Radioactive Contamination in Hanford, WA

By: Derek Ng

Introduction

In an article taken from the Journal of Environmental Health, May-June 1965, Volume 27, Number 6, pages 883-897, author Robert Fadely explains that the Atomic Energy Plant in Hanford, Washington has been a plutonium production facility since the Second World War. Some of the waste have been stored underground in the same area. Radioactive waste has been seeping into the Columbia River, and Oregon counties and the city of Portland have been exposed to radioactive contamination. The table below lists the number of cancer deaths per 100,000 residents for Portland and these counties. The table also includes an index of exposure that measures the proximity of the residents to the contamination. The index is base on the assumption that the city or county exposure is directly proportional to river frontage and inversely proportional both to the distance from Hanford, WA site and to the square of the county's or city's average distance from the river.

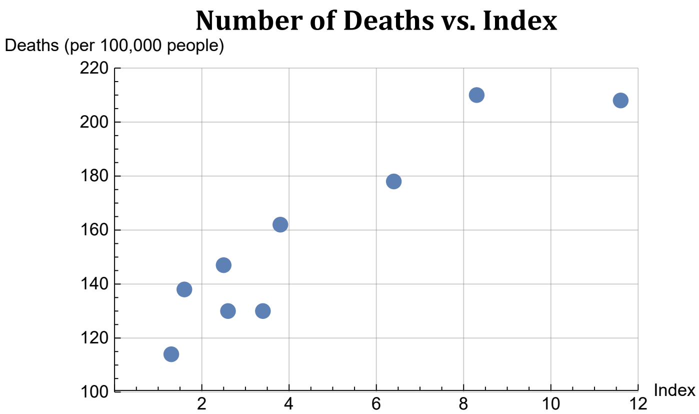
Raw Data

```
locations = List["Umatilla", "Morrow", "Gilliam",
   "Sherman", "Wasco", "Hood River", "Portland", "Columbia", "Clatsop"];

index = List[2.5, 2.6, 3.4, 1.3, 1.6, 3.8, 11.6, 6.4, 8.3];

deaths = List[147, 130, 130, 114, 138, 162, 208, 178, 210];

Grid[MapThread[Prepend, {{locations, index, deaths},
   {"Location", "Index", "Deaths (per 100,000 people)"}}], Frame → All]
```

Location	Umatilla	Morrow	Gilliam	Sherman	Wasco	Hood River	Portland	Columbia	Clatsop
Index	2.5	2.6	3.4	1.3	1.6	3.8	11.6	6.4	8.3
Deaths (per 100,000 people)	147	130	130	114	138	162	208	178	210

```
orderPairs = Transpose[{index, deaths}];

sortedPairs = Sort[orderPairs];
```

```
rawPlot = ListPlot[orderPairs, AxesLabel → {"Index", "Deaths (per 100,000 people)"},
  PlotStyle → {PointSize[0.03`]}, GridLines → Automatic,
  PlotLabel → Text[Style["Number of Deaths vs. Index", FontSize → 15,
     FontWeight → "Bold", FontFamily → "Cambria", FontColor → Black]],
  PlotRange → {{0, 12}, {100, 220}}, ImageSize → Medium]
```



The graph shows the raw data points.

Two-Point Line

To find a two-point line, I first chose two points that best represented the data: (1.6, 138) and (6.4, 178). I then found the slope of the two points.

```
twoPointSlope = (178 - 138) / (6.4 - 1.6);
```

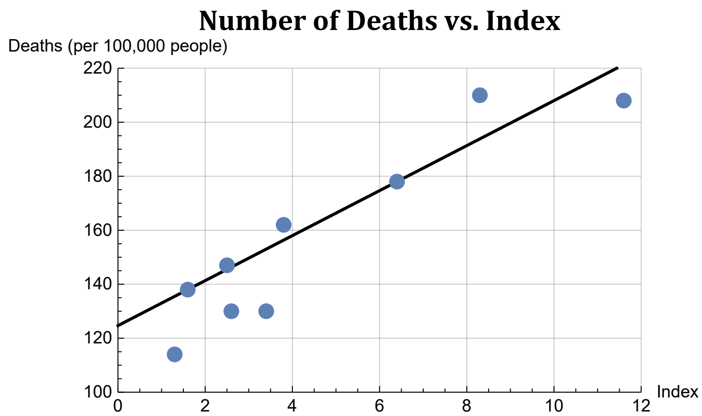
Using the slope and Point (6.4, 178), I found the equation of the line.

```
twoPointLineEq = twoPointSlope * x - twoPointSlope * 6.4 + 178
```

