Midterm Project

Introduction To Data Science

Name: Shakib Sadat Shanto

Id: 20-43074-1

Section: D

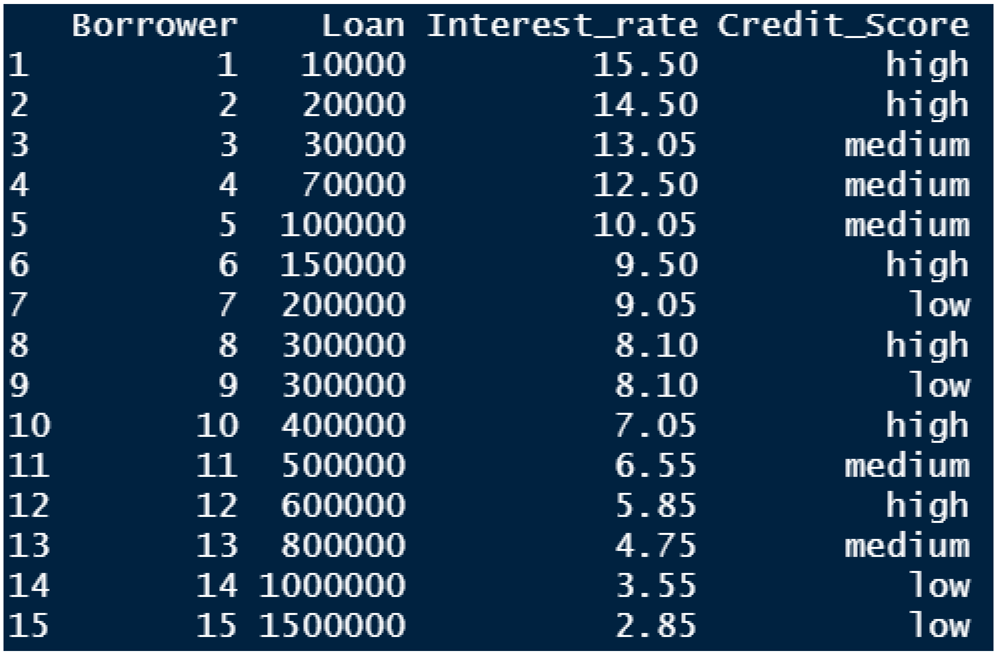
Tasks for data_1:

•Import the data set(data_1) as csv and print the data set:

Code:

```
dataset = read.csv('Dataset_1.csv')
dataset
```

Output:



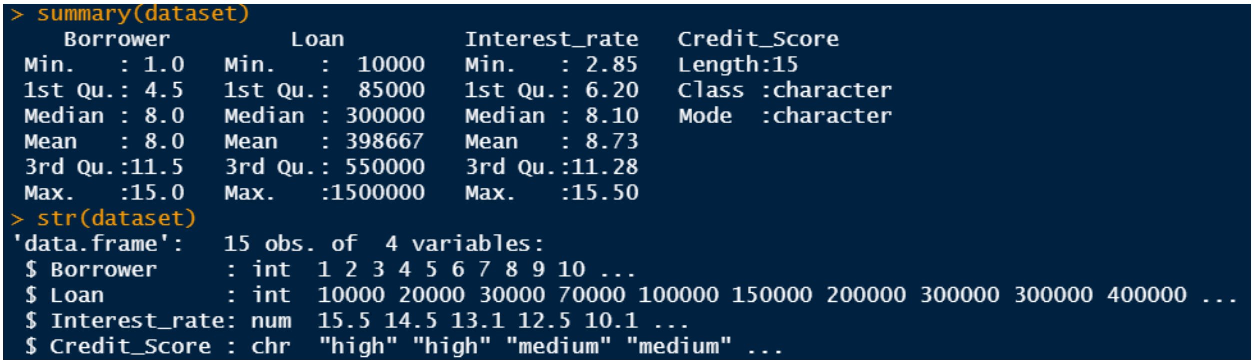
```
   Borrower    Loan Interest_rate Credit_Score
1         1   10000         15.50         high
2         2   20000         14.50         high
3         3   30000         13.05       medium
4         4   70000         12.50       medium
5         5  100000         10.05       medium
6         6  150000          9.50         high
7         7  200000          9.05          low
8         8  300000          8.10         high
9         9  300000          8.10          low
10       10  400000          7.05         high
11       11  500000          6.55       medium
12       12  600000          5.85         high
13       13  800000          4.75       medium
14       14 1000000          3.55          low
15       15 1500000          2.85          low
```

