**About this file**

**Data set taken from :** [**https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/**](https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/)

**Problem Statement**

**About Company**

**Dream Housing Finance company deals in all home loans. They have presence across all urban, semi urban and rural areas. Customer first apply for home loan after that company validates the customer eligibility for loan.**

**Problem**

**Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers.**

## **Read the dataset.**

A Data set is a collection of data. A set is normally presented in a tabular form. Every column describes a particular variable. And each row corresponds to a given member of the data set, as per the given question. This is a part of data management.

Data sets gives the values of each variable for unknown quantities such as height, weight, temperature, volume, etc of an object or values of random numbers. The values in the set are known as a datum. The data set consists of data of one or more members corresponding to each row.

In Statistics, we have different types of data sets available for different types of information. They are:

1. Numerical data sets
2. Bivariate data sets
3. Multivariate data sets
4. Categorical data sets
5. Correlation data sets

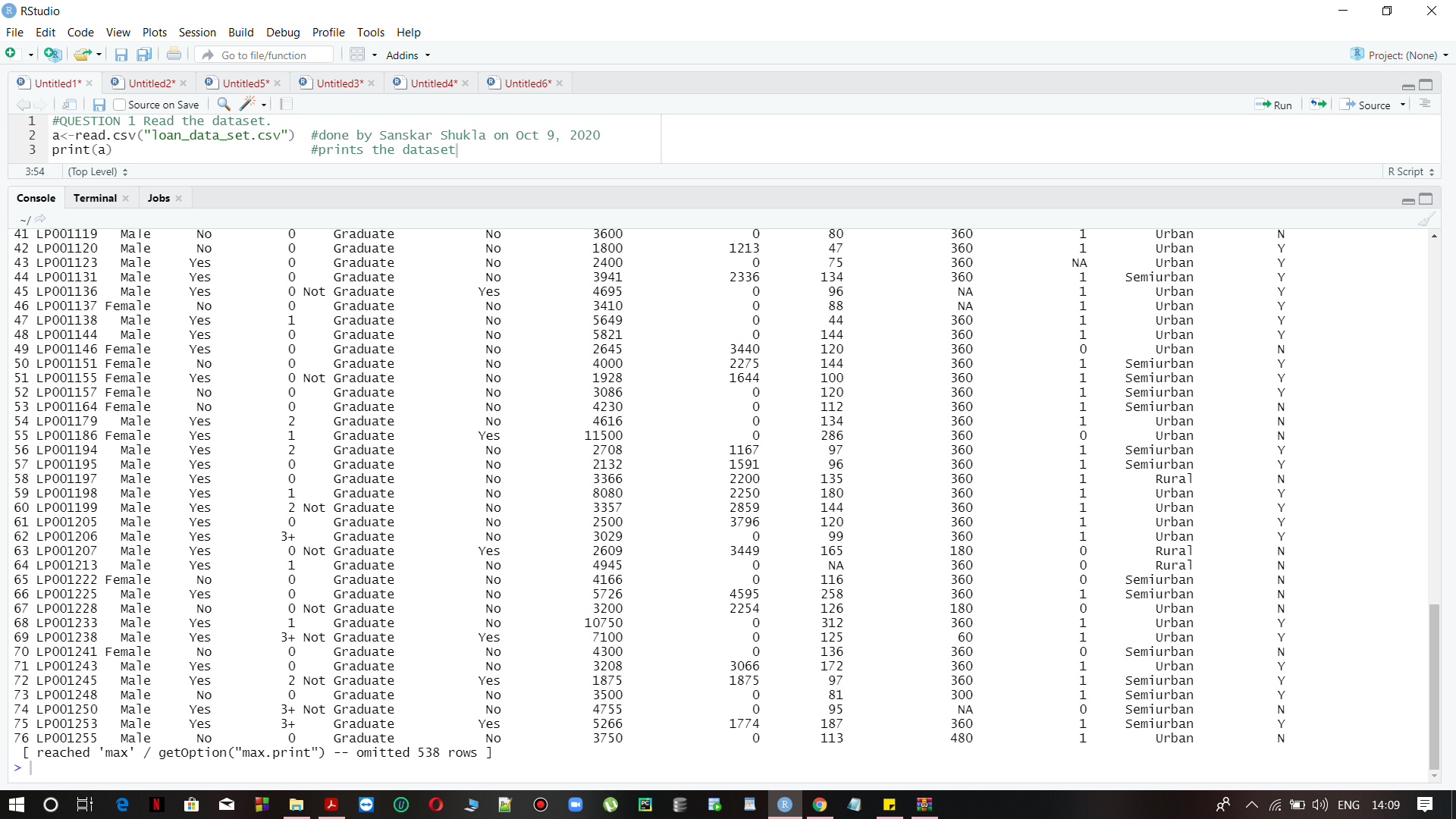
**IN THE GIVEN QUESTION THIS IS A DATA SET OF A COMPANY CALLED DREAM HOUSING FINANCE WHICH DEALS IN HOME LOANS.**

**Program:**

a<-read.csv("loan\_data\_set.csv")

print(a) #prints the dataset

**OUTPUT:**



## **Describe the dataset.**

* In the Loan Id column, there are about 614 unique values starting with code LP00xx and all unique values are assigned to different code(numbers).
* In the Gender section, there are both Male and Female, but in some columns there data is empty there. In this data set, out of 614 applicants, 80% are male , 18% are female and 2% are in other category.
* In the Married section data included in the form of YES and NO only.....and 3 columns are empty.
* In the Dependents column, the data is given in the form of 0s and 1s and >1 which comes under others category and includes 2 and 3+. In this, 56% are in 0s, 17% are in 1s and 27% includes the others category.
* In the Education column ....the whole section is divided into two categories i.e, Graduates and Non-Graduates. Here 78% are Graduates and 22% are Non-Graduates out of 614 applicants
* The Self Employed section has data in the form of YES and NO..with the around 32 empty columns which counts for 5% of the total applicants.
* In the sections of CoapplicantIncome , LoanAmount the data is in the numerical form with different values(amount).
* In CreditHistory the data is divided into 0s and 1s only.
* The category of PropertyArea is divided into three, which are 38% of Semi urban , 33% of Urban and 29% of others that includes only Rural area.
* The Loan Status has been divided into Y and N which counts for YES and No out of the total applicants. And the Loan Term section has counts of no. of months.

