qAnalyst User Manual

Giorgio Spedicato giorgio.spedicato@quantide.com
Andrea Spanò andrea.spano@quantide.com

February 12, 2008

Index

What is qAnalyst?

- qAnalyst is a R package to perform a large number of statistical quality control analysis.
- qAnalyst plots control charts and returns corresponding statistics.
- qAnalyst performs capability analysis for normal and non normal data.
- qAnalyst estimates statistical distributions parameters and execute goodness of fit analysis.
- qAnalyst can plot Pareto charts.
- qAnalyst current version is 0.2.0

Control charts overview

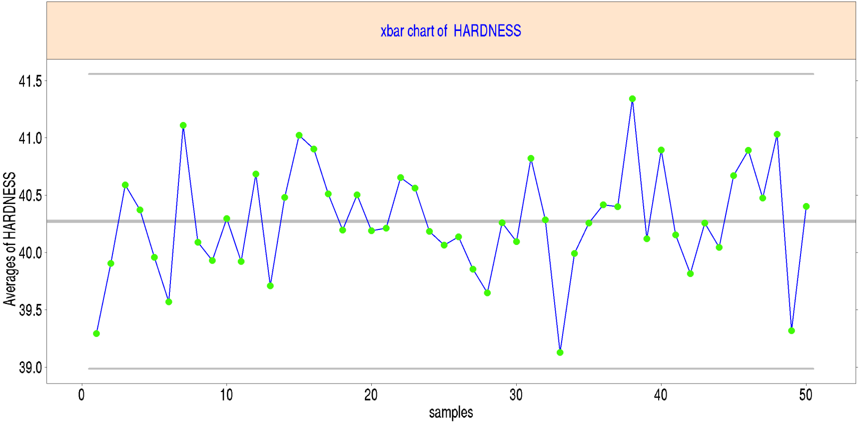
- Available control charts are:
  - x-bar, r, s charts within variable charts for subgroups.
  - i, mr charts within individual variable charts.
  - p, np, c, u within attribute charts.
- Control charts are build creating **“spc” objects**.
- Plot, print, summary methods print summary and detailed statistics and plot spc control charts.
- Tests are available to identify points out of controls according to well accepted rules.

x-bar chart example 1/2

Data of following example come from a shoe manufacturer. Shoe brakes characeristics are measured throught time by samples of five. Brake hardness variability has to be monitored. Data needed for this example are into data set "brakeCap", bundled within qAnalyst.

```
data(brakeCap)
hardnessXbar=spc(x=brakeCap$hardness,
sg=brakeCap$subgroup, type="xbar", name="HARDNESS")
plot(hardnessXbar)
```

x-bar chart example 2/2

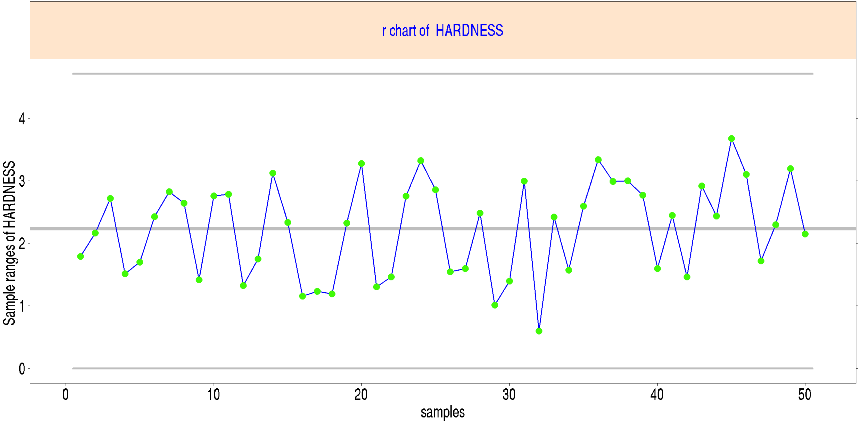


R and S charts examples

Code for R and S charts for previous data are reported below

```
data(brakeCap)
#schart
hardnessSchart=spc(x=brakeCap$hardness,
sg=brakeCap$subgroup, type="s", name="HARDNESS")
plot(hardnessSchart)
#rchart
hardnessRchart=spc(x=brakeCap$hardness,
sg=brakeCap$subgroup, type="r", name="HARDNESS")
plot(hardnessRchart)
```

r chart example



s chart example

Overview of **print** and **summary** generic methods

- `print` method returns brief informations about variable processed by spc object. Number of total, complete, missing observationss and number of subgroups is printed out. Moreover summary statistics are returned.
- `summary` returns performed test output and control chart plot elements coordinates in addition to the same `print` output informations.

