NAME

NOBONITA NONDE

ID

20-43819-2

COURSE

INTRODUCTION TO DATA SCIENCE

SECTION

E

DEPARTMENT

COMUTER SCIENCE AND ENGINEERING

SUBMISSION DATE

13.03.23

Descriptions of the Project:

This dataset contains information on caesarean section outcomes for 80 pregnant women with key features of childbirth problems in the medical field. The dataset uses various columns such as age, number of deliveries, delivery time, blood pressure, and heart condition. Here, delivery times are divided into Premature, Timely and Latecomer. Blood pressure is also classified as low, normal, and high mood. Finally, for heart problems is classified with apt and inept. There are missing values, noisy values, they should be preprocessed, and the dataset should be filled without missing. To get a clean preprocessed dataset which would be free of any discrepancies we might be using some preprocessing techniques which are data cleaning, data transformation, data reduction, data discretization, data integration, clean data, verification.

Data Preprocessing for the data set:

For age and weight we can use the mean value to find out the missing values.

Task 1:

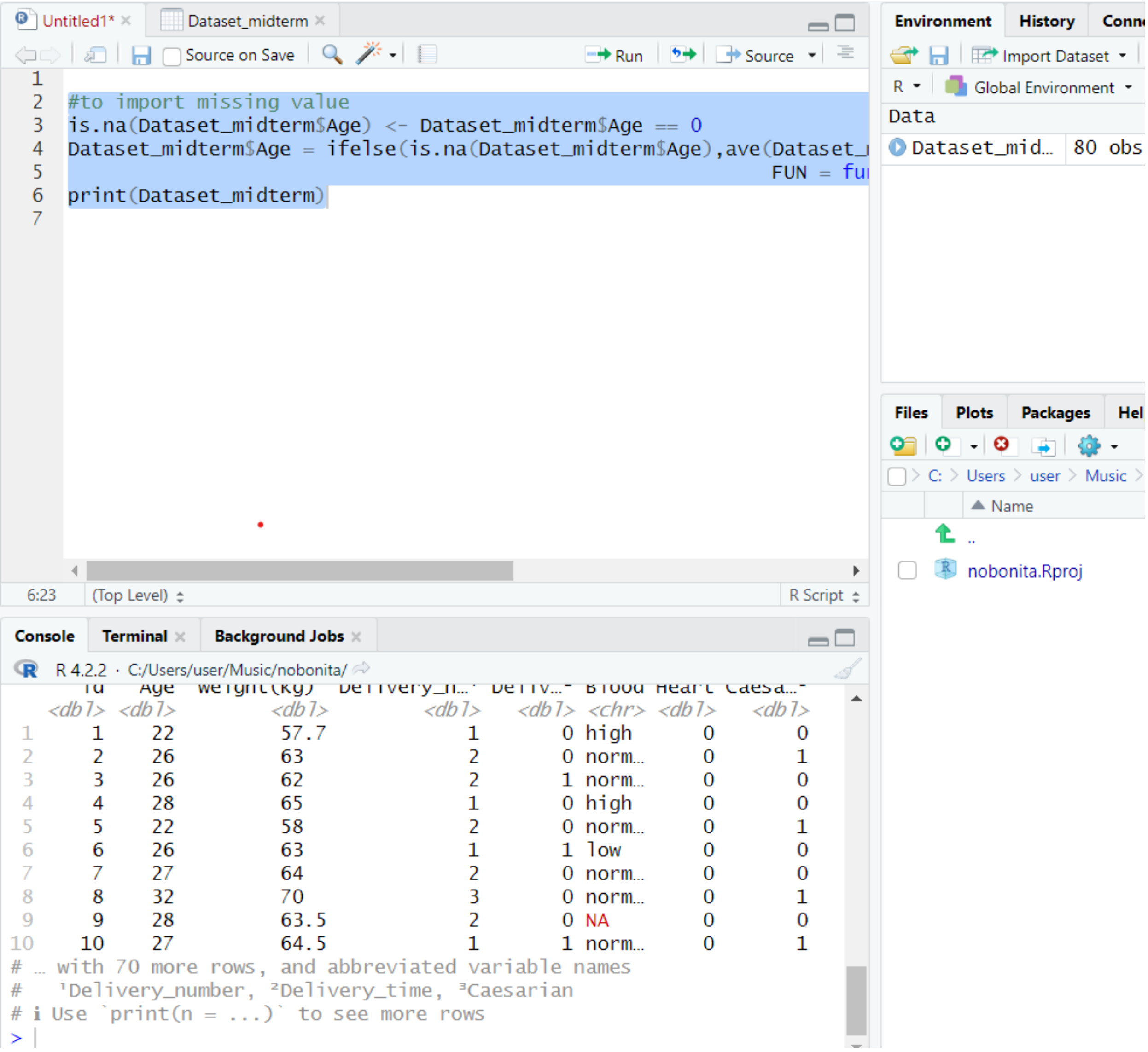
Age:

```
#to import missing value

is.na(Dataset_midterm$Age) <- Dataset_midterm$Age == 0

Dataset_midterm$Age = ifelse(is.na(Dataset_midterm$Age),ave(Dataset_midterm$Age, FUN = function(x)     mean(x, na.rm = TRUE)),Dataset_midterm$Age)

print(Dataset_midterm)
```

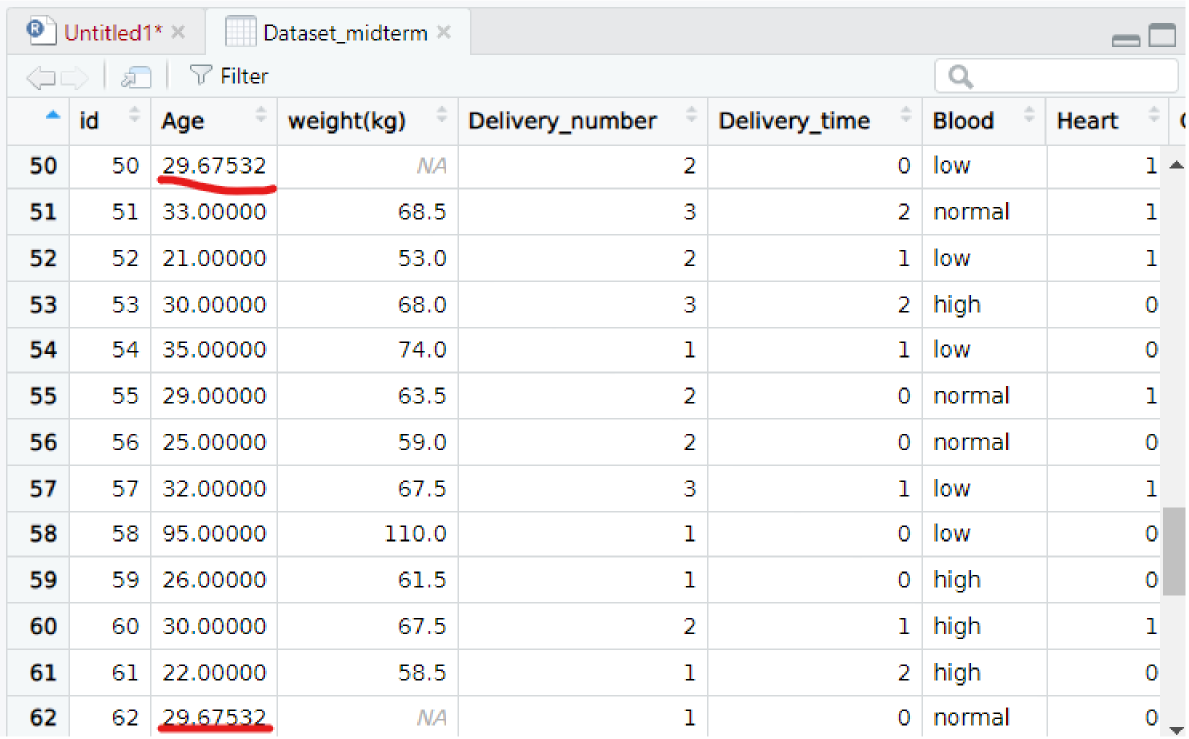


```
#to import missing value
is.na(Dataset_midterm$Age) <- Dataset_midterm$Age == 0
Dataset_midterm$Age = ifelse(is.na(Dataset_midterm$Age),ave(Dataset_
                                                      FUN = fu

print(Dataset_midterm)
```

```
      Id   Age  Weight(kg) Delivery_n…¹ Deliv…² Blood Heart Caesa…³
   <dbl> <dbl>       <dbl>        <dbl>   <dbl> <chr> <dbl>   <dbl>
 1     1    22        57.7            1       0 high      0       0
 2     2    26        63              2       0 norm…     0       1
 3     3    26        62              2       1 norm…     0       0
 4     4    28        65              1       0 high      0       0
 5     5    22        58              2       0 norm…     0       1
 6     6    26        63              1       1 low       0       0
 7     7    27        64              2       0 norm…     0       0
 8     8    32        70              3       0 norm…     0       1
 9     9    28        63.5            2       0 NA        0       0
10    10    27        64.5            1       1 norm…     0       1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
> 
```

Figure 1: Task 1



	id	Age	weight(kg)	Delivery_number	Delivery_time	Blood	Heart
50	50	29.67532	NA	2	0	low	1
51	51	33.00000	68.5	3	2	normal	1
52	52	21.00000	53.0	2	1	low	1
53	53	30.00000	68.0	3	2	high	0
54	54	35.00000	74.0	1	1	low	0
55	55	29.00000	63.5	2	0	normal	1
56	56	25.00000	59.0	2	0	normal	0
57	57	32.00000	67.5	3	1	low	1
58	58	95.00000	110.0	1	0	low	0
59	59	26.00000	61.5	1	0	high	0
60	60	30.00000	67.5	2	1	high	1
61	61	22.00000	58.5	1	2	high	0
62	62	29.67532	NA	1	0	normal	0

Figure 2: task 1

Task 2:

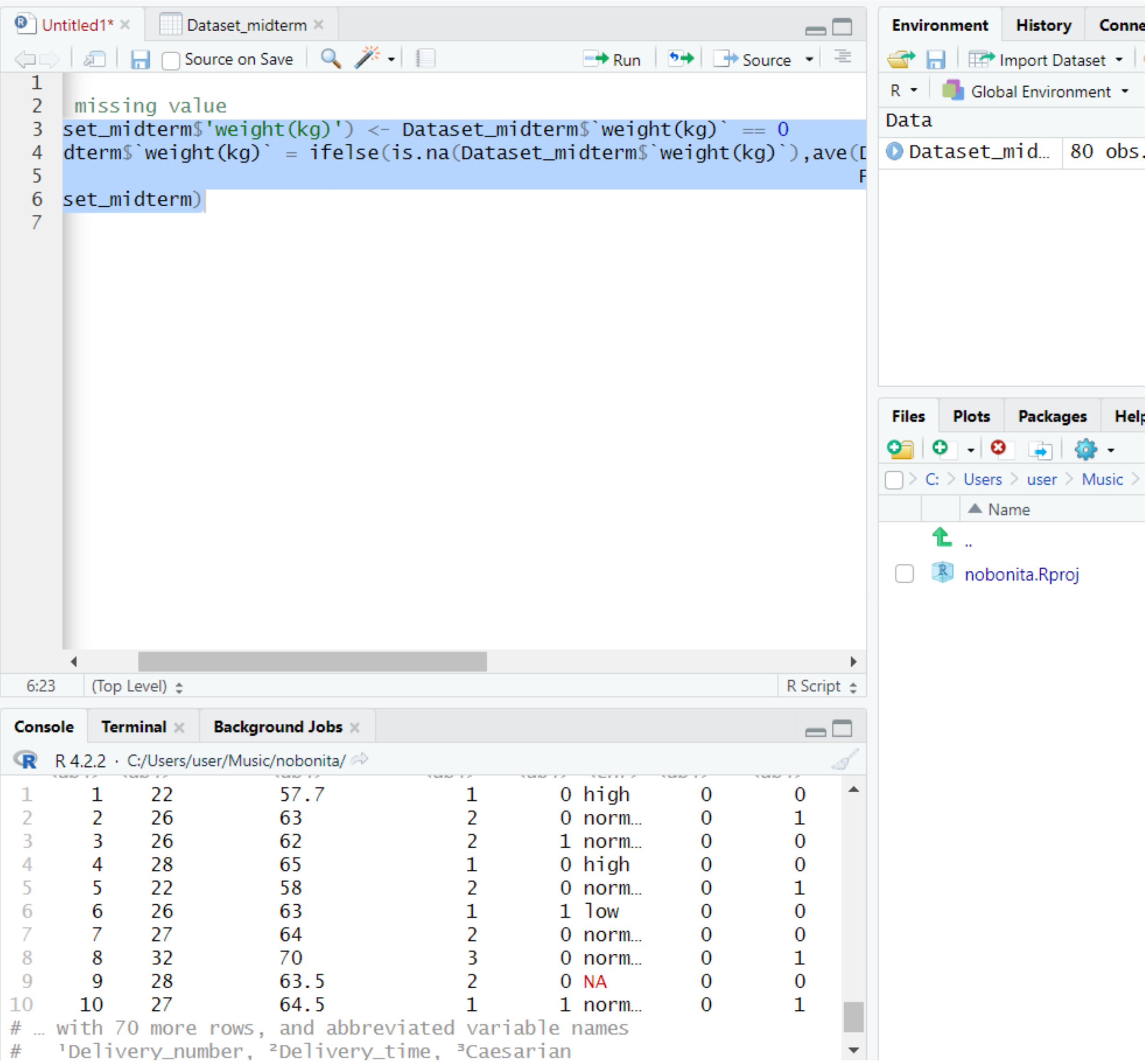
Weight:

```
#to import missing value

is.na(Dataset_midterm$'weight(kg)') <- Dataset_midterm$`weight(kg)` == 0

Dataset_midterm$`weight(kg)` =
ifelse(is.na(Dataset_midterm$`weight(kg)`),ave(Dataset_midterm$`weight(kg)`,
            FUN = function(x) mean(x, na.rm = TRUE)),Dataset_midterm$`weight(kg)`)

print(Dataset_midterm)
```



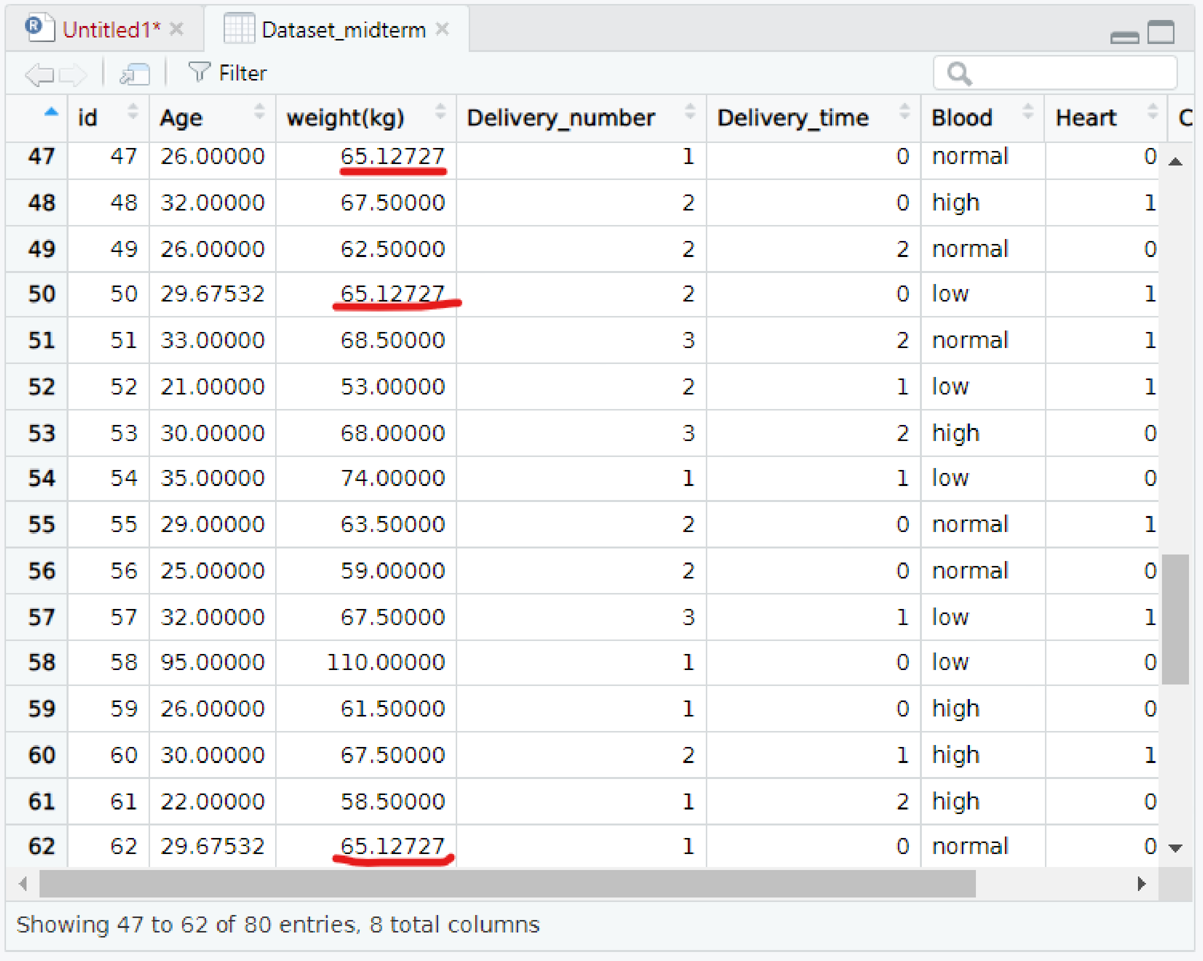
```

 missing value
set_midterm$'weight(kg)') <- Dataset_midterm$`weight(kg)` == 0
dterm$`weight(kg)` = ifelse(is.na(Dataset_midterm$`weight(kg)`),ave(D
                                                                    F
set_midterm)

```

```
 1      1    22          57.7          1        0 high      0       0
 2      2    26          63            2        0 norm…     0       1
 3      3    26          62            2        1 norm…     0       0
 4      4    28          65            1        0 high      0       0
 5      5    22          58            2        0 norm…     0       1
 6      6    26          63            1        1 low       0       0
 7      7    27          64            2        0 norm…     0       0
 8      8    32          70            3        0 norm…     0       1
 9      9    28          63.5          2        0 NA        0       0
10     10    27          64.5          1        1 norm…     0       1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
```

Figure 3: task 2



	id	Age	weight(kg)	Delivery_number	Delivery_time	Blood	Heart	C
47	47	26.00000	65.12727	1	0	normal	0	
48	48	32.00000	67.50000	2	0	high	1	
49	49	26.00000	62.50000	2	2	normal	0	
50	50	29.67532	65.12727	2	0	low	1	
51	51	33.00000	68.50000	3	2	normal	1	
52	52	21.00000	53.00000	2	1	low	1	
53	53	30.00000	68.00000	3	2	high	0	
54	54	35.00000	74.00000	1	1	low	0	
55	55	29.00000	63.50000	2	0	normal	1	
56	56	25.00000	59.00000	2	0	normal	0	
57	57	32.00000	67.50000	3	1	low	1	
58	58	95.00000	110.00000	1	0	low	0	
59	59	26.00000	61.50000	1	0	high	0	
60	60	30.00000	67.50000	2	1	high	1	
61	61	22.00000	58.50000	1	2	high	0	
62	62	29.67532	65.12727	1	0	normal	0	

Showing 47 to 62 of 80 entries, 8 total columns

Figure 4: task 2

There are some **noisy values** in Age and Weight Column. Now we have to remove the **noisy value** from the age and weight column with some mean value.

