

Introduction to Decision Making in R Programming

Decision making is significant aspect in every programming language. To check the conditions in form of TRUE and FALSE values we can get appropriate result with the help of decision making statements. R supports a decision making ability to take a decision in case of providing result by satisfying condition. R provides different decision making or conditional statements like IF, ELSE_IF.

R language also provides the looping statements for getting result in from particular range by satisfying the condition. The looping statements in R like FOR, WHILE and REPEAT etc. R supports a different operators like operator like logical (&), logical (!), arithmetic operator like (+,*) and relational operator like (==) etc. that can be used for in case of checking condition in decision making phase.

Vector: A vector is a basic data structure used in R programming. A vector stores the same type of element, the types of element is may be an integer, float, string or character. A vector concept is same as “Array” concept in procedural programming language and object oriented programming language. For creating a vector in R, R provides **c () function**.

Data Frame: A data frame is two dimensional structure of data used in R. It is a special case of list which contains equal length of components. It takes components from column and contents from rows. We can create a data data frame suing data. Frame () function provided by R. Combine all vectors in list using data. Frame () function followed by the vectors name separated by commas.

Read : A read() method used to read the contents of .csv file in R language. A read() followed by .csv file name in double quotes read the information from csv file.

Note: R programming is the case sensitive language so we have to write commands as same as commands provided by R.

For run all commands of script, select commands and press **ctrl+enter**.

Experiments:

Consider the following data set and solve the following problems.

EmpNo	Age	Height	Weight	Basic Salary	Experience	Location
101	45	6.2	67	10000	6.3	Sangli
102	23	4.7	89	6700	7.2	Satara
103	26	5.2	54	6400	2.5	Nashik
104	31	5.5	68	8400	7.5	Kolhapur
105	30	5.4	44	8450	4.8	Jalgao
106	22	4.9	56	8100	9.4	Pune
107	26	6.2	51	9500	3.5	Jalgao
108	24	4.6	55	15000	8.4	Satara
109	28	5.1	69	18450	9.0	Sangli

110	29	5.9	89	7400	4.6	Satara
111	48	5.4	95	25000	8.2	Pune
112	42	4.3	40	22000	9.4	Kolhapur
113	49	4.9	77	9000	1.5	Nashik
114	32	5.8	70	9400	1.6	Jalga
115	33	6.8	74	12400	2.7	Pune
116	46	6.2	60	45000	4.8	Satara
117	36	4.3	48	43000	8.7	Pune
118	35	5.9	59	17500	9.4	Nashik
119	32	5.5	66	18600	10.5	Kolhapur
120	27	4.5	80	17800	11.4	Pune

Introduction:-

The purpose of this experiment is to understand the concept of reading contents of .csv file using read command.

Procedure:

1. Open R studio take a new script.
2. Write `rm(list=ls())` command on script to clear workspace.
3. Read `EmpDetails.csv` file using `read` command in R and store it result on `Employee_Information` variable.
4. Print `Employee_Information`.
5. Run all commands by pressing `ctrl+enter`.
6. Save the script with name `R_code_decision` extension `.r` on desired location like `Excercise_1.r`

Code and Result :-

Open R studio take a new script.

#1.open R studio and take new script

#1.1 open R studio and go to file option

#1.2 after selecting file option and click new File and after selecting new file another window is appeared on screen in that select R script.

#1.3 the untitled R script in opened on screen.

Write `rm(list=ls())` command on script to clear workspace

#2. Write `rm(list=ls())` command in script for clearing all workspace.

```
rm(list=ls())
```

Read `EmpDetails.csv` file using `read` command in R and store the result of it on `Employee_Information` variable

#3. read EmpDetails.csv file using read command and store it on Employee_Information variable

```
Employee_Information<-read.csv("EmpDetails.csv")
```

Print Employee_Information using print function.

#use print command to print contents of file EmpDetails.csv

```
print(Employee_Information)
```

Save this script using save option from file.

Save script Exercise_1.r using save option.

Output : -

```
> #3. read Employee_Details.csv file using read command and store it on Employee_Information variable
> Employee_Information<-read.csv("EmpDetails.csv")
> #use print command to print contents of file Employee_Information
> print(Employee_Information)
```

	X.1	X	EmpNo	Age	Height	weight	BasicSalary	Experience	Location
1	1	1	101	45	6.2	67	10000	6.3	Sangli
2	2	2	102	23	4.7	89	6700	7.2	Satara
3	3	3	103	26	5.2	54	6400	2.5	Nashik
4	4	4	104	31	5.5	68	8400	7.5	Kolhapur
5	5	5	105	30	5.4	44	8450	4.8	Jalgao
6	6	6	106	22	4.9	56	8100	9.4	Pune
7	7	7	107	26	6.2	51	9500	3.5	Jalgao
8	8	8	108	24	4.6	55	15000	8.4	Satara
9	9	9	109	28	5.1	69	18450	9.0	Sangli
10	10	10	110	29	5.9	89	7400	4.6	Satara
11	11	11	111	48	5.4	95	25000	8.2	Pune
12	12	12	112	42	4.3	40	22000	9.4	Kolhapur
13	13	13	113	49	4.9	77	9000	1.5	Nashik
14	14	14	114	32	5.8	70	9400	1.6	Jalgao
15	15	15	115	33	6.8	74	12400	2.7	Pune
16	16	16	116	46	6.2	60	45000	4.8	Satara
17	17	17	117	36	4.3	48	43000	8.7	Pune
18	18	18	118	35	5.9	59	17500	9.4	Nashik
19	19	19	119	32	5.5	66	18600	10.5	Kolhapur
20	20	20	120	27	4.5	80	17800	11.4	Pune

Conclusion: -

We can read the contents of .csv file in R using read() method in R .

1. Display employees whose age is between 30 to 35.

Introduction:

The purpose of this experiment is to understand the concept of accessing elements within a range using logical operator &. And use the subset () function for display values which satisfies the condition written in subset () function.

Procedure:-

1. Create a variable as show_age . Create subset with condition Age>=30 & Age<=35
2. Print show_age
3. Run all commands by pressing ctrl+enter .
4. Save the script using save option from file.

Code And Result :-

Open R studio and open Excercise_1.r script

Create a variable as show_age . Create subset with condition Age>=30 & Age<=35

```
#Create a variable as show_age . Create subset with condition Age>=30 & Age<=35
```

```
show_age<-data.frame(subset(Employee_Information, Age>=30 & Age<=35))
```

Print show_age

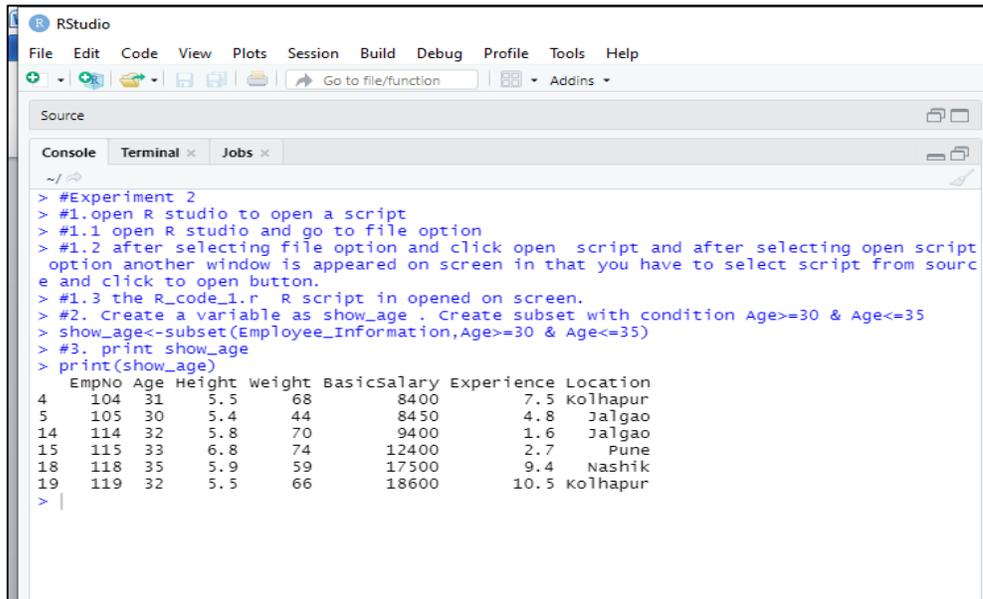
```
#3. print show_age
```

```
print(show_age)
```

Save the script Excercise_1.r on using save option..