print method output 1/2

```
#following Xbar example
print(hardnessXbar)
#ouptut
[1] "xbar chart of HARDNESS"

 HARDNESS  main stats
 ------------------------------------------
                                value
Total observations     250.0000000
complete observations 250.0000000
missing observations     0.0000000
number of groups        50.0000000
Mean                    40.2727992
min                     37.4492000
max                     43.0081000
```

print method output 2/2

```
total std. dev.         0.9858822
within std. dev.        0.9680147
between std. dev.       0.1868457
average range           2.2290060
```

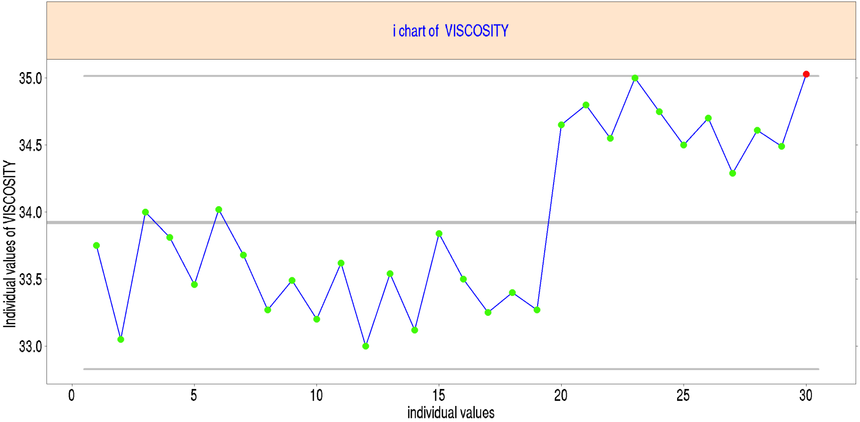
`summary` method output is presented further.

Individual values charts

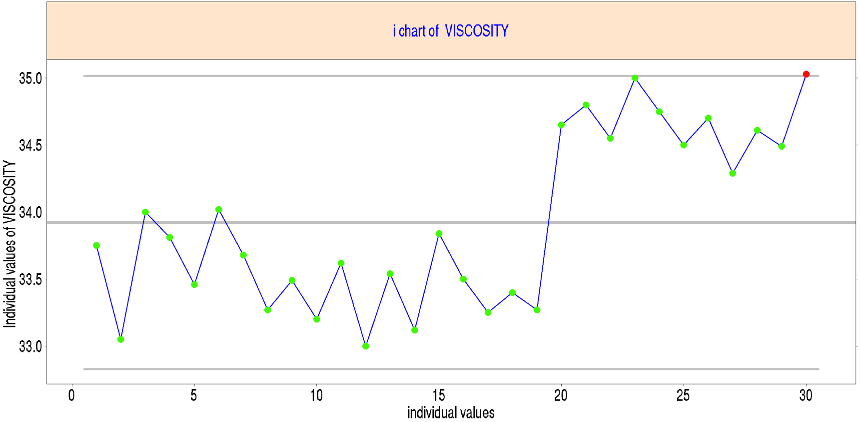
Spc object can be used to draw i and mr charts as reported in following example. Data from following example come from [Montgomery].

```
viscosity=c(33.75,33.05,34,33.81,33.46,
34.02,33.68,33.27,33.49,33.20,33.62,33,
33.54,33.12,33.84,33.5,33.25,33.4,33.27,
34.65,34.8,34.55,35,34.75,34.5,34.7,
34.29,34.61,34.49,35.03)
viscosityIchart=spc(x=viscosity,sg=1,type="i",
name="VISCOSITY", testType=1,k=1,p=1,nSigma=3)
viscosityMRchart=spc(x=viscosity,sg=1,type="i",
name="VISCOSITY", testType=1,k=1,p=1,nSigma=3)
plot(mrchartExample)
plot(viscosityMRchart)
```

