Functional Programming in *Mathematica*

A (very) brief tutorial...

Preliminaries

Functional programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids state and mutable data. It emphasizes the application of functions, in contrast with the imperative programming style that emphasizes changes in state.

General philosophy: Everything is a function

```
Clear[y];
y = Sqrt[4]
```

```
2
```

```
Clear[z];
z = Sqrt[x]
```

$\sqrt{x}$

```
z
```

$\sqrt{x}$

```
Clear[z];
```

```
z
```

```
z
```

```
??Sqrt
```

```
Sqrt[z] gives the square root of z. More...
```

```
Attributes[Sqrt] = {Listable, NumericFunction, Protected}
```

■ Function Composition

```
Sqrt[Sqrt[16]]
```

```
2
```

User-Defined Functions

■ Pure Functions

```
Function[z, z^2]
```

```
Function[z, z^2]
```

```
??Function
```

```
Function[body] or body& is a pure function. The formal parameters are # (or #1), #2,
   etc. Function[x, body] is a pure function with a single formal parameter x. Function[
   {x1, x2, ... }, body] is a pure function with a list of formal parameters. More...

Attributes[Function] = {HoldAll, Protected}
```

```
Function[z, z^2][5]
```

```
25
```

```
(#1^2 &)[5]
```

```
25
```

```
??#
```

```
# represents the first argument supplied
   to a pure function. #n represents the nth argument. More...

Attributes[Slot] = {NHoldAll, Protected}
```

■ Defining Complex Functions

```
compFunc[x_] := Sqrt[Sqrt[x]]
```

```
compFunc[16]
```

```
2
```

Lists

x = {4,1,7,1}

```
{4, 1, 7, 1}
```

y = List[1, 4, 2, 4, 2]

```
{1, 4, 2, 4, 2}
```

Sort[x]

```
{1, 1, 4, 7}
```

Sort[y]

```
{1, 2, 2, 4, 4}
```

??Sort

```
Sort[list] sorts the elements of list into canonical
   order. Sort[list, p] sorts using the ordering function p. More...

Attributes[Sort] = {Protected}
```

Sqrt[{1,2,3,4,5,6}]

$$\left\{1, \sqrt{2}, \sqrt{3}, 2, \sqrt{5}, \sqrt{6}\right\}$$

??Join

```
Join[list1, list2, ... ] concatenates lists together. Join
   can be used on any set of expressions that have the same head. More...

Attributes[Join] = {Flat, OneIdentity, Protected}
```

Join[x, y]

```
{4, 1, 7, 1, 1, 4, 2, 4, 2}
```

z = {x, y}

```
{{4, 1, 7, 1}, {1, 4, 2, 4, 2}}
```

z // TableForm

```
4      1      7      1
1      4      2      4      2
```

z // MatrixForm

$$\begin{pmatrix} \{4, 1, 7, 1\} \\ \{1, 4, 2, 4, 2\} \end{pmatrix}$$

```
Function[z, z^2][{1, 2, 3}]
```

```
{1, 4, 9}
```

```
compFunc[{16, 25}]
```

$\{2, \sqrt{5}\}$

Lets try to do Assignment 3

```
SetDirectory["c:\\data"]
```

```
c:\data
```

```
Application = "Word"
```

```
Word
```

```
depthArray = << ("FaultDepths_Terp" <> Application <> ".txt")
```

```
{1, 1, 3, 2, 1, 4, 1, 2, 2, 3, 2, 2, 1, 1, 2, 3, 1, 2, 1, 2, 4, 3, 2, 1, 2, 5, 2, 1, 4, 1, 2,
 5, 2, 2, 4, 1, 3, 1, 1, 2, 1, 1, 2, 3, 3, 2, 2, 3, 1, 1, 5, 3, 1, 5, 1, 4, 2, 5, 3, 1,
 2, 2, 1, 1, 3, 3, 1, 3, 4, 4, 1, 3, 4, 2, 2, 3, 2, 5, 1, 2, 2, 3, 2, 2, 1, 1, 3, 1,
 3, 2, 3, 1, 1, 3, 1, 1, 1, 4, 3, 1, 3, 1, 2, 2, 3, 1, 1, 4, 5, 2, 2, 2, 2, 2, 1, 2,
 4, 5, 2, 1, 1, 1, 4, 1, 5, 4, 2, 3, 4, 4, 1, 1, 3, 3, 3, 2, 1, 2, 2, 4, 3, 1, 1, 1,
 2, 1, 2, 5, 1, 3, 3, 1, 1, 3, 3, 1, 2, 1, 2, 2, 3, 2, 1, 3, 2, 5, 1, 4, 5, 1, 2, 3,
 2, 4, 1, 2, 2, 1, 4, 4, 1, 2, 1, 2, 1, 5, 3, 5, 1, 2, 4, 3, 1, 2, 4, 1, 3, 2, 1, 2}
```

```
depths = Union[depthArray]
```

```
{1, 2, 3, 4, 5}
```

```
totalPerDepth = Map[Length[Cases[depthArray, #]] &, depths]
```

