

Pareto Charts

Objective

To create a Pareto Chart in R from a collection of numbers that represent *frequencies* (or *counts*) of events or outcomes that fall into different categories.

Background

A Pareto Chart is a sorted bar chart that displays the *frequency* (or *count*) of occurrences that fall in different categories, from greatest frequency on the left to least frequency on the right, with an overlaid line chart that plots the *cumulative percentage* of occurrences. The vertical axis on the left of the chart shows *frequency* (or *count*), and the vertical axis on the right of the chart shows the *cumulative percentage*. A Pareto Chart is typically used to visualize:

- Primary types or sources of defects
- Most frequent reasons for customer complaints
- Amount of some variable (e.g. money, energy usage, time) that can be attributed to or classified according to a certain category

The Pareto Chart is typically used to separate the “vital few” from the “trivial many” using the Pareto principle, also called the 80/20 Rule, which asserts that approximately 80% of effects come from 20% of causes for many systems. Pareto analysis can thus be used to find, for example, the most critical types or sources of defects, the most common complaints that customers have, or the most essential categories within which to focus problem-solving efforts.

Data Format

To create a Pareto Chart, all you need is a vector (or array) of numbers. Typically, this vector will contain *counts* of the defects (or causes for a certain outcome) in different categories. Here is an example of data generated by surveying 50 people to ask “What are the top 2 reasons you are late to work?” The available answers were 1) bad

weather, 2) I overslept, 3) my alarm didn't go off, 4) confused by the time change to/from Daylight Savings Time, 5) traffic was bad, and 6) other.

You can work with your data as a vector:

```
defect.counts <- c(12,29,18,3,34,4)

names(defect.counts) <- c("Weather", "Overslept", "Alarm Failure",
"Time Change", "Traffic", "Other")
```

Or you can work with your data as a data frame, although for me, the data frame is only useful for displaying my data in a slightly different way:

```
df.defects <- data.frame(defect.counts)
```

Note that the *order of the counts* must be the same as the *order of the name labels*! So for example, there were 12 reports of being late due to bad weather, 29 reports of being late due to oversleeping, 18 reports of being late due to alarm clock problems, and so on.

This creates *two different data structures* that you can use to create your Pareto Chart. First, you have the vector called `defect.counts`. Additionally, you have the data frame called `df.defects` which also contains the name labels. It is easiest to generate the Pareto Chart using the vector, but it is often useful to print out the raw data as columns in the data frame, which can be easier to read.

You can tell R to show you the contents of each data structure:

```
> defect.counts
  Weather  Overslept Alarm Failure  Time Change  Traffic  Other
      12         29         18         3         34         4

> df.defects
      defect.counts
Weather          12
Overslept        29
Alarm Failure    18
Time Change       3
Traffic          34
Other            4
```

Code and Results

There are a few ways to build Pareto Charts in R. The easiest, in my opinion, is to use the `pareto.chart` function in the `qcc` package. **(Be sure that you have installed the `qcc` package and typed in `library(qcc)` to access this function before you begin!)** To create your Pareto Chart, type:

```
pareto.chart(defect.counts)
```

This command will display a summary of the values used to produce the Pareto Chart, as well as the Pareto Chart itself (shown in Figure 1).

```
> pareto.chart(defect.counts)
```

```
Pareto chart analysis for defect.counts
```

	Frequency	Cum.Freq.	Percentage	Cum.Percent.
Traffic	34	34	34	34
Overslept	29	63	29	63
Alarm Failure	18	81	18	81
Weather	12	93	12	93
Other	4	97	4	97
Time Change	3	100	3	100

There are many options that you can add to your Pareto Chart, as well:

Option	What it does
<code>main</code>	Specify a main title to be displayed above your chart
<code>xlab</code>	Specify a label to display on the x-axis
<code>ylab</code>	Specify a label to display on the y-axis
<code>col=type(length(defects))</code>	Sets the color palette to use for the bars in the bar chart (in place of <code>type</code> use <code>heat.colors</code> for reds and oranges, <code>rainbow</code> for ROYGBIV colors, <code>terrain.colors</code> for earthy greens into grays, or <code>topo.colors</code> to start with deep blues)
<code>cex.names=0.5</code>	Shrinks the fonts on the category labels (values in 0.5-0.8 range usually best)
<code>las=1</code>	Controls orientation of labels on axes (1=all horizontal, 2=all vertical, 3=perpendicular to axes)

