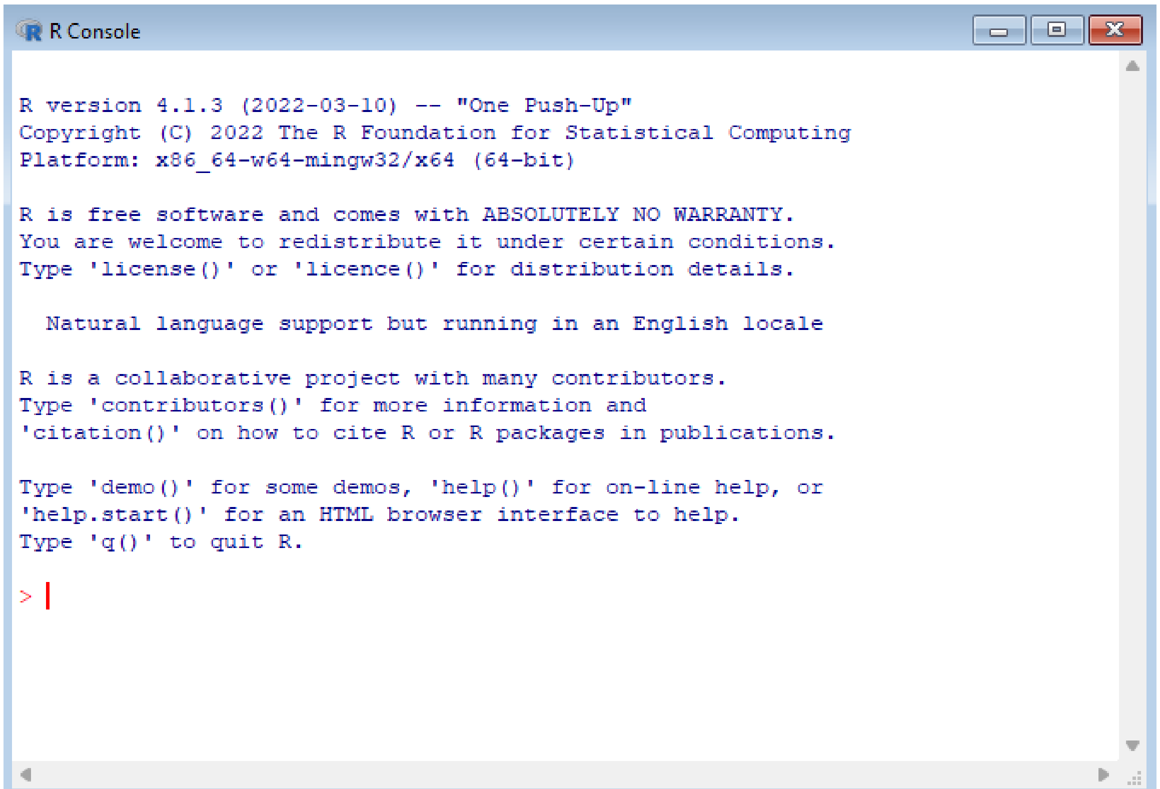
R: Advanced

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

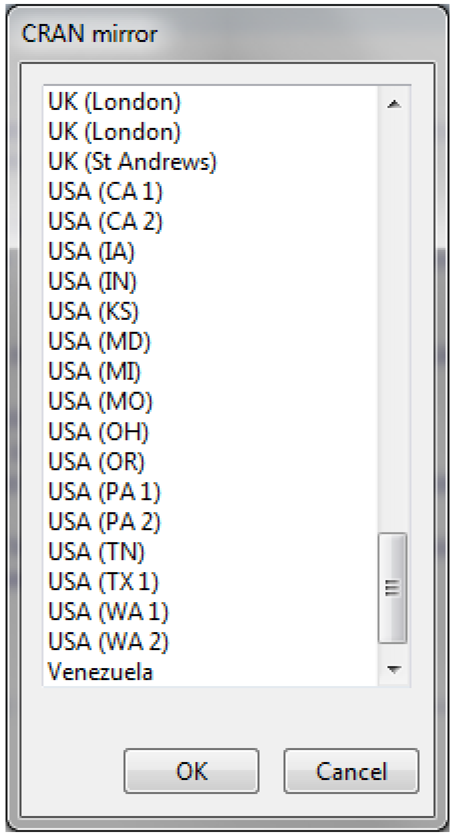


3. This is the command line screen. You can enter commands but need to know the syntax.
4. There is a simpler approach to running R, called Rcmdr (R Commander). If you are running a Whitman computer, Rcmdr is already installed. If not, you need to install it.

Installing R Commander

Follow these steps only if you do not already have Rcmdr installed.

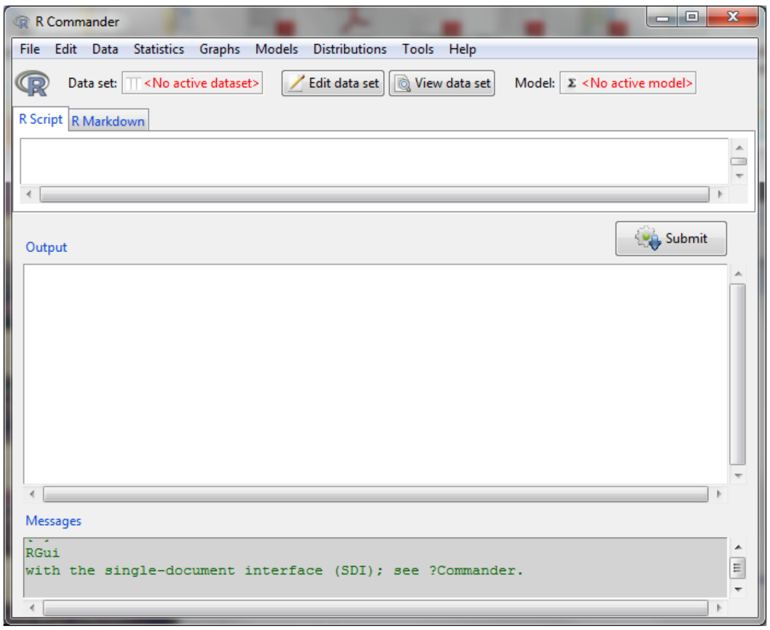
1. In R, type the command:
   install.packages("Rcmdr", dependencies = TRUE)
2. In the CRAN mirror, select the location closest to you; use a USA location near you, then click OK
3. If prompted to create a personal library, click Yes
4. If prompted to add missing packages, click Yes



Launch Rcmdr (R Commander)

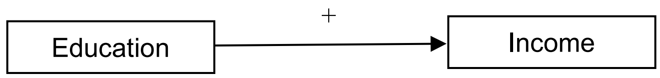
Rcmdr is a graphical user interface (GUI) that is easier to use than the command line. To launch Rcmdr:

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



Modeling – Regression

So far, we have been performing regressions on a dependent variable Y against an independent variable X. For example, we can examine how education (X) affects income (Y). Pictorially, this would appear as:



The line and arrow identify a relationship between education and income. The plus sign above the line indicates that the relationship is positive, i.e., if education increases, then income increases.

This relationship can be written as:

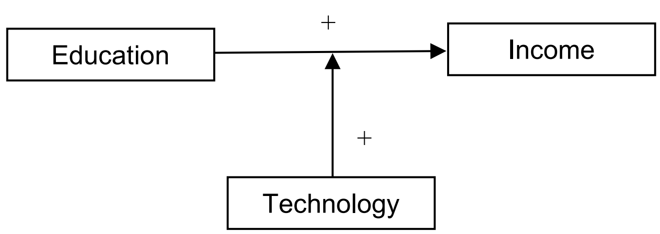
Income = f(Education)

Which means that income is a function of education. One formulation of this could be the linear relationship:

$$\text{Income} = \beta_0 + \beta_1 * \text{Education}$$

Where β_0 is the intercept and β_1 is the coefficient for education.

Now consider a third variable: technology. Technology has the potential for increasing the value of educated employees. Technology itself does not generate income for an employee but affects the value of education. This is called a moderating variable and is shown as:



This new model means that as education increases, income increases. The moderating effect of technology on education implies that technology further increases the value of education. This is modeled as an interaction term:

$$\text{Income} = \beta_0 + \beta_1 * \text{Education} + \beta_2 * \text{Education} * \text{Technology}$$

Therefore, the effect of education on income is influenced by the level of technology that an employee has.

Summary

A dummy variable changes the intercept. A moderating effect (interaction) changes the slope.

Download Datasets

Download the Titanic dataset from BlackBoard or the G: drive.

Loading Data

To load data into R:

1. Click on Data at the top of the screen
2. Click on Import Data > from Excel file …
3. Enter the name that you would like to use for this data set; type in titanic
4. Click OK
5. Find the titanic file on your computer, then Open

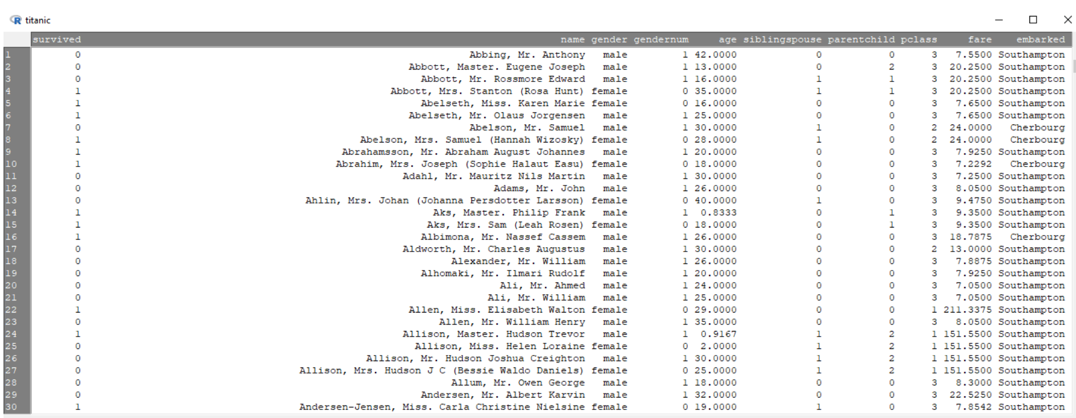
Viewing data fields

In the following example, we will use passenger data from the Titanic to explore which factors relate 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 = 1st, 2 = 2nd, 3 = 3rd)
Fare	Passenger fare
Embarked	Port of embarkation

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

1. Click on the button View data set


titanic

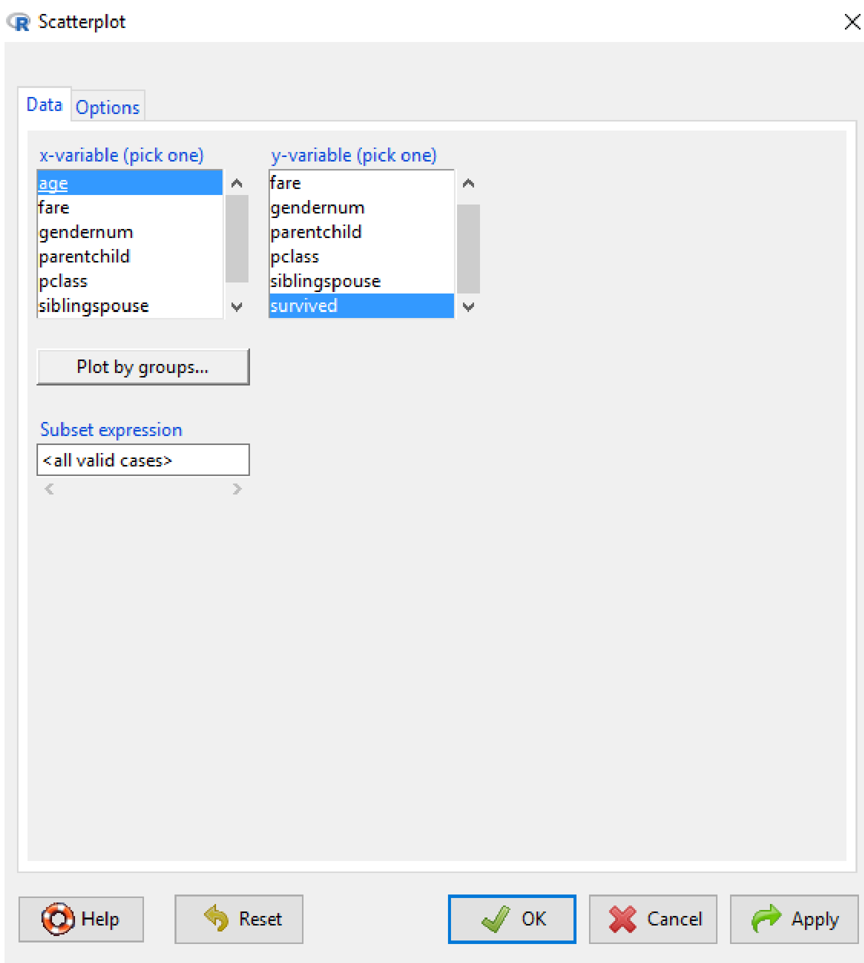
	survived	name	gender	gendernum	age	siblingspouse	parentchild	pclass	fare	embarked
1	0	Abbing, Mr. Anthony	male	1	42.0000	0	0	3	7.5500	Southampton
2	0	Abbott, Master. Eugene Joseph	male	1	13.0000	0	2	3	20.2500	Southampton
3	0	Abbott, Mr. Rossmore Edward	male	1	16.0000	1	1	3	20.2500	Southampton
4	1	Abbott, Mrs. Stanton (Rosa Hunt)	female	0	35.0000	1	1	3	20.2500	Southampton
5	1	Abelseth, Miss. Karen Marie	female	0	16.0000	0	0	3	7.6500	Southampton
6	1	Abelseth, 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 Wizosky)	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	Abrahim, 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
21	0	Ali, Mr. William	male	1	25.0000	0	0	3	7.0500	Southampton
22	1	Allen, Miss. Elisabeth Walton	female	0	29.0000	0	0	1	211.3375	Southampton
23	0	Allen, Mr. William Henry	male	1	35.0000	0	0	3	8.0500	Southampton
24	1	Allison, Master. Hudson Trevor	male	1	0.9167	1	2	1	151.5500	Southampton
25	0	Allison, Miss. Helen Loraine	female	0	2.0000	1	2	1	151.5500	Southampton
26	0	Allison, Mr. Hudson Joshua Creighton	male	1	30.0000	1	2	1	151.5500	Southampton
27	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	0	25.0000	1	2	1	151.5500	Southampton
28	0	Allum, Mr. Owen George	male	1	18.0000	0	0	3	8.3000	Southampton
29	0	Andersen, Mr. Albert Karvin	male	1	32.0000	0	0	3	22.5250	Southampton
30	1	Andersen-Jensen, Miss. Carla Christine Nielsine	female	0	19.0000	1	0	3	7.8542	Southampton

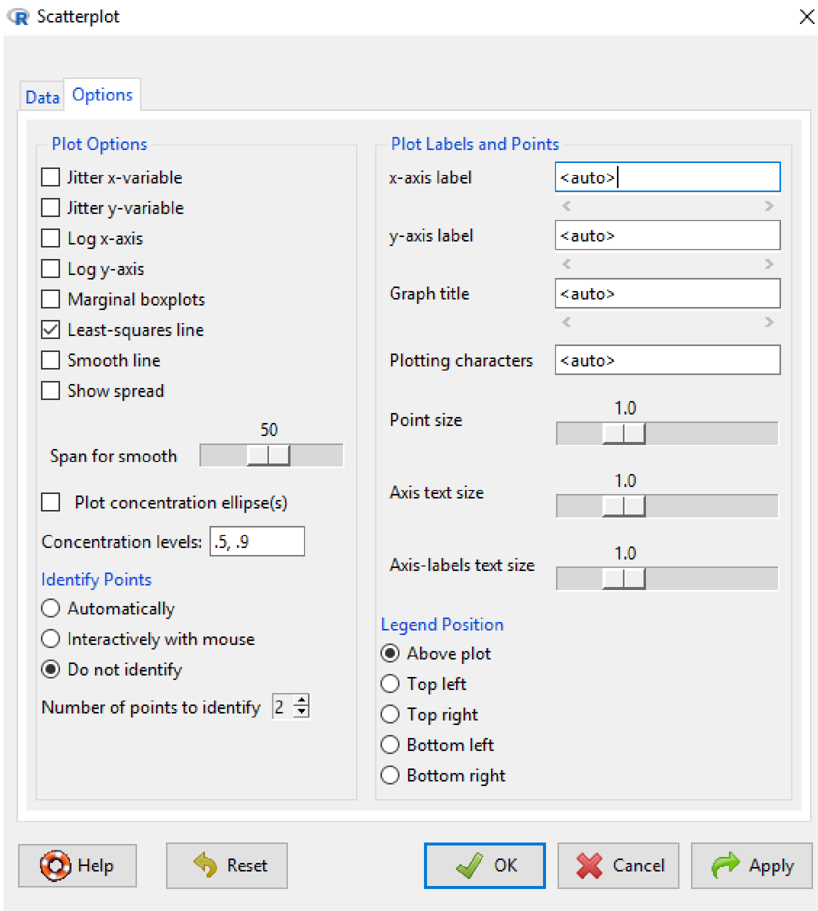
2. Which variables should affect whether someone survived? Why?
3. Click on the X in the upper right corner of the data display to close the data view.

