

# R: Advanced

## Datafiles

For these exercises, download the files:

- "Business Analytics – Week 9 Instructions.pdf"
- "Business Analytics – Week 9 Universal Bank.xls"
- "Business Analytics – Week 9 creditset.xls"
- "Business Analytics – Week 9 Titanic.xls"

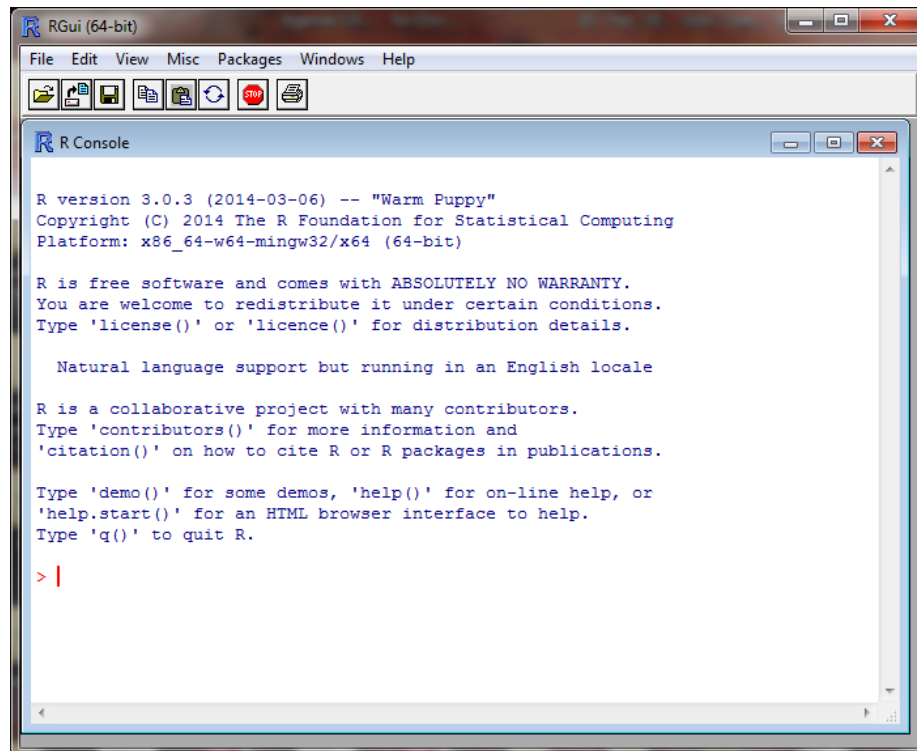
## Installation of R

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

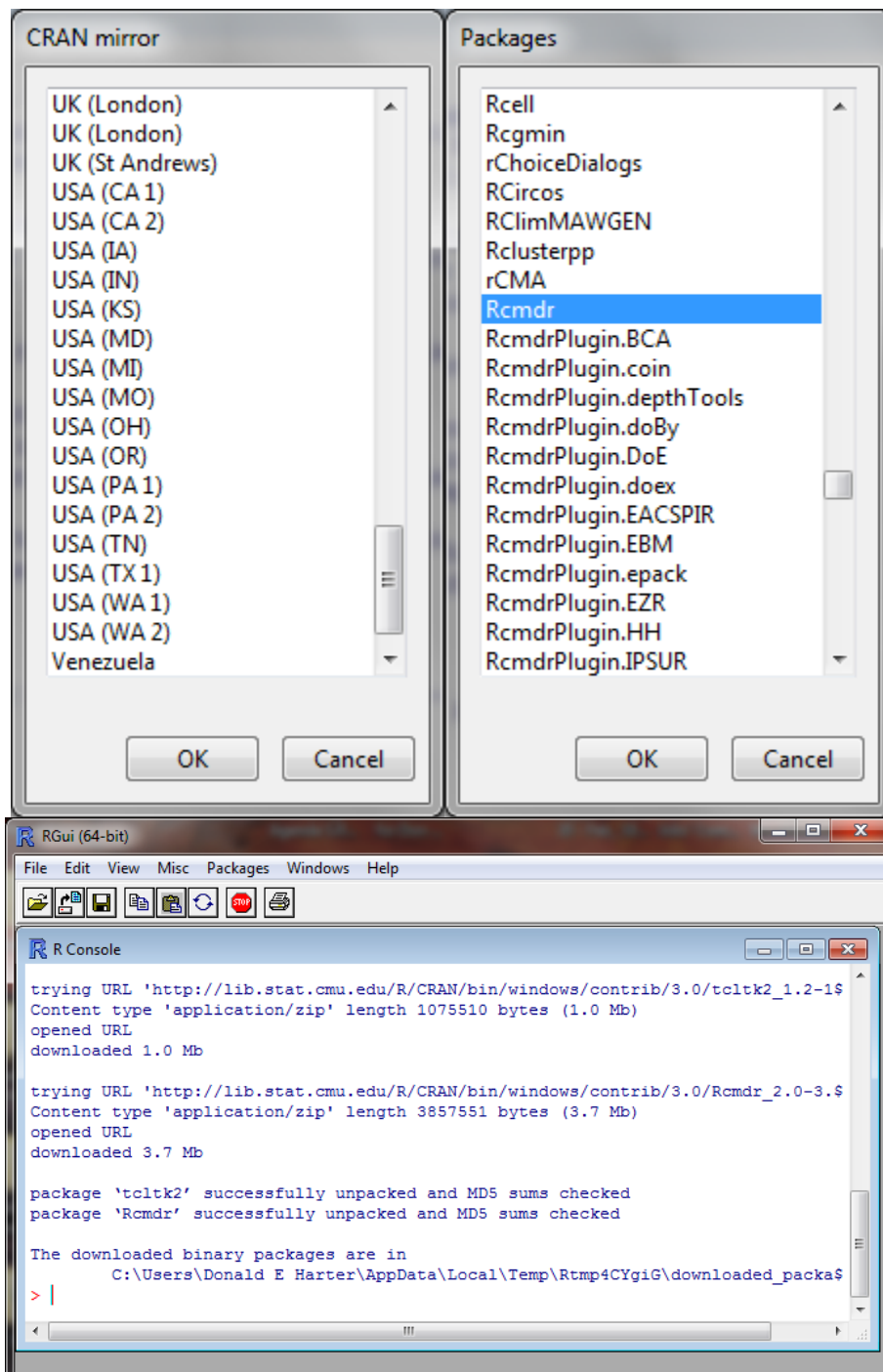


This is the command line screen. You can enter commands, but need to know the syntax. There is a simpler approach to running R, called Rcmdr (R Commander). If you are running a Whitman computer, Rcmdr is already installed. If not, you need to install it.

## Installing R Commander

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

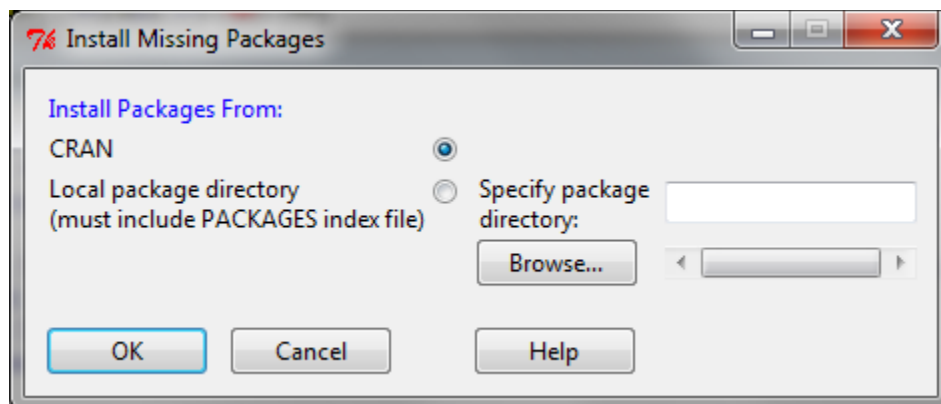
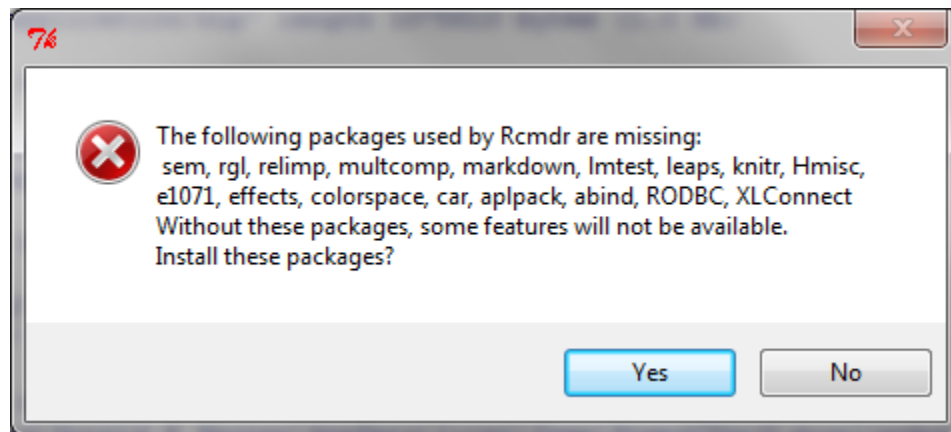
1. At the top of the screen, click on Packages
2. In the drop down menu, click on Install Package(s)
3. In the CRAN mirror, select the location closest to you; use USA (PA 1), then click OK
4. In the Packages screen, click on Rcmdr, then OK
5. When prompted to create a personal library, click Yes

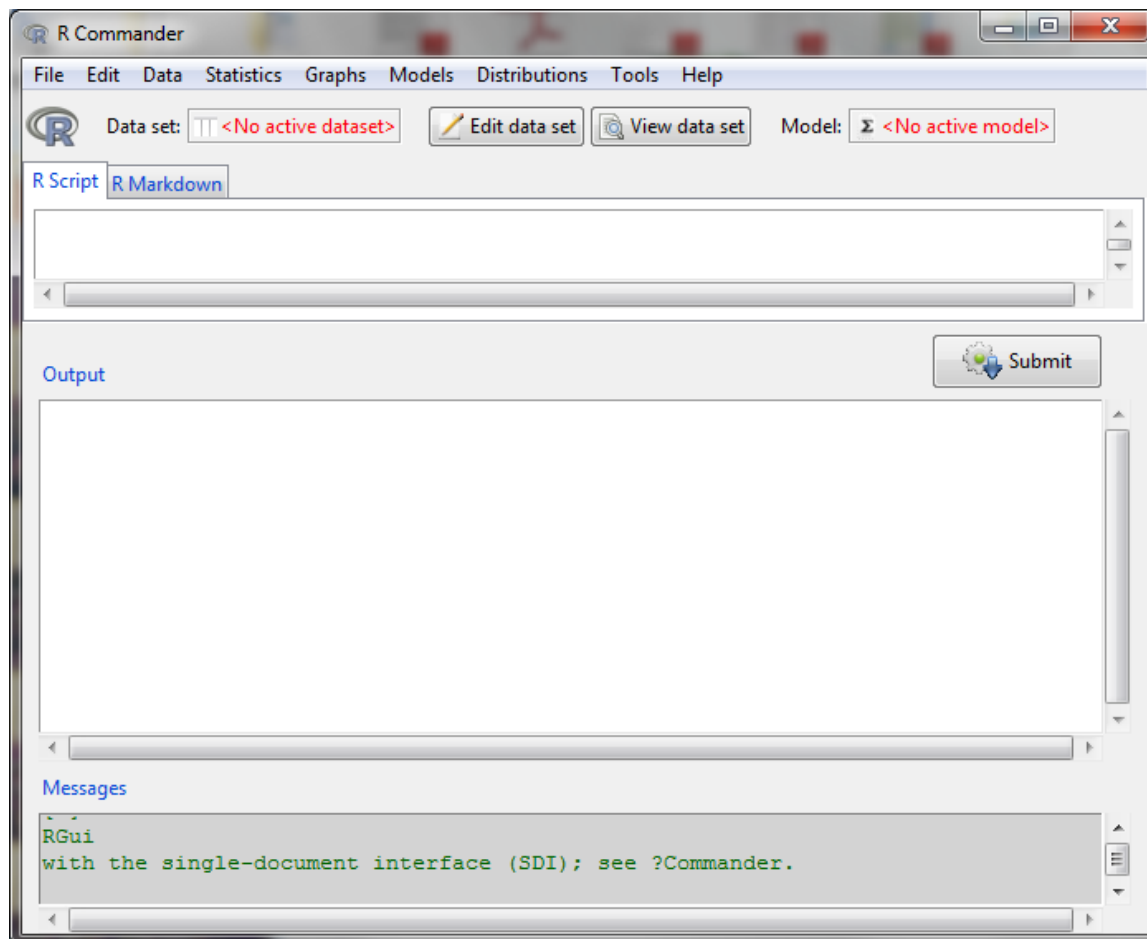


## Launch Rcmdr (R Commander)

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:

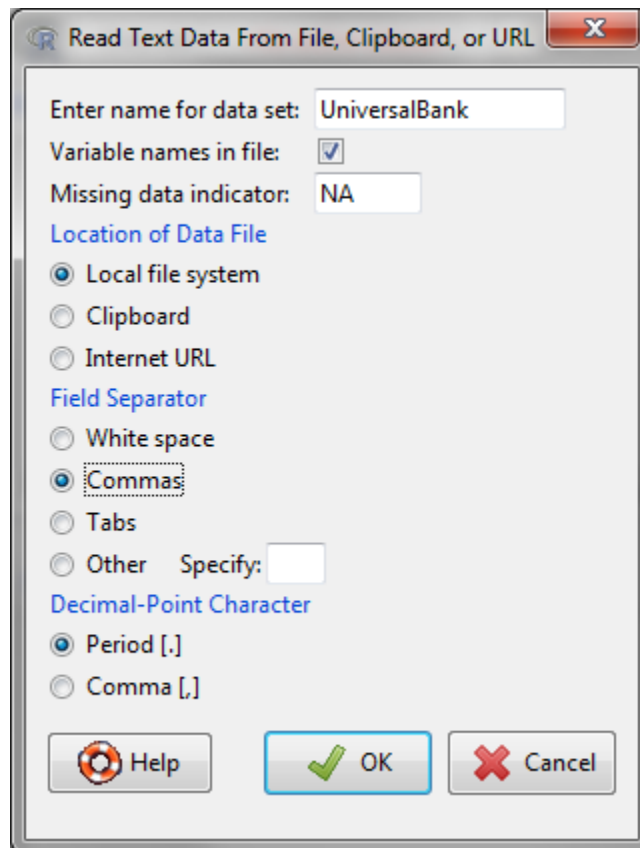
1. Go to the website:  
<http://www.biz.uiowa.edu/faculty/jledolter/DataMining>
2. Click on Data Text
3. Right click on UniversalBank.csv, then save on your computer
4. Remember where you saved the file

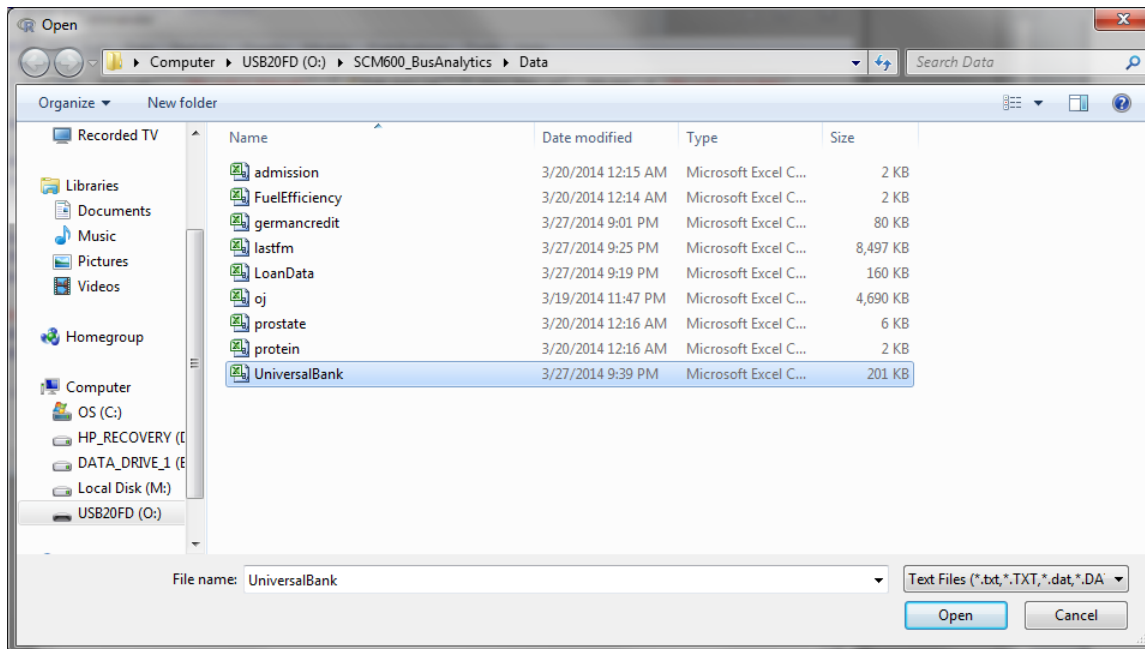
The Business Analytics - Week 9 Universal Bank.csv file can be downloaded from the course website.

### 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)

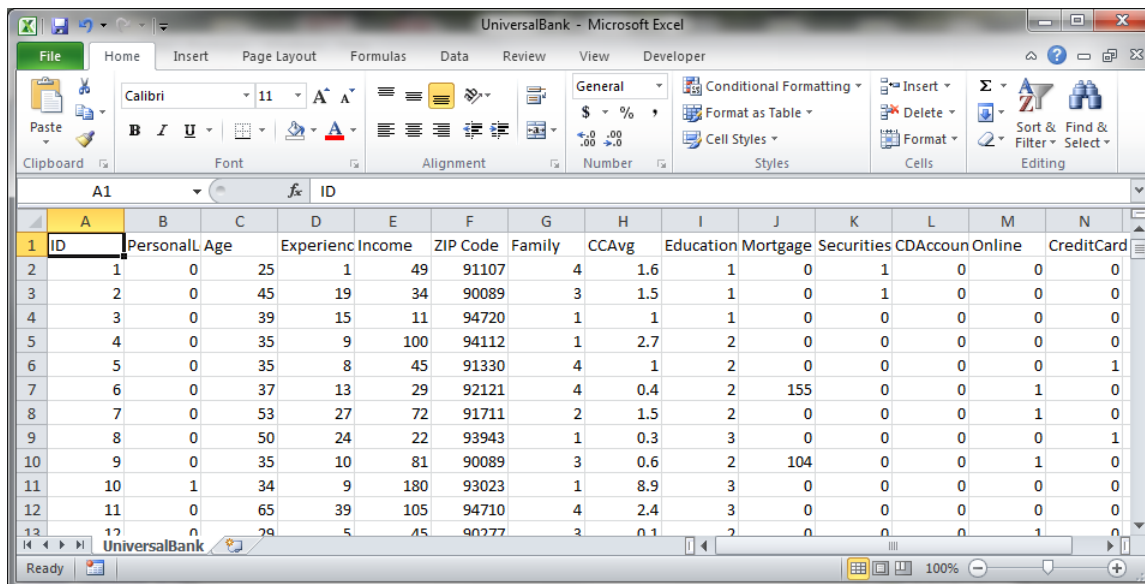
Mortgage: size of mortgage

SecuritiesAccount: No/Yes (0,1)

CDAccount: No/Yes (0,1)

Online: No/Yes (0,1)

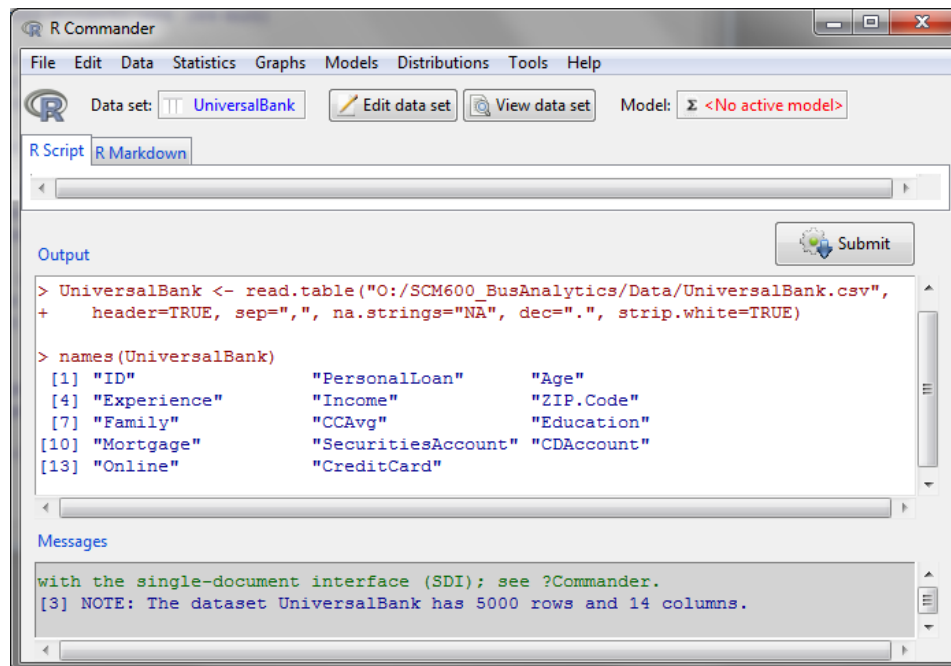
CreditCard: No/Yes (0,1)



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

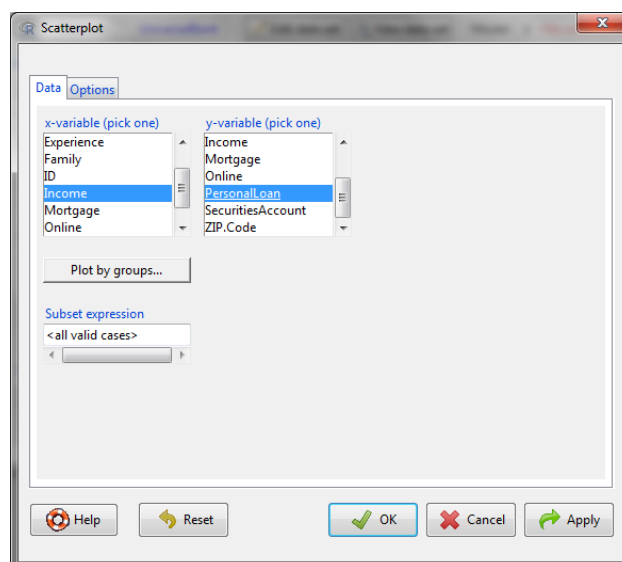


Notice that R generates the command `names(UniversalBank)`. This is the command line version.

## Scatterplots

To generate a scatter plot,

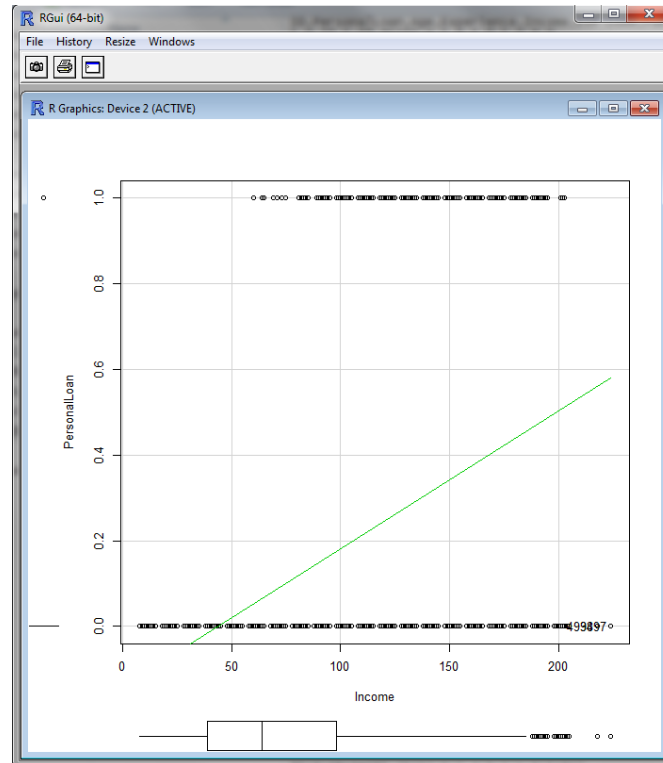
1. Click on Graphs, Scatterplot
2. Select Income as the x-variable
3. Select PersonalLoan as the y-variable
4. Click on OK



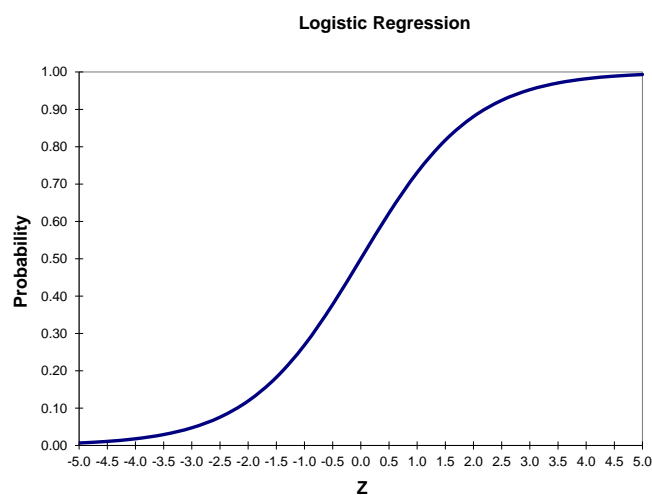


To interpret the chart:

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:



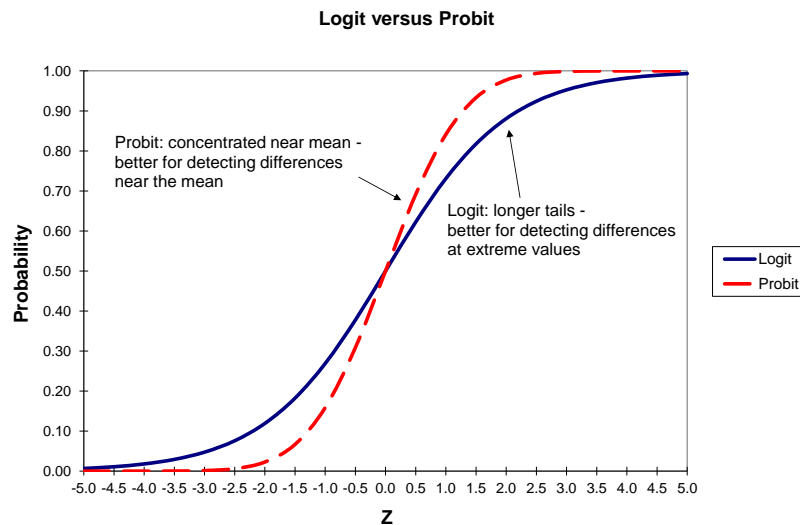
Logit and probit are techniques that assume the dependent variable (Y) is zero or one, and finds the relationship between the explanatory variables (X) and the dependent variable (Y). Logistic regression and logit are based on the logistic distribution. Probit is based on the normal distribution. Logit 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:

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



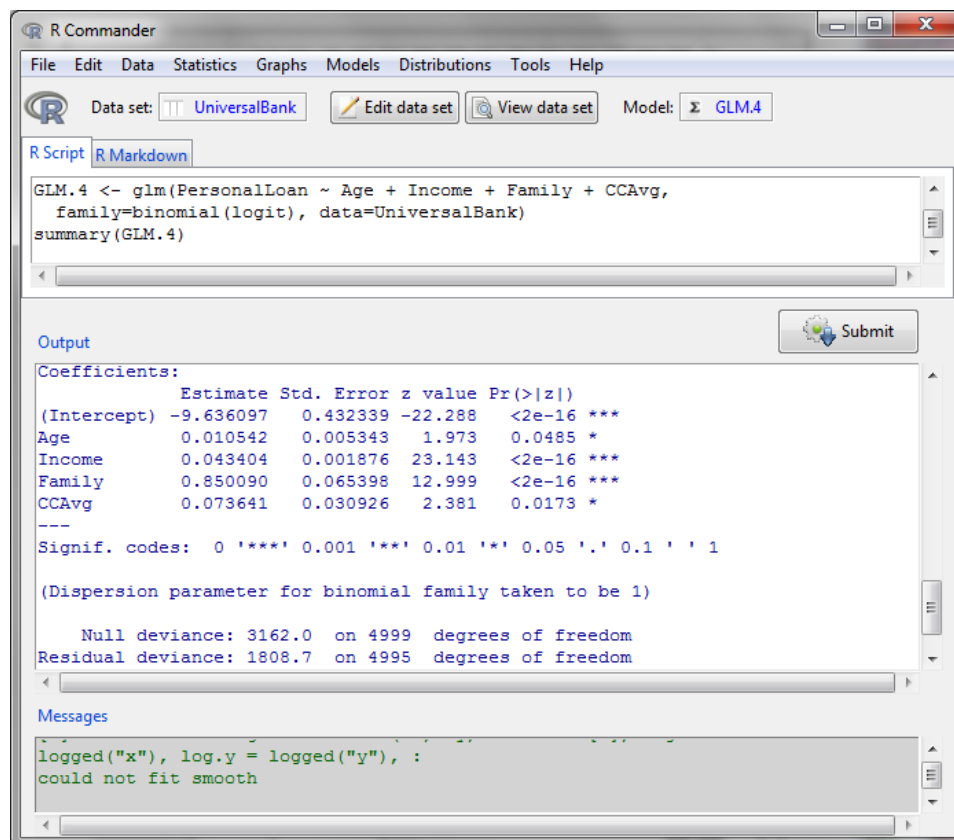
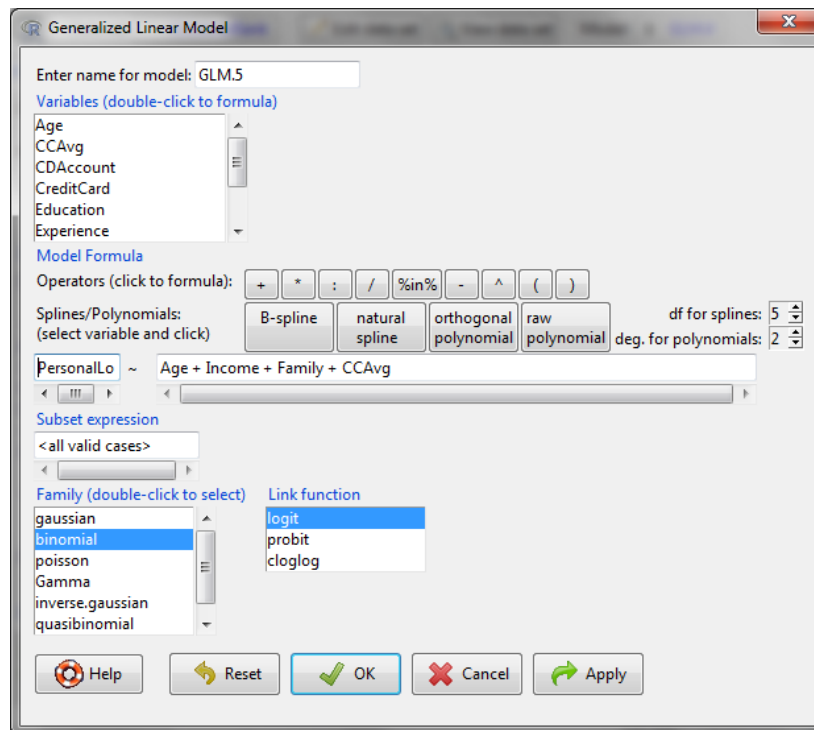
## Logit Analysis

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.,  $=\text{sum}(D3:D7)$
7. In cell D10, calculate the exponential of the sum; i.e.,  $=\text{exp}(D9)$
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

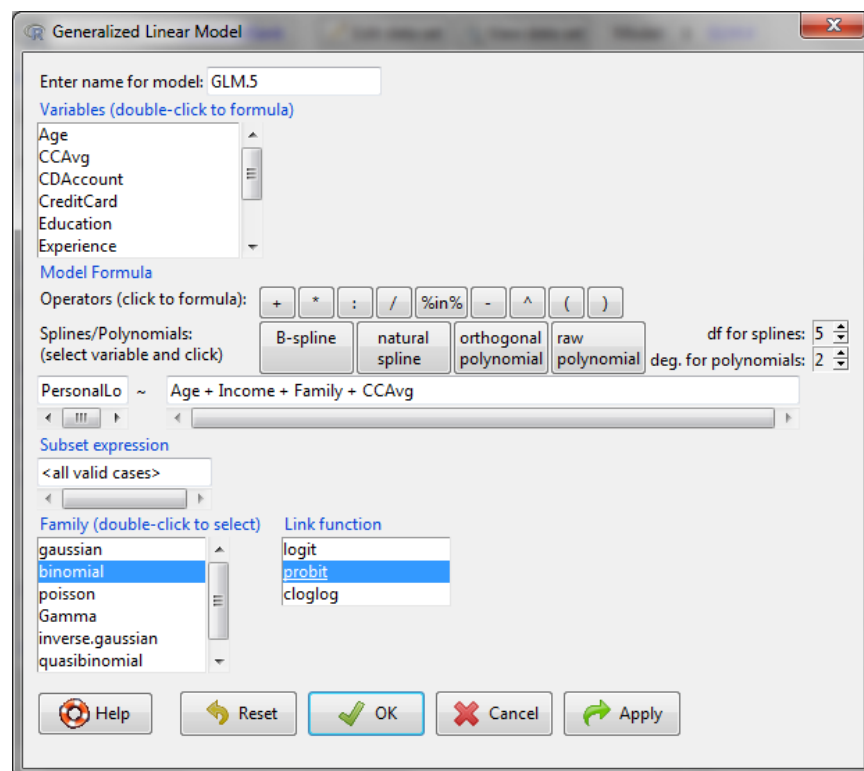
To perform a probit analysis on our data, where the Y variable is PersonalLoan and the explanatory variables are age, income, family size, and credit card average balance:

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?



R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: UniversalBank Edit data set View data set Model: GLM.5

R Script R Markdown

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

Output

Submit

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

Messages

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

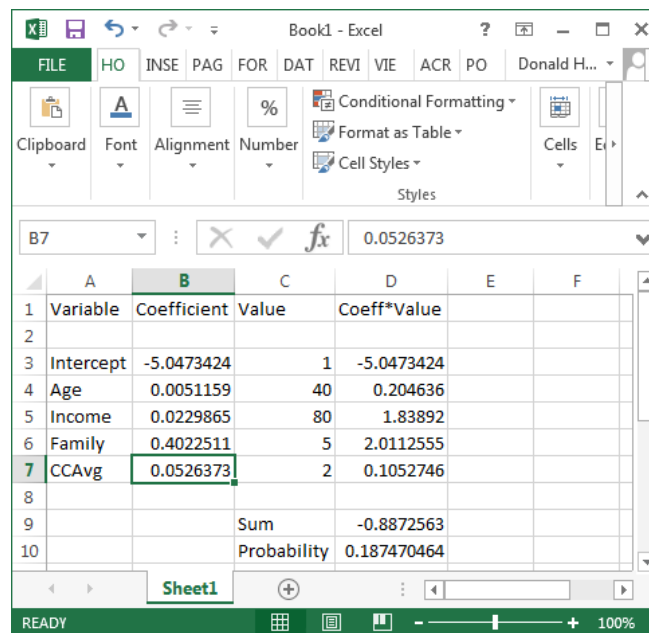
As an in class exercise, try adding Online as an explanatory variable. Is it significant in either logit or probit?

## 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		

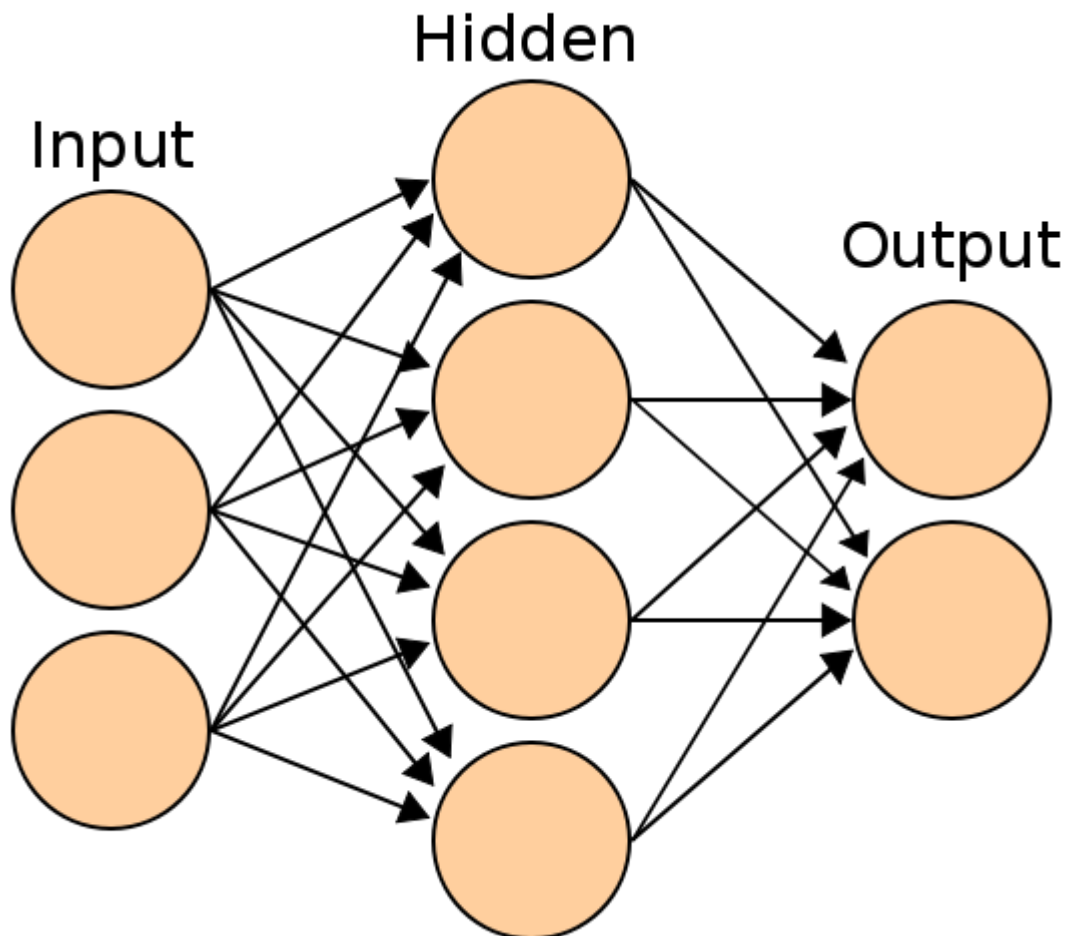
## Session 9.8: Neural Networks

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.

