DAY 2 PROGRAMS

1.Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors. Print the second row of the second matrix of the array and the element in the 3rd row and 3rd column of the 1st matrix.

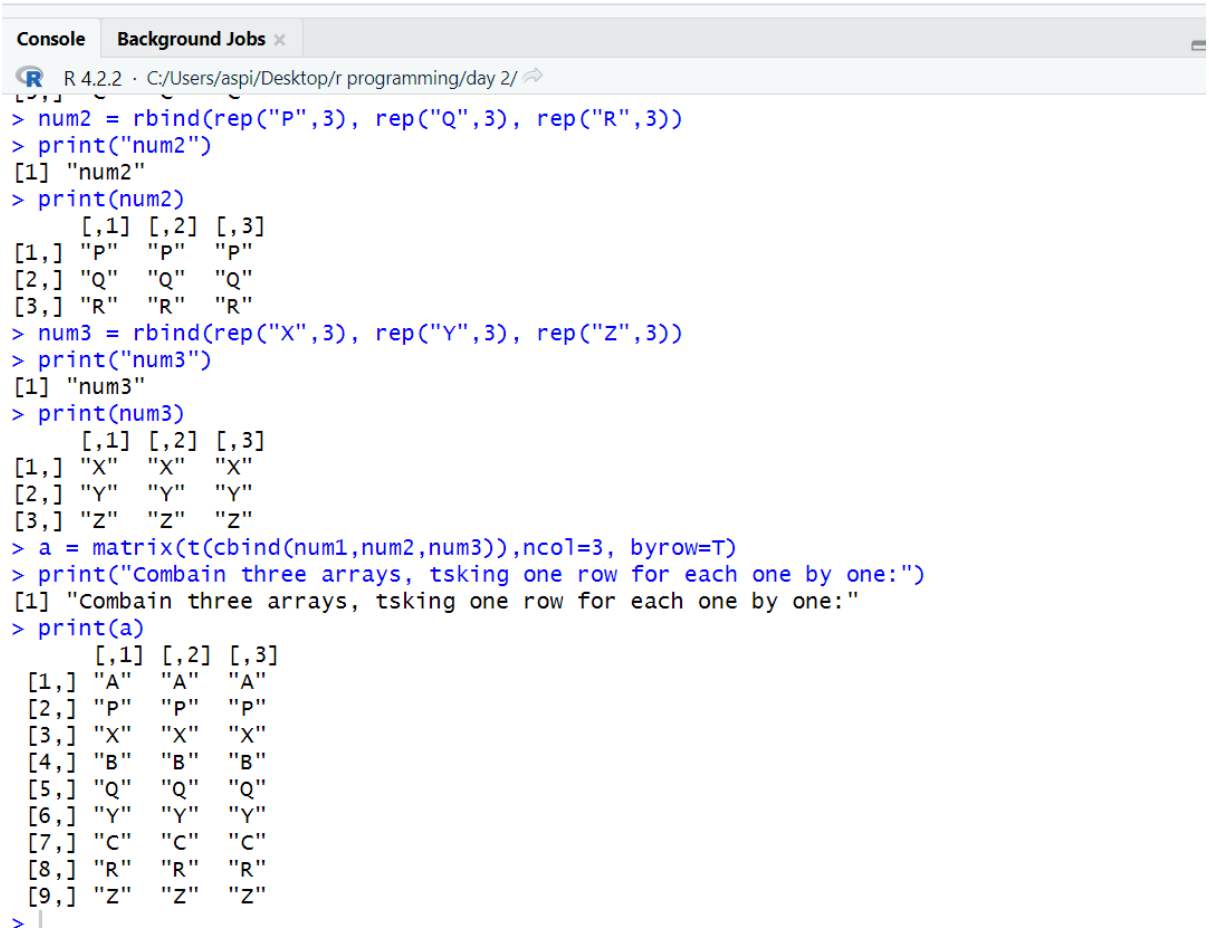
```
print("Two vectors of different lengths : ")
v1 = c(1,3,4,5)
v2 = c(10,11,12,13,14,15)
print(v1)
print(v2)
result = array(c(v1,v2),dim = c(3,3,2))
print("New array :")
print(result)
print("The second row of the second matrix of the array:")
print(result[2,,2])
print("The element in the 3rd row and 3rd colum of the 1st matrix:")
print(result[3,3,1])
```

```
R 4.2.2 · C:/Users/aspi/Desktop/r programming/day 2/
> print("Two vectors of different lengths : ")
[1] "Two vectors of different lengths : "
> v1 = c(1,3,4,5)
> v2 = c(10,11,12,13,14,15)
> print(v1)
[1] 1 3 4 5
> print(v2)
[1] 10 11 12 13 14 15
> result = array(c(v1,v2),dim = c(3,3,2))
> print("New array :")
[1] "New array :"
> print(result)
, , 1

     [,1] [,2] [,3]
[1,]    1    5   12
[2,]    3   10   13
[3,]    4   11   14

, , 2

     [,1] [,2] [,3]
[1,]   15    4   11
[2,]    1    5   12
[3,]    3   10   13

> print("The second row of the second matrix of the array:")
[1] "The second row of the second matrix of the array:"
> print(result[2,,2])
[1]  1  5 12
> print("The element in the 3rd row and 3rd colum of the 1st matrix:")
```