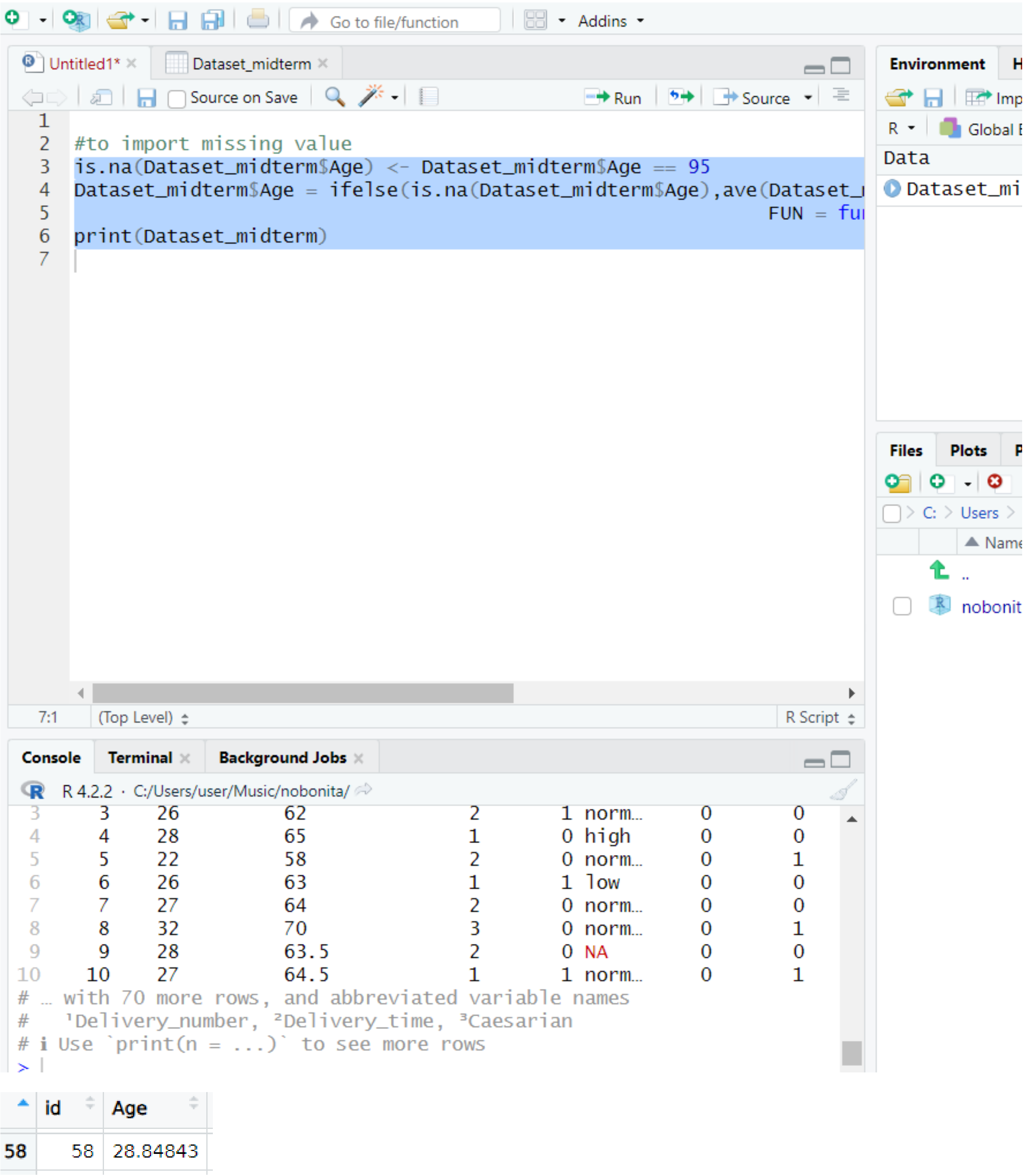
Age:

```
is.na(Dataset_midterm$Age) <- Dataset_midterm$Age == 95

Dataset_midterm$Age = ifelse(is.na(Dataset_midterm$Age),ave(Dataset_midterm$Age,

FUN = function(x) mean(x, na.rm = TRUE)),Dataset_midterm$Age)

print(Dataset_midterm)
```



```r
#to import missing value
is.na(Dataset_midterm$Age) <- Dataset_midterm$Age == 95
Dataset_midterm$Age = ifelse(is.na(Dataset_midterm$Age),ave(Dataset_
                                                        FUN = fu
print(Dataset_midterm)
```

```
 3     3    26        62          2     1 norm…     0      0
 4     4    28        65          1     0 high      0      0
 5     5    22        58          2     0 norm…     0      1
 6     6    26        63          1     1 low       0      0
 7     7    27        64          2     0 norm…     0      0
 8     8    32        70          3     0 norm…     0      1
 9     9    28        63.5        2     0 NA        0      0
10    10    27        64.5        1     1 norm…     0      1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
>
```

	id	Age
58	58	28.84843

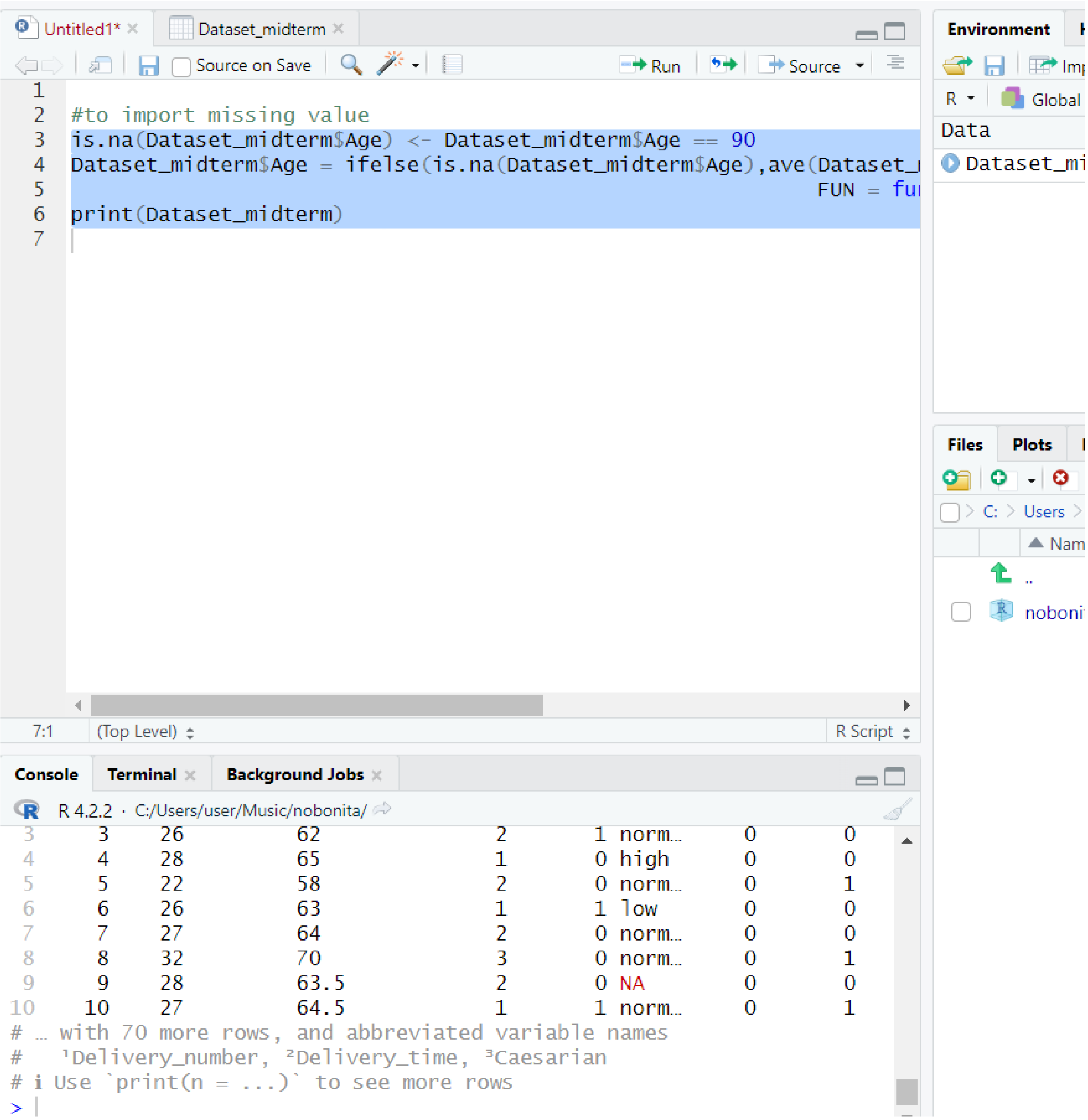
Figure 5: task 3

is.na(Dataset_midterm$Age) <- Dataset_midterm$Age == 90

Dataset_midterm$Age = ifelse(is.na(Dataset_midterm$Age),ave(Dataset_midterm$Age,

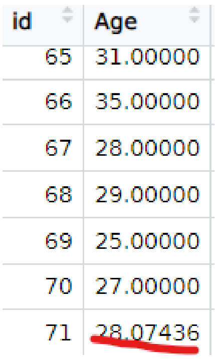
FUN = function(x) mean(x, na.rm = TRUE)),Dataset_midterm$Age)

print(Dataset_midterm)



```
#to import missing value
is.na(Dataset_midterm$Age) <- Dataset_midterm$Age == 90
Dataset_midterm$Age = ifelse(is.na(Dataset_midterm$Age),ave(Dataset_
                                                           FUN = fu
print(Dataset_midterm)
```

```
 3     3  26          62             2        1 norm…     0       0
 4     4  28          65             1        0 high      0       0
 5     5  22          58             2        0 norm…     0       1
 6     6  26          63             1        1 low       0       0
 7     7  27          64             2        0 norm…     0       0
 8     8  32          70             3        0 norm…     0       1
 9     9  28          63.5           2        0 NA        0       0
10    10  27          64.5           1        1 norm…     0       1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
> 
```



id	Age
65	31.00000
66	35.00000
67	28.00000
68	29.00000
69	25.00000
70	27.00000
71	28.07436

