name: v.ramya sri
regno: 192124039

day 2 lab exercise

IMPLEMENTATION OF VECTOR RECYCLING, APPLY FAMILY &amp; RECURSION

1. Demonstrate Vector Recycling in R.

```
> vec1=1:6
> vec2=1:2
>
> print(vec1+vec2)
[1] 2 4 4 6 6 8
>
> vec1=20:25
> vec2=4:6
> print(vec1+vec2)
[1] 24 26 28 27 29 31
>
```

2. Demonstrate the usage of apply function in R

```
> sample_matrix <- matrix(C<-(1:10),nrow=3, ncol=10)
>
> print( "sample matrix:")
[1] "sample matrix:"
> sample_matrix
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    4    7   10    3    6    9    2    5     8
[2,]    2    5    8    1    4    7   10    3    6     9
[3,]    3    6    9    2    5    8    1    4    7    10
> print("sum across rows:")
[1] "sum across rows:"
> apply( sample_matrix, 1, sum)
[1] 55 55 55
> print("mean across columns:")
[1] "mean across columns:"
> apply( sample_matrix, 2, mean)
 [1] 2.000000 5.000000 8.000000 4.333333 4.000000 7.000000 6.666667 3.000000 6.000000
9.000000
```

```
>
```

3. Demonstrate the usage of lapply function in R

```
> names <- c("priyank", "abhiraj","pawananjani",
+           "sudhanshu","devraj")
> print( "original data:")
[1] "original data:"
> names
[1] "priyank"     "abhiraj"     "pawananjani" "sudhanshu"   "devraj"
> print("data after lapply():")
[1] "data after lapply():"
> lapply(names, toupper)
[[1]]
[1] "PRIYANK"

[[2]]
[1] "ABHIRAJ"

[[3]]
[1] "PAWANANJANI"

[[4]]
[1] "SUDHANSHU"

[[5]]
[1] "DEVRAJ"
```

4. Demonstrate the usage of sapply function in R

```
 sample_data<- data.frame( x=c(1,2,3,4,5,6),
+                        y=c(3,2,4,2,34,5))
> print( "original data:")
[1] "original data:"
> sample_data
  x  y
1 1  3
2 2  2
```

```
3 3  4
4 4  2
5 5 34
6 6  5
> print("data after sapply():")
[1] "data after sapply():"
> sapply(sample_data, max)
 x  y 
 6 34 
> 
> 
```

5. Demonstrate the usage of tapply function in R

```
data(iris)
> tapply(iris$Sepal.Width, iris$Species, median)
    setosa versicolor  virginica 
       3.4        2.8        3.0 
```

6. Demonstrate the usage of mapply function in R

```
> vec1 <- c(1, 2, 3, 4)
> vec2 <- c(2, 4, 6, 8)
> vec3 <- c(3, 6, 9, 12)
> mapply(function(val1, val2, val3) val1*val2*val3, vec1, vec2, vec3)
[1]   6  48 162 384
```

7. Sum of Natural Numbers using Recursion

```
 sum<-function(n){
+     if (n<=1){
+         return(n)
+         }else{
+         return(n+sum(n-1))
+     }
+ }
> sum(7)
[1] 28
> 
```

8. Write a program to generate Fibonacci sequence using Recursion in R

```
 Fibonacci <- numeric(10)
> Fibonacci[1] <- Fibonacci[2] <- 1
> for (i in 3:10) Fibonacci[i] <- Fibonacci[i - 2] + Fibonacci[i - 1]
> print("First 10 Fibonacci numbers:")
[1] "First 10 Fibonacci numbers:"
> print(Fibonacci)
 [1]  1  1  2  3  5  8 13 21 34 55
```

9. Write a program to find factorial of a number in R using recursion.

```
 recur_fact <- function(n) {
+ if(n <= 1) {
+ return(1)
+ } else {
+ return(n * recur_fact(n-1))
+ }
+ }
> recur_fact(8)
[1] 40320
> num<-8
> print(factorial(num))
[1] 40320
> fact <- 1
> if (num < 0) {
+
+  print("Factorial for negative numbers not allowed!")
+ } else if (num == 0) {
+  print("The factorial of 0 is 1")
+ } else {
+  for(i in 1:num){
+  fact=fact*i
+  }
+  print(fact)
+ }
[1] 40320
```

CREATION AND MANIPULATION OF DATAFRAMES IN R

Exercise 1

Consider two vectors: x=seq(1,43,along.with=Id)
y=seq(-20,0,along.with=Id)
Create a data frame ‘df’ as shown below.

```
&gt;df
Id Letter x y
1 1 a 1.000000 -20.000000
2 1 b 4.818182 -18.181818
3 1 c 8.636364 -16.363636
4 2 a 12.454545 -14.545455
5 2 b 16.272727 -12.727273
6 2 c 20.090909 -10.909091
7 3 a 23.909091 -9.090909
8 3 b 27.727273 -7.272727
9 3 c 31.545455 -5.454545

10 4 a 35.363636 -3.636364
11 4 b 39.181818 -1.818182
12 4 c 43.000000 0.000000
```

prgram:

```
> Id <- rep(1:4, each = 3)
> x=seq(1,43,along.with=Id)
> y=seq(-20,0,along.with=Id)
> Letter=rep(letters[1:3],4)
>
> df <- data.frame(Id,Letter,x,y)
> df
```

out put::