Scatterplots

To generate a scatter plot,

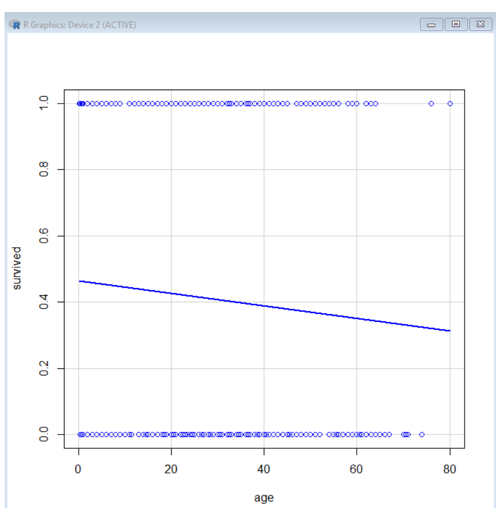
1. Click on Graphs, Scatterplot
2. Select age as the x-variable
3. Select survived as the y-variable
4. Click on the Options tab, then select check the box for Least-squares line
5. Click on OK



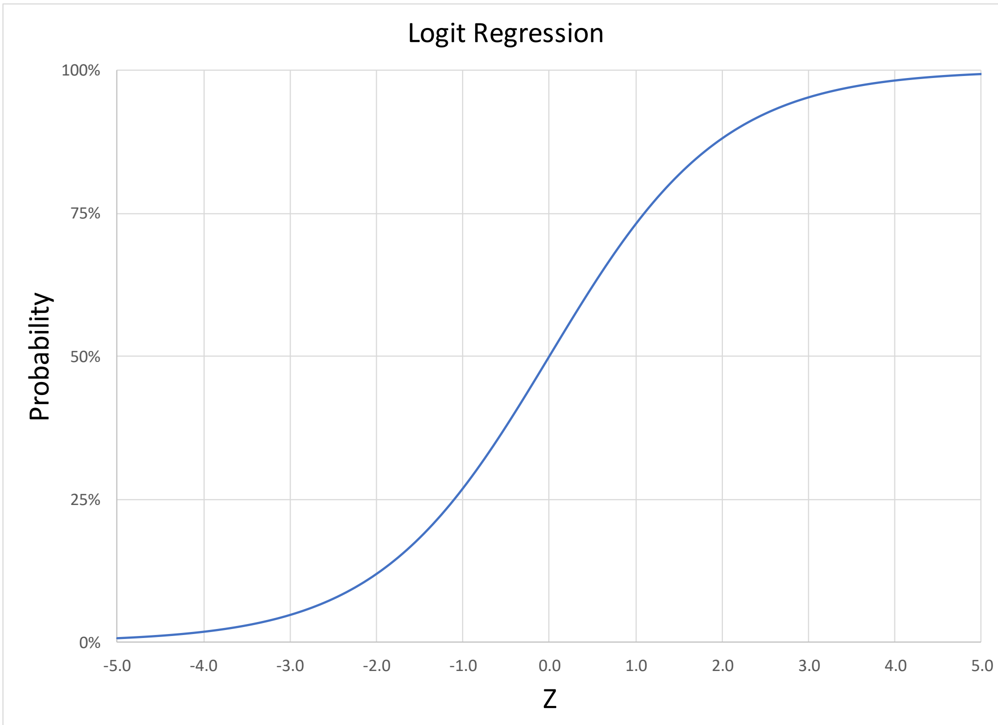


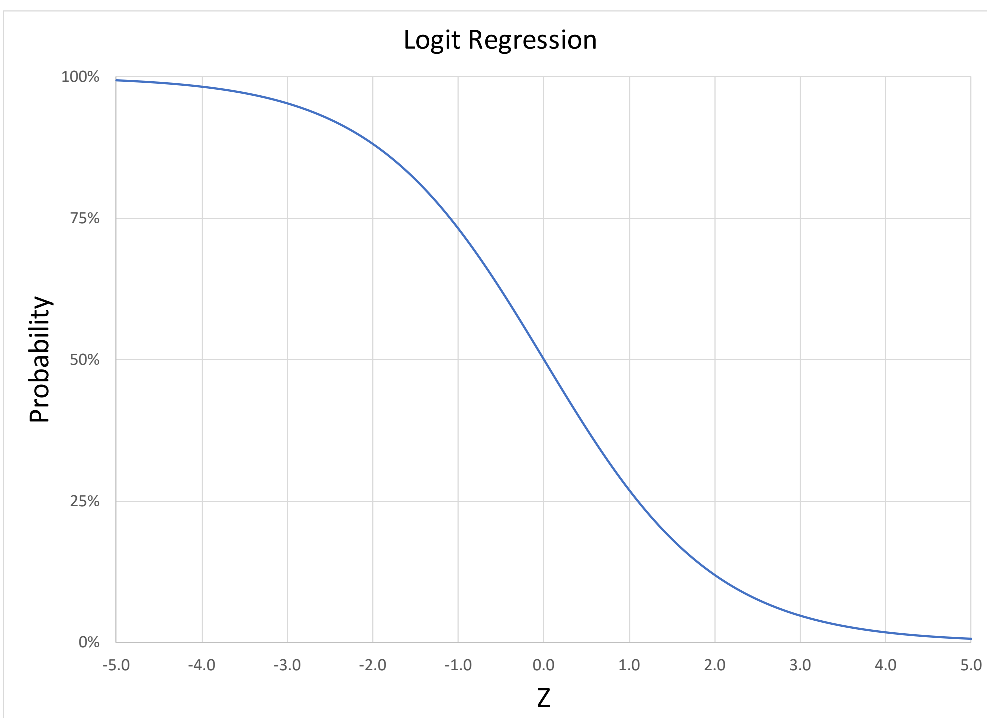
To interpret the chart:

1. The blue dots are the age versus survived (1) or not survived (0)
2. The blue 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:





The Logit regression uses the logistic function; it is either an S-shaped curve monotonically increasing or an S-shaped curve monotonically decreasing. Probit is similar, but uses the normal distribution function.

Key differences between Logit and Probit

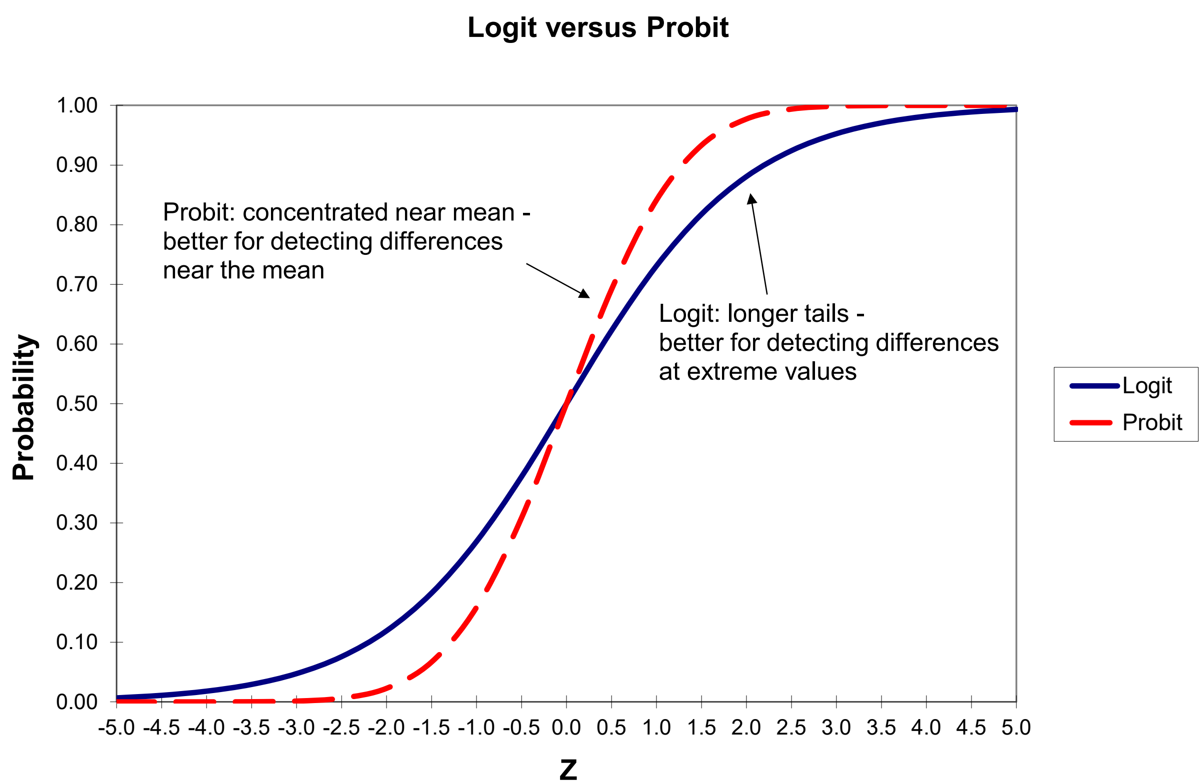
Logit and probit are techniques that assume the dependent variable (Y) is zero or one, and finds the relationship between the explanatory variables (X) and the dependent variable (Y). Logistic regression and logit are based on the logistic distribution. Probit is based on the normal distribution. Logit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

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

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

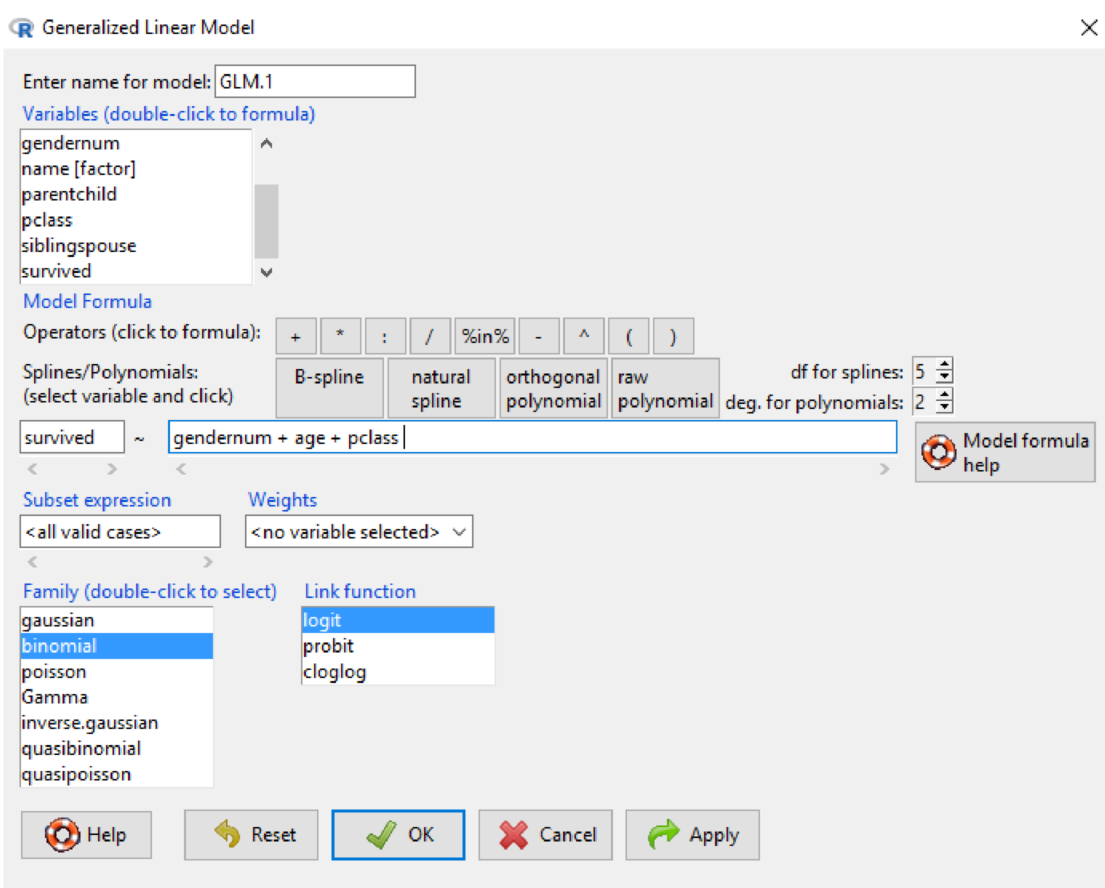
$P(Y=1) = \Phi(\Sigma\beta_i X_i)$ where Φ is the normal distribution

