Computer Algorithms and Modelling

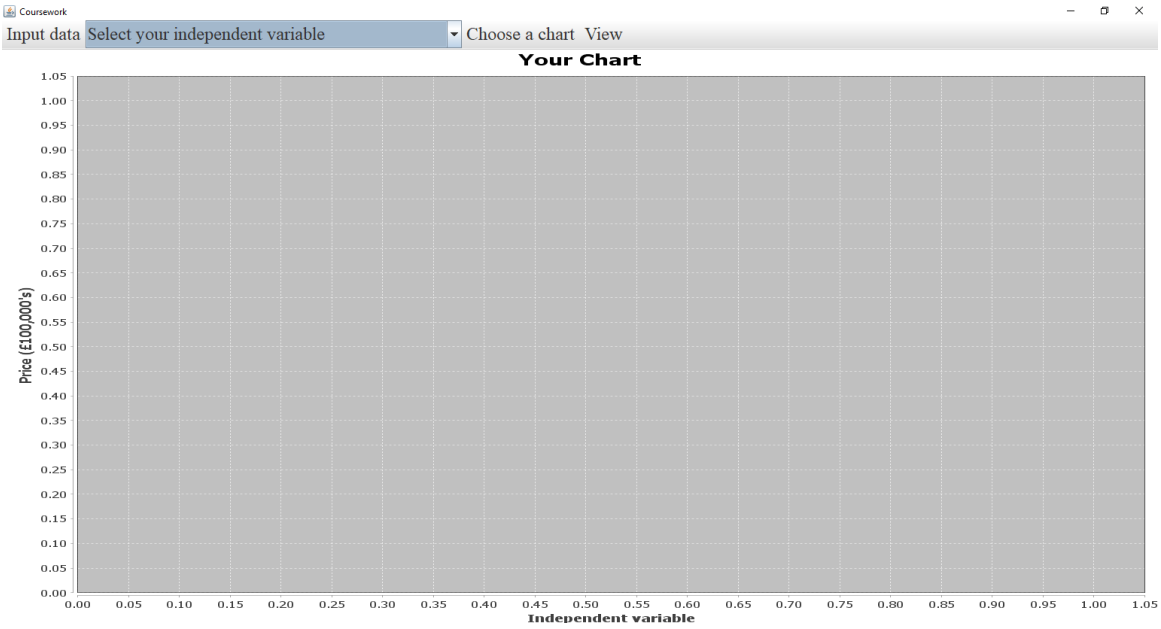
Group member Name	ID
Roshaan Nazir	000834566
Maks Smirnov	000979299
Thomas Stoyles	000990057
Michael Mitchell	000978660

Contents

Introduction

Using Java, we have made a multifunctional program relating to the price of a house given different factors that may affect this price. Within the program itself, it is possible to view a scatter chart showing the relationship of a particular variable (e.g. age of a property) with the price associated with the property. It is also possible to draw a regression line and predict the price of a property by entering a value for a given variable. We have ensured that our program is fully functional, and meets the specification for both the basic, and the extended solutions. This report details the process that we went through to complete this program.

Design



Shown above, is the Graphical User Interface (GUI from hereon) that is displayed when the user runs the program. Consisting of a single combo box that can be clicked to select the list of independent variables, the program also features three menus to provide functionalities such as reading in a text file or plotting different kinds of scatter charts. The GUI has been designed with error detection and prevention in mind – to avoid errors we have prevented access to certain parts of the program (such as viewing data tables) until the necessary information is provided. This ensures that the program runs smoothly and increases ease of use, as the program informs the user what is causing the problem. This is achieved through 'JOptionPane' – the user is told which information is missing, and so can add the necessary information before proceeding. We have also regularly added message boxes informing the user that a requested action has taken place.

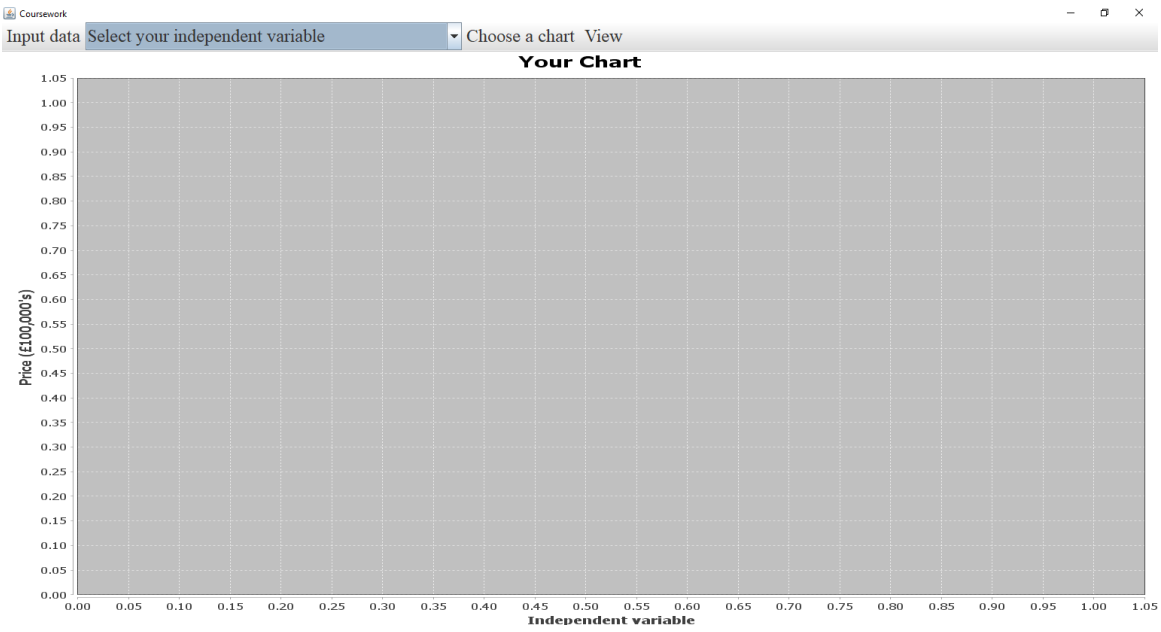
With respect to the use of libraries, we opted to use the 'JFreeChart' library that we downloaded online (link available in the references section). We considered using Java swing only, however given the time allocated to us for the assignment this did not seem like a feasible idea. The decision to choose JFreeChart over other libraries was made due to JFreeChart's functionalities – it has the ability to draw the charts that we needed (XY scatter and line) all while being fully customizable. The chart produced can easily be zoomed into to identify the exact value on any axis. While this library has the functionality to perform regression analysis, we chose to create our own class to perform regression analysis as well as calculating the correlation coefficients in an attempt to demonstrate our understanding and knowledge of least squares regression.

Assumptions

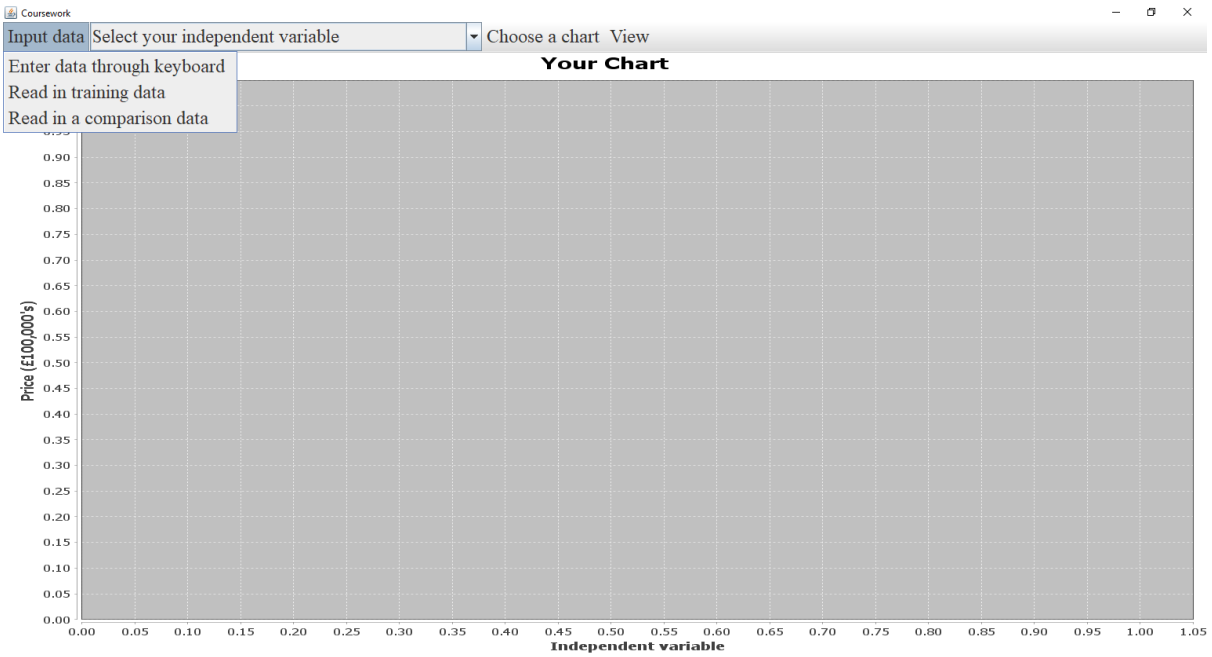
1. With respect to the data sets given to us (training and comparison) we have assumed that there are no anomalous results, therefore the linear regression line will take into consideration all values in the data sets.
2. We have assumed in our regression analysis of the data, that the only factors that affect the price of a property have been given to us, in the form of the independent variables. When forecasting the price of a house, our program therefore only gives the option of predicting the price of a house given one of the seven variables. It is not to be used to predict the price of a house where other factors influence the price (eg the location of a property) as it will not provide an accurate estimate.
3. As our program has an added functionality that allows the user to add additional comparison sets (not including Town A, B and C), we have assumed that the data that will be added will have the same format as the previous comparison sets (variables all separated by a 'tab'). In addition to this, we have assumed that no nonsensical values will have been added to this text file, such as negative numbers or strings/symbols. We have also assumed that there will be no additional variables in this comparison set and that the file containing the data is in the same directory as the comparison set and training data set.

User Instructions/Documentation

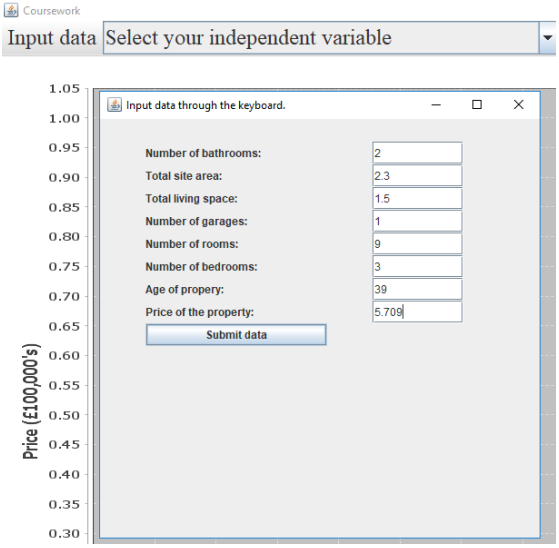
When the program is run, the user is presented with the following screen.



