

KING KHALID UNIVERSITY

College: Computer Science

Department: Information Systems

Course Instructor: Ms. Wejdan Mansoor



Course code: 373 CIS-4

Course Name: Data Mining

Total Marks: 10 Marks

## LAB PROJECT

Students Names :	Student ID.No :

### INSTRUCTIONS:

- 1) The deadline is on 10\2\2023, 19\7\1444.
- 2) Groups members are from 3 to 4 members.
- 3) Copied assignments **will lead to marks deduction**.
- 4) Make screens shot for all steps and write down all the related codes also arrange document properly.

### • Open RStudio software and apply the following questions:

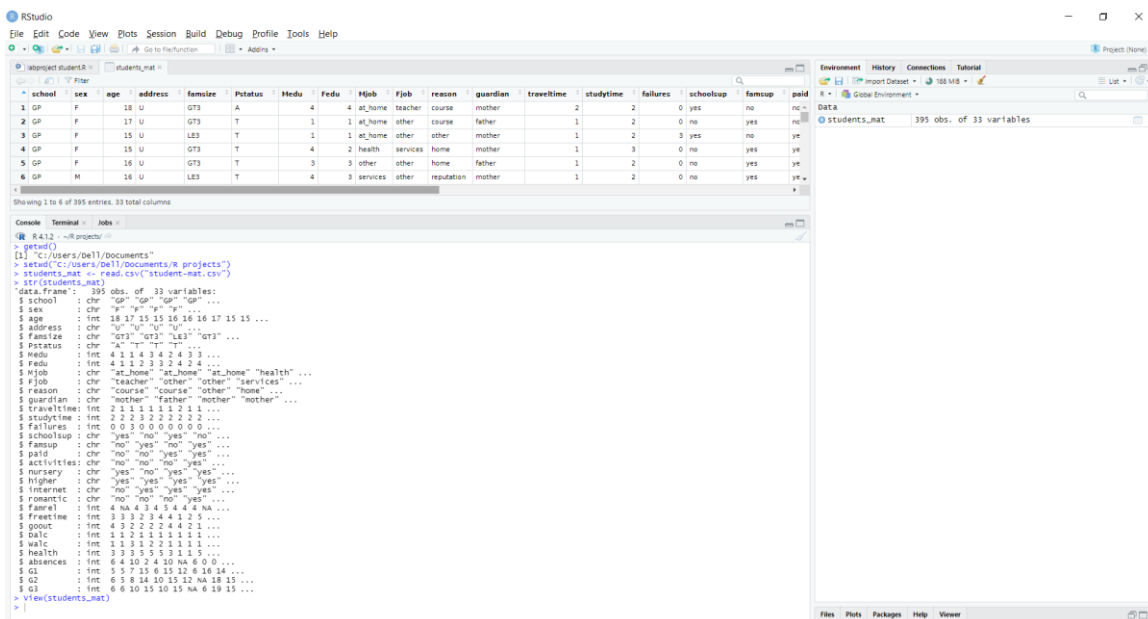
1. Import the Students-mat to R studio, check the attribute information click on the following link:

[Math Students | Kaggle](#)

```
students_mat <- read.csv("student-mat.csv")
```

```
str(students_mat)
```

```
View(students_mat)
```



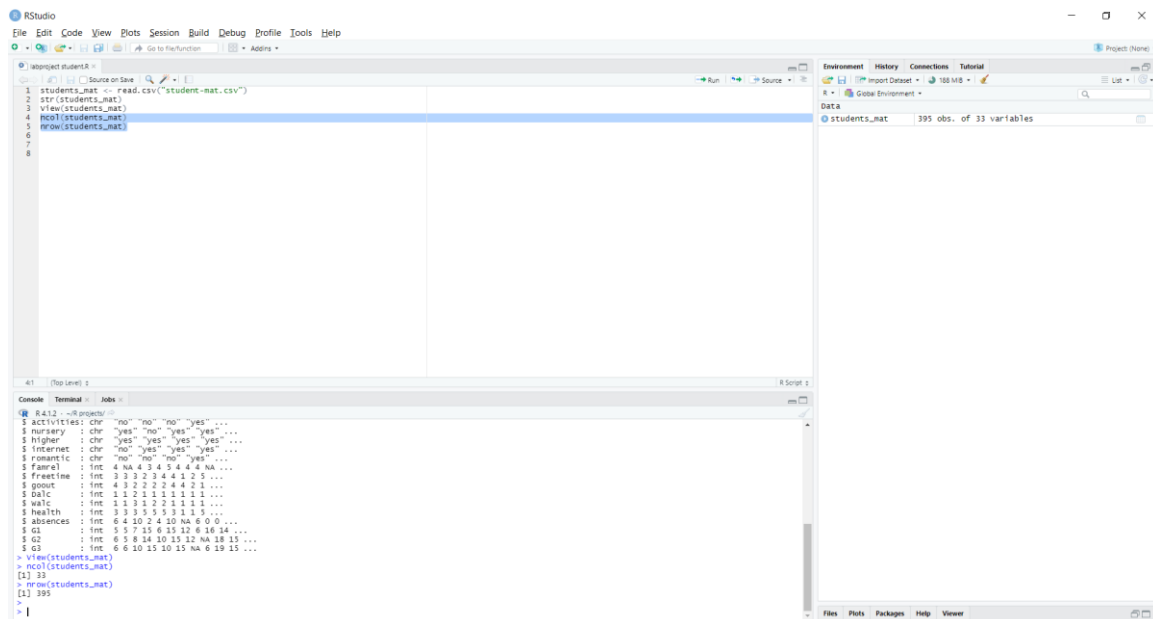
2. How many columns and observations in the dataset.

```
ncol(students_mat)
```

```
nrow(students_mat)
```

```
columns = 33
```

```
rows      = 395
```



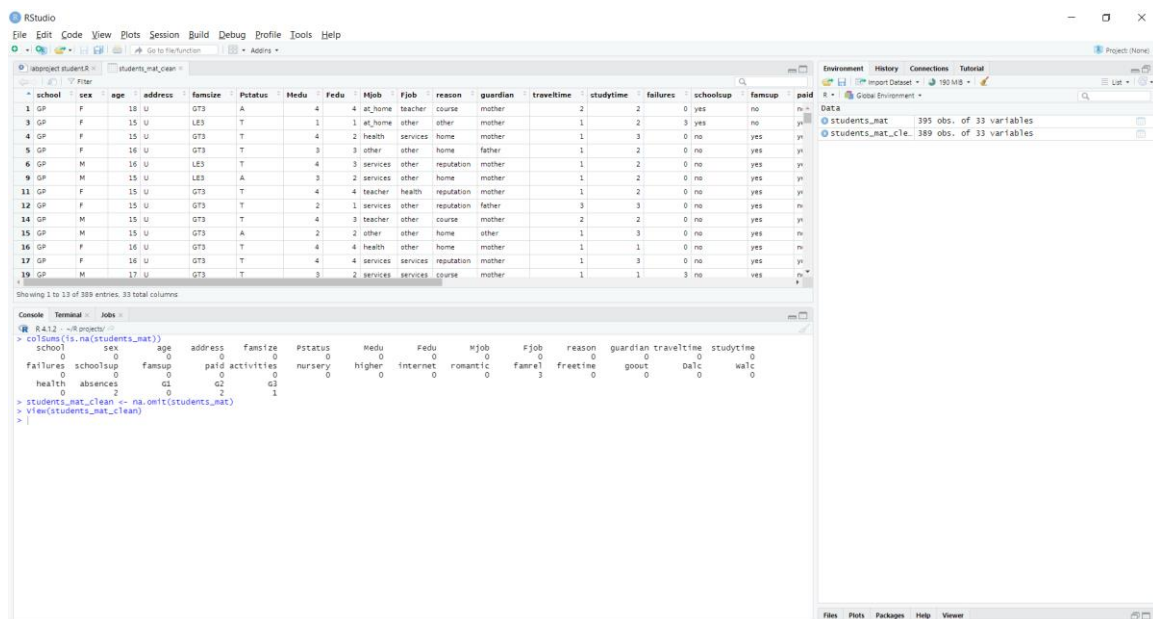
```
1 students_mat <- read.csv("student-mat.csv")
2 str(students_mat)
3 view(students_mat)
4 ncol(students_mat)
5 nrow(students_mat)
```

```
R 4.1.2 --R project--
> activities: chr "no" "no" "no" "yes" ...
> nursery: chr "yes" "no" "yes" "yes" ...
> higher: chr "yes" "yes" "yes" "yes" ...
> internet: chr "no" "yes" "yes" "yes" ...
> romantic: chr "no" "no" "no" "yes" ...
> famrel: fct 4 NA 4 3 4 5 4 4 NA ...
> freetime: fct 3 3 2 3 4 4 5 2 5 ...
> goout: fct 4 3 2 2 2 4 4 2 1 ...
> dalc: fct 1 1 1 1 1 1 1 1 1 ...
> walc: fct 1 0 1 3 2 2 0 1 1 ...
> health: fct 3 3 3 5 5 5 3 1 5 ...
> absences: fct 6 4 10 2 4 10 NA 6 0 ...
> g1: fct 5 5 7 15 6 15 12 6 16 14 ...
> g2: fct 6 5 8 14 10 15 12 NA 18 15 ...
> g3: fct 6 6 10 15 10 15 NA 6 18 15 ...
> view(students_mat)
> ncol(students_mat)
[1] 33
> nrow(students_mat)
[1] 395
> |
```

