Quantitative Risk

Financial and Data Quality Toolkit

Contents

[**Application Overview** 2](#_Toc36824925)

[**Navigating the Application** 2](#_Toc36824926)

[**1.** **Data Cleaning** 2](#_Toc36824927)

[**2.** **Statistical Overview** 2](#_Toc36824928)

[**3.** **Analysis** 2](#_Toc36824929)

[**4.** **Statistics per Sector** 2](#_Toc36824930)

[**5.** **Outlier Analysis** 2](#_Toc36824931)

[**6.** **Correlation Matrix** 3](#_Toc36824932)

[**7.** **Company Analysis** 3](#_Toc36824933)

[**8.** **Population Stability Index** 3](#_Toc36824934)

[**9.** **Macroeconomic Indicators** 3](#_Toc36824935)

[**10.** **Data Quality Report** 3](#_Toc36824936)

[**f\_all\_data\_wide** 4](#_Toc36824937)

[**f\_binded\_data** 4](#_Toc36824938)

[**f\_borrowers** 5](#_Toc36824939)

[**f\_columns** 6](#_Toc36824940)

[**f\_lagged\_df** 6](#_Toc36824941)

[**f\_last\_year** 7](#_Toc36824942)

[**f\_outliers** 8](#_Toc36824943)

[**f\_ratio** 8](#_Toc36824944)

[**f\_remove\_wrong** 9](#_Toc36824945)

[**f\_reshape** 10](#_Toc36824946)

[**f\_ui\_build** 11](#_Toc36824947)

[**f\_t\_ratio\_build** 11](#_Toc36824948)

[**f\_statistical\_overview** 12](#_Toc36824949)

[**f\_standardise\_colnames** 13](#_Toc36824950)

[**f\_standard\_size** 14](#_Toc36824951)

[**f\_standard\_format** 15](#_Toc36824952)

[**f\_standard\_clean\_final** 15](#_Toc36824953)

[**f\_server\_build** 16](#_Toc36824954)

# **Application Overview**

The Quantitative Risk financial and data quality tool allows the user to assess their financial data in terms of missing rate, stability over time and other techniques described below. The application can be run in R by running the “template\_package.R” script. In this process, the latest GTquantrisk package will automatically be installed and loaded in to the library for use. The user may wish to change the financial data to be analysed which is easily done by changing the file path location within the read\_csv() function. The app will automatically sort the data into a standard format based off the mapping data frame that is preloaded in the GTquantrisk package. If the column names of the raw data are not already located in the second column of the mapping document, the user can update the document using the cbind() function (or otherwise) within R (note any additions to the document must be added to the furthest right column in order for f\_standard\_clean\_final () function to change these column names to the names located in column 1 of that same row). From there, the data is used to create 43 different financial ratios that can be further analysed within the app itself under.

# **Navigating the Application**

## **Data Cleaning**

The user can filter their data by enforcing a restriction on the maximum number of missing financial ratios per company. This is based off the missing percentage per company. If the user wants to exclude the companies that are missing 10% of their financial ratios then 0.1 should be selected on the slider and the Filter data frame button should be selected in the shiny side panel.

## **Statistical Overview**

The data can further be broken down for analysis by a range of years using the slider in the side panel of the app. A statistical overview table can be seen in the second tab of the application. Here figures such as the min, max, quartiles and mean per ratio can be viewed in order to get a feel for the data. Just below it, a missing analysis table id displayed. It shows the percentage of missing elements per ratio in order to assess the overall quality of the data. A report of useful findings can be produced by selecting Generate Report

## **Analysis**

The Analysis tab allows the user to select a financial factor in shiny side panel which they would like to analyse. Appearing on the dashboard of the application will be the definition of the selected factor, the risk direction (linked to the probability of default) and the formula and full formula behind the calculation of the financial factor. In the graph below this, we can see how the chosen financial factors percentiles have changed over time.

## **Statistics per Sector**

The statistics per sector tab gives the user the option to choose a sector and then specify a ratio in the side panel. Two data tables then appear, one with the overall summary statistics for the sector as a whole, and the other for each company in that sector. The figures appear in red or green depending whether they are above or below the mean.

## **Outlier Analysis**

A boxplot and distribution table, with and without outliers can be viewed in the Outlier Analysis tab. The user can select their desired type of ratio and the specified metric in the side panel and these tables will automatically update. The user can then go one further and filter the data to a specific sector and date range. A report of useful findings can be produced by selecting Generate Report.