I chart example



MR chart example



I chart example details 1/4

We asked the spc function to perform test “one”. That allowed us to mark all points beyond three standard deviation from center line. Plot method Detailed test result are tabulated within the output of `summary(mrchartExample)`

```
[1] "i chart of VISCOSITY"


 VISCOSITY  main stats

 ----------------------------------------

                              value
Total observations     30.0000000
complete observations 30.0000000
missing observations   0.0000000
number of groups       1.0000000
Mean                  33.9213333
min                   33.0000000
max                   35.0300000
```

I chart example details 2/4

```
total std. dev.        0.6413603
within std. dev.       0.0000000
between std. dev.      0.6413603
average range          0.4110345
Control chart tests results
 ------------------------------------------
Matrix of points failing required tests
index Test1
   30     1
 Control chart elements table
 ------------------------------------------
   points  lcl3s  lcl2s  lcl1s center line  ucl1s ucl2s  ucl3s
1   33.75 32.828 33.193 33.557       33.921 34.286 34.65 35.015
2   33.05 32.828 33.193 33.557       33.921 34.286 34.65 35.015
3   34.00 32.828 33.193 33.557       33.921 34.286 34.65 35.015
```

I chart example details 3/4

4	33.81	32.828	33.193	33.557	33.921	34.286	34.65	35.015
5	33.46	32.828	33.193	33.557	33.921	34.286	34.65	35.015
6	34.02	32.828	33.193	33.557	33.921	34.286	34.65	35.015
7	33.68	32.828	33.193	33.557	33.921	34.286	34.65	35.015
8	33.27	32.828	33.193	33.557	33.921	34.286	34.65	35.015
9	33.49	32.828	33.193	33.557	33.921	34.286	34.65	35.015
10	33.20	32.828	33.193	33.557	33.921	34.286	34.65	35.015
11	33.62	32.828	33.193	33.557	33.921	34.286	34.65	35.015
12	33.00	32.828	33.193	33.557	33.921	34.286	34.65	35.015
13	33.54	32.828	33.193	33.557	33.921	34.286	34.65	35.015
14	33.12	32.828	33.193	33.557	33.921	34.286	34.65	35.015
15	33.84	32.828	33.193	33.557	33.921	34.286	34.65	35.015
16	33.50	32.828	33.193	33.557	33.921	34.286	34.65	35.015
17	33.25	32.828	33.193	33.557	33.921	34.286	34.65	35.015
18	33.40	32.828	33.193	33.557	33.921	34.286	34.65	35.015

I chart example details 4/4

19	33.27	32.828	33.193	33.557		33.921	34.286	34.65	35.015
20	34.65	32.828	33.193	33.557		33.921	34.286	34.65	35.015
21	34.80	32.828	33.193	33.557		33.921	34.286	34.65	35.015
22	34.55	32.828	33.193	33.557		33.921	34.286	34.65	35.015
23	35.00	32.828	33.193	33.557		33.921	34.286	34.65	35.015
24	34.75	32.828	33.193	33.557		33.921	34.286	34.65	35.015
25	34.50	32.828	33.193	33.557		33.921	34.286	34.65	35.015
26	34.70	32.828	33.193	33.557		33.921	34.286	34.65	35.015
27	34.29	32.828	33.193	33.557		33.921	34.286	34.65	35.015
28	34.61	32.828	33.193	33.557		33.921	34.286	34.65	35.015
29	34.49	32.828	33.193	33.557		33.921	34.286	34.65	35.015
30	35.03	32.828	33.193	33.557		33.921	34.286	34.65	35.015

Attribute charts

Attribute charts can be created by spc object. Available attribute charts are:

1. p chart
2. np chart
3. c chart
4. u chart

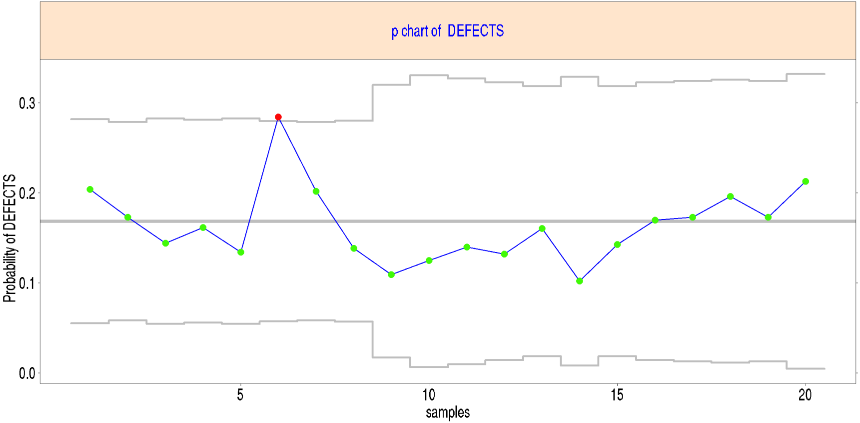
that handle binomial and Poisson data. Subgroup dimension is indicated by sg field with the exception of *c charts* where sg field is not needed. It is easy to deal with unequal subgroups size by parameter sg.

