

Contents

1	Cause and Effect Diagrams	2
2	Pareto Chart Analysis.	3
3	Pareto Analysis (Implementation with qcc package)	4
4	Operating Characteristic (OC) Curves	6
4.1	pistonrings Data	7
4.2	orangejuice Data	8
4.3	circuit Data	9
5	Moving Average (MA) Chart	10
6	Exponentially-weighted MA (EWMA) Chart	10
6.0.1	Grouped Data	11
6.0.2	Individual observations	13

1 Cause and Effect Diagrams

```
cause.and.effect(cause=list(  
  Measurements=c("Micrometers", "Microscopes", "Inspectors"),  
  Materials=c("Alloys", "Lubricants", "Suppliers"),  
  Personnel=c("Shifts", "Supervisors", "Training", "Operators"),  
  Environment=c("Condensation", "Moisture"),  
  Methods=c("Brake", "Engager", "Angle"),  
  Machines=c("Speed", "Lathes", "Bits", "Sockets")),  
effect="Surface Flaws")
```

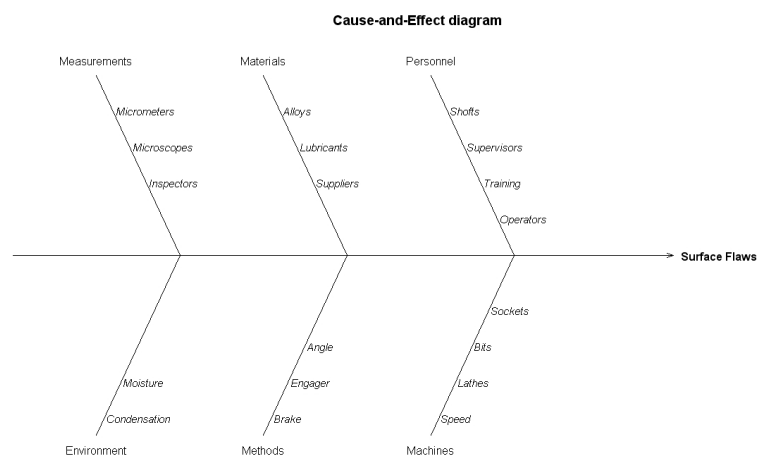


Figure 1:

2 Pareto Chart Analysis.

Quality problems are rarely spread evenly across the different aspects of the production process or different plants. Rather, a few "bad apples" often account for the majority of problems. This principle has come to be known as the Pareto principle, which basically states that quality losses are mal-distributed in such a way that a small percentage of possible causes are responsible for the majority of the quality problems. For example, a relatively small number of "dirty" cars are probably responsible for the majority of air pollution; the majority of losses in most companies result from the failure of only one or two products. To illustrate the "bad apples", one plots the Pareto chart,

3 Pareto Analysis (Implementation with qcc package)

```
defect <- c(80, 27, 66, 94, 33)

names(defect) <- c("price code", "schedule date",
  "supplier code", "contact num.", "part num.")

# 1
pareto.chart(defect, ylab = "Error frequency")

#2
pareto.chart(defect, ylab = "Error frequency", xlab = "Error causes", las=1)

#3
pareto.chart(defect, ylab = "Error frequency", col=rainbow(length(defect)))

#4
pareto.chart(defect, cumperc = seq(0, 100, by = 5),
  ylab2 = "A finer tickmarks grid")
```

Output to accompany graphs

Pareto chart analysis for defect

	Frequency	Cum.Freq.	Percentage	Cum.Percent.
contact num.	94	94	31.33333	31.33333
price code	80	174	26.66667	58.00000
supplier code	66	240	22.00000	80.00000
part num.	33	273	11.00000	91.00000
schedule date	27	300	9.00000	100.00000

