Generates a histogram plot

Description

This function plots a non-disclosive histogram

Usage

```
ds.histogram(x = NULL, type = "split", num.breaks = 10,
  method = "smallCellsRule", k = 3, noise = 0.25,
  vertical.axis = "Frequency", datasources = NULL)
```

Arguments

`x`	a charcater, the name of the vector of values for which the histogram is desired.
`type`	a character which represent the type of graph to display. If `type` is set to 'combine', a histogram that merges the single plot is displayed. Each histogram is plotted separately if `type` is set to 'split'.
`num.breaks`	a numeric specifying the number of breaks of the histogram. The default value is set to 10.
`method`	a character which defines which histogram will be created. If `method` is set to 'smallCellsRule' (default option), the histogram of the actual variable is created but bins with low counts are removed. If `method` is set to 'deterministic' the histogram of the scaled centroids of each k nearest neighbours of the original variable where the value of `k` is set by the user. If the `method` is set to 'probabilistic', then the histogram shows the original distribution disturbed by the addition of random stochastic noise. The added noise follows a normal distribution with zero mean and variance equal to a percentage of the initial variance of the input variable. This percentage is specified by the user in the argument `noise`.
`k`	the number of the nearest neghbours for which their centroid is calculated. The user can choose any value for k equal to or greater than the pre-specified threshold used as a disclosure control for this method and lower than the number of observations minus the value of this threshold. By default the value of k is set to be equal to 3 (we suggest k to be equal to, or bigger than, 3). Note that the function fails if the user uses the default value but the study has set a bigger threshold. The value of k is used only if the argument `method` is set to 'deterministic'. Any value of k is ignored if the argument `method` is set to 'probabilistic' or 'smallCellsRule'.
`noise`	the percentage of the initial variance that is used as the variance of the embedded noise if the argument `method` is set to 'probabilistic'. Any value of noise is ignored if the argument `method` is set to 'deterministic' or 'smallCellsRule'. The user can choose any value for noise equal to or greater than the pre-specified threshold 'nfilter.noise'. By default the value of noise is set to be equal to 0.25.
`vertical.axis,`	a character which defines what is shown in the vertical axis of the plot. If `vertical.axis` is set to 'Frequency' then the histogram of the frequencies is returned. If `vertical.axis` is set to 'Density' then the histogram of the densities is returned.
`datasources`	a list of opal object(s) obtained after login in to opal servers; these objects hold also the data assign to R, as `dataframe`, from opal datasources.

Details

It calls a datashield server side function that produces the histogram objects to plot. Two options are possible as identified by the argument `method`. The first option creates a histogram that excludes bins with counts smaller than the allowed threshold. The second option creates a histogram of the centroids of each k nearest neighbours. The function allows for the user to plot disctinct histograms (one for each study) or a combine histogram that merges the single plots.

Value

one or more histogram objects and plots depending on the argument `type`

Author(s)

Amadou Gaye, Demetris Avraam for DataSHIELD Development Team

Examples

```
## Not run:
```

```
  # load that contains the login details
  data(logindata)

  # login to the servers
  opals <- opal::datashield.login(logins=logindata, assign=TRUE)

  # Example 1: generate a histogram for each study separately (the default behaviour)
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', type="split")

  # Example 2: generate a combined histogram with the default small cells counts
               suppression rule
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='smallCellsRule', type='combine')

  # Example 3: if a variable is of type factor then the function returns an error
  ds.histogram(x='LD$PM_BMI_CATEGORICAL')

  # Example 4: generate a combined histogram with the deterministic method
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='deterministic', type='combine')

  # Example 5: same as Example 4 but with k=50
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', k=50, method='deterministic', type='combine')

  # Example 6: same as Example 4 but with k=1740 (here we see that as k increases we have
               big utility loss)
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', k=1740, method='deterministic', type='combine')

  # Example 7: same as Example 6 but for split analysis
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', k=1740, method='deterministic', type='split')

  # Example 7: if k is less than the pre-specified threshold then the function returns an error
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', k=2, method='deterministic')

  # Example 8: generate a combined histogram with the probabilistic method
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='probabilistic', type='combine')

  # Example 9: generate a histogram with the probabilistic method for each study separately
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='probabilistic', type='split')

  # Example 10: same as Example 9 but with higher level of noise
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='probabilistic', noise=0.5, type='split')

  # Example 11: if 'noise' is less than the pre-specified threshold then the function returns
                an error
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='probabilistic', noise=0.1, type='split')

  # Example 12: same as Example 9 but with bigger number of breaks
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='probabilistic', type='split', num.breaks=30)

  # Example 13: same as Example 12 but the vertical axis shows densities instead of frequencies
  ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='probabilistic', type='split', num.breaks=30,
                vertical.axis='Density')

  # Example 14: create a histogram and the probability density on the plot
  hist <- ds.histogram(x='LD$PM_BMI_CONTINUOUS', method='probabilistic', type='combine',
                       num.breaks=30, vertical.axis='Density')
  lines(hist$mids, hist$density)

  # clear the Datashield R sessions and logout
  opal::datashield.logout(opals)


## End(Not run)
```