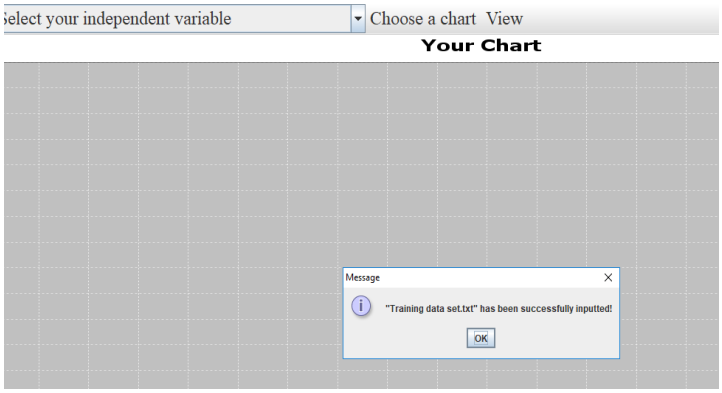
The basic GUI consists of three menus (each containing different functionalities that can be used), a combo box and a blank scatter chart. The program is designed to support the input of data from multiple sources. These are listed when clicking on the 'Input data' menu, as shown below.



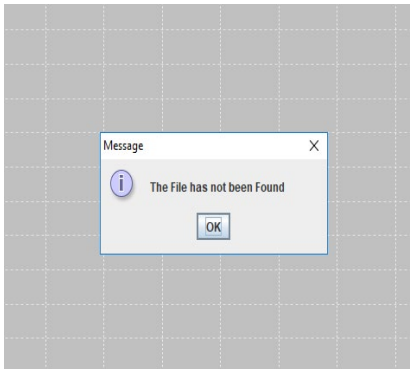
Each option allows for a unique way of adding data to the program. Clicking on the 'Enter data through keyboard' option presents the user with a form that they enter data into which is then stored within an arraylist in the program, as shown below



The program is trained using the 'training data set' which is stored in a text file within the program directory. To add this data to the program, the user must click on the 'Read in training data' option found within the 'Input data' menu. The user is then presented with a confirmation that this data has been added, or a message to say that the file was not found.

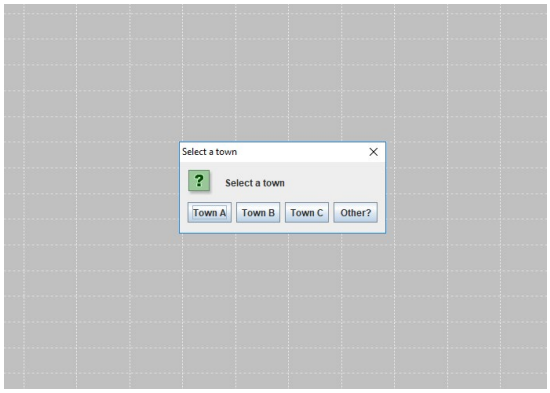


Data successfully added shown above.

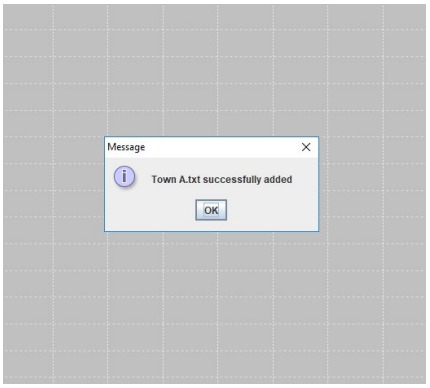


Data not found shown above

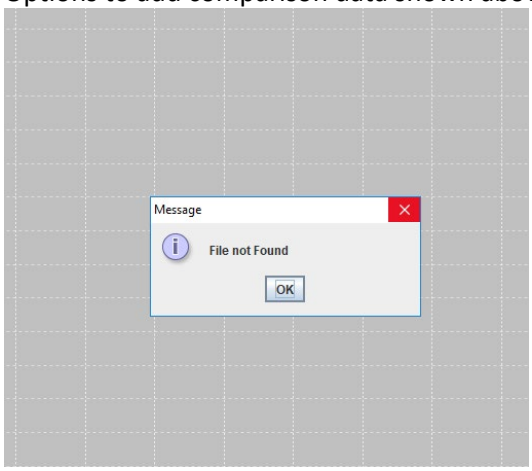
The user can also add in comparison data, either from towns A, B or C or alternatively their own data given it is in the same format and within the same directory. This is shown below.



Options to add comparison data shown above.

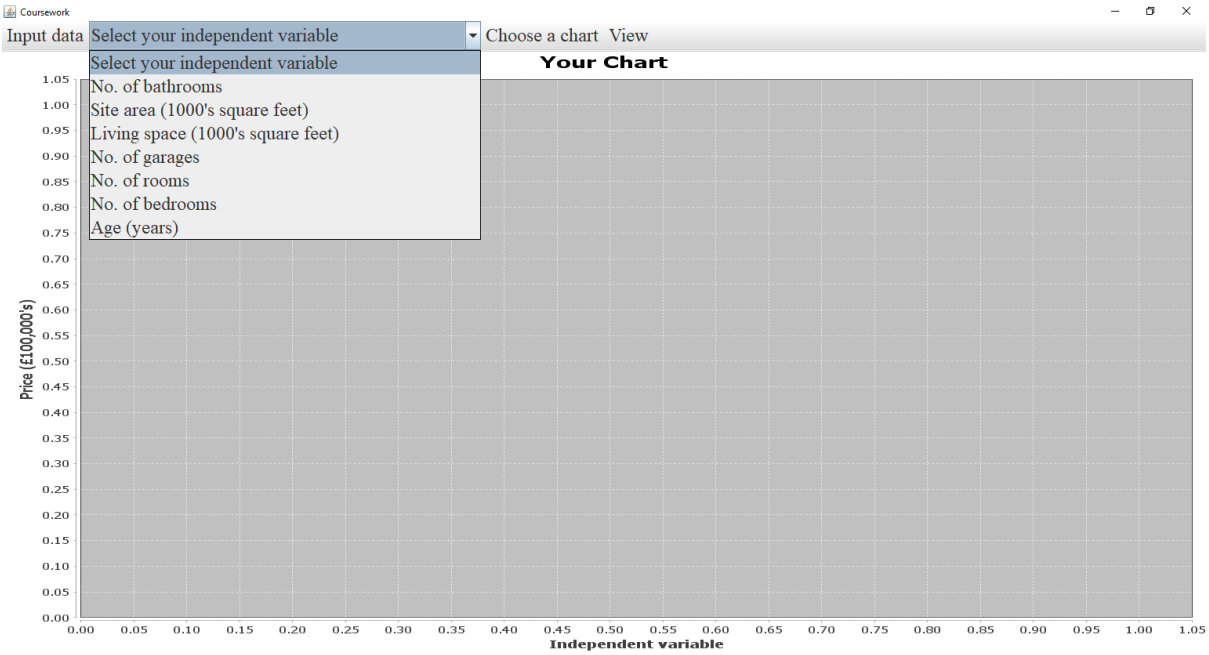


Above message displayed when data is added

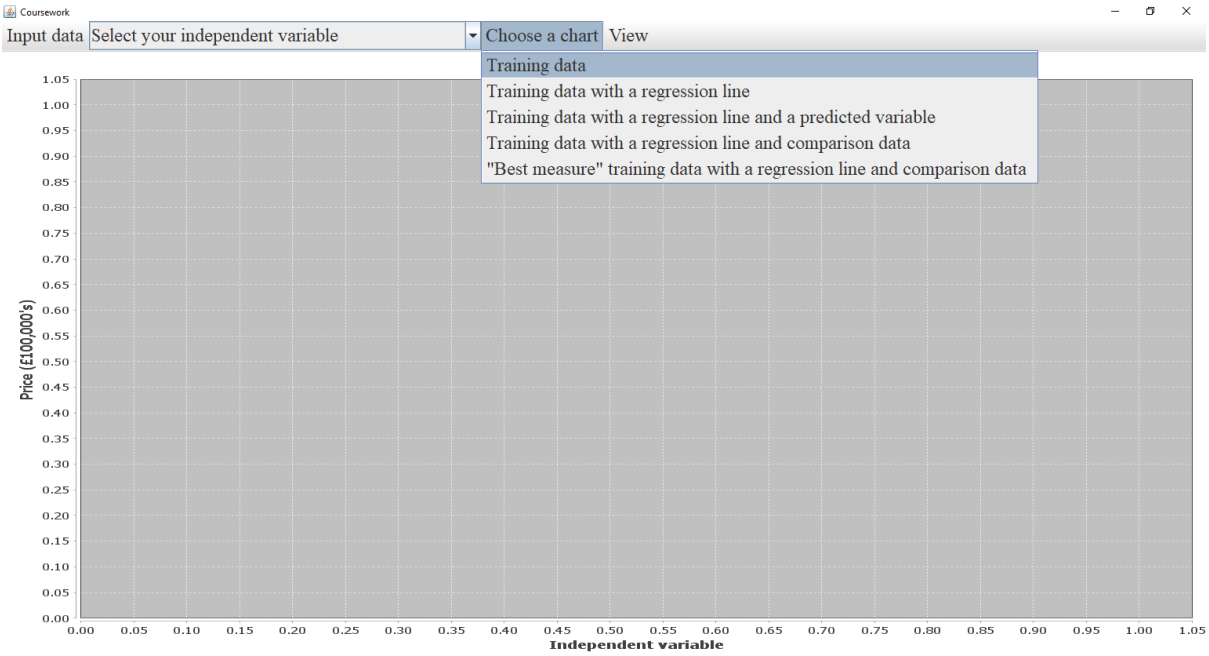


Message displayed when the file is not found

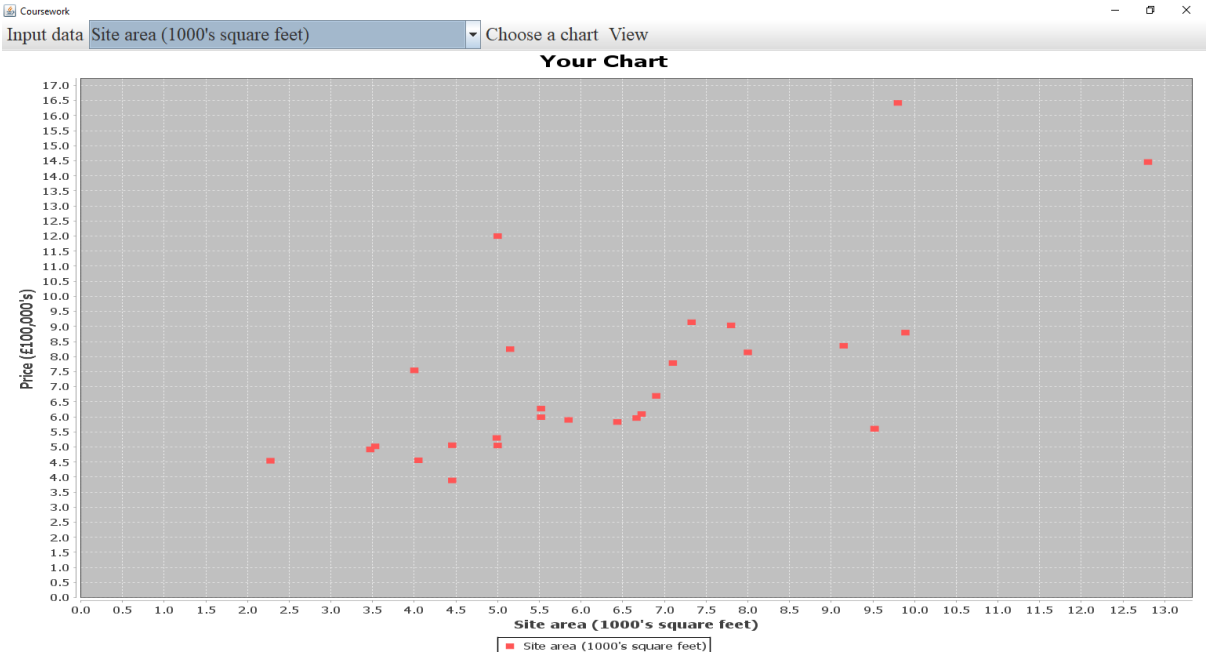
The combo box is populated with the names of the independent variables used to plot the charts.