Save script with name Exercise_1.r

Output –



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal x Jobs x
~/
> #Experiment 2
> #1.open R studio to open a script
> #1.1 open R studio and go to file option
> #1.2 after selecting file option and click open script and after selecting open script
option another window is appeared on screen in that you have to select script from source
and click to open button.
> #1.3 the R_code_1.r R script in opened on screen.
> #2. Create a variable as show_age . Create subset with condition Age>=30 & Age<=35
> show_age<-subset(Employee_Information,Age>=30 & Age<=35)
> #3. print show_age
> print(show_age)
  EmpNo Age Height weight BasicSalary Experience Location
4    104  31   5.5    68         8400         7.5  Kolhapur
5    105  30   5.4    44        8450         4.8   Jalgao
14   114  32   5.8    70        9400         1.6   Jalgao
15   115  33   6.8    74       12400         2.7    Pune
18   118  35   5.9    59       17500         9.4   Nashik
19   119  32   5.5    66       18600        10.5  Kolhapur
> |
```

Conclusion –

We understand the concept of using logical operator & ‘AND’. We can use subset () function to make a subset of given data. We can perform basic arithmetic operations on elements using logical operator and make separate set of those operations result using subset () function.

2. Display employees whose salary >15000.

1. A relational > operator used when you want to display values greater than from particular value.
2. We can use subset() function to perform this kind of basic operations to get result in another subset of data.
3. For run all commands of script, select commands and press **ctrl+enter**.

Introduction:

The purpose of this experiment is to understand the concept of conditional operator '>' greater than for performing operations and store a result in subset.

Procedure:-

1. Create a variable as show salary. Create subset with condition BasicSalary>15000
2. Print show_salary.
3. Run all commands by pressing ctrl+enter.
4. Save the script using save option from file.

Code And Result :-

Create a variable as show salary. Create subset with condition BasicSalary>15000

#Create a variable as show salary. Create subset with condition BasicSalary>15000

```
show_salary <-subset(Employee_Information,BasicSalary>15000)
```

Print show_salary

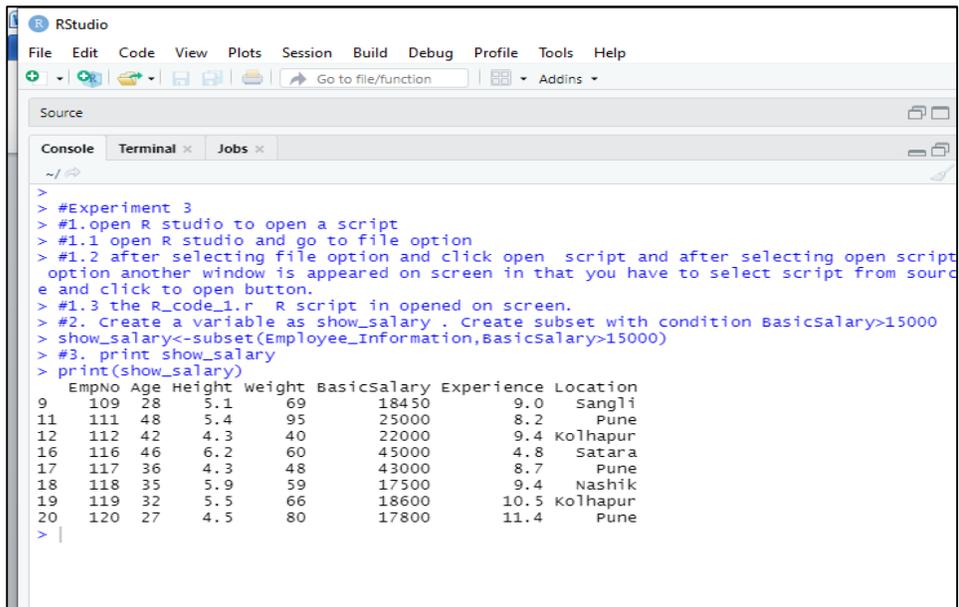
#3. print show_salary

```
print (show_salary)
```

Save the script Excercise_1.r on using save option..

Save script with name Excercise_1.r

Output –



```
>
> #Experiment 3
> #1.open R studio to open a script
> #1.1 open R studio and go to file option
> #1.2 after selecting file option and click open script and after selecting open script
option another window is appeared on screen in that you have to select script from source
and click to open button.
> #1.3 the R_code_1.r R script in opened on screen.
> #2. create a variable as show_salary . Create subset with condition Basicsalary>15000
> show_salary<-subset(Employee_Information,Basicsalary>15000)
> #3. print show_salary
> print(show_salary)
  EmpNo Age Height weight Basicsalary Experience Location
9     109  28    5.1    69      18450         9.0  Sangli
11    111  48    5.4    95      25000         8.2   Pune
12    112  42    4.3    40      22000         9.4 Kolhapur
16    116  46    6.2    60      45000         4.8  Satara
17    117  36    4.3    48      43000         8.7   Pune
18    118  35    5.9    59      17500         9.4  Nashik
19    119  32    5.5    66      18600        10.5 Kolhapur
20    120  27    4.5    80      17800        11.4   Pune
> |
```

Conclusion –

We understand the concept of using relational operator > ‘Greater Than’ and storing a result of operation in subset.

3. Display employees details whose EmpNo is even.

1. **Arithmetic %% operator:** - A %% operator used when you want to calculate remainder of value.
2. **Relational == operator:** - A '==' exact equals to operator is used when display result when resulted values are exactly equals to expected values.
3. We can use subset () function to perform this kind of basic operations to get result in another subset of data.
4. For run all commands of script, select commands and press **ctrl+enter**.

Introduction:

The purpose of this experiment is to understand the concept of use of Relational '%%' modulus operator and '==' exact equals to operator.

Procedure:-

1. Open R studio open R_code_decision.r script.
2. Create a variable as show even. Create subset with condition for even number on EmpNo element using %% and == operators.
3. Print show even.
4. Run all commands by pressing ctrl+enter.
5. Save the script using save option from file.

Code & Result:-

Open R studio and open Excercise_1.r script

Create a variable as show_even. Create subset with condition for even number on EmpNo element using %% and == operators.

#1. Create a variable as show_even. Create subset with condition for even number on EmpNo element using %% and == operators.