Figure 6: task 3

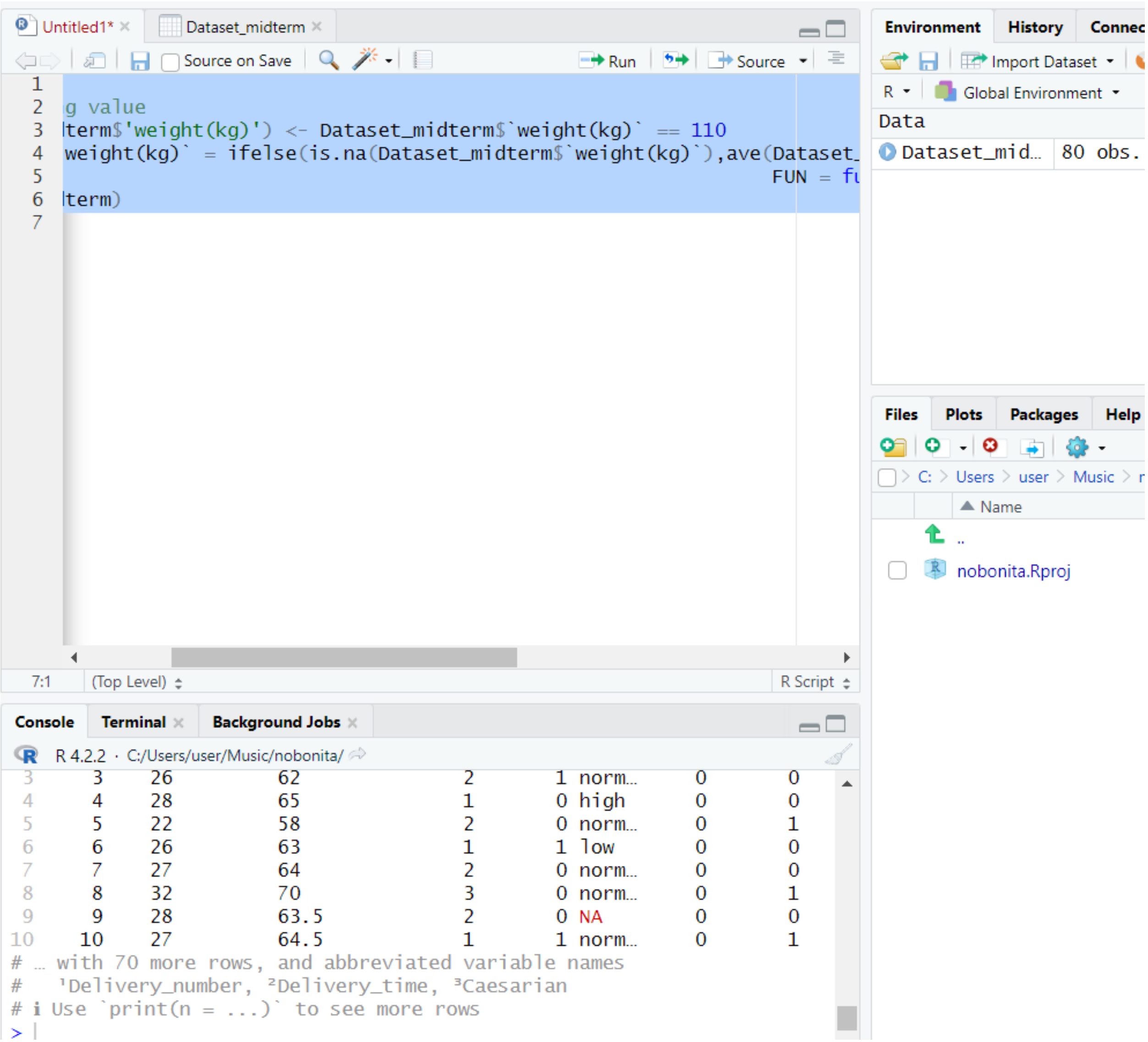
Weight:

```
is.na(Dataset_midterm$'weight(kg)') <- Dataset_midterm$`weight(kg)` == 110

Dataset_midterm$`weight(kg)` =
ifelse(is.na(Dataset_midterm$`weight(kg)`),ave(Dataset_midterm$`weight(kg)`,

                                                     FUN = function(x) mean(x, na.rm =
TRUE)),Dataset_midterm$`weight(kg)`)

print(Dataset_midterm)
```



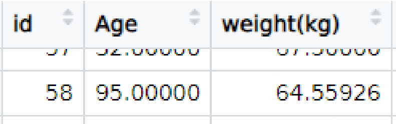
```

g value
lterm$'weight(kg)') <- Dataset_midterm$`weight(kg)` == 110
weight(kg)` = ifelse(is.na(Dataset_midterm$`weight(kg)`),ave(Dataset_
                                                    FUN = fu

lterm)

```

```
 3     3    26          62           2      1 norm…    0      0
 4     4    28          65           1      0 high     0      0
 5     5    22          58           2      0 norm…    0      1
 6     6    26          63           1      1 low      0      0
 7     7    27          64           2      0 norm…    0      0
 8     8    32          70           3      0 norm…    0      1
 9     9    28          63.5         2      0 NA       0      0
10    10    27          64.5         1      1 norm…    0      1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
> |
```



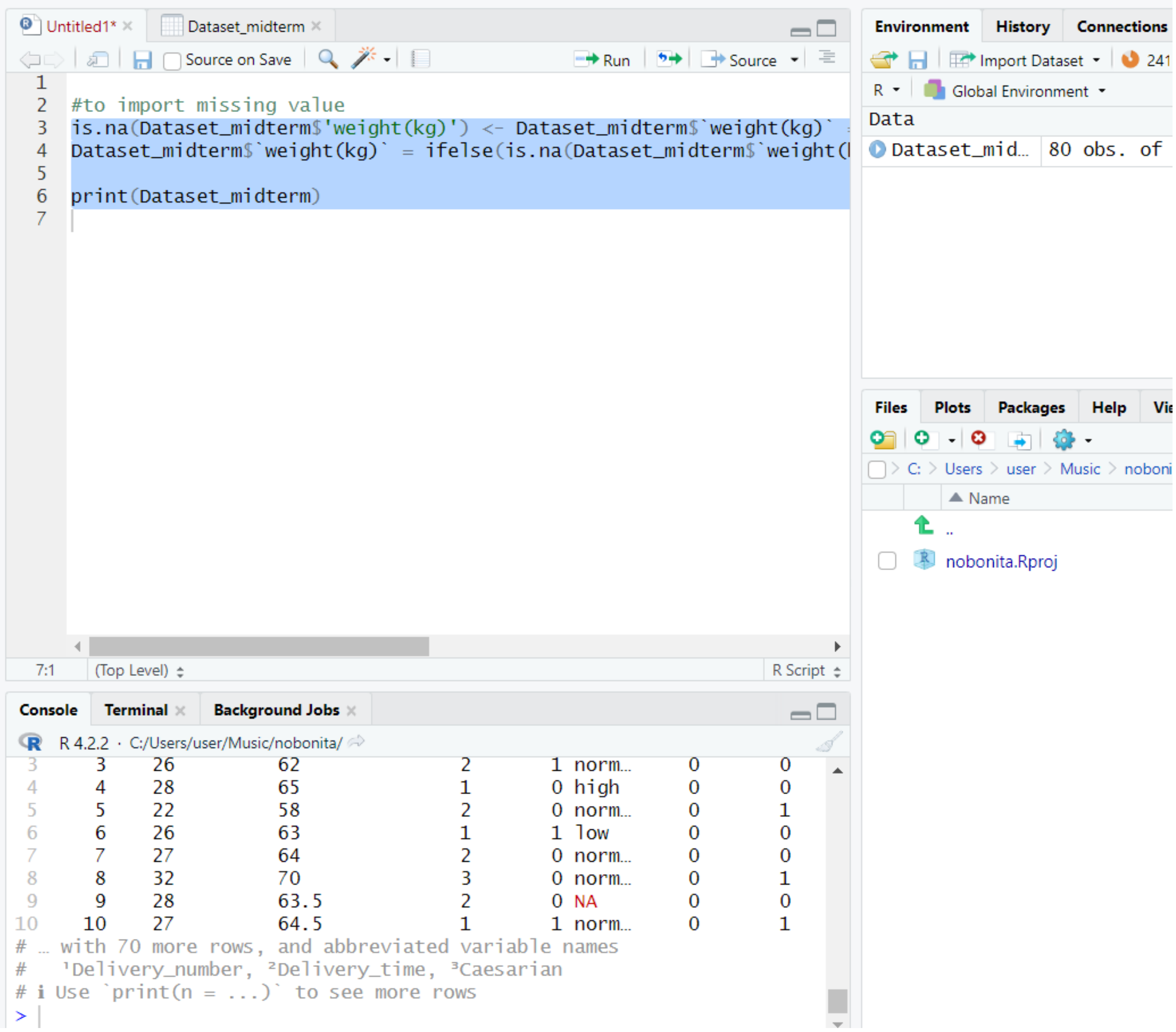
id	Age	weight(kg)
57	32.00000	67.50000
58	95.00000	64.55926

Figure 7: task 4

is.na(Dataset_midterm$'weight(kg)') <- Dataset_midterm$`weight(kg)` == 105

Dataset_midterm$`weight(kg)` = ifelse(is.na(Dataset_midterm$`weight(kg)`),ave(Dataset_midterm$`weight(kg)`,

FUN = function(x) mean(x, na.rm = TRUE)),Dataset_midterm$`weight(kg)`) print(Dataset_midterm)