•Find the shape of the data set:

Code:

```
summary(dataset)
str(dataset)
```

Output:



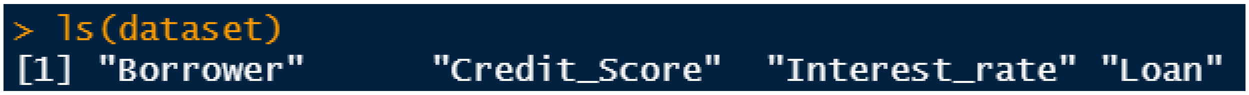
```
> summary(dataset)
    Borrower          Loan           Interest_rate   Credit_Score
 Min.   : 1.0   Min.   :  10000   Min.   : 2.85   Length:15
 1st Qu.: 4.5   1st Qu.:  85000   1st Qu.: 6.20   Class :character
 Median : 8.0   Median : 300000   Median : 8.10   Mode  :character
 Mean   : 8.0   Mean   : 398667   Mean   : 8.73
 3rd Qu.:11.5   3rd Qu.: 550000   3rd Qu.:11.28
 Max.   :15.0   Max.   :1500000   Max.   :15.50
> str(dataset)
'data.frame':   15 obs. of  4 variables:
 $ Borrower     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Loan         : int  10000 20000 30000 70000 100000 150000 200000 300000 300000 400000 ...
 $ Interest_rate: num  15.5 14.5 13.1 12.5 10.1 ...
 $ Credit_Score : chr  "high" "high" "medium" "medium" ...
```

•Show the attributes name of the data set:

Code:

ls(dataset)

Output:



```
> ls(dataset)
[1] "Borrower"      "Credit_Score"   "Interest_rate" "Loan"
```

•Find the types of data for all attributes:

Code:

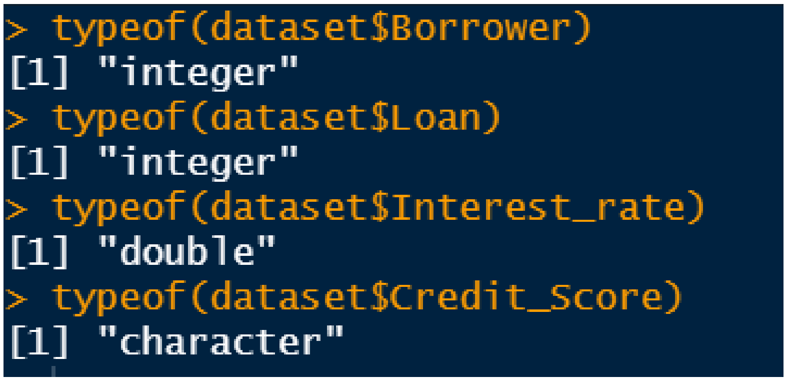
typeof(dataset$Borrower)

typeof(dataset$Loan)

typeof(dataset$Interest_rate)

typeof(dataset$Credit_Score)

Output:



```
> typeof(dataset$Borrower)
[1] "integer"
> typeof(dataset$Loan)
[1] "integer"
> typeof(dataset$Interest_rate)
[1] "double"
> typeof(dataset$Credit_Score)
[1] "character"
```

•Measure of center (mean, median and mode) for Loan and Interest_rate attributes:

Code:

install.packages('dplyr')

library(dplyr)

dataset[,2:3] %>% summarise_if(is.numeric, mean)

dataset[,2:3] %>% summarise_if(is.numeric, median)

