

R: Advanced

Datafiles

For these exercises, download the files:

- “Business Analytics – Week 9 Instructions.doc”
- “Business Analytics – Week 9 Universal Bank.xls”
- “Business Analytics – Week 9 creditset.xls”

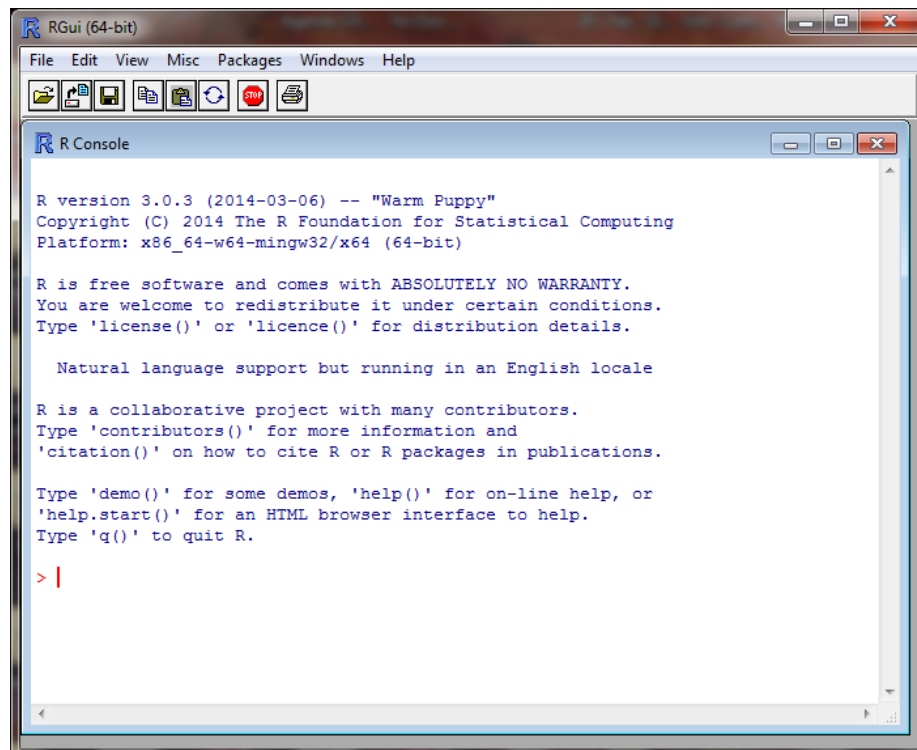
Installation of R

R is a free downloadable package capable of performing sophisticated statistical analysis and data mining. The software is already installed on the classroom laptops. To install on your own personal computer:

1. Go to the website: <http://cran.r-project.org/bin/windows/base/>
2. For a Mac, go to <http://cran.r-project.org/bin/macosx/>
3. Click on Download R 3.0.3 for Windows
4. Click on Run, and follow the install instructions

Starting R

1. Click on the Start button in the lower left corner of Windows
2. Click on All Programs, then click on the R folder, then R

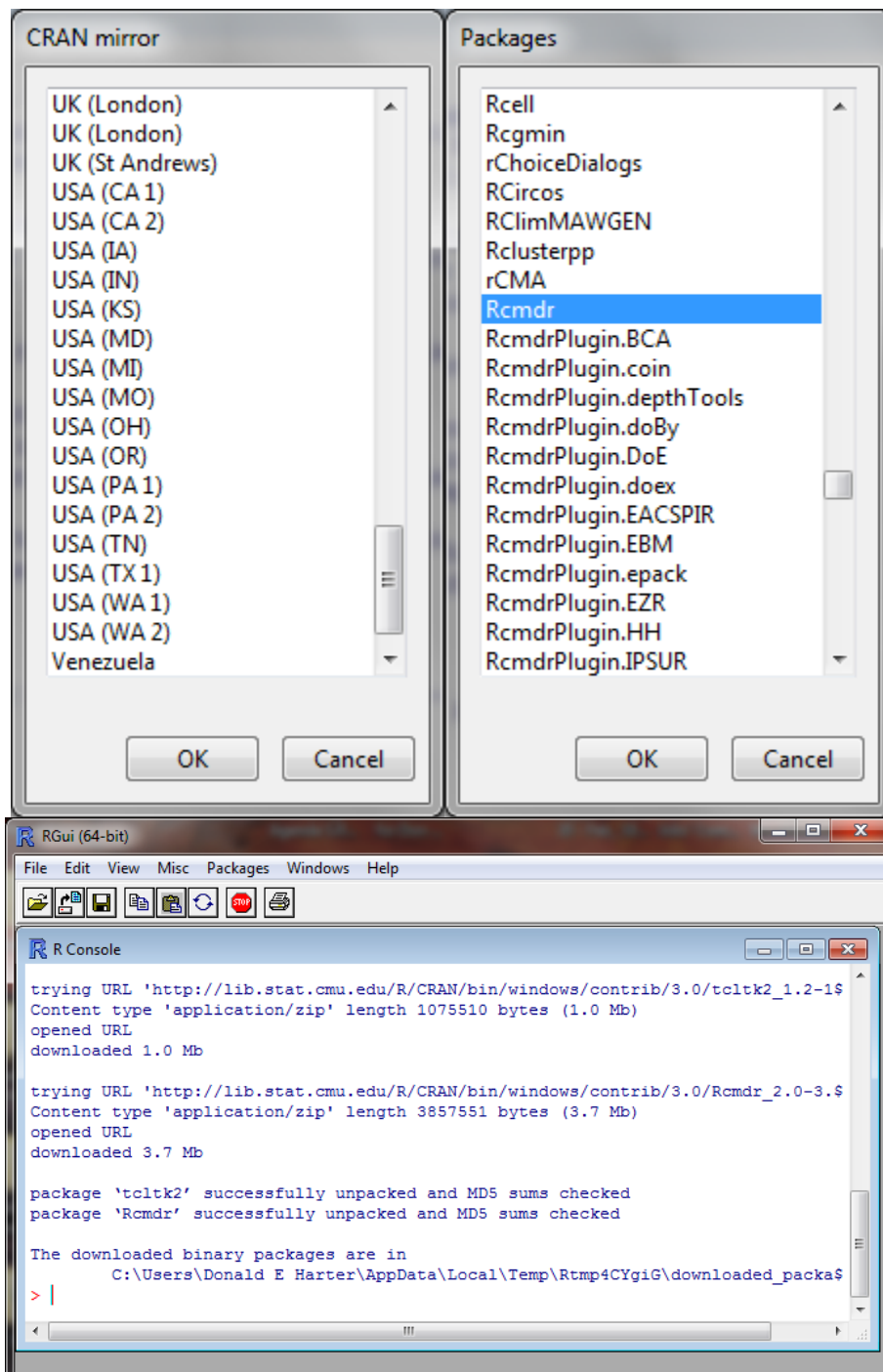


This is the command line screen. You can enter commands, but need to know the syntax. There is a simpler approach to running R, called Rcmdr (R Commander). If you are running a Whitman computer, Rcmdr is already installed. If not, you need to install it.

Installing R Commander

Follow these steps only if you don't already have Rcmdr installed.

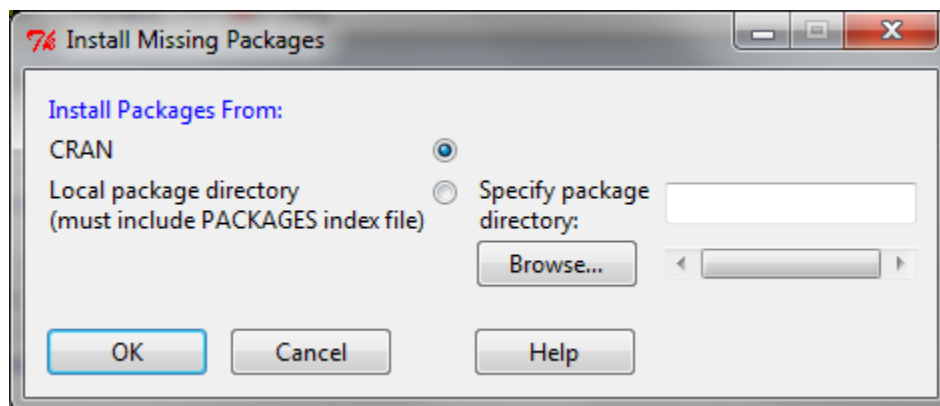
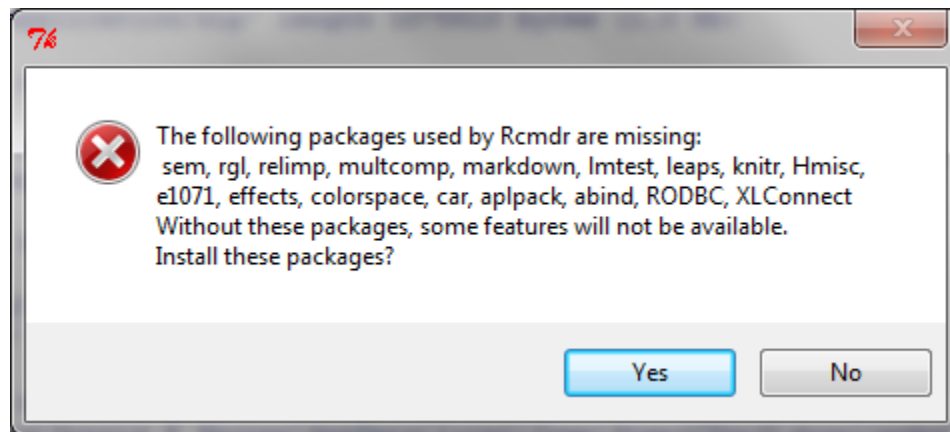
1. At the top of the screen, click on Packages
2. In the drop down menu, click on Install Package(s)
3. In the CRAN mirror, select the location closest to you; use USA (PA 1), then click OK
4. In the Packages screen, click on Rcmdr, then OK
5. When prompted to create a personal library, click Yes