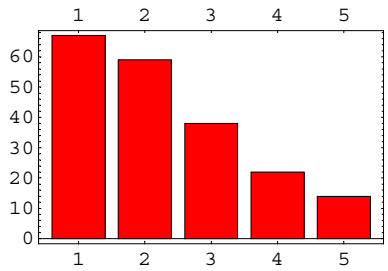
```
{67, 59, 38, 22, 14}
```

```
Apply[Plus, totalPerDepth]
```

```
200
```

```
<<Graphics`Graphics`
```

```
gr = BarChart[N[totalPerDepth], Frame → True, ImageSize → {200, Automatic}]
```




```
- Graphics -
```

Smoke Tests

```
smokeMatrix = << ("Smoke_Terp" <> Application <> ".txt");
```

```
Length[smokeMatrix]
```

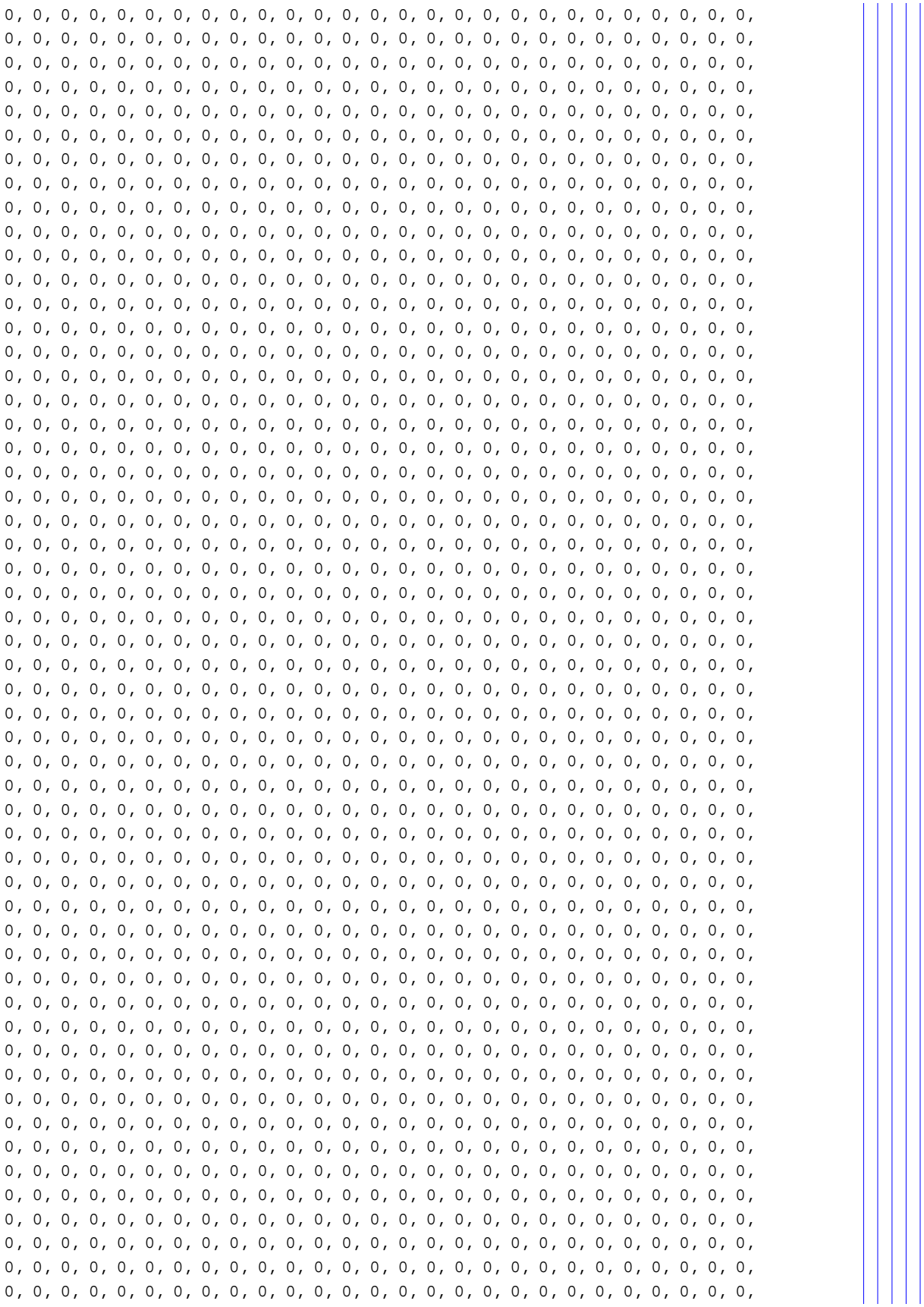
```
6782
```

```
Length[Transpose[smokeMatrix]]
```

```
200
```

```
Transpose[smokeMatrix][[1]]
```