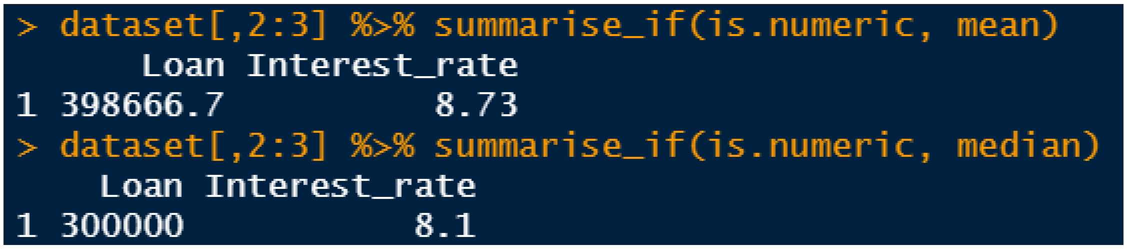
install.packages("DescTools")

library(DescTools)

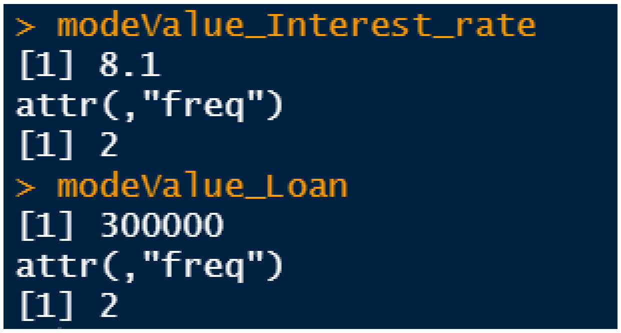
modeValue_Loan <- Mode(dataset$Loan)

modeValue_Interest_rate <- Mode(dataset$Interest_rate)

Output:



```
> dataset[,2:3] %>% summarise_if(is.numeric, mean)
      Loan Interest_rate
1 398666.7          8.73
> dataset[,2:3] %>% summarise_if(is.numeric, median)
    Loan Interest_rate
1 300000           8.1
```



```
> modeValue_Interest_rate
[1] 8.1
attr(,"freq")
[1] 2
> modeValue_Loan
[1] 300000
attr(,"freq")
[1] 2
```

•Measure of Spread (range and standard Deviation) for Loan and Interest_rate attributes:

Code:

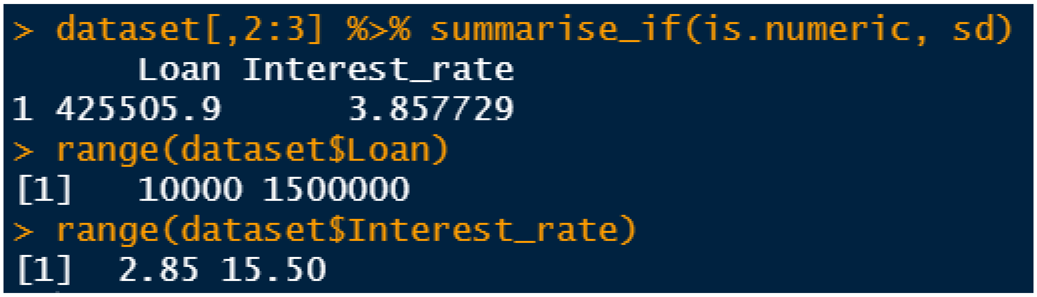
library(dplyr)

dataset[,2:3] %>% summarise_if(is.numeric, sd)

range(dataset$Loan)

range(dataset$Interest_rate)

Output:



```
> dataset[,2:3] %>% summarise_if(is.numeric, sd)
      Loan Interest_rate
1 425505.9      3.857729
> range(dataset$Loan)
[1]   10000 1500000
> range(dataset$Interest_rate)
[1]  2.85 15.50
```