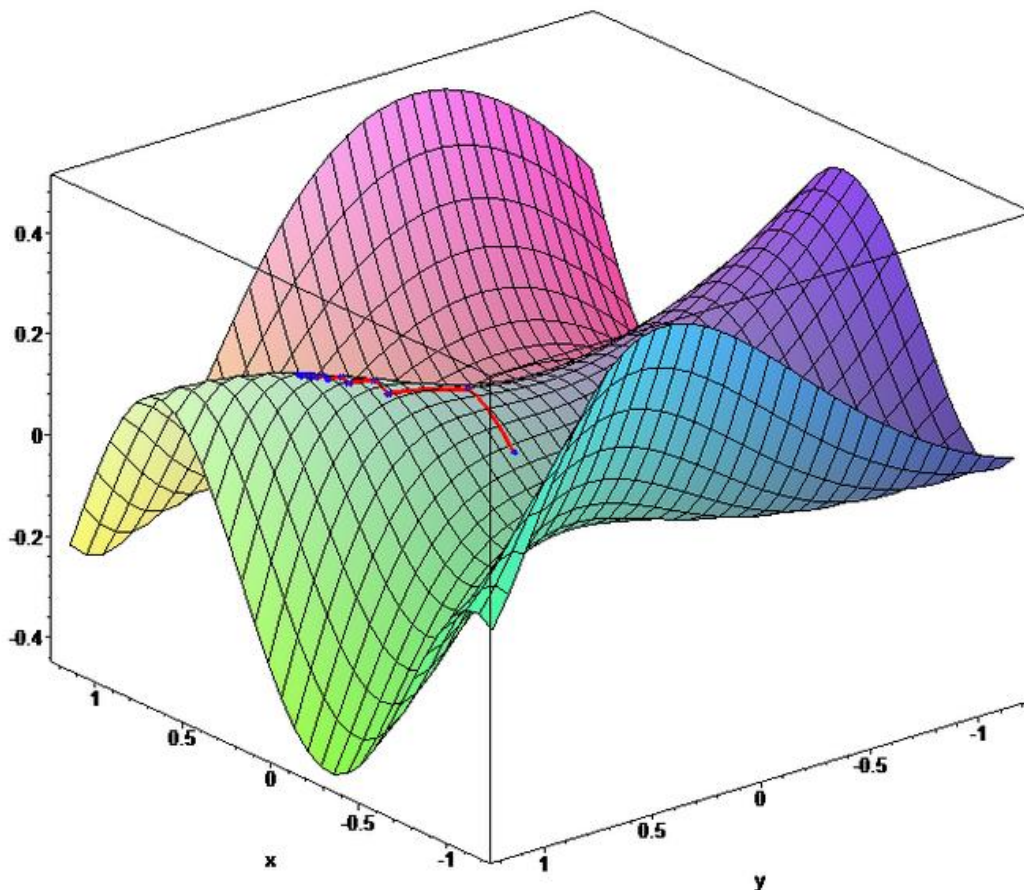
1. Use the logistic function to represent non-linear behavior
2. Add another layer (called the hidden layer) to produce more complex functions and represent more complex relationships





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

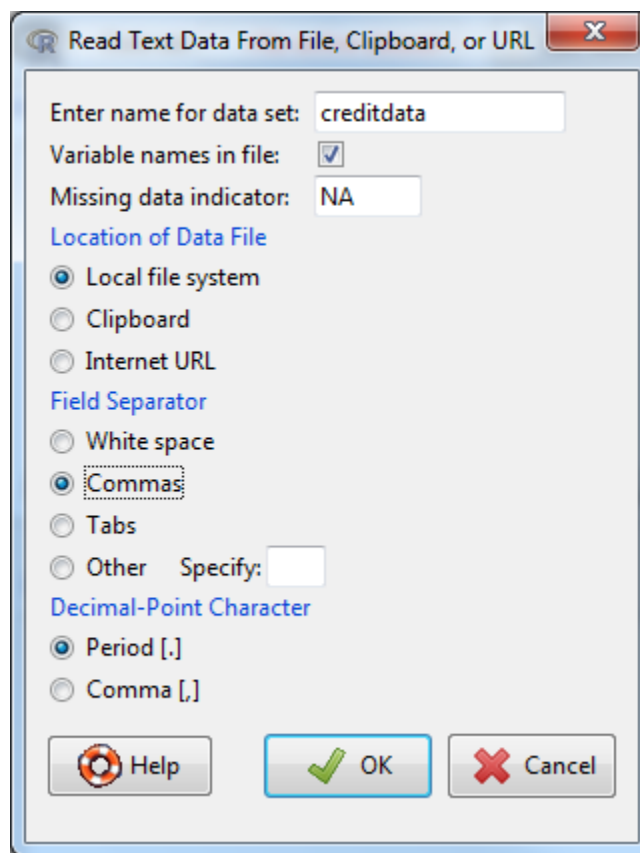
### Download Datasets

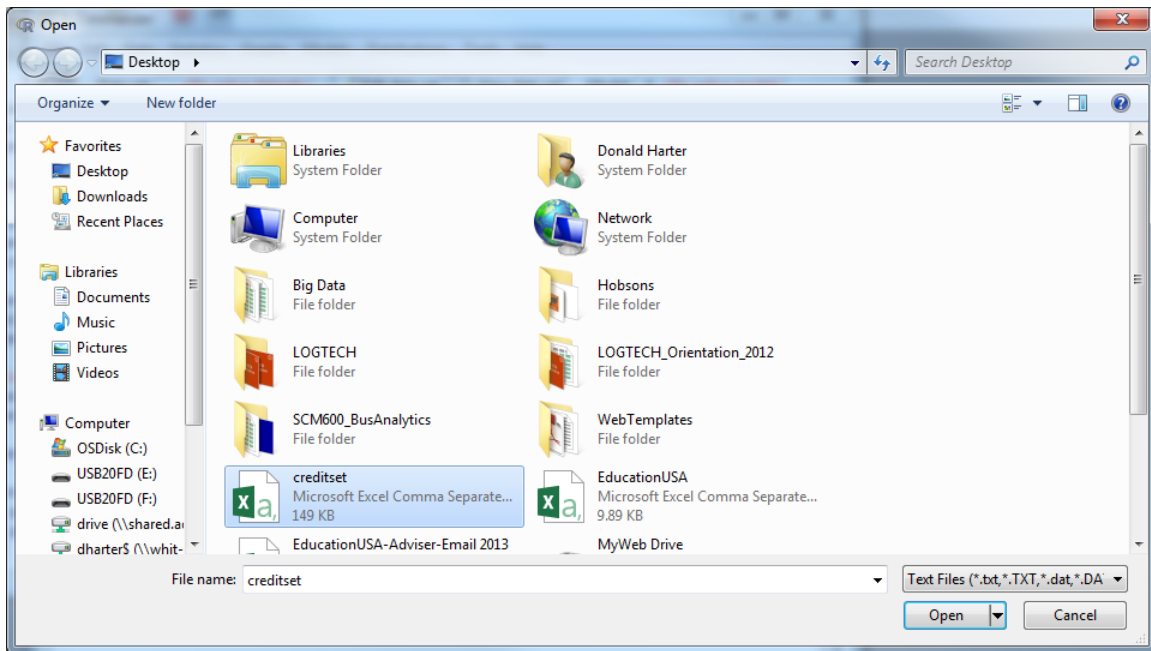
The Business Analytics - Week 9 creditset.csv file will be used for this exercise.

### Loading Data

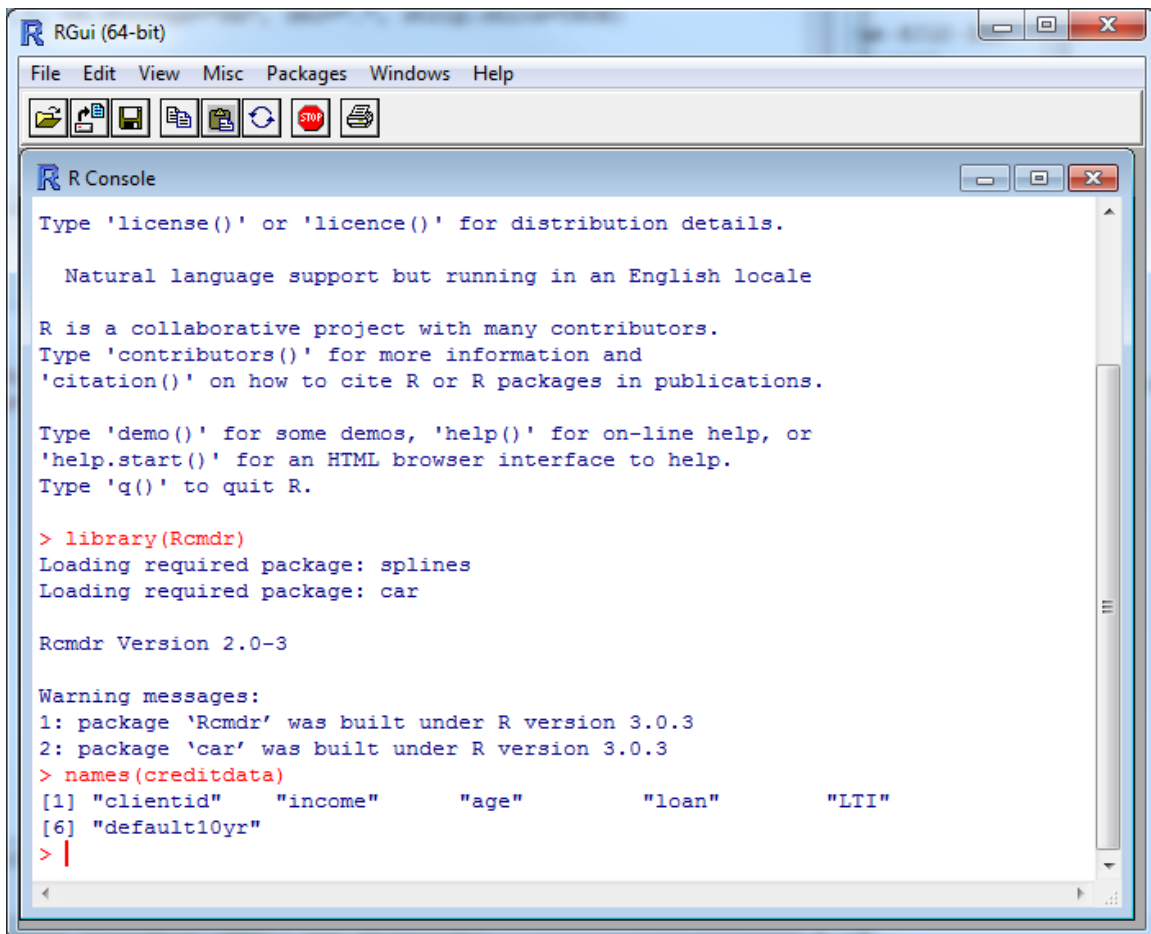
To load data into R:

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

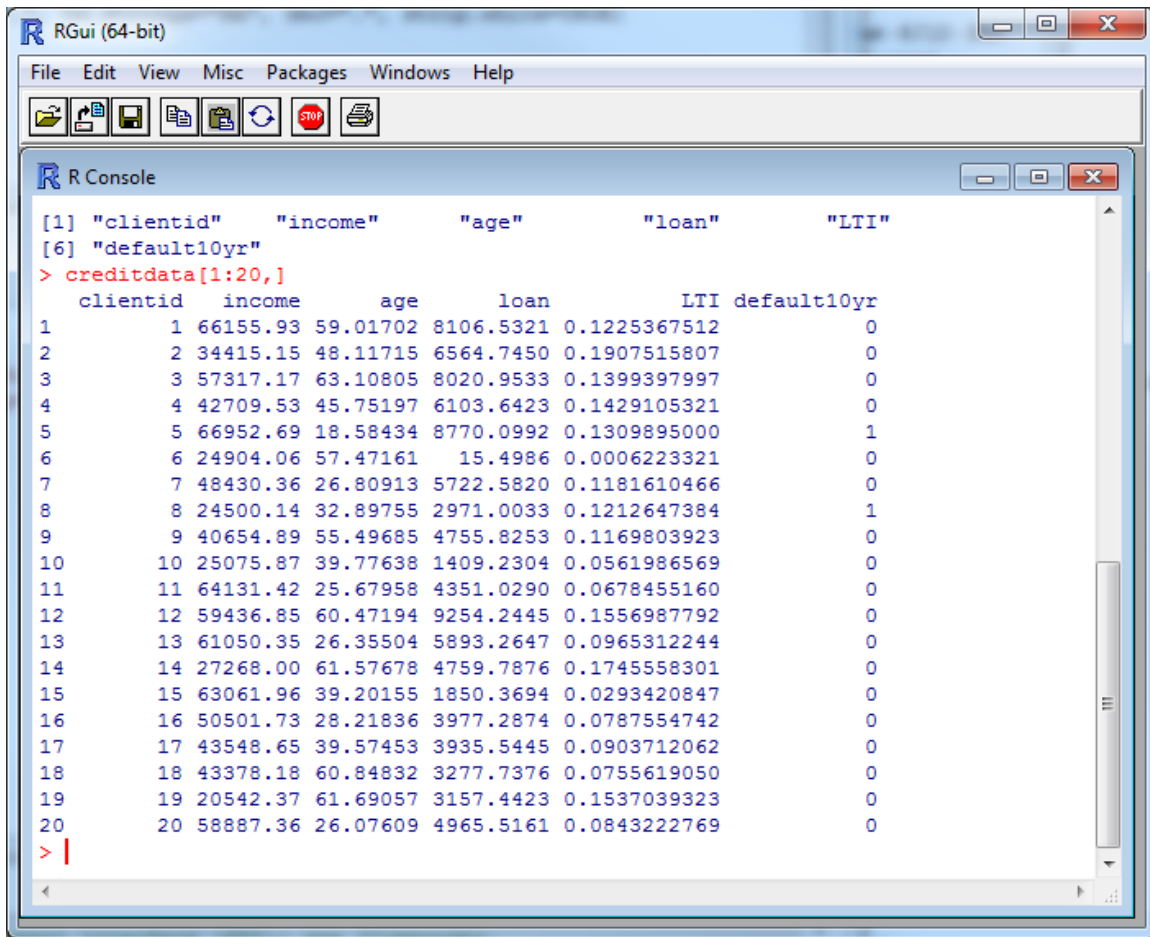




In the R Console (RGui), type `names(creditdata)`.