## **Find mean,median and mode of columns.**

**Program:**

#reads the csv file

a<-read.csv("loan\_data\_set.csv")

#for ApplicantIncome

print(mean(a$ApplicantIncome,na.rm = FALSE))#mean

print(median(a$ApplicantIncome,na.rm = FALSE))#median

#mode

getmode <- function(v) {

uniqv <- unique(v)

uniqv[which.max(tabulate(match(v, uniqv)))]

}

v <- a$ApplicantIncome

result <- getmode(v)

print(result)

#for CopplicantIncome

print(mean(a$CopplicantIncome,na.rm = FALSE))#mean

print(median(a$CopplicantIncome,na.rm = FALSE))#median

#mode

getmode <- function(v) {

uniqv <- unique(v)

uniqv[which.max(tabulate(match(v, uniqv)))]

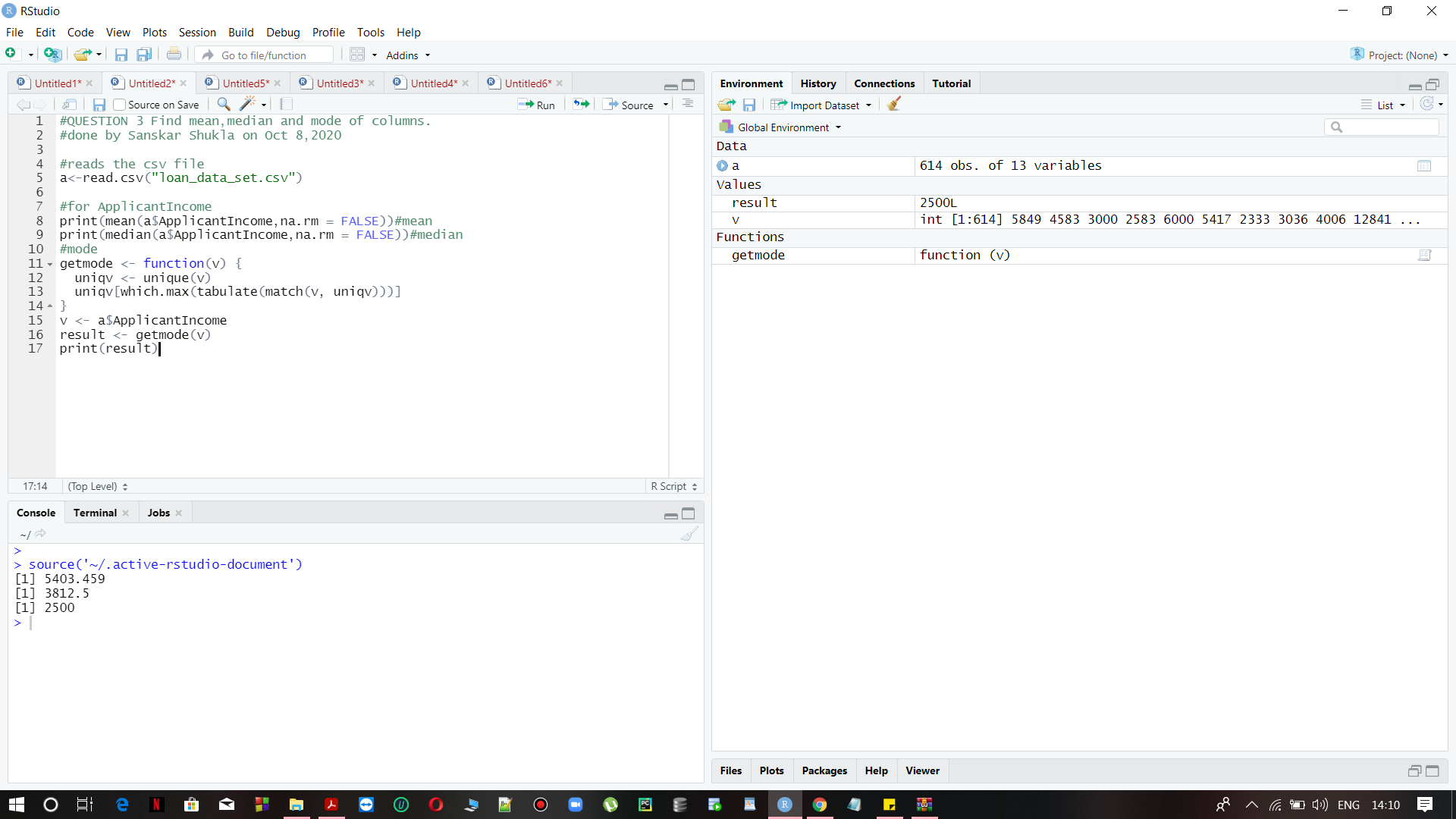
}

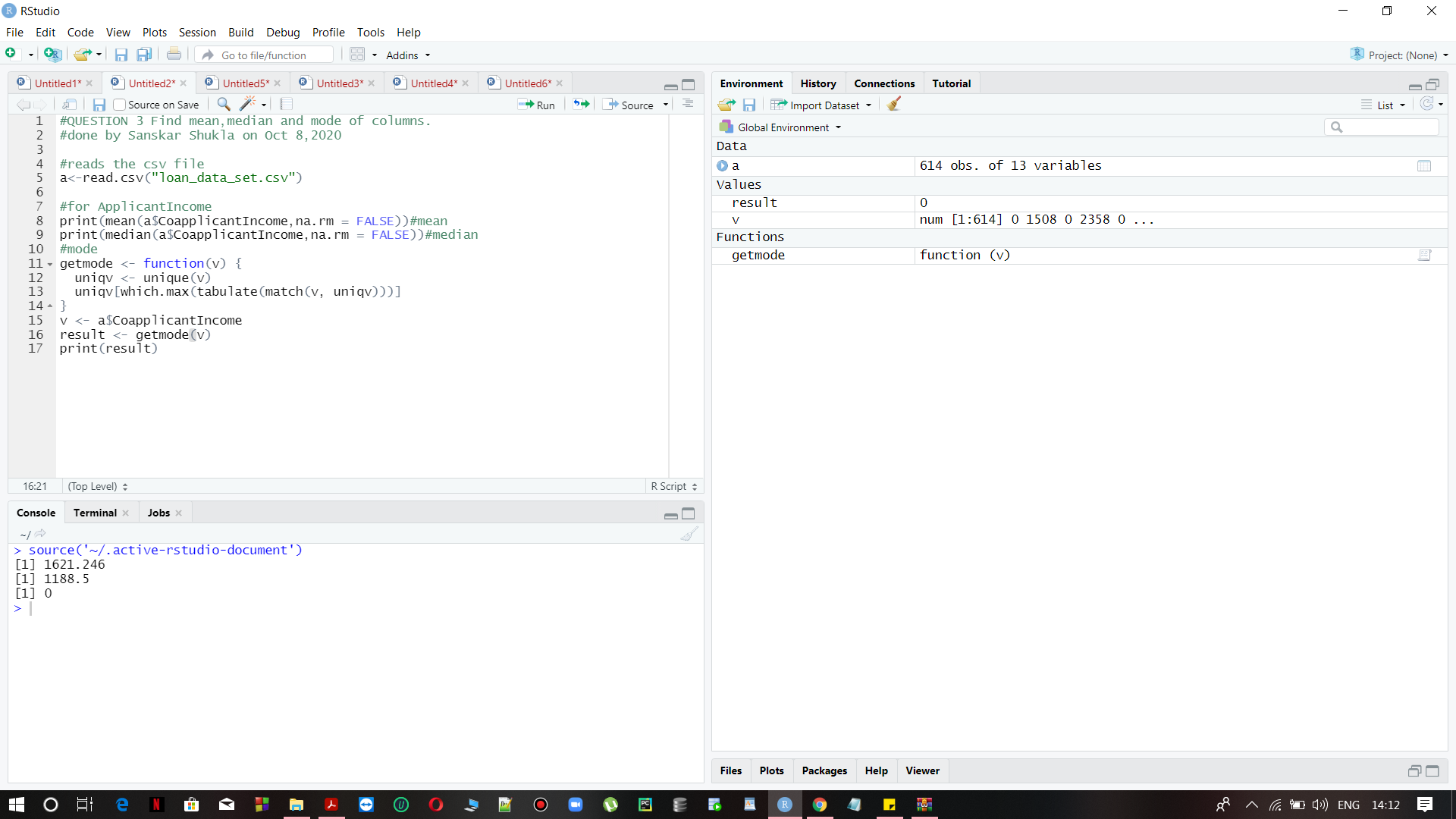
v <- a$CopplicantIncome

result <- getmode(v)

print(result)

**OUTPUT:**





## **Find the distribution of columns with numerical data. Evaluate if they are normal or not.**

**YES IT IS!!!**

In order to get the distribution of columns we will plot a density histogram and check if it forms a bell shaped curve.

readfile <- read.csv(loan\_data\_set.csv') #reading a csv file

**PROGRAM:**

applicantincome <- readfile$Values #getting values stored in the header 'Values'

mean(a$CopplicantIncome,na.rm = FALSE)

sd(applicantincome) #calculates the standard deviation

[1] 5403.459

[1] 6109.042

So from 68-95-99.7 rule

The data Between the range Mean-+ 2(Standard deviation) should be 95% of the total data