```
show_even<-subset (Employee_Information,EmpNo%%2==0)
```

Print show_even

#2. print show_even

```
print(show_even)
```

Save the script Excercise_1.r on using save option..

Save script with name Excercise_1.r

Output –

```
> #Experiment 4
> #1. open R studio to open a script
> #1.1 open R studio and go to file option
> #1.2 after selecting file option and click open script and after selecting open script
option another window is appeared on screen in that you have to select script from source
and click to open button.
> #1.3 the R_code_1.r R script in opened on screen.
> #2. Create a variable as show_even . Create subset with condition for even number on Em
pNo element using %% and == operators.
> show_even<-subset(Employee_Information,EmpNo%%2==0)
> #3. print show_even
> print(show_even)
  EmpNo Age Height weight BasicSalary Experience Location
2    102  23   4.7    89      6700         7.2   Satara
4     104  31   5.5    68      8400         7.5  Kolhapur
6     106  22   4.9    56      8100         9.4     Pune
8     108  24   4.6    55     15000         8.4   Satara
10    110  29   5.9    89      7400         4.6   Satara
12    112  42   4.3    40     22000         9.4  Kolhapur
14    114  32   5.8    70      9400         1.6  Jalgaon
16    116  46   6.2    60     45000         4.8   Satara
18    118  35   5.9    59     17500         9.4   Nashik
20    120  27   4.5    80     17800        11.4    Pune
```

Conclusion –

We understand the concept of using arithmetic operator %% ‘Modulus operator’ and relational ‘==’ ‘exact equals to’ to performing logic of even numbers on elements .It shows result only when given condition satisfies and storing a result of operation in subset.

4. Display employees whose height <5.5 and weight>60.

How to extract a values of elements by satisfying given conditions using relational operators ?

1. **Relational > operator:** - A greater than > operator used when you want to display value greater than particular values specified in condition.
2. **Relational < operator:** - A less than > operator used when you want to display value less than particular values specified in condition.
3. We can use subset () function to perform this kind of basic operations to get result in another subset of data.
4. For run all commands of script, select commands and press **ctrl+enter**.

Introduction:

The purpose of this experiment is to understand the concept of use of Relational operators greater than and less than.

Procedure:-

1. Open R studio open Excercise_1.r script.
2. Create a variable as show_height . Create subset with condition displaying Height<5.5 .
3. Print show_height.
4. Create a variable as show_weight . Create subset with condition displaying Weight>60 .
5. Print show_weight
6. Run all commands by pressing ctrl+enter .
7. Save the script using save option from file.

Code and Result:-

Open R studio and open Excercise_1.r script

Create a variable as show_height . Create subset with condition displaying Height<5.5 .

#2. Create a variable as show_height . Create subset with condition for displaying height<5.5.

```
show_height<-subset(Employee_Information,Height<5.5)
```

Print show_height

#3. print show_height

```
print(show_height)
```

Create a variable as show_weight . Create subset with condition displaying Weight>60 .

#4. Create a variable as show_weight . Create subset with condition for displaying weight>60.

```
show_weight<-subset(Employee_Information,Weight>60)
```

Print show_weight

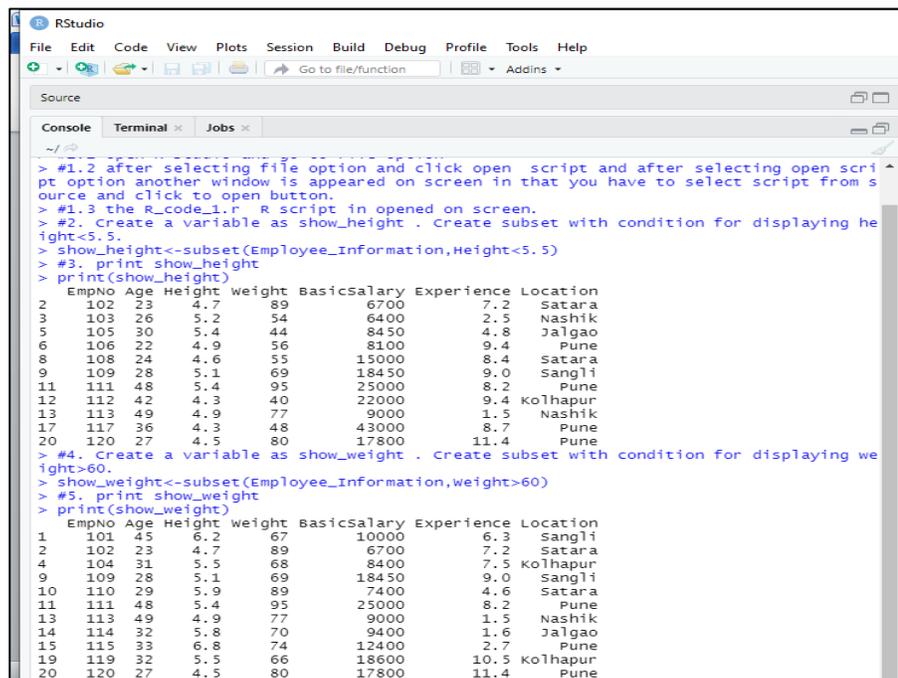
#5. print show_weight

```
print(show_weight)
```

Save the script Excercise_1.r on using save option..

Save script with name Excercise_1.r

Output –



```
> #1.2 after selecting file option and click open script and after selecting open script option another window is appeared on screen in that you have to select script from source and click to open button.
> #1.3 the R_code_1.r R script is opened on screen.
> #2. Create a variable as show_height . Create subset with condition for displaying height<5.5.
> show_height<-subset(Employee_Information,Height<5.5)
> #3. print show_height
> print(show_height)
  EmpNo Age Height weight BasicSalary Experience Location
2 102 23 4.7 89 6700 7.2 Satara
3 103 26 5.2 54 6400 2.5 Nashik
5 105 30 5.4 44 8450 4.8 Jalgaon
6 106 22 4.9 56 8100 9.4 Pune
8 108 24 4.6 55 15000 8.4 Satara
9 109 28 5.1 69 18450 9.0 Sangli
11 111 48 5.4 95 25000 8.2 Pune
12 112 42 4.3 40 22000 9.4 Kolhapur
13 113 49 4.9 77 9000 1.5 Nashik
17 117 36 4.3 48 43000 8.7 Pune
20 120 27 4.5 80 17800 11.4 Pune
> #4. Create a variable as show_weight . Create subset with condition for displaying weight>60.
> show_weight<-subset(Employee_Information,weight>60)
> #5. print show_weight
> print(show_weight)
  EmpNo Age Height weight BasicSalary Experience Location
1 101 45 6.2 67 10000 6.3 Sangli
2 102 23 4.7 89 6700 7.2 Satara
4 104 31 5.5 68 8400 7.5 Kolhapur
9 109 28 5.1 69 18450 9.0 Sangli
10 110 29 5.9 89 7400 4.6 Satara
11 111 48 5.4 95 25000 8.2 Pune
13 113 49 4.9 77 9000 1.5 Nashik
14 114 32 5.8 70 9400 1.6 Jalgaon
15 115 33 6.8 74 12400 2.7 Pune
19 119 32 5.5 66 18600 10.5 Kolhapur
20 120 27 4.5 80 17800 11.4 Pune
```

Conclusion –

We understand the concept of using relational operators like greater than “>” and less than “<”. It shows result only when given condition satisfies and storing a result of operation in subset.

5. Check employees height, weight and display whether persons fit, overweight or underweight.

Introduction:

The purpose of this experiment is to understand the concept of creating/ adding a new column/element in data frame and use of decision statements and looping statements in R.

Procedure:-

1. Open R studio open Excercise_1.r script.
2. Create a variable temp and assign operation of converting height fit to meters.
3. Create a new element Height_in_meter in Employee_Information and assign temp variable values to Height_in_meter.
4. Print Employee_Information\$Height_in_meter
5. Create a variable temp1 and assign operation of making square of Height_in_meter element.
6. Print temp1
7. Create a new element Height_sqr in Employee_Information and assign temp1 variable values to Height_sqr.
8. Print Employee_Information\$Height_sqr
9. Create element BMI_ratio and assign operation of calculating BMI ration using Weight and Height_sqr element.
10. print Employee_Information\$BMI_ratio
11. View Employee_Information\$BMI_ratio
12. In for loop use e1 variable and take range 1: nrow(Employee_Information) for whole dataset.
13. check condition using IF statement for fit , underweight and overweight
14. Add new column as a Result_of_BMI and add assign the value as Underweight when the given condition is true.
15. Use else if statements to check whether the element satisfies a second condition when the first condition is false.
16. Assign values as Fit to Result_of_BMI column when the given condition is true.
17. Use else to display else part.
18. Assign values as Overweight to Result_of_BMI column when the above conditions are false and else part is executed.
19. Print Employee_Information data frame to check desired result.
20. View Employee_Information data frame to check desired result in tabular form.
21. Run all commands by pressing ctrl+enter.
22. Save the script using save option from file.

Code and Result:-

#1.Create a variable temp and assign operation of converting height fit to meters

```
temp<-Employee_Information$Height*0.3048
```

#2.Create a new element Height_in_meter in Employee_Information and assign temp variable values to Height_in_meter.