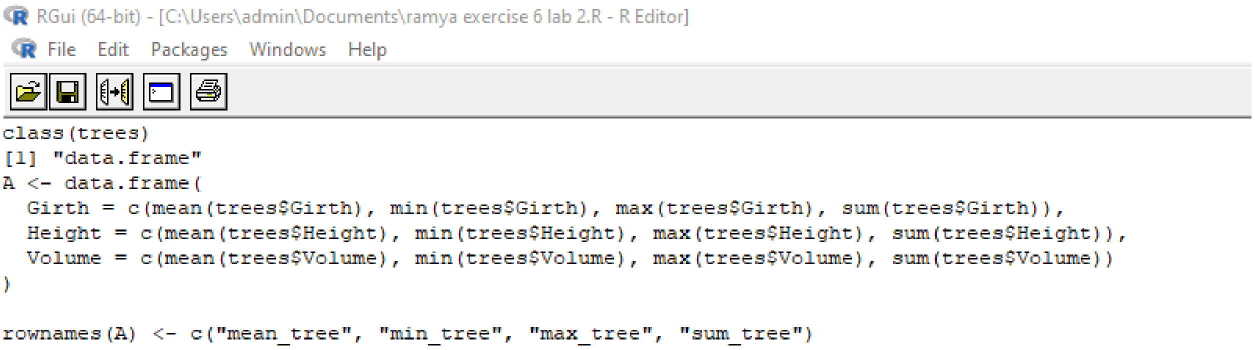
```
   Id Letter         x          y
1   1      a  1.000000 -20.000000
2   1      b  4.818182 -18.181818
3   1      c  8.636364 -16.363636
4   2      a 12.454545 -14.545455
5   2      b 16.272727 -12.727273
6   2      c 20.090909 -10.909091
```

```
7   3      a 23.909091  -9.090909
8   3      b 27.727273  -7.272727
9   3      c 31.545455  -5.454545
10  4      a 35.363636  -3.636364
11  4      b 39.181818  -1.818182
12  4      c 43.000000   0.000000
>
```

Exercise 6
For this exercise, we’ll use the (built-in) dataset trees.
a) Make sure the object is a data frame, if not change it to a data frame.
b) Create a new data frame A:
&gt;A
Girth Height Volume
mean_tree 13.24839 76 30.17097
min_tree 8.30000 63 10.20000
max_tree 20.60000 87 77.00000
sum_tree 410.70000 2356 935.30000

```
class(trees)
[1] "data.frame"
A <- data.frame(
  Girth = c(mean(trees$Girth), min(trees$Girth), max(trees$Girth), sum(trees$Girth)),
  Height = c(mean(trees$Height), min(trees$Height), max(trees$Height), sum(trees$Height)),
  Volume = c(mean(trees$Volume), min(trees$Volume), max(trees$Volume),
sum(trees$Volume))
)

rownames(A) <- c("mean_tree", "min_tree", "max_tree", "sum_tree")
```

```
OUTPUT:
Girth   Height    Volume
mean_tree 13.24839 76.00000  30.17097
min_tree   8.30000 63.00000  10.20000
max_tree  20.60000 87.00000  77.00000
sum_tree 410.70000 2356.00000 935.30000
```


RGui (64-bit) - [C:\Users\admin\Documents\ramya exercise 6 lab 2.R - R Editor]
File
Edit
Packages
Windows
Help

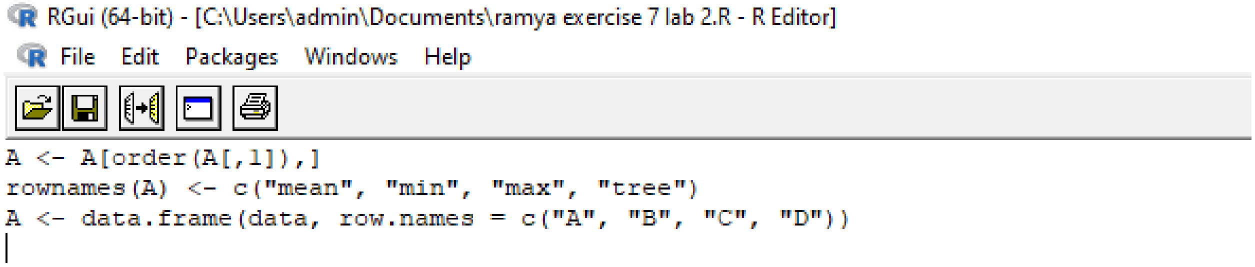
```
class(trees)
[1] "data.frame"
A <- data.frame(
  Girth = c(mean(trees$Girth), min(trees$Girth), max(trees$Girth), sum(trees$Girth)),
  Height = c(mean(trees$Height), min(trees$Height), max(trees$Height), sum(trees$Height)),
  Volume = c(mean(trees$Volume), min(trees$Volume), max(trees$Volume), sum(trees$Volume))
)

rownames(A) <- c("mean_tree", "min_tree", "max_tree", "sum_tree")
```

Exercise 7

Consider the data frame A:

1)Order the entire data frame by the first column.

