

HW2

Lin Pin Tzu (Ruby)

2022-07-05

```
Vector1 <- (c( 10, 19, 121, 83, 63, 7, 77, 61, 51, 97, 123, 41))
Vector1
```

```
## [1] 10 19 121 83 63 7 77 61 51 97 123 41
```

1) For the vector given above, use and show two methods of R coding to extract the first element and the last element.

```
Vector1[c(1,12)]
```

```
## [1] 10 41
```

```
Vector1[c(-2:-11)]
```

```
## [1] 10 41
```

2) For the vector given above, use and show two methods of R coding to extract all of the elements that are less than 60.

```
Vector1[Vector1<60]
```

```
## [1] 10 19 7 51 41
```

```
Vector1[!(Vector1 > 60)]
```

```
## [1] 10 19 7 51 41
```

3) For the vector given above, use and show two methods of R coding to extract all numbers that are not divisible by 2 or 3.

```
Vector1[!Vector1 %%2 == 0 | !Vector1 %%3 == 0]
```

```
## [1] 10 19 121 83 63 7 77 61 51 97 123 41
```

```
Vector1[Vector1 %%2 != 0 | Vector1 %%3 != 0]
```

```
## [1] 10 19 121 83 63 7 77 61 51 97 123 41
```

```
Vector1[]
```

```
## [1] 10 19 121 83 63 7 77 61 51 97 123 41
```

4) Use and show two R coding methods to confirm that Vector1 does not have missing values

```
Vector1[is.na(Vector1)]

## numeric(0)
which(is.na(Vector1))

## integer(0)
sum(is.na(Vector1)) #It shows zero which means here we don't have missing value

## [1] 0
is.na(Vector1)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
#There is no missing value in here

myList <- list(TRUE,12.35,"pear",48,c=3:8,list(23,"team"))
myList

## [[1]]
## [1] TRUE
##
## [[2]]
## [1] 12.35
##
## [[3]]
## [1] "pear"
##
## [[4]]
## [1] 48
##
## $c
## [1] 3 4 5 6 7 8
##
## [[6]]
## [[6]][[1]]
## [1] 23
##
## [[6]][[2]]
## [1] "team"
```

5) For the list given above, use and show R coding to confirm that “pear” is a character element.

```
str(myList[[3]])

## chr "pear"
```

6) For the list given above, use and show R coding to extract the first three elements of the list.

```
str(myList[1:3])

## List of 3
## $ : logi TRUE
## $ : num 12.3
## $ : chr "pear"
```

7) Use the \$ operator to extract the element “pear” from your list. Be sure to use and show required R code to produce the requested output.

```
myList <- list(TRUE,12.35,p="pear",48,c=3:8,list(23,"team"))
myList$p

## [1] "pear"
```

8) Use and show R code to write a function to solve the following quadratic equations by using the quadratic formula. (all equations have two real number solutions)

a) $x^2 - 3x - 28 = 0$

```
quadratic_equations <- function(a,b,c) {
formula1 <-(-b + sqrt(b^2 - 4 * a * c)) / (2 * a)
formula2 <-(-b - sqrt(b^2 - 4 * a * c)) / (2 * a)
return(c(formula1 ,formula2))
}
quadratic_equations(1, -3,-28)
```

```
## [1] 7 -4
```

b) $x^2 + x - 30 = 0$

```
quadratic_equations <- function(a,b,c) {
formula1 <-(-b + sqrt(b^2 - 4 * a * c)) / (2 * a)
formula2 <-(-b - sqrt(b^2 - 4 * a * c)) / (2 * a)
return(c(formula1 ,formula2))
}
quadratic_equations(1, 1,-30)
```

```
## [1] 5 -6
```

c) $3x^2 + 14x + 8 = 0$

```
quadratic_equations <- function(a,b,c) {
formula1 <-(-b + sqrt(b^2 - 4 * a * c)) / (2 * a)
formula2 <-(-b - sqrt(b^2 - 4 * a * c)) / (2 * a)
return(c(formula1 ,formula2))
```

```
}
quadratic_equations(3,14,8)
```

```
## [1] -0.6666667 -4.0000000
```

d) $2x^2 + 11x = 6$

```
quadratic_equations <- function(a,b,c) {
formula1 <-(-b + sqrt(b^2 - 4 * a * c)) / (2 * a)
formula2 <-(-b - sqrt(b^2 - 4 * a * c)) / (2 * a)
return(c(formula1 ,formula2))
}
quadratic_equations(2,11,-6)
```

```
## [1] 0.5 -6.0
```

9) In your book (towards the end of chapter 16) a special set of vectors are defined as Augmented Vectors. One such augmented vector is a Tibble. Use and show R code that will produce the Tibble shown below. Do not simply type or copy and paste. You must show and use R coding that will output the tibble.

```
tb <- tibble::tibble(x = 1:10, y = 10:1, z =y/2)
typeof(tb)
```

```
## [1] "list"
```

```
tb
```

```
## # A tibble: 10 x 3
##       x     y     z
##   <int> <int> <dbl>
## 1     1    10     5
## 2     2     9   4.5
## 3     3     8     4
## 4     4     7   3.5
## 5     5     6     3
## 6     6     5   2.5
## 7     7     4     2
## 8     8     3   1.5
## 9     9     2     1
## 10    10     1   0.5
```

10) In statistics, the Interquartile Range is the difference between Q3 and Q1. Now show and use map function coding to find the Interquartile Range for each column of the tibble from number 9.

```
tb %>%
  map_dbl(IQR)
```

```
##      x      y      z
## 4.50 4.50 2.25
```