Name: Francine Camiña
Course, Year and Section:
BSIT – 2A

<div align="center">Worksheet-1 in R</div>

Worksheet for R
   Programming
   Instructions:
- Use RStudio or the RStudio Cloud accomplish this worksheet. + Save the R script as
   RWorksheet_lastname#1.R. DONE
- Create your own *GitHub repository* and push the R script as well as this pdf worksheet to your own repo. DONE

Accomplish this worksheet by answering the questions being asked and writing the code manually.
Using functions:
   seq(), assign(), min(), max(), c(), sort(), sum(), filter()
1. Set up a vector named age, consisting of 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, 42, 53, 41, 51, 35, 24, 33, 41.

   a. How many data
  points? 34 data
  points

   b. Write the R code and its output.
```
>data.frame(
 age) age
1    34
2    28
3    22
4    36
5    27
6    18
7    52
8    39
9    42
10  29
11  35
12  31
13  27
14  22
15  37
16  34
17  19
18  20
19  57
20  49
21  50
22  37
23  46
24  25
```

```
25  17
26  37
27  42
```

```
28  53
29  41
30  51
31  35
32  24
33  33
34  41
```

2. Find the reciprocal of the
    values for age. Write the R
    code and its output.
    ```
    > rec_age <- 1/age
    > rec_age
     [1] 0.02941176 0.03571429 0.04545455 0.02777778 0.03703704 0.05555556
     0.01923077
     [8] 0.02564103 0.02380952 0.03448276 0.02857143 0.03225806 0.03703704
     0.04545455
    [15] 0.02702703 0.02941176 0.05263158 0.05000000 0.01754386 0.02040816
     0.02000000
    [22] 0.02702703 0.02173913 0.04000000 0.05882353 0.02702703 0.02380952
     0.01886792
    [29] 0.02439024 0.01960784 0.02857143 0.04166667 0.03030303 0.02439024
    ```

3. Assign also new age <-
    c(age, 0, age). What
    happen to the new_age?
    Contrast to the object "age", in "new_age" a number "0" Is added after the
    values of "age" and recurrent   the values of the "age" again.
    Resulting:
    ```
    [1] 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
    [27] 42 53 41 51 35 24 33 41 0 34 28 22 36 27 18 52 39 42 29 35 31 27 22 37
    34 19
    [53] 20 57 49 50 37 46 25 17 37 42 53 41 51 35 24 33 41
    ```

4. Sort the values for age.
    Write the R code and its output.
    ```
    > sort(age)
     [1] 17 18 19 20 22 22 24 25 27 27 28 29 31 33 34 34 35 35 36 37 37 37 39 41
     41 42
    [27] 42 46 49 50 51 52 53 57
    ```

5. Find the minimum and maximum
    value for age. Write the R code
    and its output.
    ```
    > min(ag
    e) [1]
    17
    ```

```
> max(ag
e) [1]
57
```

6. Set up a vector named data, consisting of 2.4, 2.8, 2.1, 2.5, 2.4, 2.2, 2.5, 2.3, 2.5, 2.3, 2.4, and 2.7.
   a. How many data points?
   `12 data points`

b. Write the R code and its output.
```
>data <- c(2.4, 2.8, 2.1, 2.5, 2.4, 2.2, 2.5, 2.3,
+    2.5, 2.3, 2.4, 2.7)
>data
 [1] 2.4 2.8 2.1 2.5 2.4 2.2 2.5 2.3 2.5 2.3 2.4 2.7
```

7. Generates a new vector for data where you double every value of the data.
| What happen to the data?
```
>data * 2
 [1] 4.8 5.6 4.2 5.0 4.8 4.4 5.0 4.6 5.0 4.6 4.8 5.4
```

8. Generate a sequence for the following scenario:

8.1 Integers from 1 to 100.
```
>seq(1:100)
 [1]  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19
[20]20 21 22 23 24  2 26 27 28 29  3 31 32  3 34  3 36 37  3
]              5           0        3     5        8
[39]39 40 41 42 43  4 45 46 47 48 49 50 51  5 53  5 55 56  5
]              4                    2     4        7
[58]58 59 60 61 62  6 64 65 66 67  6 69 70  7 72  7 74 75  7
]              3              8        1     3        6
[77]77 78 79 80 81  8 83 84 85 86  8 88 89  9 91  9 93 94  9
]              2              7        0     2        5
[96]96 97 98 99 10
]              0
```