3. Specify the missing values in which columns then clean your dataset by drop the tuples the include the missing data.

```
colSums(is.na(students_mat))
```

```
students_mat_clean <- na.omit(students_mat)
```



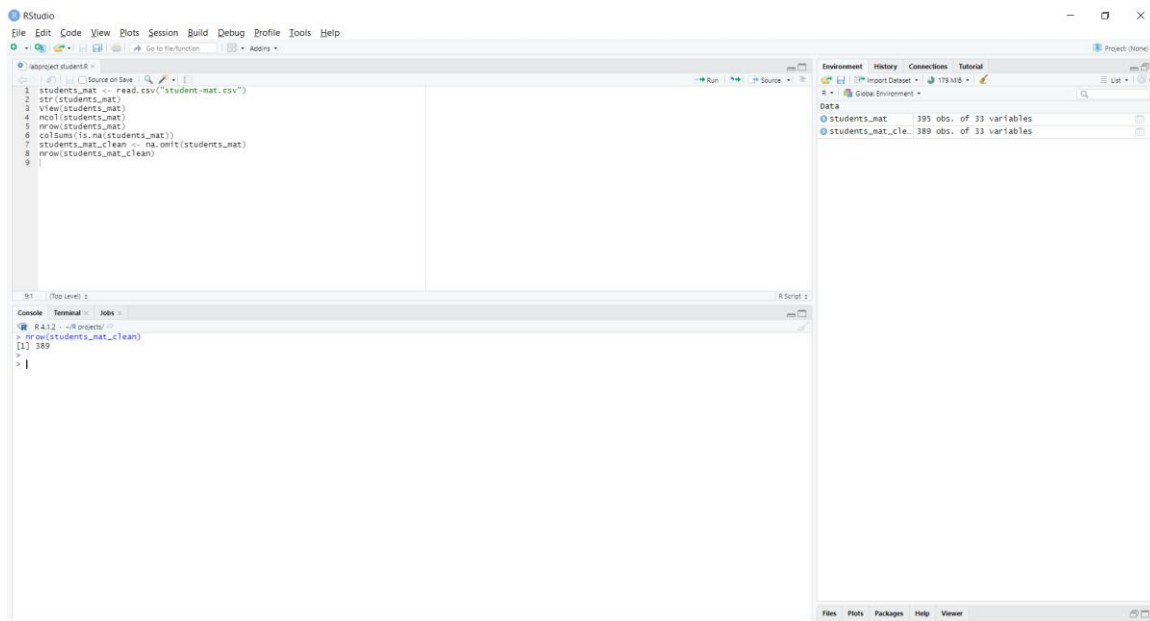
```
> colSums(is.na(students_mat))
 school sex age address famsize Ptstatus Medu Fedu Fjob Rjob reason guardian traveltime studytime failures schoolsup famsup paid
 0      0    0    0      0          0      0      0      0      0      0          0          0          0          0          0
failures schoolsup famsup paid activities nursery higher internet romantic famrel freetime goout dalc walc
 0      0      0      0          0      0      0      0      0      0      0          0          0          0
health absences g1 g2 g3
 0      0      0      0      0
> students_mat_clean <- na.omit(students_mat)
> view(students_mat_clean)
```

school	sex	age	address	famsize	Ptstatus	Medu	Fedu	Fjob	Rjob	reason	guardian	traveltime	studytime	failures	schoolsup	famsup	paid
GP	F	18	U	GTS	A	4	4	at_home	teacher	course	mother	2	2	0	yes	no	no
GP	F	15	U	LES	T	1	1	at_home	other	other	mother	1	2	3	yes	no	no
GP	F	15	U	GTS	T	4	2	health	services	home	mother	1	3	0	no	yes	no
GP	F	16	U	GTS	T	3	3	other	other	home	father	1	2	0	no	yes	no
GP	M	16	U	LES	T	4	3	services	other	reputation	mother	1	2	0	no	yes	no
GP	M	15	U	LES	A	3	2	services	other	home	mother	1	2	0	no	yes	no
GP	F	15	U	GTS	T	4	4	teacher	health	reputation	mother	1	2	0	no	yes	no
GP	F	15	U	GTS	T	2	1	services	other	reputation	father	3	3	0	no	yes	no
GP	M	15	U	GTS	T	4	3	teacher	other	course	mother	2	2	0	no	yes	no
GP	M	15	U	GTS	A	2	2	other	other	home	other	1	3	0	no	yes	no
GP	F	16	U	GTS	T	4	4	health	other	home	mother	1	1	0	no	yes	no
GP	F	16	U	GTS	T	4	4	services	services	reputation	mother	1	3	0	no	yes	no
GP	M	17	U	GTS	T	3	2	services	services	course	mother	1	1	3	no	yes	no

4. How many tuples do you have now after cleaning the data set?

```
nrow(students_mat_clean)
```

```
rows = 389
```



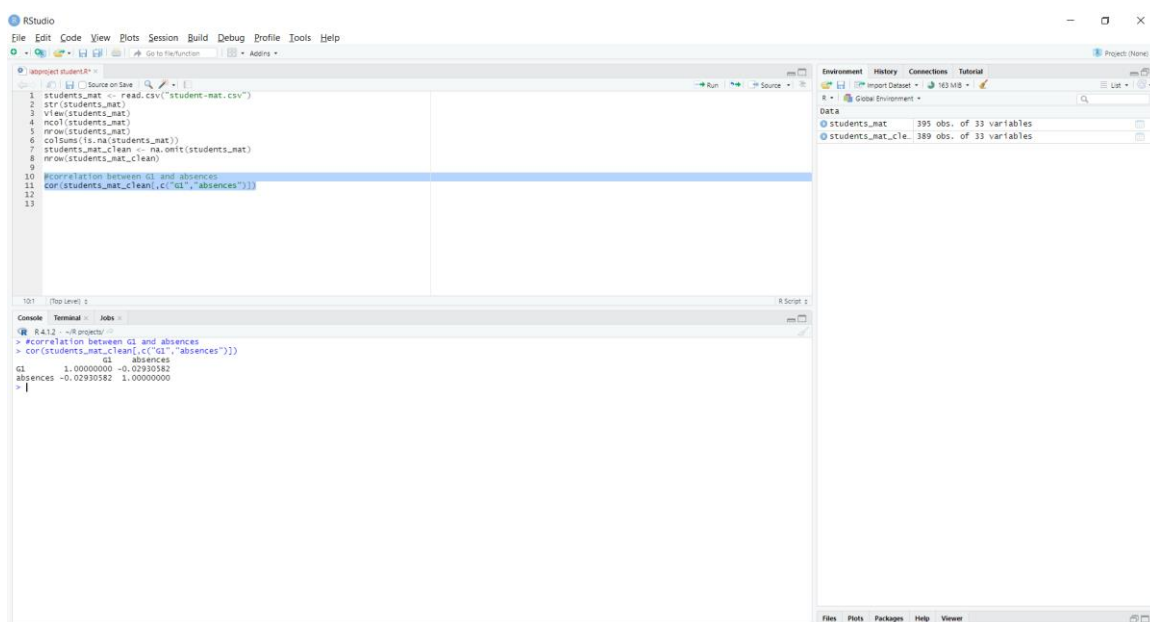
5. Find the correlation between "G1" attribute (Grade1) and the following attributes:

- " absences "
- " studytime "

- " G2 "
- " G3 "

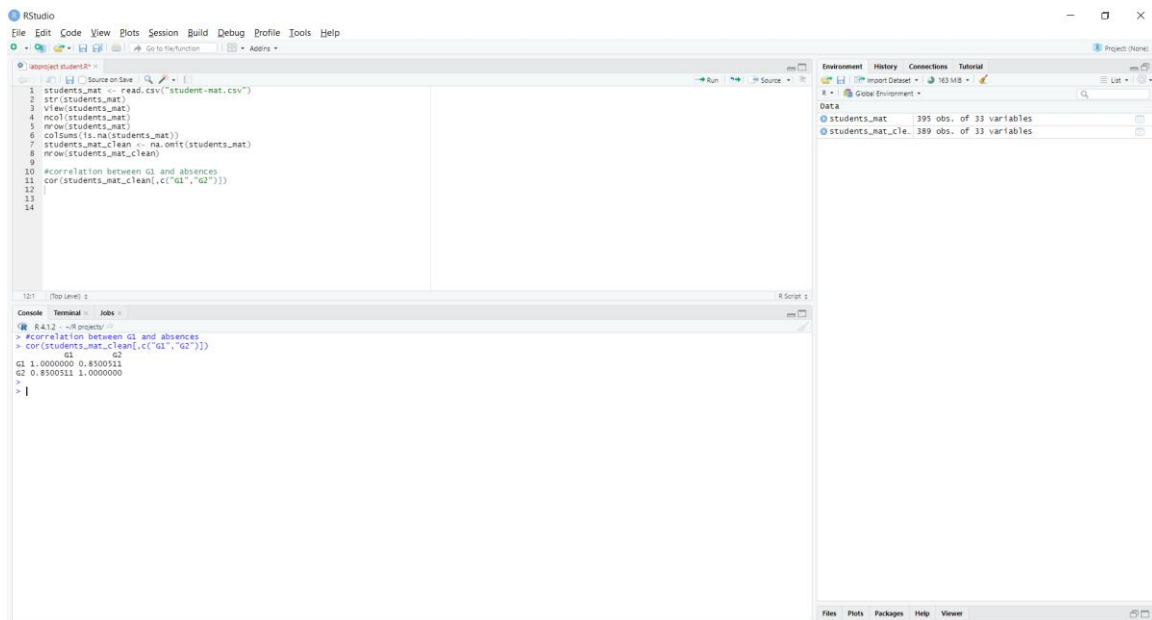
#correlation between G1 and absences

```
cor(students_mat_clean[,c("G1", "absences")])
```

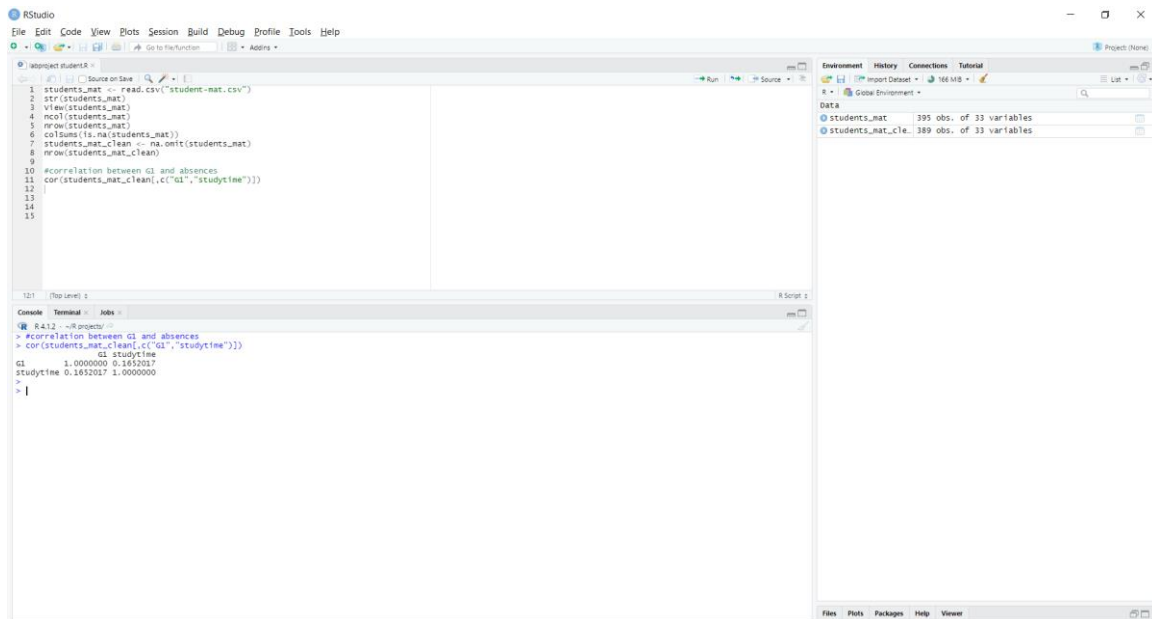


#correlation between G1 and absences

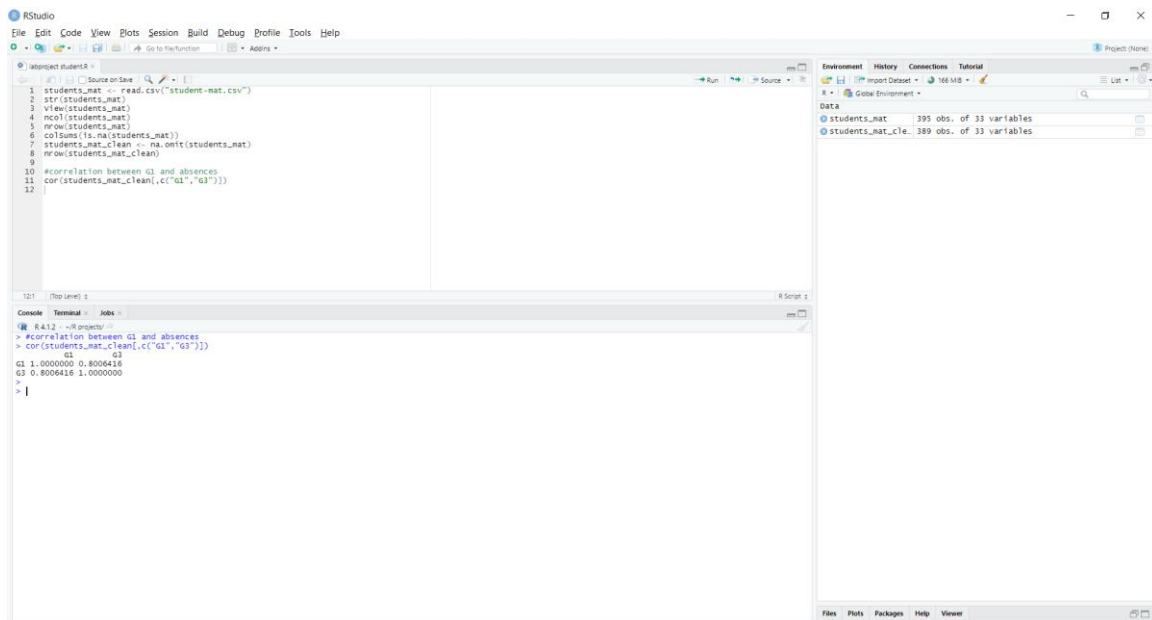
```
cor(students_mat_clean[,c("G1", "G2")])
```



#correlation between G1 and absences  
`cor(students_mat_clean[,c("G1","studytime")])`

