FINAL PROJECT

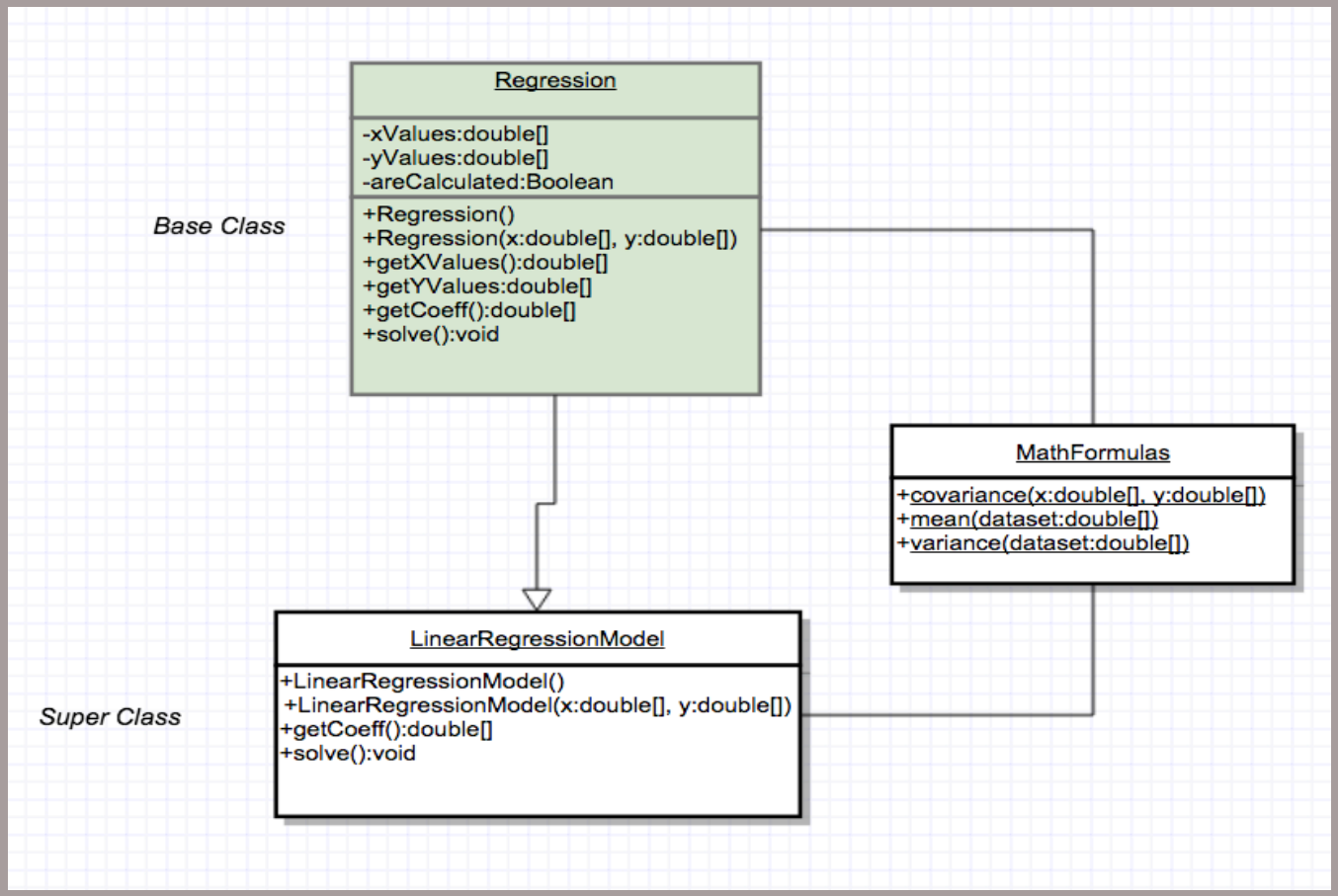
Tyler Bobik

Original Proposal Description

- My original proposal was to create a java program that took in two arrays and calculates a simple linear regression, and various statistical calculations.
- A simple linear regression predicts scores on one variable from the scores on a second variable.
- Linear regression consists of finding the best fitting straight line through the points.