To actually plot a chart the user must click on the 'choose a chart' menu and select one of the options



Since the program requires data and an independent variable to plot a chart, these must be added before selecting the type of chart otherwise the chart will not be plotted.

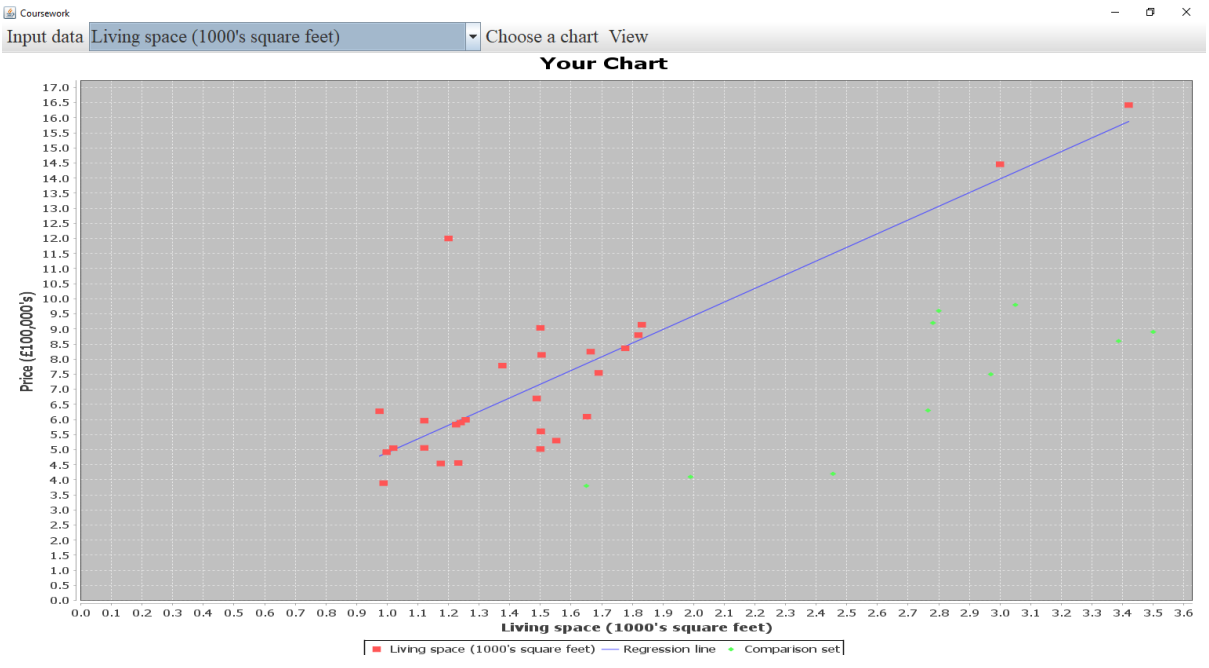


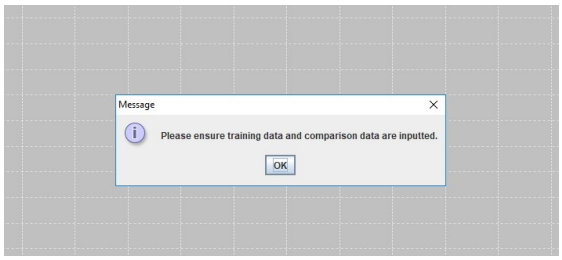
Shown above is a standard scatter chart using ‘site area’ as the independent variable

Error message user will see if no data is added

Error message user will see if no variable is selected

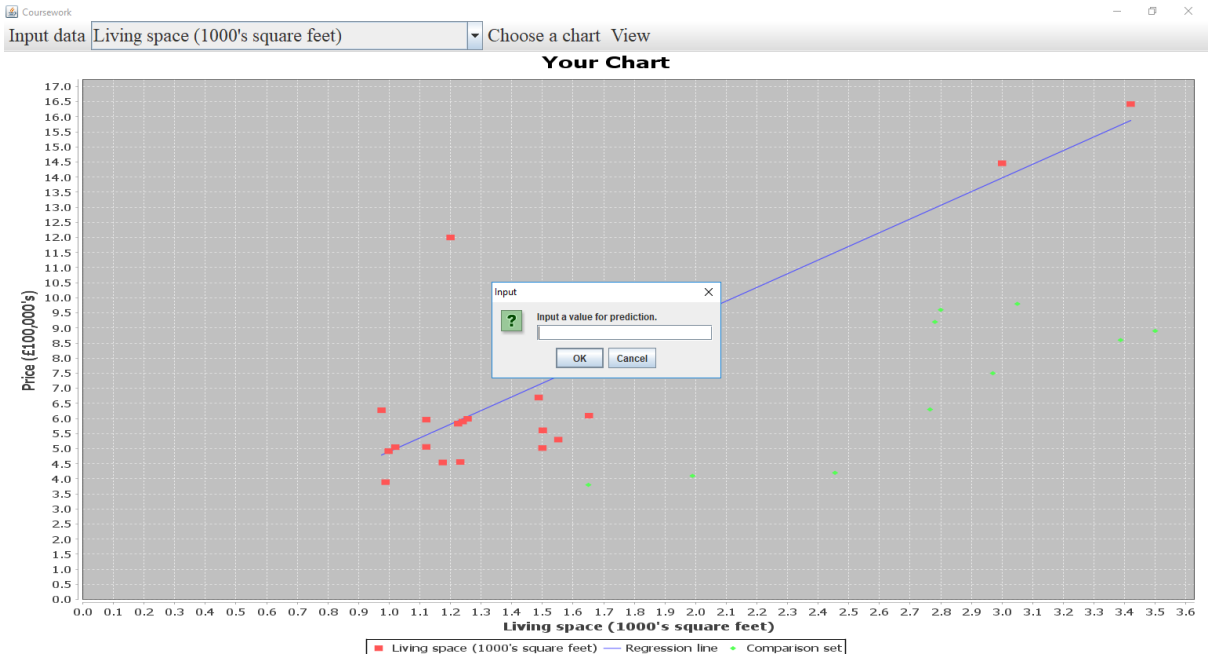
When plotting with the comparison data, in addition to adding the training data set and selecting a variable, the user must also add a comparison data set. Shown below is a scatter plot with comparison data set.



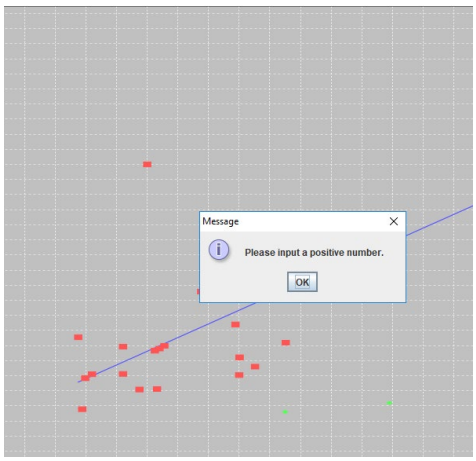


Error message shown if both data sets are not added

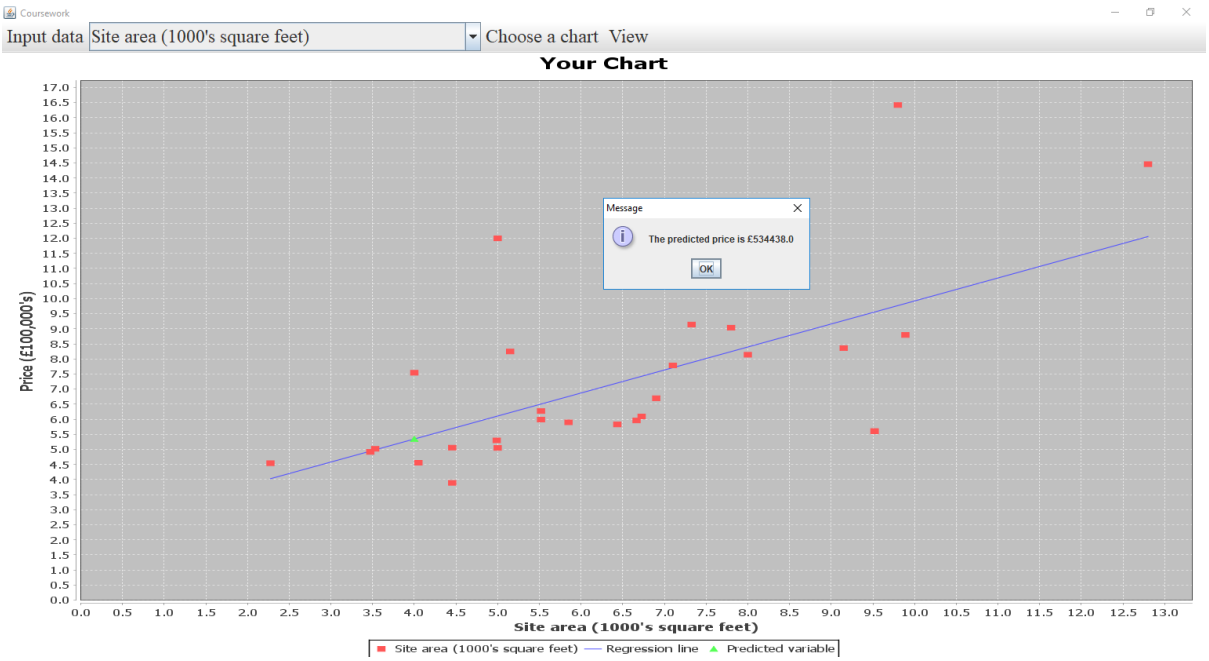
The program also has the functionality to predict the price of a property by entering a value for the independent variable. This is done by selecting the 'Training data with a regression line and a predicted variable' option found within the 'choose a chart' menu. The following popup box appears.