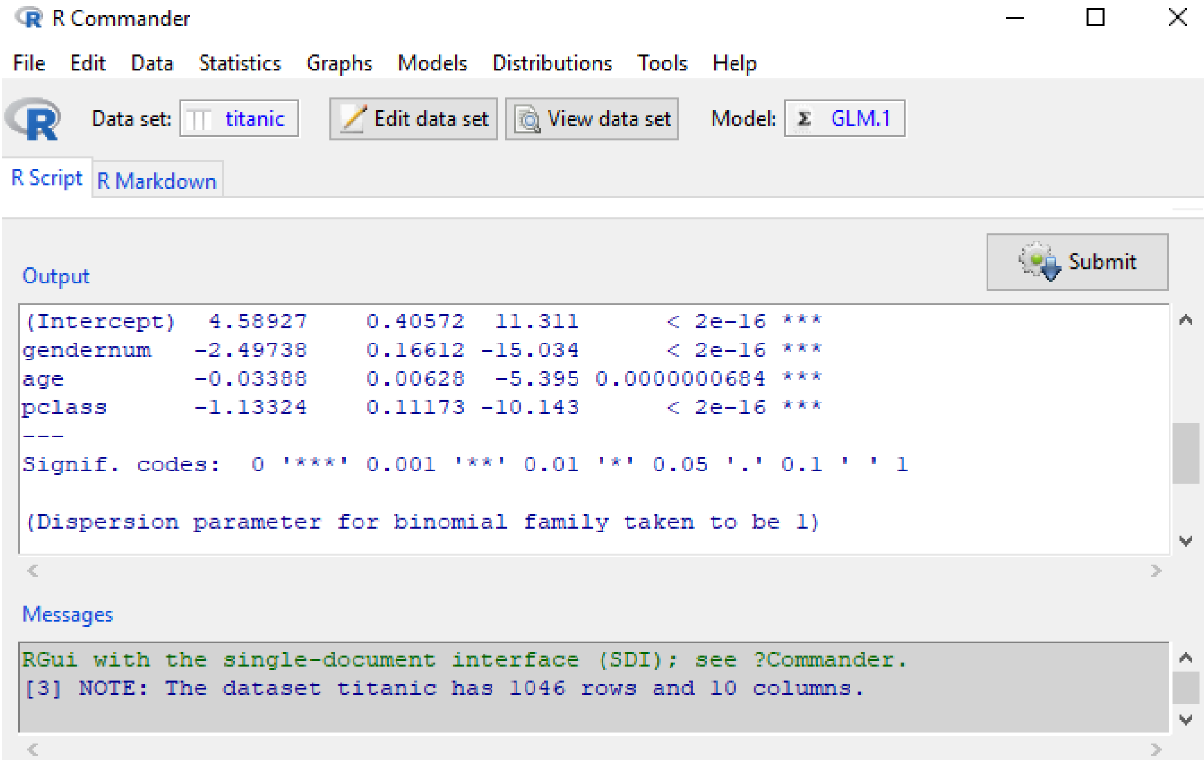


Logit Analysis

To perform a logit analysis on our data, where the Y variable is survived and the explanatory variables are gendernum, age, pclass:

1. Click on Statistics, Fit models, Generalized linear model
2. Double click on survived for the dependent variable
3. Double click on gendernum, age, pclass for the explanatory variables
4. Select Family as binomial
5. Select Link function as logit
6. Click OK



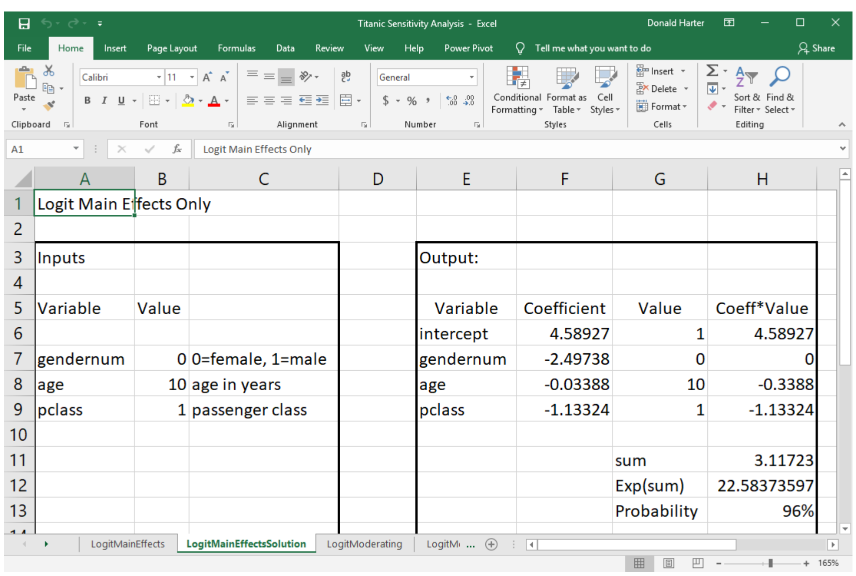


7. Are the coefficients positive or negative?
8. As gendernum goes from 0 to 1, what happens to survivability? What does this mean?
9. As age increases, what happens to survivability? What does this mean?
10. As pclass increases, what happens to survivability? What does this mean?
11. Are the coefficients statistically significant?

Calculating Logit Probabilities

You can use a spreadsheet to calculate logit probabilities for different combinations of the explanatory variable values. Let's calculate the probability of a 10-year-old girl in 1st class surviving the Titanic. Use the data from the previous page.

1. For this exercise, use the template called Titanic Sensitivity Analysis.
2. Click on the tab LogitMainEffects
3. The left section allows you to enter the values for gendernum, age, and pclass.
   a. In cell B7, enter 0 for a girl
   b. In cell B8, enter 10 for 10 years old
   c. In cell B9, enter 1 for first class
4. Next, enter the coefficients for each variable in cells F6 through F9.
5. Enter a formula for the values in G7 through G9 pointing to the input values
   a. In G7, enter =B7
   b. In G8, enter =B8
   c. In G9, enter =B9
   d. For G6, enter 1
6. In cells H6 through H9, enter formulas multiplying the coefficients and values
7. The logistic function is the exponential of the sum divided by (1+ exponential of the sum)
8. First, calculate the sum. In cell H11, enter =sum(H6:H9)
9. Next, calculate the exponential of the sum. In cell H12, enter =exp(H11)
10. Finally, in cell H13, enter =H12/(1+H12). This is your probability of surviving.
11. Change the input values from male to female; change the age; change the passenger class. What happens to the probability?



	A	B	C	D	E	F	G	H
1	Logit Main Effects Only							
2								
3	Inputs				Output:			
4								
5	Variable	Value			Variable	Coefficient	Value	Coeff*Value
6					intercept	4.58927	1	4.58927
7	gendernum	0	0=female, 1=male		gendernum	-2.49738	0	0
8	age	10	age in years		age	-0.03388	10	-0.3388
9	pclass	1	passenger class		pclass	-1.13324	1	-1.13324
10								
11							sum	3.11723
12							Exp(sum)	22.58373597
13							Probability	96%

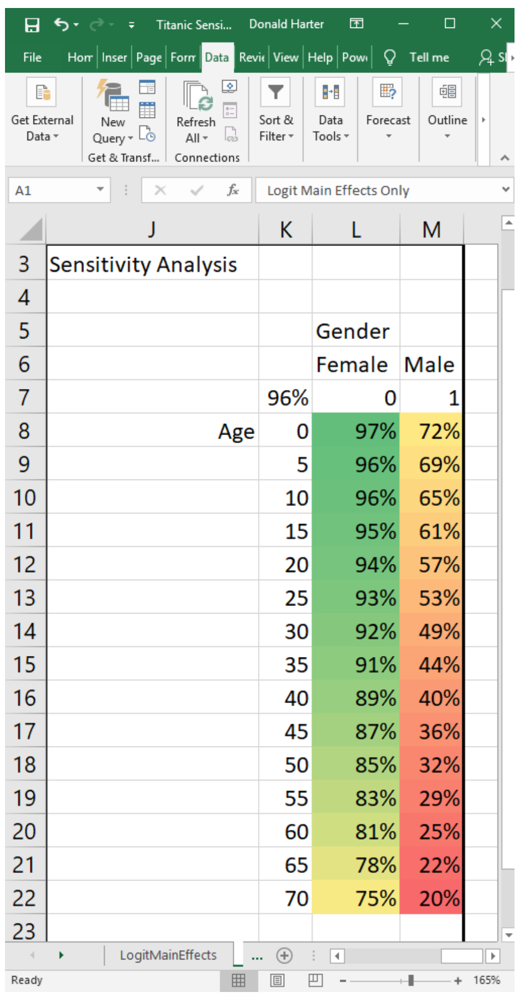
The Logit regression uses the logistic function to calculate the probability:

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

Logit Sensitivity Analysis

To perform a sensitivity analysis of gender versus age, use the Sensitivity section of the same spreadsheet.

1. In general, for a sensitivity analysis, put one dimension across the top (in this case, gender), and one dimension down the side (in this case age).
2. In the cell in the corner, you must enter the formula for the probability calculation. This is already stored in cell H13. In cell K7, enter =H13.
3. Highlight K7 through M22
4. Click on the Data tab, then What If analysis, Data Table
5. For Row input cell, point to the cell with the value for gendernum, in this case B7
6. For Column input cell, point to the cell with the value for age, in this case B8
7. Click OK
8. What pattern do you see?
9. Notice that for each gender, the curve is monotonic (always increasing or always decreasing)

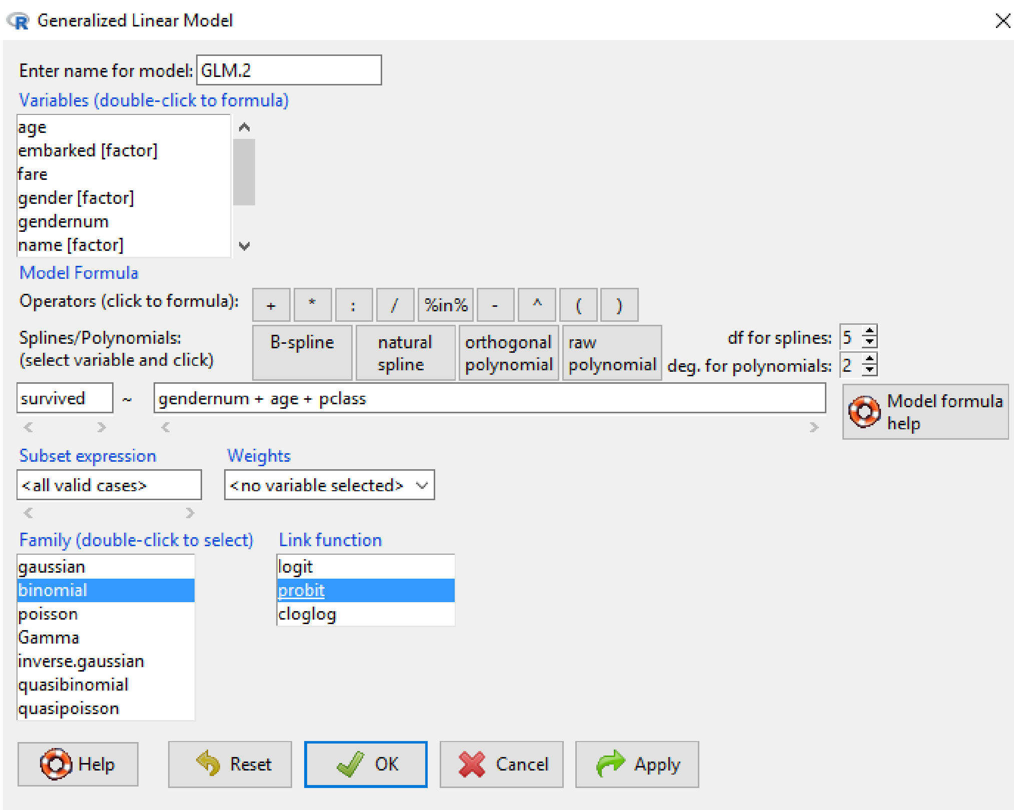


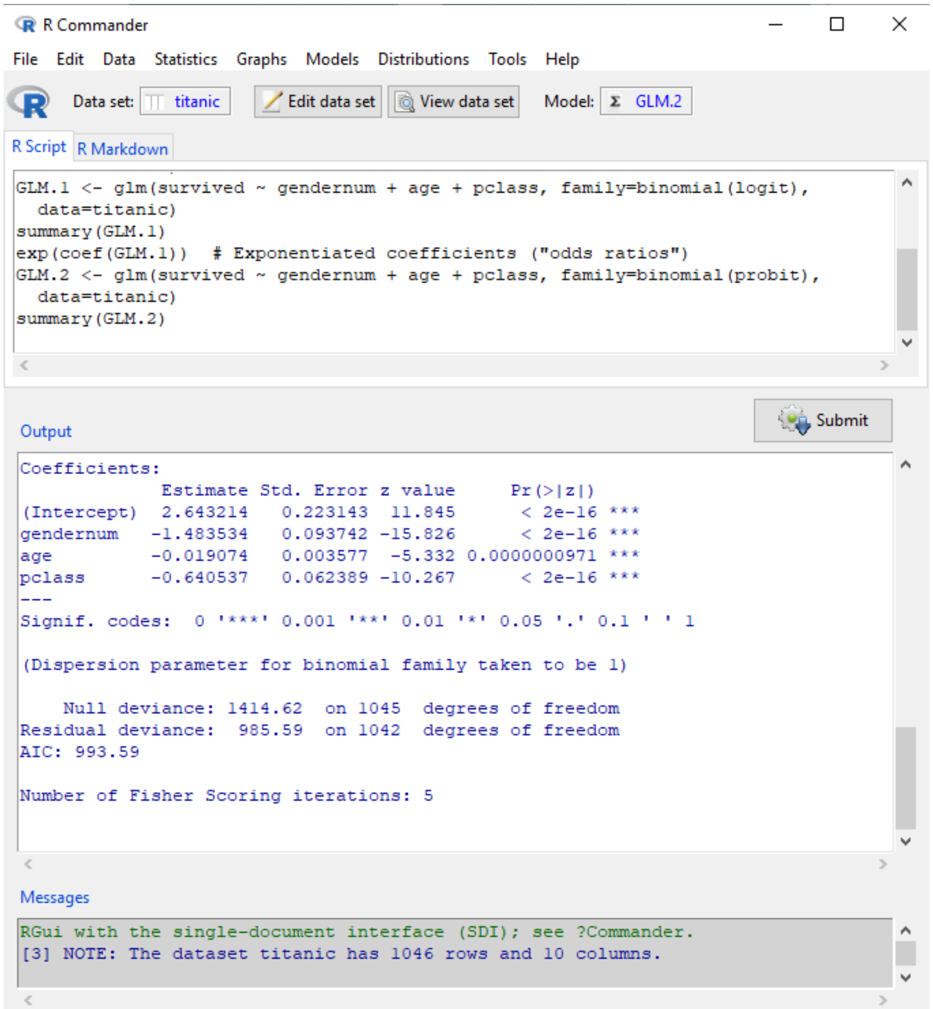
	J	K	L	M
3	Sensitivity Analysis			
4				
5			Gender	
6			Female	Male
7		96%	0	1
8	Age	0	97%	72%
9		5	96%	69%
10		10	96%	65%
11		15	95%	61%
12		20	94%	57%
13		25	93%	53%
14		30	92%	49%
15		35	91%	44%
16		40	89%	40%
17		45	87%	36%
18		50	85%	32%
19		55	83%	29%
20		60	81%	25%
21		65	78%	22%
22		70	75%	20%
23				

Probit Analysis

To perform a probit analysis on our data, where the Y variable is survived and the explanatory variables are gendernum, age, pclass:

1. Click on Statistics, Fit models, Generalized linear model
2. Double click on survived for the dependent variable
3. Double click on gendernum, age, pclass for the explanatory variables
4. Select Family as binomial
5. Select Link function as probit
6. Click OK



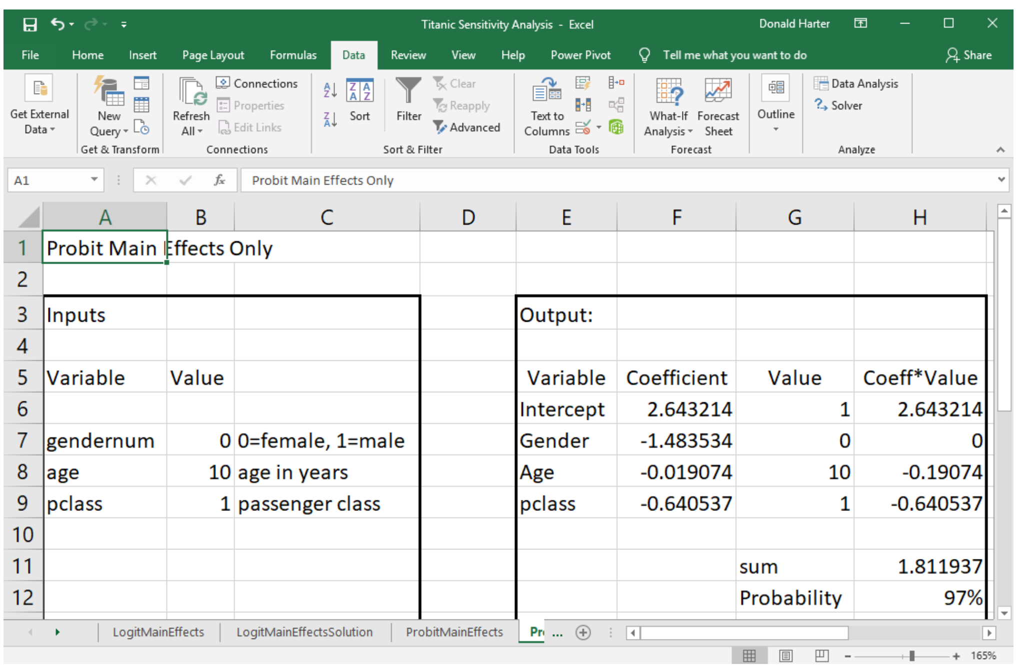


1. Are the coefficients positive or negative?
2. Are the coefficients statistically significant?
3. Is there a difference between logit and probit?