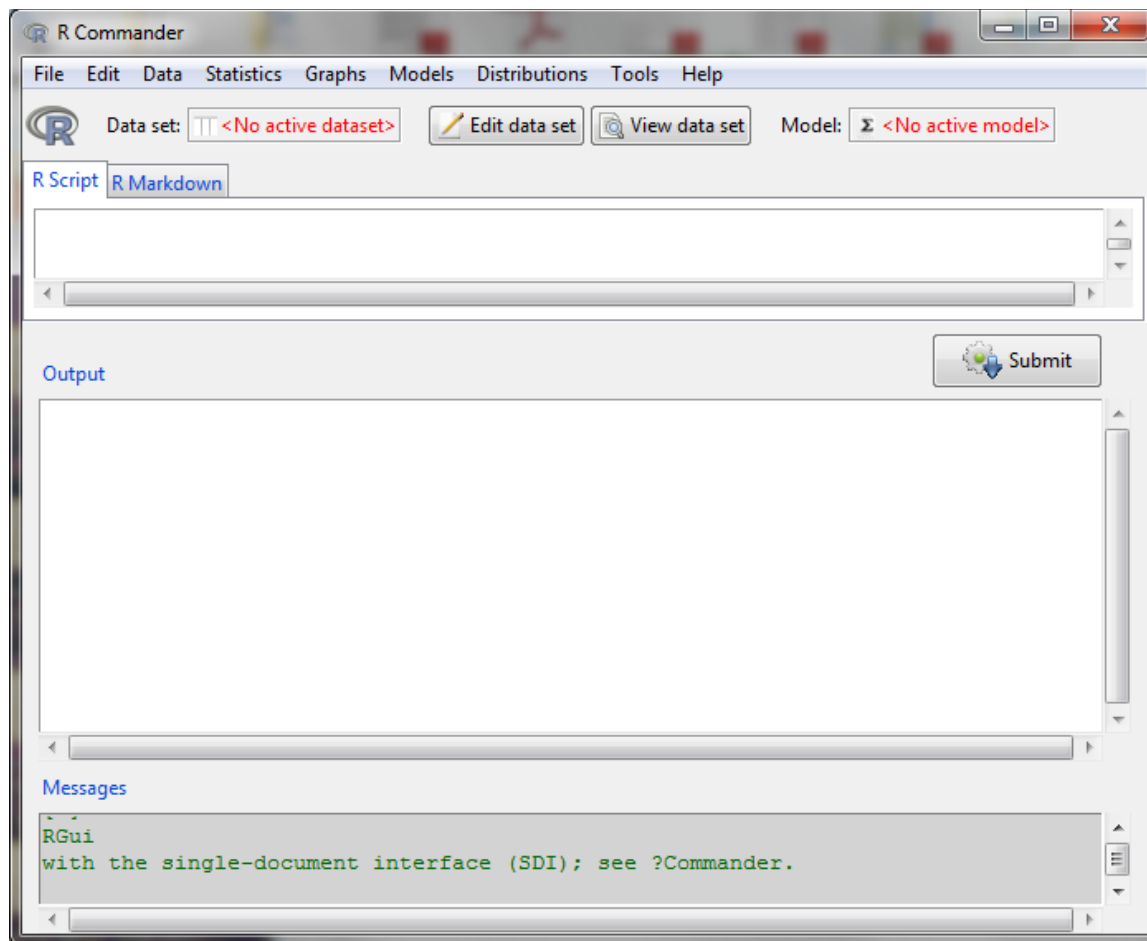


Launch Rcmdr (R Commander)

Rcmdr is a graphical user interface (GUI) that is easier to use than the command line. To launch Rcmdr:

1. Type library(Rcmdr)
2. If you receive a warning message that some packages are missing, it will ask if you want them installed. Click Yes.
3. On the Install Missing Packages screen, click OK
4. R will install the necessary software
5. The R Commander screen will appear





Session 9.4: Logit Analysis - Download Datasets

To access some excellent data sets used in the book “Data Mining and Business Analytics with R,” by Johannes Ledolter:

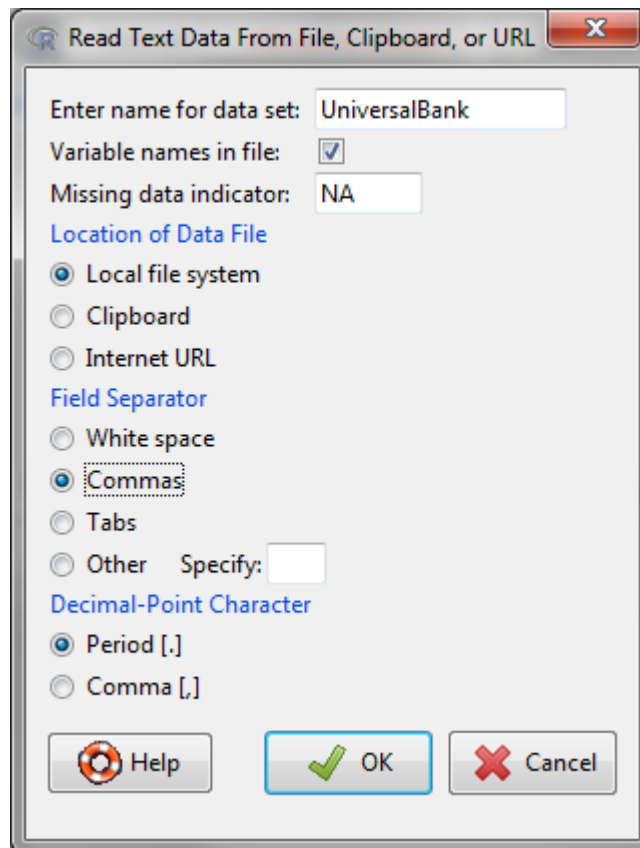
1. Go to the website:
<http://www.biz.uiowa.edu/faculty/jledolter/DataMining>
2. Click on Data Text
3. Right click on UniversalBank.csv, then save on your computer
4. Remember where you saved the file

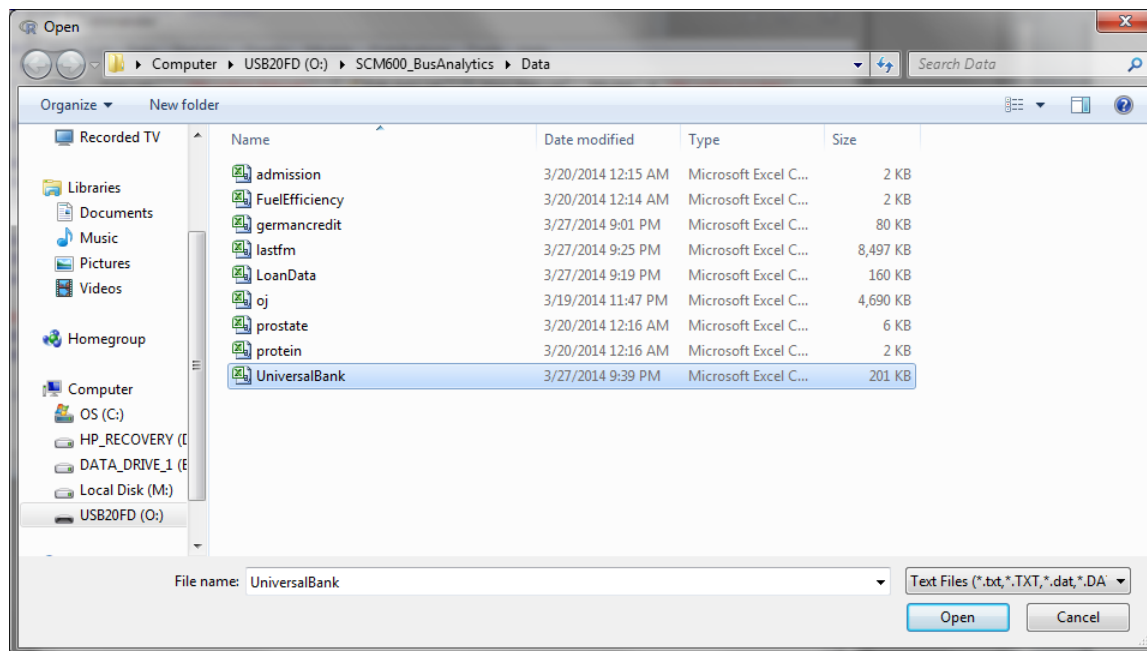
The Business Analytics - Week 9 Universal Bank.csv file can be downloaded from the course website.

Loading Data

To load data into R:

1. Click on Data at the top of the screen
2. Click on Import Data > From text file ...
3. Enter the name that you would like to use for this data set; type in UniversalBank
4. Change Field Separator to Commas, then OK
5. Click on the UniversalBank file, then Open





Note that the dataset UniversalBank has 5000 rows and 14 columns.

Viewing data fields

This data set lists loan characteristics for 5000 loan applications. Let's view the data. The easiest way to view is simply by opening the original Excel spreadsheet. Find the spreadsheet UniversalBank.csv that you downloaded and double click on it. The variables are defined below.

PersonalLoan: 0 for did not take loan, 1 if took loan

Age: age of customer

Experience: professional experience of customer

Income: income of customer

Family: family size of customer

CCAvg: average monthly credit card spending

Education: three categories (undergraduate, graduate, professional)

Mortgage: size of mortgage

SecuritiesAccount: No/Yes (0,1)

CDAccount: No/Yes (0,1)

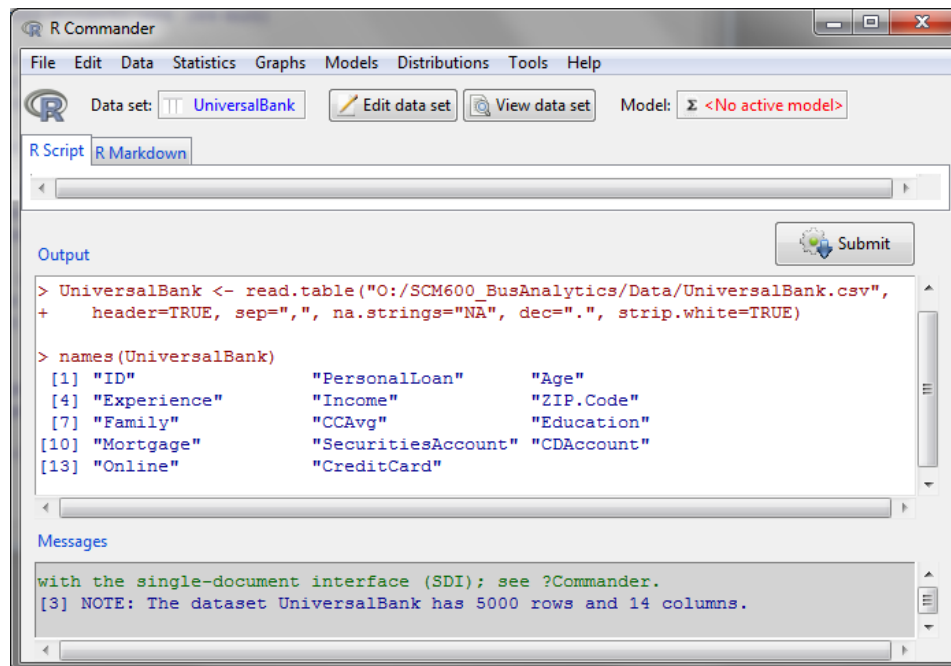
Online: No/Yes (0,1)

