Pareto Charts

Kwan Lowe

8/26/2016

Pareto Charts

In a former life I ran an IT support desk. It wasn't at a large company, but when I took over IT support, the problem queue was always full with fully half the tickets in an overdue state. I used some basic tools to figure out how to reduce the size of the queue in order to have a chance of meeting SLAs. One of the first tools was a Pareto chart. Back then it was mostly done in a spreadsheet, but today I use R.

A Pareto chart is essentially a ranked bar chart with cumulative percentages. It's useful in determining where to concentrate efforts. In my case, I used it to determine which types of problems generated the most calls. It can also be used to help streamline processes by showing which steps of a process take the longest to complete. In short, using a Pareto chart makes it easy to visualize statements such as, "80% of support calls were generated because of connectivity issues or users selecting the wrong service."

For this example I generated a list of problems and their totals. For your own data, you would likely query a database directly. The data in the input file remdata.dat shows the problem categorization and the number of calls for that categorization.

```r
library(qcc)
```

```
## Package 'qcc', version 2.6
```

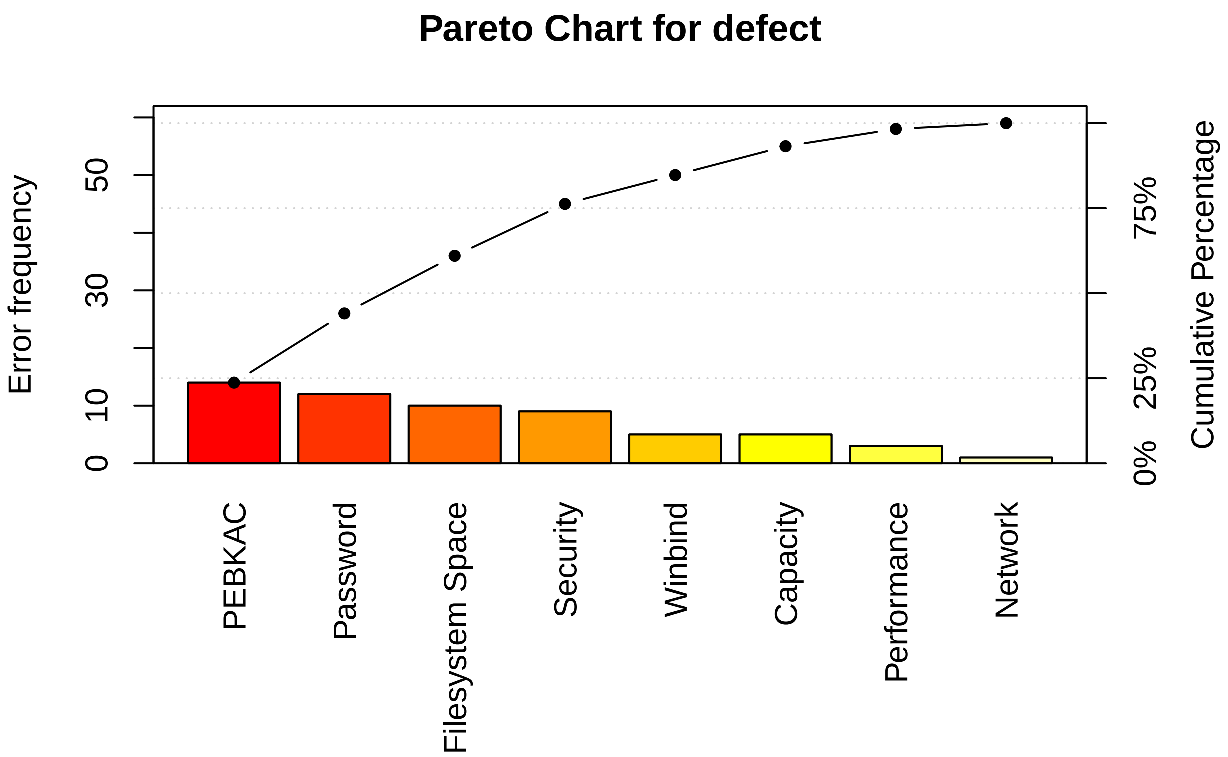
```
## Type 'citation("qcc")' for citing this R package in publications.
```

```r
library(gridExtra)

datafile <- "remdata.dat"
remedy_data <- read.table("remdata.dat", sep=",")
defect <- remedy_data[,2]
names(defect) <- remedy_data[,1]
```