2)Rename the row names as follows: mean, min, max, tree


RGui (64-bit) - [C:\Users\admin\Documents\ramya exercise 7 lab 2.R - R Editor]
File
Edit
Packages
Windows
Help

```
A <- A[order(A[,1]),]
rownames(A) <- c("mean", "min", "max", "tree")
A <- data.frame(data, row.names = c("A", "B", "C", "D"))
```

Exercise 8

Create an empty data frame with column types:

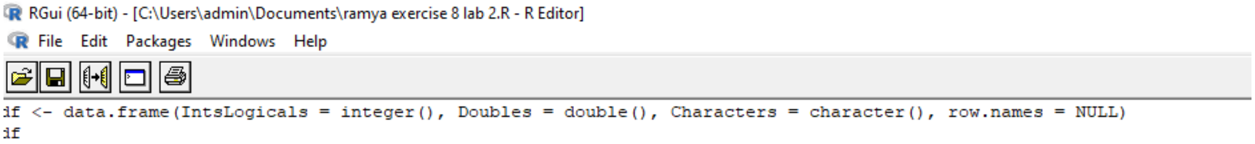
&gt;df

IntsLogicals Doubles Characters

(or 0-length row.names)

> df <- data.frame(IntsLogicals = integer(), Doubles = double(), Characters = character(), row.names = NULL)
> df

```
[1] IntsLogicals Doubles      Characters
<0 rows> (or 0-length row.names)
```


RGui (64-bit) - [C:\Users\admin\Documents\ramya exercise 8 lab 2.R - R Editor]
File
Edit
Packages
Windows
Help

```
df <- data.frame(IntsLogicals = integer(), Doubles = double(), Characters = character(), row.names = NULL)
df
```

Exercise 9

Create a data frame XY

X=c(1,2,3,1,4,5,2)

Y=c(0,3,2,0,5,9,3)

&gt; XY

X Y

1 1 0

2 2 3

3 3 2

4 1 0

5 4 5

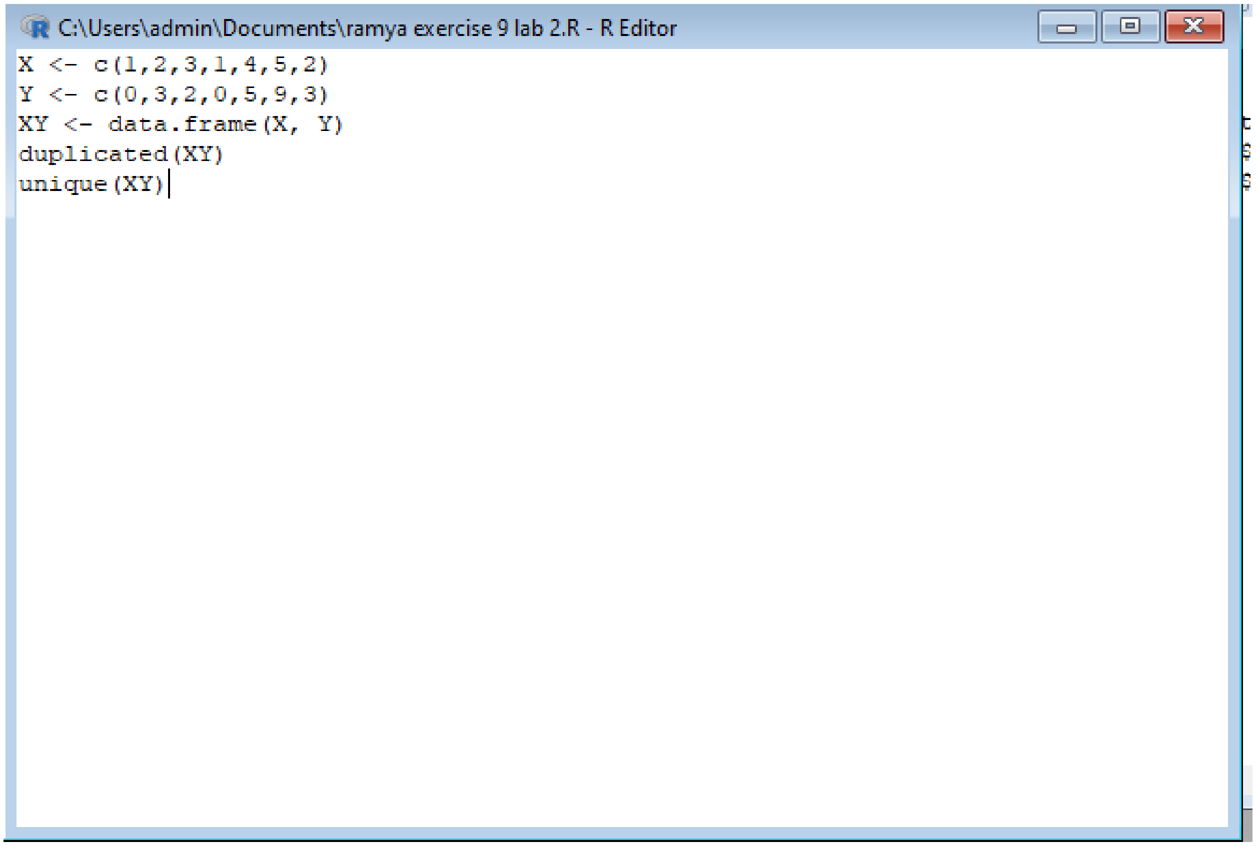
6 5 9

7 2 3

1) look at duplicated elements using a provided R function.