CreditCard: No/Yes (0,1)

ID	PersonalLoan	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Securities	CDAccount	Online	CreditCard
1	0	25	1	49	91107	4	1.6	1	0	1	0	0	0
2	0	45	19	34	90089	3	1.5	1	0	1	0	0	0
3	0	39	15	11	94720	1	1	1	0	0	0	0	0
4	0	35	9	100	94112	1	2.7	2	0	0	0	0	0
5	0	35	8	45	91330	4	1	2	0	0	0	0	1
6	0	37	13	29	92121	4	0.4	2	155	0	0	1	0
7	0	53	27	72	91711	2	1.5	2	0	0	0	1	0
8	0	50	24	22	93943	1	0.3	3	0	0	0	0	1
9	0	35	10	81	90089	3	0.6	2	104	0	0	1	0
10	1	34	9	180	93023	1	8.9	3	0	0	0	0	0
11	0	65	39	105	94710	4	2.4	3	0	0	0	0	0
12	0	29	5	45	90277	3	0.1	2	0	0	0	1	0

Now return to R. To view the variables in R,

1. Click on Data, Active Data Set, Variables in Active Data Set

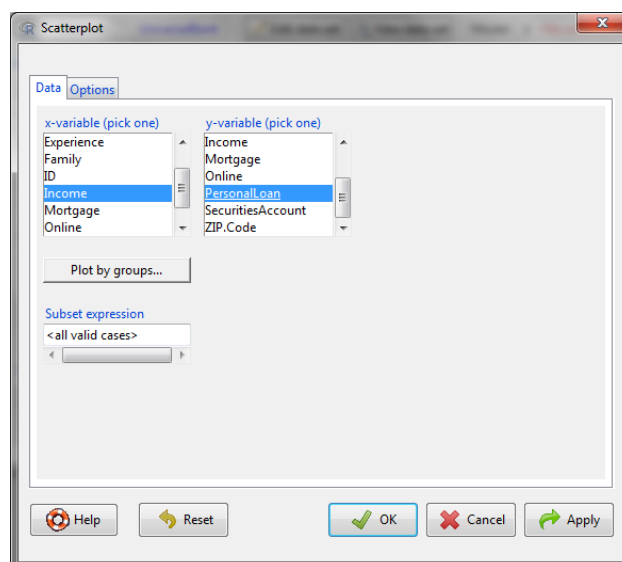


Notice that R generates the command names(UniversalBank). This is the command line version.

Scatterplots

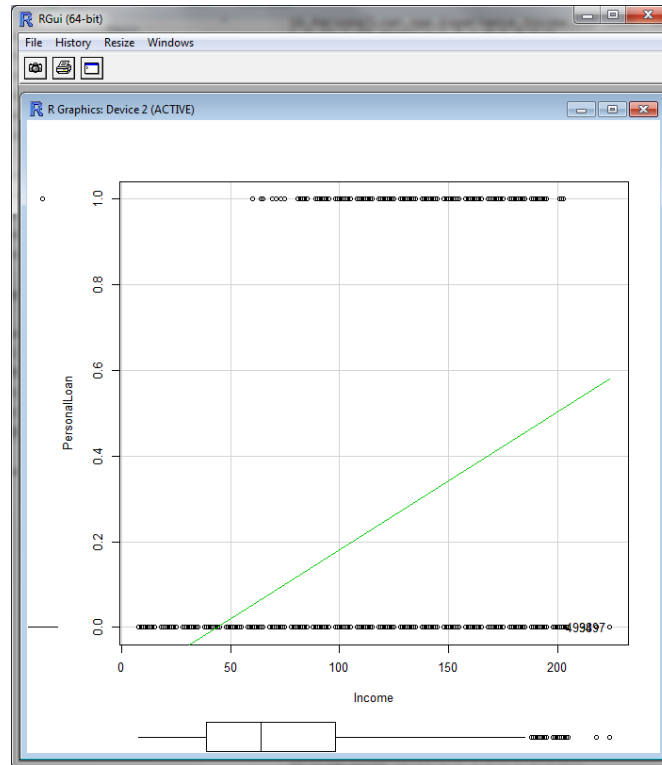
To generate a scatter plot,

1. Click on Graphs, Scatterplot
2. Select Income as the x-variable
3. Select PersonalLoan as the y-variable
4. Click on OK

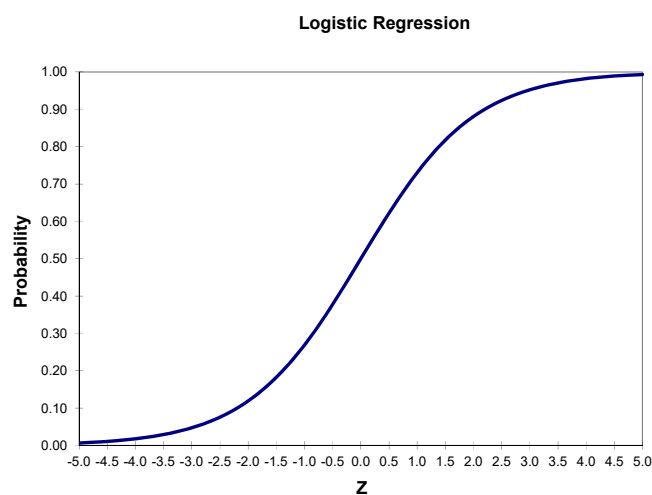


To interpret the chart:

1. The black dots are the Income versus Loan (1) or No Loan (0)
2. The green line is the linear regression line through the data
3. Does the linear regression line make sense?



Linear regression assumes that there is a linear relationship between the X and Y variables. In this case, that doesn't make sense. A better solution looks like this:



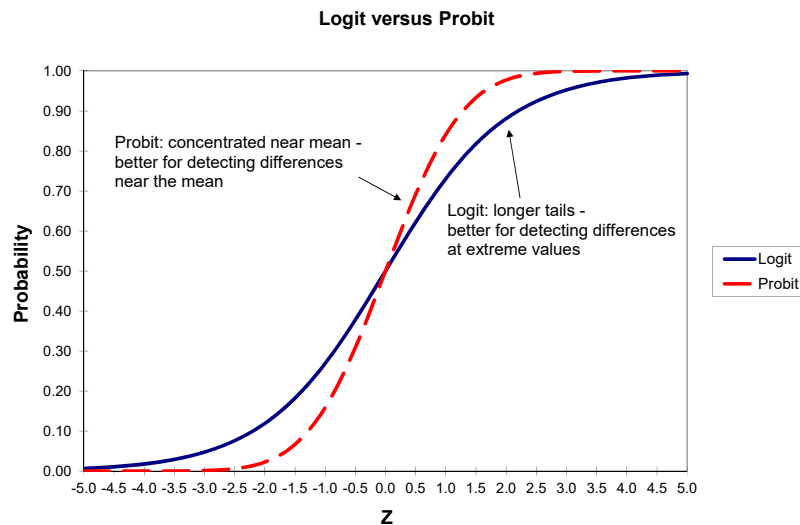
Logit and probit are techniques that assume the dependent variable (Y) is zero or one, and finds the relationship between the explanatory variables (X) and the dependent variable (Y). Logistic regression and logit are based on the logistic distribution. Probit is based on the normal distribution. Logit is more sensitive to extreme values of the X variable. Probit is more sensitive to values near the mean.

The Logit regression uses the logistic function to calculate the probability:

$$P(Y=1) = \exp(\sum \beta_i X_i) / [1 + \exp(\sum \beta_i X_i)]$$

The Probit regression uses the normal distribution to calculate the probability:

$$P(Y=1) = \Phi(\sum \beta_i X_i) \text{ where } \Phi \text{ is the normal distribution}$$