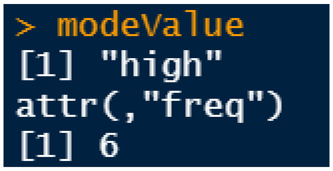
•Find the mode for Credit_Score attribute:

Code:

library("DescTools")

modeValue <- Mode(dataset$Credit_Score)

Output:



```
> modeValue
[1] "high"
attr(,"freq")
[1] 6
```

Tasks for data_2:

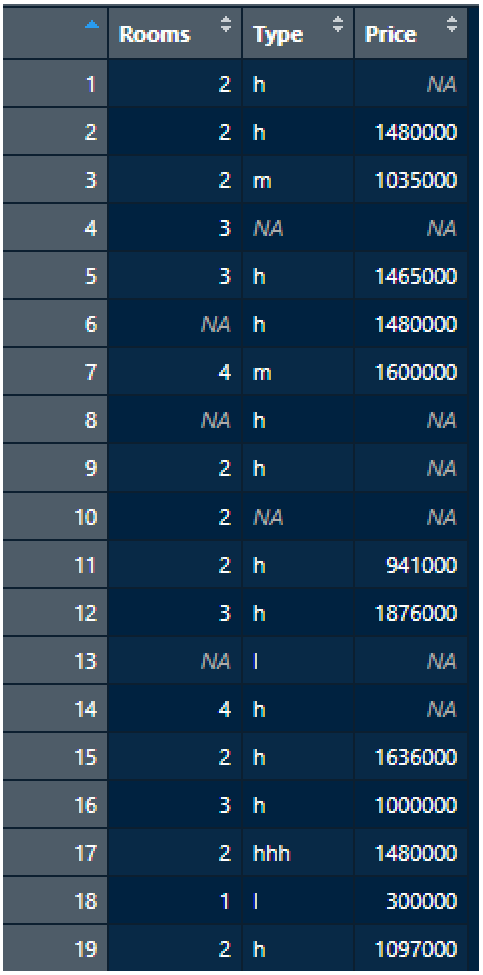
•Import the data set 2(data_2) as csv and print the data set:

•Find the missing value for all attributes:

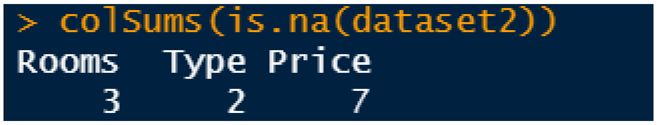
Code:

```
dataset2$Type[dataset2$Type==""] <- NA
colSums(is.na(dataset2))
```

Output:



	Rooms	Type	Price
1	2	h	NA
2	2	h	1480000
3	2	m	1035000
4	3	NA	NA
5	3	h	1465000
6	NA	h	1480000
7	4	m	1600000
8	NA	h	NA
9	2	h	NA
10	2	NA	NA
11	2	h	941000
12	3	h	1876000
13	NA	l	NA
14	4	h	NA
15	2	h	1636000
16	3	h	1000000
17	2	hhh	1480000
18	1	l	300000
19	2	h	1097000



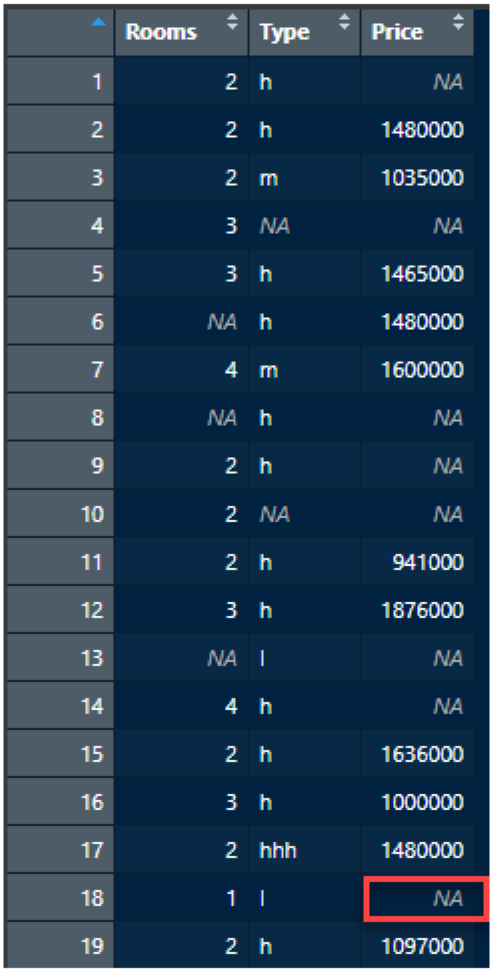
```
> colSums(is.na(dataset2))
Rooms  Type Price 
    3     2     7 
```