2. Write a R program to combine three arrays so that the first row of the first array is followed by the first row of the second array and then first row of the third array.

```
num1 = rbind(rep("A",3), rep("B",3), rep("C",3))
print("num1")
print(num1)
num2 = rbind(rep("P",3), rep("Q",3), rep("R",3))
print("num2")
print(num2)
num3 = rbind(rep("X",3), rep("Y",3), rep("Z",3))
print("num3")
print(num3)
a = matrix(t(cbind(num1,num2,num3)),ncol=3, byrow=T)
print("Combain three arrays, tsking one row for each one by one:")
```

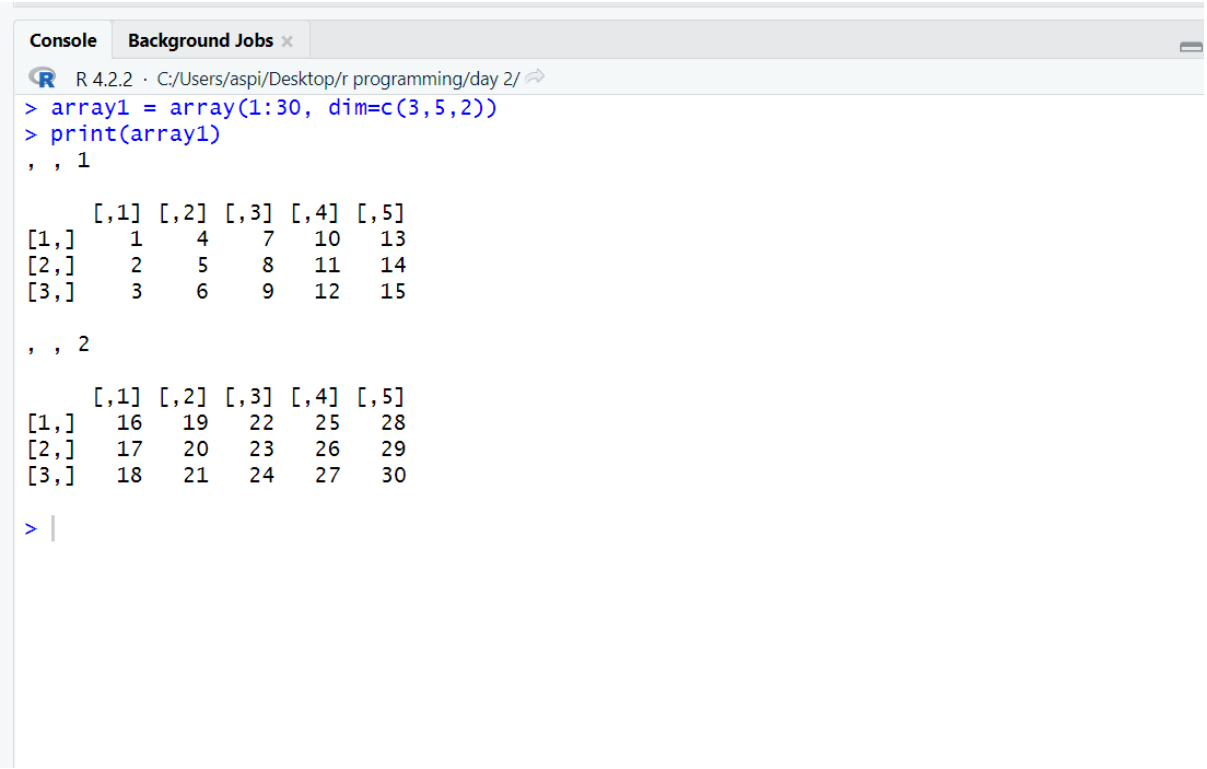
print(a)



3. Write a R program to create an array using four given columns, three given rows, and two given tables and display the content of the array.

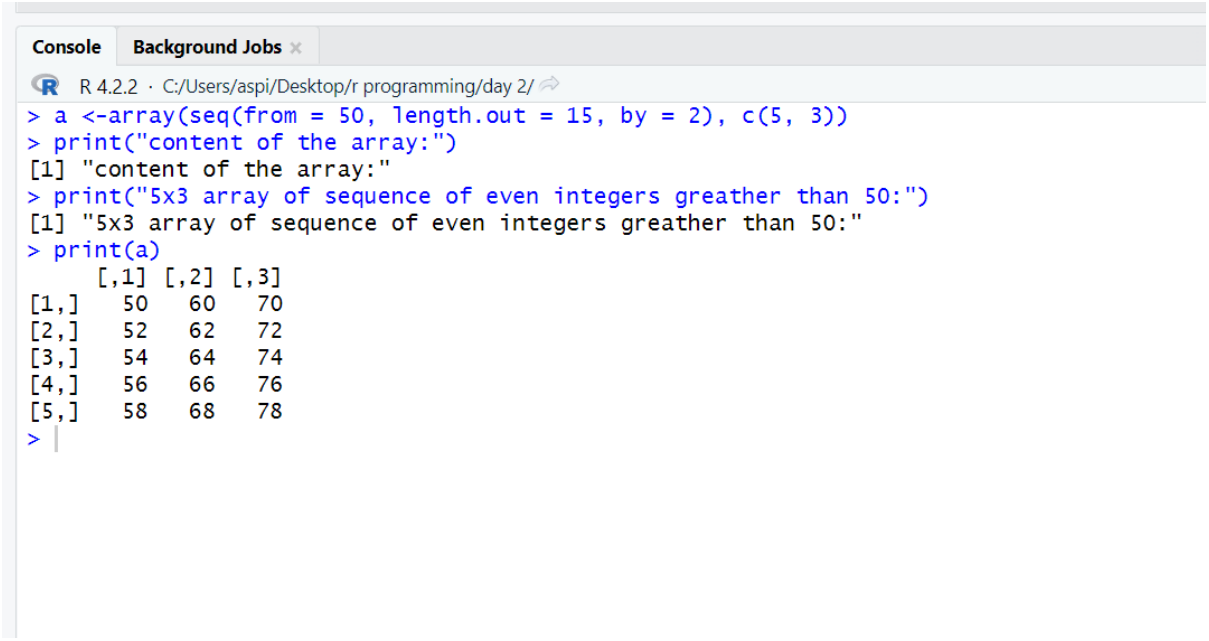
array1 = array(1:30, dim=c(3,5,2))