2) keep only the unique lines on XY using a provided R function.

```
> X <- c(1,2,3,1,4,5,2)
> Y <- c(0,3,2,0,5,9,3)
> XY <- data.frame(X, Y)
> duplicated(XY)
[1] FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
> unique(XY)
  X Y
1 1 0
2 2 3
3 3 2
5 4 5
6 5 9
>
```


C:\Users\admin\Documents\ramya exercise 9 lab 2.R - R Editor

```
X <- c(1,2,3,1,4,5,2)
Y <- c(0,3,2,0,5,9,3)
XY <- data.frame(X, Y)
duplicated(XY)
unique(XY)
```

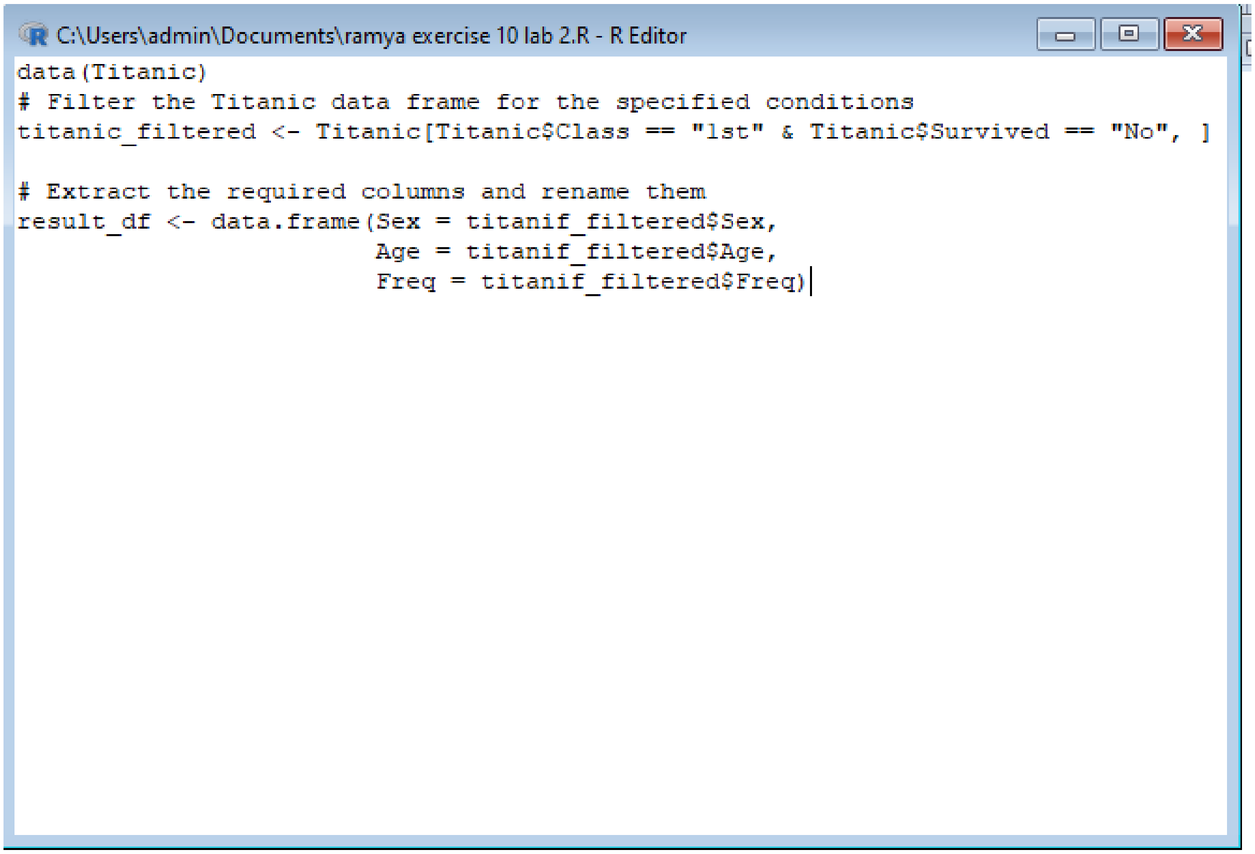
Exercise 10
Use the (built-in) dataset Titanic.
a) Make sure the object is a data frame, if not change it to a data frame.
b) Define a data frame with value 1st in Class variable, and value NO in Survived variable and variables Sex, Age and Freq.

```
data(Titanic)
# Filter the Titanic data frame for the specified conditions
titanic_filtered <- Titanic[Titanic$Class == "1st" & Titanic$Survived == "No", ]

# Extract the required columns and rename them
result_df <- data.frame(Sex = titanif_filtered$Sex,
                        Age = titanif_filtered$Age,
                        Freq = titanif_filtered$Freq)
```