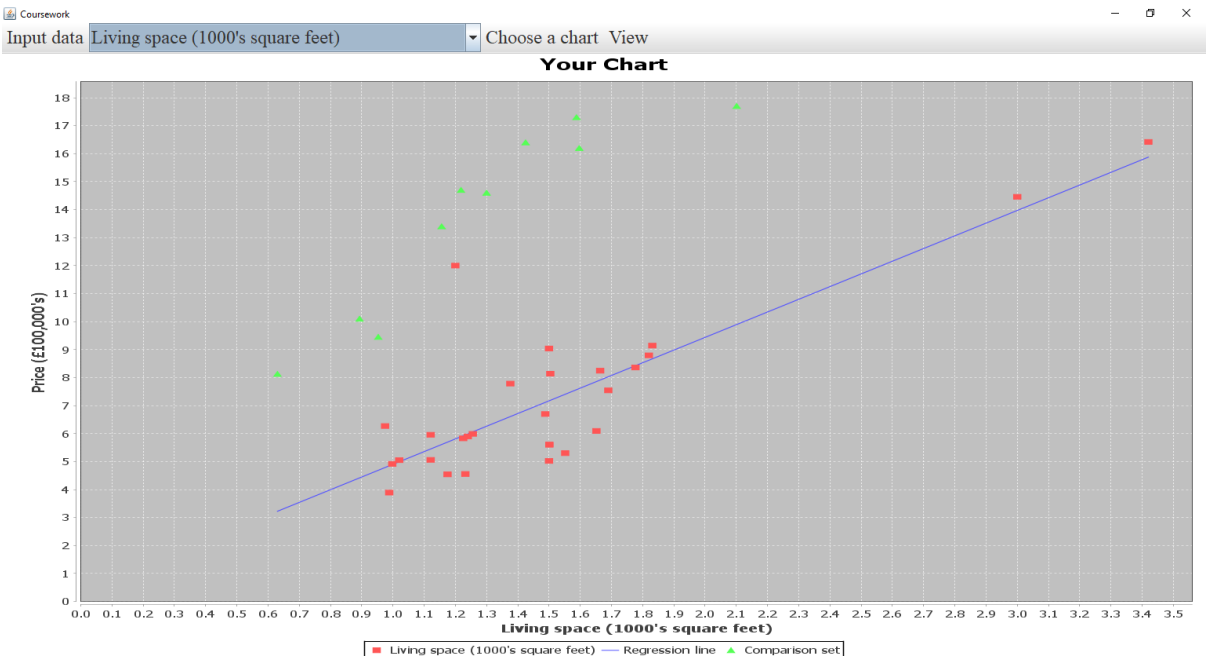
Error prevention and detection has been thought of, so entering a nonsensical value provides an error message shown below.



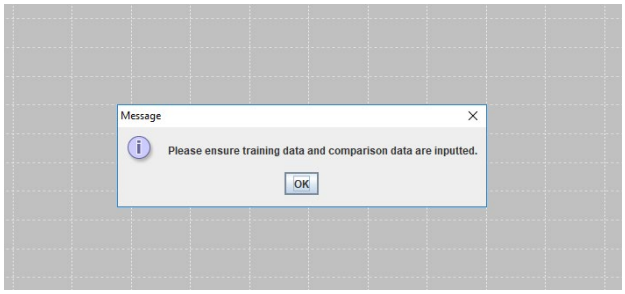
Entering a real number in the box will show a message box telling the user what the predicted price will be (rounded up to the nearest £) and also plots the point on the graph. Note that the point lies on the regression line.



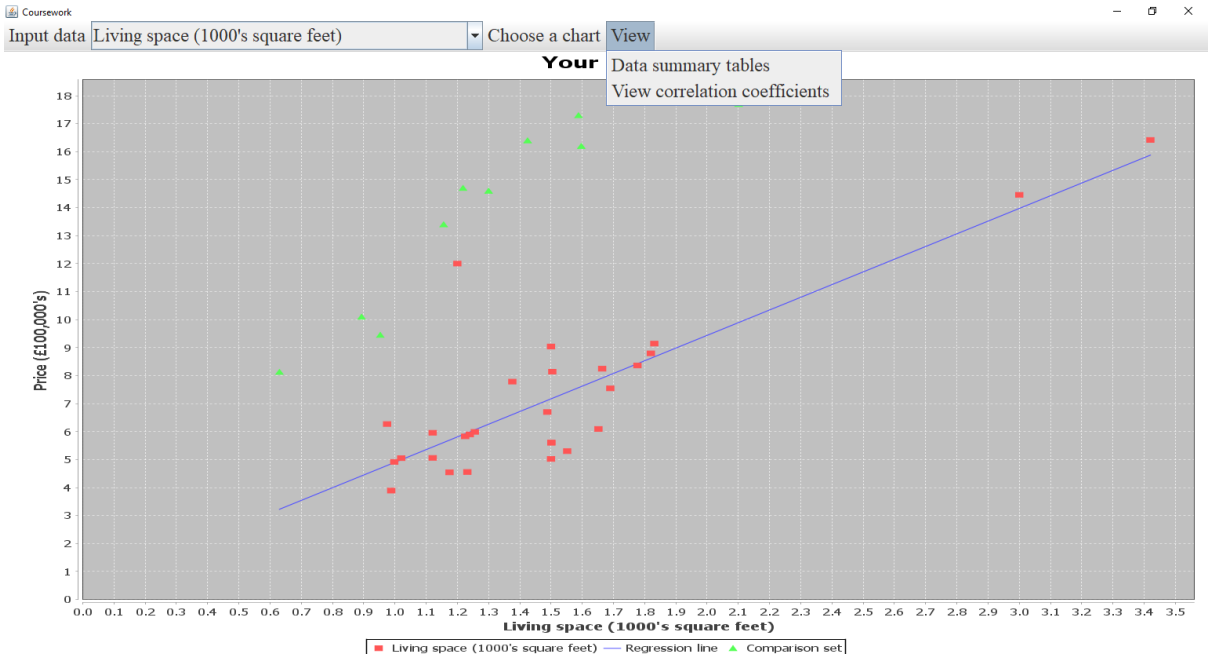
To view the training data set and comparison data set for the variable with the strongest correlation, the user needs to click on the ‘” Best Measure” training data with a regression line and comparison data’ option found within the ‘choose a chart’ menu. This requires the data to be added from the comparison data set and the training data set. Note the user does not need to select an independent variable as this function will show only the variable with the highest correlation.



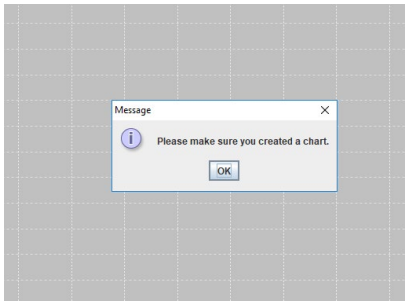
Failure to add one or more of the data sets will result in another error message box to appear, as shown below.



The final menu allows the user to view data summary tables, or the correlation coefficients. To view these options the user must click on the 'View' menu.

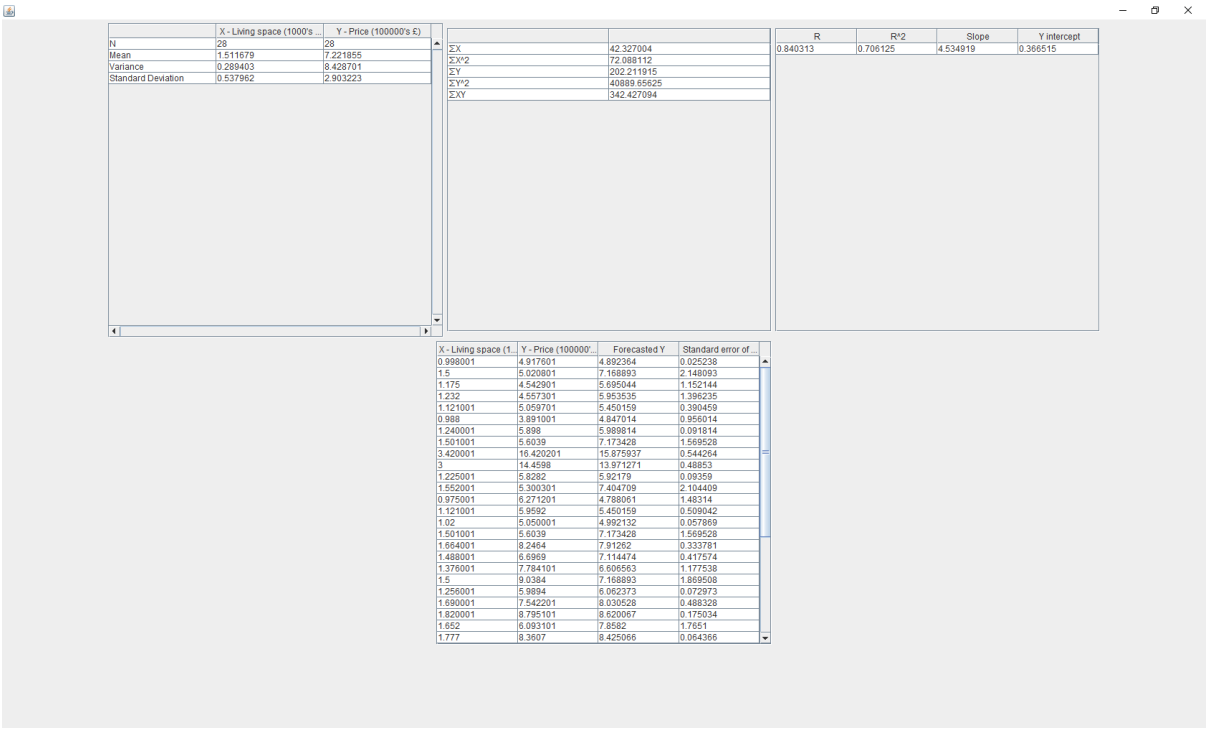


To view the data summary tables, it is necessary to add the relevant data and plot a scatter chart. In addition to the data error messages, the program will also show an additional error message if a chart has not been selected. This is illustrated below



Error message shown if chart not created

If all the necessary data has been added, the program will display the tables in a new window, as illustrated below



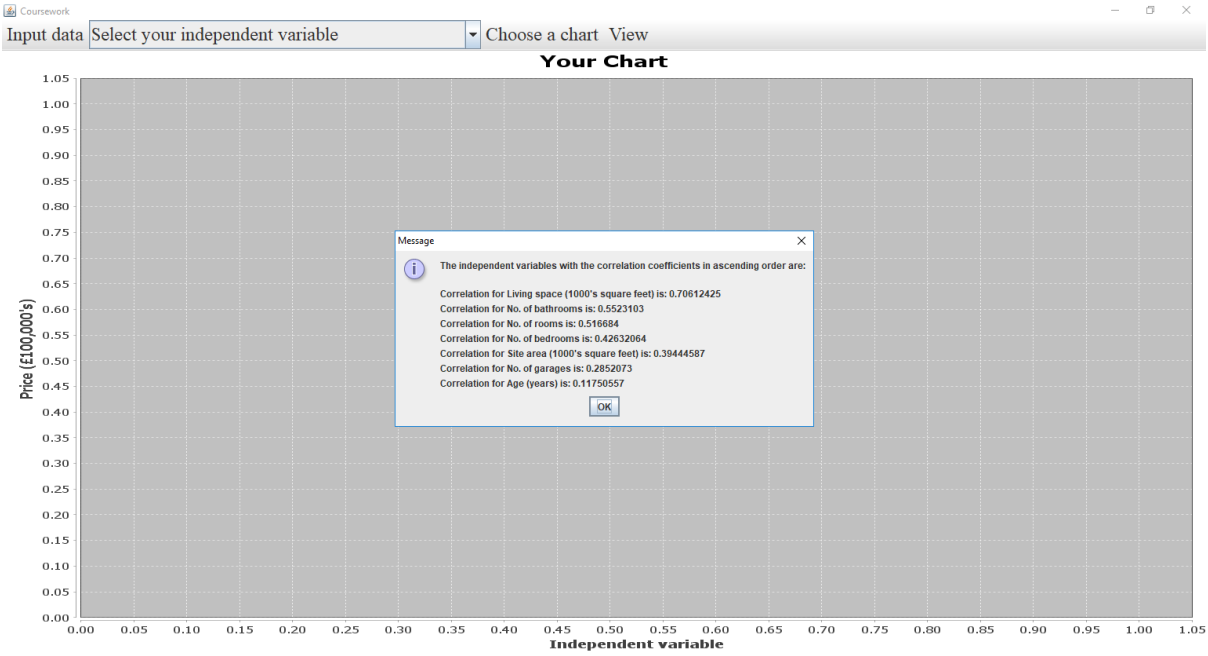
	X - Living space (1000's ...	Y - Price (100000's £)
N	28	28
Mean	1.511679	7.221855
Variance	0.289403	8.428701
Standard Deviation	0.537962	2.903223