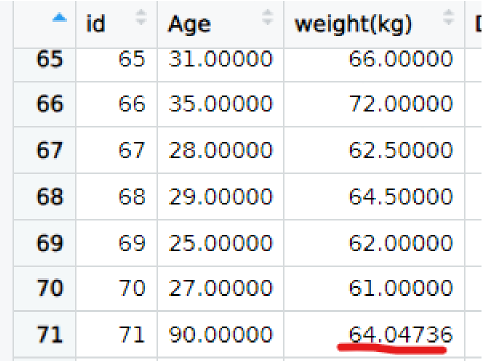


```
#to import missing value
is.na(Dataset_midterm$'weight(kg)') <- Dataset_midterm$`weight(kg)`
Dataset_midterm$`weight(kg)` = ifelse(is.na(Dataset_midterm$`weight(

print(Dataset_midterm)
```

```
Data
Dataset_mid…  80 obs. of
```

```
 3      3    26          62               2          1  norm…    0       0
 4      4    28          65               1          0  high     0       0
 5      5    22          58               2          0  norm…    0       1
 6      6    26          63               1          1  low      0       0
 7      7    27          64               2          0  norm…    0       0
 8      8    32          70               3          0  norm…    0       1
 9      9    28          63.5             2          0  NA       0       0
10     10    27          64.5             1          1  norm…    0       1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
> |
```



	id	Age	weight(kg)	D
65	65	31.00000	66.00000	
66	66	35.00000	72.00000	
67	67	28.00000	62.50000	
68	68	29.00000	64.50000	
69	69	25.00000	62.00000	
70	70	27.00000	61.00000	
71	71	90.00000	64.04736	

Figure 8: task 4

Delivery_number missing value finding:

Task 5:

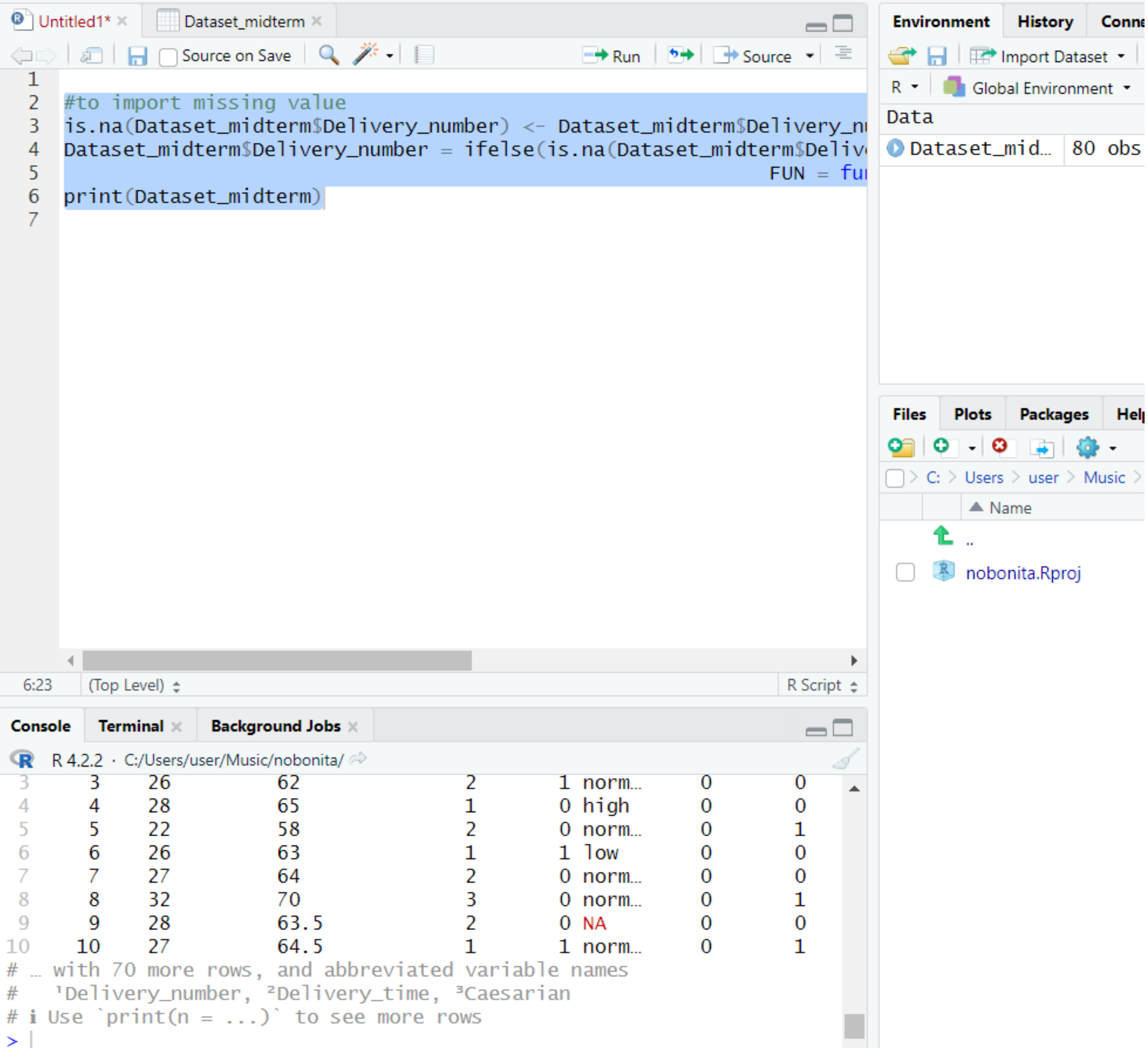
#to import missing value

is.na(Dataset_midterm$Delivery_number) <- Dataset_midterm$Delivery_number == 0

Dataset_midterm$Delivery_number = ifelse(is.na(Dataset_midterm$Delivery_number),ave(Dataset_midterm$Delivery_number,

FUN = function(x) mean(x, na.rm = TRUE)),Dataset_midterm$Delivery_number)

print(Dataset_midterm)



```

#to import missing value
is.na(Dataset_midterm$Delivery_number) <- Dataset_midterm$Delivery_nu
Dataset_midterm$Delivery_number = ifelse(is.na(Dataset_midterm$Delive
                                               FUN = fu
print(Dataset_midterm)

```

```
 3      3    26        62          2       1 norm…      0       0
 4      4    28        65          1       0 high       0       0
 5      5    22        58          2       0 norm…      0       1
 6      6    26        63          1       1 low        0       0
 7      7    27        64          2       0 norm…      0       0
 8      8    32        70          3       0 norm…      0       1
 9      9    28        63.5        2       0 NA         0       0
10     10    27        64.5        1       1 norm…      0       1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
> 
```

Figure 9: task 5

Delivery_time missing value finding:

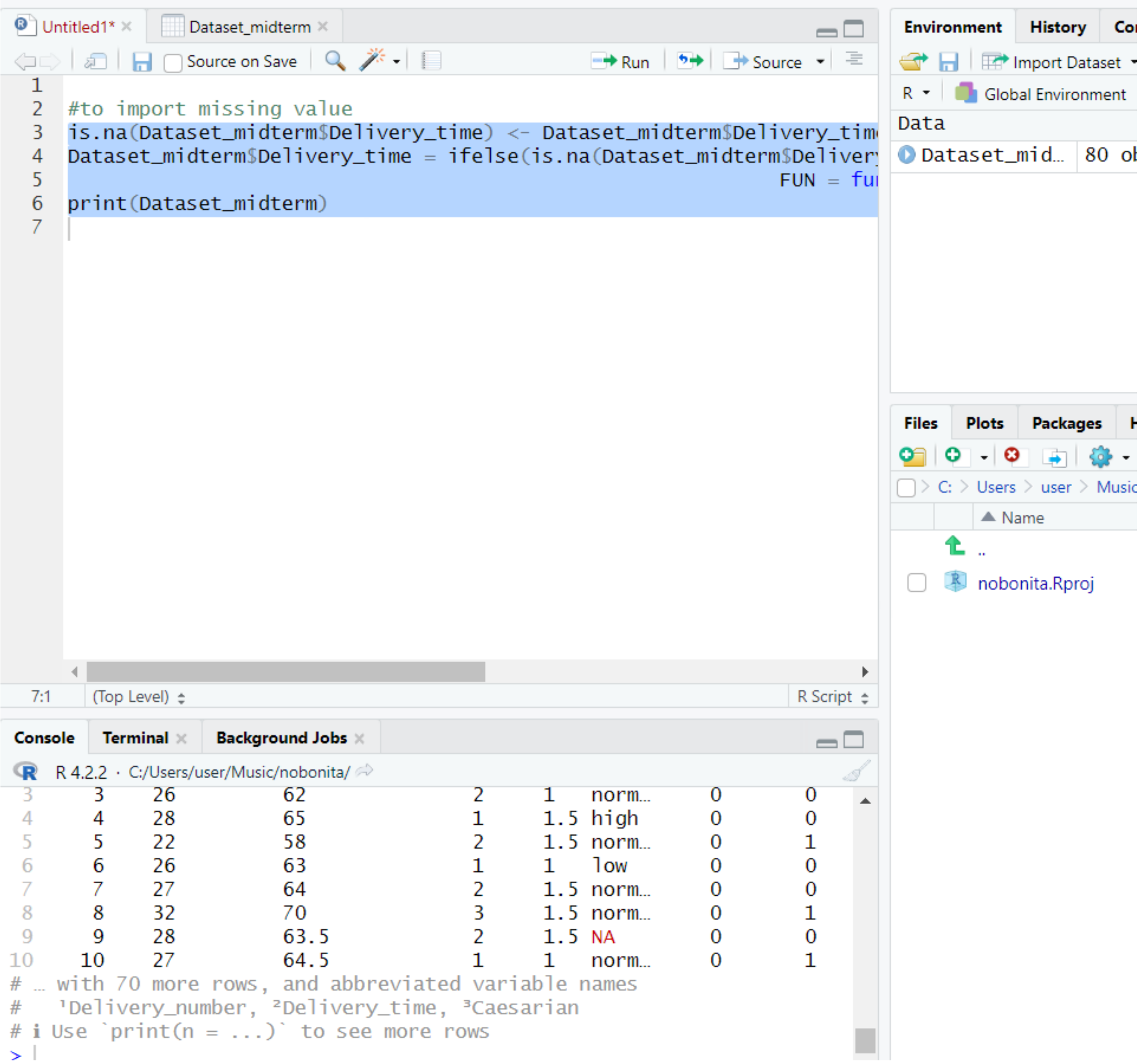
Task 6:

is.na(Dataset_midterm$Delivery_time) <- Dataset_midterm$Delivery_time == 0

Dataset_midterm$Delivery_time = ifelse(is.na(Dataset_midterm$Delivery_time),ave(Dataset_midterm$Delivery_time,