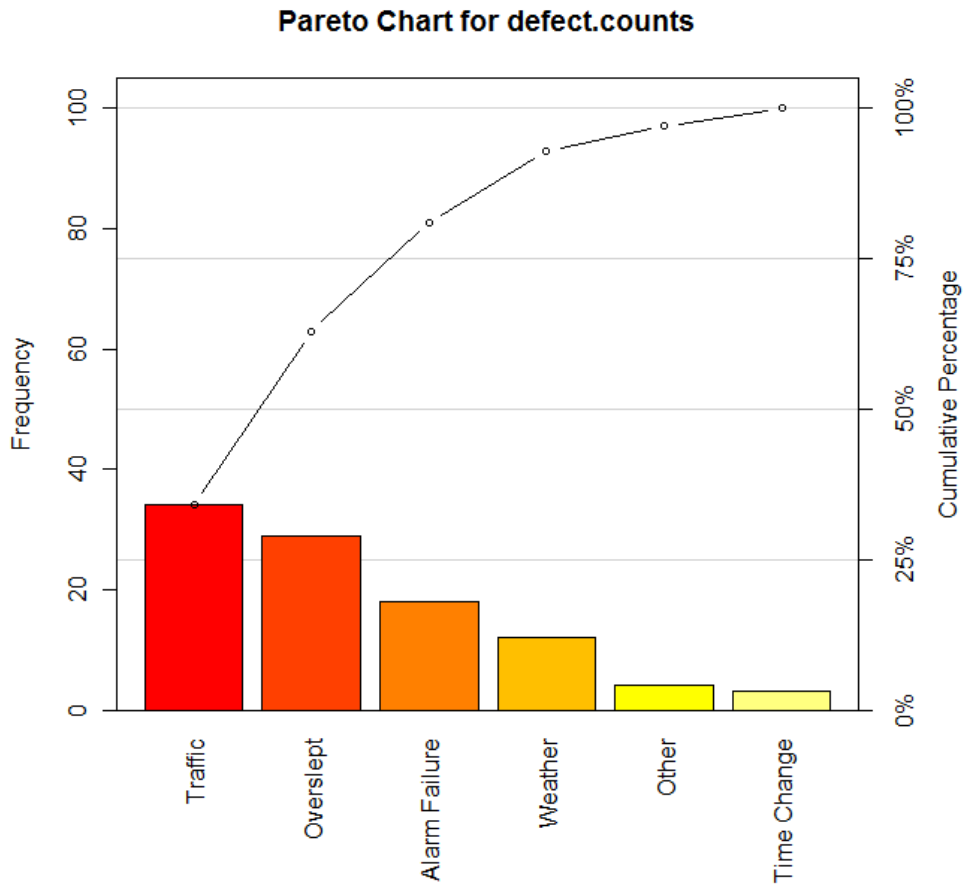


Figure 1: Chart produced by `pareto.chart(defect.counts)` that applies all of the default options

If your data is already in the form of a data frame, you can create a Pareto Chart using the command `pareto.chart(df.defects$defect.counts)`, but the chart loses the category names and replaces them with letters starting with A. I use this approach if my data is in a data frame and I just want to take a quick look at a Pareto Chart.

```
> pareto.chart(df.defects$defect.counts)
```

Pareto chart analysis for `df.defects$defect.counts`

	Frequency	Cum.Freq.	Percentage	Cum.Percent.
E	34	34	34	34
B	29	63	29	63
C	18	81	18	81
A	12	93	12	93
F	4	97	4	97
D	3	100	3	100

My favorite options are to create a main title, label the axes, make the labels on the categories horizontal so that they are more easily readable, and add an 80% line so I can easily distinguish the *vital few* from the *trivial many*:

```
> pareto.chart(defect.counts,main="My Pareto Chart",xlab="Reasons",  
ylab="Frequency", cex.names=0.6,las=1,col=topo.colors(6))
```

Pareto chart analysis for `defect.counts`

	Frequency	Cum.Freq.	Percentage	Cum.Percent.
Traffic	34	34	34	34
Overslept	29	63	29	63
Alarm Failure	18	81	18	81
Weather	12	93	12	93
Other	4	97	4	97
Time Change	3	100	3	100

Here's what the options in the command `pareto.chart` below that created the chart in Figure 2 *do*:

- First, `pareto.chart` reates a Pareto Chart out of the data in `defect.counts`
- Then `main` sets the main title of the graph to be "My Pareto Chart"
- The `xlab` option labels the x-axis with "Reasons"
- The `ylab` option labels the y-axis with "Frequency"
- The `cex` option shrinks the fonts on the x axis to 60% (or 0.6) of original size
- The `las=1` option orients the category labels horizontally instead of vertically
- And the color scheme is chosen from the `topo.colors` palette, with one color for each of the 6 categories in our original dataset (we could have also specified `topo.colors(length(defect.counts))`)