•Detect the outlier as a missing value:

Code:

```
dataset2$Price[dataset2$Price<=300000] <- NA
```

Output:



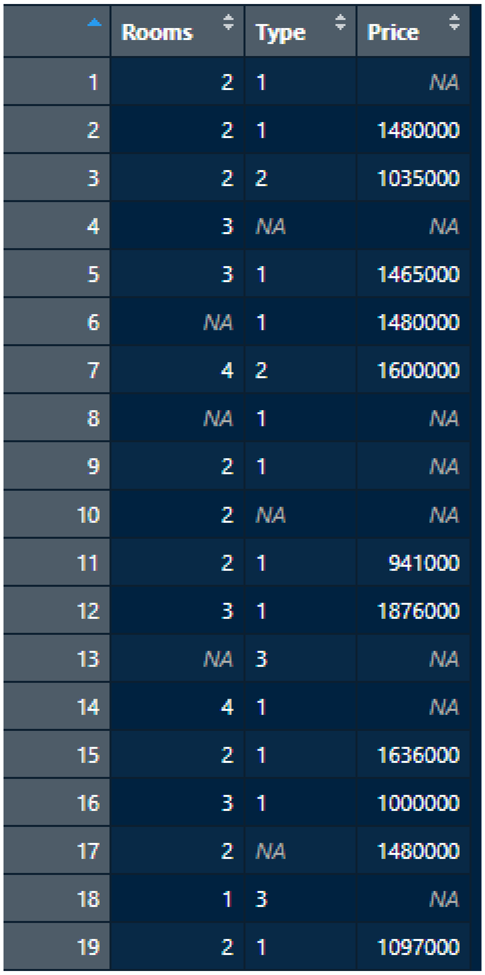
	Rooms	Type	Price
1	2	h	NA
2	2	h	1480000
3	2	m	1035000
4	3	NA	NA
5	3	h	1465000
6	NA	h	1480000
7	4	m	1600000
8	NA	h	NA
9	2	h	NA
10	2	NA	NA
11	2	h	941000
12	3	h	1876000
13	NA	l	NA
14	4	h	NA
15	2	h	1636000
16	3	h	1000000
17	2	hhh	1480000
18	1	l	NA
19	2	h	1097000

•Annotate h as 1, m as 2 , and l as 3 from “Type” attribute:

Code:

```
dataset2$Type = factor(dataset2$Type,
                        levels = c("h","m","l"),
                        labels = c(1,2,3))
```

Output:



	Rooms	Type	Price
1	2	1	NA
2	2	1	1480000
3	2	2	1035000
4	3	NA	NA
5	3	1	1465000
6	NA	1	1480000
7	4	2	1600000
8	NA	1	NA
9	2	1	NA
10	2	NA	NA
11	2	1	941000
12	3	1	1876000
13	NA	3	NA
14	4	1	NA
15	2	1	1636000
16	3	1	1000000
17	2	NA	1480000
18	1	3	NA
19	2	1	1097000