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:

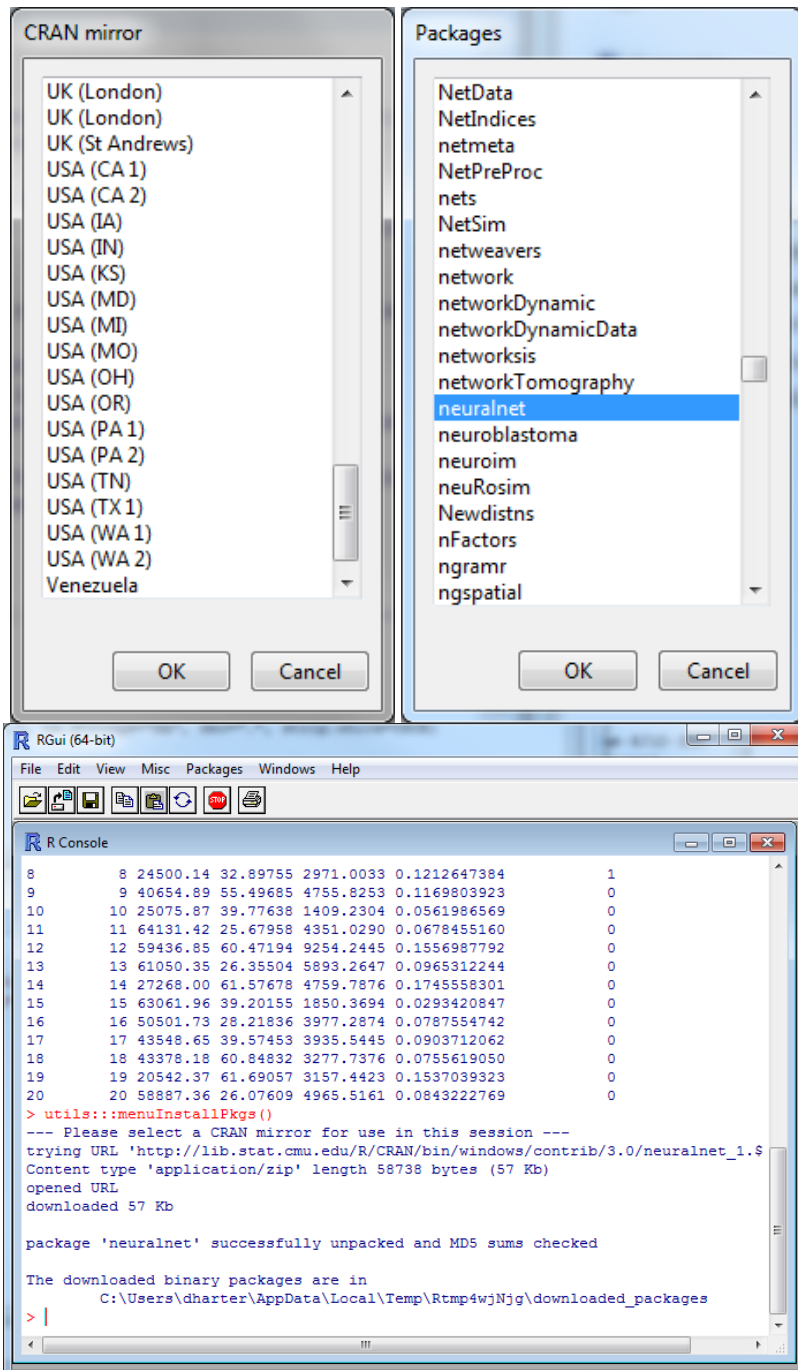
Client id	unique identifier for each loan client
Income	annual income in Euros
Age	age of customer
Loan	loan size in Euros
LTI	loan to yearly income ratio
Default10yr	1 if a default occurred in 10 years; 0 if no default occurred in 10 years

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

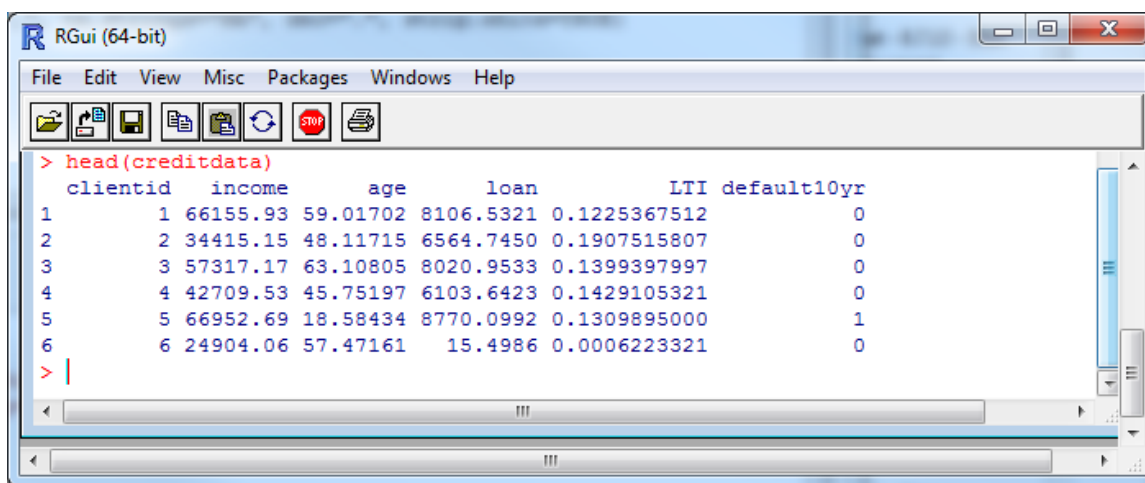
neuralnet is the R software that performs neural network calculations:

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

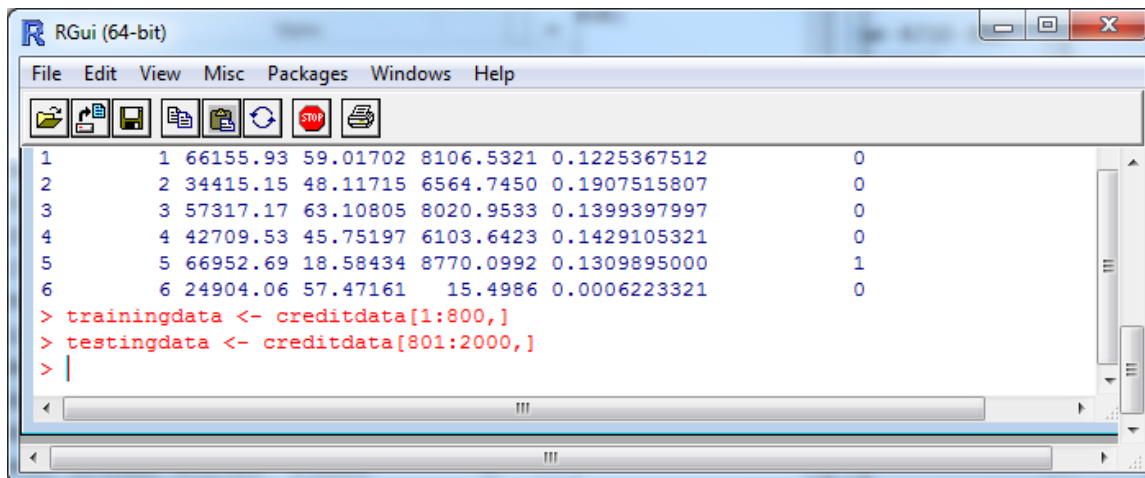
Whenever you build analytics models, it's a good idea to use some data to build the model and other data to test the model. R allows you to split the dataset into training and testing subsets.

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`.

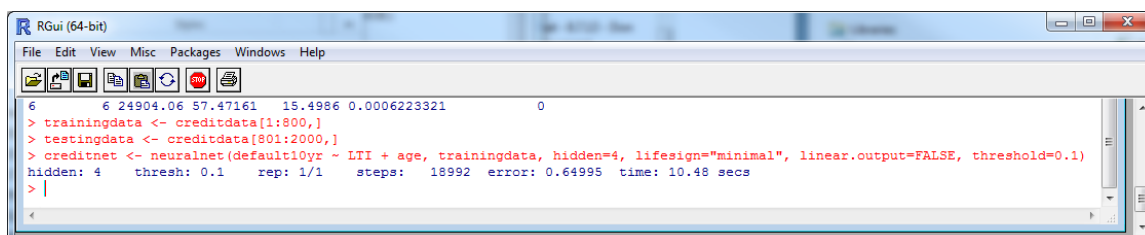


## Neural network analysis

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

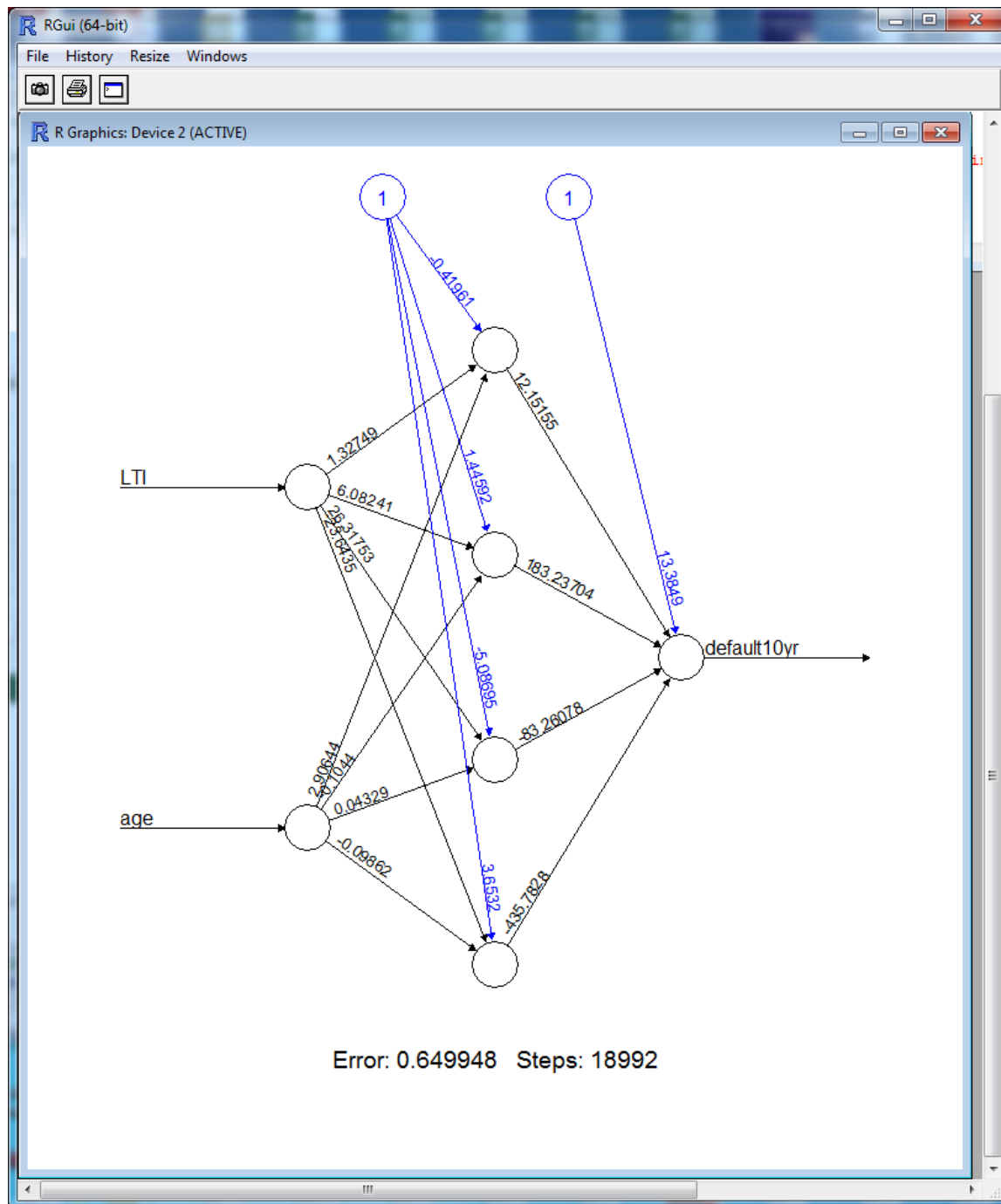


The neural network algorithm will perform a gradient search to find a solution that minimizes the error of making a mistake. In this case, the algorithm took 18,992 iterations or steps.