$124.667 + 8.33333\, x$

```
twoPointPlot = Plot[twoPointLineEq, {x, 0, 12}, PlotStyle → {RGBColor[0, 0, 0]},
   GridLines → Automatic, PlotRange → {{0, 12}, {100, 220}}, ImageSize → Medium];
```

```
Show[twoPointPlot, rawPlot,
 PlotLabel → Text[Style["Number of Deaths vs. Index", FontSize → 15,
    FontWeight → "Bold", FontFamily → "Cambria", FontColor → Black]],
 AxesLabel → {"Index", "Deaths (per 100,000 people)"}]
```

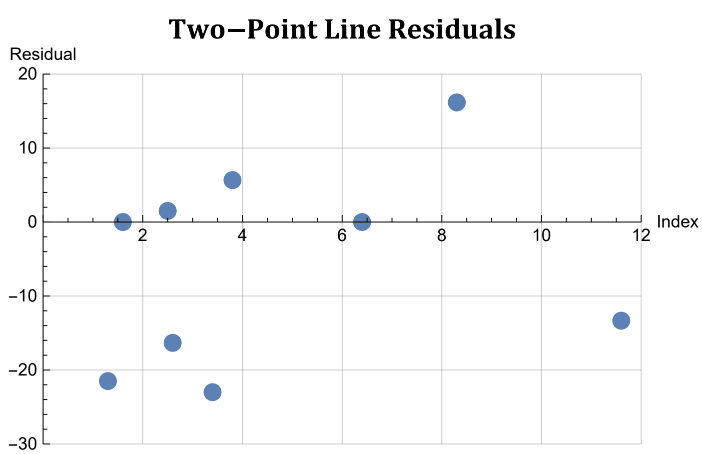


The graph shows the raw data points in addition to the two-point line.

Two-Point Line Residuals

To find the residuals for the two-point line, I subtracted the predicted value from the actual value.

```
twoPointResidual = List[];

For[i = 1, i ≤ 9, i++, AppendTo[twoPointResidual, List[sortedPairs[[i, 1]],
    (sortedPairs[[i, 2]]) - (twoPointLineEq /. x → sortedPairs[[i, 1]])]]];

twoPoint = ListPlot[twoPointResidual, AxesLabel → {"Index", "Residual"},
  PlotStyle → {PointSize[0.03`]}, GridLines → Automatic,
  PlotLabel → Text[Style["Two-Point Line Residuals", FontSize → 15,
     FontWeight → "Bold", FontFamily → "Cambria", FontColor → Black]],
  PlotRange → {{0, 12}, {-30, 20}}, ImageSize → Medium]
```



This graph shows the residual values of the predicted data compared to the actual data for the two-point line.

Sum of two-point line residuals:

```
Total[twoPointResidual[[All, 2]]]
```

−50.8333

Median-Median Line

Summary Points

To find the summary points, I divided the nine points by x-value into three different groups. The points with the lowest three x-values, the middle three x-values, and the highest three x-values were grouped together. I then used the middle point of each group as the summary point.

```
medianLeft = Median[sortedPairs[[{1, 2, 3}]]];
medianCenter = Median[sortedPairs[[{4, 5, 6}]]];
medianRight = Median[sortedPairs[[{7, 8, 9}]]];
```

Median-Median Line Equation Method 1

In my first method to find the median-median line, I first found the slope of the left and right summary point.

```
slopeLR = (medianRight[[2]] - medianLeft[[2]]) / (medianRight[[1]] - medianLeft[[1]]);
```

I then found the y-intercepts for each of the summary points using the point itself and the slope of the left and right summary point.

```
yIntLeft = medianLeft[[2]] - slopeLR * medianLeft[[1]];
yIntCenter = medianCenter[[2]] - slopeLR * medianCenter[[1]];
yIntRight = medianRight[[2]] - slopeLR * medianRight[[1]];
```

