DataIO
Manual

The package DataIO provides easy functions for writing and reading *.data Files. The *.data file format enables the user to import data to Microsoft Excel, Mathworks Matlab and allows to inspect and edit data with every text editor. This Manual describes how to use the functions for writing and reading *.data files.

General informations
A *.data file is an specific file format optimized for the needs of matrices. To prevent failure which can occur while altering the *.data file in an editor or they occur when transmitting the *.data files are saved with checksums for each column.
To save the data a data.frame is used, so it is possible to store strings and numbers within the same .data file. DataIO also provides simple functions for writing and reading plain text files (*.txt).

Easy examples for WriteData
The next few examples describe the basic usage of the function WriteData.

```
mat = matrix(
   c(1,0,0,0,1,0,0,0,1),
   nrow=3,
   ncol=3)

WriteData(DataFilename = "test.data", Data = mat)
```

DataFilename is the name of the file, which *WriteData* will create. If this file already exists it will be overwritten. It is also possible to write a data.frame:

```
data1 = c(1,2,3,4,5)
data2 = c("a","b","c","d","e")
data = data.frame(data1,data2)

WriteData(DataFilename = "test.data", Data = data)
```

Each column name will be used to generate the header for the *.data file. A
To add a comment, say a simple description of the data or a disclaimer, just add a comment attribute. The comment should be a string.

```
comment = "File for testing the comment function"
WriteData(DataFilename = "test.data",Data = mat,Comment = comment)
```

For the use of DataDefined, which is a further description of each column and its data make sure you have the correct length of the DataDefined vector. Numbers are stored using DataDefined = 9,0,1. Strings are stored as Names using DataDefined = 6,7. DataDefined = 2,3 and 8 are used for internal purposes.
Make also sure that you have only one column selected as the Key column (9). The key column is a unique column of integer values. One value codes exactly one data record, where this one line of data equals one observation.

There is also a attribute for adding description and names for each row. Use the names attribute to name each row.

```
names = c("row 1","row2","row3")
mat = matrix(
  c(1,0,0,0,1,0,0,0,1),
  nrow=3,
  ncol=3)

WriteData(DataFilename = "test.data",Data = mat, Names = names)
```

Or use Description for a detailed description of each row

```
description = c("description for row 1","description for row 2", "description for row 3")
mat = matrix(
 c(1,0,0,0,1,0,0,0,1),
 nrow=3,
 ncol=3)

WriteData(DataFilename = "test.data",Data = mat, Description = description)
```

Writing data with a large amount of decimal places is also possible.

```
data1 = c(1.0000000000002,2,3,4,5)
data2 = c("a","b","c","d","e")
data = data.frame(data1,data2)

WriteData(DataFilename = "test.data",Data = data)
```

This will cause the function round the data to a maximum of 13 decimal places. If this happen there is a warning that the data is round. A way to avoid rounding is casting the data into a character form.

```
data1 = c(1.987654321987654321,2,3,4,5)

data_as_char = as.character(data1);

data2 = c("a","b","c","d","e")
data = data.frame(data_as_char,data2)

WriteData(DataFilename = "test.data",Data = data)
```

If you use this method,

Easy examples for ReadData
The next few examples describe the basic usage of the function \code{ReadData}
The output of ReadData is a list of length 6. Where

```
ReadData(DataFilename="test.data")
```

This will read the *.data file from the current Directory (*getwd()*). To read a file from a different directory add a DataDirectory attribute.

```
ReadData(DataFilename="test.data",DataDirectory="~/Desktop") #for macOSX
ReadData(DataFilename="test.data",DataDirectory="C:/folder") #for windows
```

Open a *.data file in an external Editor

To open a *.data in Excel click in the Excel menu open, select all files and select the *.data file.
Next the "Text Import Wizard" window will open. *.data file are tab stop separated. Also make sure that for each column a point is selected as decimal separator. : it is sufficient to click the finish button. The data is now visible in Excel.

To open a *.data inside an Text editor perform a right click onto your file. Select open with and choose your favorite editor.

After altering a *.data file inside an external Editor is very importend to verify the checksums again.
This avoids warn messages if you read the *.data file in R using ReadData.
To verify a *.data file simply use verifyData.

```
verifyData(DataFilename="test.data")
```

Easy Examples for WriteTXT
This function is a easy method to write string matrices in simple plain text files (*.txt).

```
mat = matrix(
c("a","b" ,"c", "d", "e", "f", "g","h","i"), nrow=3, ncol=3)

WriteTXT(FileName= "text.txt", Names = mat)
```

Easy examples for ReadTXT

To read a txt file containing a string matrix use the function ReadTXT.

```
ReadTXT(FileName="test.txt")
```

This call will read a *.txt file from the current directory.

Error descriptions and solutions

Error Message	Solution
arguments imply differing number of rows:	Check the vector used for names and/or description
Length of Key isnt equal to the length of rows	Check the Key vector
Numbers in Data exceed the maximum number of digits. The Data will be round bevor writing in column(s): 2	Cast the column to a string column so the data wont be round
wrong number of ColumnNames: is n, but has to be i	Check if the Header contains enough elements (including a name for the Key)
In ReadData("test.data") : Incorrect Checksum for Column: n	Your file is broken. Or you altered the *.data file by hand. Use verifyData to correct the checkums