Calculating Probit Probabilities

You can use a spreadsheet to calculate logit probabilities for different combinations of the explanatory variable values. Let's calculate the probability of a 10-year-old girl in 1st class surviving the Titanic. Use the data from the previous page.

1. For this exercise, use the template called Titanic Sensitivity Analysis.
2. Click on the tab ProbitMainEffects
3. The left section allows you to enter the values for gendernum, age, and pclass.
    a. In cell B7, enter 0 for a girl
    b. In cell B8, enter 10 for 10 years old
    c. In cell B9, enter 1 for first class
4. Next, enter the coefficients for each variable in cells F6 through F9.
5. Enter a formula for the values in G7 through G9 pointing to the input values
    a. In G7, enter =B7
    b. In G8, enter =B8
    c. In G9, enter =B9
    d. For G6, enter 1
6. In cells H6 through H9, enter formulas multiplying the coefficients and values
7. The probit uses the normal function
8. First, calculate the sum. In cell H11, enter =sum(H6:H9)
9. Next, calculate the standard normal distribution of the sum. In cell H12, enter =norm.s.dist(H11,TRUE)
10. Change the input values from male to female; change the age; change the passenger class. What happens to the probability?



	A	B	C	D	E	F	G	H
1	Probit Main Effects Only							
2								
3	Inputs				Output:			
4								
5	Variable	Value			Variable	Coefficient	Value	Coeff*Value
6					Intercept	2.643214	1	2.643214
7	gendernum	0	0=female, 1=male		Gender	-1.483534	0	0
8	age	10	age in years		Age	-0.019074	10	-0.19074
9	pclass	1	passenger class		pclass	-0.640537	1	-0.640537
10								
11							sum	1.811937
12							Probability	97%

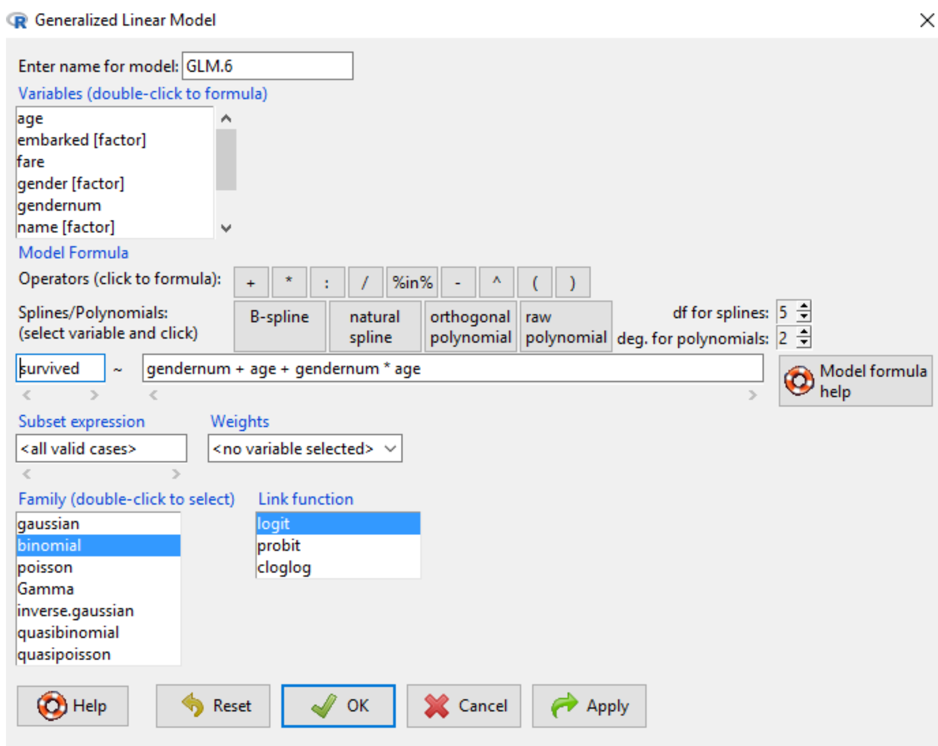
The Probit regression uses the normal distribution to calculate the probability:

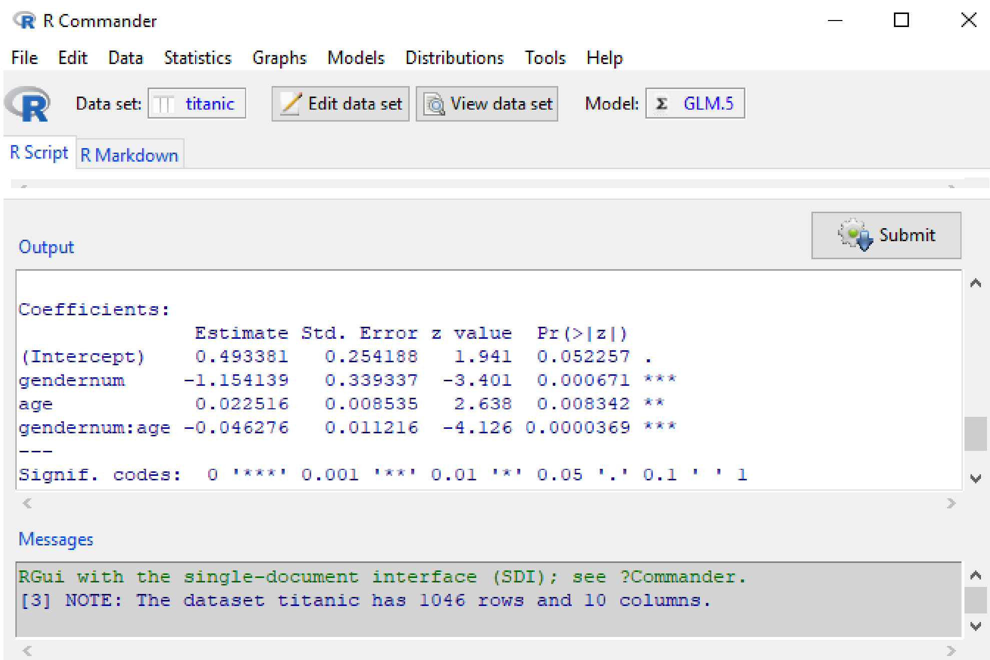
$P(Y=1) = \Phi(\Sigma\beta_i X_i)$ where Φ is the normal distribution

Logit with Moderating Effect

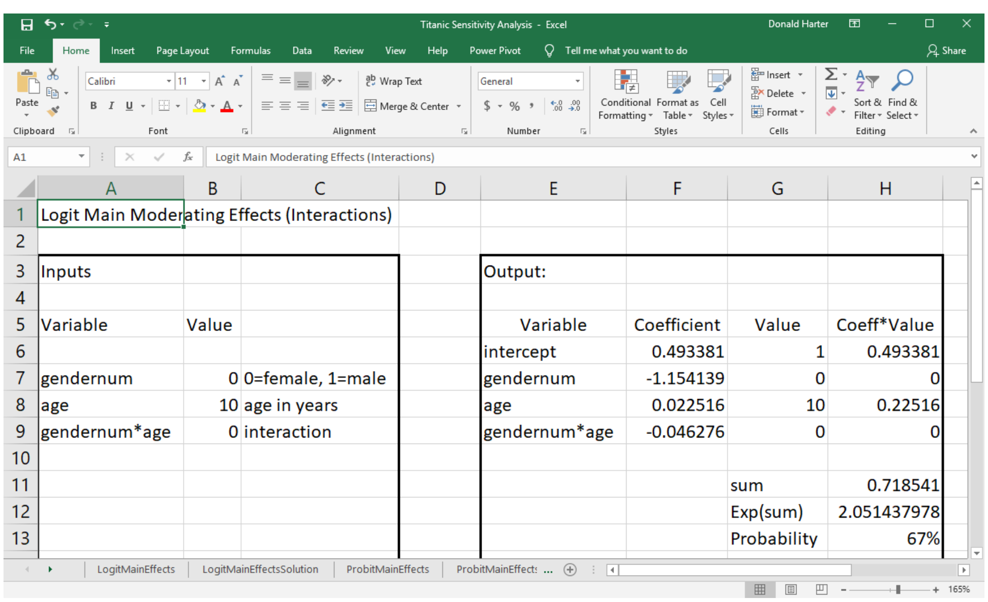
To perform a logit analysis on our data, where the Y variable is survived and the explanatory variables are gendernum, age, pclass, with a moderating effect of gendernum*age:

1. In R, click on Statistics, Fit models, Generalized linear model
2. Double click on survived for the dependent variable
3. Double click on gendernum and age for the explanatory variables; don’t include pclass
4. To add a moderating effect, double click on gendernum again, click on * to multiply, then double click on age
5. Select Family as binomial
6. Select Link function as logit
7. Click OK





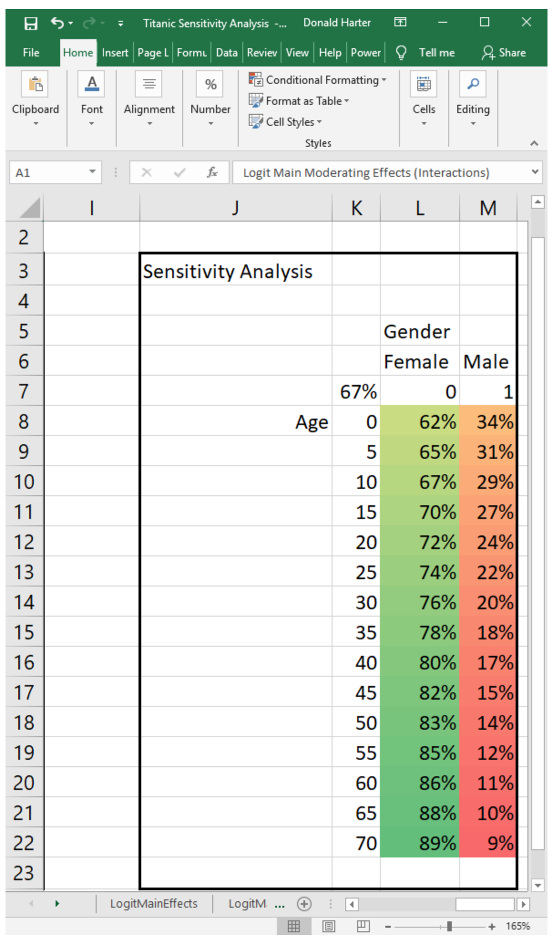
8. Use the spreadsheet tab LogitModeratingEffects to build to prediction model.
9. To predict survivability of a 10-year-old girl, enter 0 for gendernum and 10 for age in cells B7 and B8
10. In cell B9, to calculate gendernum*age, enter =B7*B8
11. Enter the coefficients in cells F6 through F9
12. Enter the formulas for the values in cells G6 through G9; recall that the value for the intercept is 1
13. Enter the formulas for Coefficient*Values in cell H6 through H9
14. In cell H11, enter the formula for the sum of H6:H9
15. In cell H12, enter the formula for the exponential of H11
16. In cell H13, enter the formula for the logistic function, =H12/(1+H12)



Logit Sensitivity Analysis with Moderating Effect

To perform a sensitivity analysis of gender versus age with a moderating effect, use the Sensitivity section of the same spreadsheet.

1. In general, for a sensitivity analysis, put one dimension across the top (in this case, gender), and one dimension down the side (in this case age).
2. In the cell in the corner, you must enter the formula for the probability calculation. This is already stored in cell H13. In cell K7, enter =H13.
3. Highlight K7 through M22
4. Click on the Data tab, then What If analysis, Data Table
5. For Row input cell, point to the cell with the value for gendernum, in this case B7
6. For Column input cell, point to the cell with the value for age, in this case B8
7. Click OK
8. What pattern do you see?
9. Notice that for each gender, the curve is monotonic (always increasing or always decreasing)
10. Are the male and female curves monotonic in the same direction?



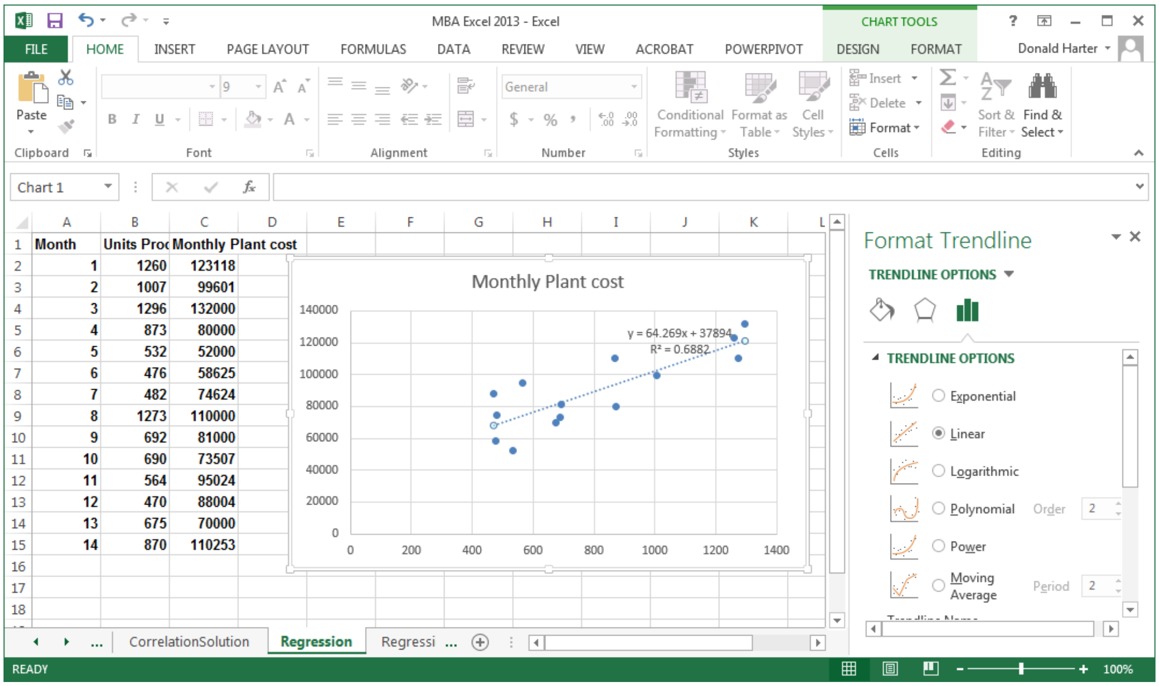
Sensitivity Analysis

		Gender	
		Female	Male
	67%	0	1
Age	0	62%	34%
	5	65%	31%
	10	67%	29%
	15	70%	27%
	20	72%	24%
	25	74%	22%
	30	76%	20%
	35	78%	18%
	40	80%	17%
	45	82%	15%
	50	83%	14%
	55	85%	12%
	60	86%	11%
	65	88%	10%
	70	89%	9%

Concept Review: A Different Perspective

Review – Regression

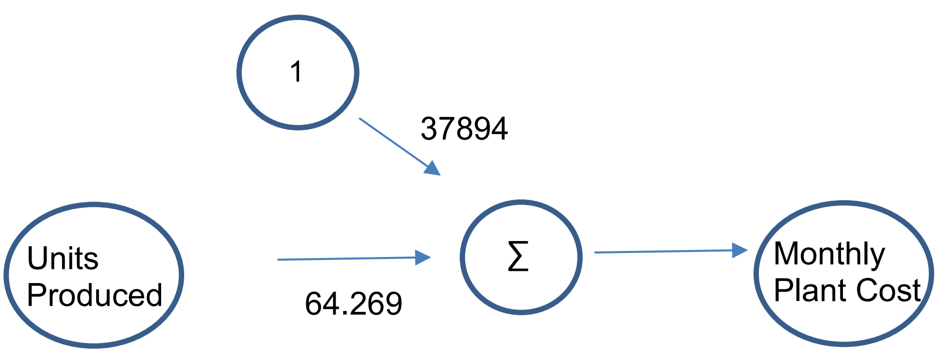
The purpose of linear regression was to identify the linear relationship between a dependent (response) variable and one or more independent (explanatory) variables. For the data below, we created the regression line that best fits the data.



