Programowanie narzędzi analitycznych Z03

Rafał Woźniak

Faculty of Economic Sciences, University of Warsaw

Warszawa, 21-10-2021

Wykresy w R

Wybrane funkcje do tworzenia wykresów:

- `plot` - tworzy wykresy: liniowe lub rozproszenia
- `hist` - histogram
- `barplot` - wykres słupkowy
- `boxplot` - wykres pudełkowy
- `qqnorm` - wykres kwantylowy
- `curve(x^2, from=0, to=2)` - tworzy wykres krzywej określonej wzorem, argumentem jest x

QQplot

- In statistics, a Q–Q (quantile-quantile) plot is a probability plot, which is a graphical method for comparing two probability distributions by plotting their quantiles against each other.
- First, the set of intervals for the quantiles is chosen. A point (x, y) on the plot corresponds to one of the quantiles of the second distribution (y-coordinate) plotted against the same quantile of the first distribution (x-coordinate).
- If the two distributions being compared are similar, the points in the Q–Q plot will approximately lie on the line $y = x$. If the distributions are linearly related, the points in the Q–Q plot will approximately lie on a line, but not necessarily on the line $y = x$.
- Q–Q plots can also be used as a graphical means of estimating parameters in a location-scale family of distributions.

Źródło: https://en.wikipedia.org/wiki/Q%E2%80%93Q_plot (link)

Jarque-Bera test

In statistics, the Jarque–Bera test is a goodness-of-fit test of whether sample data have the skewness and kurtosis matching a normal distribution. The test statistic JB is defined as

$$JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(C-3)^2\right) \tag{1}$$

where n is the number of observations; S is the sample skewness, C is the sample kurtosis:

$$S = \frac{\hat{\mu}_3}{\hat{\sigma}^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{3/2}}, \tag{2}$$

$$C = \frac{\hat{\mu}_4}{\hat{\sigma}^4} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{2}}, \tag{3}$$

where $\hat{\mu}_3$ and $\hat{\mu}_4$ are the estimates of third and fourth central moments, respectively, $\bar{x}$ is the sample mean, and $\hat{\sigma}^2$ is the estimate of the variance.

Źródło: https://en.wikipedia.org/wiki/Jarque%E2%80%93Bera_test

Jarque-Bera test

- If the data comes from a normal distribution, the JB statistic asymptotically has a chi-squared distribution with two degrees of freedom, so the statistic can be used to test the hypothesis that the data are from a normal distribution.
- The null hypothesis is a joint hypothesis of the skewness being zero and the excess kurtosis being zero.
- Samples from a normal distribution have an expected skewness of 0 and an expected excess kurtosis of 0 (which is the same as a kurtosis of 3). As the definition of JB shows, any deviation from this increases the JB statistic.
- For small samples the chi-squared approximation is overly sensitive, often rejecting the null hypothesis when it is true.
- Furthermore, the distribution of p-values departs from a uniform distribution and becomes a right-skewed uni-modal distribution, especially for small p-values. This leads to a large Type I error rate.

Źródło: https://en.wikipedia.org/wiki/Jarque%E2%80%93Bera_test

Różnica między skryptami a funkcjami

http://www.mathworks.com/help/matlab/matlab_prog/scripts-and-functions.html

Skrypt

Each time you run it [script], the script stores the result in a variable [...] that is in the base workspace.

Funkcja

[...] instead of manually updating the script each time, you can make your program more flexible by converting it to a function. [...] Functions have their own workspace, separate from the base workspace. Therefore, none of the calls to the function [...] overwrite the value of a in the base workspace.

Składnia funkcji

Składnia

```
y = function(arg) {
    expression
}
```

Przykład

```
norm <- function(x) {
    if is.vector(x)==FALSE {
        stop("x nie jest wektorem")
    }
    norm = sqrt(x%*%x)
    return(norm)
}
```

Wywołanie

norm(1:4)

Pętla for

Składnia

```
for(var in seq) {
    expression
}
```

Przykład

```
for(v in 1:10) {
    print(v)
}
```

Pętla while

Składnia

```
while(cond) {
    expression
}
```

Przykład

```
n = 1;
while(n<10) {
    print(n)
    n=n+1
}
```

Wyrażenie if

Składnia

```
if ( test_expression1) {
   statement1
} else if ( test_expression2) {
   statement2
} else
   statement4
```

Przykład

```
if (x<0) {
    print("x jest ujemny")
} else if (x==0) {
    print("x jest rowny zero")
} else {
    print("x jest dodatni")
}
```