print(array1)



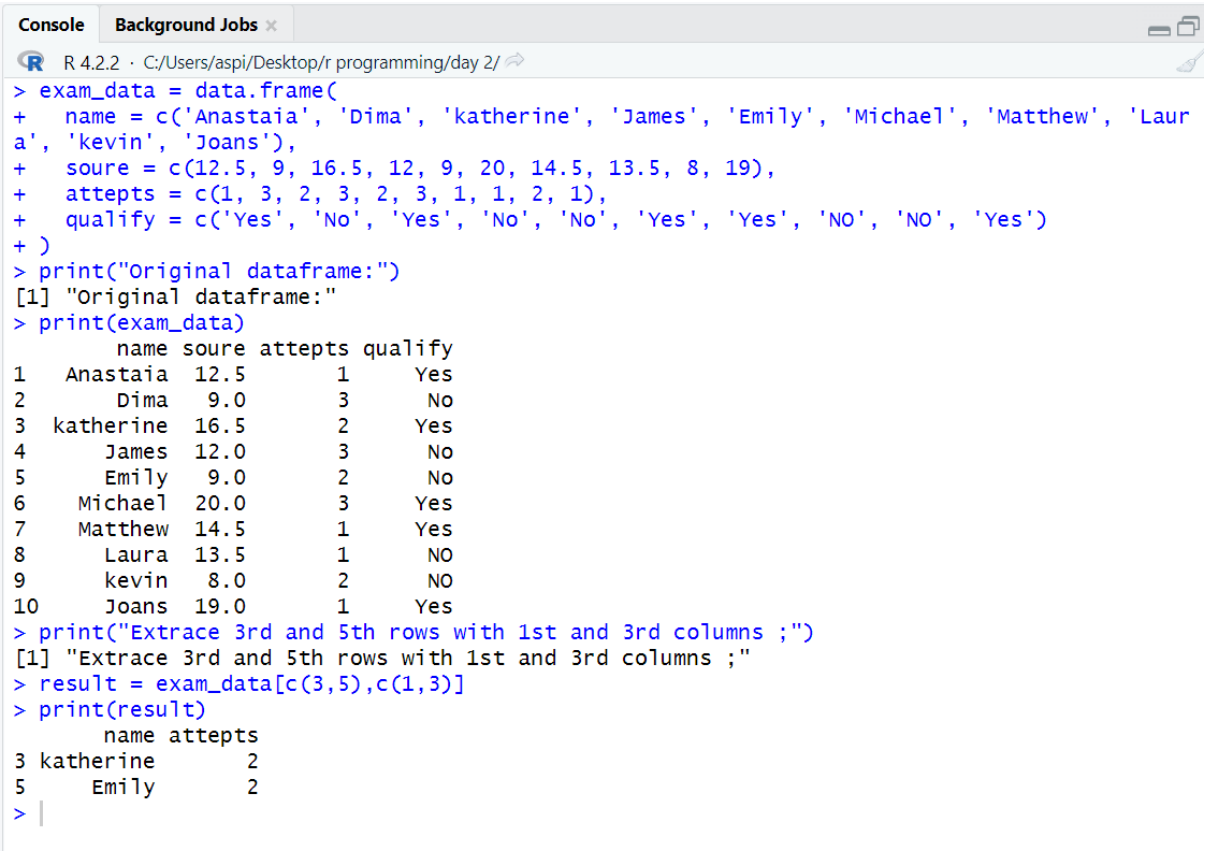
4. Write a R program to create a two-dimensional 5x3 array of sequence of even integers greater than 50.

```
a <-array(seq(from = 50, length.out = 15, by = 2), c(5, 3))
print("content of the array:")
print("5x3 array of sequence of even integers greather than 50:")
print(a)
```



5. Write a R program to extract 3 rd and 5 th rows with 1 st and 3 rd columns from a given data frame

```
exam_data = data.frame(
  name = c('Anastaia', 'Dima', 'katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'kevin', 'Joans'),
  soure = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attepts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'NO', 'NO', 'Yes')
)
print("Original dataframe:")
print(exam_data)
print("Extrace 3rd and 5th rows with 1st and 3rd columns ;")
result = exam_data[c(3,5),c(1,3)]
print(result)
```