Original Proposal Description

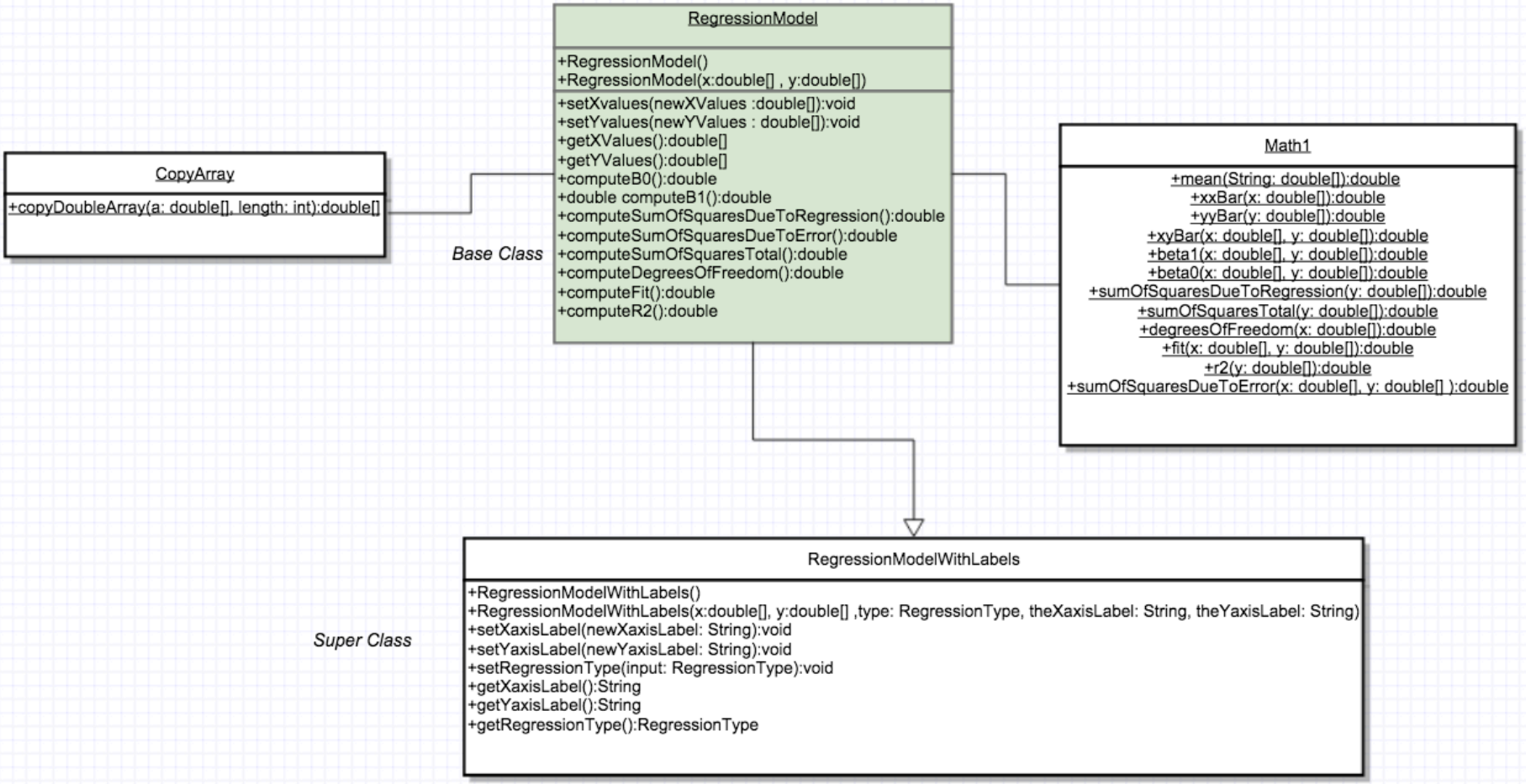
- In order to implement this on my original project proposal I planned on creating three classes.