```
Employee_Information$Height_in_meter<-temp  
Print Employee_Information$Height_in_meter  
#print Employee_Information$Height_in_meter  
print(Employee_Information$Height_in_meter)
```

#3.Create a variable temp1 and assign operation of making square of Height_in_meter element.

```
temp1<-Employee_Information$Height_in_meter*2  
Print temp1  
#print temp1  
print(temp1)
```

#4.Create a new element Height_sqr in Employee_Information and assign temp1 variable values to Height_sqr.

```
Employee_Information$Height_sqr<-temp1  
Print Employee_Information$Height_sqr  
#print Employee_Information$Height_sqr  
print (Employee_Information$Height_sqr)
```

#5. Create element BMI_ratio and assign operation of calculating BMI ration using Weight and Height_sqr element.

```
Employee_Information$BMI_ratio<-  
Employee_Information$Weight/Employee_Information$Height_sqr
```

```
Print Employee_Information$BMI_ratio  
#print Employee_Information$BMI_ratio  
print(Employee_Information$BMI_ratio)  
View Employee_Information$BMI_ratio  
#View Employee_Information$BMI_ratio  
View(Employee_Information$BMI_ratio)
```

In for loop use e1 variable and take range 1:nrow(Employee_Information) for whole dataset.

#6.In for loop use e1 variable and take range 1:nrow(Employee_Information) for whole dataset.

```
for(e1 in 1:nrow(Employee_Information))
```

```
{
```

Check condition using IF statement for fit , underweight and overweight

```
#check condition using IF statement for fit , underweight and overweight
```

```
if(Employee_Information$BMI[e1]>18.5)
```

```
{
```

Add new column as a Result_of_BMI and add assign the value as Underweight when the given condition is true.

```
#Add new column as a Result_of_BMI and add assign the value as Underweight when the given condition is true.
```

```
Employee_Information$Result_of_BMI[e1]<-"UnderWeight"
```

```
}
```

Use else if statements to check whether the element satisfies a second condition when the first condition is false

```
# Use else if statements to check whether the element satisfies a second condition when the first condition is false.
```

```
else if(Employee_Information$BMI[e1]>=18.5 Employee_Information$BMI[e1]<=24)
```

```
{
```

Assign values as Fit to Result_of_BMI column when the given condition is true.

```
#Assign values as Fit to Result_of_BMI column when the given condition is true.
```

```
Employee_Information$Result_of_BMI[e1]<-"Fit"
```

```
}
```

Use else to display else part.

```
# Use else to display else part.
```

```
else
```

```
{
```

Assign values as Overweight to Result_of_BMI column when the above conditions are false and else part is executed.

#Assign values as Overweight to Result_of_BMI column when the above conditions are false and else part is executed.

```
Employee$Result_of_BMI[e1]<-"OverWeight  
}  
}
```

Print Employee_Information data frame to check desired result.

#8.Print Employee_Information data frame to check desired result.

```
print(Employee_Information)
```

View Employee_Information data frame to check desired result in tabular form.

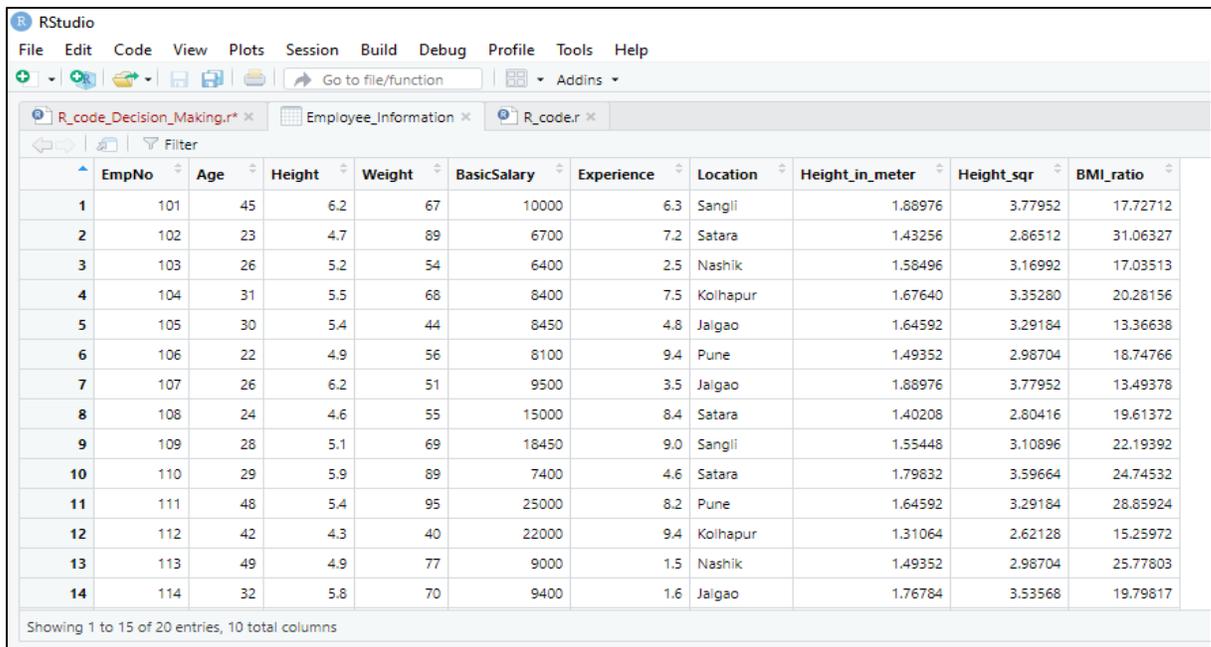
#9.Print Employee_Information data frame to check desired result.