## **Correlation Matrix**

A correction matrix of all the ratios can be viewed in the sixth tab. The user can filter the data down based on sector and/or type of ratio. This is useful for visualising highly correlated ratios which subsequently could be omitted from the variables used in a model.

## **Company Analysis**

The company analysis tab makes use of HTML scraping to access company data from the internet to show that company’s current share price. Below this, a graph of the user’s chosen ratio over time can be viewed for that company.

## **Population Stability Index**

The eighth tab contains the PSI calculation of a specified ratio through time. The stability across time of a ratio can be viewed in percentiles as well as at an overall level of the financial ratio. There are two methods used in the PSI calculation, the first one is the “Base” case where PSI is calculated from a specified starting year for all years going forward. This captures the stability of the ratio over a broad time from. The second PSI calculation is the “Rolling” case where each PSI figure is calculated based off the year before. This captures minor fluctuations in the stability of the ratio year on year that may not be captured in the “Base” case calculation.

## **Macroeconomic Indicators**

If yearly trends appear in the data for all sectors and companies, then macroeconomic factors may be influencing this. Both US and ECB inflation and interest rate history can be seen on a graph produced in the dashboard of the application.

## **Data Quality Report**

The final tab produces an overall Data Quality Diagnosis Report. The user has the option to choose a specific type of ratio and/or a specific sector grouping to be used in the report. The report displays overall statistics of the specified data along with a more detailed analysis of data outlier per variable. This report pops us in a separate window.

**Function Name**

# **f\_all\_data\_wide**

**Package**

GTquantrisk

**Description**

This function transforms wide data to long data and groups by the "symbol" (corporate tickers) column. To be used when initial financial data is long and want to transform it into wide data frame, of easy use in R. This function is later used in f\_binded\_data().

**Usage**

f\_all\_data\_wide(t\_input)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| data | The Dataset, ie a Data frame, the default is set to t\_definitions, contained in the package. | data <- t\_definitions |

**Function Name**

# **f\_binded\_data**

**Package**

GTquantrisk

**Description**

The function gets the lag of a data frame in R and binds the original factors with the lagged factors. For the Financial toolkit it column binds the outputs from f\_all\_data\_wide() and f\_lagged\_df(), so as to calculate trend ratios i.e – Net Margin Trend, Average Assets, EBITDA Trend etc.

**Usage**

f\_binded\_data(t\_input)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| t\_input | Data frame with financial information. | t\_input <- t\_financial\_data\_before\_lag |

**Function Name**

# **f\_borrowers**

**Package**

GTquantrisk

**Description**

Extracts list of borrowers from initial set of financial data to be used for data frames consistently throughout the Financial Tool. Returns a vector of the borrowers (i.e. companies) in alphabetical order to be binded onto final data frames.

**Usage**

f\_borrowers(data)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| data | The Dataset, ie a Data frame. | data <- t\_financial\_data |

**Function Name**

# **f\_columns**

**Package**

GTquantrisk

**Description**

Creates the names to be used for the final data frame. Uses the ratio names from the t\_definitions data frame and also includes "symbol" & "date". Returns a vector of the column names to be used for the final data frame.

**Usage**

f\_columns(data = t\_definitions)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| data | The Dataset, ie a Dataframe, the default is set to t\_definitions. | definitions= t\_definitions |

**Function Name**

# **f\_lagged\_df**

**Package**

GTquantrisk

**Description**

This function lags the financial data to the previous year’s values and joins these values on to the original data frame. This is done in order to calculate trend ratios. Returns the original data frame passed through it along with the lagged figures of that data frame.

**Usage**

f\_lagged\_df(t\_input)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| t\_input | Data frame of financial figures that you wish to be lagged by one year and amalgamated to the original data frame. | t\_input <-  t\_financial\_data |

**Function Name**

# **f\_last\_year**

**Package**

GTquantrisk

**Description**

Compute a lagged version of a time series, shifting the time base back by one year. Returns a time series object with the same class as x with lagged variables.

**Usage**

f\_last\_year(x)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| x | A vector or matrix or univariate or multivariate time series. | x <- t\_financial data |

**Function Name**

# **f\_outliers**

**Package**

GTquantrisk

**Description**

Extracts outliers per each financial ratio for use in Outlier Analysis tab of the Financial Tool shinyApp. Allows user to view the observations (companies / borrowers) considered as outliers. Returns a data frame with the rows that are considered outliers.

