

## Person.java

```
1 import java.util.Scanner;
2
3
4
5
6 /**
7  * Person represents a single line from the file Experiment-Student.txt
8  *   trait: the first column (0 or 1)
9  *   without: the second column (without treatment)
10 *   with: the third column (with treatment)
11 *
12 * @author Nick Jones
13 * @version 12/3/2014
14 */
15 public class Person {
16     int index;
17     Double without;
18     Double with;
19     int trait;
20
21     /**
22      * Constructor
23      * @param index which line in the file is it?
24      * @param without result without treatment
25      * @param with result with treatment
26      * @param trait trait of person
27      */
28     public Person(int index, Double without, Double with, int trait) {
29         this.index = index;
30         this.without = without;
31         this.with = with;
32         this.trait = trait;
33     }
34
35
36     /**
37      * Get a string of this Person
38      * @return index: trait without with
39      */
40     @Override
41     public String toString() {
42         return String.format("%4d:  %1d  %3.8f  %3.8f", index, trait, without, with);
43     }
44
45
46     /**
47      * What is the mean of the values of the people in the array
48      * @param ar The array of people to find the mean of
49      * @param control Is this the control group?
50      * @return If control, the mean of without. Else, the mean of with.
51      */
52     public static Double mean(ArrayList<Person> ar, boolean control) {
53         Double result = 0.0;
54
55         if (control)
56             for (Person p : ar)
57                 result += p.without;
58         else
59             for (Person p : ar)
60                 result += p.with;
```

## Person.java

```
61
62     return result/ar.size();
63 }
64
65 /**
66  * What is the standard deviation of the values of the people in the array.
67  * @param ar The array of people
68  * @param control Is this the control group.
69  * @return If control, std(without). Else, std(with).
70  */
71 public static Double std(ArrayList<Person> ar, boolean control) {
72     Double result = 0.0;
73     Double mean = Person.mean(ar, control);
74
75     if (control)
76         for (Person p : ar)
77             result += Math.pow(p.without - mean, 2);
78     else
79         for (Person p : ar)
80             result += Math.pow(p.with - mean, 2);
81
82     return Math.sqrt(result/ar.size());
83 }
84
85 /**
86  * What percent of the Person have a trait of zero?
87  * @param ar The array of people
88  * @return What percent of people have zero as their trait. [0.0, 100.0]
89  */
90 public static Double percentZero(ArrayList<Person> ar) {
91     Double result = 0.0;
92
93     for(Person p : ar)
94         result += (p.trait + 1) % 2;
95
96     return result / ar.size() * 100;
97 }
98
99 /**
100  * Main Method
101  *
102  * Read in Experiment-Student.txt and perform statistical analysis on it.
103  *
104  * @param args UNUSED
105  */
106 public static void main(String[] args) {
107     int numControl = 20; // Number of people in control group
108     int numTest = 20; // Number of people in treatment group.
109
110     assert(numControl + numTest <= 1000);
111
112     try {
113         Scanner scnr = new Scanner(new File("Experiment-Student.txt"));
114
115         ArrayList<Person> ar = new ArrayList<Person>(); // All people in file
116         ArrayList<Person> control = new ArrayList<Person>(); // Control group
117         ArrayList<Person> test = new ArrayList<Person>(); // Treatment group
```

Person.java

```
118
119     int i = 1;
120
121     while(scnr.hasNextLine()) {
122         String line = scnر.nextLine();
123         Scanner lScnr = new Scanner(line); // scans the line
124
125         // PARSE LINE FROM FILE
126         int trait = lScnr.nextInt();
127         Double without = lScnr.nextDouble();
128         Double with = lScnr.nextDouble();
129         Person p = new Person(i++, without, with, trait);
130
131         // ADD TO ARRAY OF ALL PEOPLE
132         ar.add(p);
133     }
134
135
136     // SELECT PEOPLE FOR CONTROL GROUP
137     for (i=0; i<numControl; i++) {
138         int rand = (int)(ar.size() * Math.random());
139         control.add(ar.get(rand));
140         ar.remove(rand);
141     }
142
143     // SELECT PEOPLE FOR TREATMENT GROUP
144     for (i=0; i<numTest; i++) {
145         int rand = (int)(ar.size() * Math.random());
146         test.add(ar.get(rand));
147         ar.remove(rand);
148     }
149
150     // DISPLAY CONTROL PEOPLE
151     System.out.println("\nControl:\n-----");
152     i=1;
153     for(Person p : control)
154         System.out.printf("%2d: %s\n", i++, p);
155
156     // DISPLAY TREATMENT PEOPLE
157     System.out.println("\nTest:\n-----");
158     i=1;
159     for(Person p : test)
160         System.out.printf("%2d: %s\n", i++, p);
161
162     // CALCULATE CONTROL STATS
163     Double controlMean = Person.mean(control, true);
164     Double controlSTD = Person.std(control, true);
165     Double controlZeroPercent = Person.percentZero(control);
166
167     // CALCULATE TREATMENT STATS
168     Double testMean = Person.mean(test, false);
169     Double testSTD = Person.std(test, false);
170     Double testZeroPercent = Person.percentZero(test);
171
172     // FIND TOTAL PERCENT ZERO
173     ArrayList<Person> all = new ArrayList<Person>(control);
174     all.addAll(test);
```

Person.java

```
175         Double totalZeroPercent = Person.percentZero(all);
176
177         // DISPLAY CONTROL STATS
178         System.out.println("\nControl:\n-----");
179         System.out.println("mean: " + controlMean);
180         System.out.println("std: " + controlSTD);
181         System.out.println("%0: " + controlZeroPercent);
182
183         // DISPLAY TREATMENT STATS
184         System.out.println("\nTest:\n-----");
185         System.out.println("mean: " + testMean);
186         System.out.println("std: " + testSTD);
187         System.out.println("%0: " + testZeroPercent);
188
189         // DISPLAY TOTAL PERCENT ZERO
190         System.out.println("\nTotal %0:\n-----");
191         System.out.println("" + totalZeroPercent);
192
193         // CALCULATE CONCLUSION
194         Double ts = (testMean - controlMean) /
195                     Math.sqrt(Math.pow(testSTD,2)/test.size() +
196                               Math.pow(controlSTD,2)/control.size());
197         Double cv = 1.96;
198         Boolean concl = Math.abs(ts) > cv;
199
200         // DISPLAY CONCLUSION
201         System.out.println("\nResult:\n-----");
202         System.out.printf("|%f| %s %f => %s H0",
203                             ts,
204                             concl ? ">" : "<=",
205                             cv,
206                             concl ? "reject" : "fail to reject");
207
208         // CLOSE SCANNER
209         scnr.close();
210     }
211     catch (FileNotFoundException e) { System.out.println("File not found");
212         System.exit(1); }
213 }
214
```