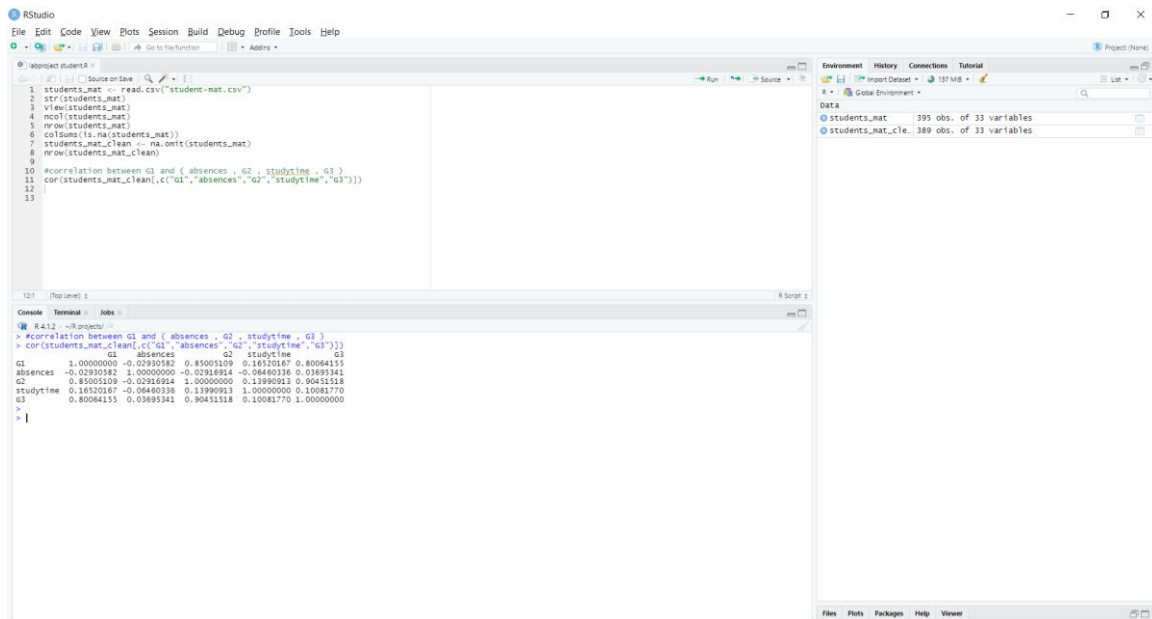


#correlation between G1 and absences  
`cor(students_mat_clean[,c("G1","G3")])`



#correlation between G1 and ( absences , G2 , studytime , G3 )

cor(students\_mat\_clean[,c("G1", "absences", "G2", "studytime", "G3")])



6. Specify the correlation types in point 2 and plot the relations using **scatter plot diagram** (Note: plot each relation and explain it).

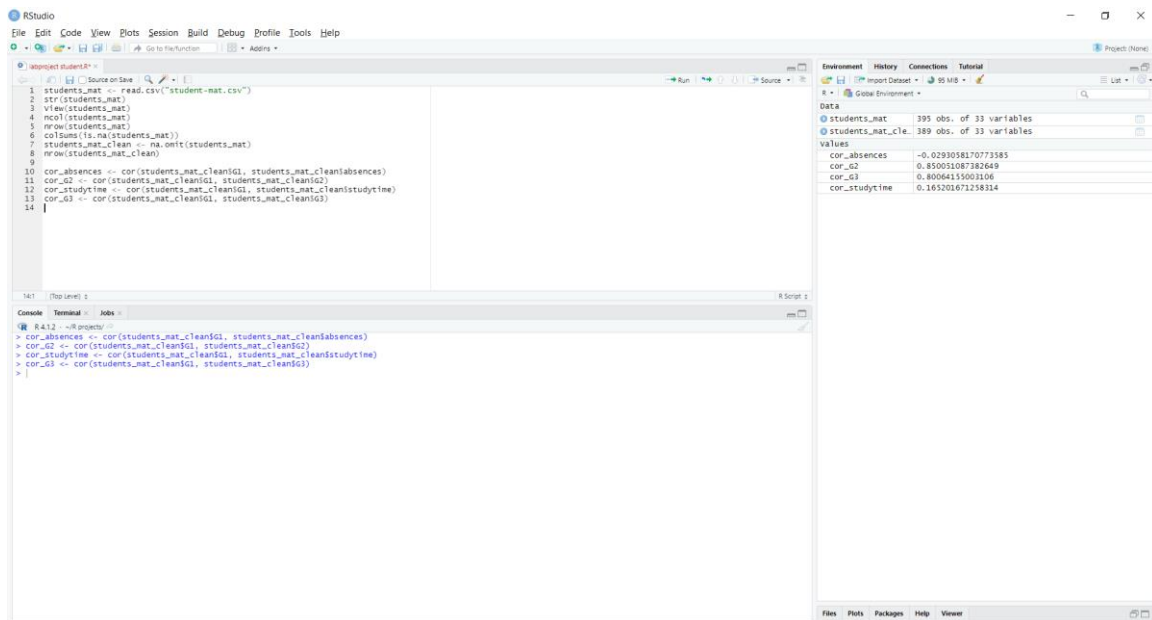
To find the correlation types between "G1" and "absences", "G2", "studytime", "G3", you can use the cor function in R:

```
cor_absences <- cor(students_mat_clean$G1, students_mat_clean$absences)
```

```
cor_G2 <- cor(students_mat_clean$G1, students_mat_clean$G2)
```

```
cor_studytime <- cor(students_mat_clean$G1, students_mat_clean$studytime)
```

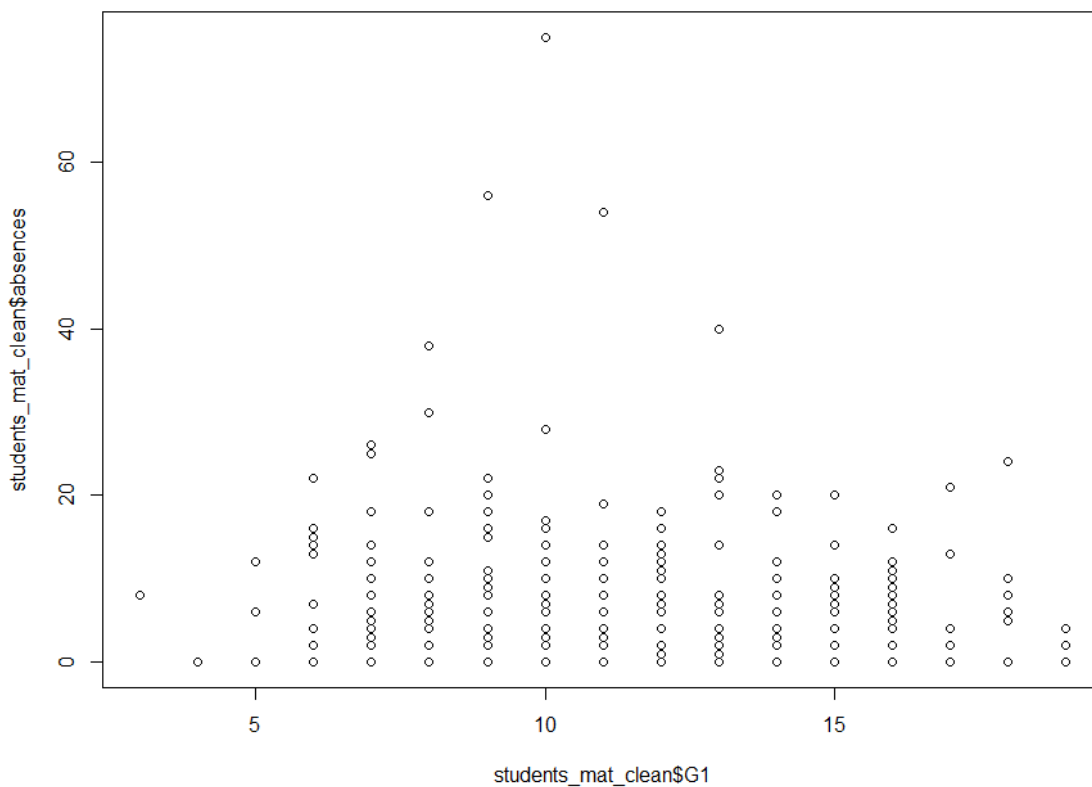
```
cor_G3 <- cor(students_mat_clean$G1, students_mat_clean$G3)
```