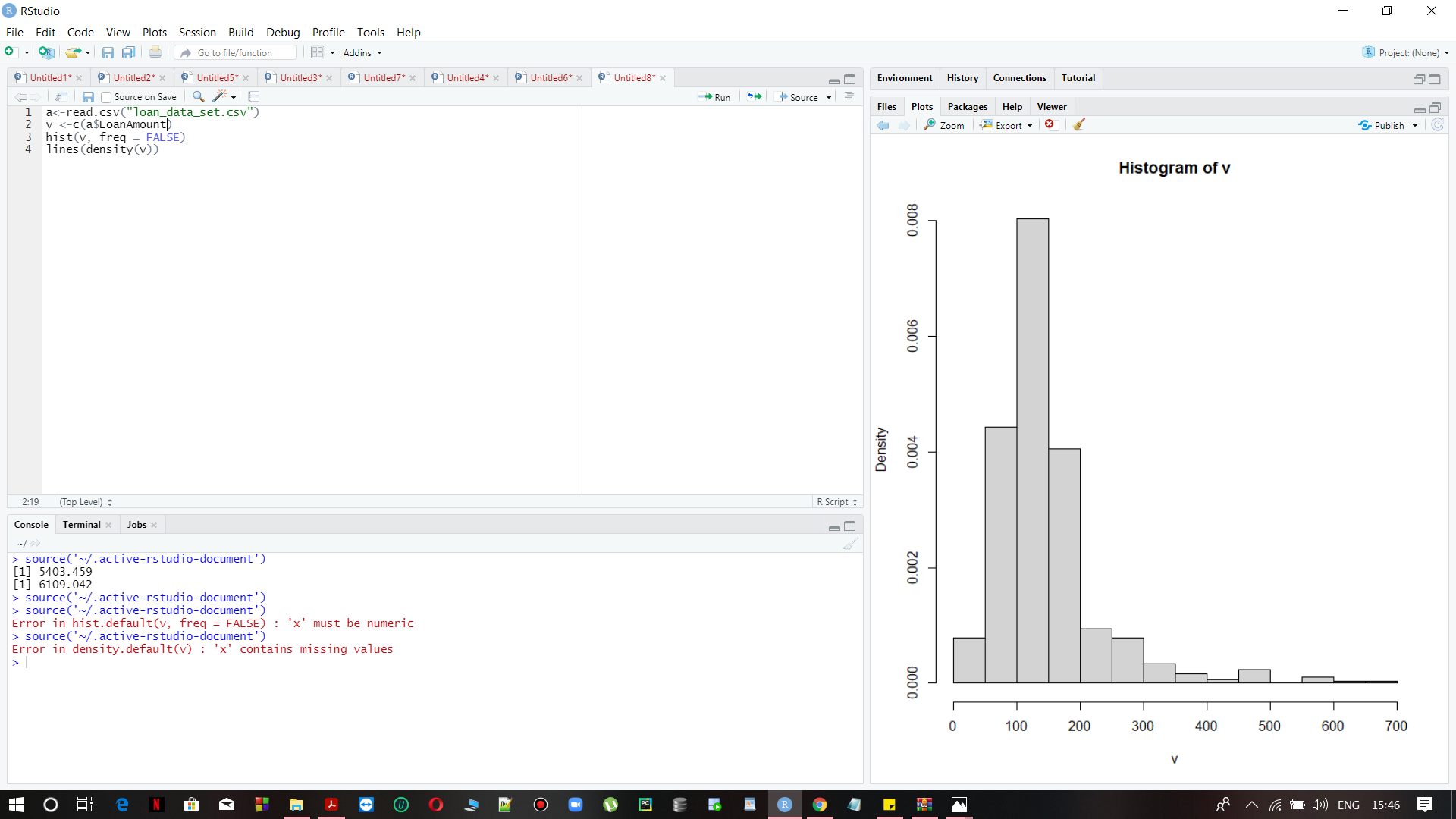
The data Between given range is 598

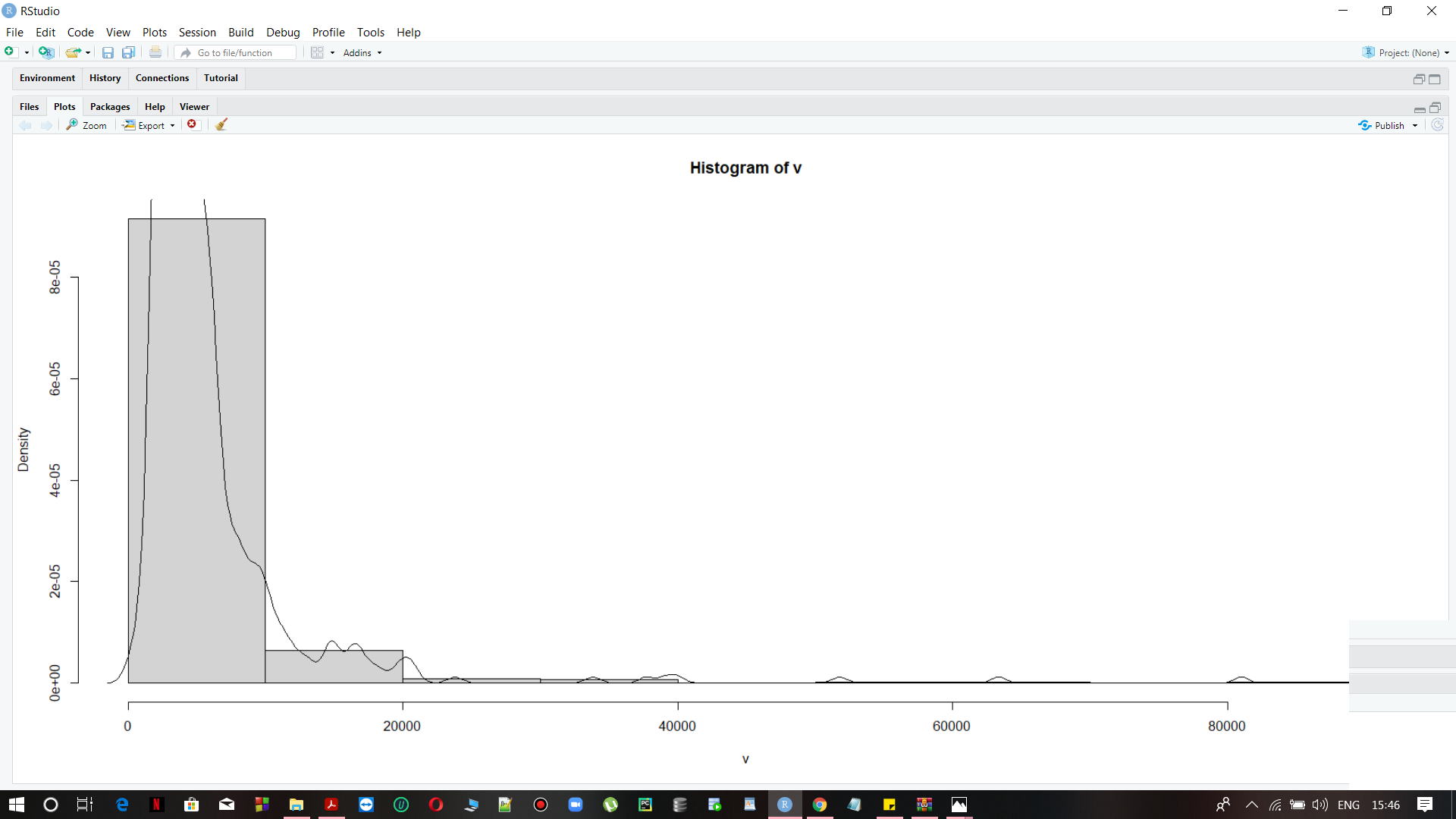