```
View(Employee_Information)
```

Save the script Excercise_1.r on using save option..

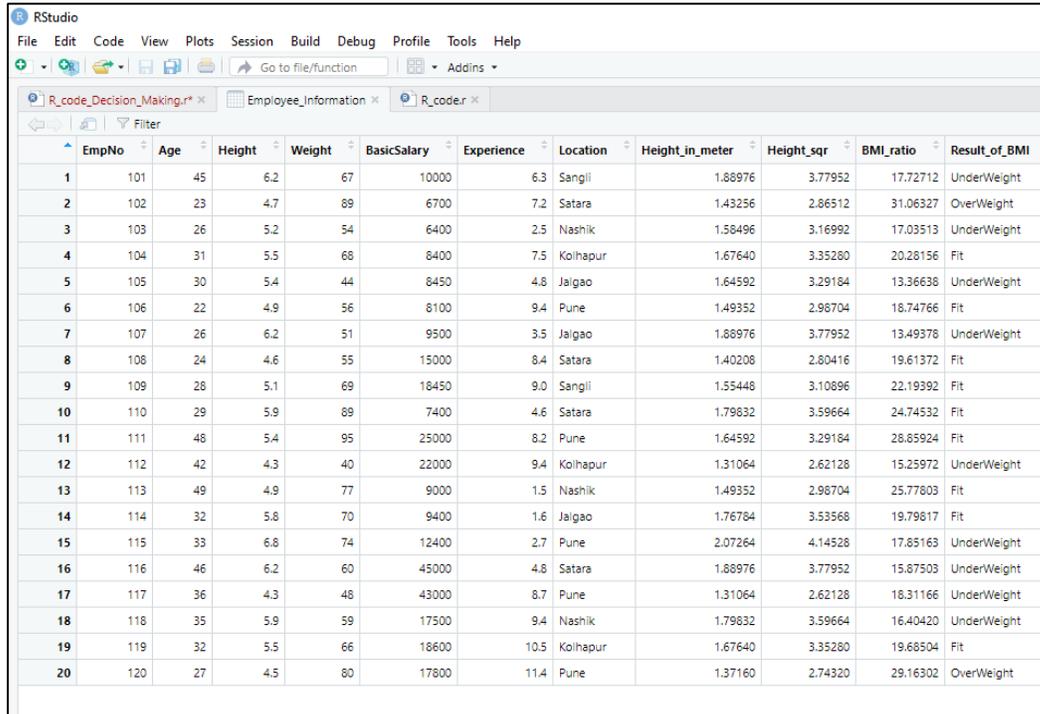
Save script with name Excercise_1.r

Output – 1. BMI ratio :



	EmpNo	Age	Height	Weight	BasicSalary	Experience	Location	Height_in_meter	Height_sqr	BMI_ratio
1	101	45	6.2	67	10000	6.3	Sangli	1.88976	3.77952	17.72712
2	102	23	4.7	89	6700	7.2	Satara	1.43256	2.86512	31.06327
3	103	26	5.2	54	6400	2.5	Nashik	1.58496	3.16992	17.03513
4	104	31	5.5	68	8400	7.5	Kolhapur	1.67640	3.35280	20.28156
5	105	30	5.4	44	8450	4.8	Jalgaon	1.64592	3.29184	13.36638
6	106	22	4.9	56	8100	9.4	Pune	1.49352	2.98704	18.74766
7	107	26	6.2	51	9500	3.5	Jalgaon	1.88976	3.77952	13.49378
8	108	24	4.6	55	15000	8.4	Satara	1.40208	2.80416	19.61372
9	109	28	5.1	69	18450	9.0	Sangli	1.55448	3.10896	22.19392
10	110	29	5.9	89	7400	4.6	Satara	1.79832	3.59664	24.74532
11	111	48	5.4	95	25000	8.2	Pune	1.64592	3.29184	28.85924
12	112	42	4.3	40	22000	9.4	Kolhapur	1.31064	2.62128	15.25972
13	113	49	4.9	77	9000	1.5	Nashik	1.49352	2.98704	25.77803
14	114	32	5.8	70	9400	1.6	Jalgaon	1.76784	3.53568	19.79817

2. For Fit, Overweight, and Underweight Condition:



The screenshot shows the RStudio interface with a data table. The table has 12 columns: EmpNo, Age, Height, Weight, BasicSalary, Experience, Location, Height_in_meter, Height_sqr, BMI_ratio, and Result_of_BMI. The data is sorted by EmpNo from 1 to 20. The Result_of_BMI column contains values like 'UnderWeight', 'OverWeight', and 'Fit'.

EmpNo	Age	Height	Weight	BasicSalary	Experience	Location	Height_in_meter	Height_sqr	BMI_ratio	Result_of_BMI	
1	101	45	6.2	67	10000	6.3	Sangli	1.88976	3.77952	17.72712	UnderWeight
2	102	23	4.7	89	6700	7.2	Satara	1.43256	2.86512	31.06327	OverWeight
3	103	26	5.2	54	6400	2.5	Nashik	1.58496	3.16992	17.03513	UnderWeight
4	104	31	5.5	68	8400	7.5	Kolhapur	1.67640	3.35280	20.28156	Fit
5	105	30	5.4	44	8450	4.8	Jalgao	1.64592	3.29184	13.36638	UnderWeight
6	106	22	4.9	56	8100	9.4	Pune	1.49352	2.98704	18.74766	Fit
7	107	26	6.2	51	9500	3.5	Jaigao	1.88976	3.77952	13.49378	UnderWeight
8	108	24	4.6	55	15000	8.4	Satara	1.40208	2.80416	19.61372	Fit
9	109	28	5.1	69	18450	9.0	Sangli	1.55448	3.10896	22.19392	Fit
10	110	29	5.9	89	7400	4.6	Satara	1.79832	3.59664	24.74532	Fit
11	111	48	5.4	95	25000	8.2	Pune	1.64592	3.29184	28.85924	Fit
12	112	42	4.3	40	22000	9.4	Kolhapur	1.31064	2.62128	15.25972	UnderWeight
13	113	49	4.9	77	9000	1.5	Nashik	1.49352	2.98704	25.77803	Fit
14	114	32	5.8	70	9400	1.6	Jaigao	1.76784	3.53568	19.79817	Fit
15	115	33	6.8	74	12400	2.7	Pune	2.07264	4.14528	17.85163	UnderWeight
16	116	46	6.2	60	45000	4.8	Satara	1.88976	3.77952	15.87503	UnderWeight
17	117	36	4.3	48	43000	8.7	Pune	1.31064	2.62128	18.31166	UnderWeight
18	118	35	5.9	59	17500	9.4	Nashik	1.79832	3.59664	16.40420	UnderWeight
19	119	32	5.5	66	18600	10.5	Kolhapur	1.67640	3.35280	19.68504	Fit
20	120	27	4.5	80	17800	11.4	Pune	1.37160	2.74320	29.16302	OverWeight

Conclusion –

We understand the concept of decision making statements and looping statements in R. We can add the new column or element of data frame in existing data frame. We can perform different operations in R is easily possible using R contents

6. Calculate gross salary of employees (TA is 10%, DA is 15%, HRA is 20%) on basic salary.

How to calculate Gross salary on the basis of given conditions using given dataset?

1. **Gross Salary:** - **Gross salary** is mathematical calculation for salary payable to employee after adding different allowances and deducting some taxes from his basic salary.

Syntax for Gross Salary:

$$\text{Gross} = \text{BasicSalary} - \text{TA} + \text{DA} + \text{HRA}$$

2. For run all commands of script, select commands and press **ctrl+enter**.

Introduction:

The purpose of this experiment is to understand the concept to calculate gross salary from given data set.

Procedure:-

1. Open R studio open Excercise_1.r script.
2. Add new column as TA add calculate TA 10% on BasicSalary.
3. Add new column as DA add calculate DA 15% on BasicSalary.
4. Add new column as HRA add calculate HRA 20% on BasicSalary.
5. View Employee_Information
6. Add new column Gross_Salary and calculate Gross salary and assign the result to this column.
7. View Emp_saldata data frame to get desired result.
8. Run all commands by pressing ctrl+enter.
9. Save the script using save option from file.

Code And Result :-

Add new column as TA add calculate TA 10% on BasicSalary.

#1.Add new column as TA add calculate TA 10% on BasicSalary.

```
Employee_Information$TA<-Employee_Information$BasicSalary*10/100
```

Add new column as DA add calculate DA 15% on BasicSalary.

#2.Add new column as DA add calculate DA 15% on BasicSalary.

```
Employee_Information$DA<-Employee_Information$BasicSalary*15/100
```

Add new column as HRA add calculate HRA 20% on BasicSalary.

#3.Add new column as HRA add calculate HRA 20% on BasicSalary.