Out put:

```
   Sex    Age    Freq
```

```
1    Male Child    0
5  Female Child    0
9    Male Adult  118
13 Female Adult    4
```


C:\Users\admin\Documents\ramya exercise 10 lab 2.R - R Editor

```
data(Titanic)
# Filter the Titanic data frame for the specified conditions
titanic_filtered <- Titanic[Titanic$Class == "1st" & Titanic$Survived == "No", ]

# Extract the required columns and rename them
result_df <- data.frame(Sex = titanif_filtered$Sex,
                        Age = titanif_filtered$Age,
                        Freq = titanif_filtered$Freq)
```

Exercise 11 a)
Create the following dataframes to merge:
buildings&lt;- data.frame(location=c(1, 2, 3), name=c(&quot;building1&quot;, &quot;building2&quot;,&quot;building3&quot;))
data &lt;- data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))

The dataframes, buildingsand datahave a common key variable called, “location”. Use the merge() function to merge the two dataframes by “location”, into a new dataframe,“buildingStats”.

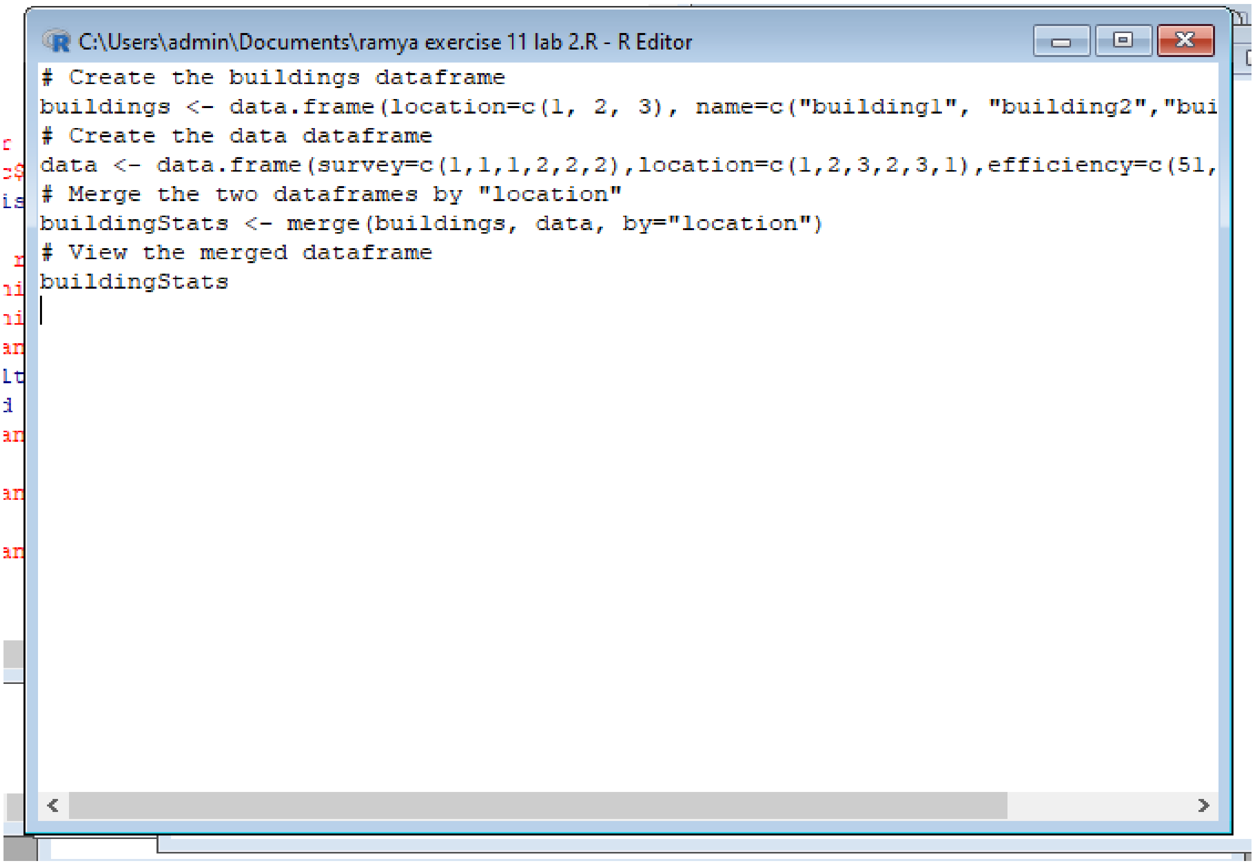
INPUT:

```
# Create the buildings dataframe
buildings <- data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))
```

```
# Create the data dataframe
data <- data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,64,70,7,80,58))
# Merge the two dataframes by "location"
buildingStats <- merge(buildings, data, by="location")
# View the merged dataframe
buildingStats
```

OUTPUT:

```
location   name    survey efficiency
1       1 building1     1       51
2       2 building2     1       64
3       2 building2     2        7
4       3 building3     1       70
5       3 building3     2       80
6       1 building1     2       58
```



