

Validation Procedure of QuantLets according to the Styleguide

The proposed validation procedure should be followed sequentially. All steps are illustrated by examples included in the styleguide.

Instructions

1. Check Code (Align QuantLet Code to Style Guide standards)

1.1. The code should begin with the following segment (1)

```
# clear all variables
rm(list = ls(all = TRUE))
graphics.off()
```

1.2. If packages are used, then all code segments which are related to packages should be written at the segment (2) [directly after segment (1)] like this:

```
# install and load packages
libraries = c("rpart", "rpartScore")
lapply(libraries, function(x) if (!(x %in% installed.packages()))
{
  install.packages(x)
})
lapply(libraries, library, quietly = TRUE, character.only = TRUE)
```

1.3. Check whether the code runs properly

1.4. Check whether datafile is appropriate and available

1.5. Format the code with the package FormatR

1.5.1. Install/ load package FormatR and execute tidy_source command

1.5.2. Check whether all lines of code are still written properly

1.5.3. Check that the code still works properly

1.6. Check whether plots\pictures are appropriate to be uploaded on the website

1.6.1. Check whether all plots have appropriate titles and legends

1.7. Check whether the descriptive comments are appropriate and written as one line

1.8. Delete unnecessary empty lines

1.8.1. There should be no empty line between comment and corresponding code segment

1.8.2. There should be one empty line after commented code segment (could be two, if it is one of few major code segments)

- 1.8.3. There should be not more than 2 empty lines in a row
- 1.9. Change all "<" with "="
- 1.10. Align all subsequent assignments by "="
- 1.11. Check indentation
- 2. **Check all items of MetaInfo.txt** for order, completeness and correctness [optional items are written in *italic*] - Click to see example
 - 2.1. Name of QuantLet (e.g. SFEDAXlogreturns)
 - SFE_DAXlogreturns - bad style*
 - 2.2. Published in Book/ Paper
 - 2.3. Description - at least 10 words; should begin with the verb and with capital, e.g. "Plots the time series ..."
 - 2.4. Keywords (at least 5 words; the more the merrier) - Choose from list of keywords
 - 2.5. *See also* - list related Quantlets
 - 2.6. Author - check at the list of authors on the website [if new, than write [New] in this field]
 - 2.7. Submitted
 - 2.8. Datafile
 - 2.9. *Input (optional)* Should contain some new info, which is not written in other meta-info fields
 - 2.10. *Output (optional)* Should contain some new info, which is not written in other meta-info fields
 - 2.11. Example - check whether there is appropriate info on the website. Should contain some new info, which is not written in other meta-info fields

Examples of validated Q's

- http://quantnet.wiwi.hu-berlin.de/index.php?p=SFE_hq
- http://quantnet.wiwi.hu-berlin.de/index.php?p=SFS_hq

Styleguide

0.1 Complete and ordered Metainfo.txt

The provided information is critical for the availability, presence and relevance of your submitted Quantlet. These information will be used for clustering, filtering and recommendation engines. With accurate and dense information

you can help other users to find your QuantLet. Because YAML is used as a formatting language, the provided metainfo.txt must look similar in style to the example below.

```
1 Name of QuantLet : SFEustb
2
3 Published in : Statistics of Financial Markets
4
5 Description : 'Shows a 3-month U.S. Treasury bill daily
6               yield from 1998 to 2008 as an approximation of the
7               short rate.'
```

0.2 Use FormatR to clean up your code

You can easily preprocess your code with the FormatR package. Below is an example of how this is done.

```
1 # Cleaning up the source code in an R script file
2 # "input.R",
3 # Indentation is set to two space characters. Maximum
4 # line width is 80 characters.
5 # The formatted code is written into a new script file
6 # "output.R"
7
8 tidy_source(source = "input.R", indent = 2, width.cutoff = 80,
9 file = "output.R")
10
11 # similar to the previous example, but using the
12 # clipboard instead of an input file
13
14 tidy_source(indent = 2, width.cutoff = 80, file = "output.R")
15
16 # when omitting function parameters the defaults
17 # indent = 4 and width.cutoff = 80 are being used.
18 # For simplicity, we recommend these for use on Quantnet.
19
20 tidy_source(file = "output.R")
```

More details of the package FormatR are available in the package documentation

0.3 Change all "<-" with "="

A QuantNet specific style requirement concerns the assignment operator. All "<-" should be replaced with "=" like shown below.

```
1 #BAD
2 foo <- 5.0
3 bar <- function(x) {
4     return x^2
5 }
6
7 #GOOD
8 foo = 5.0
9 bar = function(x) {
10     return x^2
11 }
```

0.4 Align assignments in subsequent lines by "="

```
1 foo      = 5.0
2 foobar   = 7.0
3 bar      = 8.0
```

0.5 Set four space characters per indentation level *it should be done automatically by formatR*

```
1 while (i <n){
2     i = i + 1
3 }
```