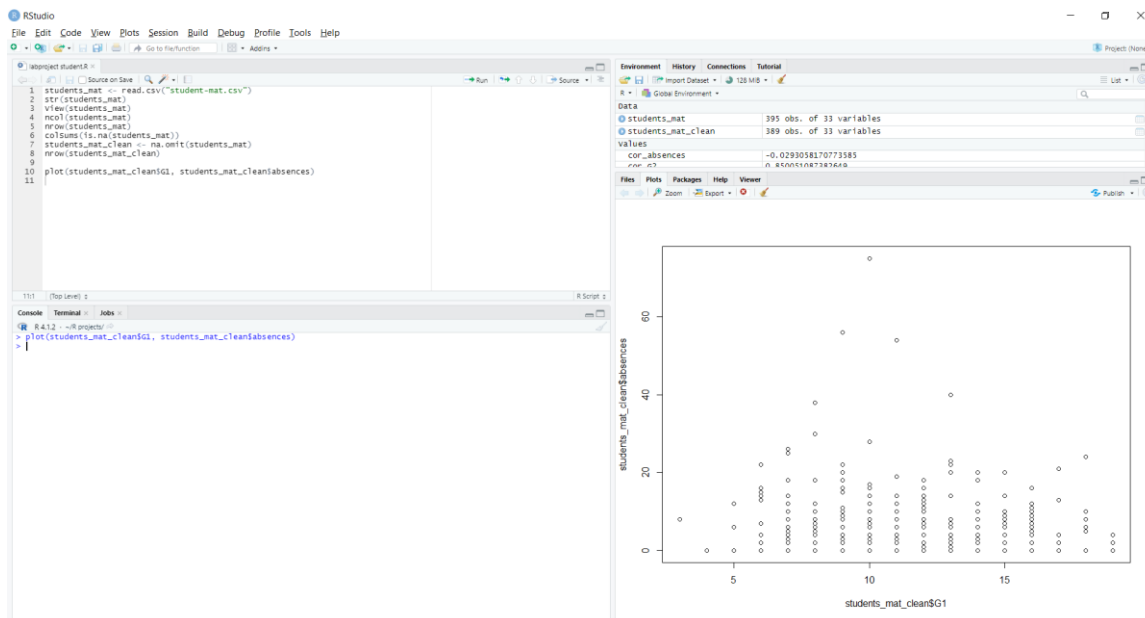


The `cor` function returns the Pearson's correlation coefficient, which measures the linear relationship between two variables. A positive correlation means that as one variable increases, the other variable increases as well. A negative correlation means that as one variable increases, the other variable decreases. A value of 1 indicates a perfect positive correlation, and a value of -1 indicates a perfect negative correlation. A value of 0 indicates no correlation.

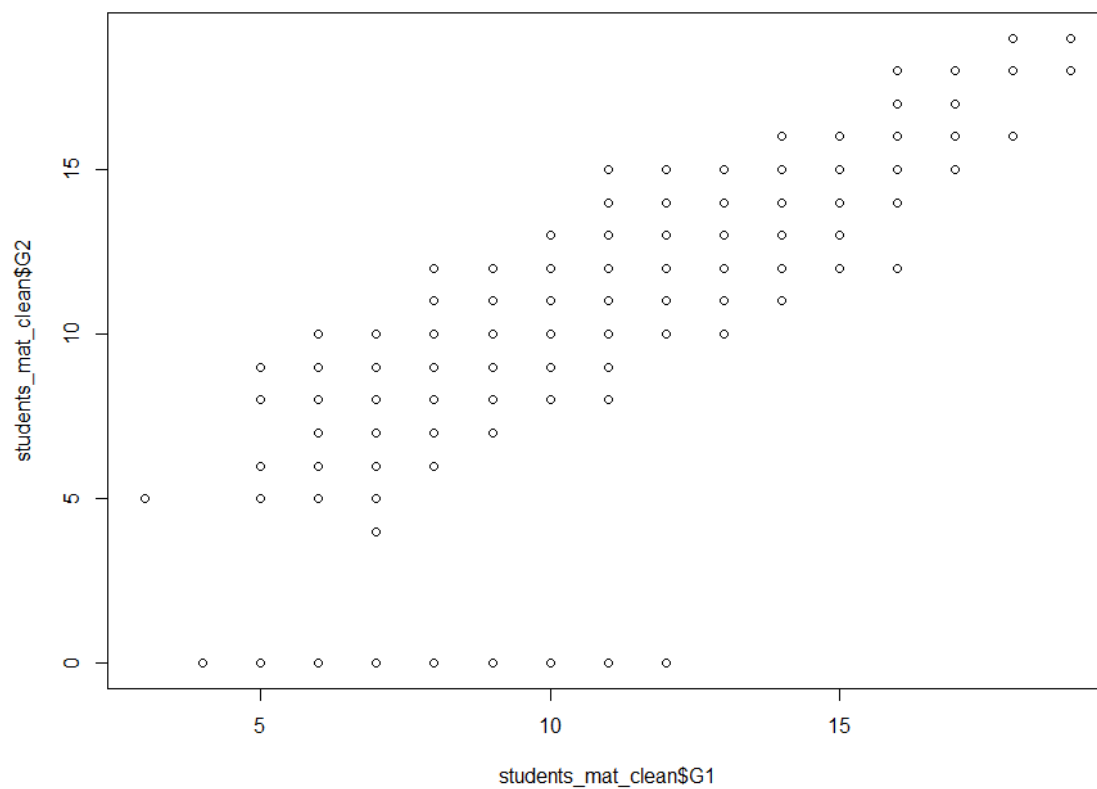
To plot the scatter plots, you can use the `plot` function in R:

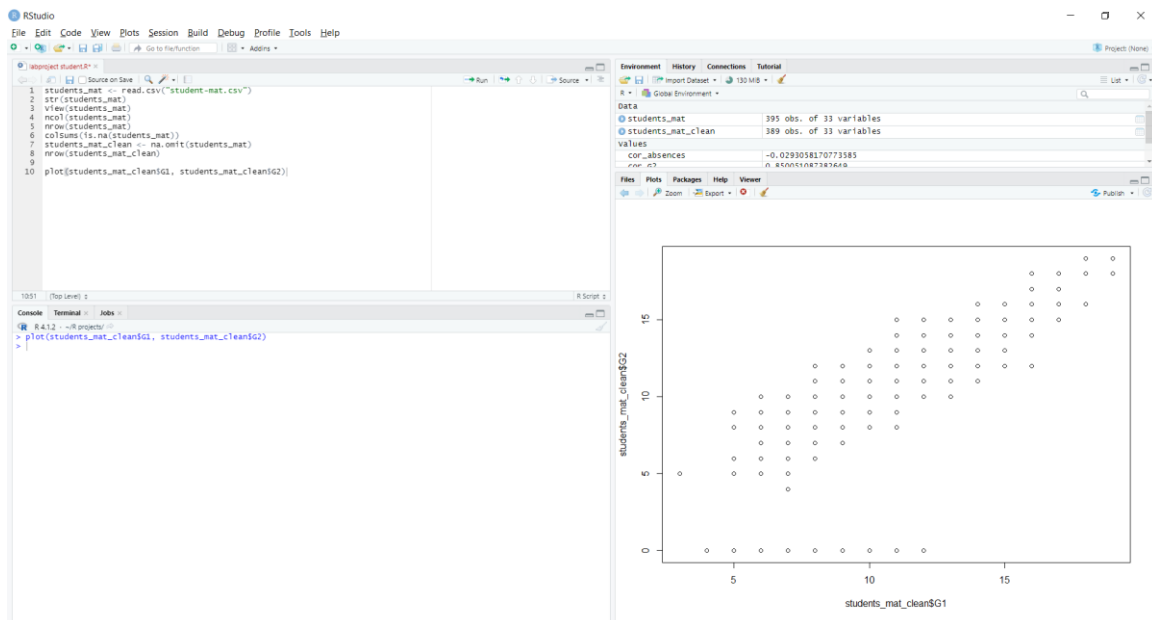
```
plot(students_mat_clean$G1, students_mat_clean$absences)
```