C:\Users\admin\Documents\ramya exercise 11 lab 2.R - R Editor

```
# Create the buildings dataframe
buildings <- data.frame(location=c(1, 2, 3), name=c("building1", "building2","bui
# Create the data dataframe
data <- data.frame(survey=c(1,1,1,2,2,2),location=c(1,2,3,2,3,1),efficiency=c(51,
# Merge the two dataframes by "location"
buildingStats <- merge(buildings, data, by="location")
# View the merged dataframe
buildingStats
```

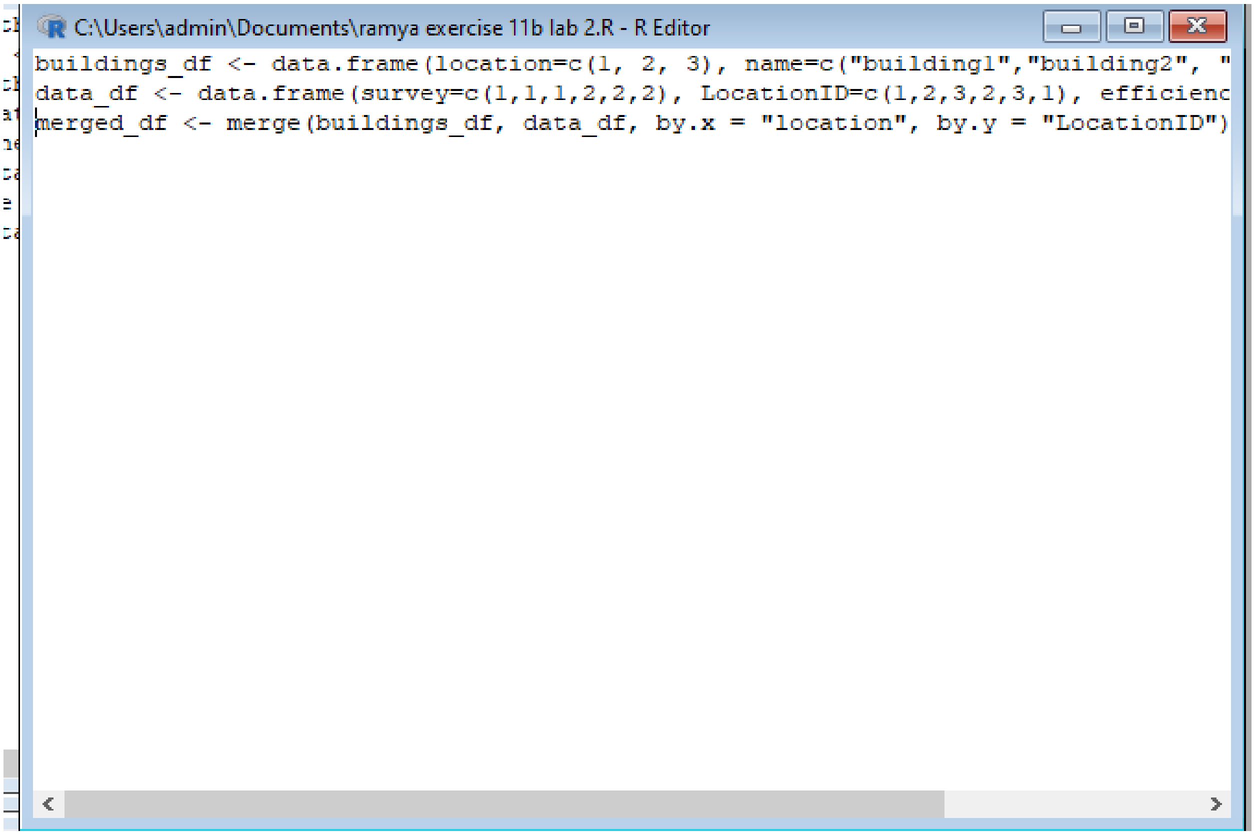
Exercise 11 b)

Give the dataframes different key variable names:

buildings&lt;- data.frame(location=c(1, 2, 3),
name=c(&quot;building1&quot;,&quot;building2&quot;, &quot;building3&quot;))
data &lt;- data.frame(survey=c(1,1,1,2,2,2), LocationID=c(1,2,3,2,3,1),

efficiency=c(51,64,70,71,80,58))
The dataframes, buildings and data have corresponding variables called, location, and LocationID. Use the merge() function to merge the columns of the two dataframes by the corresponding variables.

buildings_df <- data.frame(location=c(1, 2, 3), name=c("building1","building2", "building3"))
data_df <- data.frame(survey=c(1,1,1,2,2,2), LocationID=c(1,2,3,2,3,1),
efficiency=c(51,64,70,71,80,58))
merged_df <- merge(buildings_df, data_df, by.x = "location", by.y = "LocationID")