Which is approximately 95% Of total data

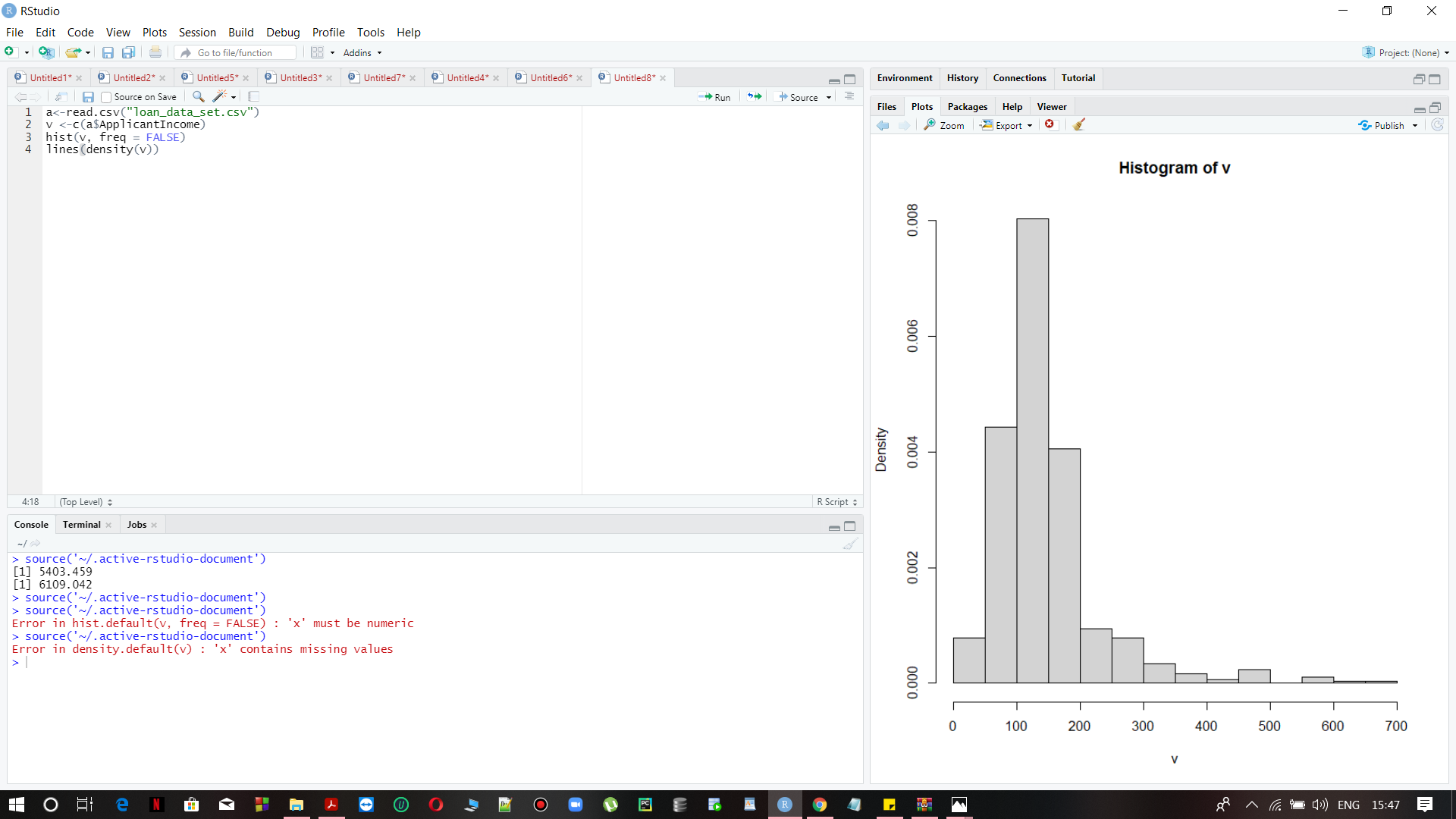
Hence the applicant income is normally distributed

For coapplicant income.