Original Proposal Failure Technical Explanation

- I soon realized that my project was not going to work as I planned:
- Not fully understanding what inheritance was
- My super class ended up being pointless
- Also, I thought in order to use my getCoeff and solve methods in my super class I had to create the same method in my base class but leave it empty, which was incorrect

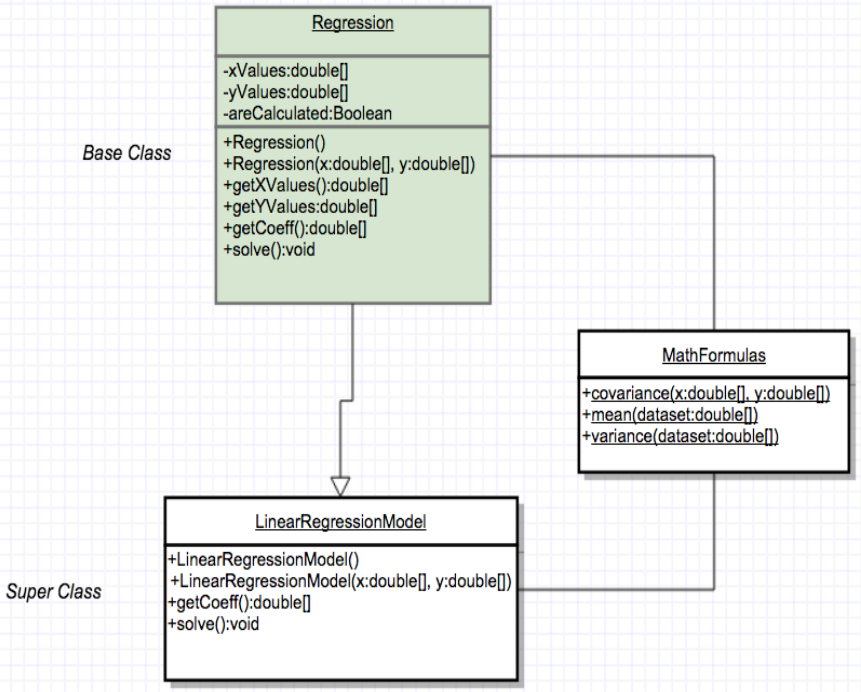
Fixing my Project



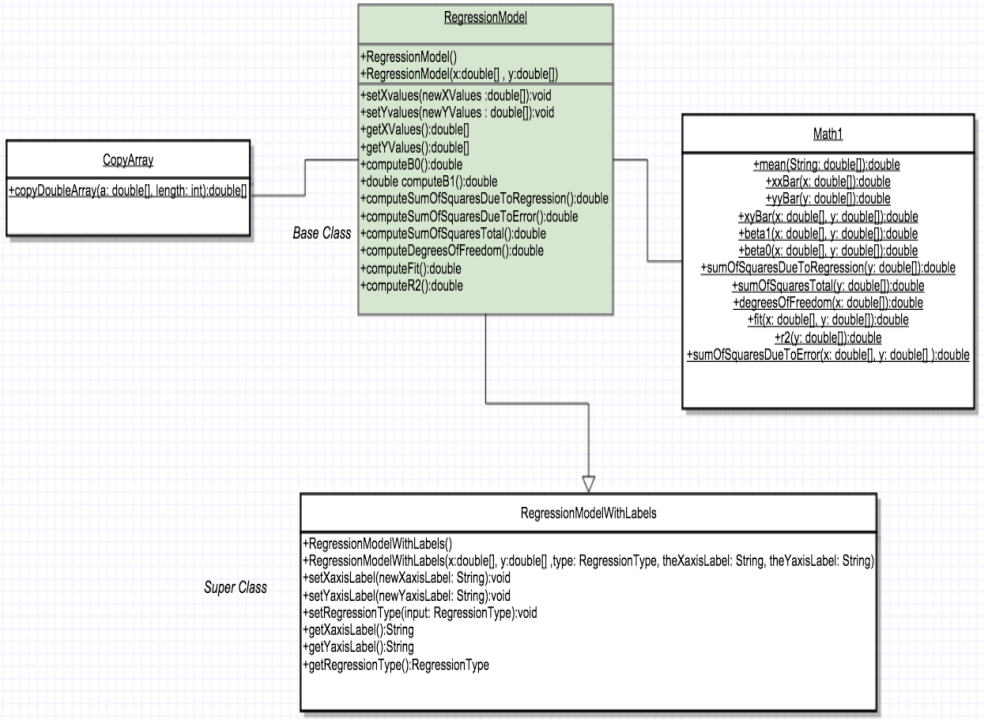
Fixing my Project

- I fixed my derived class so that it now has a purpose to label the x and y axis of my graph
- Also, removed the pointless compute method in my derived class