To find the intercept of the median-median line, I took the average of the three summary point intercepts.

```
medMedIntercept = (yIntLeft + yIntCenter + yIntRight) / 3;
```

Using the average intercept and the slop, I was able to form my equation for the median-median line.

```
medMedLineEq = slopeLR * x + medMedIntercept
```

112.348 + 10.4478 x

Median-Median Line Equation Method 2

In my second method to find the median-median line, I first found the average x and y coordinate of the three summary points.

```
avgX = (medianLeft[[1]] + medianCenter[[1]] + medianRight[[1]]) / 3;

avgY = (medianLeft[[2]] + medianCenter[[2]] + medianRight[[2]]) / 3;
```

I also found the slope of the left and right summary points.

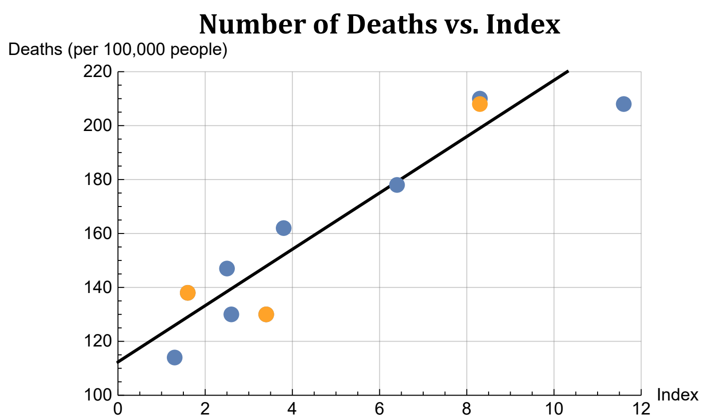
```
slopeLR = (medianRight[[2]] - medianLeft[[2]]) / (medianRight[[1]] - medianLeft[[1]]);
```

Using the average x and y coordinate and the slope, I was able to form my equation for the median-median line.

```
medMedLineEq = slopeLR * x - slopeLR * avgX + avgY
```

112.348 + 10.4478 x

```
medMedLinePlot = Plot[medMedLineEq, {x, 0, 12}, PlotStyle → {RGBColor[0, 0, 0]},
   GridLines → Automatic, PlotRange → {{0, 12}, {100, 220}}, ImageSize → Medium];

medPointsPlot = ListPlot[{medianLeft, medianCenter, medianRight},
   PlotStyle → {RGBColor[1, 0.64, 0.16], PointSize[0.03`]},
   PlotRange → {{0, 12}, {100, 220}}, ImageSize → Medium];

Show[medMedLinePlot, rawPlot, medPointsPlot,
 PlotLabel → Text[Style["Number of Deaths vs. Index", FontSize → 15,
    FontWeight → "Bold", FontFamily → "Cambria", FontColor → Black]],
 AxesLabel → {"Index", "Deaths (per 100,000 people)"}]
```



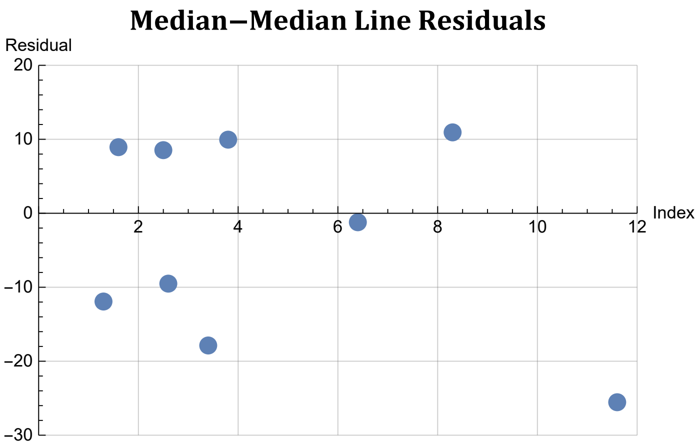
The blue dots show the raw data points, the orange dots show the three summary points, and the line shows the median-median line.