FUN = function(x) mean(x, na.rm = TRUE)),Dataset_midterm$Delivery_time)

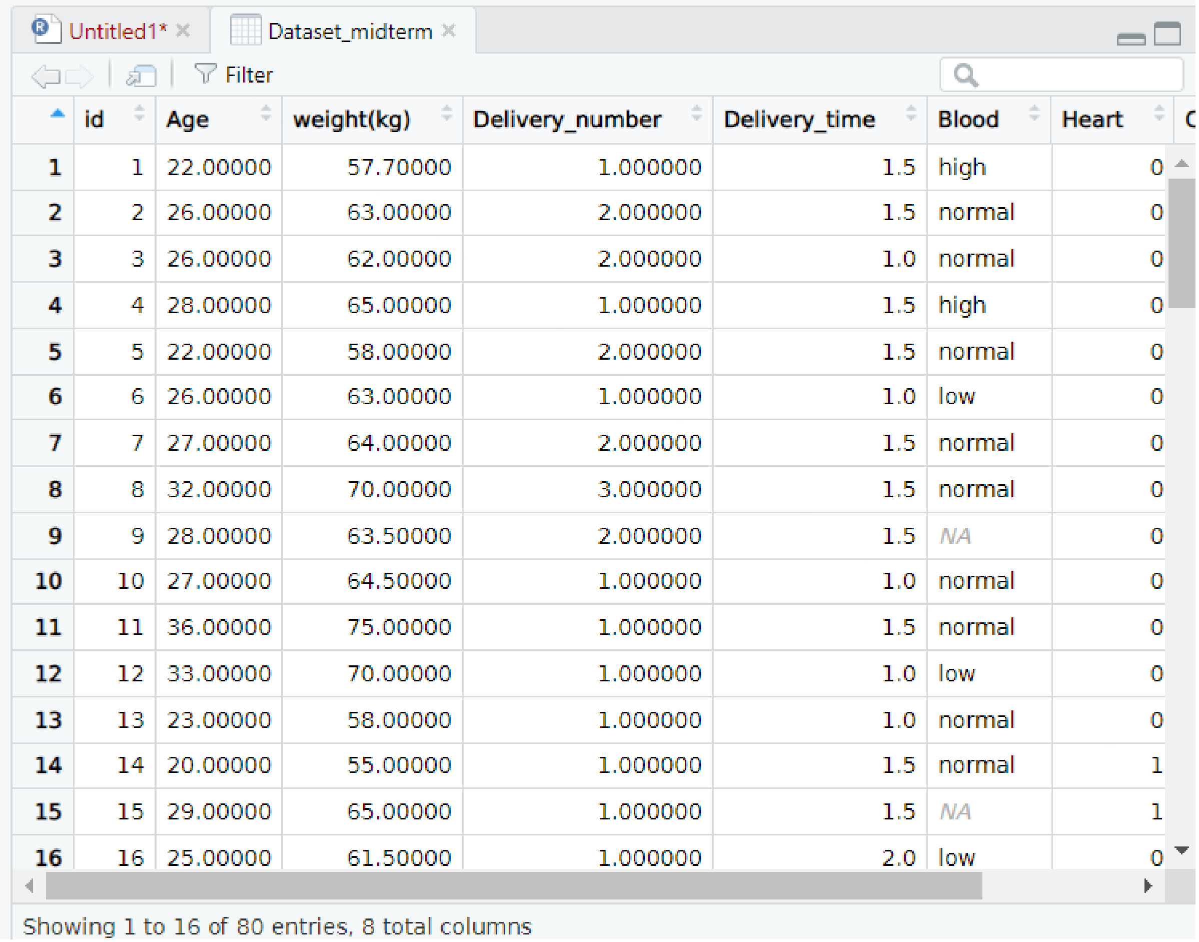
print(Dataset_midterm)



```
#to import missing value
is.na(Dataset_midterm$Delivery_time) <- Dataset_midterm$Delivery_tim
Dataset_midterm$Delivery_time = ifelse(is.na(Dataset_midterm$Deliver
                                                    FUN = fu
print(Dataset_midterm)
```

```
 3     3    26          62              2     1    norm…     0       0
 4     4    28          65              1     1.5  high      0       0
 5     5    22          58              2     1.5  norm…     0       1
 6     6    26          63              1     1    low       0       0
 7     7    27          64              2     1.5  norm…     0       0
 8     8    32          70              3     1.5  norm…     0       1
 9     9    28          63.5            2     1.5  NA        0       0
10    10    27          64.5            1     1    norm…     0       1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
> 
```

Figure 10: task 6



	id	Age	weight(kg)	Delivery_number	Delivery_time	Blood	Heart	C
1	1	22.00000	57.70000	1.000000	1.5	high	0	
2	2	26.00000	63.00000	2.000000	1.5	normal	0	
3	3	26.00000	62.00000	2.000000	1.0	normal	0	
4	4	28.00000	65.00000	1.000000	1.5	high	0	
5	5	22.00000	58.00000	2.000000	1.5	normal	0	
6	6	26.00000	63.00000	1.000000	1.0	low	0	
7	7	27.00000	64.00000	2.000000	1.5	normal	0	
8	8	32.00000	70.00000	3.000000	1.5	normal	0	
9	9	28.00000	63.50000	2.000000	1.5	NA	0	
10	10	27.00000	64.50000	1.000000	1.0	normal	0	
11	11	36.00000	75.00000	1.000000	1.5	normal	0	
12	12	33.00000	70.00000	1.000000	1.0	low	0	
13	13	23.00000	58.00000	1.000000	1.0	normal	0	
14	14	20.00000	55.00000	1.000000	1.5	normal	1	
15	15	29.00000	65.00000	1.000000	1.5	NA	1	
16	16	25.00000	61.50000	1.000000	2.0	low	0	

Showing 1 to 16 of 80 entries, 8 total columns

Figure 11: task 6

For missing value of Blood pressure:

Task 7:

Suppose,

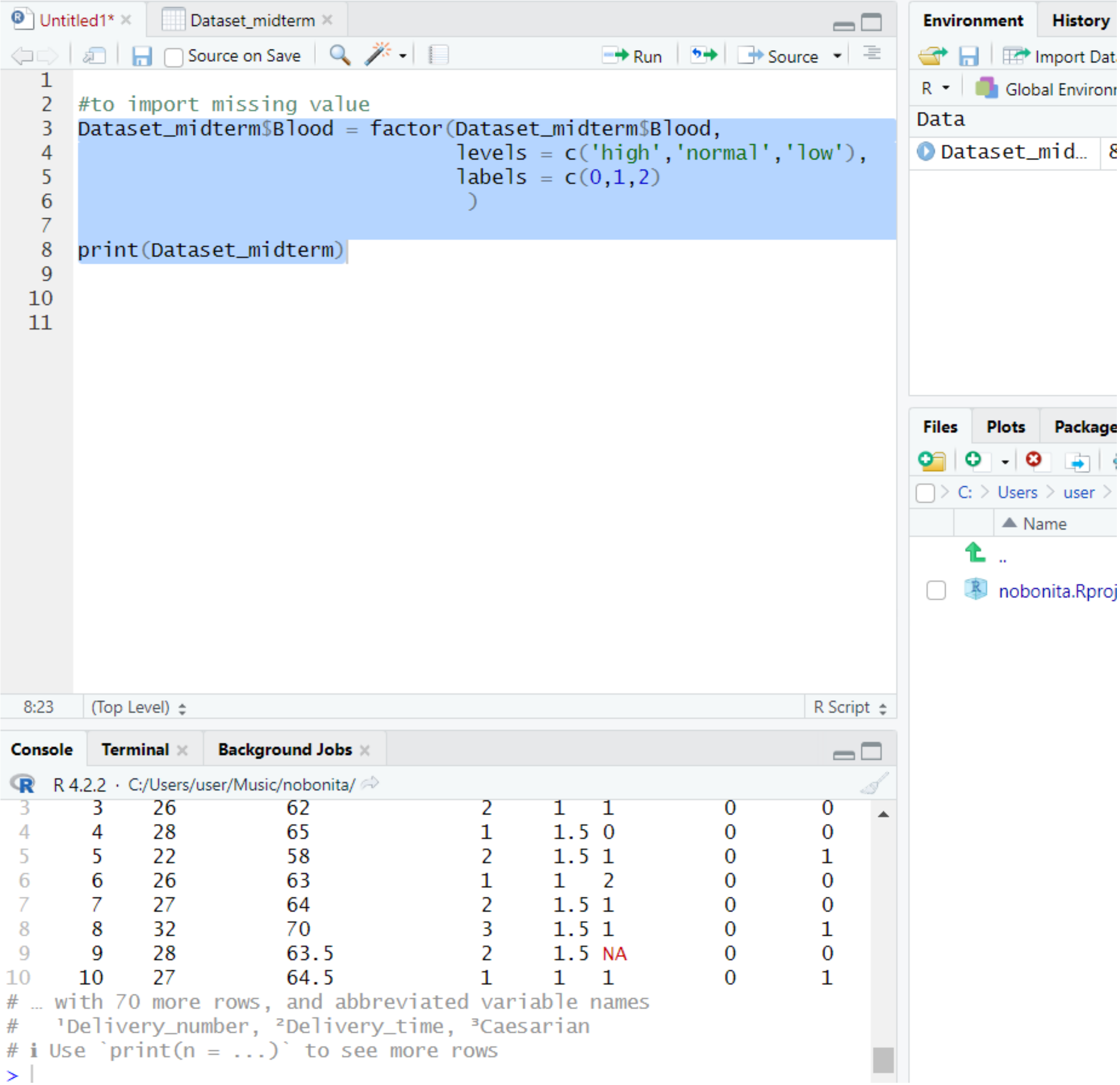
High = 0

Normal = 1

Low = 2

```
Dataset_midterm$Blood = factor(Dataset_midterm$Blood,
                        levels = c('high','normal','low'),
                        labels = c(0,1,2)
                        )
```

print(Dataset_midterm)