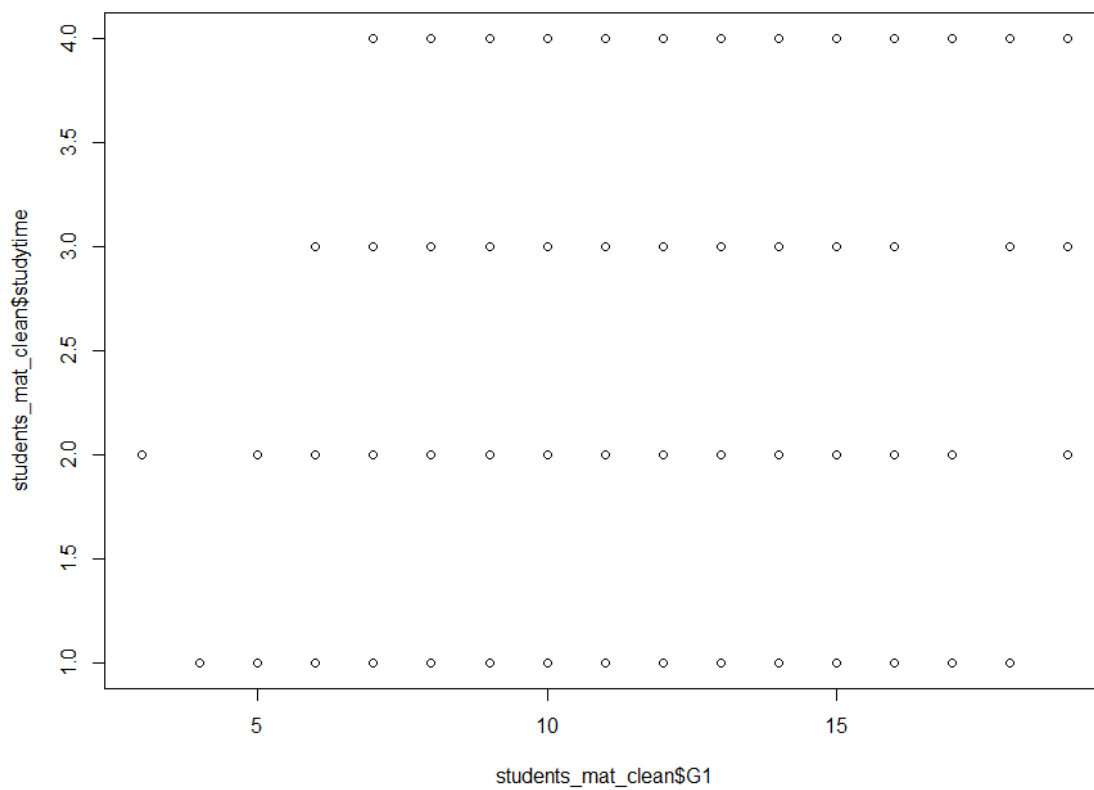


`plot(students_mat_clean$G1, students_mat_clean$G2)`

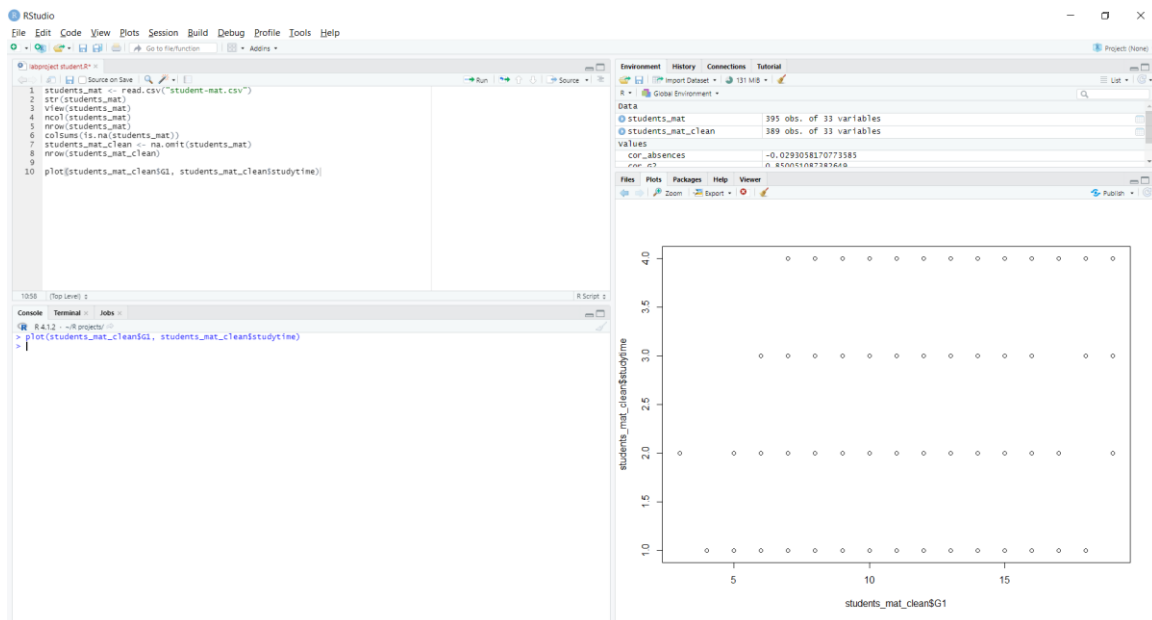




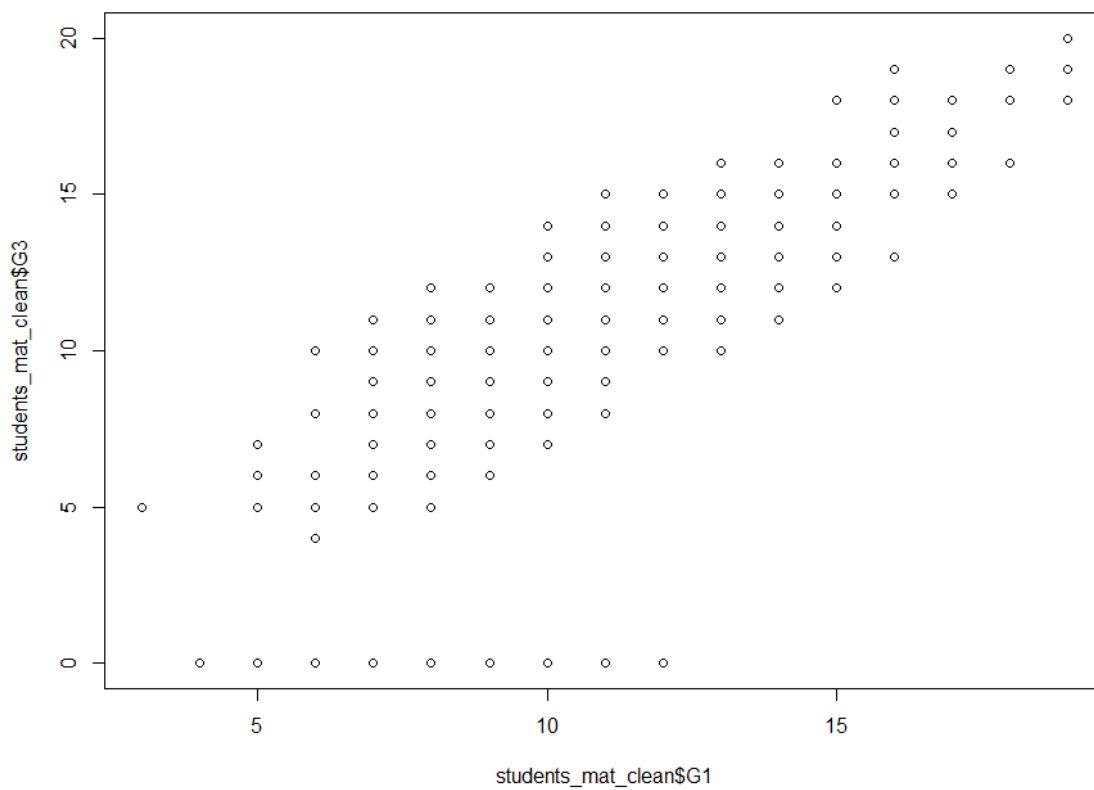
`plot(students_mat_clean$G1, students_mat_clean$studytime)`