Median-Median Line Residuals

To find the residuals for the median-median line, I subtracted the predicted value from the actual value.

```
medMedResidual = List[];

For[i = 1, i ≤ 9, i++, AppendTo[medMedResidual, List[sortedPairs[[i, 1]],
    (sortedPairs[[i, 2]]) - (medMedLineEq /. x → sortedPairs[[i, 1]])]]];
```

```
medMed = ListPlot[medMedResidual, AxesLabel → {"Index", "Residual"},
  PlotStyle → {PointSize[0.03`]}, GridLines → Automatic,
  PlotLabel → Text[Style["Median-Median Line Residuals", FontSize → 15,
     FontWeight → "Bold", FontFamily → "Cambria", FontColor → Black]],
  PlotRange → {{0, 12}, {-30, 20}}, ImageSize → Medium]
```



This graph shows the residual values of the predicted data compared to the actual data for the median-median line.

Sum of the median-median residuals:

```
Total[medMedResidual[[All, 2]]]
```

−27.7164

Least Squares Line

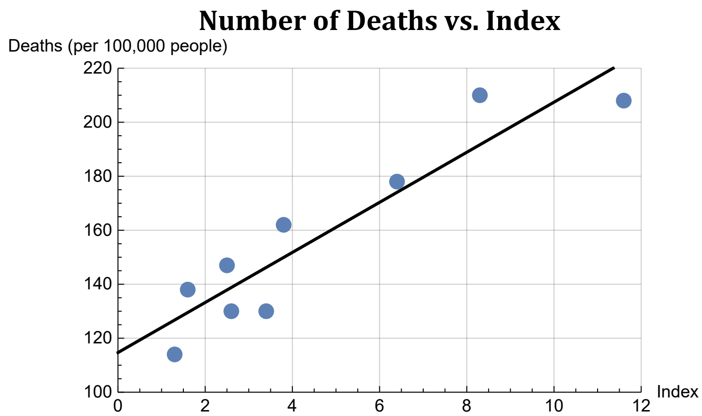
To find the equation for the least squares line, I used the fit function.

```
leastSquaresLineEq = Fit[orderPairs, {x, 1}, x]
```

114.682 + 9.27386 x

```
leastSqauresPlot = Plot[leastSquaresLineEq, {x, 0, 12}, PlotStyle → {RGBColor[0, 0, 0]},
   GridLines → Automatic, PlotRange → {{0, 12}, {100, 220}}, ImageSize → Medium];
```

```
Show[leastSqauresPlot, rawPlot,
 PlotLabel → Text[Style["Number of Deaths vs. Index", FontSize → 15,
    FontWeight → "Bold", FontFamily → "Cambria", FontColor → Black]],
 AxesLabel → {"Index", "Deaths (per 100,000 people)"}]
```

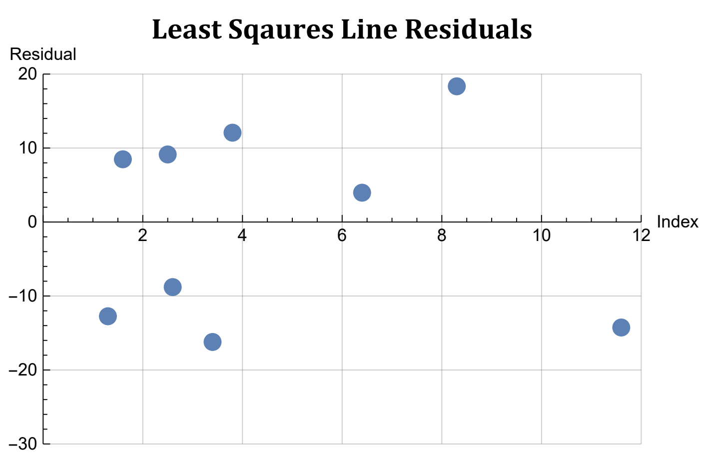


The graph shows the raw data points in addition to the least squares line.