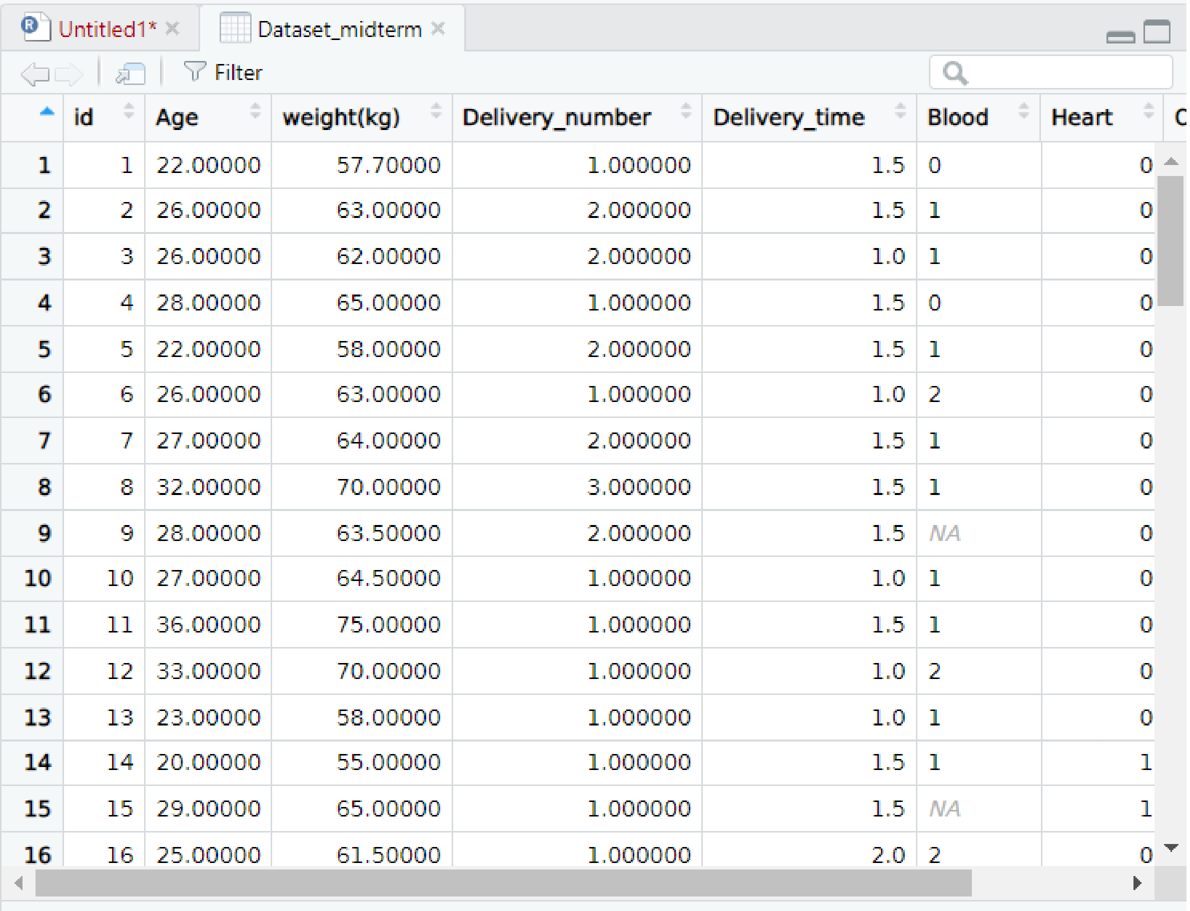
```

#to import missing value
Dataset_midterm$Blood = factor(Dataset_midterm$Blood,
                          levels = c('high','normal','low'),
                          labels = c(0,1,2)
                           )

print(Dataset_midterm)
```

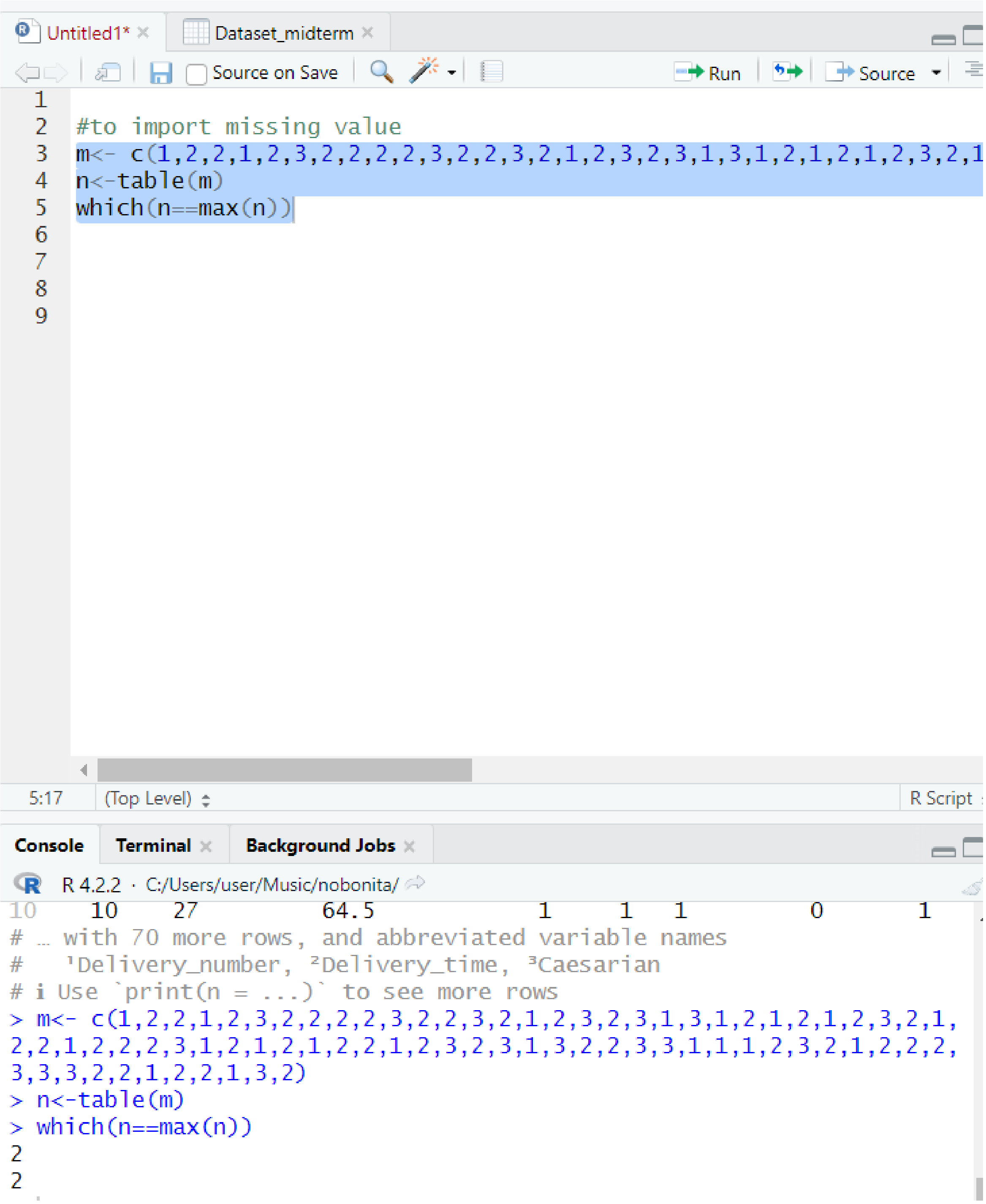
```
 3     3    26     62          2     1    1        0       0
 4     4    28     65          1     1.5  0        0       0
 5     5    22     58          2     1.5  1        0       1
 6     6    26     63          1     1    2        0       0
 7     7    27     64          2     1.5  1        0       0
 8     8    32     70          3     1.5  1        0       1
 9     9    28     63.5        2     1.5  NA       0       0
10    10    27     64.5        1     1    1        0       1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
> |
```

Figure 12: task 7



	id	Age	weight(kg)	Delivery_number	Delivery_time	Blood	Heart	C
1	1	22.00000	57.70000	1.000000	1.5	0	0	
2	2	26.00000	63.00000	2.000000	1.5	1	0	
3	3	26.00000	62.00000	2.000000	1.0	1	0	
4	4	28.00000	65.00000	1.000000	1.5	0	0	
5	5	22.00000	58.00000	2.000000	1.5	1	0	
6	6	26.00000	63.00000	1.000000	1.0	2	0	
7	7	27.00000	64.00000	2.000000	1.5	1	0	
8	8	32.00000	70.00000	3.000000	1.5	1	0	
9	9	28.00000	63.50000	2.000000	1.5	NA	0	
10	10	27.00000	64.50000	1.000000	1.0	1	0	
11	11	36.00000	75.00000	1.000000	1.5	1	0	
12	12	33.00000	70.00000	1.000000	1.0	2	0	
13	13	23.00000	58.00000	1.000000	1.0	1	0	
14	14	20.00000	55.00000	1.000000	1.5	1	1	
15	15	29.00000	65.00000	1.000000	1.5	NA	1	
16	16	25.00000	61.50000	1.000000	2.0	2	0	

Figure 13: task 7



```
#to import missing value
m<- c(1,2,2,1,2,3,2,2,2,2,3,2,2,3,2,1,2,3,2,3,1,3,1,2,1,2,1,2,3,2,1
n<-table(m)
which(n==max(n))
```

```
10     10     27          64.5            1      1    1          0        1
# … with 70 more rows, and abbreviated variable names
#   ¹Delivery_number, ²Delivery_time, ³Caesarian
# ℹ Use `print(n = ...)` to see more rows
> m<- c(1,2,2,1,2,3,2,2,2,2,3,2,2,3,2,1,2,3,2,3,1,3,1,2,1,2,1,2,3,2,1,
2,2,1,2,2,2,3,1,2,1,2,1,2,2,1,2,3,2,3,1,3,2,2,3,3,1,1,1,2,3,2,1,2,2,2,
3,3,3,2,2,1,2,2,1,3,2)
> n<-table(m)
> which(n==max(n))
2
2
```