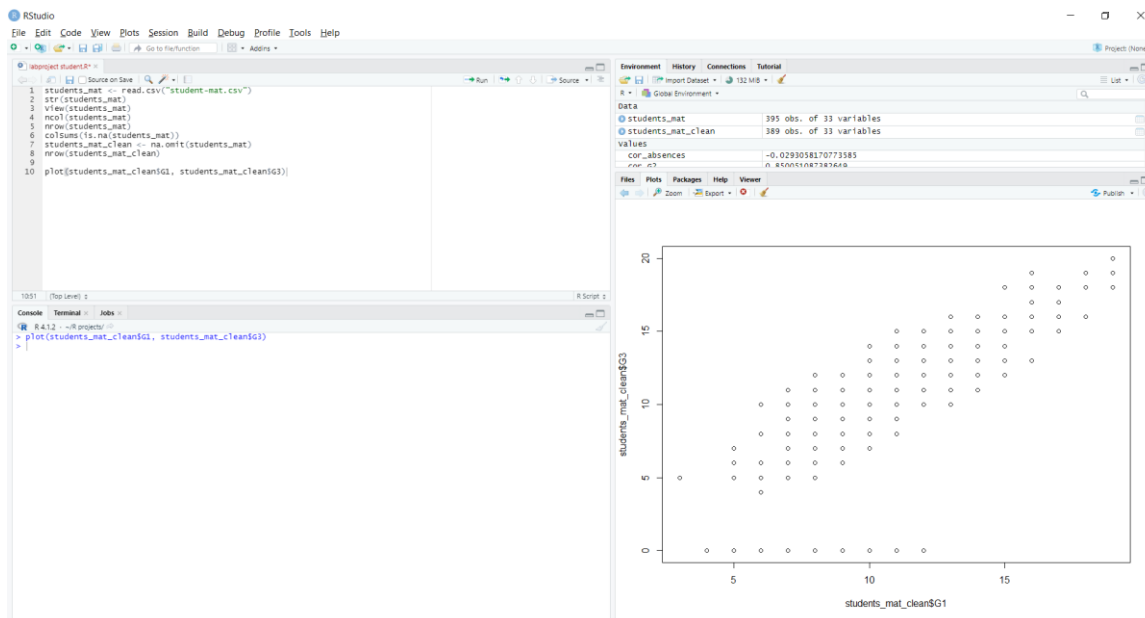






`plot(students_mat_clean$G1, students_mat_clean$G3)`

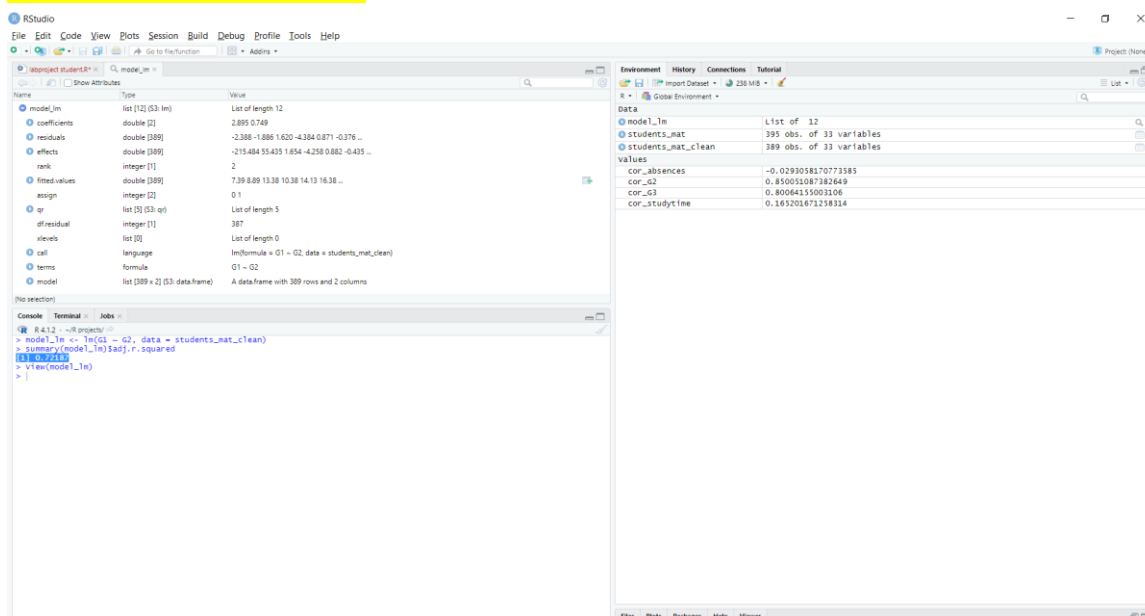




7. Apply linear regression algorithm on "**Grade1**" and "**Grade2**" attributes to calculate the Adjusted R Squared.

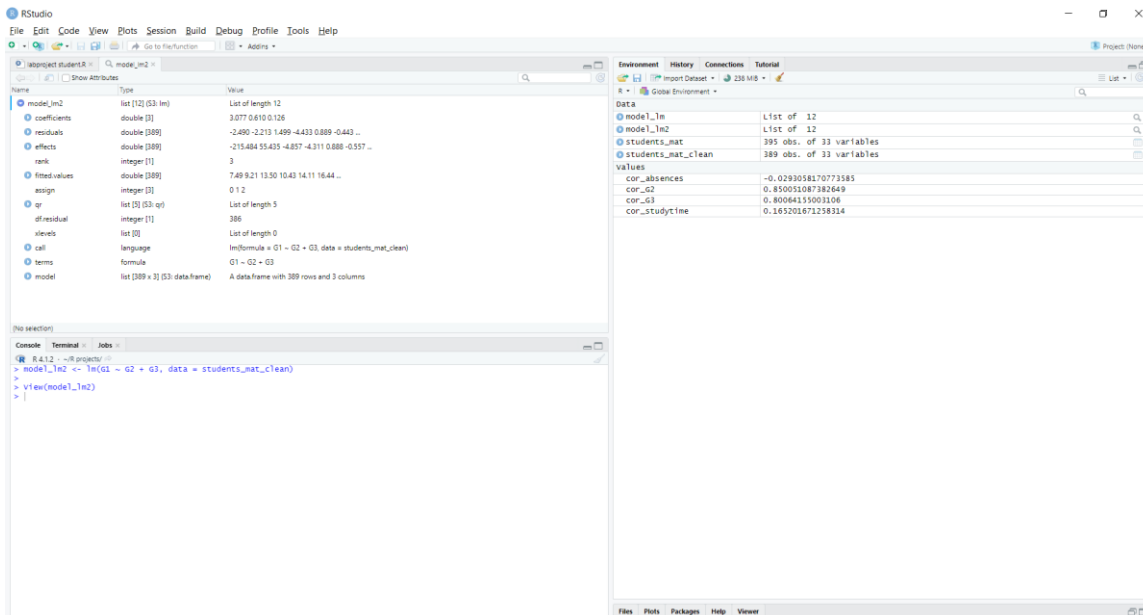
```
model_lm <- lm(G1 ~ G2, data = students_mat_clean)
summary(model_lm)$adj.r.squared
```

**Adjusted R Squared = 0.72187**



8. Apply the multiple regression algorithm to (G1,G2,G3) columns where the G1 as dependent variable and the two others as independent.