Original



Revised



Added Features

- Added more calculation methods and formulas
- Made just base class contain my computation methods instead of my base and my derived class
- Made my program output and save a scatter plot
- Also, made a correct derived class that takes in labels for the x and y axis for the graph

Topic #1: Console Input With Error Checking

- Console Input With Error Checking: In Test.java:44 takes input for x values
- Error checking occurs when the array is inputted in the constructor

```
43        // takes input for y
44        System.out.println("Please enter y values that are numeric and seperated by commas:");
45        inputLine2 = keyboard.nextLine();
```

```
25    public RegressionModel(double[] xValues, double[] yValues)
26    {
27        if(xValues == null || yValues == null)
28        {
29            System.out.println("Fatal Error creating regression model.");
30            System.exit(0);
31        }
32        else if(xValues.length == 0 || yValues.length == 0)
33        {
34            System.out.println("Fatal Error one or more zero arrays have zero lengths.");
35            System.exit(0);
36        }
37        else if(xValues.length != yValues.length)
38        {
39            System.out.println("Fatal Error array lengths are not equal.");
40            System.exit(0);
```

Topic #2: Console Output & Formatted Output

- Console Output: in Test.java:53
- Formatted Output: in RegressionModel.java:244
- Formatted output mostly with Decimal Format

```
The x values are: [0.0, 1.0, 3.0, 5.0, 8.0, 4.0, 3.0, 4.0, 8.0, 2.0]
The y values are: [20.0, 30.0, 15.0, 35.0, 80.0, 95.7, 60.15, 100.25, 11.5, 71.31]
The equation for the linear equation is: y = 44.90 + 1.84x
The sum of squares due to regresson (SSR) is: 10267.18
The sum of squares due to error (SSE) is: 35318.40
The sum of squares total (SST) is: 10267.18
The dregrees of freedom (DF) are:8.00
The fit is: 48.58
The R2 is: -3.00
The x axis label is: Higher Education (Years)
The y axis label is: Income (Thousands)
```

```java
    public String toString()
    {
        DecimalFormat roundedTwoDec = new DecimalFormat("#.00");
        return "The x values are: " + Arrays.toString(xValues) + "\nThe y values are: " + Arrays.toString(yValues) +
                "\nThe equation for the linear equation is: y = " + roundedTwoDec.format(computeB0()) + " + " +
                roundedTwoDec.format(computeB1()) + "x \n" + "The sum of squares due to regresson (SSR) is: " +
                roundedTwoDec.format(computeSumOfSquaresDueToRegression()) + "\nThe sum of squares due to error (SSE) is: " +
                roundedTwoDec.format(computeSumOfSquaresDueToError()) + "\nThe sum of squares total (SST) is: " +
                roundedTwoDec.format(computeSumOfSquaresTotal()) +
                "\nThe dregrees of freedom (DF) are:" + roundedTwoDec.format(computeDegreesOfFreedom()) +
                "\nThe fit is: " + roundedTwoDec.format(computeFit()) + "\nThe R2 is: " +
                roundedTwoDec.format(computeR2());
```

Topic #3: Selection Statements

- Selection Statement: in Test.Java:95
- Asks user to press x to exit, if statement used

```
92          //to exit program
93          System.out.println("Press X to exit");
94          String answer = keyboard.nextLine();
95          if(answer.equalsIgnoreCase("x"))
96          {
97              System.exit(0);
98          }
```