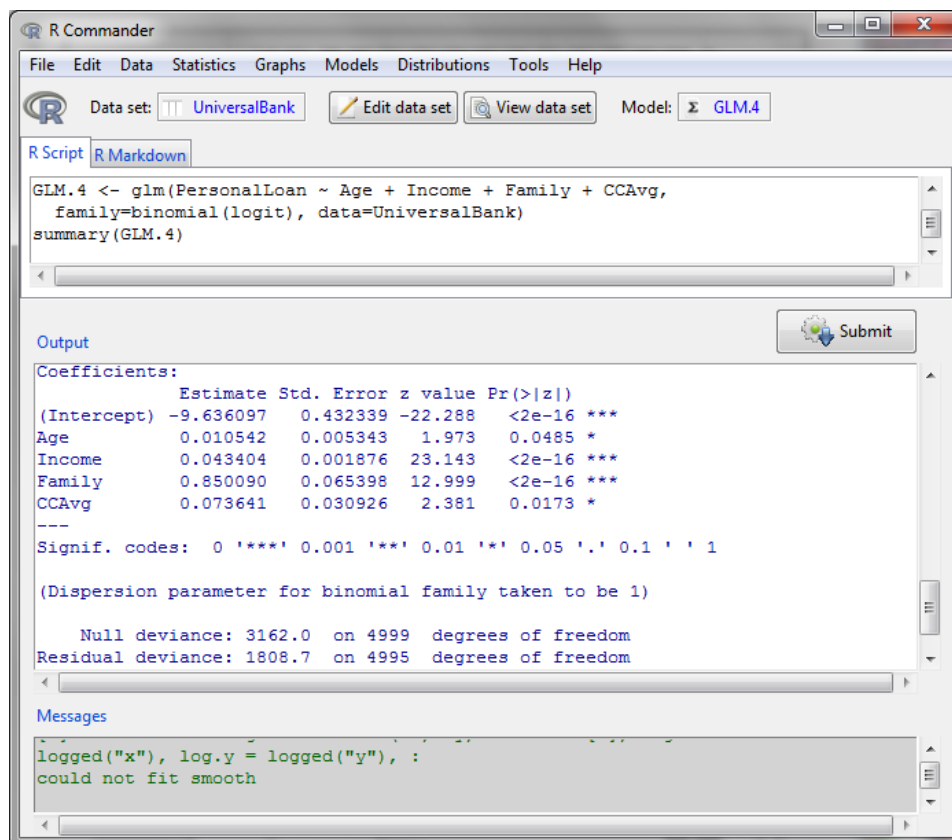
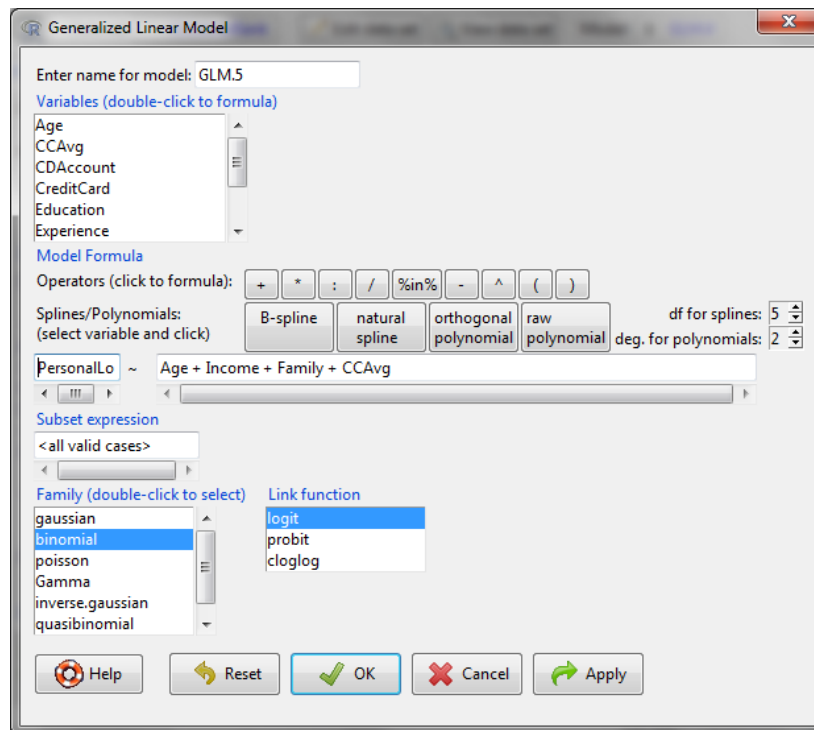
Logit Analysis

To perform a logit analysis on our data, where the Y variable is PersonalLoan and the explanatory variables are age, income, family size, and credit card average balance:

1. Click on Statistics, Fit models, Generalized linear model
2. Double click on PersonalLoan for the dependent variable
3. Double click on Age, Income, Family and CCAvg for the explanatory variables
4. Select binomial family
5. Select link function as logit
6. Click OK

Are the coefficients positive or negative?

Are the coefficients statistically significant?



Session 9.5: Logit Predictions

You can use a spreadsheet to calculate logit probabilities for different combinations of the explanatory variable values. Let's calculate the probability of taking out a loan for someone who is 40 years old, with \$80,000 in income, family of 5, and credit card average balance of \$2,000. Use the data from the previous page.

1. Create the labels in row 1 for Variable, Coefficient, Value, and Coeff*Value
2. In Column A, below Variable, list Intercept and each of the explanatory variables
3. In Column B, below Coefficient, enter the coefficients from your Logit regression
4. In Column C, below Value, enter 1 for intercept and the values that you want to evaluate
 - a. Enter 40 for age
 - b. Since income is in the dataset in thousands, enter 80 for \$80,000
 - c. Enter 5 for family
 - d. Since credit card average balance is in the dataset in thousands, enter 2 for \$2,000
5. In Column D, below Coeff*Value, enter the formula to multiply the coefficient and value; e.g., for cell D3, enter =B3*C3
6. In cell D9, enter the formula for the sum of the column D calculations; i.e., =sum(D3:D7)
7. In cell D10, calculate the exponential of the sum; i.e., =exp(D9)
8. In cell D11, calculate the probability; i.e., =D10/(1+D10)

Vary the age, income, family size, and credit card balance to determine the effect on the probability of taking out a loan.

	A	B	C	D	E	F
1	Variable	Coefficient	Value	Coeff*Value		
2						
3	Intercept	-9.636097	1	-9.636097		
4	Age	0.010542	40	0.42168		
5	Income	0.043404	80	3.47232		
6	Family	0.850090	5	4.25045		
7	CCAvg	0.073641	2	0.147282		
8						
9			Sum	-1.344365		
10			Exp(sum)	0.260705203		
11			Probability	0.206793152		

Session 9.6: Probit Analysis

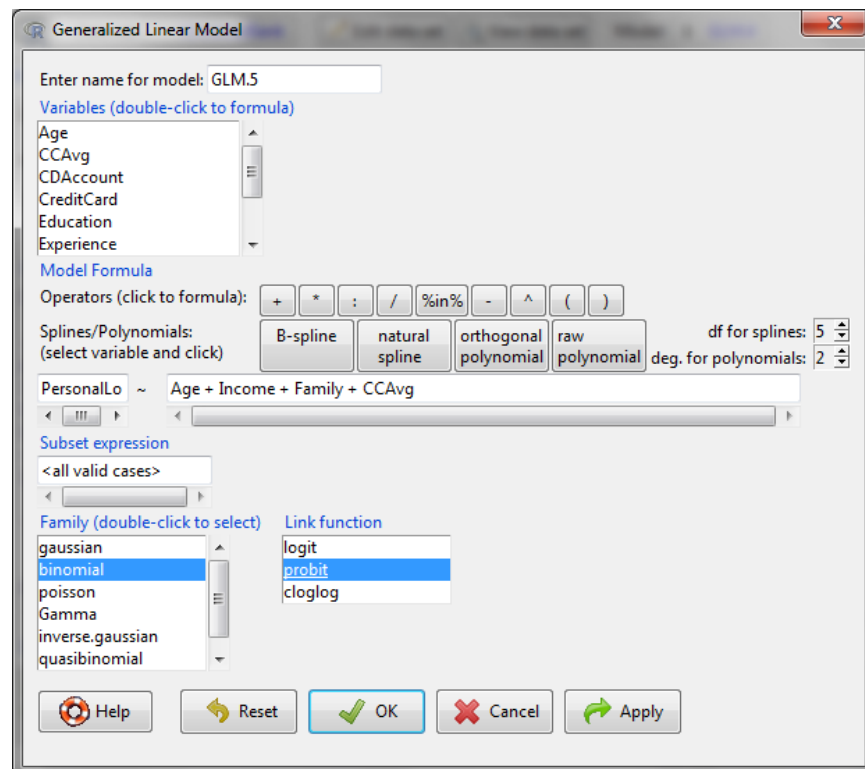
To perform a probit analysis on our data, where the Y variable is PersonalLoan and the explanatory variables are age, income, family size, and credit card average balance:

1. Click on Statistics, Fit models, Generalized linear model
2. Double click on PersonalLoan for the dependent variable
3. Double click on Age, Income, Family and CCAvg for the explanatory variables
4. Select binomial family
5. Select link function as probit
6. Click OK

Are the coefficients positive or negative?

Are the coefficients statistically significant?

Is there a difference between logit and probit?



The screenshot shows the R Commander interface. The 'Data set' is 'UniversalBank' and the 'Model' is 'GLM.5'. The R script in the editor is:

```
GLM.5 <- glm(PersonalLoan ~ Age + Income + Family + CCAvg,  
             family=binomial(probit), data=UniversalBank)  
summary(GLM.5)
```

The 'Output' pane displays the following results:

```
Coefficients:  
              Estimate Std. Error z value Pr(>|z|)  
(Intercept) -5.0473424  0.2174640 -23.210 < 2e-16 ***  
Age           0.0051159  0.0028504   1.795  0.07269 .  
Income        0.0229865  0.0009718  23.654 < 2e-16 ***  
Family        0.4022511  0.0335304  11.997 < 2e-16 ***  
CCAvg         0.0526373  0.0172616   3.049  0.00229 **  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for binomial family taken to be 1)  
  
Null deviance: 3162.0 on 4999 degrees of freedom  
Residual deviance: 1796.5 on 4995 degrees of freedom
```

The 'Messages' pane shows a warning:

```
logged("x"), log.y = logged("y"), :  
could not fit smooth
```

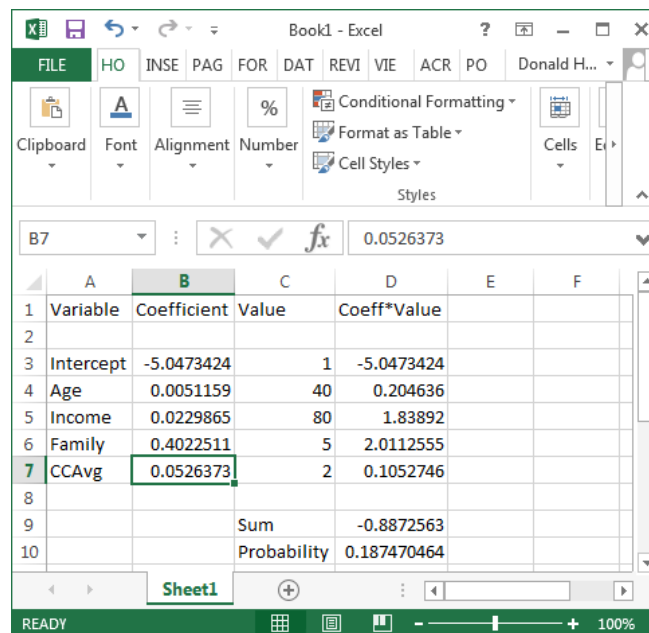
As an in class exercise, try adding Online as an explanatory variable. Is it significant in either logit or probit?

Session 9.7: Probit Predictions

You can use a spreadsheet to calculate probit probabilities for different combinations of the explanatory variable values. Let's calculate the probability of taking out a loan for someone who is 40 years old, with \$80,000 in income, family of 5, and credit card average balance of \$2,000. Use the data from the previous page.