6. Write a R program to add a new column named country in a given data frame

Country<-c("USA","USA","USA","USA","UK","USA","USA","India","USA","USA")

exam_data = data.frame(

name = c('Anastaia', 'Dima', 'katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'kevin', 'Joans'),

soure = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),

attepts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),

qualify = c('Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'NO', 'NO', 'Yes')

)

print("Original dataframe:")

print(exam_data)

print("New data ftame after adding the 'country' column : ")

exam_data$country = c('USA', 'USA', 'USA', 'USA', 'UK', 'USA', 'USA', 'INDIA', 'USA', 'USA')

print(exam_data)

```
> print("Original dataframe:")
[1] "Original dataframe:"
> print(exam_data)
        name soure attepts qualify
1   Anastaia  12.5       1     Yes
2       Dima   9.0       3      No
3  katherine  16.5       2     Yes
4      James  12.0       3      No
5      Emily   9.0       2      No
6    Michael  20.0       3     Yes
7    Matthew  14.5       1     Yes
8      Laura  13.5       1      NO
9      kevin   8.0       2      NO
10     Joans  19.0       1     Yes
> print("New data ftame after adding the 'country' column : ")
[1] "New data ftame after adding the 'country' column : "
> exam_data$country = c('USA', 'USA', 'USA', 'USA', 'UK', 'USA', 'USA', 'INDIA', 'USA',
SA')
> print(exam_data)
        name soure attepts qualify country
1   Anastaia  12.5       1     Yes     USA
2       Dima   9.0       3      No     USA
3  katherine  16.5       2     Yes     USA
4      James  12.0       3      No     USA
5      Emily   9.0       2      No      UK
6    Michael  20.0       3     Yes     USA
7    Matthew  14.5       1     Yes     USA
8      Laura  13.5       1      NO   INDIA
9      kevin   8.0       2      NO     USA
10     Joans  19.0       1     Yes     USA
>
```

7. Write a R program to add new row(s) to an existing data frame

new_exam_data = data.frame(name = c('Robert', 'Sophia'),score =

c(10.5, 9), attempts = c(1, 3),qualify = c('yes', 'no'))

exam_data = data.frame(

 name = c('Anastaia', 'Dima', 'katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'kevin', 'Joans'),

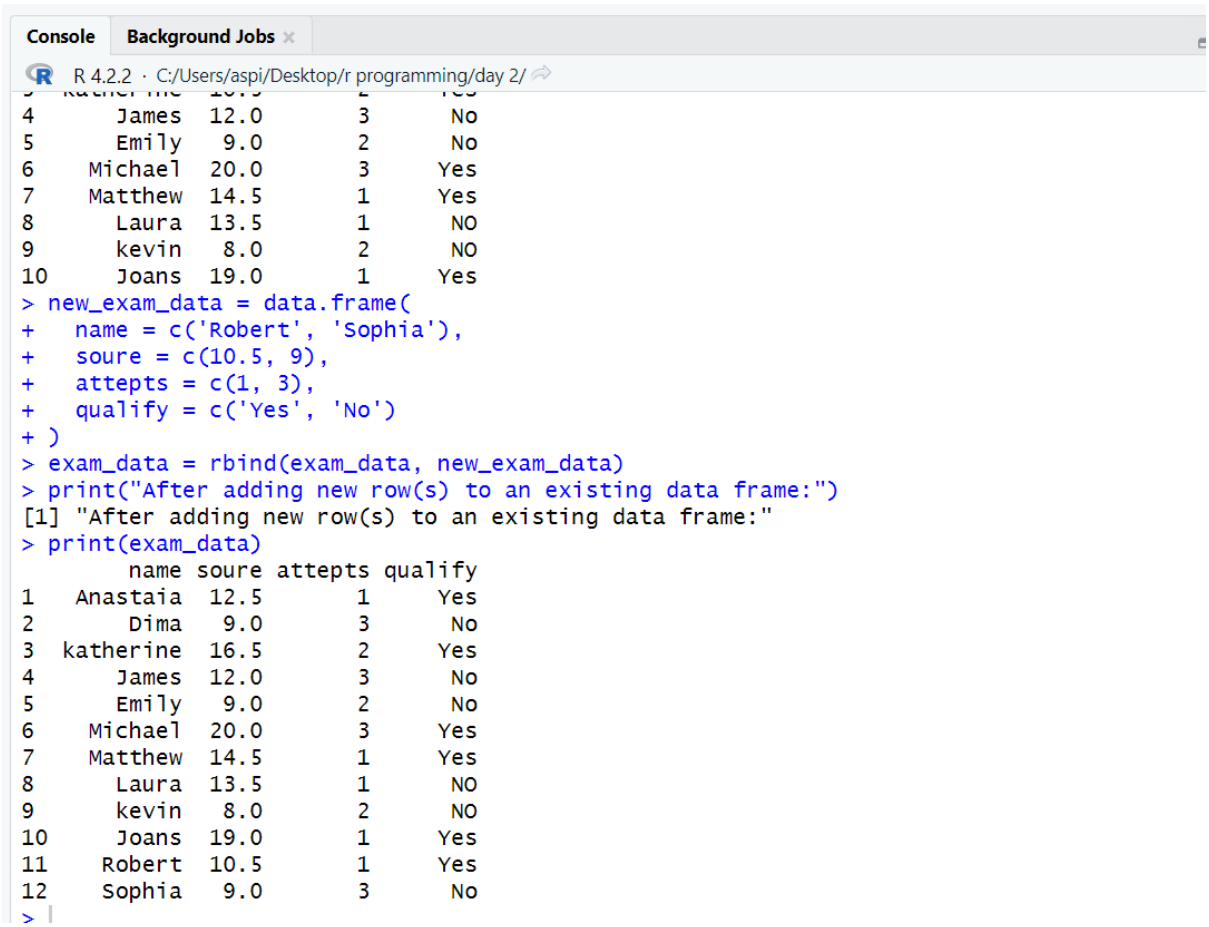
 soure = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),