8.2 Numbers from 20 to 60
```
>seq(20,60)
 [1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
43 44 45
[27] 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
```
*8.3 Mean of numbers from 20 to 60
```
>mean(20:
60) [1] 40
```
*8.4 Sum of numbers from 51 to 91
```
>sum(51:
91) [1]
2911
```
*8.5 Integers from 1 to 1,000
```
>seq(1:1000)
 [1] 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
[16 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
]
[31 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
]
[46 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
]
[61 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
]
[76 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
]
```

```
 [91]  91  92  93  94  95  96  97  98  99 100 101
102 103 104 105
[106] 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
[121] 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
[136] 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
[151] 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
[166] 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
[181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195
[196] 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210
[211] 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
[226] 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
```

```
[24 24 24 24 24 24 24 24 24 24 25 25 25 25 25 255
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[25 25 25 25 25 26 26 26 26 26 26 26 26 26 26 270
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[27 27 27 27 27 27 27 27 27 27 28 28 28 28 28 285
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[28 28 28 28 28 29 29 29 29 29 29 29 29 29 29 300
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[30 30 30 30 30 30 30 30 30 30 31 31 31 31 31 315
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[31 31 31 31 31 32 32 32 32 32 32 32 32 32 32 330
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[33 33 33 33 33 33 33 33 33 33 34 34 34 34 34 345
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[34 34 34 34 34 35 35 35 35 35 35 35 35 35 359 360
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8
[36 36 36 36 36 36 36 36 36 36 37 37 37 37 37 375
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[37 37 37 37 37 38 38 38 38 38 38 38 38 38 38 390
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[39 39 39 39 39 39 39 39 39 39 40 40 40 40 40 405
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[40 40 40 40 40 41 41 41 41 41 41 41 41 41 41 420
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[42 42 42 42 42 42 42 42 42 42 43 43 43 43 43 435
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[43 43 43 43 43 44 44 44 44 44 44 44 44 44 44 450
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[45 45 45 45 45 45 45 45 45 45 46 46 46 46 46 465
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[46 46 46 46 46 47 47 47 47 47 47 47 47 47 479 480
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8
[48 48 48 48 48 48 48 48 48 48 49 49 49 49 49 495
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[49 49 49 49 49 50 50 50 50 50 50 50 50 50 50 510
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[51 51 51 51 51 51 51 51 51 51 52 52 52 52 524 525
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3
[52 52 52 52 52 53 53 53 53 53 53 53 53 53 53 540
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[54 54 54 54 54 54 54 54 54 54 55 55 55 55 55 555
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[55 55 55 55 55 56 56 56 56 56 56 56 56 56 569 570
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8
[57 57 57 57 57 57 57 57 57 57 58 58 58 58 58 585
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3  4
[58 58 58 58 58 59 59 59 59 59 59 59 59 59 59 600
 6]  6  7  8  9  0  1  2  3  4  5  6  7  8  9
[60 60 60 60 60 60 60 60 60 60 61 61 61 61 614 615
 1]  1  2  3  4  5  6  7  8  9  0  1  2  3
[61 61 61 61 61 62 62 62 62 62 62 62 62 62 62 630
```

```
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[63 63  63  63  63  63  63  63  63  63 64  64  64  64  64 645
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[64 64  64  64  64 65  65  65  65  65  65  65  65  65 659 660
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8
[66 66  66  66  66  66  66  66  66  66 67  67  67  67  67 675
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[67 67  67  67  67 68  68  68  68  68  68  68  68  68  68 690
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[69 69  69  69  69  69  69  69  69  69 70  70  70  70 704 705
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3
[70 70  70  70  70 71  71  71  71  71  71  71  71  71  71 720
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[72 72  72  72  72  72  72  72  72  72 73  73  73  73  73 735
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[73 73  73  73  73 74  74  74  74  74  74  74  74  74  74 750
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[75 75  75  75  75  75  75  75  75  75 76  76  76  76  76 765
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[76 76  76  76  76 77  77  77  77  77  77  77  77  77  77 780
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[78 78  78  78  78  78  78  78  78  78 79  79  79  79  79 795
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[79 79  79  79  79 80  80  80  80  80  80  80  80  80  80 810
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[81 81  81  81  81  81  81  81  81  81 82  82  82  82  82 825
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[82 82  82  82  82 83  83  83  83  83  83  83  83  83  83 840
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[84 84  84  84  84  84  84  84  84  84 85  85  85  85  85 855
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[85 85  85  85  85 86  86  86  86  86  86  86  86  86  86 870
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[87 87  87  87  87  87  87  87  87  87 88  88  88  88  88 885
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[88 88  88  88  88 89  89  89  89  89  89  89  89  89 899 900
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8
[90 90  90  90  90  90  90  90  90  90 91  91  91  91  91 915
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3   4
[91 91  91  91  91 92  92  92  92  92  92  92  92  92  92 930
 6]  6   7   8   9   0   1   2   3   4   5   6   7   8   9
[93 93  93  93  93  93  93  93  93  93 94  94  94  94 944 945
 1]  1   2   3   4   5   6   7   8   9   0   1   2   3
```

```
[94  94   94    94   94   95    95   95   95    95 955 95    95    95    95    96
6]  6    7    8    9    0    1    2    3    4        6    7    8    9    0
[96  96   96    96   96   96    96   96   96    96 970 97    97    97    97    97
1]  1    2    3    4    5    6    7    8    9         1    2    3    4    5
[97  97   97    97   97   98    98   98   98    98 985 98    98    98    98    99
6]  6    7    8    9    0    1    2    3    4        6    7    8    9    0
[99  99   99    99   99   99    99   99   99    99 100
1]  1    2    3    4    5    6    7    8    9 0
```