1. Create the labels in row 1 for Variable, Coefficient, Value, and Coeff*Value
2. In Column A, below Variable, list Intercept and each of the explanatory variables
3. In Column B, below Coefficient, enter the coefficients from your Logit regression
4. In Column C, below Value, enter 1 for intercept and the values that you want to evaluate
 - a. Enter 40 for age
 - b. Since income is in the dataset in thousands, enter 80 for \$80,000
 - c. Enter 5 for family
 - d. Since credit card average balance is in the dataset in thousands, enter 2 for \$2,000
5. In Column D, below Coeff*Value, enter the formula to multiply the coefficient and value; e.g., for cell D3, enter =B3*C3
6. In cell D9, enter the formula for the sum of the column D calculations; i.e., =sum(D3:D7)
7. In cell D10, calculate the probability for the standard normal distribution using =NORM.S.DIST(D9,TRUE)

Vary the age, income, family size, and credit card balance to determine the effect on the probability of taking out a loan.



	A	B	C	D	E	F
1	Variable	Coefficient	Value	Coeff*Value		
2						
3	Intercept	-5.0473424	1	-5.0473424		
4	Age	0.0051159	40	0.204636		
5	Income	0.0229865	80	1.83892		
6	Family	0.4022511	5	2.0112555		
7	CCAvg	0.0526373	2	0.1052746		
8						
9			Sum	-0.8872563		
10			Probability	0.187470464		

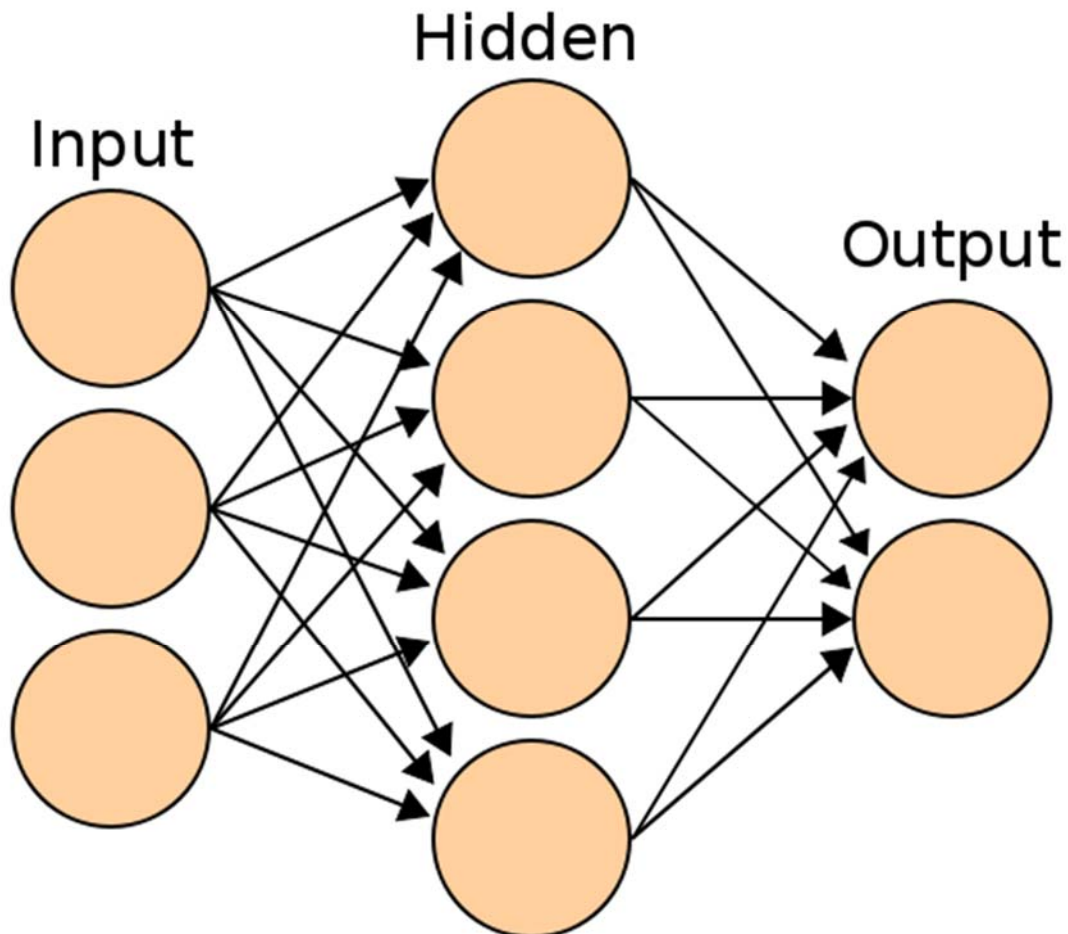
Session 9.8: Neural Networks

Reference:

Rumelhart, D.E; James McClelland (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press.

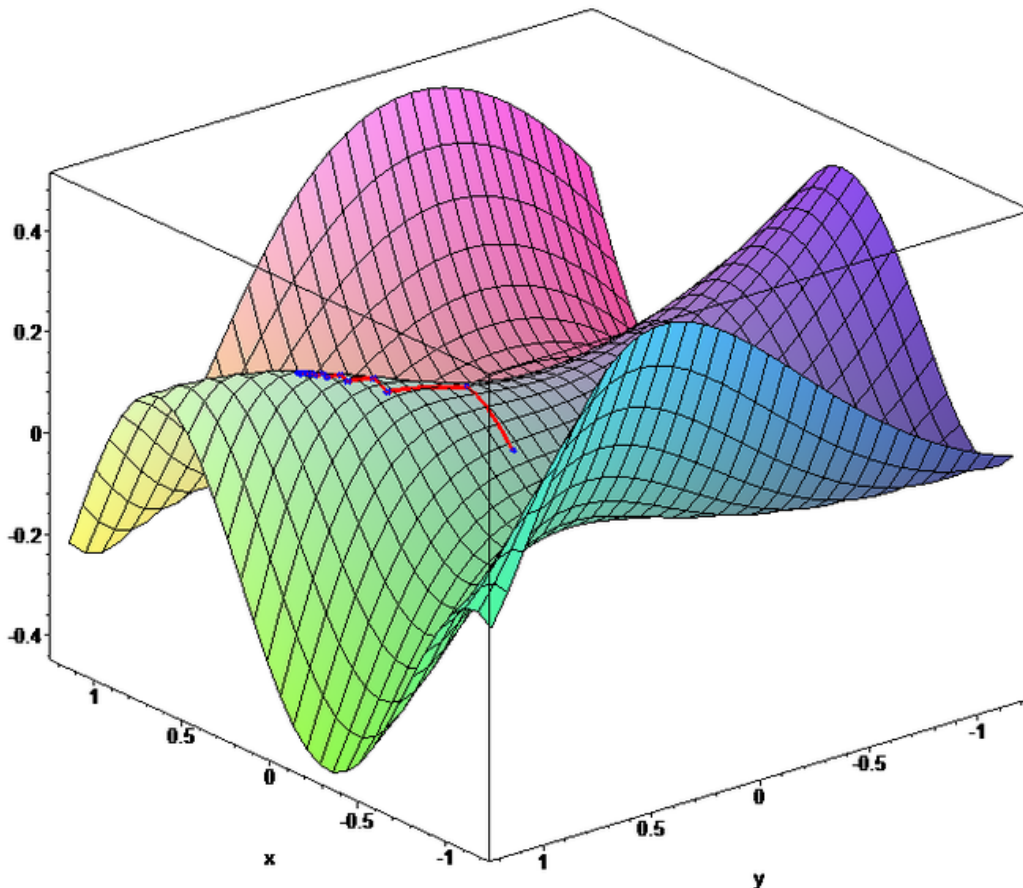
David Rumelhart and Jay McClelland recognized the limitation of a linear perceptron and proposed two innovations in 1986.

1. Use the logistic function to represent non-linear behavior
2. Add another layer (called the hidden layer) to produce more complex functions and represent more complex relationships



Why use the logistic function (used in Logit) rather than the normal distribution (used in Probit)? The logistic function has a very simple derivative. Why is this important?

Neural network searches use gradient search. Imagine that you are climbing a hill. To reach the peak in the shortest amount of time, look at where you are standing and find the direction with the steepest slope. Head in that direction, then decide in a new direction.



The risk is that there might be multiple high points, where some are local optima. Gradient search uses multiple starting points to find the global optimum.

So, why is a simply derivative for the logistic function important? The derivative gives you the slope so you can determine search direction. Other functions could be used, but the logistic function is the most popular because the derivative is easy to calculate.

If $f(x)$ is the logistic function, then the derivative is $f(x) * (1 - f(x))$.