**OUTPUT:**



****

****

## 

**Clearly ApplicantIncome and Loan Amount are Normally distributed because of the bell shape.**

## **Draw different types of graphs to represent information hidden in the dataset.**

**PROGRAM 1 TO DRAW A PIE CHART**

a<-read.csv("loan\_data\_set.csv")

m<-nrow(subset(a,a$Dependents==1))

print(m)

f<-nrow(subset(a,a$Dependents==2))

print(f)

n<-nrow(subset(a,a$Dependents>=3))

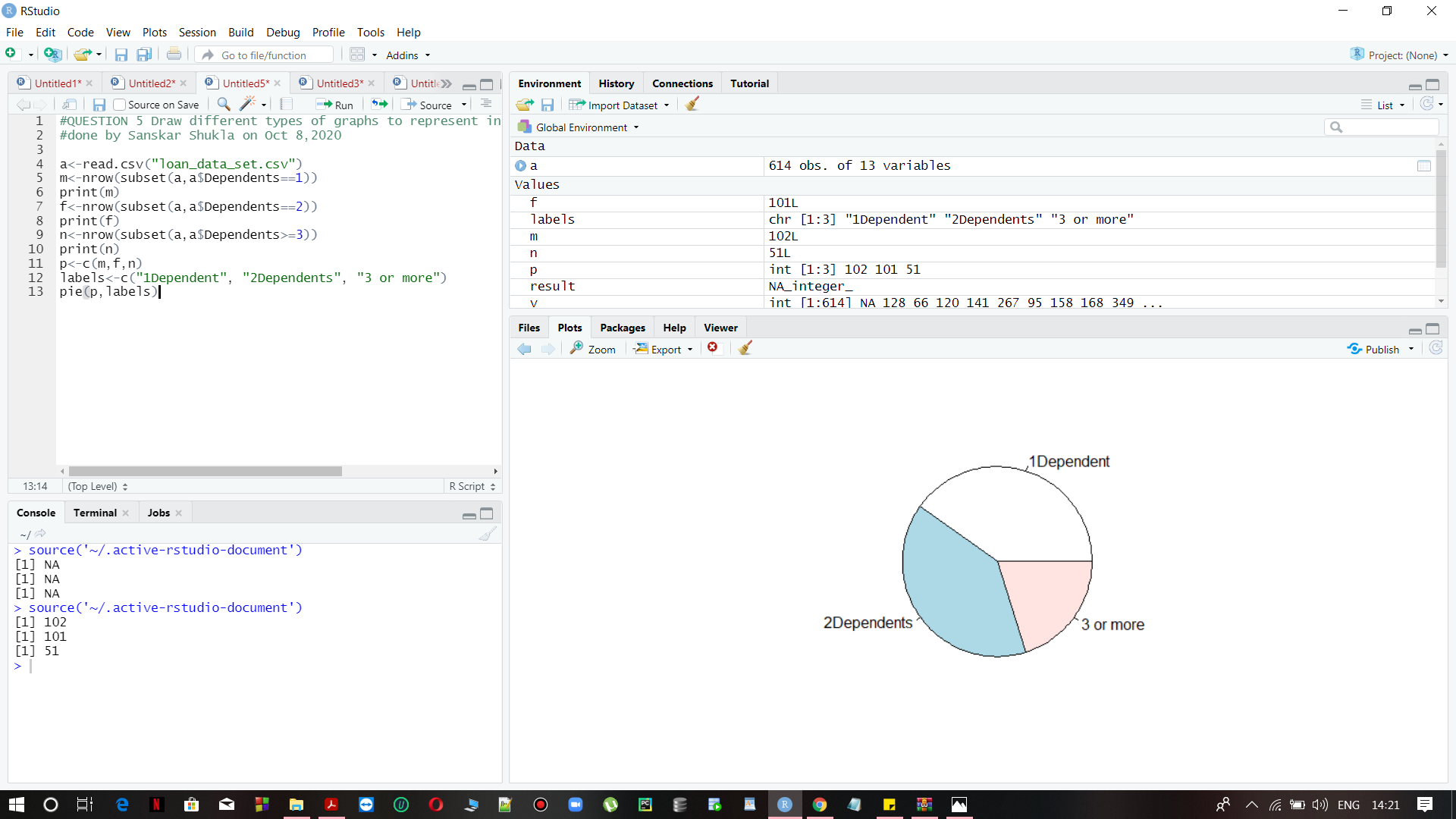
print(n)

p<-c(m,f,n)

labels<-c("1Dependent", "2Dependents", "3 or more")

pie(p,labels)

**OUTPUT:**



**PROGRAM 2 TO DRAW A SCATTER PLOT**

a<-read.csv("loan\_data\_set.csv")