ΣX	42.327004
ΣX^2	72.088112
ΣY	202.211915
ΣY^2	40889.65625
ΣXY	342.427094

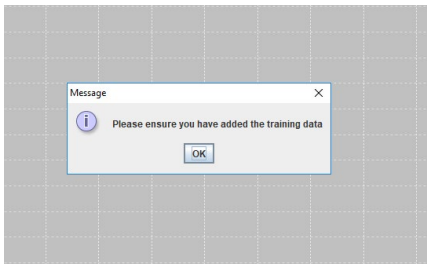
R	R^2	Slope	Y intercept
0.840313	0.706125	4.534919	0.366515

X - Living space (1...	Y - Price (100000'...	Forecasted Y	Standard error of ...
0.998001	4.917601	4.892364	0.025238
1.5	5.020801	7.168893	2.148093
1.175	4.542901	5.695044	1.152144
1.232	4.557301	5.953535	1.396235
1.121001	5.059701	5.450159	0.390459
0.988	3.891001	4.847014	0.956014
1.240001	5.898	5.989814	0.091814
1.501001	5.6039	7.173428	1.569528
3.420001	16.420201	15.875937	0.544264
3	14.4598	13.971271	0.48853
1.225001	5.8282	5.92179	0.09359
1.552001	5.300301	7.404709	2.104409
0.975001	6.271201	4.788061	1.48314
1.121001	5.9592	5.450159	0.509042
1.02	5.050001	4.992132	0.057869
1.501001	5.6039	7.173428	1.569528
1.664001	8.2464	7.91262	0.333781
1.488001	6.6969	7.114474	0.417574
1.376001	7.784101	6.606563	1.177538
1.5	9.0384	7.168893	1.869508
1.256001	5.9894	6.062373	0.072973
1.690001	7.542201	8.030528	0.488328
1.820001	8.795101	8.620067	0.175034
1.652	6.093101	7.8582	1.7651
1.777	8.3607	8.425066	0.064366

The final functionality of the program displays, in ascending order, the correlation coefficients for each independent variable. In order to do this, data must be added either through the keyboard or by adding the training data set. The following output is shown if data has been added



If no data has been added, a familiar error message is shown to the user



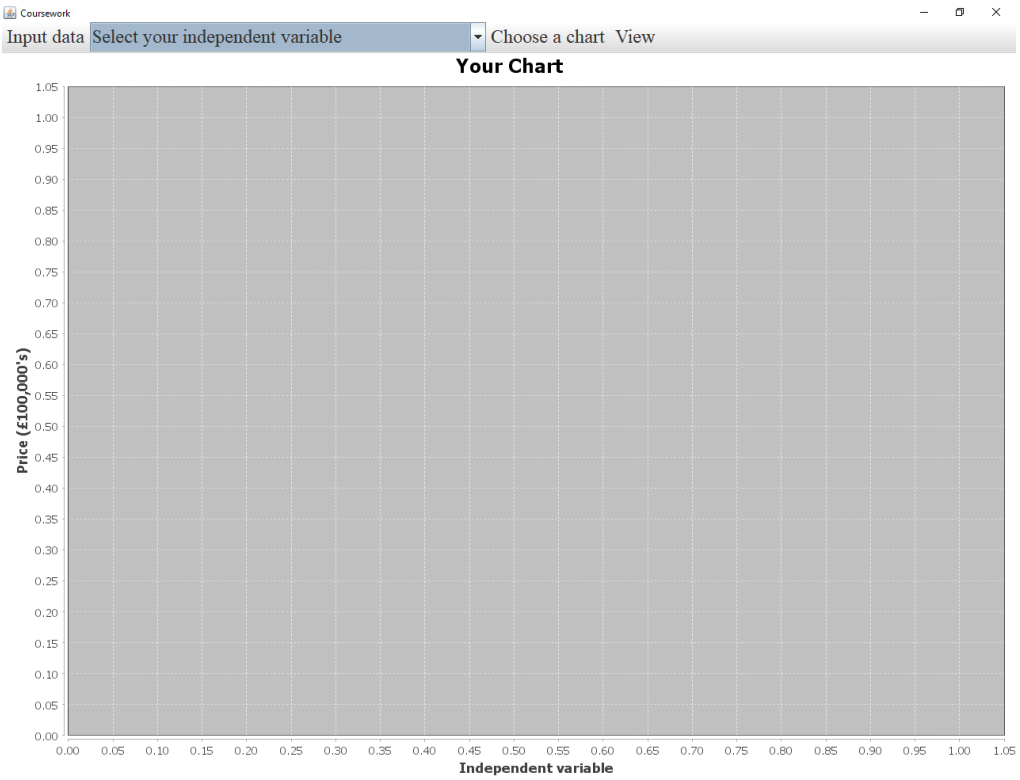
Known flaws within the program and how we could improve the program further

While our program has several error detection and prevention processes in place, there is still a flaw that we have identified. When predicting the price of a property, it is possible to enter nonsensical values in and still receive a prediction. An example is predicting the price of a property, if it has 1.79 bathrooms. The program calculates the predicted price using the regression line but this is of no value. Although the prediction will be in line with the regression line, it is not possible to have 1.79 bathrooms and so it is considered nonsensical. The issue surrounding this is that it is in fact possible to have 1.5 bathrooms, so coding the prediction to only take an integer is not a good idea either. Given more time, we would improve the program by taking this into account, and perhaps rounding a prediction to the nearest 0.5 or prohibiting the user from entering a value that is not a multiple of 0.5.

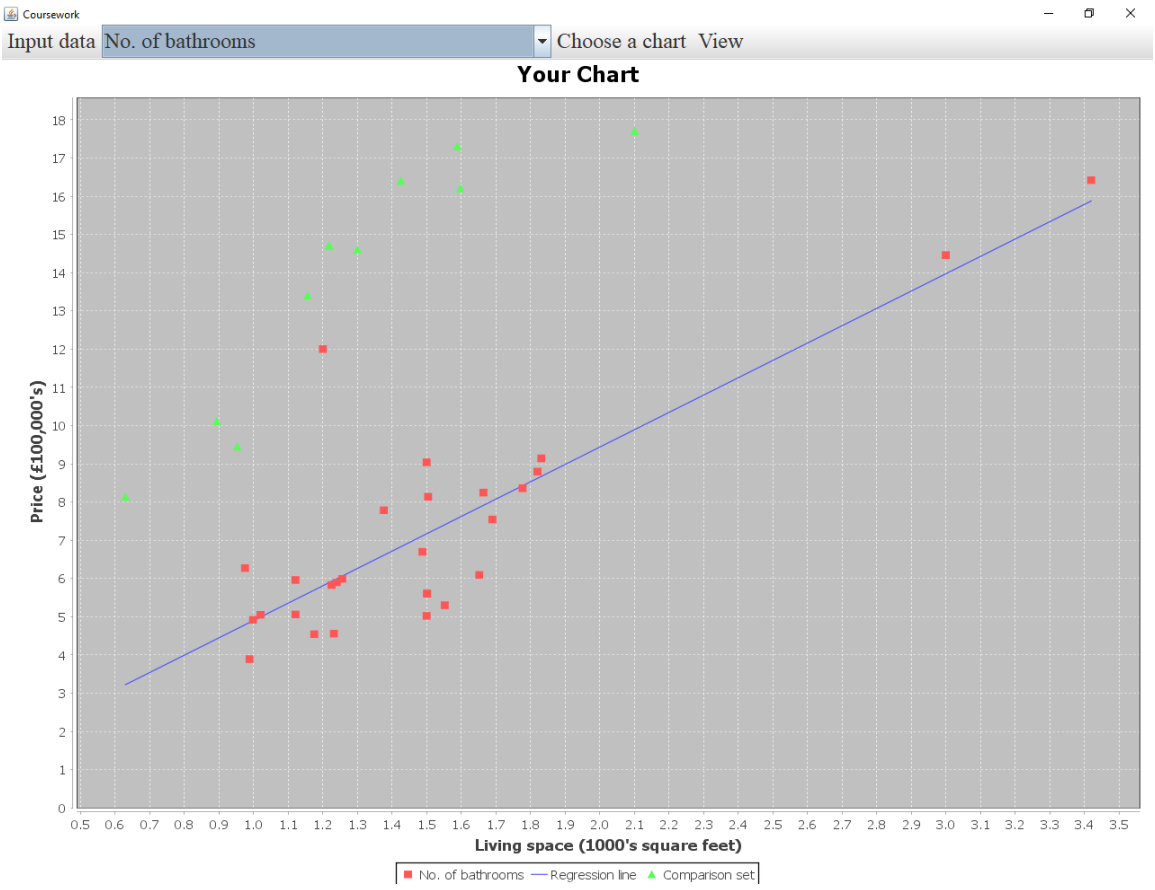
Another flaw that we found was that entering nonsensible data as a predicted variable, actually throws an exception. For example, entering 1000000000000000000 as a predicted number of bathrooms will not display anything on the chart, but will throw an exception visible in Netbeans. Given more time we would find a way to catch this exception or prevent the user from entering nonsensible data. Although an exception was thrown, the program did not crash and continued to function.

Testing that we did

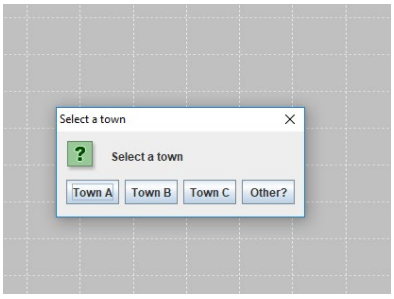
For the program to work properly, it is required for it to open properly and display the GUI, we had to test for this to make sure that there are no errors to stop the program from opening. This was done by pressing run project. When we run the project, the GUI had displayed without faults and the design was shown as expected.