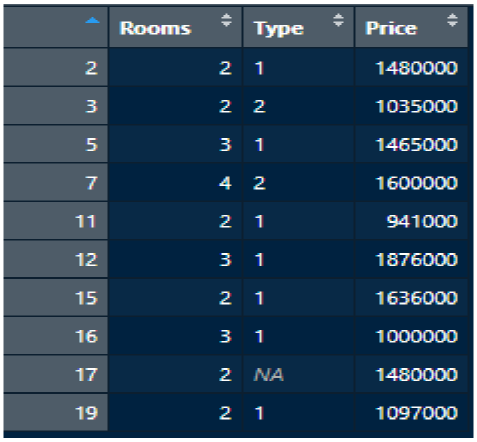
•Recover missing values by the following strategies for Rooms and Price attributes:

I. Delete the rows with missing values:

Code:

```
remove_row =
dataset2[complete.cases(dataset2$Rooms,dataset2$Price), ]
```

Output:



	Rooms	Type	Price
2	2	1	1480000
3	2	2	1035000
5	3	1	1465000
7	4	2	1600000
11	2	1	941000
12	3	1	1876000
15	2	1	1636000
16	3	1	1000000
17	2	NA	1480000
19	2	1	1097000

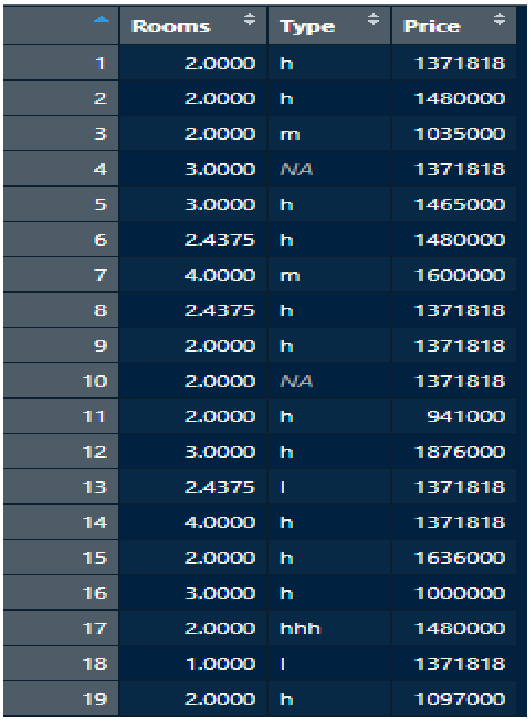
II. Recover missing values with the mean value:

Code:

dataset2$Rooms[is.na(dataset2$Rooms)] = mean(dataset2$Rooms, na.rm = TRUE)

dataset2$Price[is.na(dataset2$Price)] = mean(dataset2$Price, na.rm = TRUE)

Output:



	Rooms	Type	Price
1	2.0000	h	1371818
2	2.0000	h	1480000
3	2.0000	m	1035000
4	3.0000	NA	1371818
5	3.0000	h	1465000
6	2.4375	h	1480000
7	4.0000	m	1600000
8	2.4375	h	1371818
9	2.0000	h	1371818
10	2.0000	NA	1371818
11	2.0000	h	941000
12	3.0000	h	1876000
13	2.4375	l	1371818
14	4.0000	h	1371818
15	2.0000	h	1636000
16	3.0000	h	1000000
17	2.0000	hhh	1480000
18	1.0000	l	1371818
19	2.0000	h	1097000