a. How many data points from 8.1 to 8.4?_____

   8.1 = 100 data points
   8.2 = 41 data points
   8.3 = 1 data point
   8.4 = 1 data point
   Total of 143 data points

b. Write the R code and its output from 8.1 to 8.4.

   8.1 CODE =

```
> data.frame(1
  :100) X1.100
1    1
2    2
3    3
4    4
5    5
6    6
7    7
8    8
9    9
10   10
11   11
12   12
13   13
14   14
15   15
16   16
17   17
18   18
19   19
20   20
21   21
22   22
23   23
24   24
25   25
26   26
27   27
28   28
29   29
30   30
31   31
32   32
```

```
33    33
34    34
35    35
36    36
37    37
38    38
39    39
40    40
41    41
42    42
43    43
44    44
45    45
46    46
47    47
48    48
49    49
50    50
51    51
52    52
53    53
54    54
55    55
56    56
57    57
58    58
59    59
60    60
61    61
62    62
63    63
64    64
65    65
66    66
67    67
68    68
69    69
70    70
71    71
72    72
73    73
74    74
75    75
76    76
77    77
78    78
79    79
```

```
80   80
81   81
82   82
83   83
84   84
85   85
86   86
87   87
88   88
89   89
90   90
91   91
92   92
93   93
94   94
95   95
96   96
97   97
98   98
99   99
100 100
```

8.2 CODE =
```
> data.frame(2
  0:60) X20.60
1    20
2    21
3    22
4    23
5    24
6    25
7    26
8    27
9    28
10   29
11   30
12   31
13   32
14   33
15   34
16   35
17   36
18   37
19   38
20   39
21   40
22   41
```

```
23  42
24  43
25  44
26  45
27  46
28  47
29  48
30  49
31  50
32  51
33  52
34  53
35  54
36  55
37  56
38  57
39  58
40  59
41  60
```

8.3 CODE =
```
> data.frame(mean(2
 0:60)) mean.20.60.
 1     40
```

8.4 CODE =
```
> data.frame(sum(5
 1:91)) sum.51.91.
 1   2911
```

c. For 8.5 find only maximum data points until 10.
```
> max(1:
10) [1]
10
```

9. *Print a vector with the integers between 1 and 100 that are not divisible by 3, 5 and 7 using filter option. filter(function(i) { all(i %% c(3,5,7) != 0) }, seq(100)) Write the R code and its output.
```
> filter(function(i) { all(i %% c(3,5,7) != 0) }, seq(100)) #if
 "f" in the word "filter" is small letter then result will be
 error
 Error in attr(data, "tsp") <- c(start, end, frequency) :
  object is not a matrix
> Filter(function(i) { all(i %% c(3,5,7) != 0) }, seq(100))
 #if "F" in the word "Filter" is Capital then the result is
 not error
 [1]  1  2  4  8 11 13 16 17 19 22 23 26 29 31 32 34 37 38 41 43 44 46 47 52
 53 58
 [27] 59 61 62 64 67 68 71 73 74 76 79 82 83 86 88 89 92 94 97
```