```
Employee_Information$HRA<-Employee_Information$BasicSalary*20/100
```

View Employee_Information data frame to get desired result.

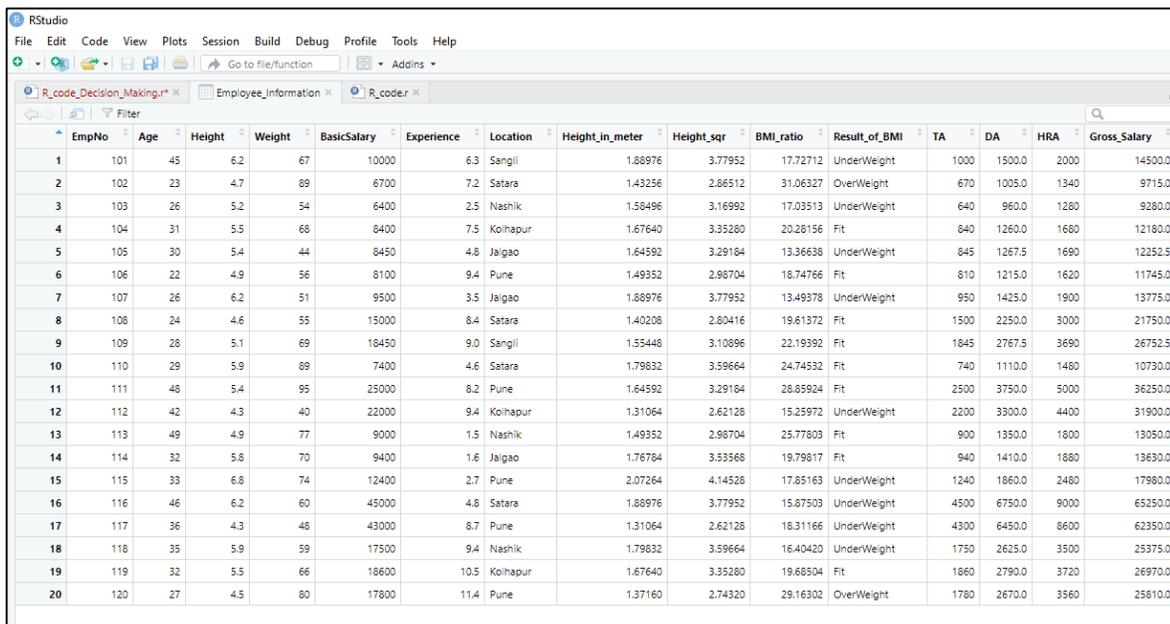
#4.View Employee_Information data frame

View (Employee_Information)

Save the script Excercise_1.r on using save option..

Save script with name Excercise_1.r

Output -



EmpNo	Age	Height	Weight	BasicSalary	Experience	Location	Height_in_meter	Height_sqr	BMI_ratio	Result_of_BMI	TA	DA	HRA	Gross_Salary	
1	101	45	6.2	67	10000	6.3	Sangli	1.88976	3.77952	17.72712	UnderWeight	1000	1500.0	2000	14500.0
2	102	23	4.7	89	6700	7.2	Satara	1.43256	2.86512	31.06327	OverWeight	670	1005.0	1340	9715.0
3	103	26	5.2	54	6400	2.5	Nashik	1.58496	3.16992	17.03513	UnderWeight	640	960.0	1280	9280.0
4	104	31	5.5	68	8400	7.5	Kolhapur	1.67640	3.35280	20.26156	Fit	840	1260.0	1680	12180.0
5	105	30	5.4	44	8450	4.8	Jaigao	1.64592	3.29184	13.36638	UnderWeight	845	1267.5	1690	12252.5
6	106	22	4.9	56	8100	9.4	Pune	1.49352	2.98704	18.74766	Fit	810	1215.0	1620	11745.0
7	107	26	6.2	51	9500	3.5	Jaigao	1.88976	3.77952	13.49378	UnderWeight	950	1425.0	1900	13775.0
8	108	24	4.6	55	15000	8.4	Satara	1.40208	2.80416	19.61372	Fit	1500	2250.0	3000	21750.0
9	109	28	5.1	69	18450	9.0	Sangli	1.55448	3.10896	22.19392	Fit	1845	2767.5	3690	26752.5
10	110	29	5.9	89	7400	4.6	Satara	1.79832	3.59664	24.74532	Fit	740	1110.0	1480	10730.0
11	111	48	5.4	95	25000	8.2	Pune	1.64592	3.29184	28.85924	Fit	2500	3750.0	5000	36250.0
12	112	42	4.3	40	22000	9.4	Kolhapur	1.31064	2.62128	15.25972	UnderWeight	2200	3300.0	4400	31900.0
13	113	49	4.9	77	9000	1.5	Nashik	1.49352	2.98704	25.77803	Fit	900	1350.0	1800	13050.0
14	114	32	5.8	70	9400	1.6	Jaigao	1.76784	3.53568	19.79817	Fit	940	1410.0	1880	13630.0
15	115	33	6.8	74	12400	2.7	Pune	2.07264	4.14528	17.85163	UnderWeight	1240	1860.0	2480	17980.0
16	116	46	6.2	60	45000	4.8	Satara	1.88976	3.77952	15.87503	UnderWeight	4500	6750.0	9000	65250.0
17	117	36	4.3	48	43000	8.7	Pune	1.31064	2.62128	18.31166	UnderWeight	4300	6450.0	8600	62350.0
18	118	35	5.9	59	17500	9.4	Nashik	1.79832	3.59664	16.40420	UnderWeight	1750	2625.0	3500	25375.0
19	119	32	5.5	66	18600	10.5	Kolhapur	1.67640	3.35280	19.68504	Fit	1860	2790.0	3720	26970.0
20	120	27	4.5	80	17800	11.4	Pune	1.37160	2.74320	29.16302	OverWeight	1780	2670.0	3560	25810.0

Conclusion –

We can calculate the different formulas like gross easily using the R contents.

7. Display employee details whose gross salary is >20000.

How to extract data element value from data frame as per the condition using relational greater than operator '>' and how to store a result in newly created subset?

1. **Relational > operator:** - A logical > operator used when you want to display values greater than from particular value.
2. We can use subset () function to perform this kind of basic operations to get result in another subset of data.
3. For run all commands of script, select commands and press **ctrl+enter**.

Introduction:

The purpose of this experiment is to understand the concept of how to use relational greater than operator to satisfy the condition and store the result in subset.

Procedure:-

1. Open R studio open Excercise_1.r script.
2. Take a new variable as salary and create a subset of Employee_Information whose Gross_Salary >20000.
3. Print salary variable to get desire a result
4. Run all commands by pressing ctrl+enter .
5. Save the script using save option from file.

Code And Result :-

Take a new variable as salary and create a subset of Employee_Information whose Gross_Salary >20000.

```
#Take a new variable as salary and create a subset of Employee_Information whose Gross_Salary >20000.
```

```
salary<-subset(Employee_Information,Gross_Salary>20000)
```

Print salary variable to get desire a result

```
#Print salary variable to get desire a result
```

```
print(salary)
```

Save the script Excercise_1.r on using save option..