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```



```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

```
Apply[Plus,Transpose[smokeMatrix][[1]]]
```

```
1
```

```
Map[Apply[Plus, #]&, Transpose[smokeMatrix]]
```

```
{1, 2, 0, 1, 1, 0, 1, 1, 1, 1, 2, 0, 1, 2, 2, 0, 2, 1, 1, 1, 2, 2, 2, 1, 2, 2, 1, 2, 0, 1,
 1, 0, 1, 0, 0, 2, 1, 1, 2, 2, 2, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 0, 1, 2, 1, 1, 1, 2, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 2, 1, 1, 3, 1, 1, 1, 1, 2,
 2, 2, 0, 2, 1, 1, 2, 1, 1, 1, 1, 2, 3, 1, 2, 0, 0, 1, 0, 2, 0, 1, 1, 2, 1, 0, 2, 2,
 0, 0, 1, 1, 1, 1, 1, 1, 2, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 2, 1, 1, 1,
 2, 2, 1, 1, 2, 1, 1, 1, 1, 2, 2, 1, 0, 1, 2, 2, 1, 2, 1, 3, 1, 0, 2, 0, 0, 2, 2, 1,
 2, 1, 0, 2, 1, 1, 0, 1, 1, 3, 1, 1, 0, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 0, 1}
```

```
??Position
```

```
Position[expr, pattern] gives a list of the positions at which objects
   matching pattern appear in expr. Position[expr, pattern, levspec] finds
   only objects that appear on levels specified by levspec. Position[expr,
   pattern, levspec, n] gives the positions of the first n objects found. More...

Attributes[Position] = {Protected}

Options[Position] = {Heads → True}
```

```
zeros = Position[Map[Apply[Plus, #]&, Transpose[smokeMatrix]], 0]
```

```
{{3}, {6}, {12}, {16}, {29}, {32}, {34}, {35}, {52}, {72}, {78},
 {91}, {104}, {105}, {107}, {109}, {114}, {117}, {118}, {126}, {129},
 {134}, {140}, {157}, {166}, {168}, {169}, {175}, {179}, {185}, {199}}
```

```
Range[Length[Transpose[smokeMatrix]]]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78,
 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112,
 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126,
 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140,
 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170,
 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185,
 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200}
```

```
??Delete
```

```
Delete[expr, n] deletes the element at position n in expr. If
   n is negative, the position is counted from the end. Delete[expr, {i,
   j, ... }] deletes the part at position {i, j, ... }. Delete[expr, {{i1,
   j1, ... }, {i2, j2, ... }, ... }] deletes parts at several positions. More...