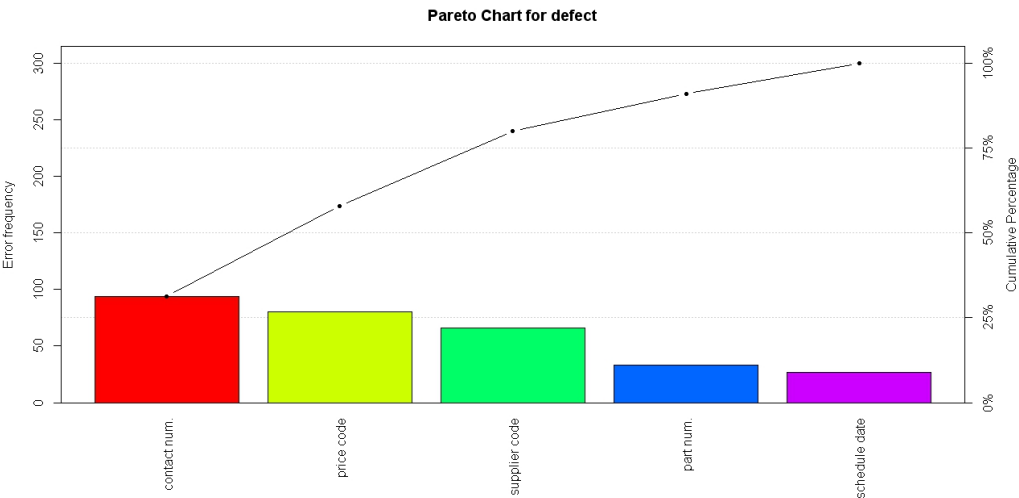


Figure 2: Third Implementation

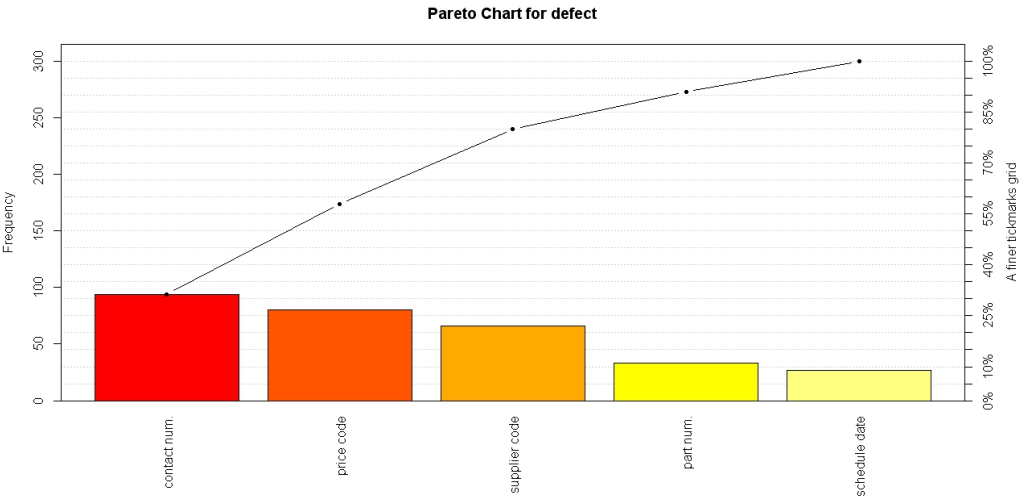


Figure 3: Fourth Implementation

4 Operating Characteristic (OC) Curves

- A common supplementary plot to standard quality control charts is the so-called operating characteristic or OC curve (see example below). One question that comes to mind when using standard variable or attribute charts is how sensitive is the current quality control procedure? Put in more specific terms, how likely is it that you will not find a sample (e.g., mean in an X-bar chart) outside the control limits (i.e., accept the production process as "in control"), when, in fact, it has shifted by a certain amount?
- This probability is usually referred to as the (beta) error probability, that is, the probability of erroneously accepting a process (mean, mean proportion, mean rate defectives, etc.) as being "in control."
- Note that operating characteristic curves pertain to the false-acceptance probability using the sample-outside-of- control-limits criterion only, and not the runs tests described earlier.
- Operating characteristic curves are extremely useful for exploring the power of our quality control procedure. The actual decision concerning sample sizes should depend not only on the cost of implementing the plan (e.g., cost per item sampled), but also on the costs resulting from not detecting quality problems. The OC curve allows the engineer to estimate the probabilities of not detecting shifts of certain sizes in the production quality.