```
Save script with name Excercise_1.r
```

Output –

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Source
Console Terminal x Jobs x
~/
> #Experiment 7
> #1.open R studio to open a script
> #1.1 open R studio and go to file option
> #1.2 after selecting file option and click open script and after selecting open script option another window is appeared on screen in that you have to
elect script from source and click to open button.
> #1.3 the R_code_1.r R script in opened on screen.
> #Take a new variable as salary and create a subset of Employee_Information whose Gross_salary >20000.
> salary<-subset(Employee_Information,Gross_Salary>20000)
> #Print salary variable to get desire a result
> print(salary)
  EmpNo Age Height Weight BasicSalary Experience Location Height_in_meter Height_sqr BMI_ratio Result_of_BMI TA DA HRA Gross_salary
8 108 24 4.6 55 15000 8.4 Satara 1.40208 2.80416 19.61372 Fit 1500 2250.0 3000 21750.0
9 109 28 5.1 69 18450 9.0 Sangli 1.55448 3.10896 22.19392 Fit 1845 2767.5 3690 26752.5
11 111 48 5.4 95 25000 8.2 Pune 1.64592 3.29184 28.85924 Fit 2500 3750.0 5000 36250.0
12 112 42 4.3 40 22000 9.4 kolhapur 1.31064 2.62128 15.25972 Underweight 2200 3300.0 4400 31900.0
16 116 46 6.2 60 45000 4.8 Satara 1.88976 3.77952 15.87503 Underweight 4500 6750.0 9000 65250.0
17 117 36 4.3 48 43000 8.7 Pune 1.31064 2.62128 18.31166 Underweight 4300 6450.0 8600 62350.0
18 118 35 5.9 59 17500 9.4 Nashik 1.79832 3.59664 16.40420 Underweight 1750 2625.0 3500 25375.0
19 119 32 5.5 66 18600 10.5 kolhapur 1.67640 3.35280 19.68504 Fit 1860 2790.0 3720 26970.0
20 120 27 4.5 80 17800 11.4 Pune 1.37160 2.74320 29.16302 Overweight 1780 2670.0 3560 25810.0
> |

```

Conclusion –

We can extract the values of element by satisfying conditions with help of relational operators and store their result in newly created subset.

8. Display employee details who's having more than 5 years' experience.

How to extract data element value from data frame as per the condition using relational greater than operator '>' and how to store a result in newly created subset?

4. **Relational > operator:** - A relational> operator used when you want to display values greater than from particular value.
5. We can use subset () function to perform this kind of basic operations to get result in another subset of data.
6. For run all commands of script, select commands and press **ctrl+enter**.

Introduction:

The purpose of this experiment is to understand the concept of how to use relational greater than operator to satisfy the condition and store the result in subset.

Procedure:-

1. Open R studio open Excercise_1.r script.
2. Take a new variable as experience and create a subset of Employee_Information whose Experience>5.
3. Print experience variable to get desire a result
4. Run all commands by pressing ctrl+enter.
5. Save the script using save option from file.

Code and Result:-

Open R studio and open Excercise_1.r script

#1.open R studio to open a script

Take a new variable as experience and create a subset of Employee_Information whose Experience >5.

Take a new variable as experience and create a subset of Employee_Information whose Experience >5.

```
experience<-subset(Employee_Information,Experience>5)
```

Print experience variable to get desire a result

#Print experience variable to get desire a result

```
print(experience)
```

Save the script Excercise_1.r on using save option.

Save script with name Excercise_1.r

Output –

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Source
Console Terminal Jobs
~/
> #Experiment 8
> #1.open R studio to open a script
> #1.1 open R studio and go to file option
> #1.2 after selecting file option and click open script and after selecting open script option another window is appeared on screen in that you have to select script from source and click to open button.
> #1.3 the R_code_1.r R script is opened on screen.
> #Take a new variable as experience and create a subset of Employee_Information whose Experience >5.
> experience<-subset(Employee_Information,Employee_Information$Experience>5)
> #Print salary variable to get desire a result
> print(experience)
  EmpNo Age Height weight BasicSalary Experience Location Height_in_meter Height_sqr BMI_ratio Result_of_BMI TA DA HRA Gross_Salary
1 101 45 6.2 67 10000 6.3 Sangli 1.88976 3.77952 17.72712 Underweight 1000 1500.0 2000 14500.0
2 102 23 4.7 89 6700 7.2 Satara 1.43256 2.86512 31.06327 overweight 670 1005.0 1340 9715.0
4 104 31 5.5 68 8400 7.5 kolhapur 1.67640 3.35280 20.28156 Fit 840 1260.0 1680 12180.0
6 106 22 4.9 56 8100 9.4 Pune 1.49352 2.98704 18.74766 Fit 810 1215.0 1620 11745.0
8 108 24 4.6 55 15000 8.4 Satara 1.40208 2.80416 19.61372 Fit 1500 2250.0 3000 21750.0
9 109 28 5.1 69 18450 9.0 Sangli 1.55448 3.10896 22.19392 Fit 1845 2767.5 3690 26752.5
11 111 48 5.4 95 25000 8.2 Pune 1.64592 3.29184 28.85924 Fit 2500 3750.0 5000 36250.0
12 112 42 4.3 40 22000 9.4 kolhapur 1.31064 2.62128 15.25972 Underweight 2200 3300.0 4400 31900.0
17 117 36 4.3 48 43000 8.7 Pune 1.31064 2.62128 18.31166 Underweight 4300 6450.0 8600 62350.0
18 118 35 5.9 59 17500 9.4 Nashik 1.79832 3.59664 16.40420 Underweight 1750 2625.0 3500 25375.0
19 119 32 5.5 66 18600 10.5 kolhapur 1.67640 3.35280 19.68504 Fit 1860 2790.0 3720 26970.0
20 120 27 4.5 80 17800 11.4 Pune 1.37160 2.74320 29.16302 overweight 1780 2670.0 3560 25810.0
> |
```

Conclusion –

We can extract the values of element by satisfying conditions with help of relational operators and store their result in newly created subset.

9. Display employee details whose location is pune.

How to extract data element value from data frame as per the condition using relational exact equal to operator ‘==’ and how to store a result in newly created subset?

1. **Relational == operator:** - A ‘==’ exact equals to operator is used when display result when resulted values are exactly equals to expected values.
2. We can use subset () function to perform this kind of basic operations to get result in another subset of data.
3. For run all commands of script, select commands and press **ctrl+enter**.

Introduction:

The purpose of this experiment is to understand the concept to use of relational operator exact equals to operator by for satisfying condition and display results in form of subset.

Procedure:-

1. Open R studio open Excercise_1.r script.
2. Take a new variable as loc and create a subset of Employee_Information whose Location=="Pune".
3. Print loc variable to get desire a result
4. Run all commands by pressing ctrl+enter.
5. Save the script using save option from file.

Code and Result:-

Take a new variable as loc and create a subset of Employee_Information whose Location==Pune.

Take a new variable as loc and create a subset of Employee_Information whose Location=="Pune".

```
loc<-subset(Employee_Information,Location=="Pune")
```

Print loc variable to get desire result

#Print loc variable to get desire result

```
print(loc)
```

Save the script Excercise_1.r on using save option.

Save script with name Excercise_1.r

Output –

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Source
Console Terminal x Jobs x
~/
> #Experiment 9
> #1.open R studio to open a script
> #1.1 open R studio and go to file option
> #1.2 after selecting file option and click open script and after selecting open script option another window is appeared on screen in that you have to s
elect script from source and click to open button.
> #1.3 the R_code_1.r R script in opened on screen.
> #Take a new variable as loc and create a subset of Employee_Information whose Location==Pune.
> loc<-subset(Employee_Information,Location=="Pune")
> #Print loc variable to get desire a result
> print(loc)
  EmpNo Age Height Weight BasicSalary Experience Location Height_in_meter Height_sqr BMI_ratio Result_of_BMI TA DA HRA Gross_salary
6    106  22   4.9   56         8100         9.4    Pune         1.49352  2.98704  18.74766          Fit  810 1215 1620    11745
11   111  48   5.4   95        25000         8.2    Pune         1.64592  3.29184  28.85924          Fit  2500 3750 5000    36250
15   115  33   6.8   74        12400         2.7    Pune         2.07264  4.14528  17.85163 Underweight 1240 1860 2480    17980
17   117  36   4.3   48        43000         8.7    Pune         1.31064  2.62128  18.31166 Underweight 4300 6450 8600    62350
20   120  27   4.5   80        17800        11.4    Pune         1.37160  2.74320  29.16302  overweight 1780 2670 3560    25810
  
```

Conclusion –

We can extract the values of element by satisfying conditions with help of relational operators exact equals to ‘==’ and store their result in newly created subset. In case of checking condition using operator by passing condition as a string value then the string value must be written in double quotes “”.