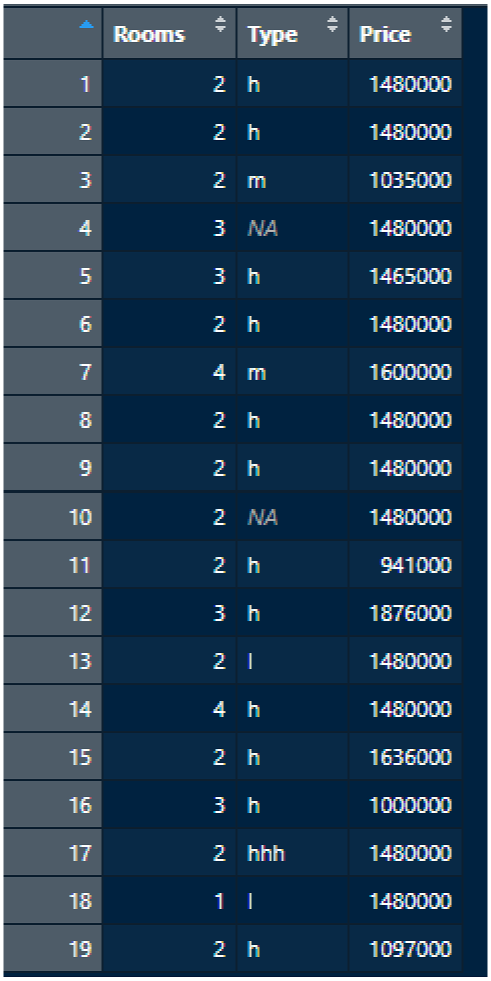
III. Recover missing values with the median value:

Code:

dataset2$Price[is.na(dataset2$Price)] = median(dataset2$Price, na.rm = TRUE)

dataset2$Rooms[is.na(dataset2$Rooms)] = median(dataset2$Rooms, na.rm = TRUE)

Output:



	Rooms	Type	Price
1	2	h	1480000
2	2	h	1480000
3	2	m	1035000
4	3	*NA*	1480000
5	3	h	1465000
6	2	h	1480000
7	4	m	1600000
8	2	h	1480000
9	2	h	1480000
10	2	*NA*	1480000
11	2	h	941000
12	3	h	1876000
13	2	l	1480000
14	4	h	1480000
15	2	h	1636000
16	3	h	1000000
17	2	hhh	1480000
18	1	l	1480000
19	2	h	1097000

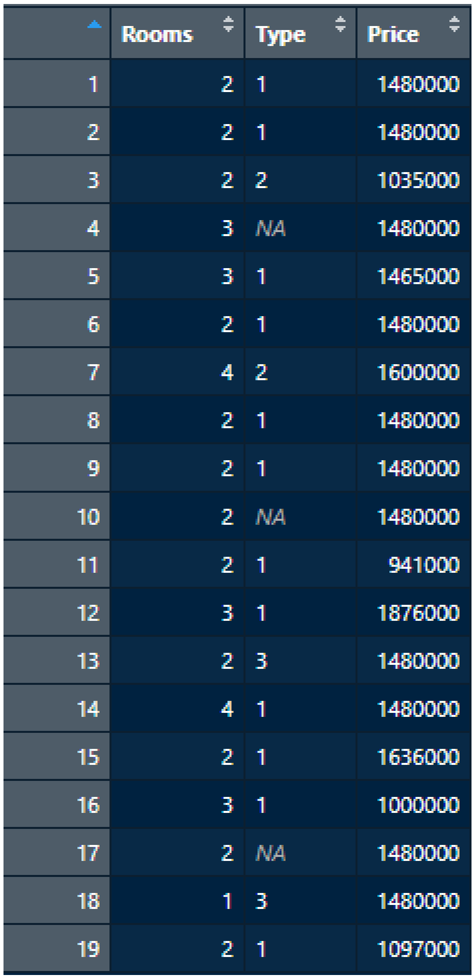
IV. Recover missing values with the mode value:

Code:

```
dataset2$Rooms[is.na(dataset2$Rooms)] <- Mode(dataset2$Rooms, na.rm = TRUE)

dataset2$Price[is.na(dataset2$Price)] <- Mode(dataset2$Price, na.rm = TRUE)
```

Output:



	Rooms	Type	Price
1	2	1	1480000
2	2	1	1480000
3	2	2	1035000
4	3	NA	1480000
5	3	1	1465000
6	2	1	1480000
7	4	2	1600000
8	2	1	1480000
9	2	1	1480000
10	2	NA	1480000
11	2	1	941000
12	3	1	1876000
13	2	3	1480000
14	4	1	1480000
15	2	1	1636000
16	3	1	1000000
17	2	NA	1480000
18	1	3	1480000
19	2	1	1097000