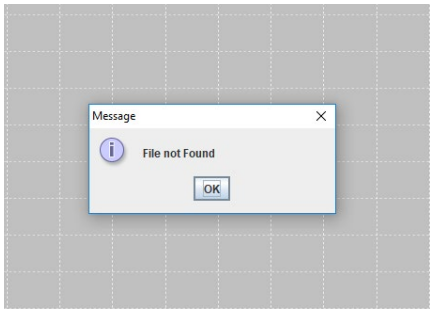
An important test we had to undertake was the display of the training and comparison data on the graph, this was to see the difference between data and to help see where the best fit would be shown. To do this we had to go to 'input data' and select both 'read training and comparison data', once this was done, we would go to the choose chart tab and select display best training data with regression line and comparison data. This gave a display of both data plus a regression line to show the best fit.



There were 3 types of comparison data the user would have the option to choose from, town A, town B, town C. Alternatively the user could add another data set given it was in the same format as the others and was located within the same directory. We had to ensure that the user had the choice to choose which town data they would want to use in the graph. We tested it by creating a pop-up message to show the options of towns.

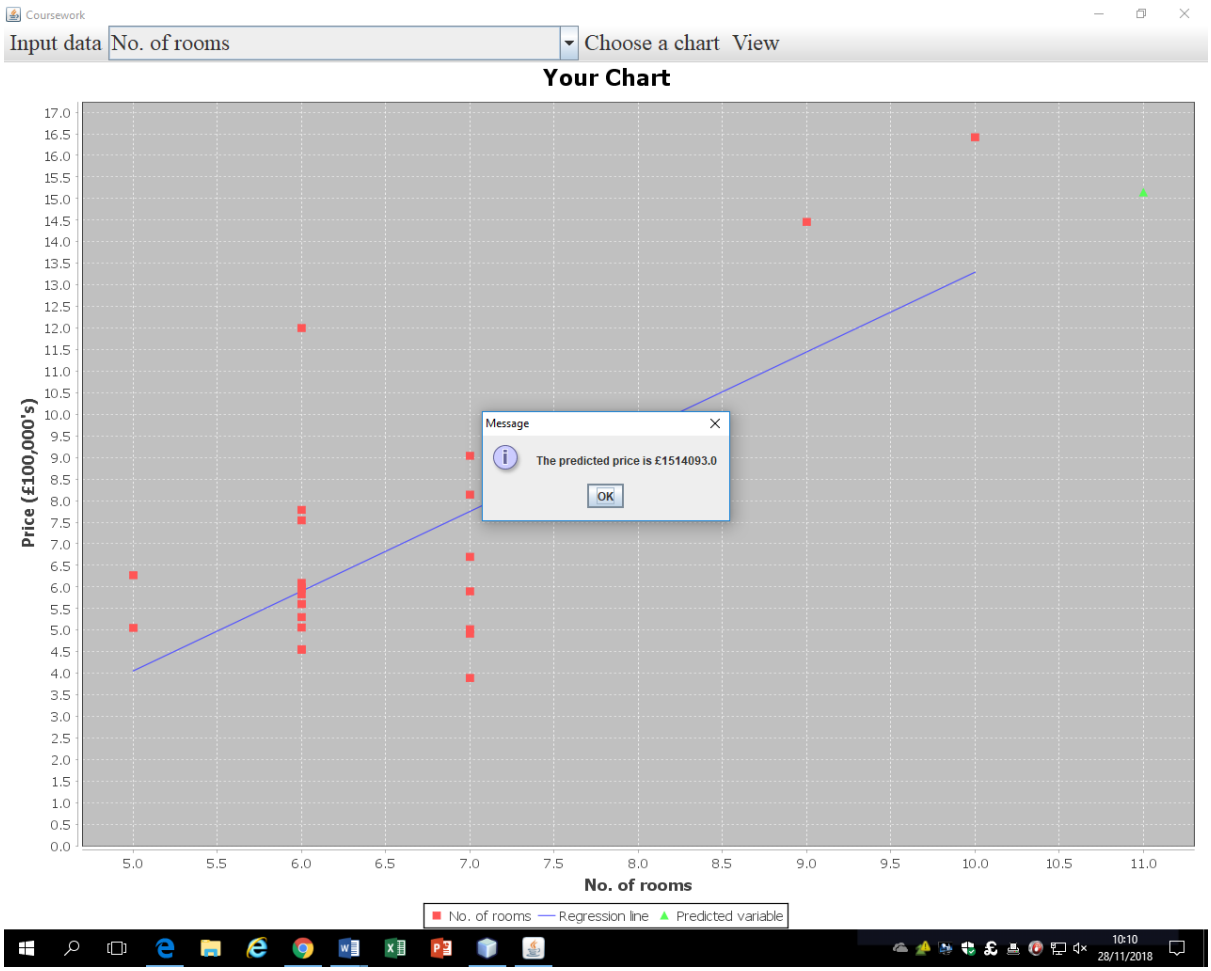


If the file does not exist or is not placed within the correct directory, an error message is shown to the user. This prevents the program from throwing an exception, while also advising the user on what the issue is.



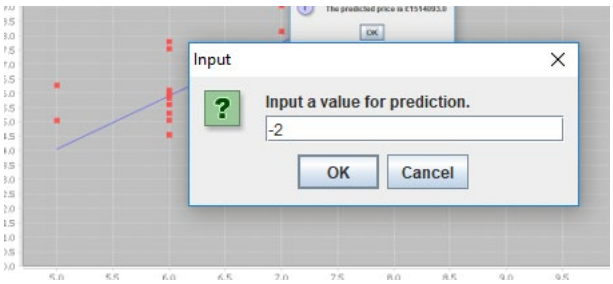
This message is also displayed if the user tries to add another comparison data file that does not exist.

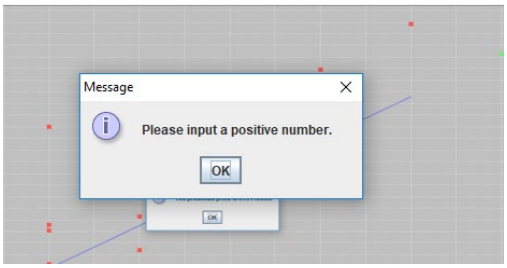
One way we tested our system was using the keyboard where we inputted our own values into the system, this was to judge how well the predicted value would be calculated. We did this by inputting the training and comparison data so that those values are stored to be displayed on the graph. After which we selected the independent variable we would want to use, in this case we chose the room variable and selected the 'training data with the predicted line' and entered 11. When we clicked 'ok' it told us the value that it would cost to have 11 rooms with a predicted triangle shape shown on the graph.



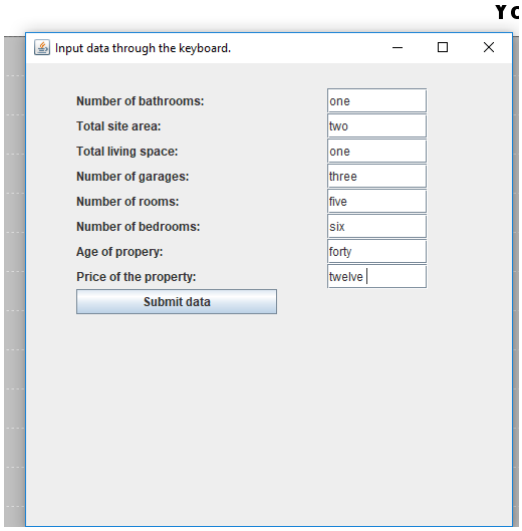
Another way we tested out the system was by trying to input a number which was equal to 0 or less than to see if the program would accept it or not. We have thought of this when programming the popup box, so included a try-catch to ensure the program ran smoothly.

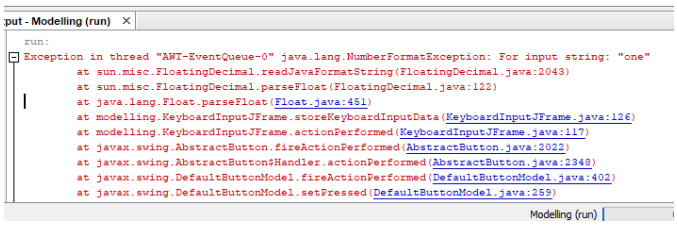
Within the input message we entered -2 and got an error message box telling us to input a Positive Number.



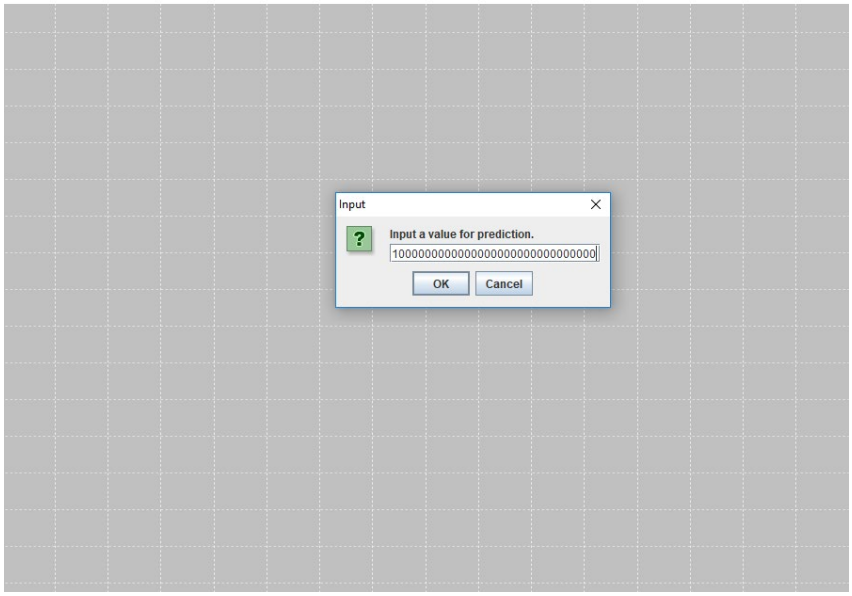


Concerning the inputting of our own data we had to make sure that the user can only enter positive numbers and not text or negative numbers. We tested to see if this was possible however when carrying the test out, we realised that we received and exception error. To sort this problem, we had to include a 'try-catch' which would stop the user form entering letters into the input.





Showcasing what happens if words are entered into the form



Entering a nonsensible value in the prediction throws an exception, resulting in the error shown below.