10. Display employee details whose location other than Nashik.

How to extract data element value from data frame as per the condition using relational operator not equal to operator '!=' and how to store a result in newly created subset?

4. **Relational!= operator :-** A '==' not equals to operator is used when display result when resulted values are not equals to expected values.
5. We can use subset () function to perform this kind of basic operations to get result in another subset of data.
6. For run all commands of script, select commands and press **ctrl+enter**.

Introduction:

The purpose of this experiment is to understand the concept to use of relational operator not equals to operator by for satisfying condition and display results in form of subset.

Procedure:-

1. Open R studio open Excercise_1.r script.
2. Take a new variable as other and create a subset of Employee_Information whose Location!="Nashik".
3. Print other variable to get desire a result
4. Run all commands by pressing ctrl+enter.
5. Save the script using save option from file.

Code and Result:-

Open R studio and open Excercise_1.r script

- #1.open R studio to open a script
- #1.1 open R studio and go to file option
- #1.2 after selecting file option and click open script and after selecting open script option another window is appeared on screen in that you have to select script from source and click to open button.
- #1.3 the Excercise_1.r R script in opened on screen.

Take a new variable as other and create a subset of Employee_Information whose Location!="Nashik".

```
# Take a new variable as other and create a subset of Employee_Information whose Location!="Nashik".
```

```
other<-subset(Employee_Information,Location!="Nashik")
```

Print other variable to get desire result

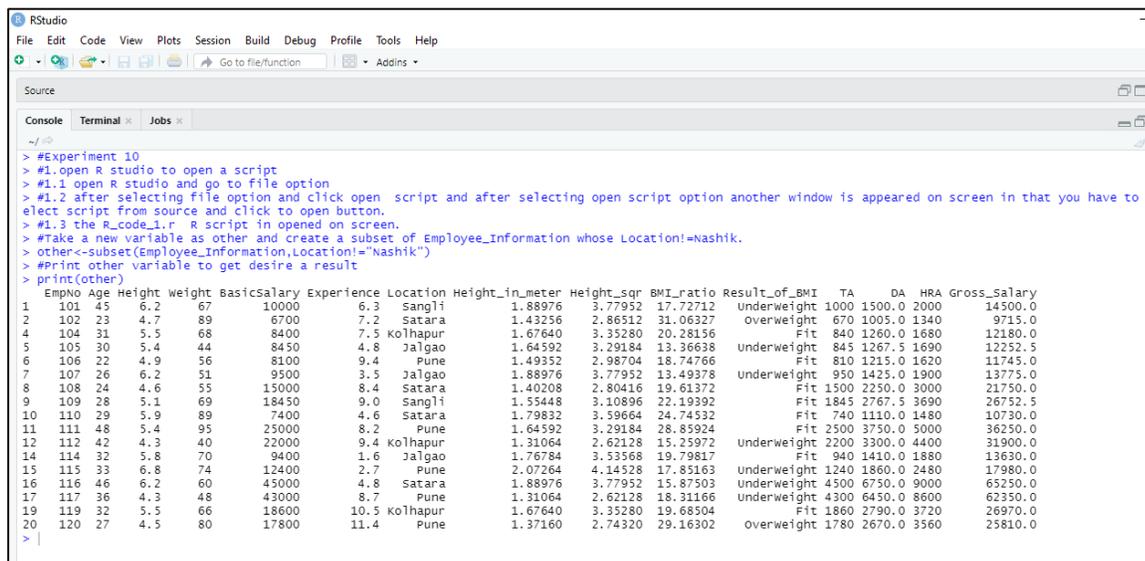
#Print other variable to get desire result

print(other)

Save the script Excersice_1.r on using save option..

Save script with name Excersice_1.r

Output –



```
> #Experiment 10
> #1. open R studio to open a script
> #1.1 open R studio and go to file option
> #1.2 after selecting file option and click open script and after selecting open script option another window is appeared on screen in that you have to select script from source and click to open button.
> #1.3 the R_code_1.r R script is opened on screen.
> #Take a new variable as other and create a subset of Employee_Information whose Location!=Nashik.
> other<-subset(Employee_Information,Location!="Nashik")
> #Print other variable to get desire a result
> print(other)
```

EmpNo	Age	Height	Weight	BasicSalary	Experience	Location	Height_in_meter	Height_sqr	BMI_ratio	Result_of_BMI	TA	DA	HRA	Gross_salary	
1	101	45	6.2	67	10000	6.3	Sangli	1.88976	3.77952	17.72712	underweight	1000	1500.0	2000	14500.0
2	102	23	4.7	89	6700	7.2	Satara	1.43256	2.86512	31.06327	overweight	670	1005.0	1340	9715.0
4	104	31	5.5	68	8400	7.5	Kolhapur	1.67640	3.35280	20.28156	Fit	840	1260.0	1680	12180.0
5	105	30	5.4	44	8450	4.8	Jalgaon	1.64592	3.29184	13.36638	underweight	845	1267.5	1690	12252.5
6	106	22	4.9	36	8100	9.4	Pune	1.49352	2.98704	18.74766	Fit	810	1215.0	1620	11745.0
7	107	26	6.2	51	9500	3.5	Jalgaon	1.88976	3.77952	13.49378	underweight	950	1425.0	1900	13775.0
8	108	24	4.6	55	15000	8.4	Satara	1.40208	2.80416	19.61372	Fit	1500	2250.0	3000	21750.0
9	109	28	5.1	69	18450	9.0	Sangli	1.55448	3.10896	22.19392	Fit	1845	2767.5	3690	26752.5
10	110	29	5.9	89	7400	4.6	Satara	1.79832	3.59664	24.74532	Fit	740	1110.0	1480	10730.0
11	111	48	5.4	95	25000	8.2	Pune	1.64592	3.29184	28.85924	Fit	2500	3750.0	5000	36250.0
12	112	42	4.3	40	22000	9.4	Kolhapur	1.31064	2.62128	15.25972	underweight	2200	3300.0	4400	31900.0
14	114	32	5.8	70	9400	1.6	Jalgaon	1.76784	3.53568	19.79817	Fit	940	1410.0	1880	13630.0
15	115	33	6.8	74	12400	2.7	Pune	2.07264	4.14528	17.85163	underweight	1240	1860.0	2480	17980.0
16	116	46	6.2	60	45000	4.8	Satara	1.88976	3.77952	15.87503	underweight	4500	6750.0	9000	65250.0
17	117	36	4.3	48	43000	8.7	Pune	1.31064	2.62128	18.31166	underweight	4300	6450.0	8600	62350.0
19	119	32	5.5	66	18600	10.5	Kolhapur	1.67640	3.35280	19.68504	Fit	1860	2790.0	3720	26970.0
20	120	27	4.5	80	17800	11.4	Pune	1.37160	2.74320	29.16302	Overweight	1780	2670.0	3560	25810.0

Conclusion –

We can use the relational operator “!=” not equals to for excluding values in given condition. In case of checking condition using != operator and the condition contains the string value then the string value must be written in double quotes .

XXXXXXXXXXXXXXXXXX