4.1 pistonrings Data

```
data(pistonrings); attach(pistonrings);

diameter <- qcc.groups(diameter, sample)
beta <- oc.curves.xbar(qcc(diameter, type="xbar", nsigmas=3, plot=FALSE))
print(round(beta, digits=4))

# or to identify points on the plot use
## Not run: oc.curves.xbar(qcc(diameter,
    type="xbar", nsigmas=3, plot=FALSE), identify=TRUE)

detach(pistonrings)
```

4.2 orangejuice Data

```
data(orangejuice)
attach(orangejuice)
beta <- oc.curves(qcc(D[trial], sizes=size[trial], type="p", plot=FALSE))
print(round(beta, digits=4))
# or to identify points on the plot use
## Not run: oc.curves(qcc(D[trial], sizes=size[trial], type="p", plot=FALSE), identify=TRUE)
detach(orangejuice)
```


4.3 circuit Data

```
data(circuit)
attach(circuit)
q <- qcc(x[trial], sizes=size[trial], type="c", plot=FALSE)
beta <- oc.curves(q)
print(round(beta, digits=4))
# or to identify points on the plot use
## Not run: oc.curves(qcc(x[trial], sizes=size[trial], type="c", plot=FALSE), identify=TRUE)
detach(circuit)
```

5 Moving Average (MA) Chart

To return to the piston ring example, suppose we are mostly interested in detecting small trends across successive sample means. For example, we may be particularly concerned about machine wear, leading to a slow but constant deterioration of quality (i.e., deviation from specification). The CUSUM chart described above is one way to monitor such trends, and to detect small permanent shifts in the process average.

Another way is to use some weighting scheme that summarizes the means of several successive samples; moving such a weighted mean across the samples will produce a moving average chart (as shown in the following graph).

6 Exponentially-weighted MA (EWMA) Chart

The idea of moving averages of successive (adjacent) samples can be generalized. In principle, in order to detect a trend we need to weight successive samples to form a moving average; however, instead of a simple arithmetic moving average, we could compute a geometric moving average (this chart (see graph below) is also called Geometric Moving Average chart, see Montgomery, 1985, 1991).

6.0.1 Grouped Data

```
data(pistonrings)
attach(pistonrings)
diameter <- qcc.groups(diameter, sample)
q <- ewma(diameter[1:25,], lambda=0.2, nsigmas=3)
summary(q)

q <- ewma(diameter[1:25,], lambda=0.2, nsigmas=2.7,
newdata=diameter[26:40,], plot = FALSE)
summary(q)

plot(q)
```

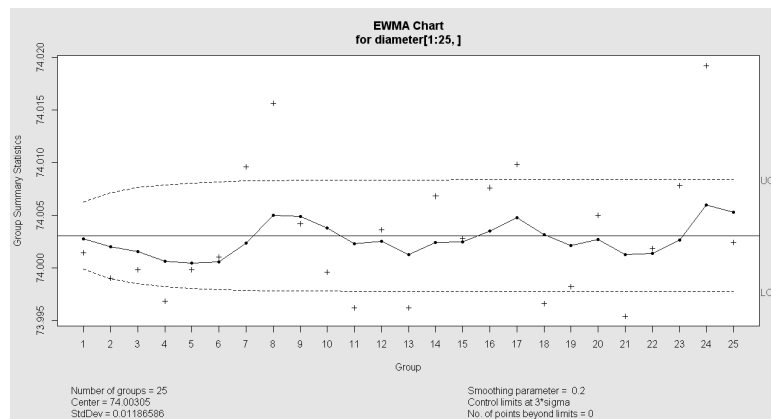


Figure 4:

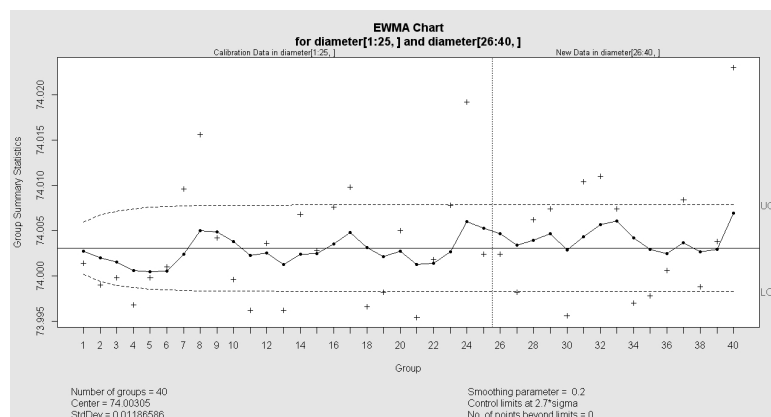


Figure 5:

```
> summary(q)
```

```
Call:
```

```
ewma(data = diameter[1:25, ], lambda = 0.2, nsigmas = 2.7, newdata = diameter[26:40,
```

```
ewma chart for diameter[1:25, ]
```

```
Summary of group statistics:
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
74.00	74.00	74.00	74.00	74.01	74.02

```
Group sample size: 5
```

```
Number of groups: 25
```

```
Center of group statistics: 74.00305
```

```
Standard deviation: 0.01186586
```

```
Summary of group statistics in diameter[26:40, ]:
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
74.00	74.00	74.00	74.00	74.01	74.02

```
Group sample size: 5
```

```
Number of groups: 15
```

```
Smoothing parameter: 0.2
```

```
Control limits:
```

	LCL	UCL
1	74.00018	74.00591
2	73.99938	74.00672
...		
40	73.99827	74.00782

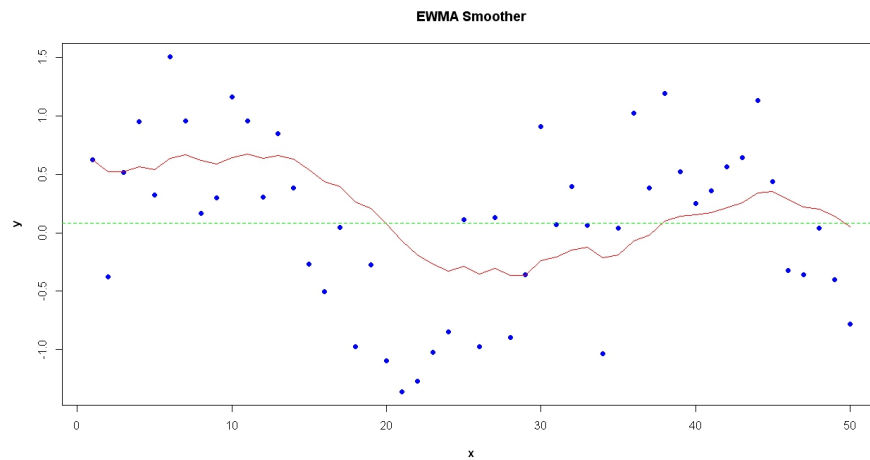


Figure 6:

6.0.2 Individual observations

```
x <- c(33.75, 33.05, 34, 33.81, 33.46, 34.02, 33.68, 33.27, 33.49, 33.20,
33.62, 33.00, 33.54, 33.12, 33.84) # viscosity data (Montgomery, pag. 242)
q <- ewma(x, lambda=0.2, nsigmas=2.7)
summary(q)
```

```
x <- 1:50
y <- rnorm(50, sin(x/5), 0.5)
plot(x,y,pch=16,col="blue",font.lab=2)
lines(ewmaSmooth(x,y,lambda=0.1), col="red")
abline(h=mean(y),col="green",lty=2)
title("EWMA Smoother")
```