## 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)`

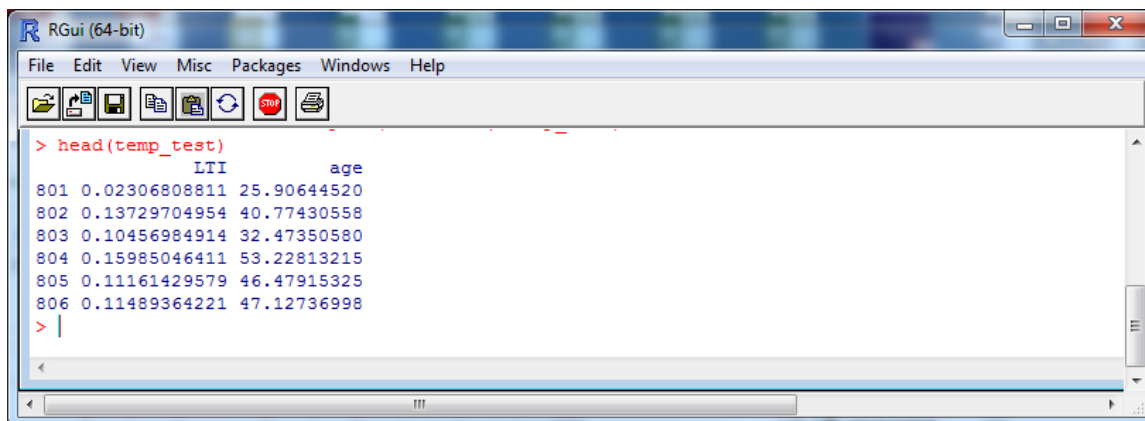




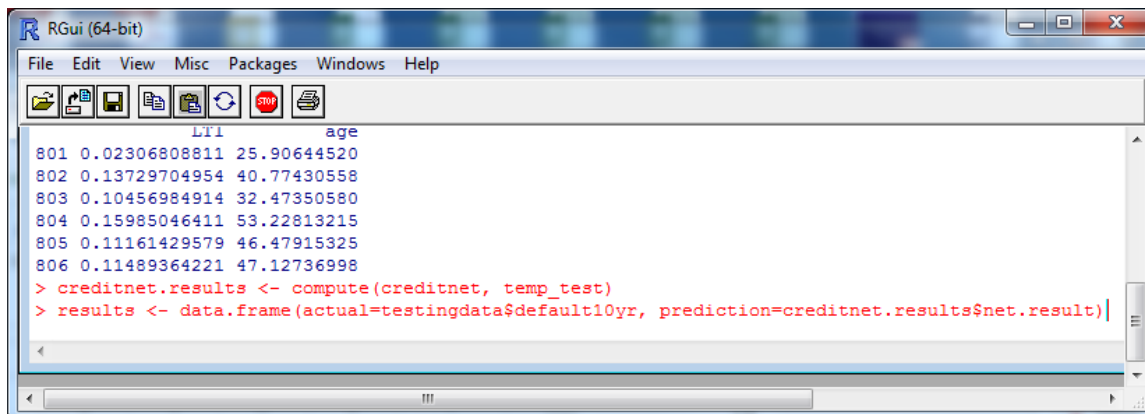
## Testing the Neural Network Prediction Ability

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)
```



## Session 9.9: Sensitivity Analysis of Neural Networks (live session)

Sensitivity analysis performed in Excel can assist in interpreting results from Neural Networks. In the following example, we will use passenger data from the Titanic to explore which factors are related to survival after the sinking of the Titanic. There is complete numeric data for 1,046 passengers. The variables in the data are:

Survived	Survival Indicator (0 = No, 1 = Yes)
Name	Passenger Name
Gender	Passenger's gender
GenderNum	Passenger's numeric gender (0 = Female, 1 = Male)
Age	Age in years
SiblingSpouse	Number of passengers on ship who are this person's brother, sister or spouse
ParentChild	Number of passengers on ship who are this person's parent or child
PClass	Passenger class (1 = 1 <sup>st</sup> , 2 = 2 <sup>nd</sup> , 3 = 3 <sup>rd</sup> )
Fare	Passenger fare
Embarked	Port of embarkation

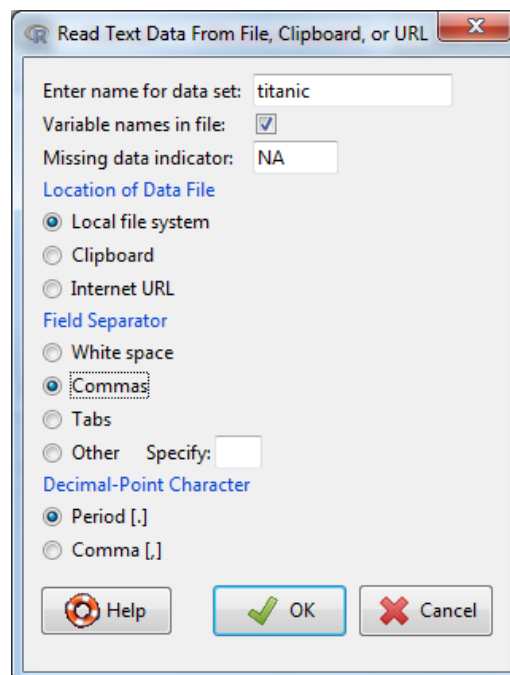
### Download Datasets

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

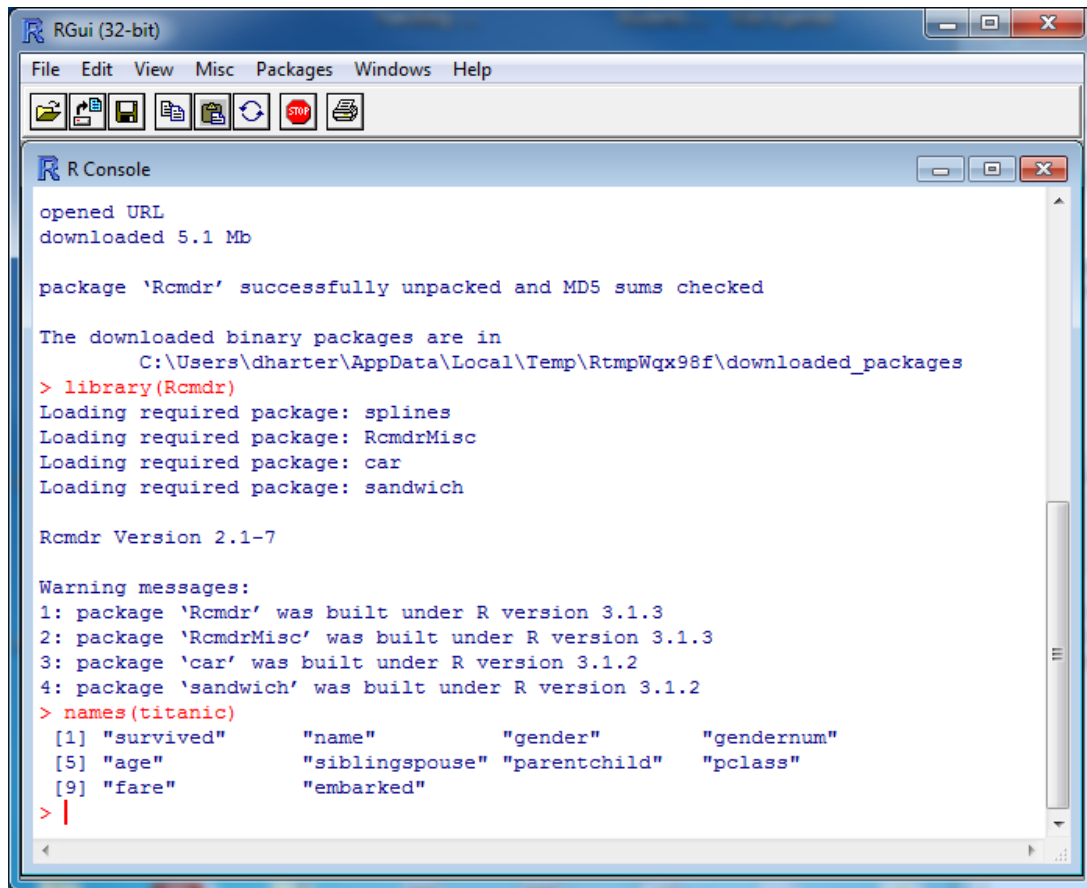
### Loading Data

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 titanic
4. Change Field Separator to Commas, then OK
5. Click on the Titanic.csv file, then Open



In the R Console (RGui), type names(titanic).



```
RGui (32-bit)
File Edit View Misc Packages Windows Help

R Console
opened URL
downloaded 5.1 Mb

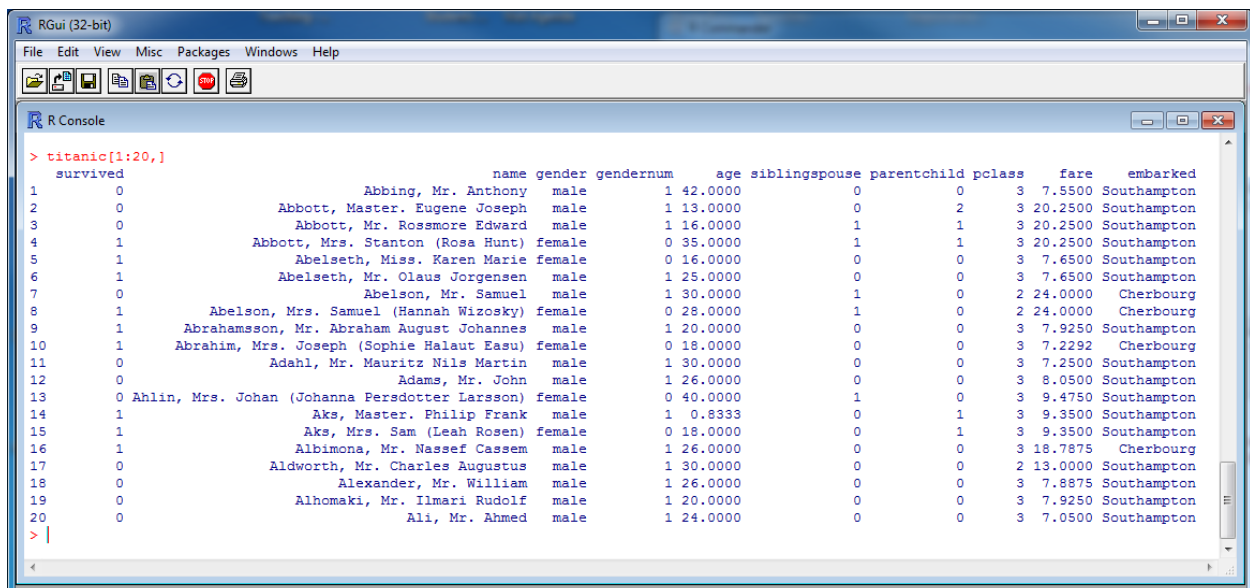
package 'Rcmdr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\dharter\AppData\Local\Temp\RtmpWqx98f\downloaded_packages
> library(Rcmdr)
Loading required package: splines
Loading required package: RcmdrMisc
Loading required package: car
Loading required package: sandwich