 attepts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),

 qualify = c('Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'NO', 'NO', 'Yes')

)

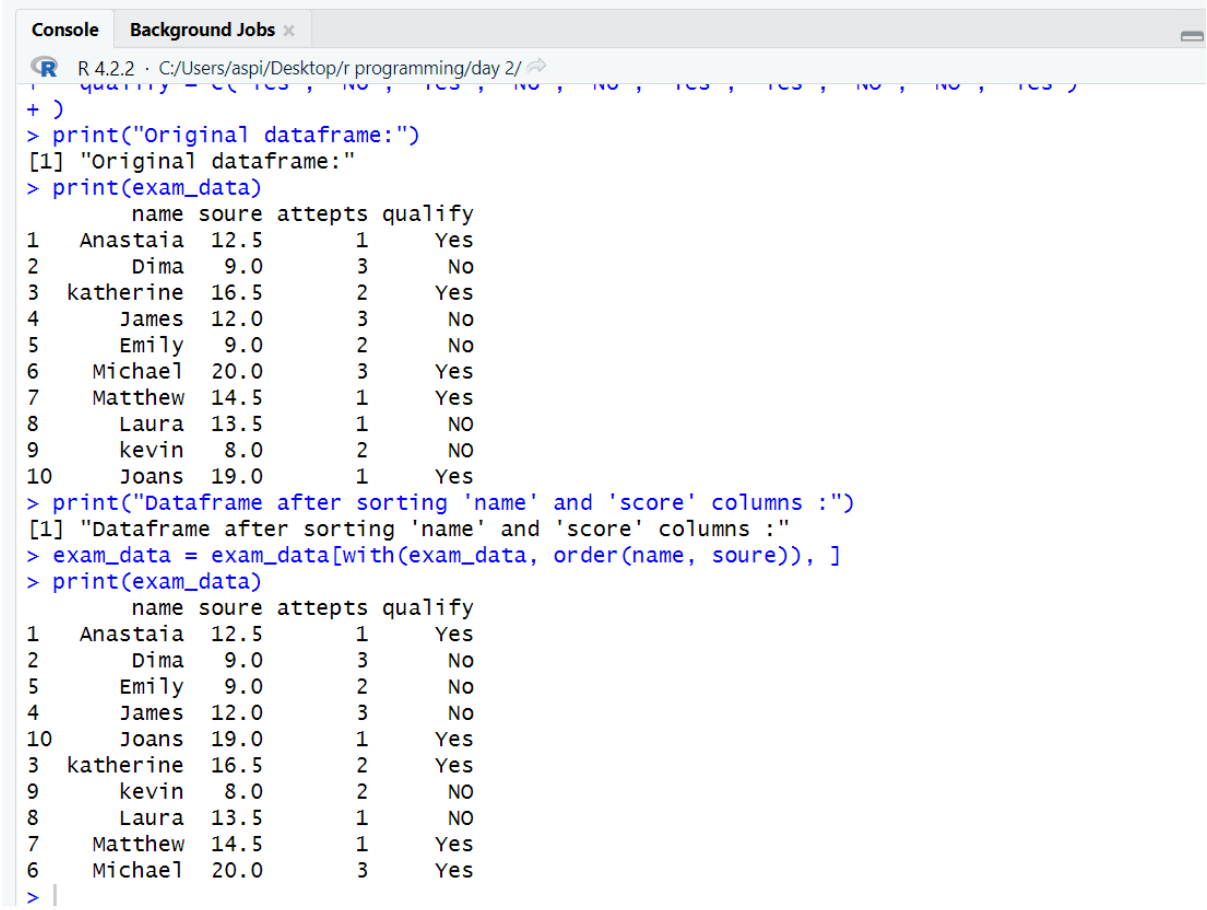
```
print("Original dataframe:")
print(exam_data)
new_exam_data = data.frame(
  name = c('Robert', 'Sophia'),
  soure = c(10.5, 9),
  attepts = c(1, 3),
  qualify = c('Yes', 'No')
)
exam_data = rbind(exam_data, new_exam_data)
print("After adding new row(s) to an existing data frame:")
print(exam_data)
```




```
4       James  12.0       3      No
5       Emily   9.0       2      No
6     Michael  20.0       3     Yes
7     Matthew  14.5       1     Yes
8       Laura  13.5       1      NO
9       kevin   8.0       2      NO
10      Joans  19.0       1     Yes
> new_exam_data = data.frame(
+   name = c('Robert', 'Sophia'),
+   soure = c(10.5, 9),
+   attepts = c(1, 3),
+   qualify = c('Yes', 'No')
+ )
> exam_data = rbind(exam_data, new_exam_data)
> print("After adding new row(s) to an existing data frame:")
[1] "After adding new row(s) to an existing data frame:"
> print(exam_data)
        name soure attepts qualify
1   Anastaia  12.5       1     Yes
2       Dima   9.0       3      No
3  katherine  16.5       2     Yes
4       James  12.0       3      No
5       Emily   9.0       2      No
6     Michael  20.0       3     Yes
7     Matthew  14.5       1     Yes
8       Laura  13.5       1      NO
9       kevin   8.0       2      NO
10      Joans  19.0       1     Yes
11     Robert  10.5       1     Yes
12     Sophia   9.0       3      No
>
```