plot(x = a$ApplicantIncome,y = a$LoanAmount,

xlab = "Applicant Income",

ylab = "Loan Amount",

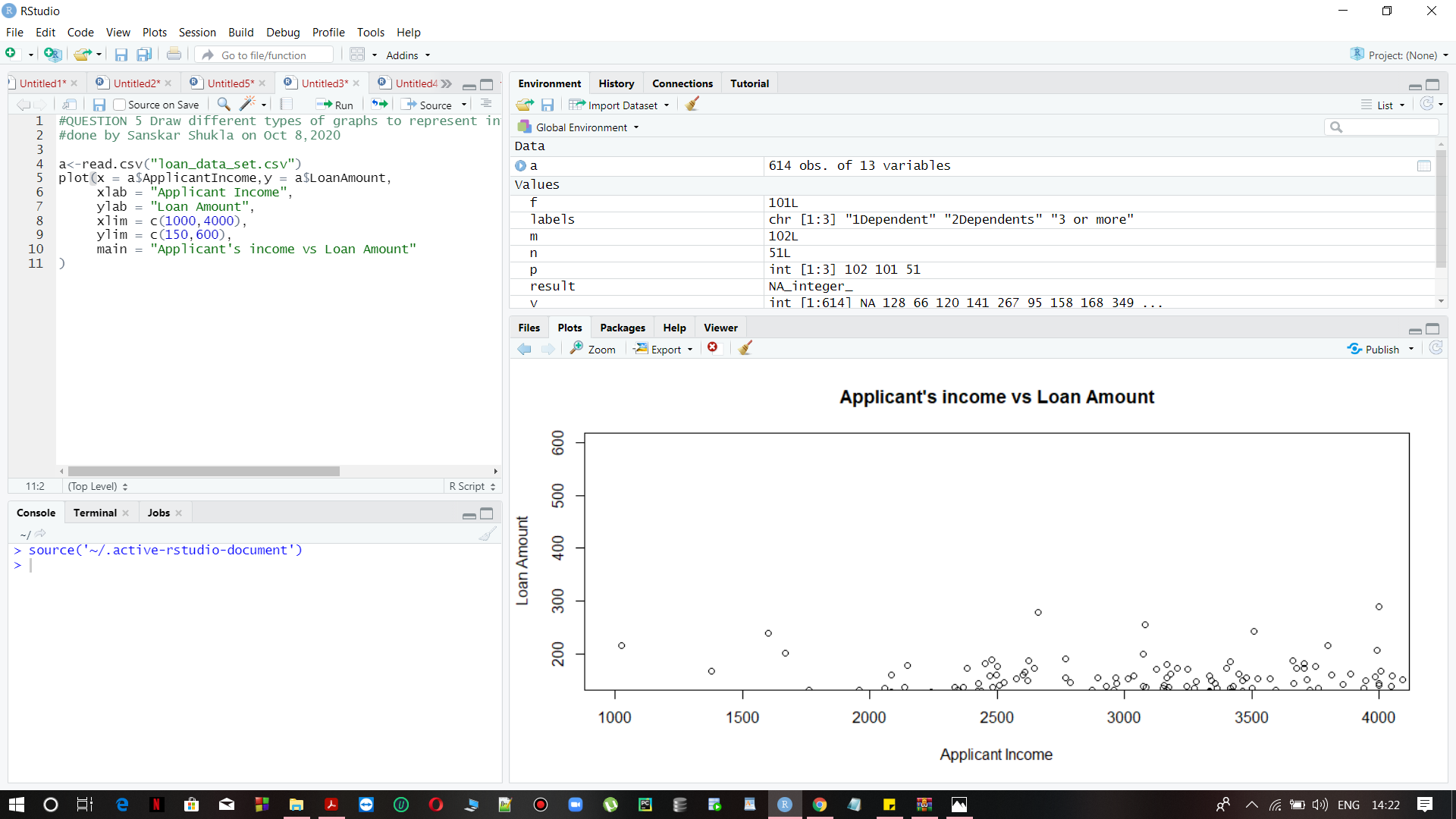
xlim = c(1000,4000),

ylim = c(150,600),

main = "Applicant's income vs Loan Amount"

)

**OUTPUT:**



## **Find columns which are correlated.**

## **Pearson Correlation**

The p-values (significance level) of the correlation can be determined by :

by using correlation coefficient given in the table for the degrees of freedom :

df=n−2, here

n is the number of the observations in the x and y variables or by calculating the t values as given under:

**t=(r√(n−2))/(1−r2)**

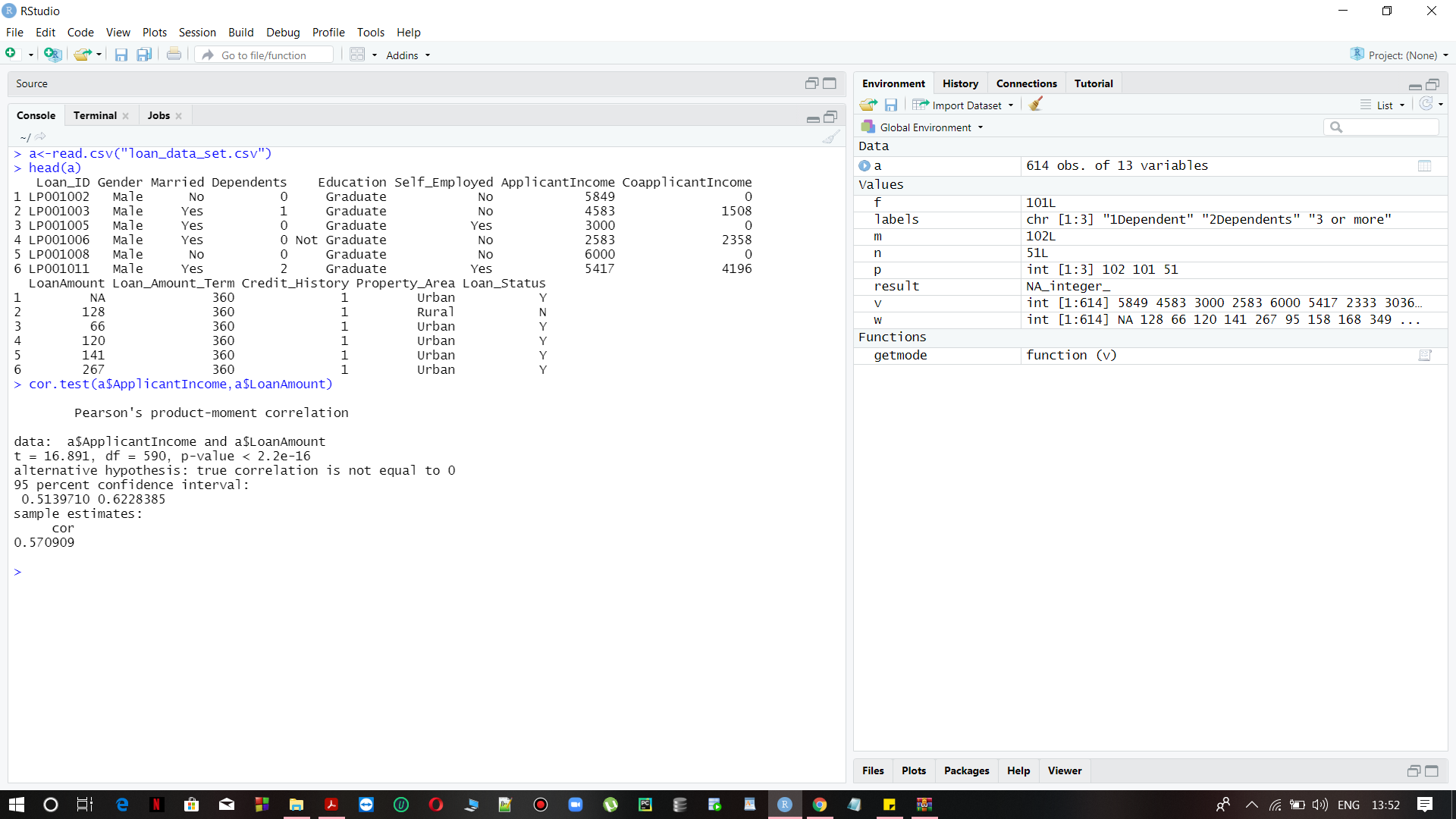
In the case 2) the corresponding p-value is determined using [t distribution table](http://www.sthda.com/english/wiki/t-distribution-table) for

