Logic Specification Template

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Hector Manuel Takami Flores | Date | 07-04-18 |
| Program | 7 | Program # | 7 |
| Instructor | Adriana Bojorquez | Language | Java |

|  |  |
| --- | --- |
| **Design** | OST 1, FST1 |
| **References** |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parameters** | Main |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| System.out.println("r:"+r);  System.out.println("r^2:"+r\*r);  System.out.println("tailArea:"+getTailArea(r, xArrayList));  System.out.println("beta0:"+beta0);  System.out.println("beta1:"+beta1);  System.out.println("yk:"+y);  System.out.println("Range:"+range);  System.out.println("UPI:"+(y+range));  System.out.println("LPI:"+(y-range)); |

|  |  |
| --- | --- |
| **Class Name** | Correlacion |

|  |  |
| --- | --- |
| **Design** | OST 1, FST1 |
| **References** |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Method Name** | calR |

|  |  |
| --- | --- |
| **Parameters** | double[][] matrix |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| double cal = 0;  cal = xArrayList.size()\*multiple(xArrayList, yArrayList)-multiple(xArrayList)\*multiple(yArrayList);  double minus = 0;  minus = xArrayList.size()\*multiple(xArrayList, xArrayList)-multiple(xArrayList)\*multiple(xArrayList);  minus \*= yArrayList.size()\*multiple(yArrayList, yArrayList)-multiple(yArrayList)\*multiple(yArrayList);  cal = cal / Math.sqrt(minus);  return cal; |

|  |  |
| --- | --- |
| **Class Name** | Correlacion |

|  |  |
| --- | --- |
| **Design** | OST 1, FST1 |
| **References** |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Method Name** | getTailArea |

|  |  |
| --- | --- |
| **Parameters** | (double r, ArrayList<Double> arrayList |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |
| double x = Math.abs(r) \* Math.sqrt(arrayList.size() - 2) / Math.sqrt(1 - r \* r);  return 1 - 2 \* cal(x, arrayList.size() - 2); |

|  |  |
| --- | --- |
| **Class Name** | Correlacion |

|  |  |
| --- | --- |
| **Design** | OST 1, FST1 |
| **References** |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Method Name** | sigma |

|  |  |
| --- | --- |
| **Parameters** | ArrayList<Double> xArrayList, ArrayList<Double> yArrayList,double b0, double b1 |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| ArrayList<Double> cal = new ArrayList<Double>();  for (int i = 0; i < xArrayList.size(); i++){  cal.add(yArrayList.get(i) - b0 - b1 \* xArrayList.get(i));  }  return Math.sqrt(multiple(cal,cal) / (xArrayList.size() - 2)); | | |
| **Class Name** | | Correlacion |

|  |  |
| --- | --- |
| **Design** | OST 1, FST1 |
| **References** |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Method Name** | Range |

|  |  |
| --- | --- |
| **Parameters** | A rrayList<Double> xArrayList, ArrayList<Double> yArrayList,double b0, double b1,double xk |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| ouble range = getX(0.35, xArrayList.size() - 2) \* getSigma(xArrayList, yArrayList, b0, b1);  ArrayList<Double> cal = new ArrayList<Double>();  for (int i = 0; i < xArrayList.size(); i++){  cal.add(xArrayList.get(i) - getAverage(xArrayList));  }  return range \* Math.sqrt( 1 + (1.0 / xArrayList.size()) + ((xk - getAverage(xArrayList)) \* (xk - getAverage(xArrayList)) / multiple(cal,cal)));  } |