8. Write a R program to sort a given data frame by name and score

```
exam_data = data.frame(
  name = c('Anastaia', 'Dima', 'katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'kevin', 'Joans'),
  soure = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attepts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'NO', 'NO', 'Yes')
)
print("Original dataframe:")
print(exam_data)
print("Dataframe after sorting 'name' and 'score' columns :")
exam_data = exam_data[with(exam_data, order(name, soure)), ]
print(exam_data)
```



9. Write a R program to save the information of a data frame in a file and display the information of the file.

```
exam_data = data.frame(
  name = c('Anastaia', 'Dima', 'katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'kevin', 'Joans'),
  soure = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
  attepts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c('Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'NO', 'NO', 'Yes')
)
print("Original dataframe:")
print(exam_data)
save(exam_data, file="data.rda")
load("data.rda")
file.info("data.rda")
```

```
R 4.2.2 · C:/Users/aspi/Desktop/r programming/day 2/
> exam_data = data.frame(
+   name = c('Anastaia', 'Dima', 'katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laur
a', 'kevin', 'Joans'),
+   soure = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),
+   attepts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+   qualify = c('Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'NO', 'NO', 'Yes')
+ )
> print("Original dataframe:")
[1] "Original dataframe:"
> print(exam_data)
        name soure attepts qualify
1   Anastaia  12.5       1     Yes
2       Dima   9.0       3      No
3  katherine  16.5       2     Yes
4      James  12.0       3      No
5      Emily   9.0       2      No
6    Michael  20.0       3     Yes
7    Matthew  14.5       1     Yes
8      Laura  13.5       1      NO
9      kevin   8.0       2      NO
10     Joans  19.0       1     Yes
> save(exam_data, file="data.rda")
> load("data.rda")
> file.info("data.rda")
         size isdir mode               mtime               ctime               atime
data.rda  306 FALSE  666 2023-02-02 16:01:06 2023-02-02 16:01:06 2023-02-02 16:01:06
         exe
data.rda  no
>
```

10. Write a R program to call the (built-in) dataset airquality. Check whether it is a data frame or not? Order the entire data frame by the first and second column. remove the variables 'Solar.R' and 'Wind' and display the data frame.

```
data("airquality")
if (is.data.frame(airquality)){
  print("airquality is a data frame.")
}else {
  print("airquality is not a data frame.")
}
airquality <- airquality[order(airquality$Ozone, airquality$Month),]
airquality <- airquality[ , !(names(airquality) %in% c("Solar.R", "Wind"))]
print(airquality)
```

Console | Background Jobs

R 4.2.2 · C:/Users/aspi/Desktop/r programming/day 2/

```
+ }
[1] "airquality is a data frame."
> airquality <- airquality[order(airquality$Ozone, airquality$Month),]
> airquality <- airquality[ , !(names(airquality) %in% c("Solar.R", "Wind"))]
> print(airquality)
    Ozone Temp Month Day
21      1   59     5  21
23      4   61     5  23
18      6   57     5  18
11      7   74     5  11
76      7   80     7  15
147     7   69     9  24
9       8   61     5   9
94      9   81     8   2
114     9   72     8  22
137     9   71     9  14
73     10   73     7  12
13     11   66     5  13
20     11   62     5  20
22     11   73     5  22
3      12   74     5   3
50     12   73     6  19
51     13   76     6  20
138    13   71     9  15
141    13   76     9  18
144    13   64     9  21
14     14   68     5  14
16     14   64     5  16
148    14   63     9  25
151    14   75     9  28
12     16   69     5  12
```

11. Write a R program to create a factor corresponding to height of women data set , which inbuild in R, contains height and weights for a sample of women.

