

Rankit Plot

Shaillay Kumar Dogra
Scientific Editor – QSAR World
editor@qsarworld.com

Notes:

1. This Jython script works in Sarchitect Designer version 2.2
2. Learn about Sarchitect Designer – <http://www.strandls.com/sarchitect/index.html>
3. Get Sarchitect – <http://www.strandls.com/sarchitect/freetrial.php>

The actual script follows this discussion. It is also accessible directly from the webpage in .py format.

Discussion:

Assumption of normal distribution of data is an important prerequisite for some statistical tests (parametric) and regression methods. This assumption can be tested by using various graphical methods like rankit plots, normal probability plots and tests like Shapiro-Wilk (SW) or Kolmogorov-Smirnov (KS) tests.

A simple way to test the normality assumption could be to just look at the distribution of data as a histogram and see if it assumes a bell-shaped distribution that is characteristic of a normal or Gaussian distribution. If this is not the case, some transformations like log transformation can be attempted to see if now the new distribution takes more of a bell shaped curve.

Also, parameters like skewness and kurtosis can be checked. Quantile distribution of values can be looked at to see if a given mass of distribution is falling below a given percentile position patterning the indicative trend of a normal distribution (ND).

A graph plotting the rankits versus the data points is known as a rankit plot. If the sample is sufficiently large and comes from a normally distributed population, such a plot should approximate a straight line. Significant deviations from straightness can indicate evidence against normality of the distribution (rankit).

For more discussion about normality tests read [GraphPad, Eng-Stat, wikipedia].

A rankit plot is quite simple to generate. We have a list of values with us whose normal distribution we wish to check. We need to get an equal number of data points from a normal distribution (mean 0, variance 1). We then plot these two lists against each other after sorting them both and putting corresponding ranks against each other as the x, y pairs.

As mentioned above, I needed to generate random numbers from a Gaussian distribution with mean 0 and variance 1 for which purpose I used python's 'random' module (python).

A plot of particular interest to look at is a rankit plot of the 'standardized residuals'.

In the script given here, the name of the column containing values for which rankit plot needs to be generated is hard-coded (line #56) as 'Standardized Residuals'. Change it per your column name for which you want to generate the plot.

As output, it launches a scatter plot of (sorted) standardized residuals (or input column) and (sorted) random values. This is the rankit plot. These sorted values also get appended to the spreadsheet as 2 columns and are not in the same order as your identifier because of the sorting.

References:

[SW] <http://www.itl.nist.gov/div898/handbook/prc/section2/prc213.htm>

[KS] <http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm>

[ND] http://en.wikipedia.org/wiki/Normal_distribution

[rankit] http://en.wikipedia.org/wiki/Rankit_plot

[python] <http://docs.python.org/lib/module-random.html>

[GraphPad] http://www.graphpad.com/library/BiostatsSpecial/article_197.htm

[Eng-Stat] <http://www.itl.nist.gov/div898/handbook/prc/section2/prc21.htm>

[wikipedia] http://en.wikipedia.org/wiki/Normality_test

Cite this as:

Dogra, Shaillay K., "Script for getting Rankit plot" from QSARWorld – free online resource for QSAR modeling.

<http://www.qsarworld.com/virtual-workshop.php>

```
##
##
##
##  sarchitect designer script to get 'Rankit plot'
##
##  Shaillay Kumar Dogra
##  05 Oct 2006
##  editor@qsarworld.com
##
##
##  INPUT: a column labeled as "Standardized Residuals".
##  Script can be modified to change the input column and check
##  'normality' assumption
##  for any set of values (in a given column).
##
##  Generates random nos. belonging to Gaussian distribution of mean ##  0 and variance 1;
##  sorts them.
##  Sorts the input column.
##
##  OUTPUT: scatter plot of 'sorted random nos.' vs. 'sorted input ##  column'.
##
##  Appends 2 columns to dataset - "Sorted Random Values" & "Sorted ##  Standardized
##  Residuals".
##
##
##  References :
##  (1) http://en.wikipedia.org/wiki/Normal\_probability\_plot
##  (2) http://docs.python.org/lib/module-random.html
##
##
```

```
import script
from script.dataset import *
from script.project import *
from script.omega import createComponent, showDialog
from javax.swing import *
from math import *
from script.view import *
import random  ## python 'random' module

dataset = getActiveDataset()
row_count = dataset.getRowCount()

rand_col = [0] * row_count
for i in range(row_count):
    rand_col[i] = random.gauss(0,1)

rand_col.sort()

addCol=createFloatColumn("Sorted Random Values",rand_col)
dataset.addColumn(addCol)

## CHECK GAUSSIAN-DISTRIBUTION 'QUALITY' OF GENERATED RANDOM NOS.
```

```
#rand_idx = dataset.index("Sorted Random Values")
#Histogram(column = rand_idx).show()

residual_col = dataset.index("Standardized Residuals")
sorted_resid = [ ]
for i in dataset[residual_col]:
    sorted_resid.append(i)

sorted_resid.sort()

addCol=createFloatColumn("Sorted Standardized Residuals",sorted_resid)
dataset.addColumn(addCol)

## SCATTER PLOT
x_col = dataset.index("Sorted Random Values")
y_col = dataset.index("Sorted Standardized Residuals")
ScatterPlot(yaxis = y_col, xaxis = x_col).show()

## REPORT COMPLETION
parent=script.tool.getTool().getFrame()
mesg = "Done With Script Execution."
JOptionPane.showMessageDialog(parent,mesg,"STATUS!",JOptionPane.INFORMATION_MESS
AGE)

##
## END
##
```

End of Document