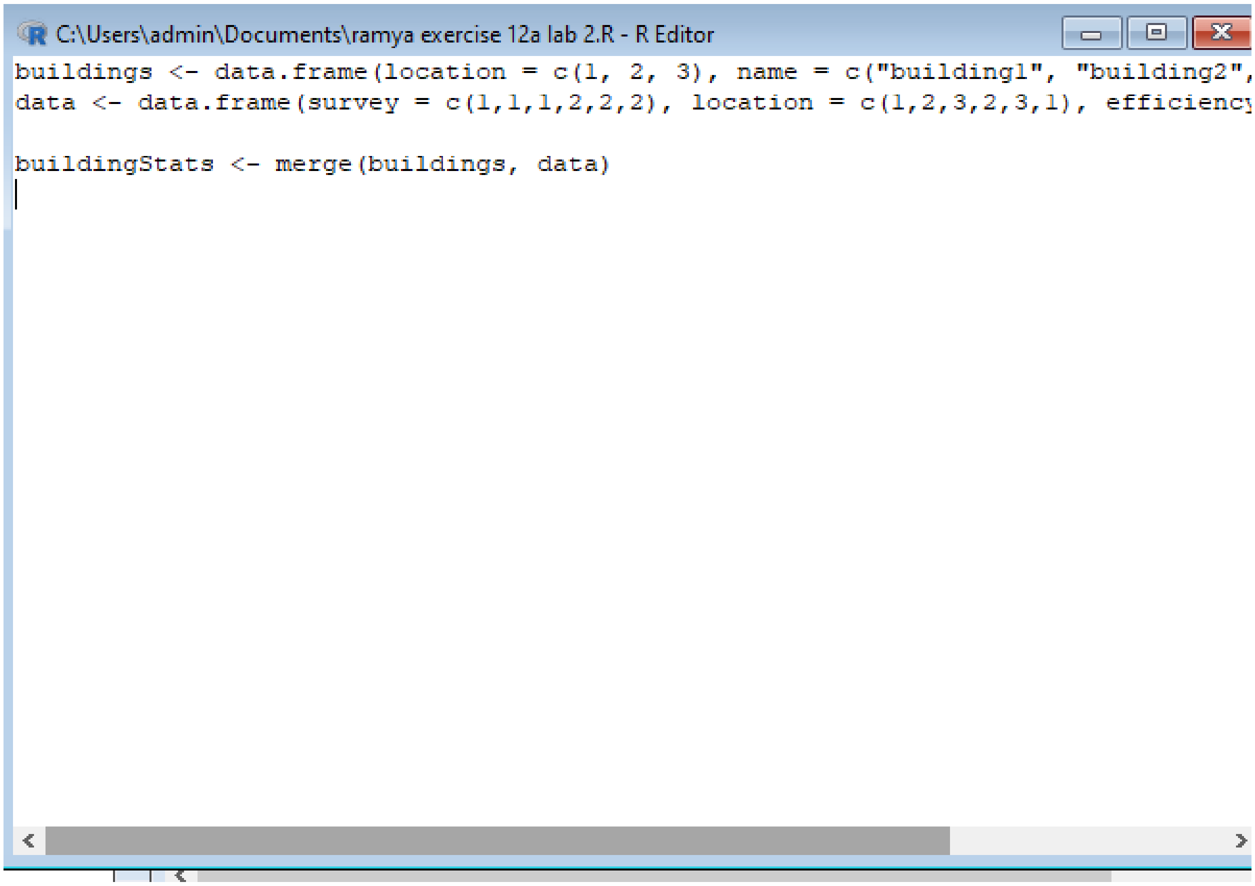

C:\Users\admin\Documents\ramya exercise 11b lab 2.R - R Editor

```
buildings_df <- data.frame(location=c(1, 2, 3), name=c("building1","building2", "
data_df <- data.frame(survey=c(1,1,1,2,2,2), LocationID=c(1,2,3,2,3,1), efficienc
merged_df <- merge(buildings_df, data_df, by.x = "location", by.y = "LocationID")
```

Exercise 12a)InnerJoin:
The R merge() function automatically joins the frames by common variable names. In that case, demonstrate how you would perform the merge in Exercise 11a without specifying the
key variable.

buildings <- data.frame(location = c(1, 2, 3), name = c("building1", "building2", "building3"))

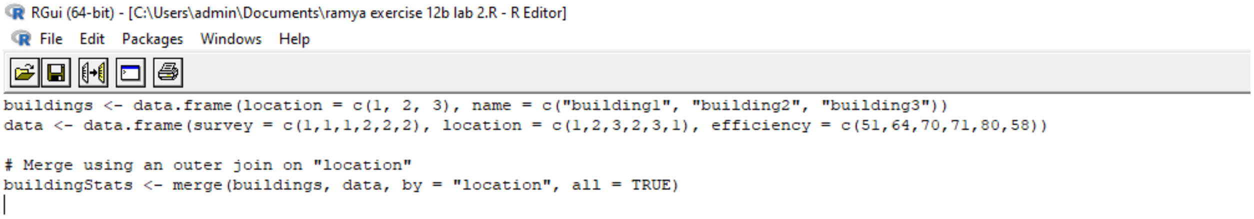
```
data <- data.frame(survey = c(1,1,1,2,2,2), location = c(1,2,3,2,3,1), efficiency =
c(51,64,70,71,80,58))

buildingStats <- merge(buildings, data)
```


C:\Users\admin\Documents\ramya exercise 12a lab 2.R - R Editor

```
buildings <- data.frame(location = c(1, 2, 3), name = c("building1", "building2",
data <- data.frame(survey = c(1,1,1,2,2,2), location = c(1,2,3,2,3,1), efficiency

buildingStats <- merge(buildings, data)
```