Figure 14: task 8

2 is mode value, so 2 will take place of the N/A

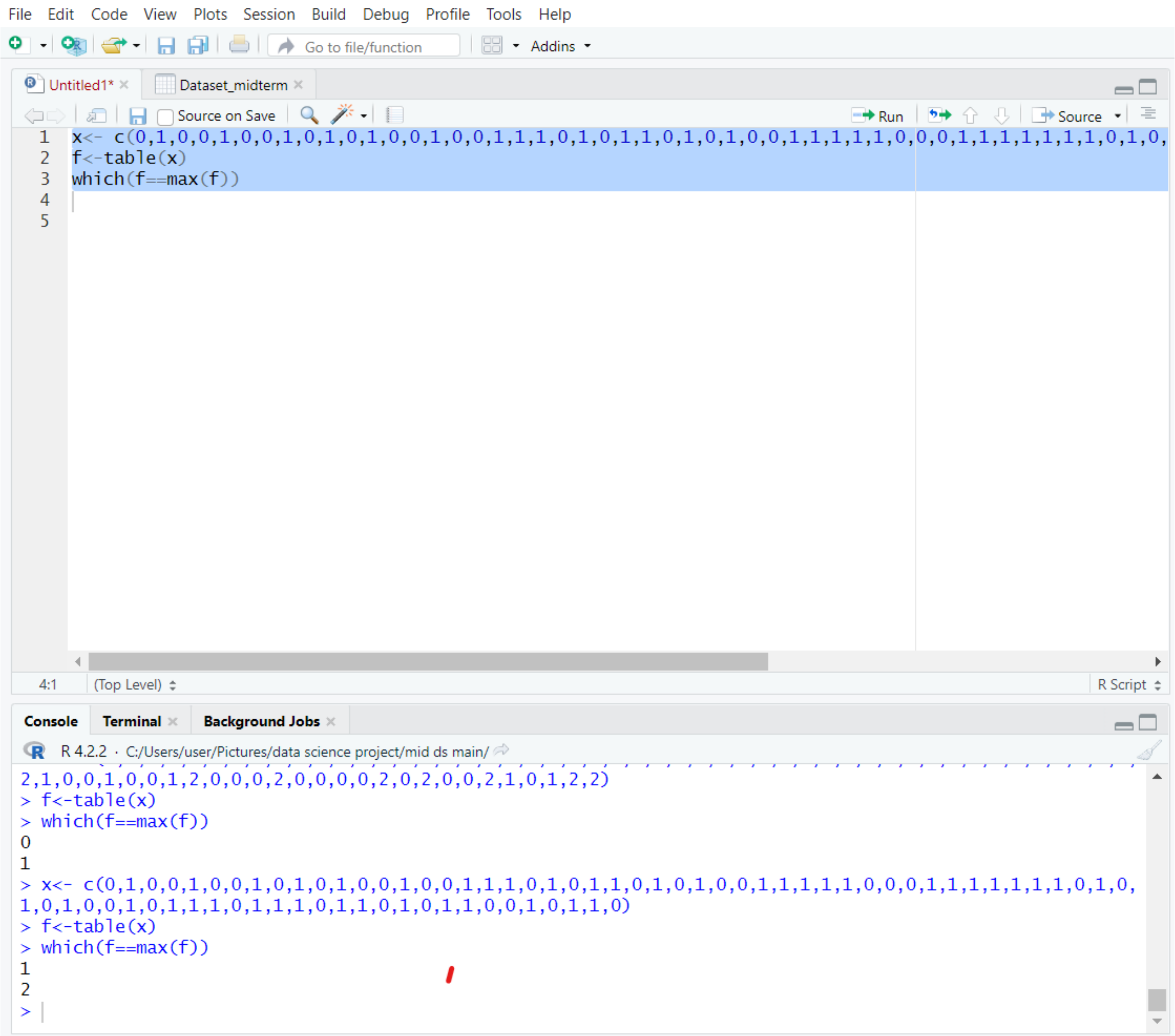
For missing value of caesarian:

Task 9:

x<-
c(0,1,0,0,1,0,0,1,0,1,0,1,0,0,1,0,0,1,1,1,0,1,0,1,1,0,1,0,1,0,0,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,1,0,1,
0,1,0,0,1,0,1,1,1,0,1,1,1,0,1,1,0,1,0,1,1,0,0,1,0,1,1,0)

f<-table(x)

which(f==max(f))



```
x<- c(0,1,0,0,1,0,0,1,0,1,0,1,0,0,1,0,0,1,1,1,0,1,0,1,1,0,1,0,1,0,0,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,1,0,
f<-table(x)
which(f==max(f))
```

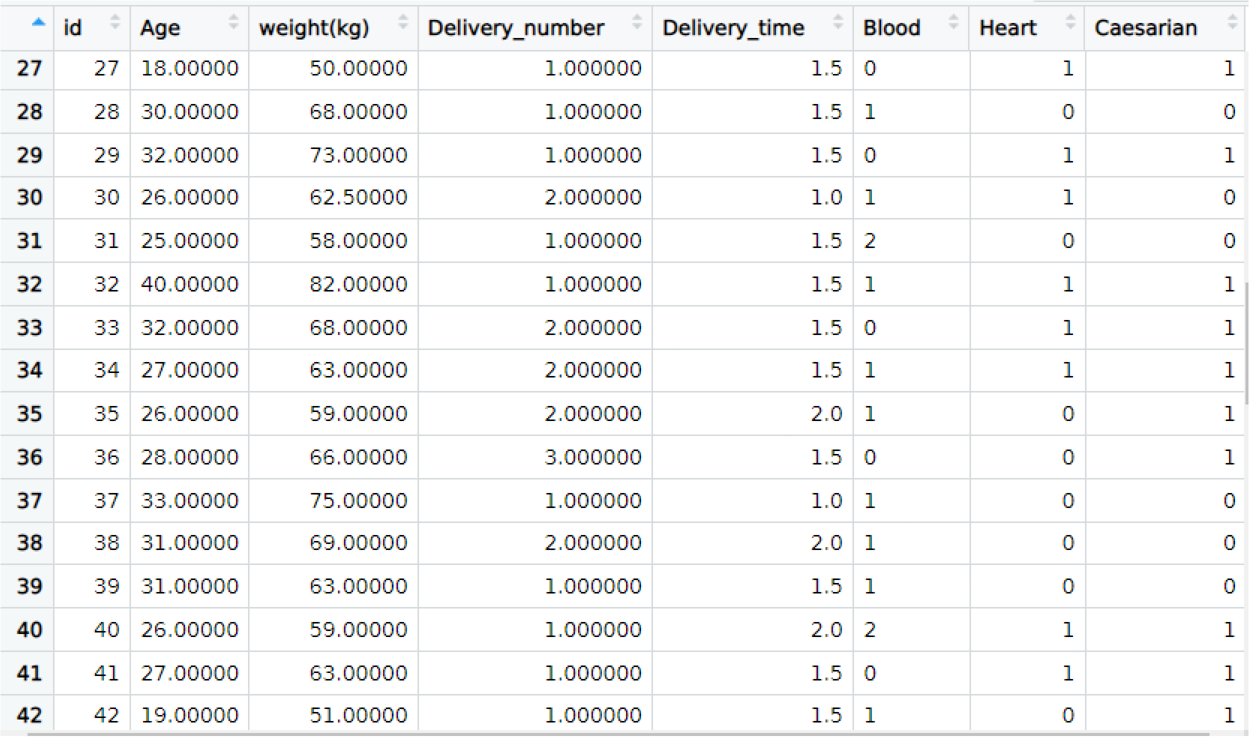
```
2,1,0,0,1,0,0,1,2,0,0,0,2,0,0,0,0,2,0,2,0,0,2,1,0,1,2,2)
> f<-table(x)
> which(f==max(f))
0
1
> x<- c(0,1,0,0,1,0,0,1,0,1,0,1,0,0,1,0,0,1,1,1,0,1,0,1,1,0,1,0,1,0,0,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,1,0,
1,0,1,0,0,1,0,1,1,1,0,1,1,1,0,1,1,0,1,0,1,1,0,0,1,0,1,1,0)
> f<-table(x)
> which(f==max(f))
1
2
> 
```

Figure 15: task 9

Here mode value is 1, so all the NA values can be replaced with 1.

Conclusion:

The target of getting a fulfilled data set is complete through the preprocessing of data set. As part of this project, we dealt with missing data with mean and mode, noisy value and discretized some data and with all this I have completed my preprocessing of data set.



	id	Age	weight(kg)	Delivery_number	Delivery_time	Blood	Heart	Caesarian
27	27	18.00000	50.00000	1.000000	1.5	0	1	1
28	28	30.00000	68.00000	1.000000	1.5	1	0	0
29	29	32.00000	73.00000	1.000000	1.5	0	1	1
30	30	26.00000	62.50000	2.000000	1.0	1	1	0
31	31	25.00000	58.00000	1.000000	1.5	2	0	0
32	32	40.00000	82.00000	1.000000	1.5	1	1	1
33	33	32.00000	68.00000	2.000000	1.5	0	1	1
34	34	27.00000	63.00000	2.000000	1.5	1	1	1
35	35	26.00000	59.00000	2.000000	2.0	1	0	1
36	36	28.00000	66.00000	3.000000	1.5	0	0	1
37	37	33.00000	75.00000	1.000000	1.0	1	0	0
38	38	31.00000	69.00000	2.000000	2.0	1	0	0
39	39	31.00000	63.00000	1.000000	1.5	1	0	0
40	40	26.00000	59.00000	1.000000	2.0	2	1	1
41	41	27.00000	63.00000	1.000000	1.5	0	1	1
42	42	19.00000	51.00000	1.000000	1.5	1	0	1

Figure 16: A complete Data Set