Neural Networks with R

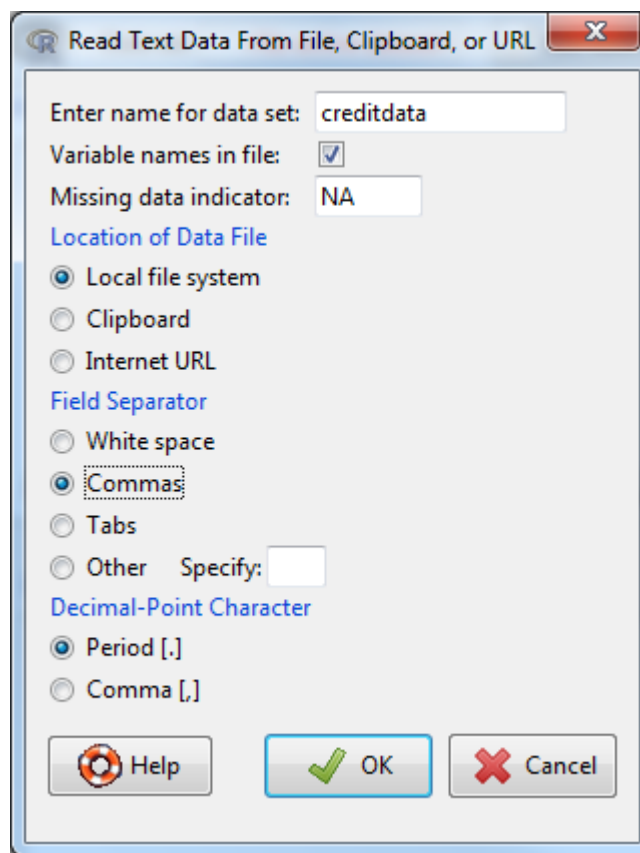
Download Datasets

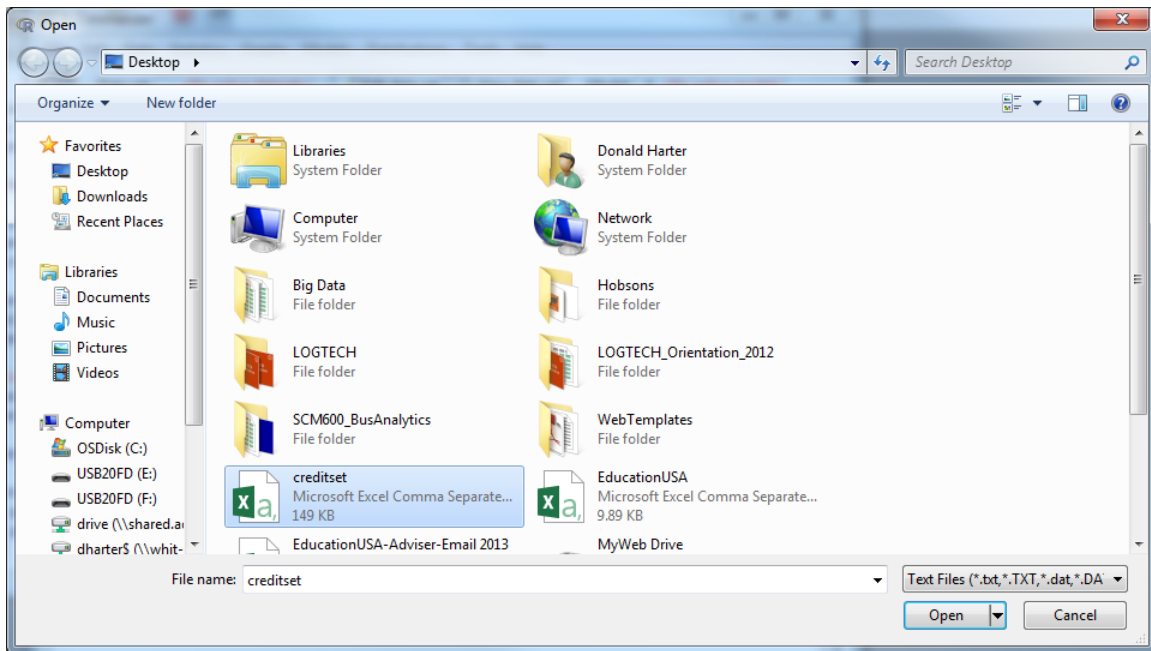
The Business Analytics - Week 9 creditset.csv file will be used for this exercise.

Loading Data

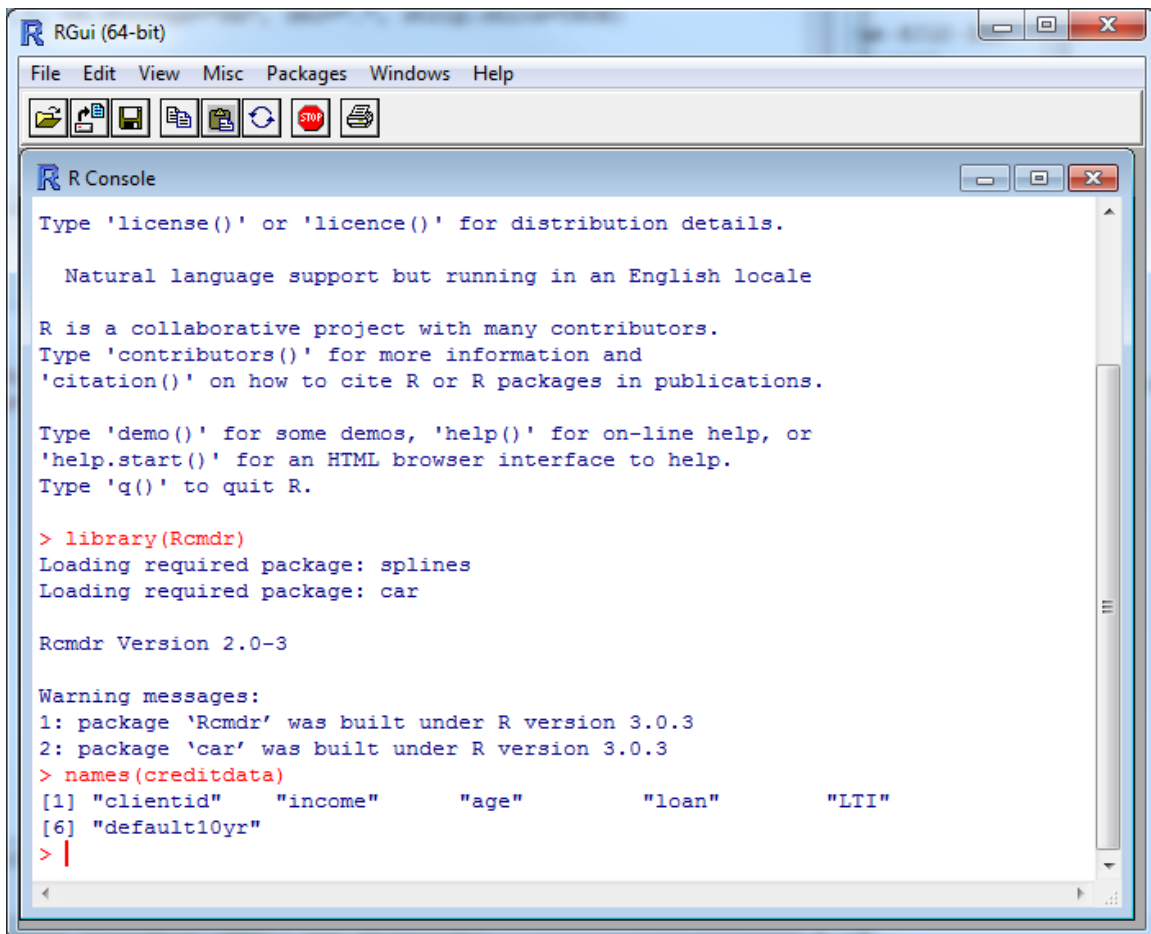
To load data into R:

1. Click on Data at the top of the Rcmdr screen
2. Click on Import Data > From text file ...
3. Enter the name that you would like to use for this data set; type in creditdata
4. Change Field Separator to Commas, then OK
5. Click on the creditset.csv file, then Open

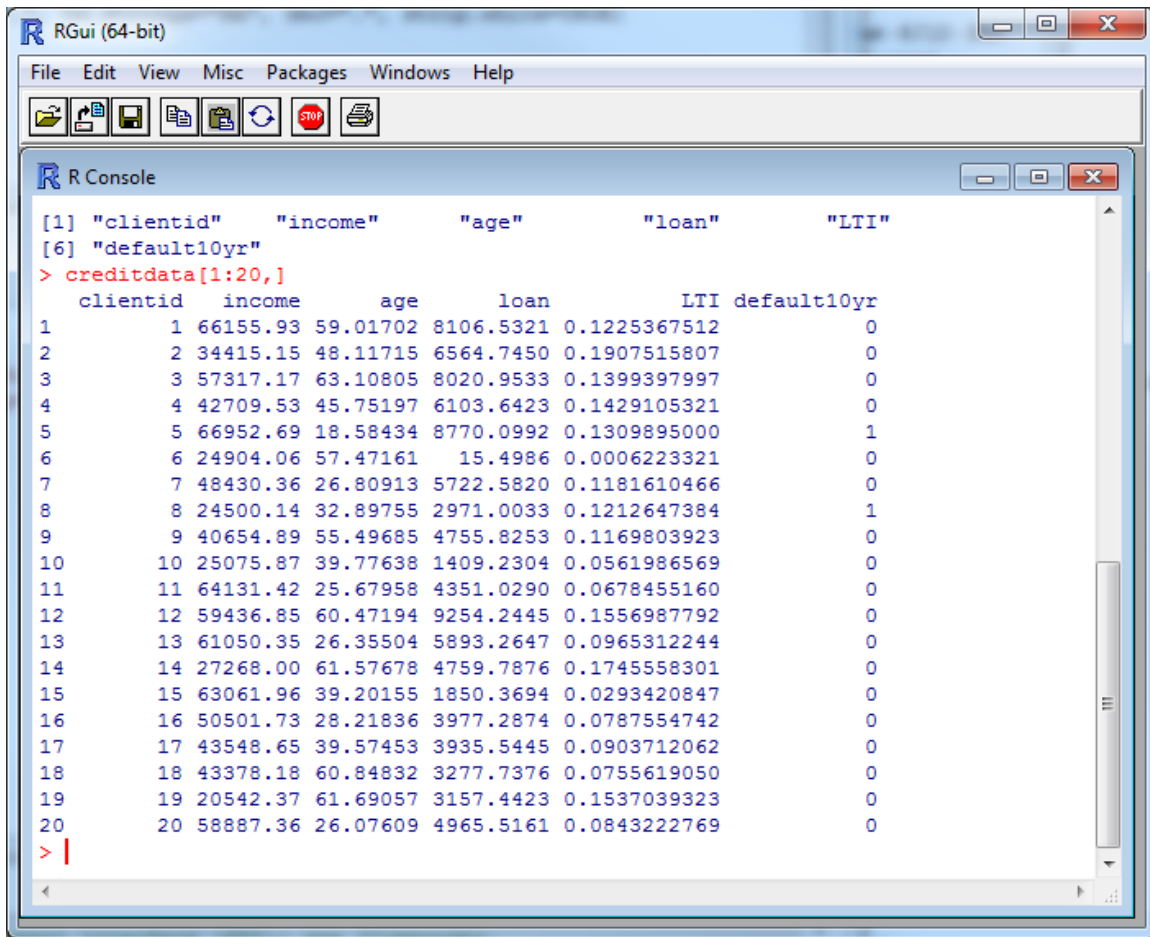




In the R Console (RGui), type `names(creditdata)`.