Topic #4: Advanced Boolean Expressions

- Advanced Boolean Expression: in RegressionModel.java:32
- Makes sure x or y arrays length do not equal zero

```
25      public RegressionModel(double[] xValues, double[] yValues)
26      {
27          if(xValues == null || yValues == null)
28          {
29              System.out.println("Fatal Error creating regression model.");
30              System.exit(0);
31          }
32          else if(xValues.length == 0 || yValues.length == 0)
33          {
34              System.out.println("Fatal Error one or more zero arrays have zero lengths.");
35              System.exit(0);
36          }
37          else if(xValues.length != yValues.length)
38          {
39              System.out.println("Fatal Error array lengths are not equal.");
40              System.exit(0);
41          }
42          else
43          {
44              this.xValues = CopyArray.copyDoubleArray(xValues, xValues.length);
45              this.yValues = CopyArray.copyDoubleArray(yValues, yValues.length);
46          }
47      }
```

Topic # 5: Repetition Statements

- Repetition Statements: in Test.java:36
- Use of for loop to fill array for x values

```
34          x = new double[xValues.length];
35          // parses the string value given by user and puts it into the double array
36          for(int index = 0; index < xValues.length; index++)
37          {
38              x[index] = Double.parseDouble(xValues[index]);
39          }
```

Topic 6#: Classes/Objects

- Classes/Objects: in Tester.java:56
- Creates a new object of RegressionModelWithLabels class

```
56        regression1 = new RegressionModelWithLabels(x,y,type,xAxisLabel,yAxisLabel);
```

Topic #7: Static Variables and Methods

- Static Variables and Methods: in Math1.java:6
- Use of static of method mean so you can call on it without creating an object of the class Math1

```
6       public static double mean(double[] values)
7       {
8           double sum;
9           sum = 0.0;
10
11          for(int index = 0; index < values.length; index++)
12          {
13              sum += values[index];
14          }
15
16          return sum / values.length;
17      }
18
```

Topic #8: Math Class

- Math Class: in Math1.java:92
- Used Math.pow so that I could square a calculation

```java
84      public static double sumOfSquaresDueToRegression(double[] y)
85      {
86          double meanY;
87          meanY = mean(y);
88
89          double sumOfSquaredDeviations = 0.0;
90          for(int index = 0; index < y.length; index++)
91          {
92              sumOfSquaredDeviations += (Math.pow(y[index] - meanY, 2));
93          }
94
95          return sumOfSquaredDeviations;
96
97      }
```

Topic# 9: Wrapper Classes

- Wrapper Classes: in Test.java:38
- I used Double.parseDouble wrapper class method to convert my String x input into a double so that I can fill an array with the input

```
34          x = new double[xValues.length];
35          // parses the string value given by user and puts it into the double array
36          for(int index = 0; index < xValues.length; index++)
37          {
38              x[index] = Double.parseDouble(xValues[index]);
39          }
40          System.out.println("Enter x axis label:");
41          xAxisLabel = keyboard.nextLine();
```

Topic #10: References (deep copying)

- References (deep copying): in RegressionModel.java:44
- I used my copyArray class to make a deep copy to actually make a new array and copy over the values

```
42          else
43          {
44              this.xValues = CopyArray.copyDoubleArray(xValues, xValues.length);
45              this.yValues = CopyArray.copyDoubleArray(yValues, yValues.length);
46          }
47      }
```

Topic #11: Arrays

- Arrays: in RegressionModel.java:10
- Initialized array instance variable xValues

```
9   /*** INSTANCE VARIABLES ***/
10  private double [] xValues;
11  private double [] yValues;
12
```

Topic #12: Enumerated Types

- Enumerated Types: in RegressionModelWithLabels.java:3
- Enum types of NONE, LINEAR, or NONLINEAR

```
1   public class RegressionModelWithLabels extends RegressionModel {
2
3       public enum RegressionType {NONE, LINEAR, NONLINEAR}
4
```

Topic #13: Inheritance

- Inheritance: in RegressionModelWithLabels.java:1
- RegressionModelWithLabels(subclass) is derived from the RegressionModel(base class), therefore inheriting its methods.