Exercise 12b)OuterJoin:
Merge the two dataframes from Exercise 11a. Use the “all=” parameter in the merge() function to return all records from both tables. Also, merge with the key variable, “location”.

```
buildings <- data.frame(location = c(1, 2, 3), name = c("building1", "building2", "building3"))
data <- data.frame(survey = c(1,1,1,2,2,2), location = c(1,2,3,2,3,1), efficiency =
c(51,64,70,71,80,58))

# Merge using an outer join on "location"
buildingStats <- merge(buildings, data, by = "location", all = TRUE)
```



RGui (64-bit) - [C:\Users\admin\Documents\ramya exercise 12b lab 2.R - R Editor]
File
Edit
Packages
Windows
Help

```
buildings <- data.frame(location = c(1, 2, 3), name = c("building1", "building2", "building3"))
data <- data.frame(survey = c(1,1,1,2,2,2), location = c(1,2,3,2,3,1), efficiency = c(51,64,70,71,80,58))

# Merge using an outer join on "location"
buildingStats <- merge(buildings, data, by = "location", all = TRUE)
```

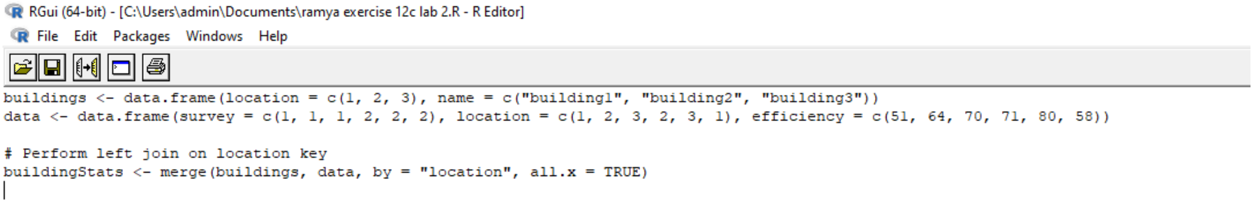
Exercise 12c)Left Join:

Merge the two dataframes from Exercise 11a, and return all rows from the left table. Specify

the matching key from Exercise 11a.

buildings <- data.frame(location = c(1, 2, 3), name = c("building1", "building2", "building3"))
data <- data.frame(survey = c(1, 1, 1, 2, 2, 2), location = c(1, 2, 3, 2, 3, 1), efficiency = c(51, 64, 70, 71, 80, 58))

Perform left join on location key
buildingStats <- merge(buildings, data, by = "location", all.x = TRUE)



RGui (64-bit) - [C:\Users\admin\Documents\ramya exercise 12c lab 2.R - R Editor]
File
Edit
Packages
Windows
Help

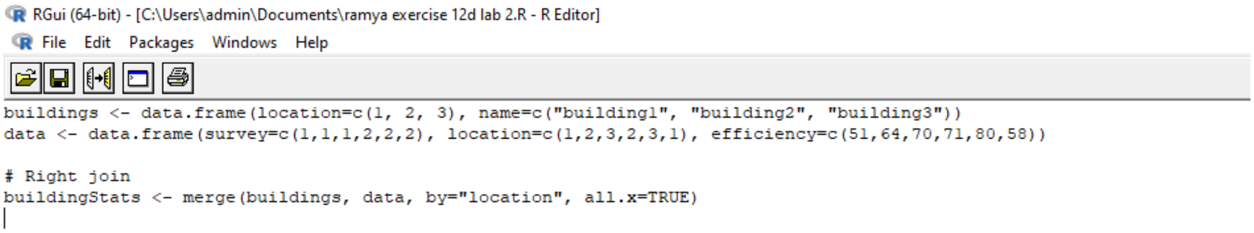
```
buildings <- data.frame(location = c(1, 2, 3), name = c("building1", "building2", "building3"))
data <- data.frame(survey = c(1, 1, 1, 2, 2, 2), location = c(1, 2, 3, 2, 3, 1), efficiency = c(51, 64, 70, 71, 80, 58))

# Perform left join on location key
buildingStats <- merge(buildings, data, by = "location", all.x = TRUE)
```

Exercise 12d)Right Join:

Merge the two dataframes from Exercise 11a, and return all rows from the right table. Use the matching key from Exercise 11a to return matching rows from the left table.

buildings <- data.frame(location=c(1, 2, 3), name=c("building1", "building2", "building3"))

```
data <- data.frame(survey=c(1,1,1,2,2,2), location=c(1,2,3,2,3,1),
efficiency=c(51,64,70,71,80,58))

# Right join
buildingStats <- merge(buildings, data, by="location", all.x=TRUE)
```



RGui (64-bit) - [C:\Users\admin\Documents\ramya exercise 12d lab 2.R - R Editor]
File
Edit
Packages
Windows
Help

```
buildings <- data.frame(location=c(1, 2, 3), name=c("building1", "building2", "building3"))
data <- data.frame(survey=c(1,1,1,2,2,2), location=c(1,2,3,2,3,1), efficiency=c(51,64,70,71,80,58))

# Right join
buildingStats <- merge(buildings, data, by="location", all.x=TRUE)
```