<u>Table showing other tests we have carried out</u>

What was Tested	Expected Results	Results Shown	Reason for Pass/Fail	Changes made
Program loaded correctly.	The GUI should load with no faults or error.	The GUI was successfully loaded, and no faults had occurred.	Passed - because the program opened properly without a ‘rum project’ displaying suggestion the code has faults.	N/A
The option to choose which town you would like to include	A pop-up box asking which town the user wanted to use.	A pop-up box asking which town the user wanted to use.	Passed - as the user was able to choose out of the 3 towns, they wanted.	N/A

when the 'Read in the comparison data' is selected				
Display of training and comparison data on the graph together.	Points showing the training data and comparison on the same graph.	Both training and comparison data are shown on the graph with a regression line.	Passed - because it showed what was intended also it shows other addition parts.	N/A
Input of data via keyboard to see the Regression Algorithm was calculated correct in the table's summary.	A table showing the calculating regression.	The table of the regression analysis showing the correct data.	Passed - as we calculated it ourselves giving us the same value.	N/A
Display of linear regression results table.	Shows a table of regression analysis showing the value of R, R^2, Slope and Y-intercept.	It shows a table of regression analysis showing the value of R, R^2, Slope and Y-intercept.	Passed - as it gives the calculation of the data shown on the graph and the values calculated are correct.	N/A
The predicted price value after input.	A green tringle point showing the variable of what was inputted and the price it should cost.	A pop-up message showing the price it will cost, also the green triangle representing the predicted value.	Passed - due to the fact it shown the prediction on the graph also it showed a message saying the price it would cost.	N/A
Value entered	An error message	When entered a	Passed - because an error message	N/A

less than 0 in the predicted input.	telling the user this value is not acceptable.	pop-up message was shown telling the user to input a positive value.	had appeared, and no calculation was undertaken.	
Tested to see if decimal numbers can be accepted in variables requiring whole numbers.	A message that shows please enter a whole number.	Showed a price for the given decimal input.	Failed - as a message did not show asking the user to enter a valid number.	Caught the exception that was thrown, and added a message to inform the user to enter a valid number
Tested to see if you can enter text and negative values into the 'enter through keyboard'	A pop-up asking the user to enter a number.	An Exception error had been taken place	Failed - because an error exception had occurred in the system. However, we resolved this by adding a 'Try-Catch'.	Caught the exception that was thrown and added a message to inform the user to enter positive numbers only.
Tested to see if you can enter a nonsensible value in to the prediction popup	A pop-up asking the user to enter a number	Index out of bounds exception	Failed – because an exception error had occurred. Although it did not crash the program, it was unable to produce a graph due to the entered value being greater than 32 bits	N/A - given more time, we would find a way to catch the exception, or prohibit the user from entering nonsensible values.

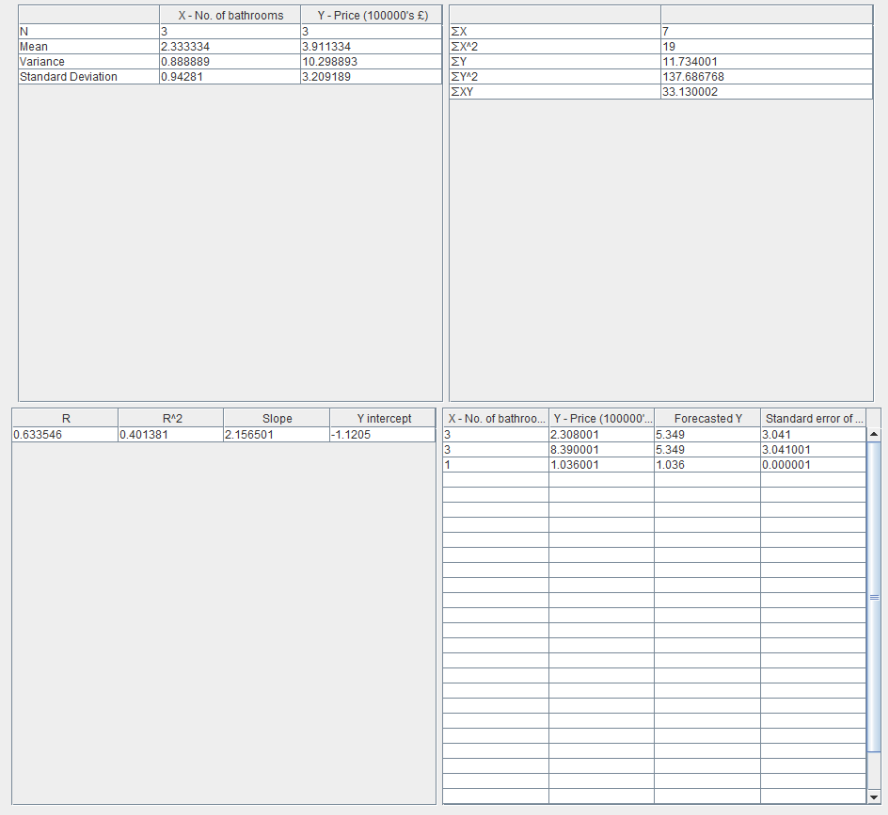
Since one of the requirements was to show a table showing the Linear Regression Analysis Results, we had to carry out a test to ensure that the calculation carried out was correct. We did this by inputting the training data into the system and displaying it on the graph, once the graph had been shown we then opened the table showing us the linear regression results.

To ensure these values are correct we had to calculate the values for ourselves.

X_i (bathroom as independent variable)	Y (Price)	Expected R^2	Expected Y intercept	Actual R^2	Actual Y intercept	Changes we made
3 3 1	2.308 8.390 1.036	0.401381	-1.1205	0.401381	-1.1205	N/A

Mean X = 2.333333’
Mean Y = 3.911334
S_{xx} = 2.6667
S_{xy} = 5.7507
Gradient = 2.1565
Y intercept = -1.1205

Our expected output and actual output match, showing that the program we have coded is accurate and provides the correct answer. Our tables in the program match the values that we calculated too. As the values are correct, we didn’t feel the need to test it for other values.



	X - No. of bathrooms	Y - Price (100000's £)
N	3	3
Mean	2.333334	3.911334
Variance	0.888889	10.298893
Standard Deviation	0.94281	3.209189

ΣX	7
ΣX^2	19
ΣY	11.734001
ΣY^2	137.686768
ΣXY	33.130002

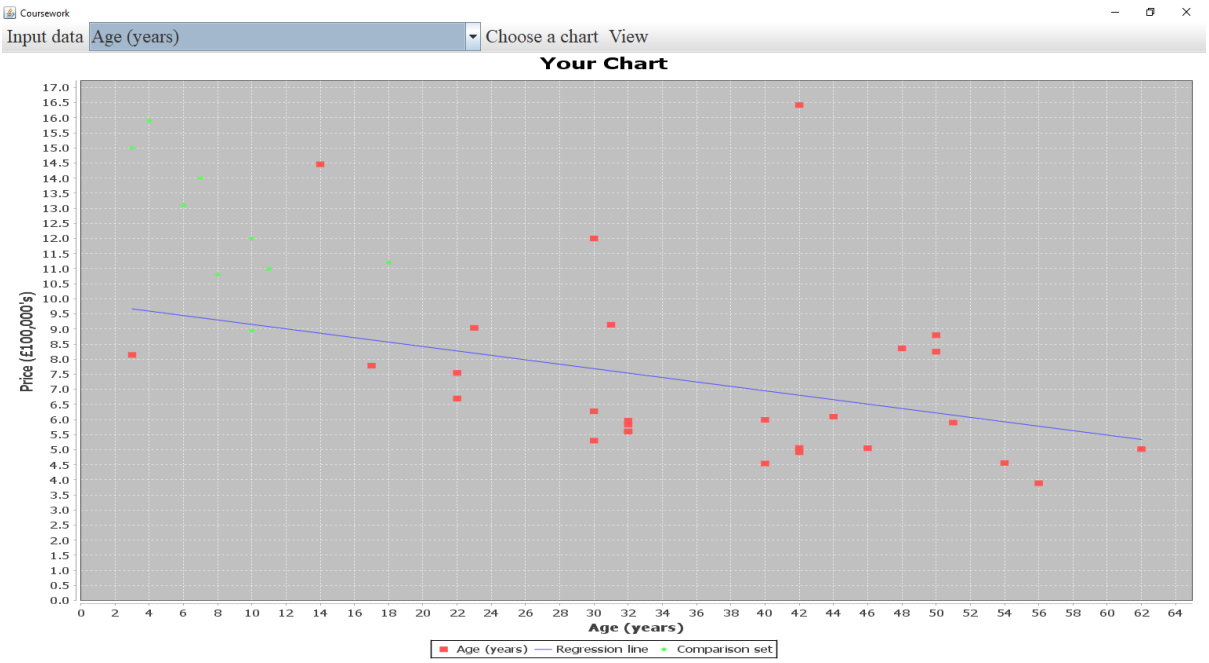
R	R^2	Slope	Y intercept
0.633546	0.401381	2.156501	-1.1205

X - No. of bathroo...	Y - Price (100000'...	Forecasted Y	Standard error of ...
3	2.308001	5.349	3.041
3	8.390001	5.349	3.041001
1	1.036001	1.036	0.000001

Analysis of results for the comparison data

Town A

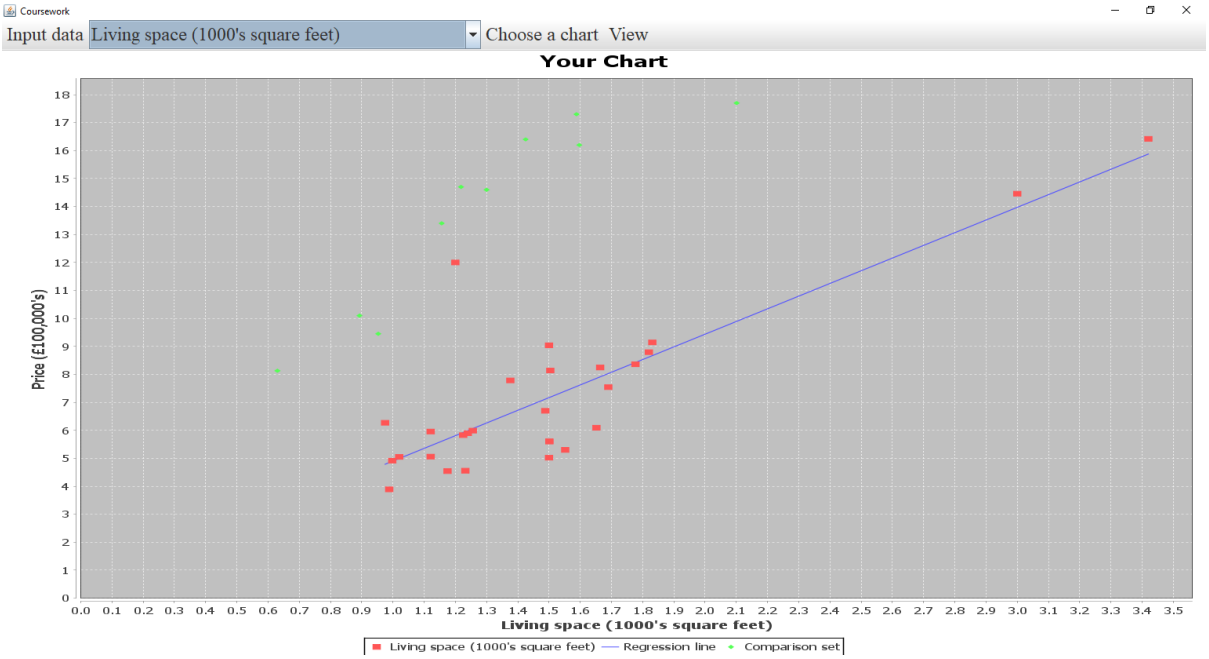
In general, the data for town A appears to follow the trend set by the training data. The variables 'number of garages' and 'age' do not fit the trend set by the training data, and deviate away from the regression line significantly. This indicates that the regression line does not provide an accurate prediction of price. Upon further analysing the data, we can see from the tables that the age of a property in fact provides the lowest correlation coefficient and as a result is not an accurate model. The number of garages is the variable that provides the second lowest correlation coefficient which also indicates that it is a poor model and will not provide accurate predictions. It was interesting to note however that a general trend of decreasing price associated with increasing age of property. Town A appears to have relatively newer properties when compared to the training data set. Other features of properties within town A show that the houses are larger, not only in living space but also with respect to site area, and have a larger number of rooms within the houses.