p chart and np chart

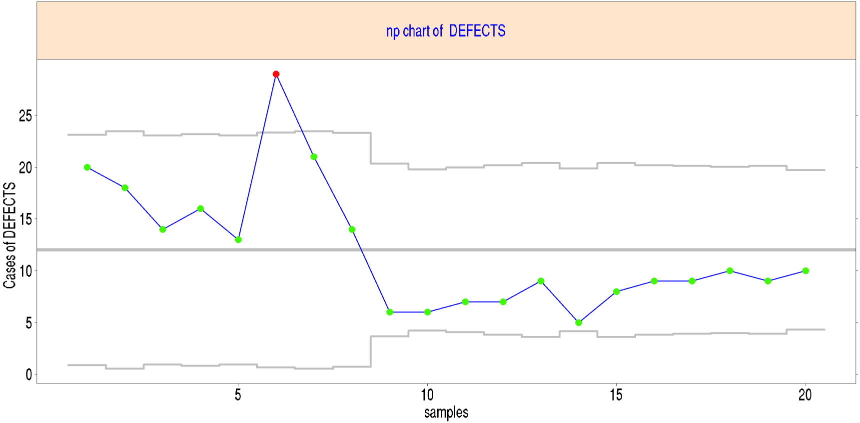
Following examples use data set *tubes*, that is bundled within qAnalyst package. Tubes data set reports results of quality inspections in a televisions tube manufacturing. This example shows how qAnalyst can easily handle different subgroup dimensions.

```
data(tubes)
rejectsPchart=spc(x=tubes$rejects, sg=tubes$sampled, type="p",
name="DEFECTS", testType=1, nSigma=3, k=1, p=1)
rejectsNPchart=spc(x=tubes$rejects, sg=tubes$sampled, type="np",
name="DEFECTS", testType=1, nSigma=3, k=1, p=1)
plot(rejectsPchart)
plot(rejectsNPchart)
```