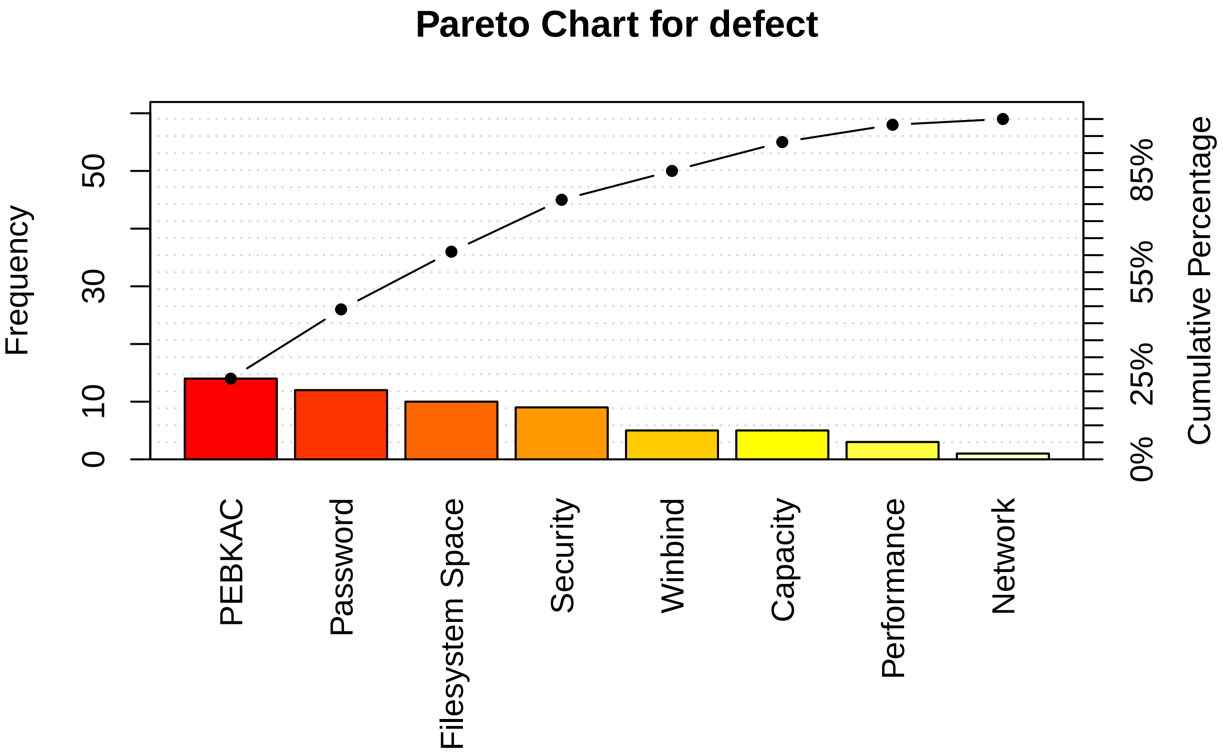
Here are some example plots:

```r
pareto.chart(defect, ylab = "Error frequency")
```



```
##
## Pareto chart analysis for defect
##                     Frequency Cum.Freq. Percentage Cum.Percent.
##   PEBKAC                   14        14  23.728814     23.72881
##   Password                 12        26  20.338983     44.06780
##   Filesystem Space         10        36  16.949153     61.01695
##   Security                  9        45  15.254237     76.27119
##   Winbind                   5        50   8.474576     84.74576
##   Capacity                  5        55   8.474576     93.22034
##   Performance               3        58   5.084746     98.30508
##   Network                   1        59   1.694915    100.00000
```

```
pareto.chart(defect, cumperc = seq(0, 100, by = 5), ylab2 = "Cumulative Percentage")
```



```
##
## Pareto chart analysis for defect
##                     Frequency Cum.Freq. Percentage Cum.Percent.
##   PEBKAC                   14        14  23.728814     23.72881
##   Password                 12        26  20.338983     44.06780
##   Filesystem Space         10        36  16.949153     61.01695
##   Security                  9        45  15.254237     76.27119
##   Winbind                   5        50   8.474576     84.74576
##   Capacity                  5        55   8.474576     93.22034
##   Performance               3        58   5.084746     98.30508
##   Network                   1        59   1.694915    100.00000
```

Conclusion

That's it. In some data sets it's quite obvious what is generating the majority of calls. In others, the graph is essentially flat so it takes some more work to find the underlying problem.

In the above data, the "Password", "Security", and "Winbind" issues could perhaps be aggregated into an "Authentication" category and suddenly the culprit stands out. Maybe adding centralized authentication can fix the problem, or going a step further, eliminate the need for endpoint authentication by changing the process.

And for my last points… Without data we're stabbing in the dark, fixing the wrong things, optimizing the 2% instead of the 80%. When tasked with fixing a problem, it's deceptively easy to hire a consultant to essentially act as a salesperson for whatever new technology is the Next Big Thing.

And you do have data to analyze, right? It may not seem like a big deal that your helpdesk folks are always selecting "Other" because that entry is the default, but when it comes time to crunching that data, you may wish that more time was spent on proper categorization.