In the R Console (RGui), type `creditdata[1:20,]`



```
RGui (64-bit)
File Edit View Misc Packages Windows Help

R Console
[1] "clientid"      "income"        "age"           "loan"          "LTI"
[6] "default10yr"
> creditdata[1:20,]
  clientid  income    age    loan    LTI default10yr
1      1 66155.93 59.01702 8106.5321 0.1225367512      0
2      2 34415.15 48.11715 6564.7450 0.1907515807      0
3      3 57317.17 63.10805 8020.9533 0.1399397997      0
4      4 42709.53 45.75197 6103.6423 0.1429105321      0
5      5 66952.69 18.58434 8770.0992 0.1309895000      1
6      6 24904.06 57.47161  15.4986 0.0006223321      0
7      7 48430.36 26.80913 5722.5820 0.1181610466      0
8      8 24500.14 32.89755 2971.0033 0.1212647384      1
9      9 40654.89 55.49685 4755.8253 0.1169803923      0
10     10 25075.87 39.77638 1409.2304 0.0561986569      0
11     11 64131.42 25.67958 4351.0290 0.0678455160      0
12     12 59436.85 60.47194 9254.2445 0.1556987792      0
13     13 61050.35 26.35504 5893.2647 0.0965312244      0
14     14 27268.00 61.57678 4759.7876 0.1745558301      0
15     15 63061.96 39.20155 1850.3694 0.0293420847      0
16     16 50501.73 28.21836 3977.2874 0.0787554742      0
17     17 43548.65 39.57453 3935.5445 0.0903712062      0
18     18 43378.18 60.84832 3277.7376 0.0755619050      0
19     19 20542.37 61.69057 3157.4423 0.1537039323      0
20     20 58887.36 26.07609 4965.5161 0.0843222769      0
> |
```

The data includes:

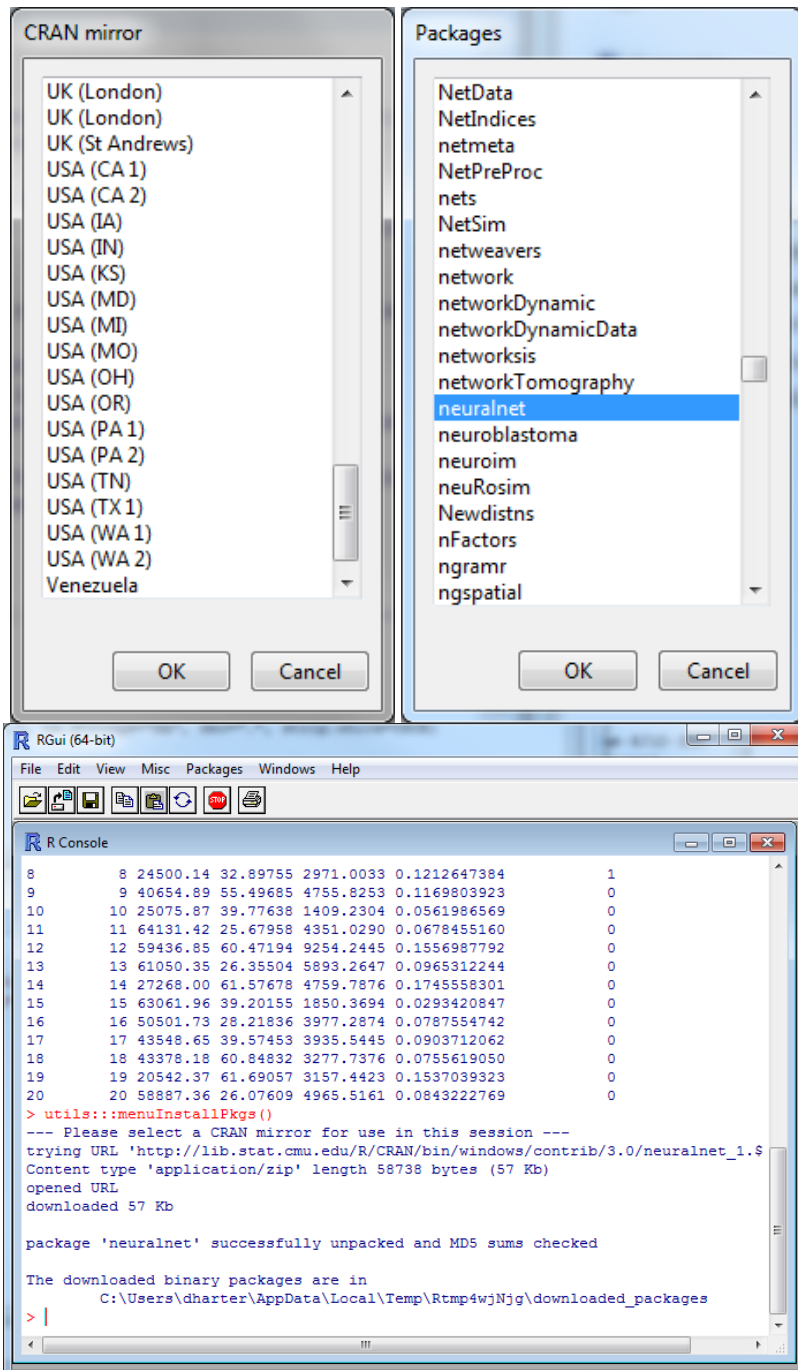
- Client id unique identifier for each loan client
- Income annual income in Euros
- Age age of customer
- Loan loan size in Euros
- LTI loan to yearly income ratio
- Default10yr 1 if a default occurred in 10 years; 0 if no default occurred in 10 years

Which variables should affect the probability of a loan default?

Installing NeuralNet

Follow these to install neuralnet.

1. In the R Console (RGui), at the top of the screen, click on Packages
2. In the drop down menu, click on Install Package(s)
3. In the CRAN mirror, select the location closest to you; use USA (PA 1), then click OK
4. In the Packages screen, click on neuralnet, then OK
5. If prompted to create a personal library, click Yes



Launch neuralnet

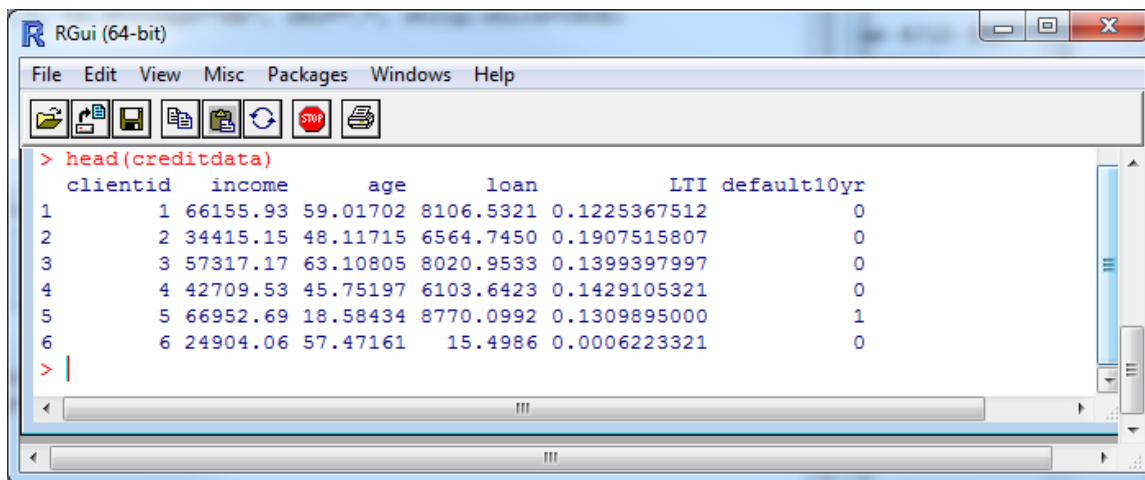
neuralnet is the R software that performs neural network calculations:

1. Type `library(neuralnet)`
2. If you receive a warning message that some packages are missing, it will ask if you want them installed. Click Yes.
3. On the Install Missing Packages screen, click OK
4. R will install the necessary software

Viewing a sample of data

Another way to view the data headers and sample data is with the `head` command.

1. In the R console (RGui), type `head(creditdata)`



Neural network training and testing data

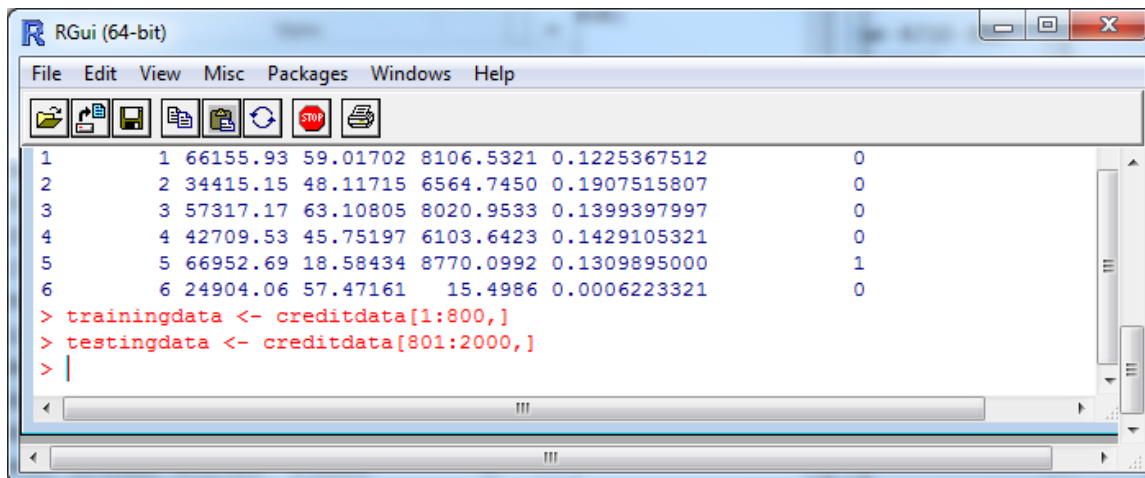
Whenever you build analytics models, it's a good idea to use some data to build the model and other data to test the model. R allows you to split the dataset into training and testing subsets.

There are 2000 observations in our `creditdata` data set. Since the data is randomly distributed (not sorted), we will select the first 800 data rows for the training data, and the remainder of the data for testing purposes.

To create the training data and testing data:

1. Type the following to create the training data
`trainingdata <- creditdata[1:800,]`
2. Type the following to create the testing data
`testingdata <- creditdata[801:2000,]`

The training data that the neural network will use to learn is stored in `trainingdata`. The testing data that we will use to test the neural network is stored in `testingdata`.