p chart example



np chart example

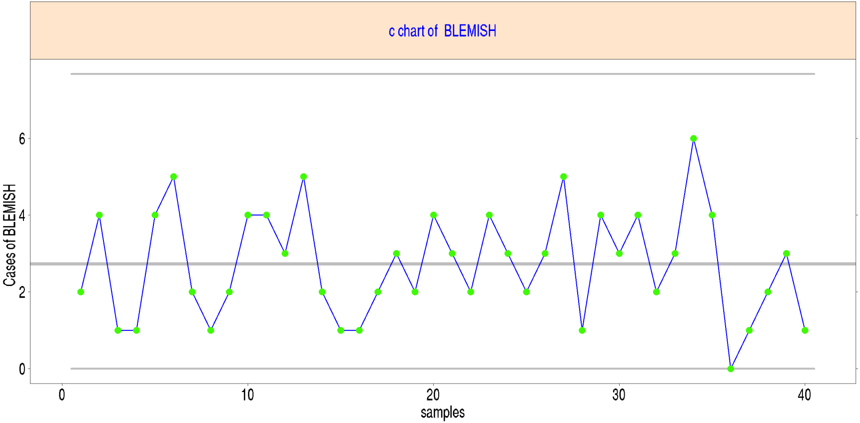


c chart and u chart

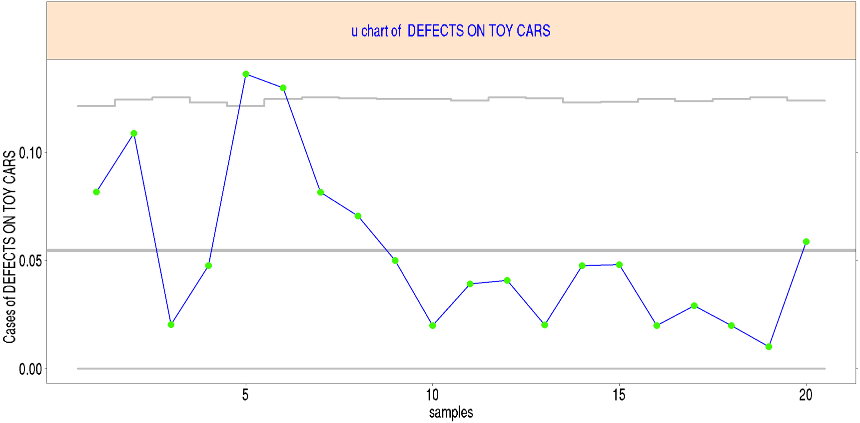
Following examples show statistical process control on Poisson data. Reported examples use data from data set *blemish* and *toys* bundled into qAnalyst package

```
data(blemish); data(toyCarsDefects)
blemishCchart=spc(x=blemish$blemish, type="c", name="BLEMISH")
toysUchart=spc(x=toyCarsDefects$defects,sg=toyCarsDefects$sampled,
type="u", name="DEFECTS ON TOY CARS")
plot(blemishCchart)
plot(toysUchart)
dev.off()
```

c chart example



u chart example



Capability functions overview

qAnalyst package performs capability analysis for both normal and non normal data. Capability analysis is performed creating **capability** object using functions **capabilityNormal** and **capabilityNotNormal**. Methods **plot, print, summary** permit to plot capability histogram and to print many capability stats. Available capability indexes are:

- PPM, PPL, PPU; these stats can be avaiable in percentage format.
- CP, CPU, CPL, CPK, CPM; these stats can also be reported in zeta format.

Capability functions overview

Two group of analysis, “overall” and “potential” capability, are avaiabile if data come from a normal distribution. Otherwise only “overall” analyses are avaiable.

Moreover, **invCPFun** performs inverse capability analysis. In other words, minimum width tolerance limits can be calculated from data variability, given a desidered $cp = cpk$.

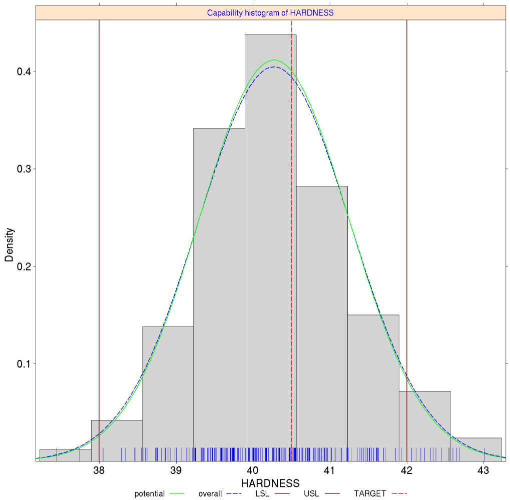
Details about capability background statistical theory can be found in [Bothe].

Capability analysis for normal data

Following example comes from *brakeCap* dataset. Specified tolerance limits for brake hardness are 38 - 42. Moreover, a target of 40.5 would be desidered. Hardness is supposed to follow a normal distribution.

```
data(brakeCap)
hardnessCap=capabilityNormal(x=brakeCap$hardness,
sg=brakeCap$subgroup, lsl=38,usl=42, target=40.5, name="HARDNESS")
plot(hardnessCap)
summary(harnessCap)
print(hardnessCap)
```

Normal capability plot



Normal capability **print** method

```
Capability analysis of  HARDNESS
mean of  HARDNESS  is  40
Total observations of  HARDNESS Non missing observations:  250
total ppm observed of  HARDNESS  are  64000
Estimated pp of  HARDNESS  is  0.68
Estimated ppl of   is  0.77
Estimated ppu of   is  0.58
Estimated ppk of   is  0.58
Estimated cpm of   is  0.49
```

Normal capability **summary** method 1/3

```
 summary of  HARDNESS  capability analysis
 Assumed distribution:  normal
 ---------------------
 Summary statistics
                   value
Total obs.       250.00000
Non missing obs. 250.00000
Missing Obs.       0.00000
Groups            50.00000
Mean              40.27280
Overall st. dev.   0.98588
Within st. dev.    0.96922
```