For this data, the line that best fits the data is:

Monthly Plant Cost = 37894 + 64.269 * Units Produced

Another way to view this equation is:



Review – Logit & the Logistic Function

The purpose of logit regression was to identify the relationship between a binary dependent variable using a logistic function and explanatory variables. The form of the logistic function is:

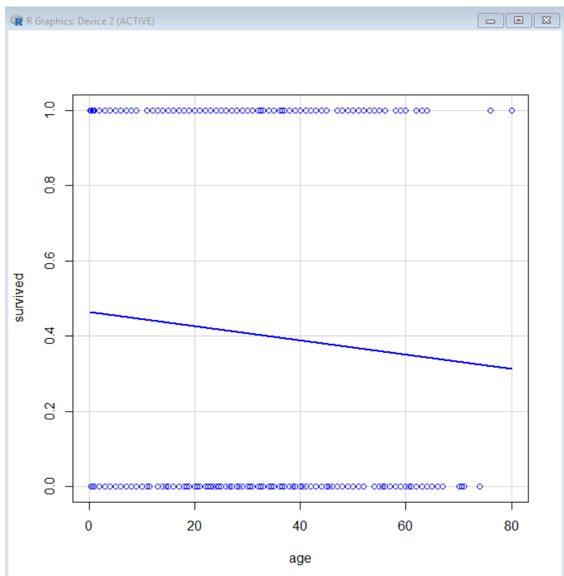
$$f(X) = \exp(\sum \beta_i X_i) / (1 + \exp(\sum \beta_i X_i))$$

Where:

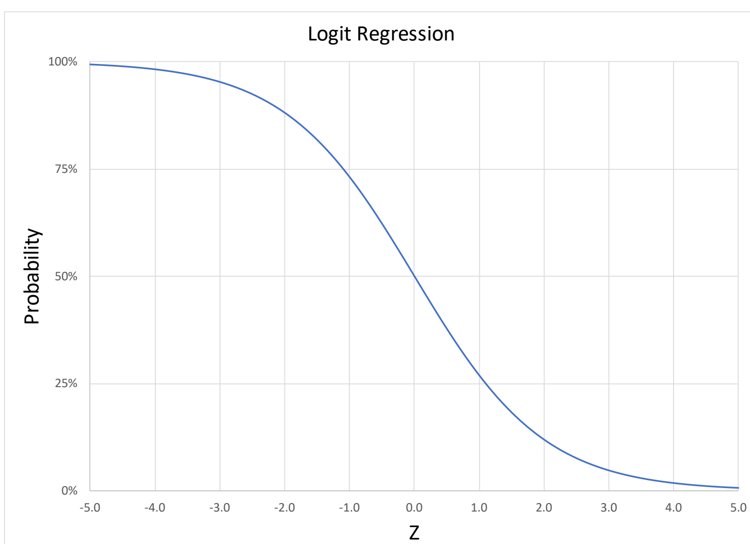
β_i = the coefficients in a logit regression

Xi = the variables in a logit regression

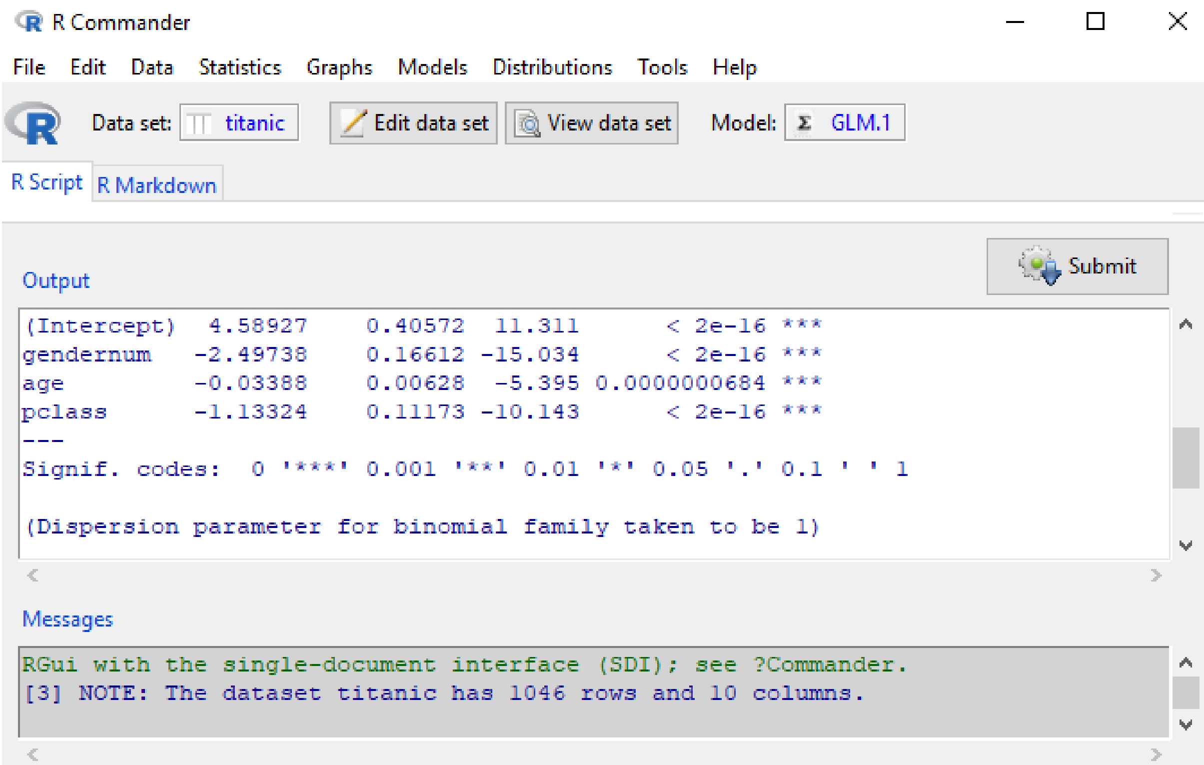
We are trying to find a function that fits the data below. A linear regression is not appropriate when the dependent variable is zeroes and ones.



A better solution looks like this:



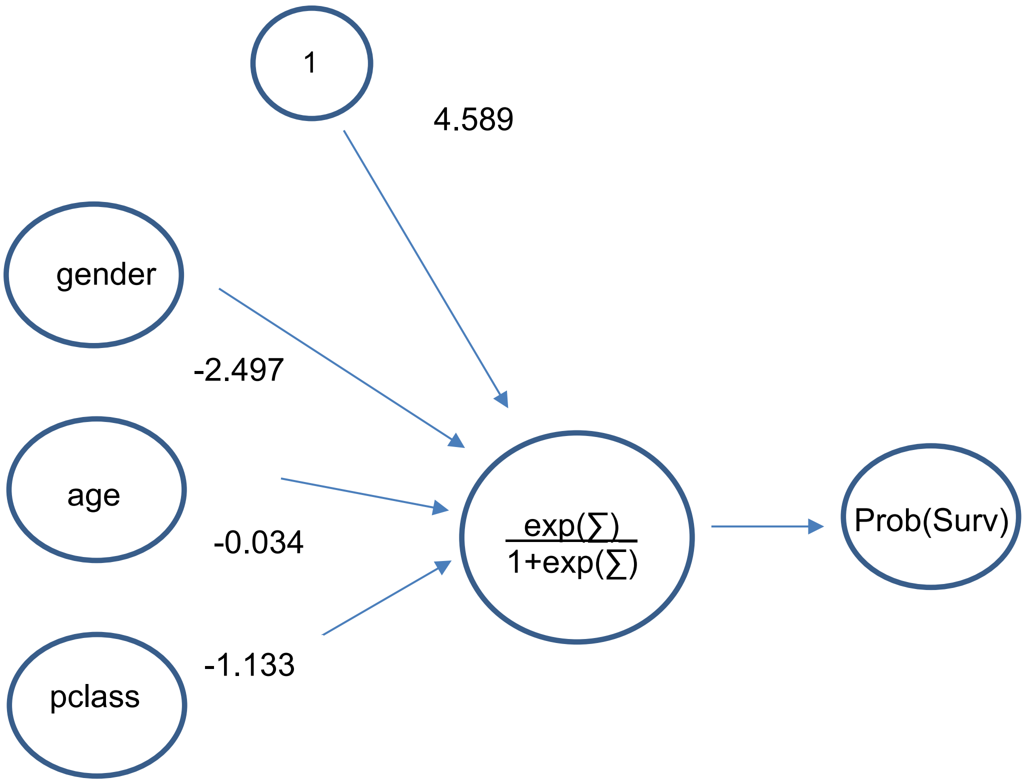
For the logit regression below, we created the logistic function that best fits the data.



The logistic function that predicts the probability of surviving is:

$$P(\text{survived}) = \frac{\exp(4.589 - 2.497*\text{gender} - 0.034*\text{age} - 1.133*\text{pclass})}{1 + \exp(4.589 - 2.497*\text{gender} - 0.034*\text{age} - 1.133*\text{pclass})}$$

Pictorially, this equation looks like:



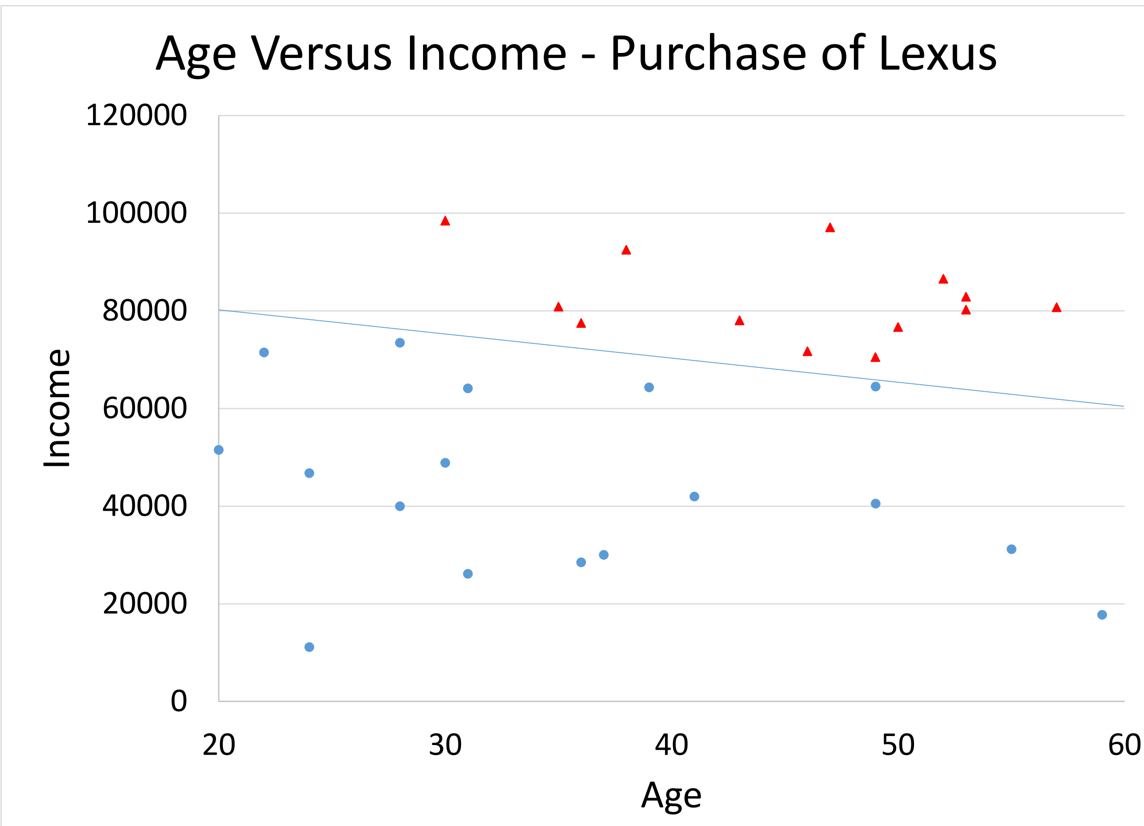
New Concept – Perceptrons

Reference:

Rosenblatt, Frank (1957), The Perceptron--a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory

Minsky M. L. and Papert S. A. 1969. *Perceptrons*. Cambridge, MA: MIT Press

A perceptron is a technique developed in the 1950s and 1960s to linearly classify data points into two groups. In the example below, red triangles identify customers who purchased a Lexus, blue dots are customers who did not purchase. The line is the linear perceptron classification line that separates the two groups.



But the perceptron could not classify all types of problems. Computer scientists recognized that it could classify data point formed by the classic “AND” condition. In the following example, the X marks the condition when a customer is both Rich and Young; O when this is not true.

	Rich	Not Rich
Young	X	O
Not Young	O	O

A line can be used to classify the X’s and O’s:

	Rich	Not Rich
Young	X	O
Not Young	O	O

Similarly, the classic “OR” condition can be classified.

	Rich	Not Rich
Young	X	X
Not Young	X	O

With the result:

	Rich	Not Rich
Young	X	X
Not Young	X	O

But researchers used the Exclusive OR condition to demonstrate the weakness of a perceptron. An Exclusive OR means Rich or Young, but not both.

	Rich	Not Rich
Young	O	X
Not Young	X	O

In this case, there is not straight line that separates the groups.

