_to_5
```

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

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

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

```r
#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
```

```r
#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.
```

```r
#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
paste(workers[3])
```

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

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

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

```r
#Access all but the 4th and 12th element is not
#included. Write the R script and its output.
workers[-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
```

```r
#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)
x
```

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

```r
#b. Write the code and its output.
print(x[c("first", "third")])
```

```
## first third
```

```
##     3     9
```

```
x <- c(-3:2)
```

```
#5. Create a sequence x from -3:2.
x <- c(-3:2)
#a. Modify 2nd element and change it to 0.
x[2] <- 0
#Describe the output
#b. Write the code and its output.
x
```

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

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

months <- c("Jan", "Feb", "March", "April", "May", "June")
price_per_liter <- c(52.50, 57.25, 62.00, 65.00, 74.25, 54.00)
purchase_quantity <- c(25L, 30L, 40L, 50L, 10L, 45L)

gas <- data.frame (Month = months, Price_per_liter_PhP = price_per_liter, Purchase_quantity_Liters = pu
gas
```

```
##   Month Price_per_liter_PhP Purchase_quantity_Liters
## 1   Jan               52.50                       25
## 2   Feb               57.25                       30
## 3 March               62.00                       40
## 4 April               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.
gas_ex <- weighted.mean(price_per_liter, purchase_quantity)
print(paste("Average Fuel Expenditure per liter from Jan to June:", gas_ex, "PhP"))
```

```
## [1] "Average Fuel Expenditure per liter from Jan to June: 59.6625 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".

#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).
#b. What are the results?
#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.

```
#a. Create vectors according to the above table.
#Write the R scripts and its output.
power_ranking <- 1:25
celebrity_name <- c("Tom Cruise", "Rolling Stones", "Oprah Winfrey", "U2", "Tiger Woods", "Steven Spiell
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, CelebrityName = celebrity_name, Pay = pay)
celebrity_data
```

```
##    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 Leno  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 the
#R scripts and its output.
celebrity_data[celebrity_data$CelebrityName == "J.K. Rowling", "PowerRanking"] <- 15
celebrity_data[celebrity_data$CelebrityName == "J.K. Rowling", "Pay"] <- 90

celebrity_data
```

```
##    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           15         J.K. Rowling  90
## 20           20           Bradd Pitt  25
## 21           21        Peter Jackson  39
## 22           22      Dr. Phil McGraw  45
## 23           23             Jay Leno  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?
write.csv(celebrity_data, file = "PowerRanking.csv", row.names = FALSE)

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

celebrity_data_imported
```

```
##    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           15         J.K. Rowling  90
## 20           20           Bradd Pitt  25
## 21           21        Peter Jackson  39
## 22           22      Dr. Phil McGraw  45
## 23           23             Jay Leno  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.
```

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

```r
#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>
```

```r
#b.How many dimensions does the dataset have?
dataset_dimensions <- dim(hotels_data)
dataset_dimensions
```

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

```r
#c. Select columns country, neighbourhood, price, stars, accomodation_type, and ratings. Write the R sc
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
```

```r
selected_columns <- hotels_data %>% select(country, neighbourhood, price, stars, accommodation_type, ra
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?
load("new.RData")

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("Talong", "Sitaw", "Ampalaya", "Okra", "Kalabasa", "Malunggay", "Kangkong", "Pechay", "S
vegetables
```

```
##  [1] "Talong"    "Sitaw"     "Ampalaya"  "Okra"      "Kalabasa"  "Malunggay"
##  [7] "Kangkong"  "Pechay"    "Sayote"    "Patola"
```

```
#b. Add 2 additional vegetables after the last vegetables in the list. What is the R script and its outp
vegetables <- c(vegetables, "Labanos", "Gabi")
vegetables
```

```
##  [1] "Talong"    "Sitaw"     "Ampalaya"  "Okra"      "Kalabasa"  "Malunggay"
##  [7] "Kangkong"  "Pechay"    "Sayote"    "Patola"    "Labanos"   "Gabi"
```

```
#c. Add 4 additional vegetables after index 5. How many data points does your vegetable list have? What
vegetables <- append(vegetables, c("Mustasa", "Kamote", "Upo", "Alugbati"), after = 5)
vegetables
```

```
##  [1] "Talong"    "Sitaw"     "Ampalaya"  "Okra"      "Kalabasa"  "Mustasa"
##  [7] "Kamote"    "Upo"       "Alugbati"  "Malunggay" "Kangkong"  "Pechay"
## [13] "Sayote"    "Patola"    "Labanos"   "Gabi"
```

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

```
##  [1] "Talong"   "Sitaw"    "Ampalaya" "Okra"     "Mustasa"  "Kamote"
##  [7] "Upo"      "Alugbati" "Kangkong" "Pechay"   "Sayote"   "Patola"
## [13] "Gabi"
```