Rcmdr Version 2.1-7

Warning messages:
1: package 'Rcmdr' was built under R version 3.1.3
2: package 'RcmdrMisc' was built under R version 3.1.3
3: package 'car' was built under R version 3.1.2
4: package 'sandwich' was built under R version 3.1.2
> names(titanic)
 [1] "survived"      "name"          "gender"        "gendernum"
 [5] "age"           "siblingspouse" "parentchild"   "pclass"
 [9] "fare"         "embarked"
> |
```

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



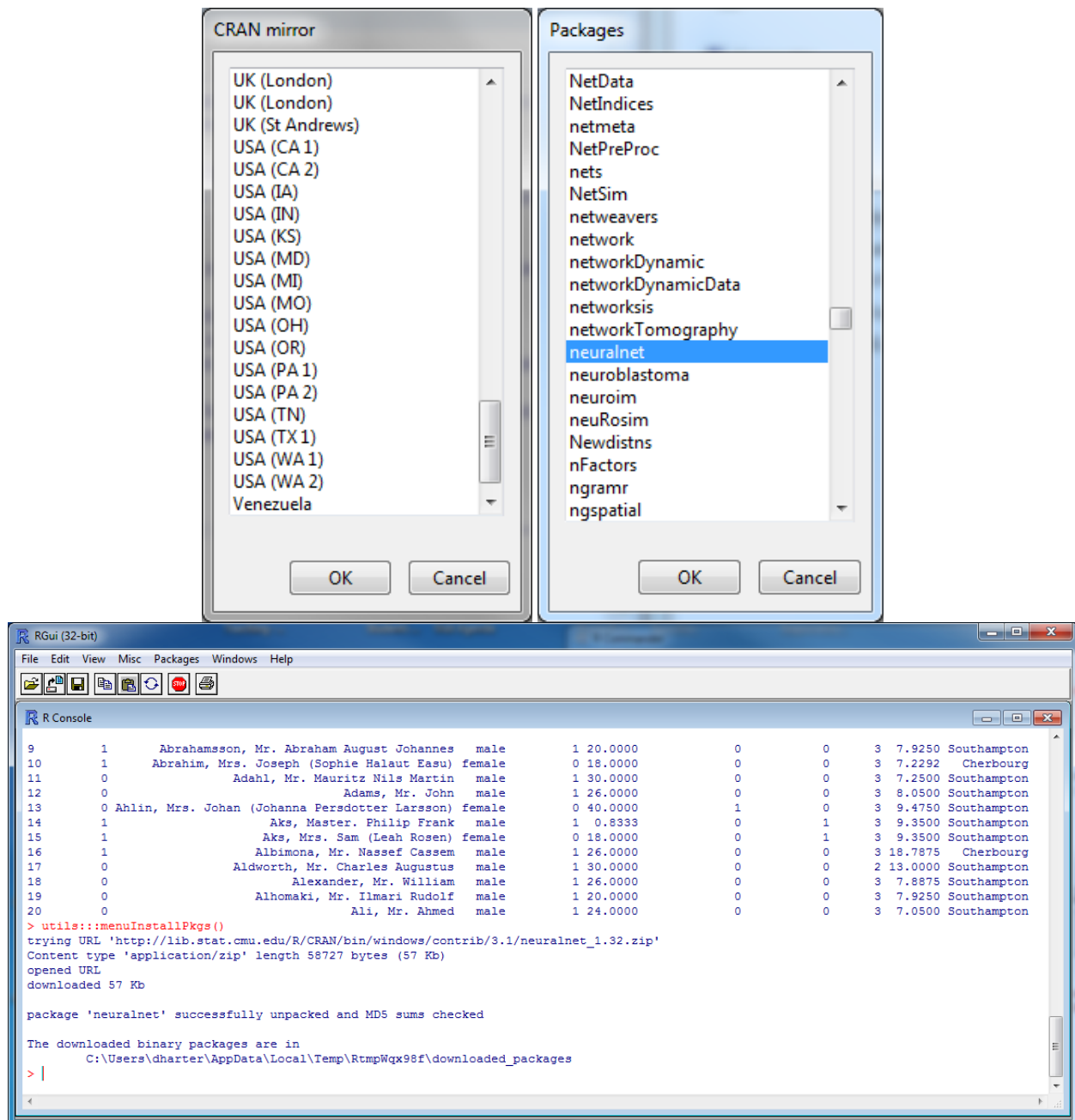
```
RGui (32-bit)
File Edit View Misc Packages Windows Help

R Console
> titanic[1:20,]
  survived      name gender gendernum   age siblingspouse parentchild pclass   fare embarked
1        0 Abbing, Mr. Anthony male    1 42.0000         0         0         3  7.5500 Southampton
2        0 Abbott, Master. Eugene Joseph male    1 13.0000         0         2         3 20.2500 Southampton
3        0 Abbott, Mr. Rossmore Edward male    1 16.0000         1         1         3 20.2500 Southampton
4        1 Abbott, Mrs. Stanton (Rosa Hunt) female    0 35.0000         1         1         3 20.2500 Southampton
5        1 Abbelseth, Miss. Karen Marie female    0 16.0000         0         0         3  7.6500 Southampton
6        1 Abbelseth, Mr. Olaus Jorgensen male    1 25.0000         0         0         3  7.6500 Southampton
7        0 Abelson, Mr. Samuel male    1 30.0000         1         0         2 24.0000 Cherbourg
8        1 Abelson, Mrs. Samuel (Hannah Wozosky) female    0 28.0000         1         0         2 24.0000 Cherbourg
9        1 Abrahamsson, Mr. Abraham August Johannes male    1 20.0000         0         0         3  7.9250 Southampton
10       1 Abraham, Mrs. Joseph (Sophie Halaut Easu) female    0 18.0000         0         0         3  7.2292 Cherbourg
11       0 Adahl, Mr. Mauritz Nils Martin male    1 30.0000         0         0         3  7.2500 Southampton
12       0 Adams, Mr. John male    1 26.0000         0         0         3  8.0500 Southampton
13       0 Ahlin, Mrs. Johan (Johanna Persdotter Larsson) female    0 40.0000         1         0         3  9.4750 Southampton
14       1 Aks, Master. Philip Frank male    1  0.8333         0         1         3  9.3500 Southampton
15       1 Aks, Mrs. Sam (Leah Rosen) female    0 18.0000         0         1         3  9.3500 Southampton
16       1 Albimona, Mr. Nassef Cassem male    1 26.0000         0         0         3 18.7875 Cherbourg
17       0 Aldworth, Mr. Charles Augustus male    1 30.0000         0         0         2 13.0000 Southampton
18       0 Alexander, Mr. William male    1 26.0000         0         0         3  7.8875 Southampton
19       0 Alhomaki, Mr. Ilmari Rudolf male    1 20.0000         0         0         3  7.9250 Southampton
20       0 Ali, Mr. Ahmed male    1 24.0000         0         0         3  7.0500 Southampton
> |
```

## Installing NeuralNet

If you do not have neuralnet installed, follow these steps 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

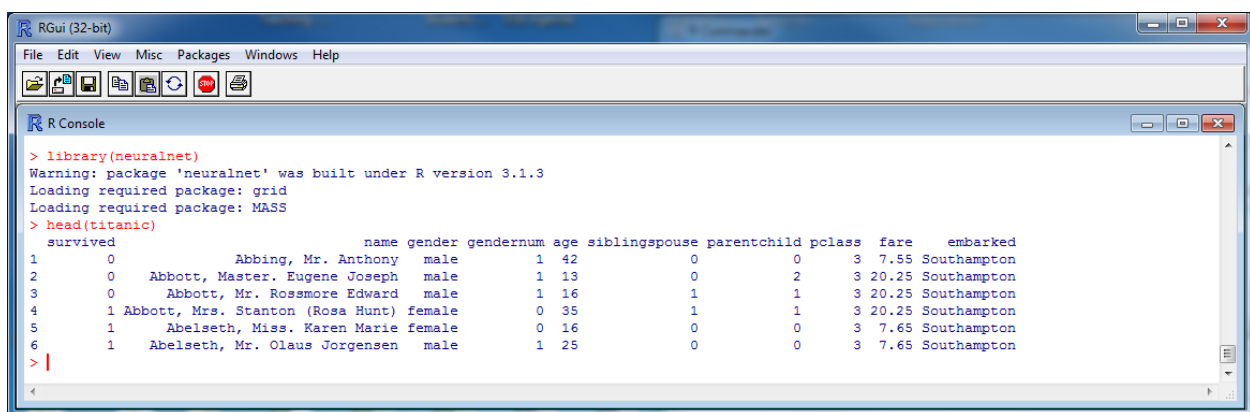
neuralnet is the R software that performs neural network calculations:

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(titanic)`



```
> library(neuralnet)
Warning: package 'neuralnet' was built under R version 3.1.3
Loading required package: grid
Loading required package: MASS
> head(titanic)
  survived      name gender gendernum age siblings spouse parent child polclass  fare embarked
1        0  Abbing, Mr. Anthony   male         1    42         0         0         0         3    7.55 Southampton
2        0 Abbott, Master. Eugene Joseph   male         1    13         0         2         3    20.25 Southampton
3        0 Abbott, Mr. Rossmore Edward   male         1    16         1         1         3    20.25 Southampton
4        1 Abbott, Mrs. Stanton (Rosa Hunt) female         0    35         1         1         3    20.25 Southampton
5        1  Abbelseth, Miss. Karen Marie female         0    16         0         0         3     7.65 Southampton
6        1  Abbelseth, Mr. Olaus Jorgensen   male         1    25         0         0         3     7.65 Southampton
```

## 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:

```
titanicnet <- neuralnet(survived ~ gendernum + age, titanic, hidden=2, lifesign="minimal",
linear.output=FALSE, threshold=0.01)
```

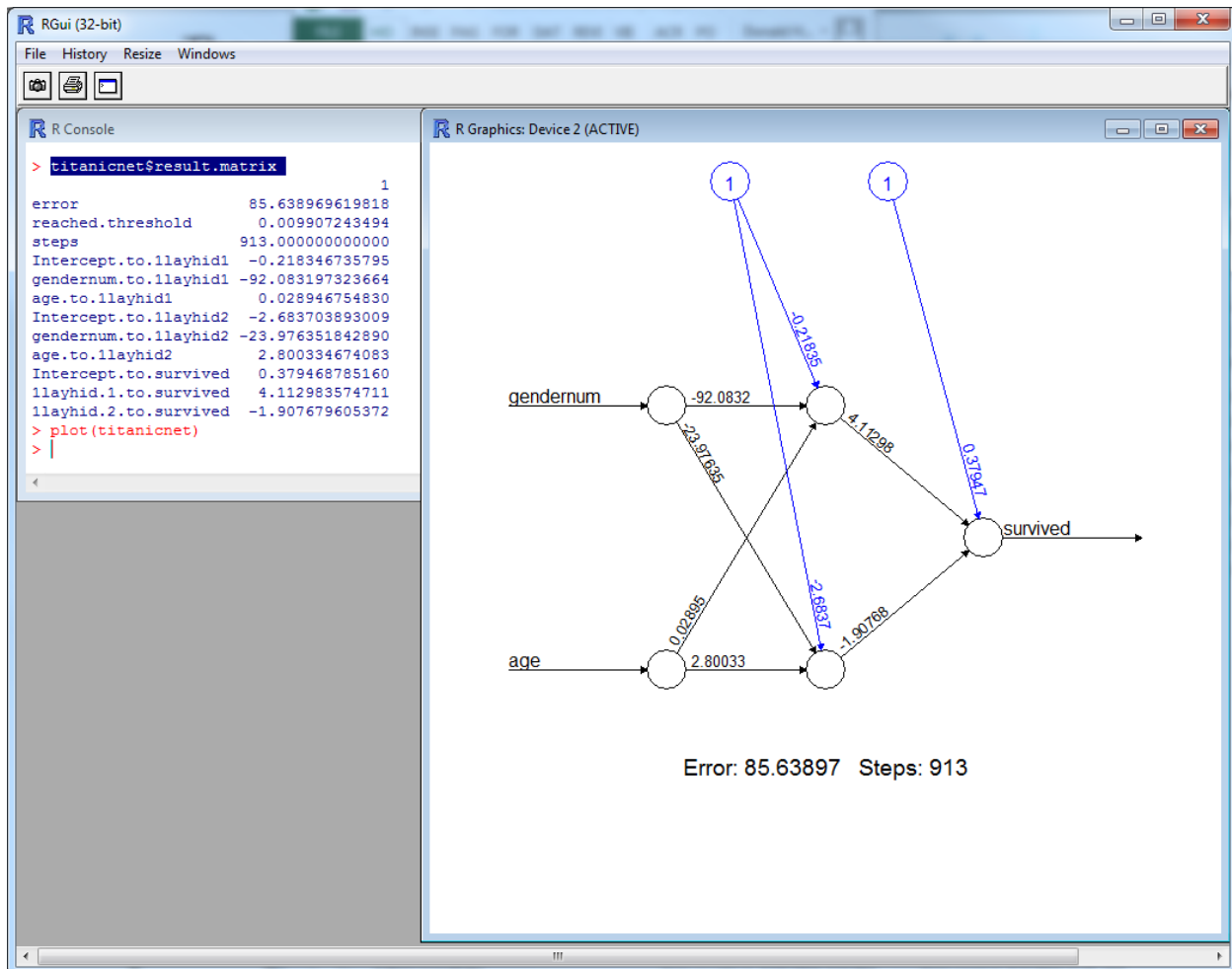
titanicnet	stores the results
neuralnet	program which runs the neural network analysis
survived	dependent variable (0 = did not survive, 1 =survived)
gendernum	independent variable – gender (0 = female, 1 = male)
age	independent variable – age in years
hidden	number of hidden nodes
lifesign	amount of 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 913 iterations or steps.

## Neural Network Model

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

1. To list the coefficients, type the command `titanicnet$result.matrix`
2. To generate the graph, in the R console (RGui), type the command `plot(titanicnet)`



## Neural Network Prediction

Interpreting the results of a neural network is often easier to do through a spreadsheet prediction and sensitivity analysis. We will take the coefficients from the neural network and build an Excel spreadsheet to calculate the predictions of survivability of Titanic passengers, using gender and age.

Let's build the neural network calculations:

1. Open a blank Excel spreadsheet
2. The two input variables which we used in the neural network were Gender and Age.
  - a. Type Gender into A1, Age into A2
  - b. Type in a sample gender number into B1 (0 for female)
  - c. Type a sample age into B2 (in this case, 10 for 10 years old)
  - d. Enter labels in D1 and D2 to make it easier to remember
3. In cell A5, enter Hidden node 1:
4. In row 7, type the label Variable, Coefficient, Value and Coeff\*Value in cells A7 through D7
5. In cells A8, A9, and A10, type the labels Intercept, Gender, and Age
6. In cells B8, B9, and B10, type the coefficients for hidden node 1 from the R results
  - a. B8, enter the results.matrix value from Intercept.to.1layhid1
  - b. B9, enter the results.matrix value from gendernum.to.1layhid1
  - c. B10, enter the results.matrix value from age.to.1layhid2
7. In cell C8, enter the number 1
8. In cell C9, point to the value for Gender by typing =B1
9. In cell C10, point to the value for Age by typing =B2
10. In cells D8, D9, D10, enter formulas to multiply coefficients and values
  - a. In D8, enter =B8\*C8
  - b. In D9, enter =B9\*C9
  - c. In D10, enter =B10\*C10
11. In cell D12, enter the formula for the sum of the column D calculations, i.e., =sum(D8:D10)
12. In cell D13, calculate the exponential of the sum, i.e., =exp(D12)
13. In cell D14, calculate the probability, i.e., =D13/(1+D13)



NeuralNetworkSensitivityAnaly...				
FILE	HO	INSE	PAG	FOR
Get External Data	Refresh All	Sort & Filter	Data Tools	Outline
Analysis				
A1 : Gender				
1	Gender	0	0=female, 1=male	
2	Age	10	age in years	
3				
4				
5	Hidden node 1:			
6				
7	Variable	Coefficient	Value	Coeff*Value
8	Intercept	-0.218346736	1	-0.218346736
9	Gender	-92.08319732	0	0
10	Age	0.028946755	10	0.289467548
11				
12			sum	0.071120813
13			Exp(sum)	1.073710936
14			Probability	0.517772712

14. Next, build similar calculations for Hidden node 2. From the result.matrix, use the coefficients:
- Intercept.to.1layhid2
  - Gendernum.to.1layhid2
  - Age.to.1layhid2

NeuralNetworkSensitivityAnalysis - Excel

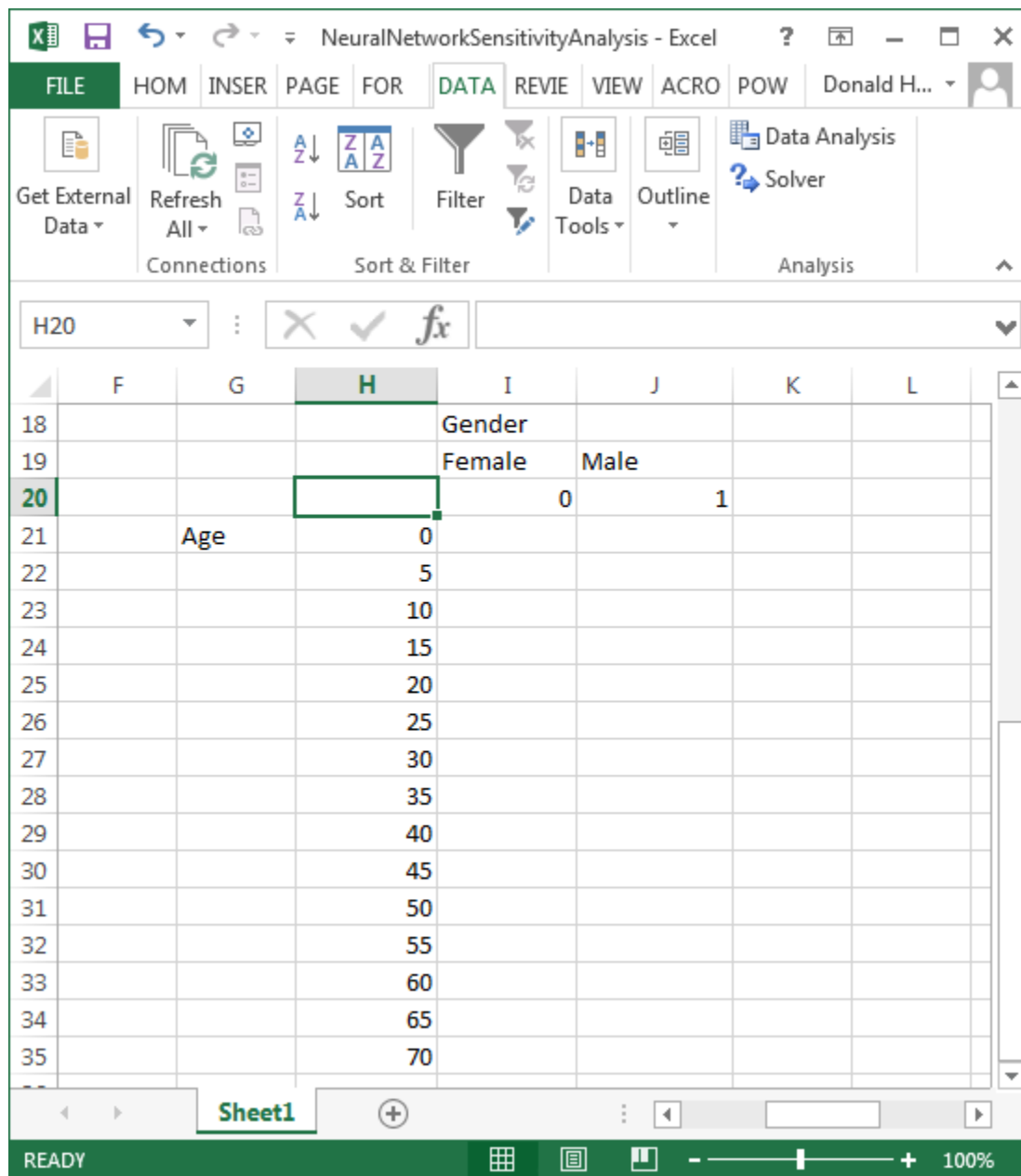
	A	B	C	D	E	F	G	H	I	J
1	Gender	0		0=female, 1=male						
2	Age	10		age in years						
3										
4										
5	Hidden node 1:						Output:			
6										
7	Variable	Coefficient	Value	Coeff*Value			Variable	Coefficient	Value	Coeff*Value
8	Intercept	-0.218346736	1	-0.218346736			Intercept	0.37946879	1	0.379468785
9	Gender	-92.08319732	0	0			Hidden1	4.11298357	0.51777271	2.129590661
10	Age	0.028946755	10	0.289467548			Hidden2	-1.9076796	1	-1.90767961
11										
12			sum	0.071120813					sum	0.601379841
13			Exp(sum)	1.073710936					Exp(sum)	1.82463477
14			Probability	0.517772712					Probability	0.645971929
15										
16	Hidden node 2:									
17										
18	Variable	Coefficient	Value	Coeff*Value						
19	Intercept	-2.683703893	1	-2.683703893						
20	Gender	-23.97635184	0	0						
21	Age	2.800334674	10	28.00334674						
22										
23			sum	25.31964285						
24			Exp(sum)	99124537178						
25			Probability	1						

15. Finally, build the calculations for the Output node.
- From the result.matrix, use the coefficients:
    - Intercept.to.survived
    - 1layhid.1.to.survived
    - 1layhid.2.to.survived
  - For values, point to:
    - Cell I9 (Hidden1), point to the probability for Hidden1, i.e., =D14
    - Cell I10 (Hidden2), point to the probability for Hidden2, i.e., =D25
16. Change the values for Gender and Age in B1 and B2. The probability of surviving the Titanic appears in J14 will reflect the changes.

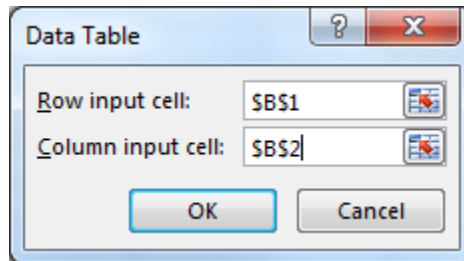
## Sensitivity Analysis

Now that the calculations have been created for the Neural Network, let's develop a two-way sensitivity analysis.

1. In cell I18, type Gender
2. In cells I19 and J19, type Female and Male; these will not be used, but will help us label options
3. In cells I20 and J20, type 0 and 1, to represent the genderum for Gender
4. In cell G21, type Age
5. In cells H21 through H35, type in zero through 70, incrementing by 5 years for age



6. In cell H20, point to the formula for probability, i.e., =J14
7. Highlight H20 through J35
8. Click on the Data tab, What-If-Analysis, then Data Table
9. Since Gender varies across the row, enter B1 for Row input cell
10. Since Age varies down the column, enter B2 for column input cell
11. Click OK
12. Highlight cells I21 through J35, change to Percentage format



NeuralNetworkSensitivityAnalysis - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW ACROBATIC POWERS

Paste Font Alignment Number Conditional Formatting Format as Table Cell Styles

G18

	F	G	H	I	J	K	L
18				Gender			
19				Female	Male		
20			0.64597193	0	1		
21		Age	0	89%	59%		
22			5	61%	59%		
23			10	65%	24%		
24			15	68%	18%		
25			20	71%	18%		
26			25	74%	18%		
27			30	76%	18%		
28			35	79%	18%		
29			40	81%	18%		
30			45	82%	18%		
31			50	84%	18%		
32			55	85%	18%		
33			60	86%	18%		
34			65	87%	18%		
35			70	88%	18%		

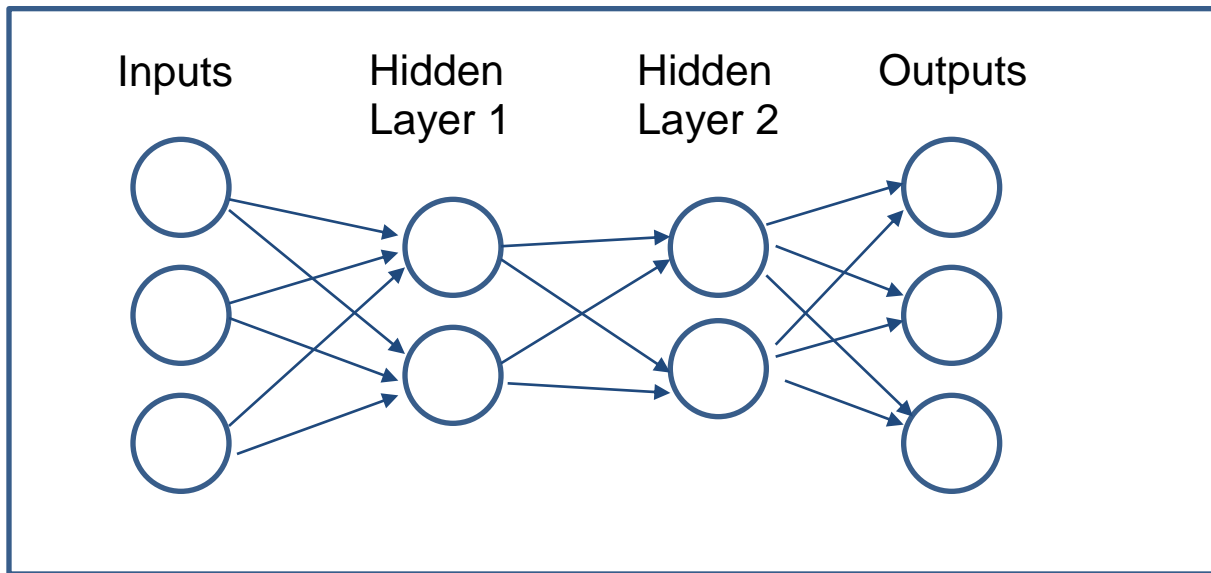
Sheet1

READY 100%

What pattern do you see?

## Session 9.10: Deep Neural Networks (supplemental material)

Neural networks are not limited to one hidden layer. Neural networks which have more than one hidden layer are called Deep Neural Networks and have the ability to learn much more subtle patterns and strategies. An example of a deep neural network might look like:



Using the Titanic data once again, let's use two inputs (gender, age), two hidden layers (with two nodes each), and one output. The command becomes

```
titanicnet <- neuralnet(survived ~ gendernum + age, titanic, hidden=c(2,2), lifesign="minimal",  
linear.output=FALSE, threshold=0.01)
```

titanicnet	stores the results
neuralnet	program which runs the neural network analysis
survived	dependent variable (0 = did not survive, 1 = survived)
gendernum	independent variable – gender (0 = female, 1 = male)
age	independent variable – age in years
hidden	number of hidden nodes at each level: c(2,4) would mean 2 in hidden layer 1, 4 in hidden layer 2
lifesign	amount of output
linear.output	whether you want linear or non-linear model
threshold	error term threshold

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

1. To generate the graph, in the R console (RGui), type the command `plot(titanicnet)`