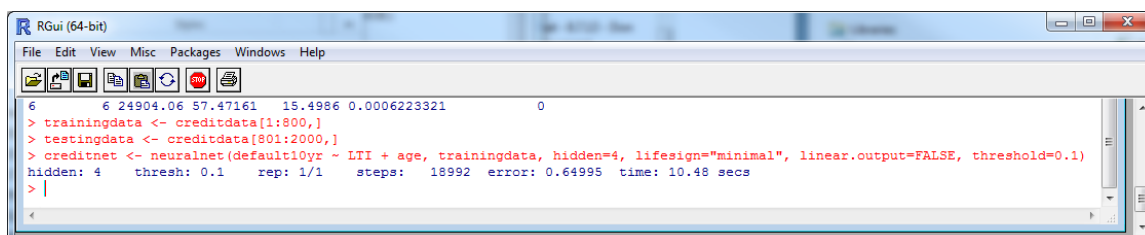


Neural network analysis

To run the neural network on loan defaults with inputs of loan to income ratio (LTI) and age, enter the command into the R console:

```
creditnet <- neuralnet(default10yr ~ LTI + age, trainingdata, hidden=4, lifesign="minimal",
linear.output=FALSE, threshold=0.1)
```

creditnet	stores the results
neuralnet	program which runs the neural network analysis
default10yr	dependent variable
LTI & age	independent variables
trainingdata	data to be used for training the network
hidden	number of hidden nodes
lifesign	amount of output
linear.output	whether you want linear or non-linear model
threshold	error term threshold

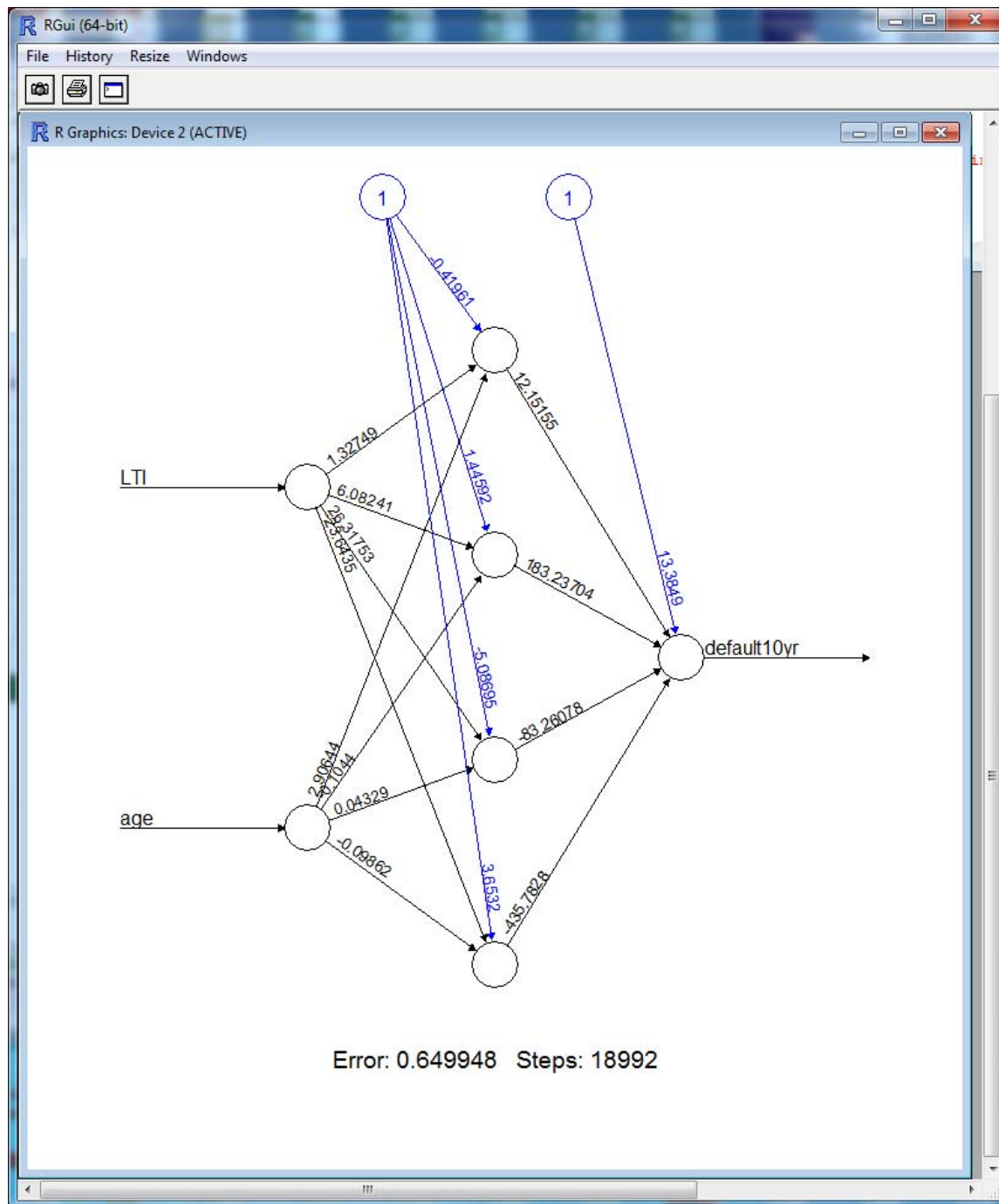


The neural network algorithm will perform a gradient search to find a solution that minimizes the error of making a mistake. In this case, the algorithm took 18,992 iterations or steps.

Neural Network Model

The result of the model can be displayed by plotting the model

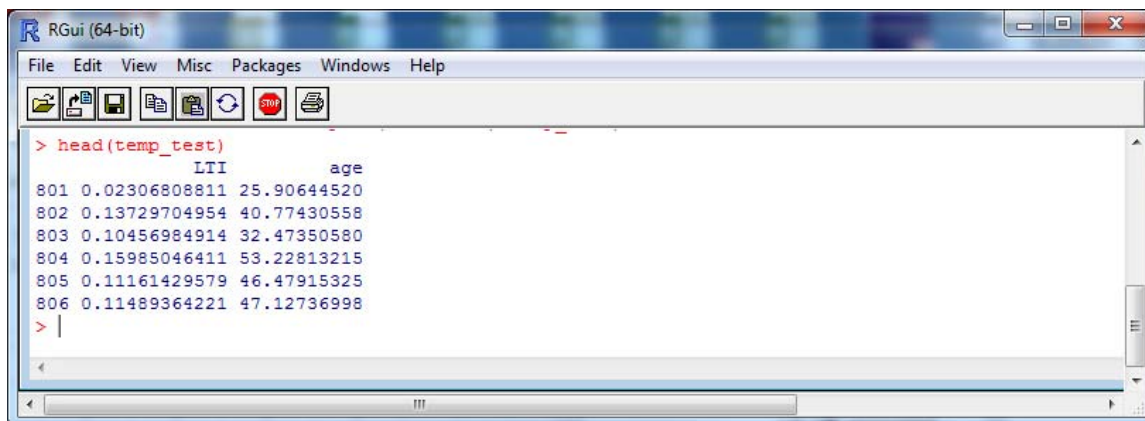
1. In the R console (RGui), type the command `plot(creditnet,rep="best")`



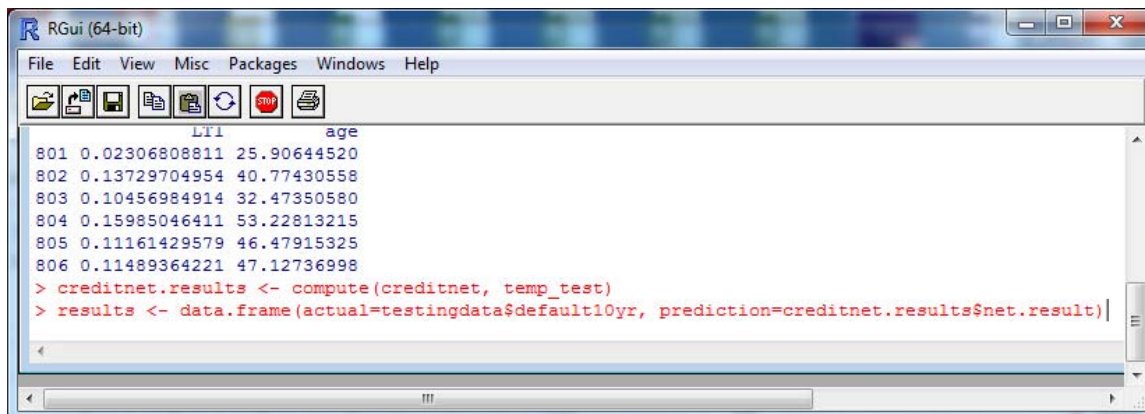
Testing the Neural Network Prediction Ability

To test our neural network with the testing data, we first need to reduce the test data to only the variables needed for the model.

1. In the R Console (RGui), type
`temp_test <- subset(testingdata, select = c("LTI", "age"))`
2. To view this subset of data, type
`head(temp_test)`
3. To predict loan defaults for the test data, type
`creditnet.results <- compute(creditnet, temp_test)`
4. To create a view of the predictions, type the two lines below onto one line
`results <- data.frame(actual=testingdata$default10yr, prediction=creditnet.results$net.result)`
5. Finally, to view the results, type
`results[1:20,]`
6. There are too many decimal places. To round off the number, type
`results$prediction <- round(results$prediction)`
`results [1:20,]`



```
> head(temp_test)
      LTI      age
801 0.02306808811 25.90644520
802 0.13729704954 40.77430558
803 0.10456984914 32.47350580
804 0.15985046411 53.22813215
805 0.11161429579 46.47915325
806 0.11489364221 47.12736998
> |
```



```
> head(temp_test)
      LTI      age
801 0.02306808811 25.90644520
802 0.13729704954 40.77430558
803 0.10456984914 32.47350580
804 0.15985046411 53.22813215
805 0.11161429579 46.47915325
806 0.11489364221 47.12736998
> |
> creditnet.results <- compute(creditnet, temp_test)
> results <- data.frame(actual=testingdata$default10yr, prediction=creditnet.results$net.result)
```