**df=n−2**

**Thus the moderately related attributes are:**

1. **LoanAmount and ApplicantIncome**

**OUTPUT:**

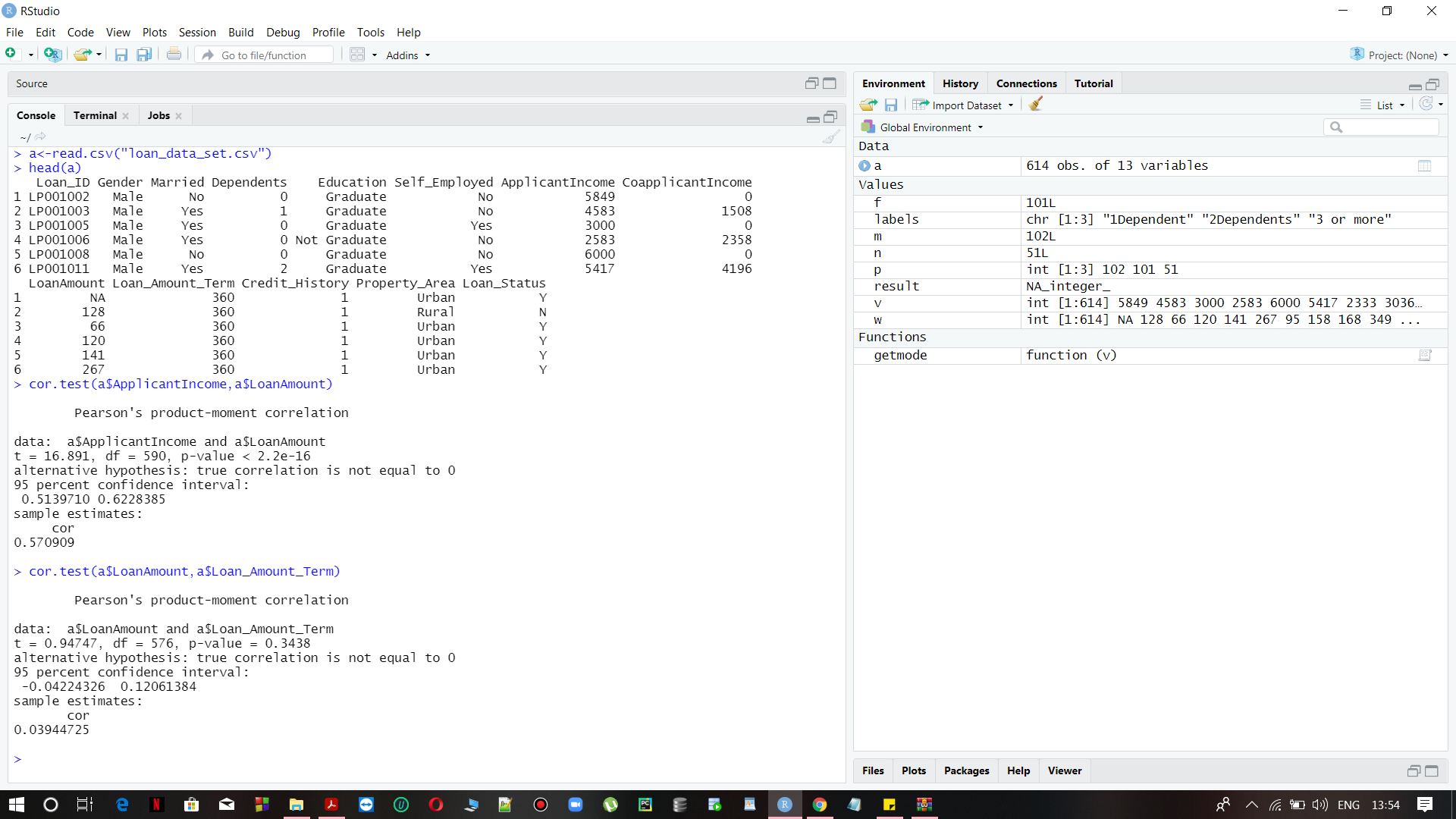
****

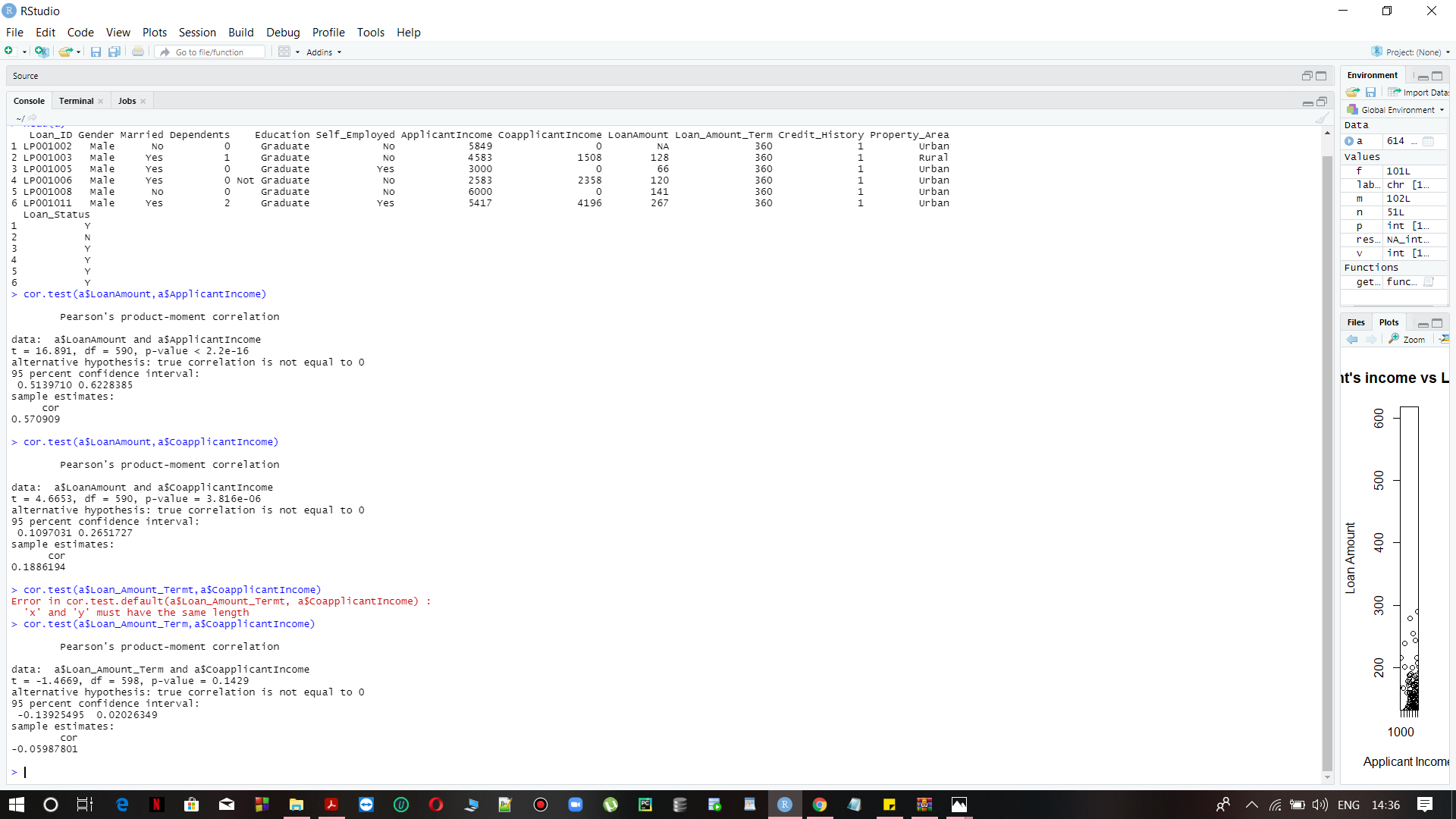
## **Find columns which are not correlated.**

**The non relating attributes are :**

1. **ApplicantIncome and CoapplicantIncome**
2. **CoapplicantIncome and LoanAmount**
3. **LoanAmount and Loan\_Amount\_Term**
4. **ApplicantIncome and Loan\_Amount\_Term**

**OUTPUT:**

****

****

## **Compare different columns of the dataset.**

To compare any set of columns in a dataset is not an easy task. Since datasets contain huge amounts of values.. We can take help of visualization in order to distinguish the attributes.