Attributes[Delete] = {Protected}
```

```
smokeFaultsFound = Delete[Range[Length[Transpose[smokeMatrix]]], zeros]
```

```
{1, 2, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24,
 25, 26, 27, 28, 30, 31, 33, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
 48, 49, 50, 51, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
 68, 69, 70, 71, 73, 74, 75, 76, 77, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88,
 89, 90, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 106, 108, 110,
 111, 112, 113, 115, 116, 119, 120, 121, 122, 123, 124, 125, 127, 128, 130,
 131, 132, 133, 135, 136, 137, 138, 139, 141, 142, 143, 144, 145, 146, 147,
 148, 149, 150, 151, 152, 153, 154, 155, 156, 158, 159, 160, 161, 162, 163,
 164, 165, 167, 170, 171, 172, 173, 174, 176, 177, 178, 180, 181, 182, 183,
 184, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 200}
```

```
Length[Delete[Range[Length[Transpose[smokeMatrix]]], zeros]]
```

```
169
```

```
smokeDepths = Map[{1, #} &, depthArray[[smokeFaultsFound]]]
```

```
{{1, 1}, {1, 1}, {1, 2}, {1, 1}, {1, 1}, {1, 2}, {1, 2}, {1, 3}, {1, 2}, {1, 1},
 {1, 1}, {1, 2}, {1, 1}, {1, 2}, {1, 1}, {1, 2}, {1, 4}, {1, 3}, {1, 2}, {1, 1},
 {1, 2}, {1, 5}, {1, 2}, {1, 1}, {1, 1}, {1, 2}, {1, 2}, {1, 1}, {1, 3}, {1, 1},
 {1, 1}, {1, 2}, {1, 1}, {1, 1}, {1, 2}, {1, 3}, {1, 3}, {1, 2}, {1, 2}, {1, 3},
 {1, 1}, {1, 1}, {1, 5}, {1, 1}, {1, 5}, {1, 1}, {1, 4}, {1, 2}, {1, 5}, {1, 3},
 {1, 1}, {1, 2}, {1, 2}, {1, 1}, {1, 1}, {1, 3}, {1, 3}, {1, 1}, {1, 3}, {1, 4},
 {1, 4}, {1, 1}, {1, 4}, {1, 2}, {1, 2}, {1, 3}, {1, 2}, {1, 1}, {1, 2}, {1, 2},
 {1, 3}, {1, 2}, {1, 2}, {1, 1}, {1, 1}, {1, 3}, {1, 1}, {1, 3}, {1, 2}, {1, 1},
 {1, 1}, {1, 3}, {1, 1}, {1, 1}, {1, 1}, {1, 4}, {1, 3}, {1, 1}, {1, 3}, {1, 1},
 {1, 2}, {1, 1}, {1, 4}, {1, 2}, {1, 2}, {1, 2}, {1, 2}, {1, 1}, {1, 2}, {1, 2},
 {1, 1}, {1, 1}, {1, 1}, {1, 4}, {1, 1}, {1, 5}, {1, 2}, {1, 3}, {1, 4}, {1, 1},
 {1, 1}, {1, 3}, {1, 3}, {1, 2}, {1, 1}, {1, 2}, {1, 2}, {1, 3}, {1, 1}, {1, 1},
 {1, 1}, {1, 2}, {1, 1}, {1, 2}, {1, 5}, {1, 1}, {1, 3}, {1, 3}, {1, 1}, {1, 1},
 {1, 3}, {1, 3}, {1, 1}, {1, 1}, {1, 2}, {1, 2}, {1, 3}, {1, 2}, {1, 1}, {1, 3},
 {1, 2}, {1, 1}, {1, 1}, {1, 2}, {1, 3}, {1, 2}, {1, 4}, {1, 2}, {1, 2}, {1, 1},
 {1, 4}, {1, 1}, {1, 2}, {1, 1}, {1, 2}, {1, 5}, {1, 3}, {1, 5}, {1, 1}, {1, 2},
 {1, 4}, {1, 3}, {1, 1}, {1, 2}, {1, 4}, {1, 1}, {1, 3}, {1, 2}, {1, 2}}
```