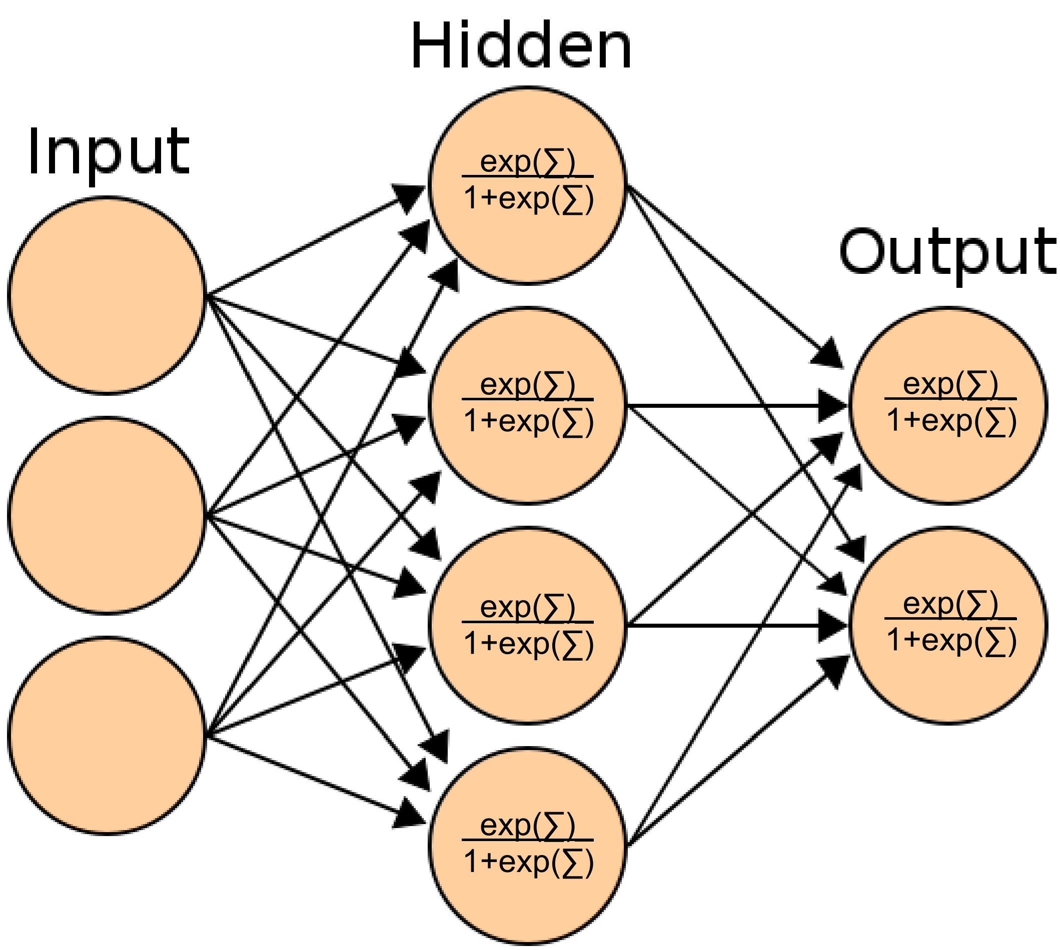
New Concept – Neural Networks

Reference:

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

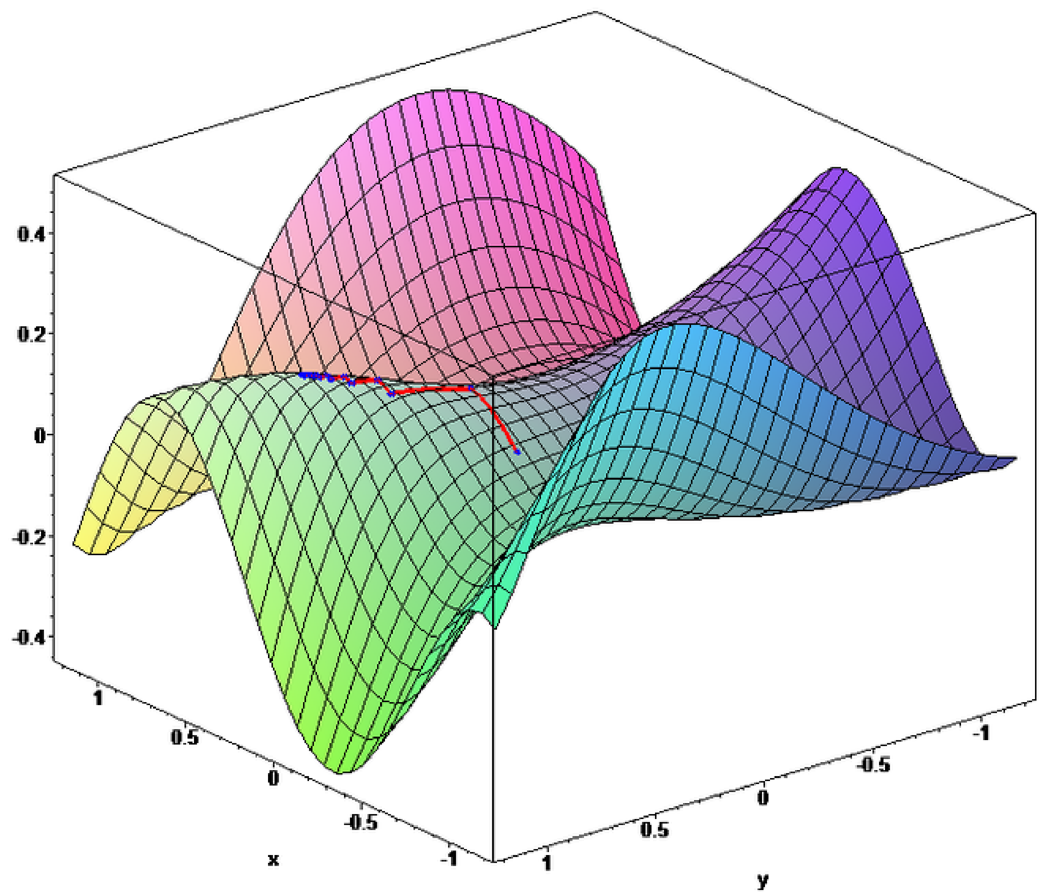
David Rumelhart and Jay McClelland recognized the limitation of a linear perceptron and proposed two innovations in 1986.

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)).

Installing NeuralNet

Follow these step to install neuralnet.

1. In R, type the command:

   install.packages("neuralnet", dependencies = TRUE)

2. If prompted for a CRAN mirror, select the location closest to you; use a USA location near you, then click OK
3. If prompted to create a personal library, click Yes
4. If prompted to add missing packages, 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

Neural network analysis

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

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

where:

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.

Neural Network Model

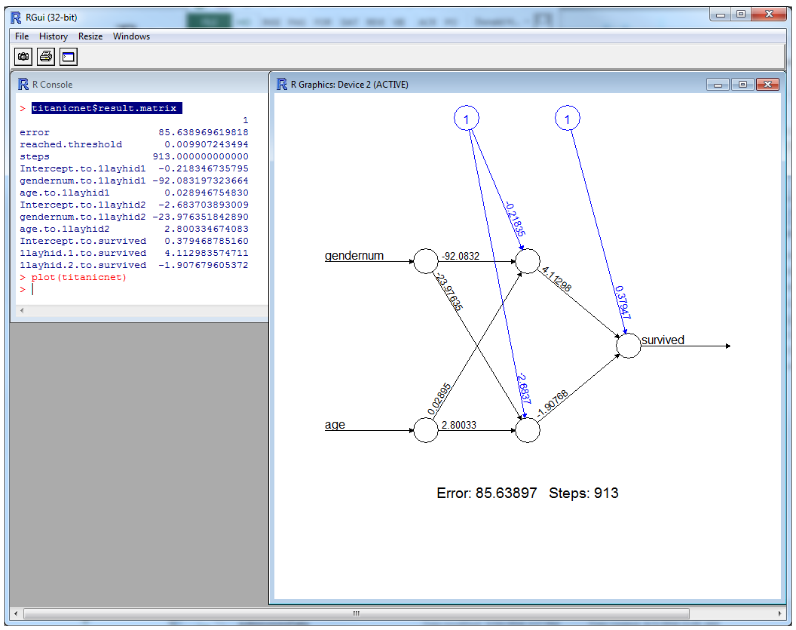
The result of the model can be displayed by plotting the model

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)

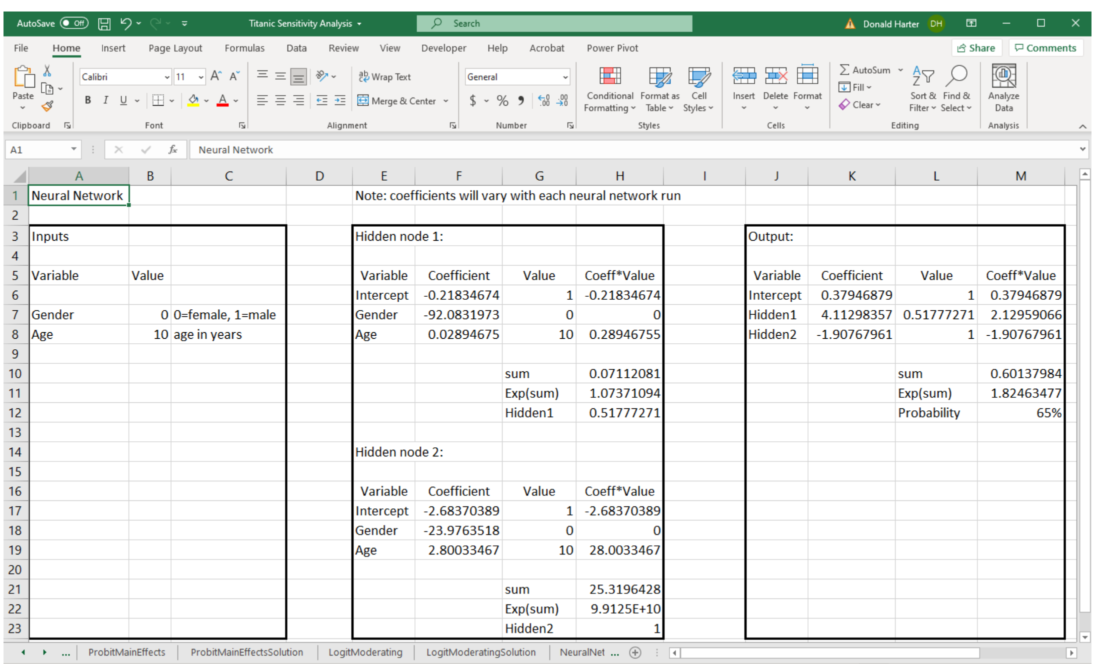


Neural Network Prediction

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. Click on the NeuralNetwork tab in the Titanic Sensitivity Analysis Excel workbook
2. The two input variables which we used in the neural network were Gender and Age.
    a. Type in a sample gender number into B7 (0 for female)
    b. Type a sample age into B8 (in this case, 10 for 10 years old)
3. For Hidden node 1, in cells F6:F8, enter the coefficients
4. In cells G6:G8, enter the value formulas (intercept value is 1)
5. In cells H6:H8, enter coefficient*value formula
6. In cell H10, enter sum formula
7. In cell H11, enter exponential of the sum
8. In cell H12, enter =H11/(1+H11) for the logistic function
9. Repeat the process for Hidden node 2
10. In the Output section, enter the coefficients in cells K6:K8
11. For the Values,
    a. Enter 1 for the intercept
    b. For Hidden 1, enter =H12 to point to the calculation for hidden node 1
    c. For Hidden 2, enter =H23 to point to the calculation for hidden node 2
12. In cells M6:M8, enter coefficient*value formula
13. In cell M10, enter sum formula
14. In cell M11, enter exponential of the sum
15. In cell M12, enter =M11/(1+M11)

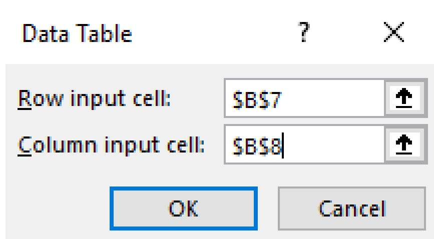


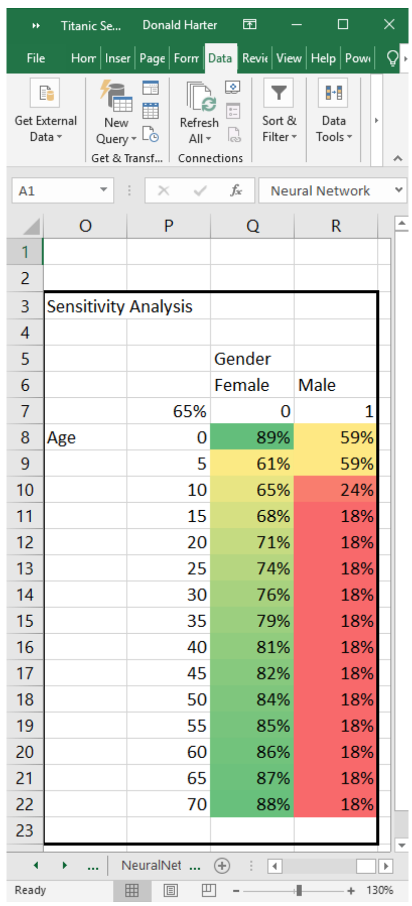
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Neural Network				Note: coefficients will vary with each neural network run								
3	Inputs				Hidden node 1:					Output:			
5	Variable	Value			Variable	Coefficient	Value	Coeff*Value		Variable	Coefficient	Value	Coeff*Value
6					Intercept	-0.21834674	1	-0.21834674		Intercept	0.37946879	1	0.37946879
7	Gender	0	0=female, 1=male		Gender	-92.0831973	0	0		Hidden1	4.11298357	0.51777271	2.12959066
8	Age	10	age in years		Age	0.02894675	10	0.28946755		Hidden2	-1.90767961	1	-1.90767961
10							sum	0.07112081				sum	0.60137984
11							Exp(sum)	1.07371094				Exp(sum)	1.82463477
12							Hidden1	0.51777271				Probability	65%
14					Hidden node 2:								
16					Variable	Coefficient	Value	Coeff*Value					
17					Intercept	-2.68370389	1	-2.68370389					
18					Gender	-23.9763518	0	0					
19					Age	2.80033467	10	28.0033467					
21							sum	25.3196428					
22							Exp(sum)	9.9125E+10					
23							Hidden2	1					

Sensitivity Analysis

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

1. Across the top of the sensitivity analysis is gender
2. Down the side of the sensitivity analysis is age
3. The formula is in the corner cell, P7. Point to the final formula output =M12
4. Click on the Data tab, What-If-Analysis, then Data Table
5. Since Gender varies across the row, enter B7 for Row input cell
6. Since Age varies down the column, enter B8 for column input cell
7. Click OK



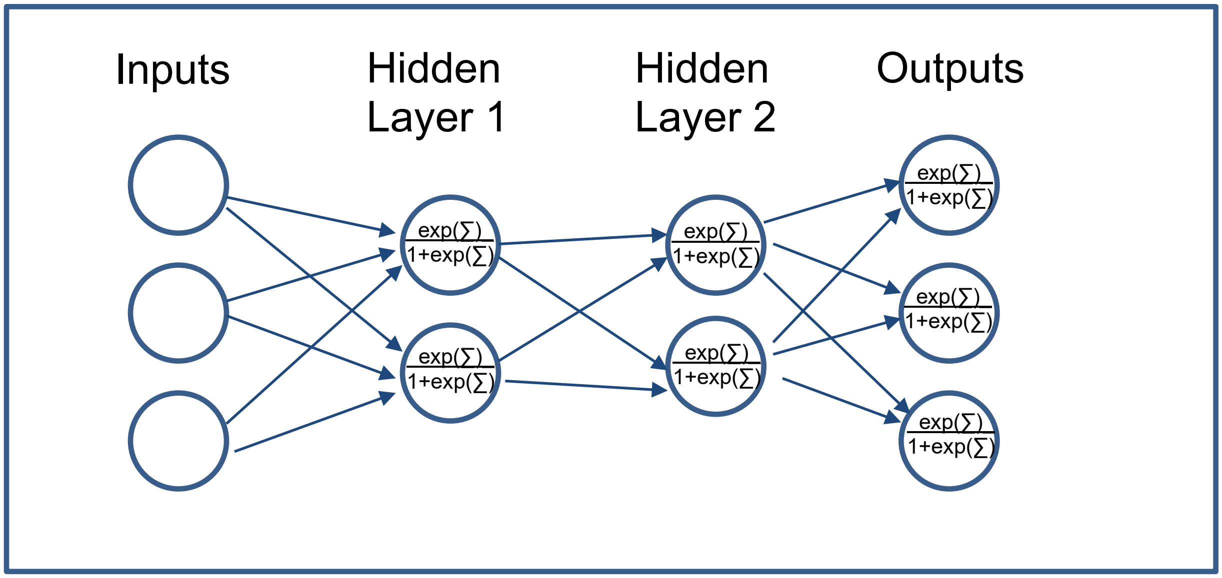


	O	P	Q	R
1				
2				
3	Sensitivity Analysis			
4				
5			Gender	
6			Female	Male
7		65%	0	1
8	Age	0	89%	59%
9		5	61%	59%
10		10	65%	24%
11		15	68%	18%
12		20	71%	18%
13		25	74%	18%
14		30	76%	18%
15		35	79%	18%
16		40	81%	18%
17		45	82%	18%
18		50	84%	18%
19		55	85%	18%
20		60	86%	18%
21		65	87%	18%
22		70	88%	18%
23				