Next, I like to add a horizontal line at the cumulative percentage of 80%. To do this, I have to figure out what position on my y-axis represents 80% of my total counts, so to do this, I add up all my counts using `sum(defect.counts)` and multiply it by 0.8. Then, I can place an A-B Line (which is just a funny way to say “line” in R) horizontally (that’s what the `h=` is for) at `(sum(defect.counts)*.8)`. I choose red for the color, and set a line width of 4 with `lwd=4`. For a thinner line, I might pick `lwd=2`.

```
> abline(h=(sum(defect.counts)*.8),col="red",lwd=4)
```

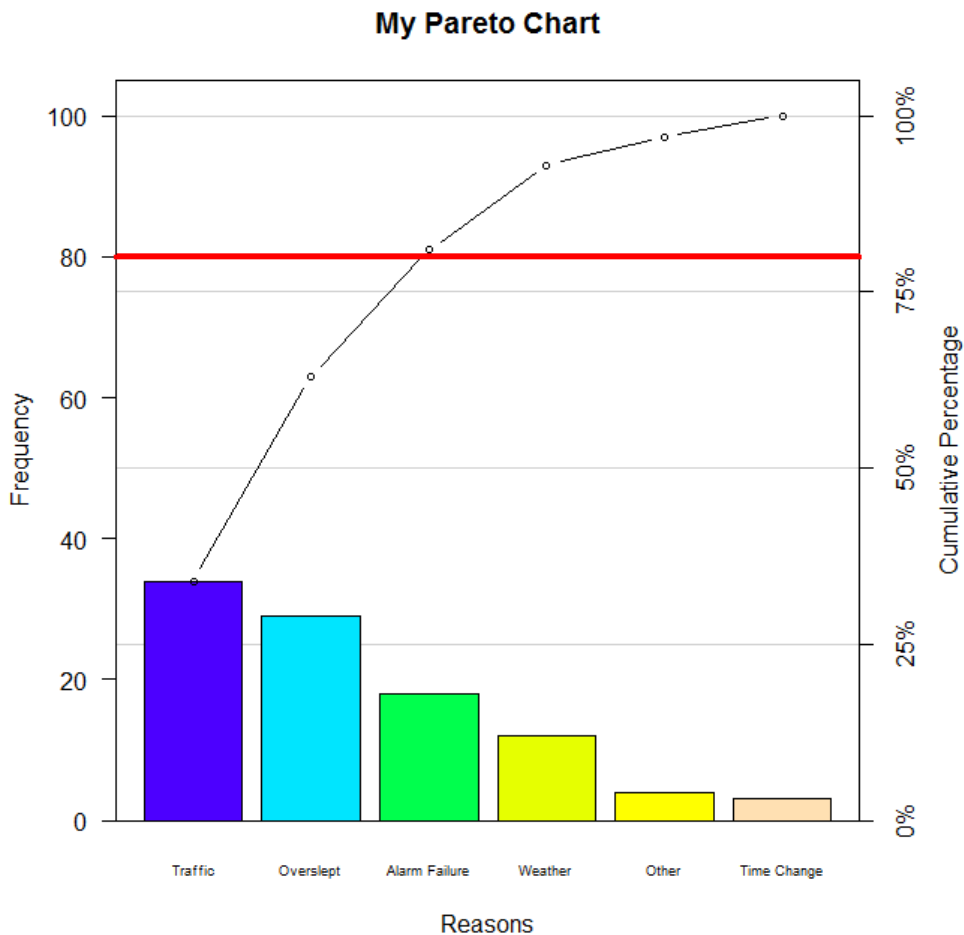


Figure 2: Chart produced by `pareto.chart` with additional options + `abline`

Conclusions

The Pareto Chart brings immediate focus to which reasons are part of the “vital few” and thus should receive attention first. By dropping a vertical line from where the horizontal line at 80% intersects the cumulative percentage line, this chart shows that traffic, oversleeping, and alarm failure are the most critical reasons that people in our survey are late for work. The next step in our problem solving activity might be to use the 5 Whys or Ishikawa/Fishbone Diagram techniques to determine the root causes of those reasons.

Other Resources:

- http://en.wikipedia.org/wiki/Pareto_chart
- http://en.wikipedia.org/wiki/Pareto_principle
- <http://www.dmaictools.com/dmaic-analyze/pareto-chart>
- <http://stackoverflow.com/questions/1735540/creating-a-pareto-chart-with-ggplot2-and-r>
- http://rgm2.lab.nig.ac.jp/RGM2/func.php?rd_id=qcc:pareto.chart