```
depths
```

```
{1, 2, 3, 4, 5}
```

```
smokePerDepth = Map[Length[Cases[depthArray[[smokeFaultsFound]], #]] &, depths]
```

```
{63, 54, 31, 13, 8}
```

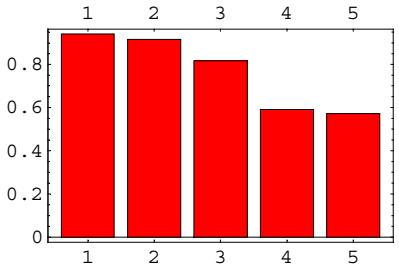
```
Apply[Plus, smokePerDepth]
```

```
169
```

```
N[smokePerDepth / totalPerDepth]
```

```
{0.940299, 0.915254, 0.815789, 0.590909, 0.571429}
```

```
gr = BarChart[N[smokePerDepth / totalPerDepth],
   Frame → True, ImageSize → {200, Automatic}];
```



Comp Tests

Putting it all Together

■ Initialization

```
<<Graphics`Graphics`
```

```
SetDirectory["c:\\data"]
```

```
c:\data
```

■ Parameters

```
type="Comp"
```

```
Comp
```

```
Application="Word"
```

```
Word
```

■ Computation

```
depthArray = << ("FaultDepths_Terp" <> Application <> ".txt");
```

```
depths = Union[depthArray];
```

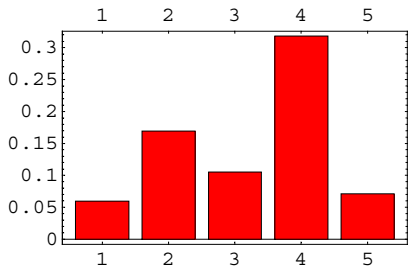
```
totalPerDepth= Map[Length[Cases[depthArray, #]]&, depths];
```

```
matrix = << (type <> "_Terp" <> Application <> ".txt");
```

```
zeros = Position[Map[Apply[Plus, #]&, Transpose[matrix]], 0];
```

```
faultsFound = Delete[Range[Length[Transpose[matrix]]], zeros];
```

```
foundDepths = Map[{1, #} &, depthArray[[faultsFound]]];

faultsPerDepth = Map[Length[Cases[depthArray[[faultsFound]], #]] &, depths];

gr = BarChart[N[faultsPerDepth / totalPerDepth],
  Frame → True, ImageSize → {200, Automatic}]
```




```
- Graphics -
```

Putting it in One Function

■ Initialization

```
<<Graphics`Graphics`

SetDirectory["c:\\data"]
```

```
c:\data
```

■ Parameters

```
type="Comp"
```

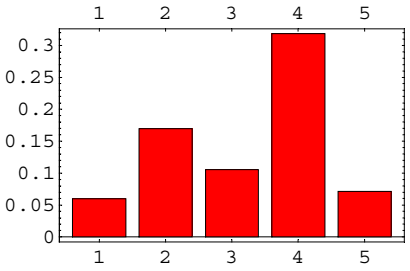
```
Comp
```

```
Application="Word"
```

```
Word
```

■ Computation

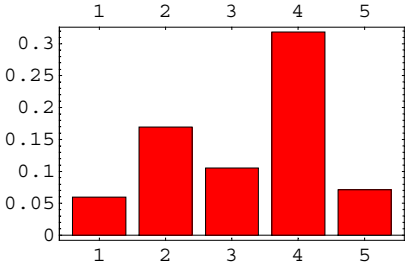
```
Clear[plotHistogram];
plotHistogram[type_, Application_]:=
Module[{},
 depthArray = Get[("FaultDepths_Terp"<>Application<>".txt")];
 depths=Union[depthArray];
 totalPerDepth= Map[Length[Cases[depthArray, #]]&, depths];
 matrix = Get[(type <> "_Terp"<>Application<>".txt")];
 zeros = Position[Map[Apply[Plus, #]&, Transpose[matrix]], 0];
 faultsFound = Delete[Range[Length[Transpose[matrix]]], zeros];
 foundDepths=Map[{1, #}&, depthArray[[faultsFound]]];
 faultsPerDepth=Map[Length[Cases[depthArray[[faultsFound]], #]]&, depths];
 BarChart[N[faultsPerDepth/totalPerDepth], Frame→True,
    ImageSize→{200, Automatic}]]