8. How is this pattern different from the Logit?

Deep Neural Networks

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



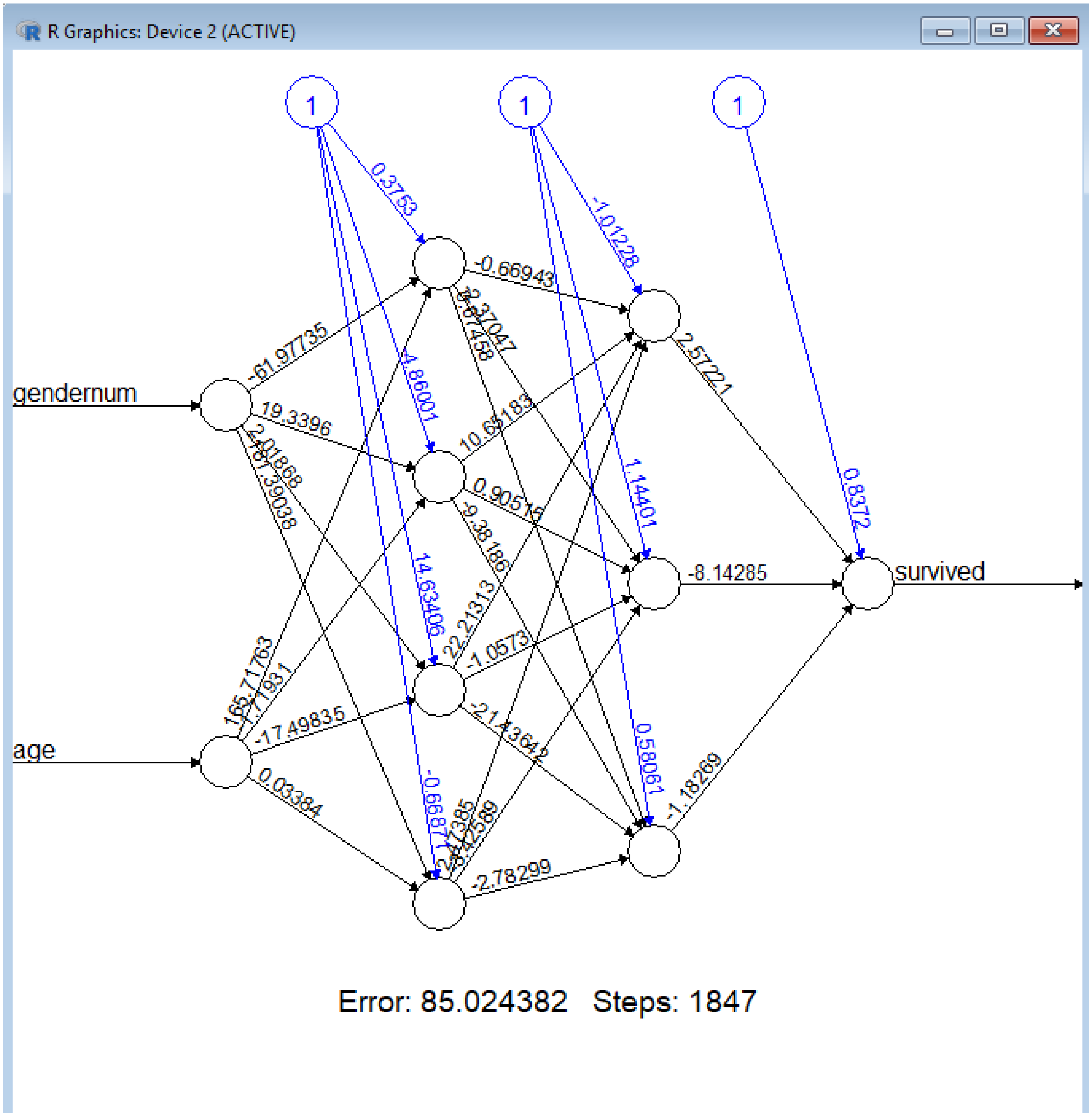
Using the Titanic data once again, let's use two inputs (gender, age), two hidden layers (with two nodes each), and one output. The command becomes

```
titanicnet <- neuralnet(survived ~ gendernum + age, titanic, hidden=c(4,3), 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)

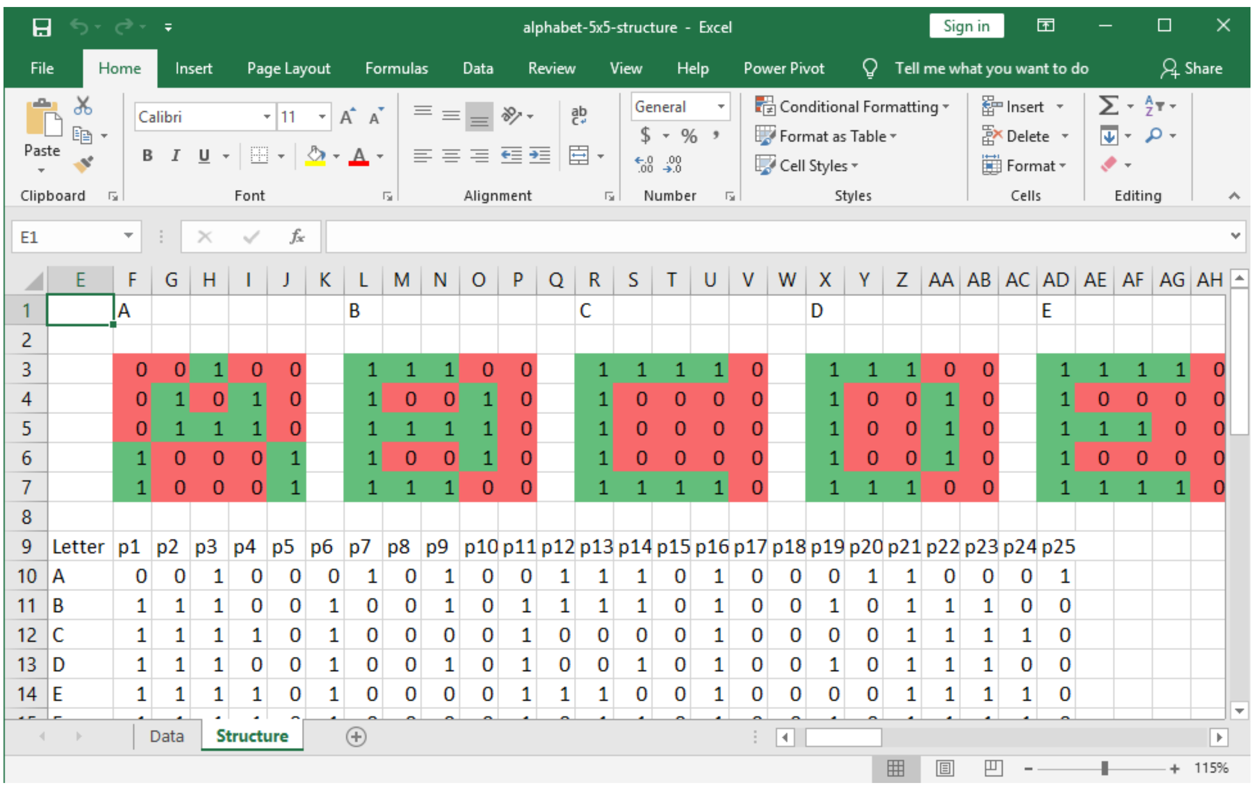


Neural Network with Linear Output

A neural network is not limited to outputs between zero and one. It can also predict any number. In the following example, the neural network will learn and predict letters of the alphabet.

The first step is to create the alphabet and convert the images into pixels. In the picture below are the letters of the alphabet with pixels either on (1) or off (0). In this example, each image is reduced to five pixels wide by five pixels deep, or 25 pixels (5x5). Each letter is then identified the sequence of pixels starting in the first column and row, moving across the first row, then the second row, etc.

The neural network will learn how the image, converted to pixels, relates to the letter of the alphabet. The prediction from the network will be numbers 1 through 26, representing the letter of the alphabet.



Letter	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19	p20	p21	p22	p23	p24	p25
A	0	0	1	0	0	0	1	0	1	0	0	1	1	1	0	1	0	0	0	1	1	0	0	0	1
B	1	1	1	0	0	1	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	0
C	1	1	1	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0
D	1	1	1	0	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	0	1	1	1	0	0
E	1	1	1	1	0	1	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	1	0

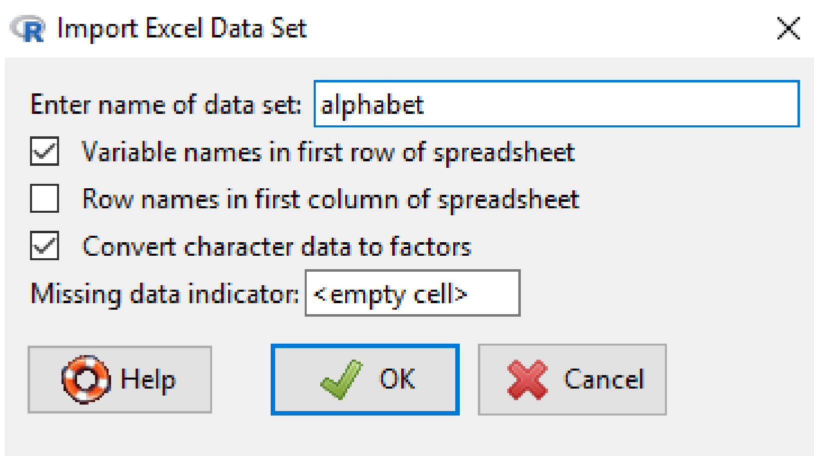
Loading Data

You can also load regular spreadsheets into R without using .csv (comma delimited) format.

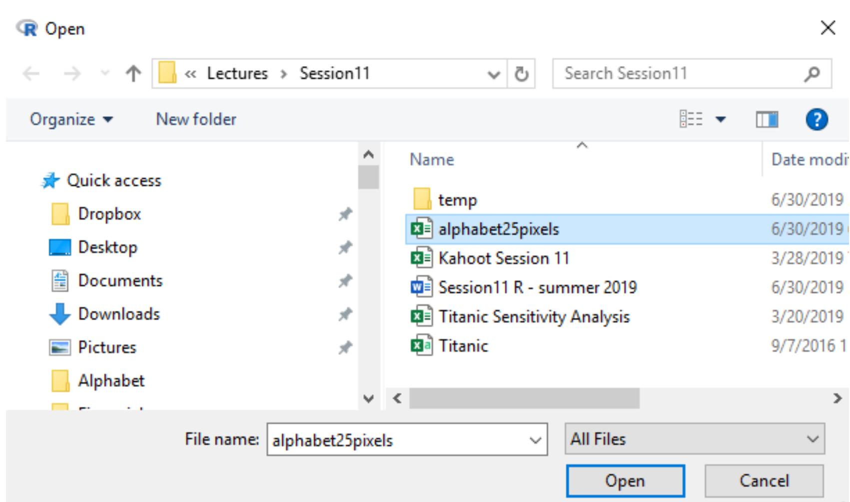
Download the spreadsheet alphabet25pixels from BlackBoard Session 10 or the G: drive.

To load a spreadsheet into R:

1. Click on Data at the top of the screen
2. Click on Import Data > From Excel file …
3. Enter the name that you would like to use for this data set; type in alphabet, then OK



4. Click on the files alphabet25pixels, then Open



Running the neural network with one output

To run the neural network, we will use a command like the Titanic example, with one key change. To have an output between zero and one, we used linear.output=FALSE. Now, to have an output from 1 to 26, we will use linear.output=TRUE.

The following command will use the num variable as the output (values 1 to 26) and the p# variables as the inputs representing the pixel for each letter. It uses two hidden layers, with 5 hidden nodes in the first layer, 4 hidden nodes in the second layer, designated by hidden=c(5,4). If you wanted five layers, with 8, 7, 6, 7, 5 hidden nodes in layers one through five, you would use the option hidden=c(8,7,6,7,5).

1. Copy the following command into the R window
2. Next, copy the following command into R

   ```
   alphabetnet <- neuralnet(LetterNum ~ p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 + p11 +
   p12 + p13 + p14 + p15 + p16 + p17 + p18 + p19 + p20 + p21 + p22 + p23 + p24 + p25, alphabet,
   hidden=c(5,4), lifesign="minimal", linear.output=TRUE, threshold=0.01)
   ```

3. To generate the plot, enter the command:

   ```
   plot(alphabetnet)
   ```

4. This is too many coefficients to enter in Excel, so we will have R generate the predictions.
5. To extract just the input data from our data set, we will create a subset with only the p# pixel values. Enter the command below. The alphabet[rows, columns] specifies that we want rows 1 through 26 and columns 3 through 27, which is the pixel data.

   ```
   inputdata <- alphabet[c(1:26),c(3:27)]
   ```

6. When you do not specify the rows or columns, then all are included. The following command has a blank for rows, so all rows are included. This is equivalent to the command above in step 5.

   ```
   inputdata <- alphabet[ ,c(3:27)]
   ```

7. Now calculate the predictions using the compute command. It takes the inputdata, which is our pixel inputs, and calculates the predictions using the neural network which it just learned.

   ```
   alphabetnet.results <- compute(alphabetnet, inputdata)
   ```

8. Finally, display the original LetterNum, which identifies the letter of the alphabet, and the prediction of the letter from the neural network based on the pixels presented.

   ```
   alphabetnet.results$net.result
   ```

R Console