Normal capability **summary** method 2/3

```
Observed performance
          value
PPM Total 64000
PPM < LSL  8000
PPM > USL 56000
 Expected overall performance
          value
PPM Total 50466
PPM < LSL 10574
PPM > USL 39893
 Expected within performance
          value
PPM Total 46884
PPM < LSL  9514
PPM > USL 37370
```

Normal capability **summary** method 3/3

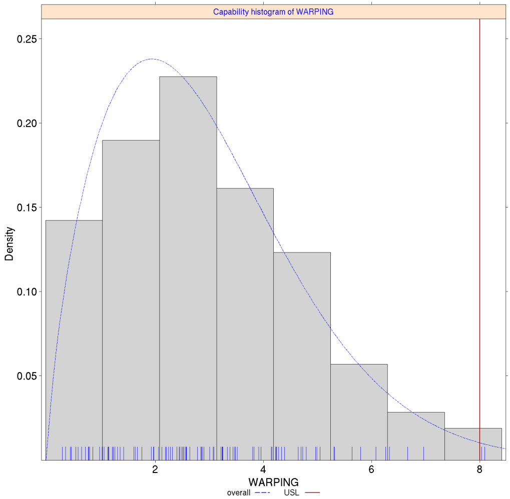
```
 Overall Capability
      value
pp   0.67621
ppl 0.76845
ppu 0.58398
ppk 0.58398
cpm 0.49416
 Potential Capability
      value
cp   0.68784
cpl 0.78166
cpu 0.59402
cpk 0.59402
```

Non normal capability example 1/3

Warptiles data set let to show an example of capability analysis on non - normal data. Warping on sampled tiles has been measured. An upper tolerance limit is equal to eigth. Warping is assumed to follow a weibull distribution.

```
data(warpTiles)
warpCap=capabilityNotNormal(x=warpTiles$warping,
usl=8,distribution="weibull", name="WARPING")
plot(warpCap)
print(warpCap)
```

Non normal capability example 2/3



Non normal capability example 3/3

```
Capability analysis of  WARPING
mean of  WARPING  is  2.9
Total observations of  WARPING Non missing observations:  100
total ppm observed of  WARPING  are  20000
Estimated ppu of   is  0.73
Estimated ppk of   is  0.73
```

Inverse capability feature

Performing an inverse capability means to state tolerance limits for a desidered *cp=cpk*. In other words, user gives a value for *cp=cpk*, $invCpFun$ returns the smallest tolerance limits interval consinstent with given specifications. E.g., in brakeCap dataset, hardness tolerance limit for a $cp = 1.33$ are $36 - 44$, obtained by

```
invCpFun(x=brakeCap$hardness, cp=4/3, fun="normal")
```

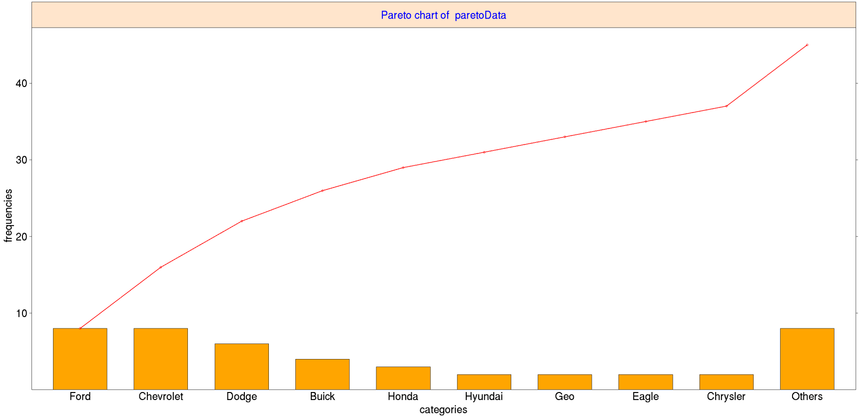
command.