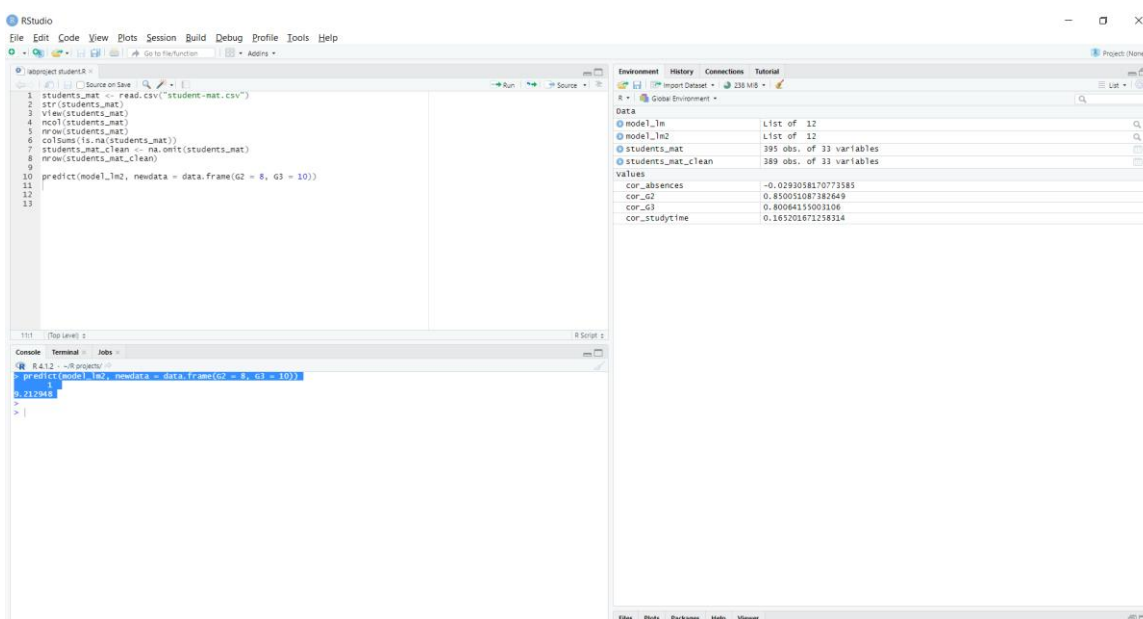
To apply multiple regression to predict "G1" based on "G2" and "G3", you can use the lm function:  
`model_lm2 <- lm(G1 ~ G2 + G3, data = students_mat_clean)`



9. If we assume that the student got G2 as 8 and G3 as 10 what will be the value of G1?  
(Note: apply the equation for multiple regression model)

To make a prediction for "G1" based on the assumption that "G2" is 8 and "G3" is 10, you can use the predict function:

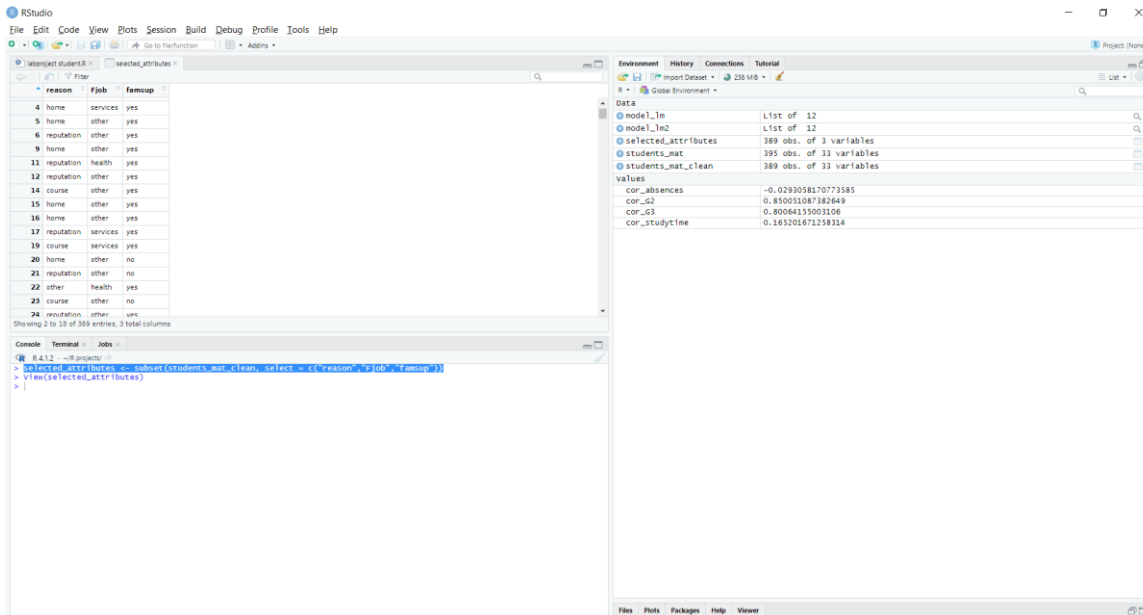
`predict(model_lm2, newdata = data.frame(G2 = 8, G3 = 10))`



10. Select the ("reason","Fjob","famsup") attributes then Apply decision tree algorithm to the selected dataset by setting the training dataset as **80%**, the seed as **100**, to predict the **"reason" attribute**. Then Show the decision tree.

To select the "reason", "Fjob", and "famsup" attributes, you can use the subset function:

```
selected_attributes <- subset(students_mat_clean, select = c("reason","Fjob","famsup"))
```



To apply decision tree to predict the "reason" attribute, you can use the rpart function from the "rpart" package:

```
library(caret)
```

```
library(rpart)
```

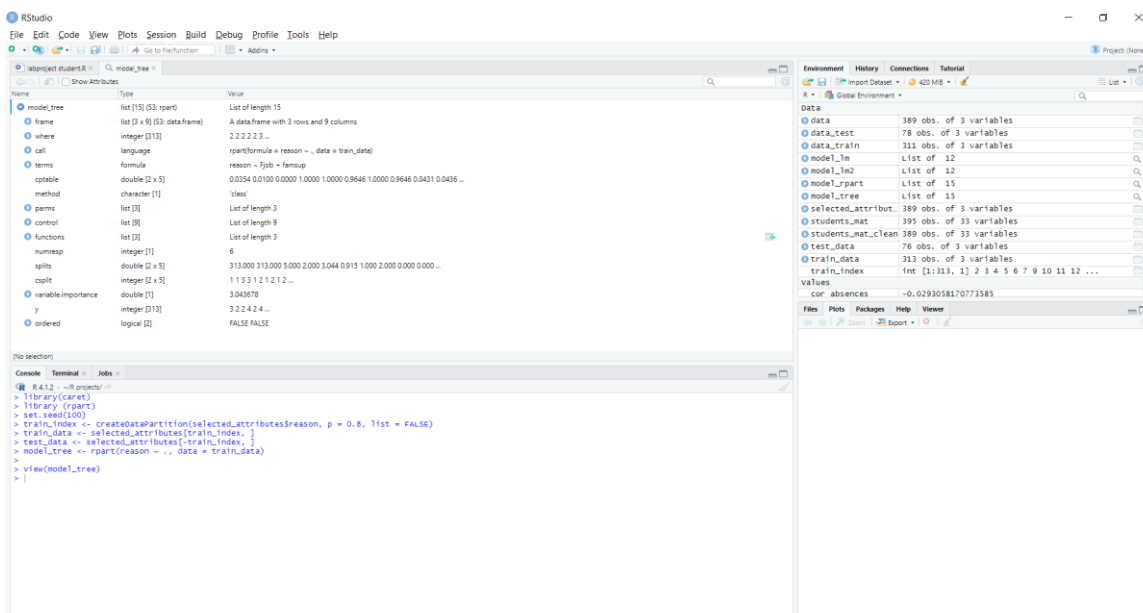
```
set.seed(100)
```

```
train_index <- createDataPartition(selected_attributes$reason, p = 0.8, list = FALSE)
```

```
train_data <- selected_attributes[train_index, ]
```

```
test_data <- selected_attributes[-train_index, ]
```

```
model_tree <- rpart(reason ~ ., data = train_data)
```



To show the decision tree, you can use the plot function from the "rpart.plot" package:

```
library(rpart.plot)
```

```
prp(model_tree)
```