As evident from the illustration above, the comparison data set does not follow a trend at all when age is selected as the independent variable

Town B

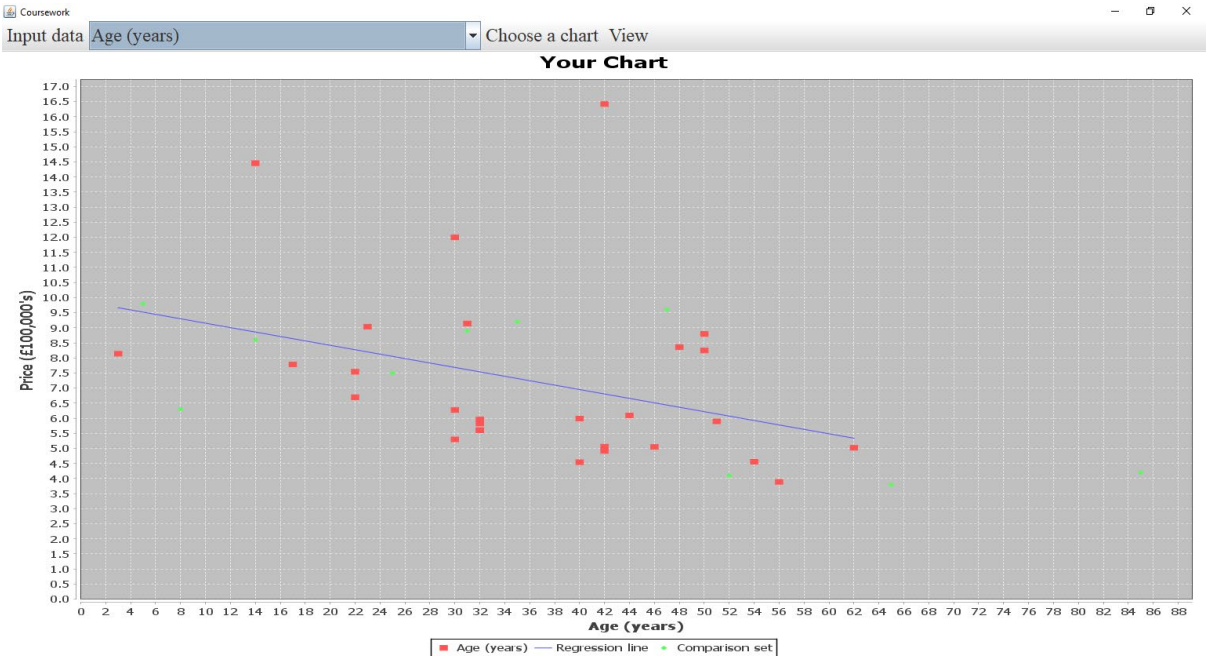
Town B does not appear to follow any trends seen with the training data set. The only variables that somewhat follow a trend are 'age' and 'number of bathrooms'. The data values in town B are very far away from the regression line, meaning this is not a good model for predicting prices of properties in town B. Even with the variable that provides the strongest correlation (living space) does not provide a sensible prediction. The data tables are also indicative of this, as the standard error of estimate is shown to be at an unreasonably high figure. From this analysis, we can infer that town B is an expensive area to live in with smaller houses being much more expensive than large houses in different towns. To provide a better prediction of price, the regression analysis should take into consideration this town, otherwise it is not a good indicator of prices at all.



As evident from the above illustration, even the variable with the strongest correlation is unable to prove meaningful with respect to town B's dataset. This means that this particular regression line is not a good model for predicting houses in town B.

Town C

Town C also does not fit in very well with the regression line created from the training data set. The only variable that provides a reasonable prediction is 'age'. The data for town C generally follows the trend of decreasing price with increasing age. However, the other variables have a much greater standard error of estimate and are therefore would not provide a meaningful prediction. The data tables reinforce this analysis by showing the increased standard error of estimate.



The illustration above shows that the age of a property in town C follows on from the trend within the training data set. But the correlation coefficient for the variable age is much less than 0.8, meaning it is unlikely to reflect an accurate prediction.

Evaluation of the choice and performance of data structures

The data structures we chose to use in our application were ArrayLists and two-dimensional arrays. We use ArrayLists to store the data about the properties. As our data will not contain key/value pairs, we ruled out the usage of maps from our application. Sets do not use indexes to order the elements, which prohibits us from accessing specific elements when we need them. We also assume that there might be two properties, which will have the same data, so we chose not to use sets as duplicate values are ignored by them and we must include all the values for correct calculations. We have considered LinkedLists for storage. LinkedLists perform better when adding and removing data from them, but as we never remove data (we only add data when the initial training/comparison files are read in and when we manually add in a single property), this ability is not very useful.

ArrayLists on the other hand, allow for constant-time retrieving and adding of the values (we only add to the end of the ArrayList and not the middle), which is useful for us when we do regression analysis, create datasets and create tables for charts. If an ArrayList, would have to be resized when a new value is added, this may pose a significant consumption of time on that operation.

We use two-dimensional arrays to create tables for charts. As an ArrayList can be resized while arrays cannot, we used our ArrayLists to store data and populate the two-dimensional arrays so that the data can be displayed by the tables. As we retrieve the data from the ArrayLists and add it to the two-dimensional arrays, the operations are in constant time.

Individual contributions

Name	ID	Contribution
Roshaan Nazir	000834566	100%
Maks Smirnov	000979299	100%
Thomas Stoyles	000990057	30%
Michael Mitchell	000978660	50%

Personal Reflection

Michael M Mitchell:

When looking back at this project that my Group and I had undertaken, I concluded that this project was not as simple as I thought. At the start of this project my Input into the task was minimal as I could not make it to some of the group sessions and reading the specification, I was slightly confused.

I am relatively decent at both programming and maths, however as the project included creating a program based on calculation, this was one of the reasons I struggled to make an input into the project.

However, after a serious talk with the group I found out there were tasks which needed doing which I could help in, such as programming the tables. With my fellow member we created the tables for the summary table and added additional code to show the required values. I also was able to give support to the team where needed especially to those that were struggling by researching on what they needed to do and help them in whatever way I could. During the report stage I was able to add contribution by doing some of the report where I was undergoing the testing of the program to ensure any errors found were fixed. I believe I could have done better if I had taken more time in understand what was asked of us and having more support from the team.

Roshaan Nazir:
My experience of completing this assignment has been generally positive and I feel that I have learnt a lot, not only from a programming perspective but I feel I have also gained a better understanding of algorithms and data structures.

When I initially read through the specification, I believed that the assignment was not as difficult as it seemed and so did not commence work on it straight away. I believe that underestimating the assignment was a mistake. As the deadline drew nearer, it became apparent that I would need to spend a lot more time working on this than I had previously anticipated. I had assumed that the most challenging part of this assignment, was performing linear regression on the sets of data and plotting this on a chart. But this was not the hardest part. The hardest part of the assignment was ensuring that the program was written in a way that promoted ease of use and provided an appropriate level of detail for the user. It was also important to ensure that we were adhering to the rules of object-oriented programming and we regularly updated code to reflect this.

Seeing as this was group work, we thought it would be useful to utilise our strengths therefore myself and another group member acted as 'principal programmers'. Together we compiled almost all of the code. I predominantly programmed the tables, creating the GUI, reading in the data files as well as helping out with other sections of the code when required. I also authored part of the report ('The design of the GUI and model including assumptions', 'Known flaws within the program' and 'Analysis of the comparison data'), as well as proofreading it for errors. As this was a group assignment, we used the aid of online code repositories (GitHub), which allowed for version control, whilst also allowing easy access to each other's code. If I could have done anything differently a second time around, I would commence work much earlier and spend more time on ensuring code was object oriented.

Thomas Stoyles:
My group collectively decided to give me a 30% contribution which is how much the group believed I contributed and to an extent I understand as to why they decided this. I originally thought that the coursework would be easy and simple, like last years, because of this I put minimal input into the coursework while my group started it. Once my group pointed out to me that the coursework is going to take more time than originally expected, I decided to start looking into how to write the java code.

For me personally I feel like I could have done multiple things differently. One of these things being instead of researching how to do the code, I should have spent more time on creating the code within the Java program regardless of whether it was correct or not. If the code was incorrect then we as a group would have been able to find a solution to the issue.
My group was fair and gave me a time limit on when they would like my code due however, as I was unable to learn the code within the allotted time limit and my group decided to move on with the coursework to make sure we met the deadline. The big issue with my contribution towards this coursework was my inability to learn.

Another thing I could have done better is start to learn the java code sooner. Looking back, I should have known my lack of java knowledge would affect me and my group in a negative way.
Due to my lack of knowledge in the area and failure to learn the code quickly, my group finished the areas that I was meant to code leaving me with some knowledge of how to create the java but without the implementation within our main program. One thing that I thought I did well however was my contribution to the report in which we wrote, mainly the User documentation, I believe this input into the report is what gave me the 30% contribution rather than something lower. Something

I would do differently if we had to do the coursework again is to start learning the Java code earlier than what I did.

Maks Domas Smirnov:
Reflecting on the coursework, I must say that I contributed a lot to it. There was a problem with miscommunication and a lack of enthusiasm with two members of our team, who have not contributed enough for a substantial contribution percentage. I was the one to try to organize meetings and set-up group gatherings but did not receive as much help as I have hoped for. In turn I was required to manage more of the workload myself, taking on more tasks as delegated jobs have consistently not been completed.
Concerning the work I did, I would have rather spent more time on OO analysis and design than implementation. This led to mid-project time-costing changes that could have been avoided. While working on this coursework, I have successfully learnt to properly use provided official java documentation.
I can honestly say that I am responsible for developing most of the functionalities in the application. The Property, RegressionAlgorithm, and KeyboardInputJFrame classes were predominantly developed by me. I have also made the decision to introduce the external chart library JFreeChart into our application. Most of the functionalities, which used that library, were coded by me. I have also provided the analysis and discussion for the choice of data structures for our report.
As per recommendation of our lecturer, to keep the code safe and easy to manage, I introduced our group to an online software repository and version control tool GitHub. As this tool was new to me too, I have learnt to use it efficiently to manage and control code versions which will definitely be useful in my future projects.

References

In our program, we have authored all of the code ourselves. However, to do this we required the aid of the following resources:

1. https://docs.oracle.com/javase/7/docs/api/javax/swing/table/DefaultTableModel.html - using JTables
2. https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html - enhanced for loops for iteration
3. https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html - different layouts to use
4. https://dzone.com/articles/arraylist-vs-linkedlist-vs [Accessed on 10:43, 28/11/2018]
5. https://stackoverflow.com/questions/322715/when-to-use-linkedlist-over-arraylist-in-java [Accessed on 10:44, 28/11/2018]
6. http://www.jfree.org/jfreechart/ - JFreeChart website
7. https://sourceforge.net/projects/jfreechart/ - JFreeChart download link
8. In addition to the above, a method within our program (makeMenuItem – found in modelling class) was influenced from lecturer Andy Wicks' code given to us in the previous term.