Pareto Charts 1/2

qAnalyst package can plot **Pareto chart** for categorical variables. Pareto chart is a quick tool to identify most frequenty occurring categories. Categories are orded decreasing by frequency and represented by bars. A percentage treshold can be set to group less frequent items after cumulated frequency exceeds defined threshold.
Cars93 dataset from package MASS contains car characteristics of a random sample of cars owner. Most common makes can be show by a Pareto Chart.

```
data(Cars93) #package MASS
paretoData=Cars93$Manufacturer[1:45]
paretoChart(x=paretoData,mergeThr=.8, addLine=TRUE)
```

Pareto Charts 2/2



Overview of distribution analysis functions

qAnalyst package provides instruments to fit parameters of most important distribution from data. Moveover it provides instruments to check goodness of fit. Box Cox and Johnson transformations are supported too. Most important functions are:

- **funInfoFun**, **rapidFit** and **andersonDarlingFun** functions for single distribution fitting
- **probplot** function to obtain probability plots
- **boxcoxFun** function to obtain Box Cox transformation
- **johnsonFun** function to obtain johnson transformation

Distribution fitting 1/3

Currently supported continuous distributions are: normal, lognormal, the gamma family, weibull, logistic, cauchy.
`funInfoFun` let user to fit parameters for a specific distribution.
`funInfoFun` creates `infoFun` objects for which `summary` method is available.

```
#R script
data(warpTiles)
infoX=funInfoFun(warpTiles$warping, "weibull")
summary(infoX)
#output
Hypotized distibution is  weibull
Estimated values for shape scale 1.694 3.278
AD statistic is 0.248 , corresponding p-value 0.753
```

Distribution fitting 2/3

rapidFitFun helps user to find best fitting distribution within supported distribution, according to Anderson Darling goodness of fit test. Use of rapidFitFun is shown in the example below.

```
#R script
data(brakeCap)
rapidFitFun(brakeCap$centering)
#output
Distributions fit output
 ---------------------
  distributions parameter1 parameter2   theta1 theta2 ADpvalue
1        normal       mean         sd   41.028  0.977    0.850
2     lognormal    meanlog      sdlog    3.714  0.024    0.871
3         gamma      shape       rate 1757.551 42.838    0.873
4       weibull      shape      scale   43.233 41.507    0.000
```

Distribution fitting 3/3

```
5      logistic   location     scale   41.024  0.558    0.794
6        cauchy   location     scale   41.018  0.582    0.000

"Distribution with higher AD p-value is gamma"
```

andersonDarlingFun function helps user to numerically assess single distribution goodness of fit, as example below shows.

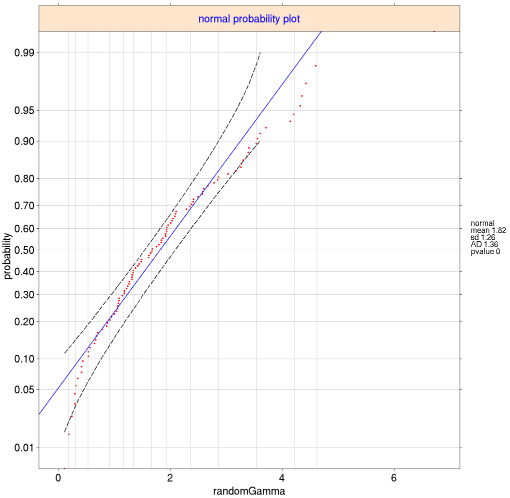
```
#R script
set.seed(100)
randomGamma=rgamma(100,2,1)
andersonDarlingFun(x=randomGamma,distribution="normal", theta=c(mean(x), sd(x)))
#output
         AD       pvalue
1.356429411 0.001635293
```

Probability plots 1/3

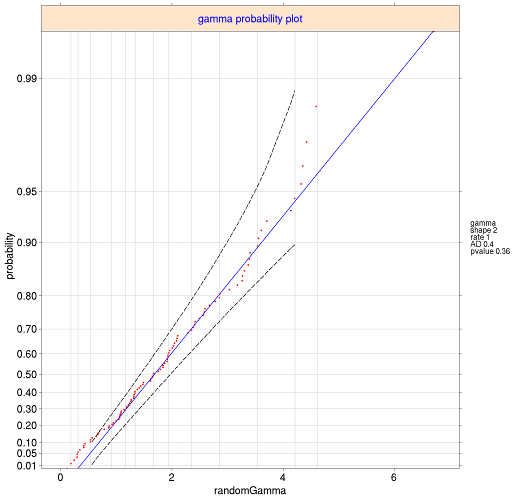
Probability plots let user to graphically asses empirical data fitting with respect to a hypotized distribution. If a theoretical distribution fits adequately data, all plotted points should lie around plotted central line and within confidence bands. A legend reporting parameter fits and Anderson Darling statistics is added to the graph.
Probability plot are available for quoted before supported distribution. Following pictures show normal probability plot and gamma probability plot for *randomGamma* data generated in the example before.

```
#R script
probplot(randomGamma,"normal", theta=c(mean(x), sd(x)),confintervals=T)
probplot(randomGamma,"gamma",theta=c(2,1),confintervals=T)
```