```
 [1,]  0.9841714
 [2,]  2.0222618
 [3,]  3.0017081
 [4,]  3.9932210
 [5,]  4.9994298
 [6,]  5.9972971
 [7,]  7.0004030
 [8,]  7.9997754
 [9,]  8.9998814
[10,]  9.9997440
[11,] 10.9995090
[12,] 11.9993021
[13,] 12.9999310
[14,] 14.0006082
[15,] 15.0000442
[16,] 16.0010406
[17,] 17.0001927
[18,] 17.9998280
[19,] 18.9997189
[20,] 20.0000473
[21,] 20.9999058
[22,] 21.9996899
[23,] 22.9998533
[24,] 23.9998659
[25,] 25.0005704
[26,] 25.9993269
> 
```

Running the neural network with 26 outputs

1. Copy the following command into the R window

   alphabetnet <- neuralnet(A + B + C + D + E + F + G + H + I + J + K + L + M + N + O + P + Q + R + S + T + U + V + W + X + Y + Z
   ~ p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 + p11 + p12 + p13 + p14 + p15 + p16 + p17 + p18 + p19 + p20 + p21 + p22 + p23 + p24 + p25,
   alphabet, hidden=c(5,4), lifesign="minimal", linear.output=FALSE, threshold=0.01)

2. To generate the plot, enter the command:

   plot(alphabetnet)

3. This is too many coefficients to enter in Excel, so we will have R generate the predictions.
4. To extract just the input data from our data set, we will create a subset with only the p# pixel values. Enter the command below. The alphabet[rows, columns] specifies that we want rows 1 through 26 and columns 3 through 27, which is the pixel data.

   inputdata <- alphabet[c(1:26),c(3:27)]

5. When you do not specify the rows or columns, then all are included. The following command has a blank for rows, so all rows are included. This is equivalent to the command above in step 5.

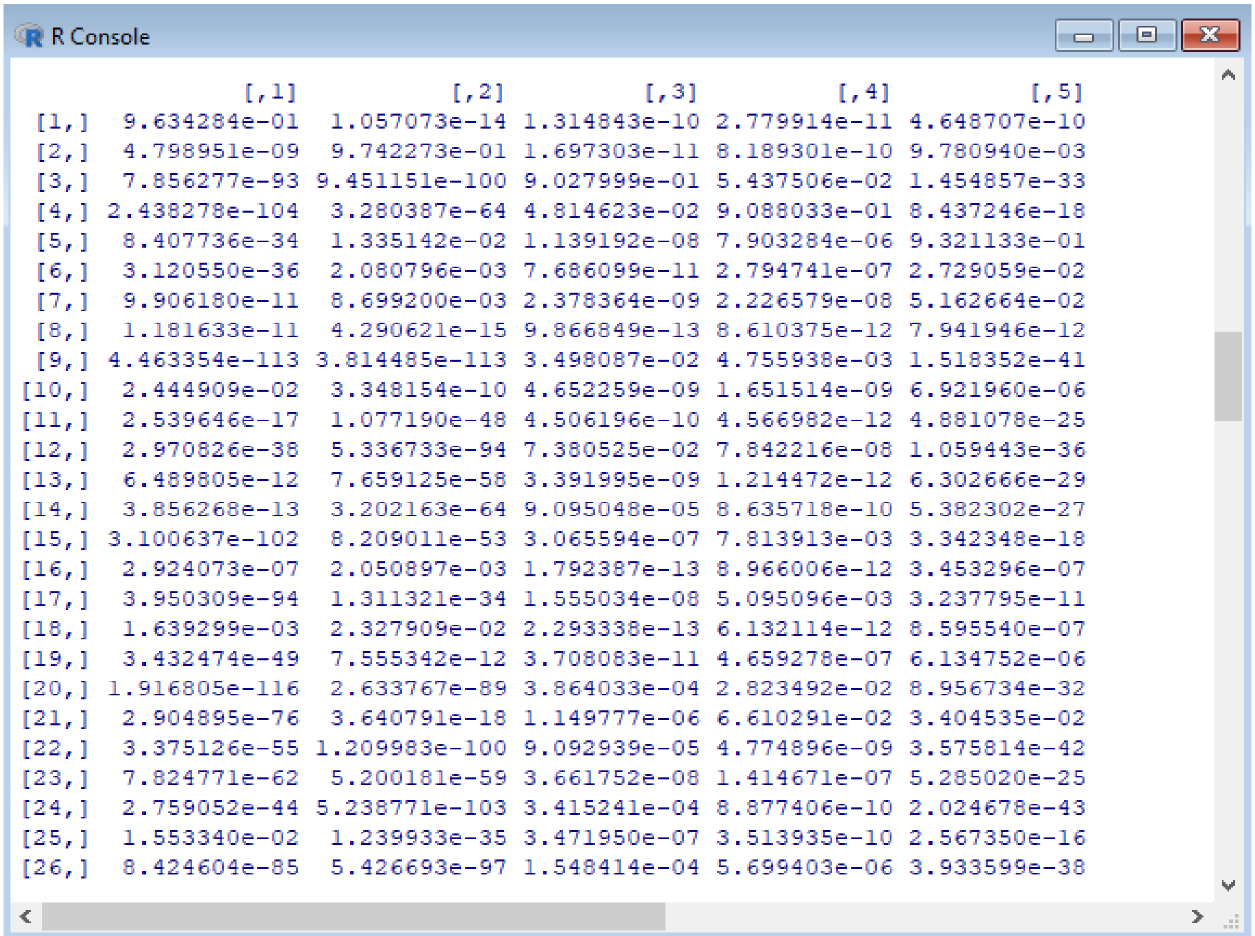
   inputdata <- alphabet[,c(3:27)]

6. Now calculate the predictions using the compute command. It takes the inputdata, which is our pixel inputs, and calculates the predictions using the neural network which it just learned.

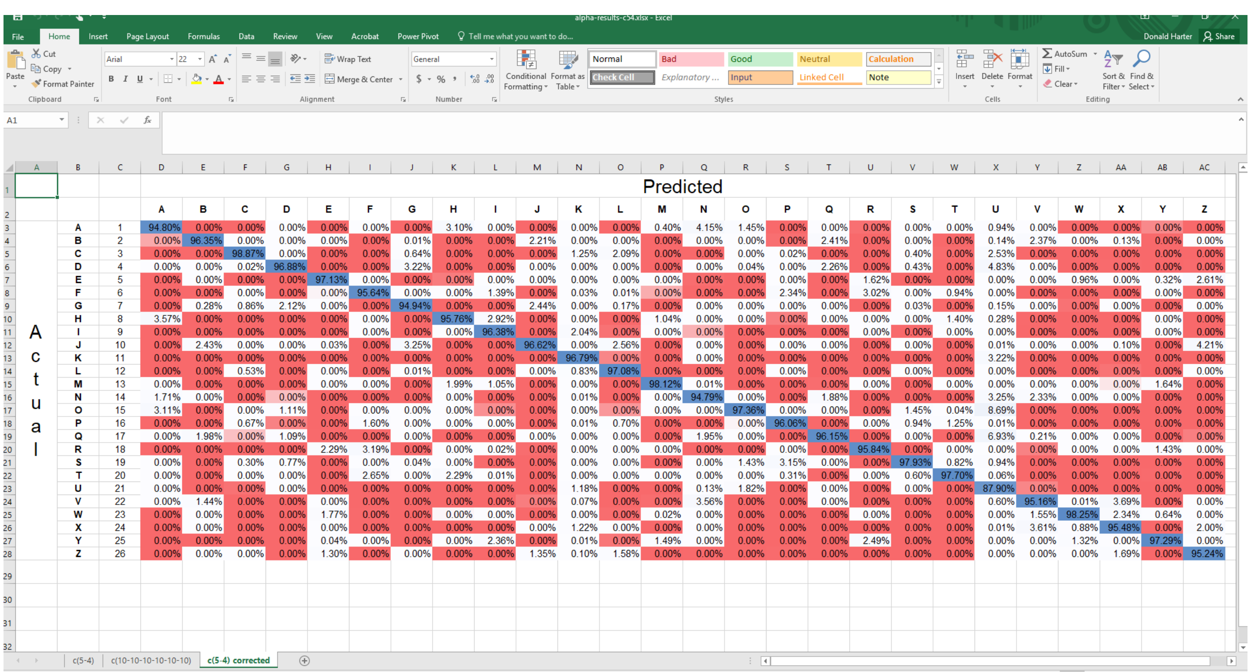
   alphabetnet.results <- compute(alphabetnet, inputdata)

7. Finally, display the original LetterNum, which identifies the letter of the alphabet, and the prediction of the letter from the neural network based on the pixels presented.

   alphabetnet.results$net.result




```
          [,1]          [,2]          [,3]         [,4]         [,5]
 [1,] 9.634284e-01  1.057073e-14 1.314843e-10 2.779914e-11 4.648707e-10
 [2,] 4.798951e-09  9.742273e-01 1.697303e-11 8.189301e-10 9.780940e-03
 [3,] 7.856277e-93 9.451151e-100 9.027999e-01 5.437506e-02 1.454857e-33
 [4,] 2.438278e-104 3.280387e-64 4.814623e-02 9.088033e-01 8.437246e-18
 [5,] 8.407736e-34  1.335142e-02 1.139192e-08 7.903284e-06 9.321133e-01
 [6,] 3.120550e-36  2.080796e-03 7.686099e-11 2.794741e-07 2.729059e-02
 [7,] 9.906180e-11  8.699200e-03 2.378364e-09 2.226579e-08 5.162664e-02
 [8,] 1.181633e-11  4.290621e-15 9.866849e-13 8.610375e-12 7.941946e-12
 [9,] 4.463354e-113 3.814485e-113 3.498087e-02 4.755938e-03 1.518352e-41
[10,] 2.444909e-02  3.348154e-10 4.652259e-09 1.651514e-09 6.921960e-06
[11,] 2.539646e-17  1.077190e-48 4.506196e-10 4.566982e-12 4.881078e-25
[12,] 2.970826e-38  5.336733e-94 7.380525e-02 7.842216e-08 1.059443e-36
[13,] 6.489805e-12  7.659125e-58 3.391995e-09 1.214472e-12 6.302666e-29
[14,] 3.856268e-13  3.202163e-64 9.095048e-05 8.635718e-10 5.382302e-27
[15,] 3.100637e-102 8.209011e-53 3.065594e-07 7.813913e-03 3.342348e-18
[16,] 2.924073e-07  2.050897e-03 1.792387e-13 8.966006e-12 3.453296e-07
[17,] 3.950309e-94  1.311321e-34 1.555034e-08 5.095096e-03 3.237795e-11
[18,] 1.639299e-03  2.327909e-02 2.293338e-13 6.132114e-12 8.595540e-07
[19,] 3.432474e-49  7.555342e-12 3.708083e-11 4.659278e-07 6.134752e-06
[20,] 1.916805e-116 2.633767e-89 3.864033e-04 2.823492e-02 8.956734e-32
[21,] 2.904895e-76  3.640791e-18 1.149777e-06 6.610291e-02 3.404535e-02
[22,] 3.375126e-55 1.209983e-100 9.092939e-05 4.774896e-09 3.575814e-42
[23,] 7.824771e-62  5.200181e-59 3.661752e-08 1.414671e-07 5.285020e-25
[24,] 2.759052e-44 5.238771e-103 3.415241e-04 8.877406e-10 2.024678e-43
[25,] 1.553340e-02  1.239933e-35 3.471950e-07 3.513935e-10 2.567350e-16
[26,] 8.424604e-85  5.426693e-97 1.548414e-04 5.699403e-06 3.933599e-38
```



Predicted

Actual		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	1	94.80%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	3.10%	0.00%	0.00%	0.00%	0.00%	0.40%	4.15%	1.45%	0.00%	0.00%	0.00%	0.00%	0.00%	0.94%	0.00%	0.00%	0.00%	0.00%	0.00%
B	2	0.00%	96.35%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	2.21%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.41%	0.00%	0.00%	0.00%	0.14%	2.37%	0.00%	0.13%	0.00%	0.00%
C	3	0.00%	0.00%	98.87%	0.00%	0.00%	0.00%	0.64%	0.00%	0.00%	0.00%	1.25%	2.09%	0.00%	0.00%	0.00%	0.02%	0.00%	0.00%	0.40%	0.00%	2.53%	0.00%	0.00%	0.00%	0.00%	0.00%
D	4	0.00%	0.00%	0.02%	96.88%	0.00%	0.00%	3.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.04%	0.00%	2.26%	0.00%	0.43%	0.00%	4.83%	0.00%	0.00%	0.00%	0.00%	0.00%
E	5	0.00%	0.00%	0.00%	0.00%	97.13%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.62%	0.00%	0.00%	0.00%	0.00%	0.96%	0.00%	0.32%	2.61%
F	6	0.00%	0.00%	0.00%	0.00%	0.00%	95.64%	0.00%	0.00%	1.39%	0.00%	0.03%	0.01%	0.00%	0.00%	0.00%	2.34%	0.00%	3.02%	0.00%	0.94%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
G	7	0.00%	0.28%	0.86%	2.12%	0.00%	0.00%	94.94%	0.00%	0.00%	2.44%	0.00%	0.17%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.03%	0.00%	0.15%	0.00%	0.00%	0.00%	0.00%	0.00%
H	8	3.57%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	95.76%	2.92%	0.00%	0.00%	0.00%	1.04%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.40%	0.28%	0.00%	0.00%	0.00%	0.00%	0.00%
I	9	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	96.38%	0.00%	2.04%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
J	10	0.00%	2.43%	0.00%	0.00%	0.03%	0.00%	3.25%	0.00%	0.00%	96.62%	0.00%	2.56%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	0.10%	0.00%	4.21%
K	11	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	96.79%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	3.22%	0.00%	0.00%	0.00%	0.00%	0.00%
L	12	0.00%	0.00%	0.53%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	0.00%	0.83%	97.08%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
M	13	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.99%	1.05%	0.00%	0.00%	0.00%	98.12%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.64%	0.00%
N	14	1.71%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	94.79%	0.00%	0.00%	1.88%	0.00%	0.00%	0.00%	3.25%	2.33%	0.00%	0.00%	0.00%	0.00%
O	15	3.11%	0.00%	0.00%	1.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	97.36%	0.00%	0.00%	0.00%	1.45%	0.04%	8.69%	0.00%	0.00%	0.00%	0.00%	0.00%
P	16	0.00%	0.00%	0.67%	0.00%	0.00%	1.60%	0.00%	0.00%	0.00%	0.00%	0.01%	0.70%	0.00%	0.00%	0.00%	96.06%	0.00%	0.00%	0.94%	1.25%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%
Q	17	0.00%	1.98%	0.00%	1.09%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.95%	0.00%	0.00%	96.15%	0.00%	0.00%	0.00%	6.93%	0.21%	0.00%	0.00%	0.00%	0.00%
R	18	0.00%	0.00%	0.00%	0.00%	2.29%	3.19%	0.00%	0.00%	0.02%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	95.84%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.43%	0.00%
S	19	0.00%	0.00%	0.30%	0.77%	0.00%	0.00%	0.04%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.43%	3.15%	0.00%	0.00%	97.93%	0.82%	0.94%	0.00%	0.00%	0.00%	0.00%	0.00%
T	20	0.00%	0.00%	0.00%	0.00%	0.00%	2.65%	0.00%	2.29%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.31%	0.00%	0.00%	0.60%	97.70%	0.06%	0.00%	0.00%	0.00%	0.00%	0.00%
U	21	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.18%	0.00%	0.00%	0.13%	1.82%	0.00%	0.00%	0.00%	0.00%	0.00%	87.90%	0.00%	0.00%	0.00%	0.00%	0.00%
V	22	0.00%	1.44%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.07%	0.00%	0.00%	3.56%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.60%	95.16%	0.01%	3.69%	0.00%	0.00%
W	23	0.00%	0.00%	0.00%	0.00%	1.77%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.02%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.55%	98.25%	2.34%	0.64%	0.00%
X	24	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	3.61%	0.88%	95.48%	0.00%	2.00%
Y	25	0.00%	0.00%	0.00%	0.00%	0.04%	0.00%	0.00%	0.00%	2.36%	0.00%	0.01%	0.00%	1.49%	0.00%	0.00%	0.00%	0.00%	2.49%	0.00%	0.00%	0.00%	0.00%	1.32%	0.00%	97.29%	0.00%
Z	26	0.00%	0.00%	0.00%	0.00%	1.30%	0.00%	0.00%	0.00%	0.00%	1.35%	0.10%	1.58%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.69%	0.00%	95.24%