Source History

```
1  public class RegressionModelWithLabels extends RegressionModel {
2
```

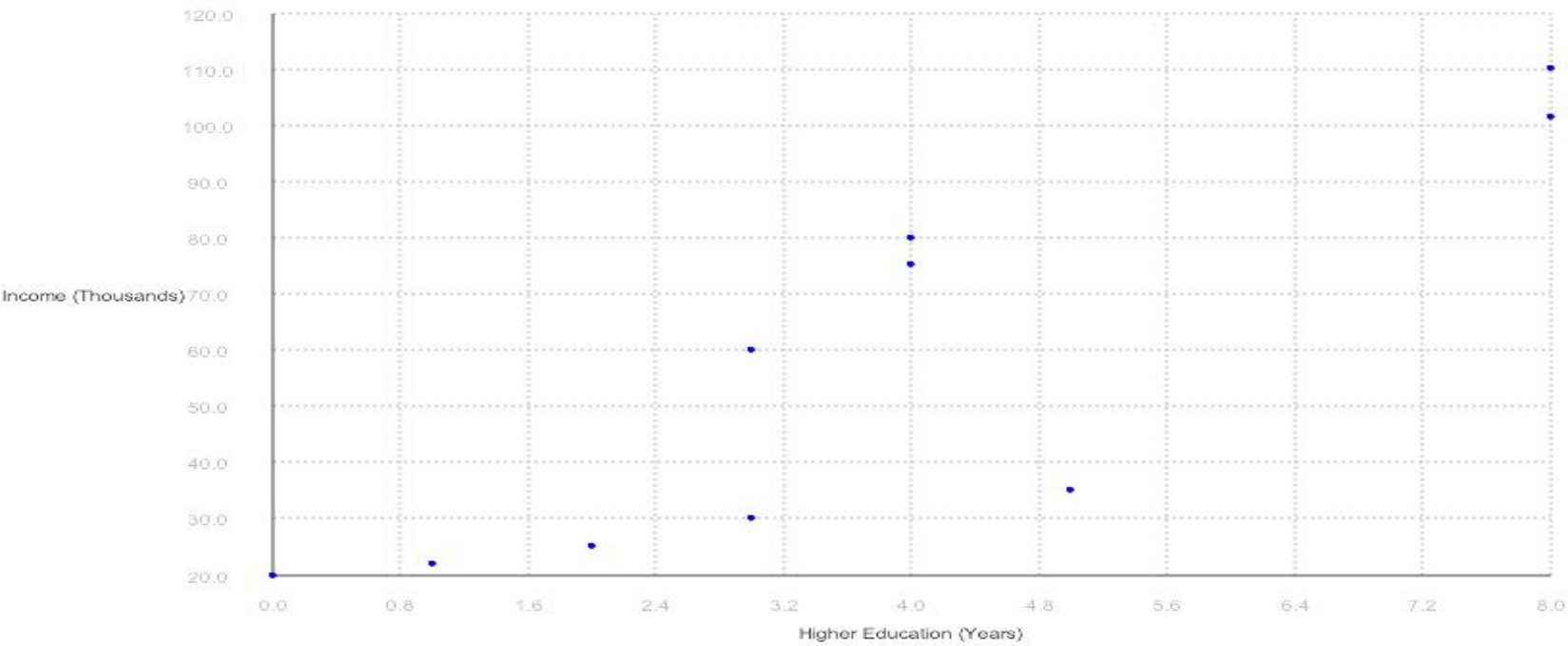
Topic #14 File Input/Output

- File Input/Output: In Test.java:84
- I save the image to my desktop as a jpeg file

```
72          // put the PlotPanel in a JFrame, as a JPanel
73          JFrame frame = new JFrame("Scatter Plot of X and Y arrays");
74          frame.setSize(1000, 1000);
75          frame.setContentPane(plot);
76          frame.setVisible(true);
77
78          // saves graph as jpeg to desktop
79          try
80          {
81              BufferedImage image = new BufferedImage(frame.getWidth(), frame.getHeight(), BufferedImage.TYPE_INT_RGB);
82              Graphics2D graphics2D = image.createGraphics();
83              frame.paint(graphics2D);
84              ImageIO.write(image,"jpeg", new File("/Users/tyler/Desktop/LinearRegression.jpeg"));
85          }
86          catch(IOException e)
87          {
88              System.out.println("Error saving image");
89              System.exit(0);
90
```

Topic #14: File Input/Output

- Was most interesting topic because I had to learn how to add and use a new library



Future Add-Ons

- I would like to add a feature that automatically parses through a CSV file and inputs it into the x and y arrays
- Also, I would like to have the linear formula that the program calculates graph a line through my scatter plot.