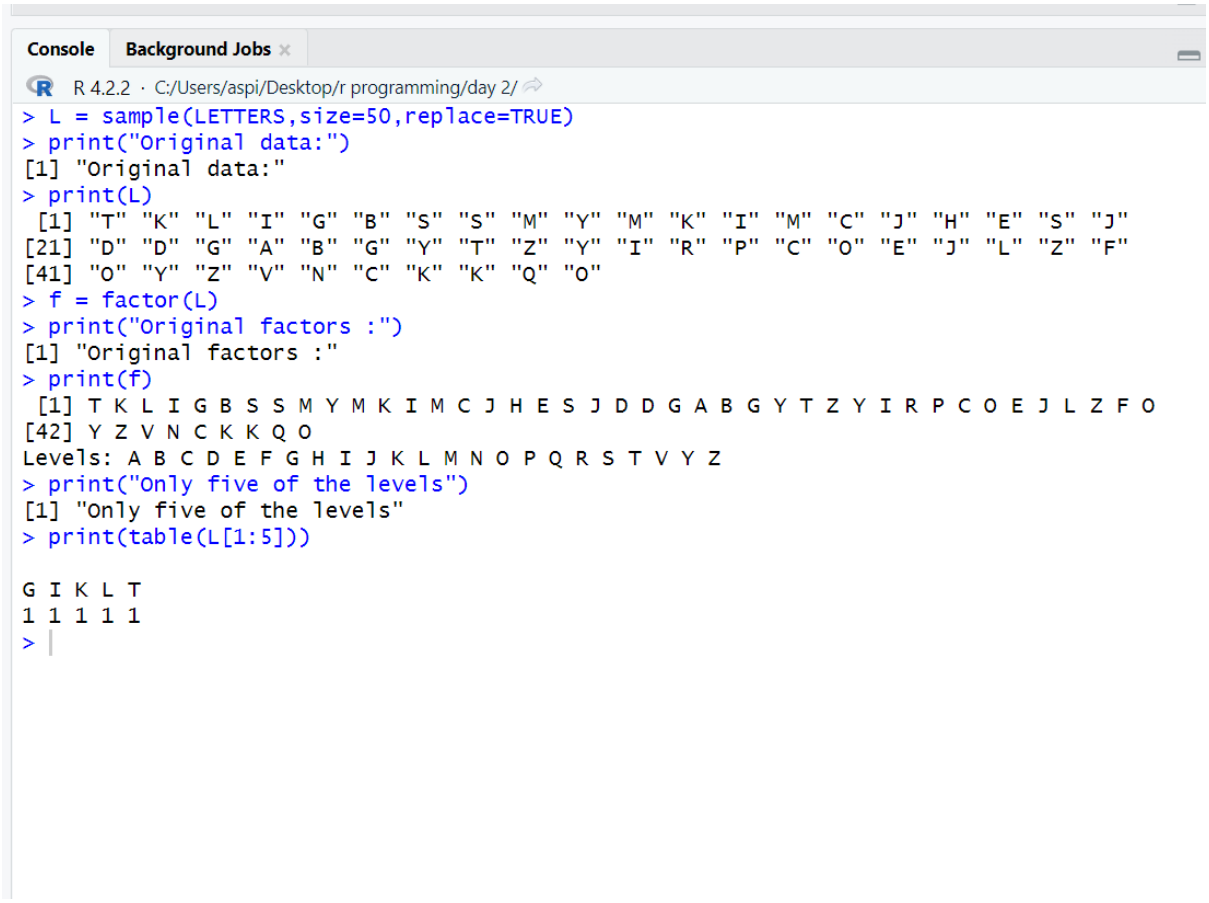
```
data = women
print("Women data set of height and weights :")
print(data)
height_f = cut(women$height,3)
print("factors corresponding to the height:")
print(table(height_f))
```

```
R 4.2.2 · C:/Users/aspi/Desktop/r programming/day 2/
> data = women
> print("Women data set of height and weights :")
[1] "Women data set of height and weights :"
> print(data)
   height weight
1      58    115
2      59    117
3      60    120
4      61    123
5      62    126
6      63    129
7      64    132
8      65    135
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
> height_f = cut(women$height,3)
> print("factors corresponding to the height:")
[1] "factors corresponding to the height:"
> print(table(height_f))
height_f
  (58,62.7] (62.7,67.3]   (67.3,72]
          5           5           5
>
```

12. Write a R program to extract the five of the levels of factor created from a random sample from the LETTERS (Part of the base R distribution.)

```
L = sample(LETTERS,size=50,replace=TRUE)
```

```
print("Original data:")
print(L)
f = factor(L)
print("Original factors :")
print(f)
print("Only five of the levels")
print(table(L[1:5]))
```



13. Iris dataset is a very famous dataset in almost all data mining, machine learning courses, and it has been an R build-in dataset. The dataset consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features(variables) were measured from each sample, they are the length and the width of sepal and petal, in centimetres. Perform the following EDA steps .

(i)Find dimension, Structure, Summary statistics, Standard Deviation of all features.

(ii)Find mean and standard deviation of features groped by three species of Iris flowers

(Iris setosa, Iris virginica and Iris versicolor)

(iii)Find quantile value of sepal width and length

(iV)create new data frame named iris1 which have a new column name

Sepal.Length.Cate that categorizes “Sepal.Length” by quantile

(V) Average value of numerical varialbes by two categorical variables: Species and

Sepal.Length.Cate:

(vi) Average mean value of numerical varialbes by Species and Sepal.Length.Cate

(vii)Create Pivot Table based on Species and Sepal.Length.Cate.

```
data("iris")

dim(iris)
str(iris)
summary(iris)
sapply(iris[,1:4], sd)

group_by_species <- aggregate(iris[,1:4], by=list(specis=iris$Species), FUN=function(x)
c(mean=mean(x), sd=sd(x)))
print(group_by_species)

quantile(iris$Sepal.Width)
quantile(iris$Sepal.Length)

irisl <- iris
irisl$Sepal.Length.Cate <- cut(irisl$Sepal.Length, quantile(irisl$Sepal.Length), lables = c("Q1",
"Q2", "Q3", "Q4"))
```