plotHistogram["Comp", "Word"]
```

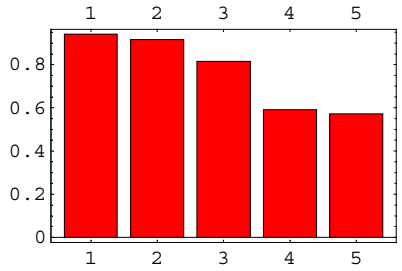


- Graphics -

Invoking the Function

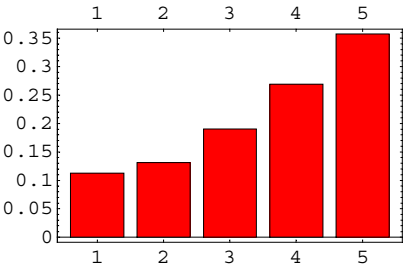
```
Map[plotHistogram[#[[1]], #[[2]]]&, {{"Comp","Word"}, {"Smoke", "Word"}}]
```

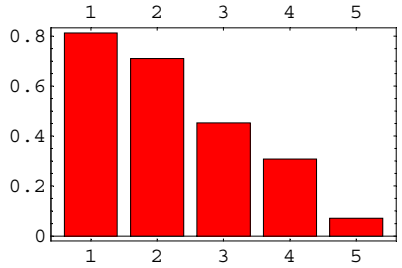




```
{- Graphics -, - Graphics -}
```

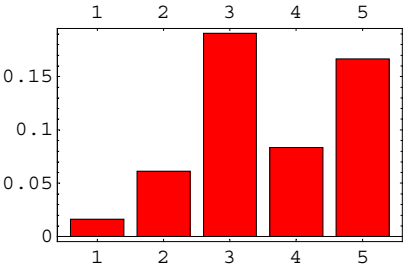
```
Map[plotHistogram[#[[1]], #[[2]]]&, {{"Comp","Paint"}, {"Smoke", "Paint"}}]
```

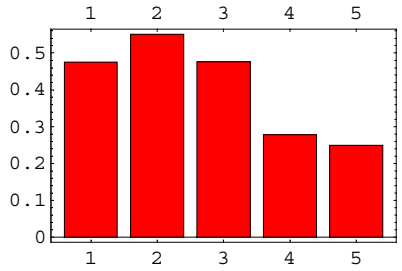




```
{- Graphics -, - Graphics -}
```

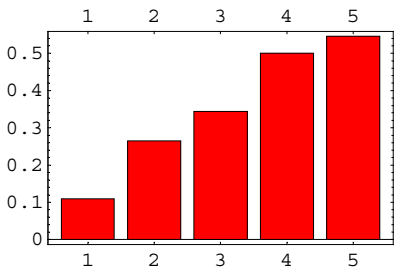
```
Map[plotHistogram[#[[1]], #[[2]]]&, {{"Comp","Present"}, {"Smoke",
"Present"}}]
```

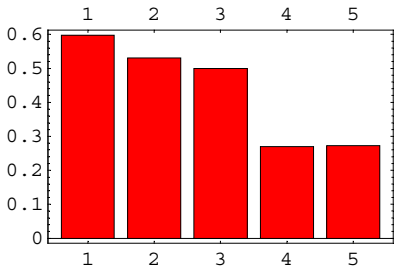




```
{- Graphics -, - Graphics -}
```

```
Map[plotHistogram[#[[1]], #[[2]]]&, {{"Comp","SpreadSheet"}, {"Smoke",
"SpreadSheet"}}]
```






```
{- Graphics -, - Graphics -}
```