Probability plots 2/3



Probability plots 3/3



Variables transformations overview

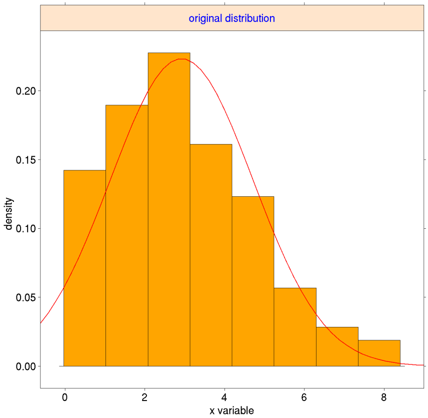
qAnalyst package performs Box Cox and Johnson transformation. Functions `boxcoxFun` and `johnsonFun` creates `transformation` objects for which `print` and `plot` methods are available. Following examples shows how to use qAnalyst function to handle data transformations. See [Box, Cox, Johnson] for further details about Box Cox and Johnson transformations.

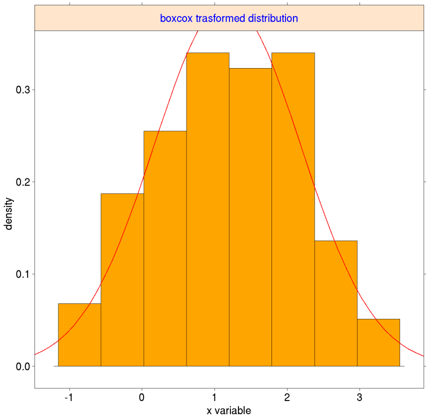
Box cox transformation 1/2

Below code for obtaining Box-Cox transformation of input data is reported.

```
#R code
data(warpTiles)
warpingBC=boxcoxFun(warpTiles$warping)
print(warpingBC)
plot(warpingBC)
#print method output
box cox  transformation of variable  warpingBC
Parameter(s) estimation
0.4272021
```

Box cox transformation 2/2





Johnson transformation 1/3

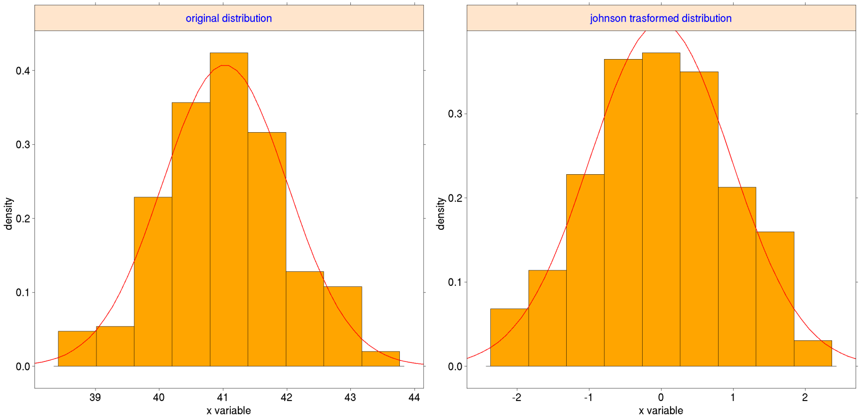
Below code for obtaining Johnson transformation of input data is reported.

```
#R code
data(brakeCap)
centeringJT=johnsonFun(brakeCap$centering)
print(centeringJT)
plot(centeringJT)
#print method output
 johnson  transformation of variable  centeringJT
Parameter(s) estimation
$gamma
[1] -0.08615386
$delta
[1] 1.775478
```

Johnson transformation 2/3

```
$xi
[1] 40.94764
$lambda
[1] 1.570807
$type
[1] "SU"
```

Johnson transformation 3/3



Bibliography

- Montgomery, D.C. Introduction to Statistical Quality Control, 5th ed, New York, John Wiley & Sons.
- Bothe (1997), Measuring Process Capability, McGraw Hill
- Box, and Cox, (1964) An analysis of transformations. Journal of the Royal Statistical Society.
- Johnson, N.L. (1949), Systems of frequency curves generated by methods of translation, Biometrika