**PROGRAM:**

a<-read.csv("loan\_data\_set.csv")

v <-c(a$ApplicantIncome)

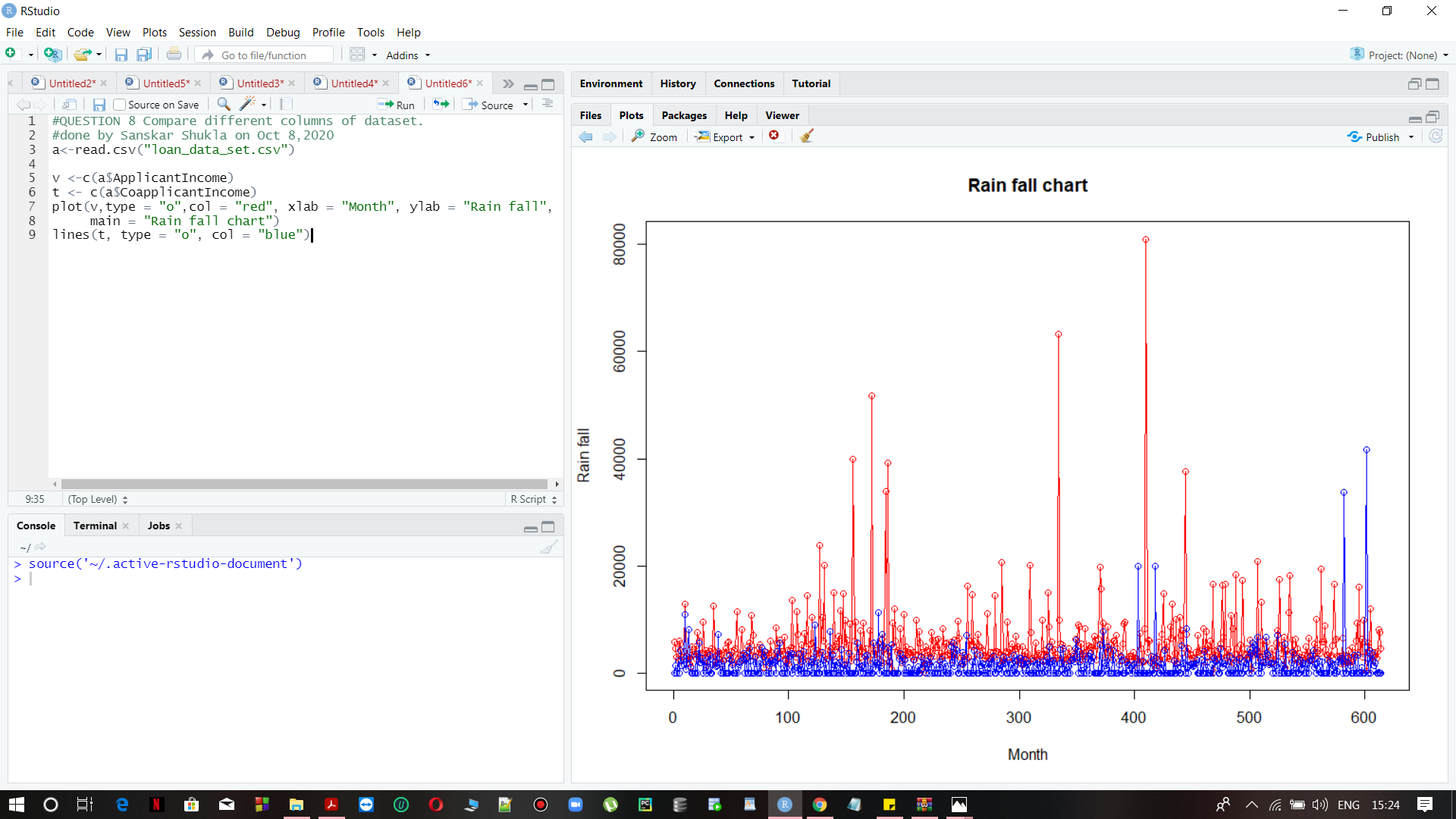
t <- c(a$CoapplicantIncome)

plot(v,type = "o",col = "red", xlab = "Month", ylab = "Rain fall",

main = "Rain fall chart")

lines(t, type = "o", col = "blue")

**OUTPUT:**

****

## **Is Any supervised machine learning possible ? if yes explain.**

First we ought to describe the supervised learning.

**Supervised Machine Learning**

The majority of sensible machine learning uses the supervised learning.

Supervised learning is wherever you've got input variables (x) associate degreed associate degree output variable (Y) and you employ the formula to find out the mapping performed the input to the output.

Y = f(X)

The goal is to approximate the mapping perform thus well that once you have new knowledge|input file|computer file} (x) that you simply will predict the output variables (Y) for that above data.

It is referred to as supervised learning as a result of method {the method} of associate degree formula learning from the coaching dataset will be thought of as a coach overseeing the training process. we all know the proper answers, the formula iteratively makes predictions on the coaching information and is corrected by the teacher. Learning stops once the formula achieves a suitable level of immeasurable performance.

* **Yes, definitely the supervised learning is possible in the above given dataset.**
* **We can perform Naive Bayes Algorithm for the given dataset.**
* **Note: The scatter plot of this dataset can be used for predictive analysis for banks in order to sanction loans and analyse the bank db for modelling.**

Naive Bayes is a Supervised Machine Learning algorithm which is based on the Bayes Theorem that is used to solve **classification problems** by following the probabilistic approach. It is based on the very idea of predictor variables in a [Machine Learning](https://www.edureka.co/blog/what-is-machine-learning/) model are independent of each other as we know it. Meaning that the outcome of a model depends on a set of independent variables that have nothing to do with each other.

**SAMPLE OUTPUT FROM OUR CONSOLE:**

# convert some numeric variables to factors

attrition <- attrition %>%

mutate(

JobLevel = factor(JobLevel),

StockOptionLevel = factor(StockOptionLevel),

TrainingTimesLastYear = factor(TrainingTimesLastYear)

)

# Create training (70%) and test (30%) sets for the attrition data.

# Use set.seed for reproducibility

set.seed(123)

split <- initial\_split(attrition, prop = .7, strata = "Attrition")

train <- training(split)

test <- testing(split)

# distribution of Attrition rates across train & test set

table(train$Attrition) %>% prop.table()

##

## No Yes

## 0.838835 0.161165

table(test$Attrition) %>% prop.table()

##

## No Yes

## 0.8386364 0.1613636