10. Generate a sequence backwards of the

integers from 1 to 100. Write the R code and
its output.

```
> seq(100:1)
```

```
 [1]  1  2  3  4  5  6  7  8  9  10   11  12 13 14 15 16  17 18  19
[20 20 21 22 2  24 25 26 27 28  2 30 31  3 33  3 35 36  3 38
]           3              9     2     4     7
[39 39 40 41 4  43 44 45 46 47  4 49 50  5 52  5 54 55  5  5
]           2              8     1     3        6 7
[58 58 59 60 6  62 63 64 65 66  6 68 69  7 71  7 73 74  7  7
]           1              7     0     2        5 6
[77 77 78 79 8  81 82 83 84 85  8 87 88  8 90  9 92 93  9 95
]           0              6     9     1        4
[96 96 97 98 9  10
]           9   0
```

11. List all the natural numbers below 25 that are multiples of 3 or 5.

```
> sum((1 : 25)[((1 : 25)%%3 == 0) | ((1:25)%%5 == 0)])
[1] 168
```

Find the sum of these multiples.

a. How many data points from 10 to 11? 101 data points

b. Write the R code and its output from 10 and 11. No. 10 output =

```
> data.frame(1
  00:1) X100.1
1   100
2    99
3    98
4    97
5    96
6    95
7    94
8    93
9    92
10   91
11   90
12   89
13   88
14   87
15   86
16   85
17   84
18   83
19   82
20   81
21   80
22   79
23   78
24   77
25   76
26   75
27   74
28   73
29   72
```

14

30 71

```
31   70
32   69
33   68
34   67
35   66
36   65
37   64
38   63
39   62
40   61
41   60
42   59
43   58
44   57
45   56
46   55
47   54
48   53
49   52
50   51
51   50
52   49
53   48
54   47
55   46
56   45
57   44
58   43
59   42
60   41
61   40
62   39
63   38
64   37
65   36
66   35
67   34
68   33
69   32
70   31
71   30
72   29
73   28
74   27
75   26
76   25
77   24
```

```
78   23
79   22
80   21
81   20
82   19
83   18
84   17
85   16
86   15
87   14
88   13
89   12
90   11
91   10
92    9
93    8
94    7
95    6
96    5
97    4
98    3
99    2
100   1
```

```
No. 11 output =
> data.frame(sum((1 : 25)[((1 : 25)%%3 == 0)
 | ((1:25)%%5 == 0)]))
 sum..1.25....1.25...3....0......1.25...5...
 .0...
1                    168
```

12. Statements can be grouped together using braces '{' and '}'. A group of statements is sometimes called a **block**. Single statements are evaluated when a new line is typed at the end of the syntactically complete statement. Blocks are not evaluated until a new line is entered after the closing brace.

   Enter this
   statement:
```
{ x <- 0+ x + 5 + }
```
   Describe the output.
```
> { x <- 0+ x + 5 + }
Error: unexpected '}' in "{ x <- 0+ x + 5 + }"
Explanation: According to the error, the Closing Brace "}" is an error
with in the given statement.
```

13. *Set up a vector named score, consisting of 72, 86, 92, 63, 88, 89, 91, 92, 75, 75 and 77. To access individual elements of an atomic vector, one generally uses the x[i] construction. Find x[2] and x[3]. Write the R
   code and its output. `2     86`
   `3  92`

14. *Create a vector a = c(1,2,NA,4,NA,6,7).
    a. Change the NA to 999 using the codes print(a,na.print="-999").
    > a = c(1,2,NA,4,NA,6,7)

```
> print(a,na.print="-999")
[1] 1  2 -999       4 -999    6    7
```
  b. Write the R code and its output. Describe the output.
```
> a = c(1,2,NA,4,NA,6,7)
> print(a,na.print="-999")
[1] 1  2 -999       4 -999    6    7
```
EXPLANATION: The "NA" was replaced by "-999".

15. A special type of function calls can appear on the left hand side of the assignmentoperator as in > class(x)
    <- "foo".
     Follow the codes below:
```
       name = readline(prompt="Input your name: ") age =
       readline(prompt="Input your age: ") print(paste("My name is",name,
       "and I am",age ,"years old.")) print(R.version.string)
```
     What is the output of the above code?
```
     [1] "My name is Francine Camiña and I am 19 years old years old."
     > print(R.version.string)
```