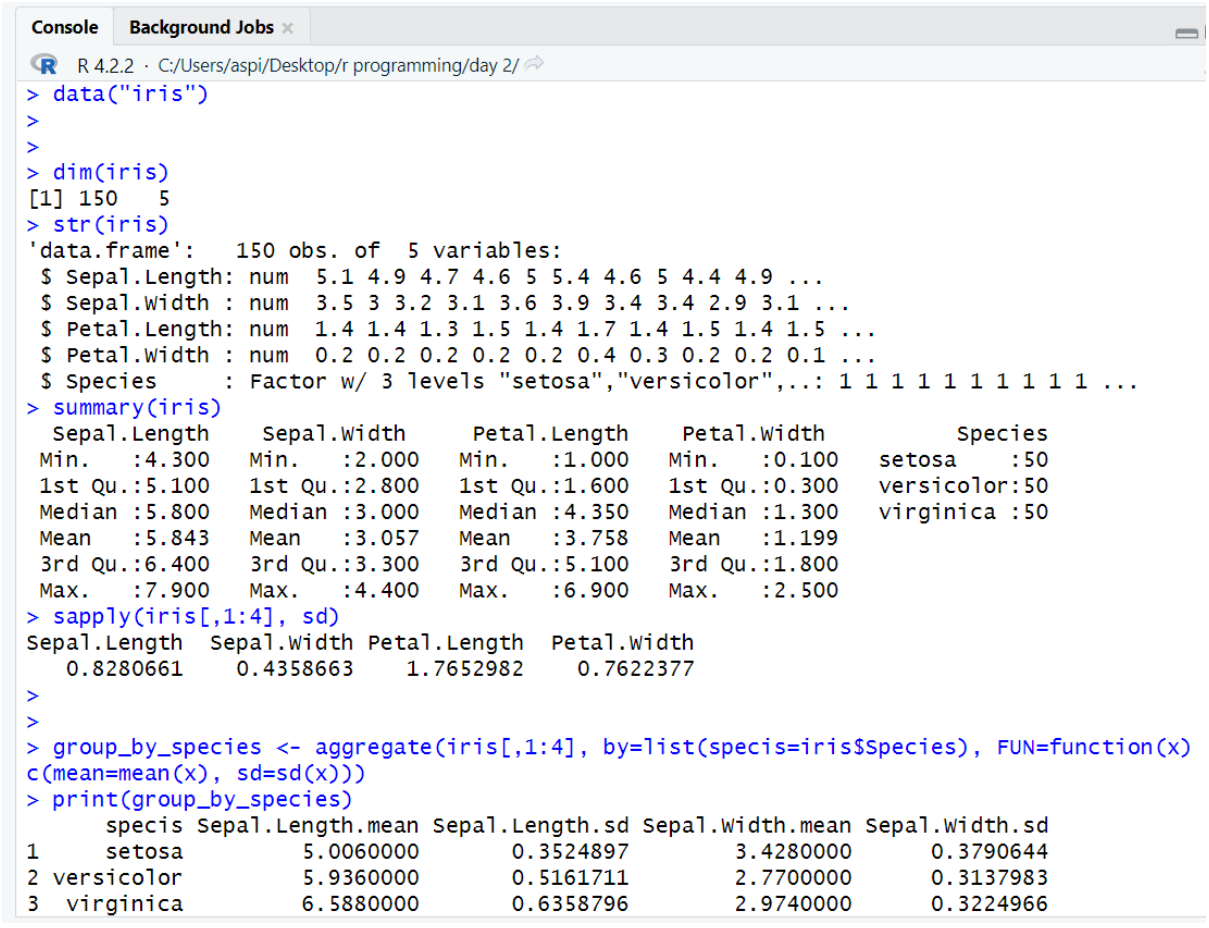
aggregate (iris[,1:4], by=list(Species=irisl$Species, Sepal.Length.Cate=irisl$Sepal.Length.Cate), mean)

aggregate (iris[,1:4], by=list(Species=irisl$Species, Sepal.Length.Cate=irisl$Sepal.Length.Cate), mean, FUN=mean)

library(reshape2)

plvot_table <- dcast(irisl, Species - sepal.Length.Cate, value.var = c("Sepal.Length", "Sepal.Width", "Peal.Length", "Petal.Width"), mean)

print(pivot_table)



```
Console   Background Jobs
R 4.2.2 · C:/Users/aspi/Desktop/r programming/day 2/
> data("iris")
>
>
> dim(iris)
[1] 150   5
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> sapply(iris[,1:4], sd)
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   0.8280661    0.4358663    1.7652982    0.7622377
>
>
> group_by_species <- aggregate(iris[,1:4], by=list(specis=iris$Species), FUN=function(x)
c(mean=mean(x), sd=sd(x)))
> print(group_by_species)
      specis Sepal.Length.mean Sepal.Length.sd Sepal.Width.mean Sepal.Width.sd
1     setosa          5.0060000       0.3524897        3.4280000      0.3790644
2 versicolor          5.9360000       0.5161711        2.7700000      0.3137983
3  virginica          6.5880000       0.6358796        2.9740000      0.3224966
```