Least Squares Line Residuals

To find the residuals for the least squares line, I subtracted the predicted value from the actual value.

```
leastSquaresResidual = List[];

For[i = 1, i ≤ 9, i++, AppendTo[leastSquaresResidual, List[sortedPairs[[i, 1]],
    (sortedPairs[[i, 2]]) - (leastSquaresLineEq /. x → sortedPairs[[i, 1]])]]];

leastSquares = ListPlot[leastSquaresResidual,
  AxesLabel → {"Index", "Residual"}, PlotStyle → {PointSize[0.03`]},
  GridLines → Automatic, PlotLabel → Text[Style["Least Sqaures Line Residuals",
     FontSize → 15, FontWeight → "Bold", FontFamily → "Cambria", FontColor → Black]],
  PlotRange → {{0, 12}, {-30, 20}}, ImageSize → Medium]
```



This graph shows the residual values of the predicted data compared to the actual data for the least squares line.

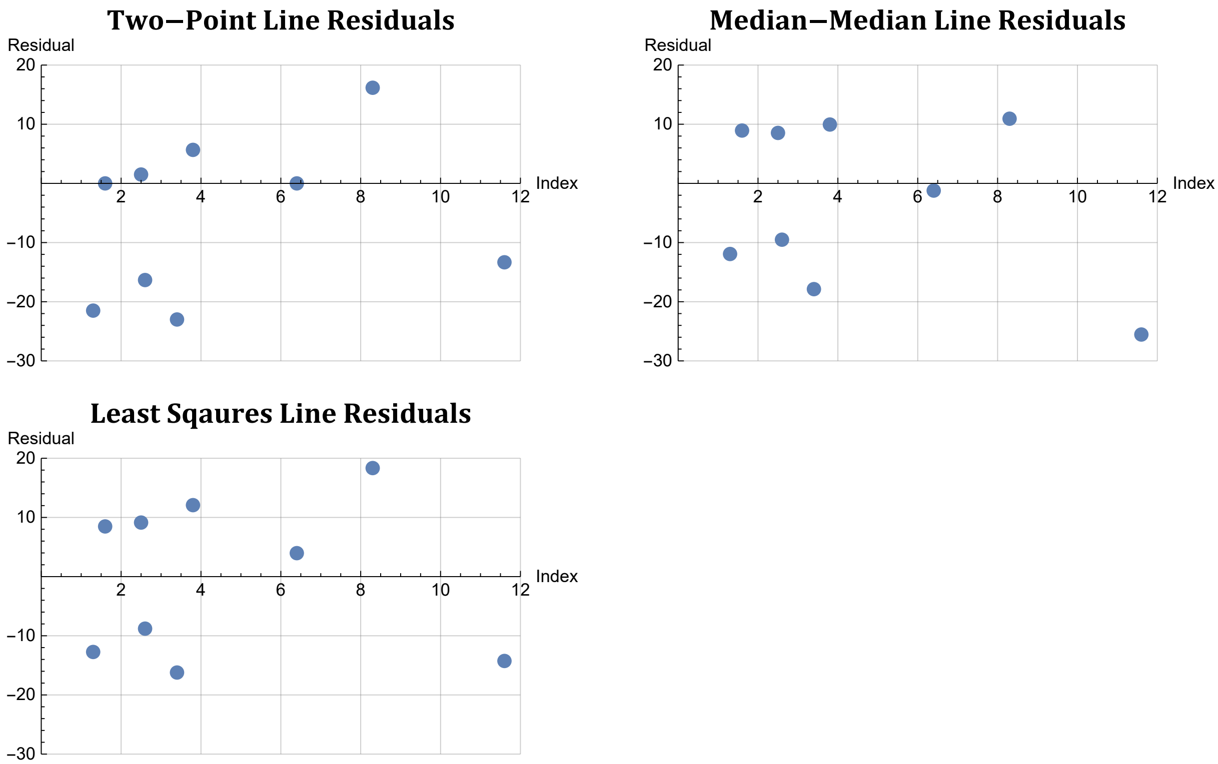
Sum of the least squares residuals:

```
Total[leastSquaresResidual[[All, 2]]]
```

-9.9476×10^{-14}

Residual Comparisons

```
GraphicsGrid[{{twoPoint, medMed}, {leastSquares}}, ImageSize → Full]
```



Comparison of the residual values for the three regression lines.