**Usage**

f\_outliers(dataframe)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| dataframe | A Dataset i.e. a data frame, from which the outliers are mapped per each financial ratio. Preferably it will be the data frame produced by f\_reshape as the ratios are subsetted by type. | dataframe <-  f\_reshape(t\_financial\_data) |

**Function Name**

# **f\_ratio**

**Package**

GTquantrisk

**Description**

This function takes a data frame containing all the financial information, and uses these figures to calculate financial ratios and then adds these ratios to an empty data frame (built by f\_t\_ratio\_build() function) to produce a data frame of all the possible financial ratios.

**Usage**

f\_ratio(df1, df2)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| df1 | Empty data frame built in f\_t\_ratio\_build() with column names labelled as all the financial ratios. | df1 <-  f\_t\_ratio\_build(data1,  data2 = t\_definitions) |
| df2 | Dataframe built in f\_lagged\_df() containing all the financial figures as well as the lagged figures. | df2 <-  f\_lagged\_df(t\_financial\_data) |

**Function Name**

# **f\_remove\_wrong**

**Package**

GTquantrisk

**Description**

Loops through each company’s financial data, calculates the median per each value and removes values that are 50 times larger than the median value. Returns a data frame with values deemed incorrect (50 times larger than median value) as NA.

**Usage**

f\_remove\_wrong(t\_input)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| t\_input | The Dataset, ie a data frame | t\_input <- t\_financial\_data |

**Function Name**

# **f\_reshape**

**Package**

GTquantrisk

**Description**

Transforms wide data frame produced by f\_ratio into a long data frame. Adds the type of financial ratio as a column which allows for sub setting of ratio by the type also makes it easy to navigate through multiple line ggplot calls used throughout the shinyapp. Returns a long data frame with each ratio value as a row.

**Usage**

f\_reshape(data1, data2 = t\_definitions)

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| data1 | A Dataset to be used. This will be the dataset that is produced by f\_ratio. This dataset will be made longer. | data1 <-  f\_ratio(df1, df2) |
| data2 | A Dataset to be used. This dataset will be the one with the names of the financial ratios and their type. The default is set to t\_definitions. | data2 <-  d\_definitions |

**Arguments**

**Function Name**

# **f\_ui\_build**

**Package**

GTquantrisk

**Description**

Creates a user interface for the shinyApp call in order to build the Financial Tool. Combination of conditional sidepanels and the layout of the body of the Application. Specifies the inputs to be used in the server and the format of the server outputs.

**Usage**

f\_ui\_build(data1 = t\_ratio, data2 = t\_definitions)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| data1 | A Dataset, i.e a Dataframe. Purpose of this dataframe is to create the grouping list to be included as inputs into the server as well as the date ranges for the slider inputs on various sidepanels. Preferably the output of f\_t\_ratio\_build(). | data1 = t\_ratio |
| data2 = t\_definitions | A Dataset, i.e a Dataframe. Purpose is to generate the list of ratio types to be selected by user on conditional sidepanels. Default is set to t\_definitions. | data2 = t\_definitions |

**Function Name**

# **f\_t\_ratio\_build**

**Package**

GTquantrisk

**Description**

Contains row and column strucutre to be used in f\_ratio.

**Usage**

f\_t\_ratio\_build(data1 = t\_financial\_data, data2 = t\_definitions)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| data1 | The first Dataset to be used. The names of the rows (borrowers) to be used for final dataframe. | data1 = t\_financial\_data |
| data2 = t\_definitions | The second Dataset to be used. The names of the columns (ratios along with symbol & date) to be used for the final dataframe. The default is set to the dataframe t\_definitions. | data2 = t\_definitions |

**Function Name**

# **f\_statistical\_overview**

**Package**

GTquantrisk

**Description**

This function produces interesting summary statistics about the financial ratios from the dataframe that have been built from f\_ratio(). The dataframe can be subsetted based on year, this can be manually inputted when used outside the Rshiny app or automatically inputted through the use of a date slider within the Rshiny app.

**Usage**

f\_statistical\_overview(year1,year2,data)

**Arguments**

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| year1 | The beginng year for which to subset the data. | year1= 2005 |
| year2 | The end year for which to subset the data. | year2= 2015 |
| data | Dataframe for which you want to create summary statistics from column 3 on ("symbol" and "date" should be the first two columns). | data = t\_financial\_data |

**Function Name**

# **f\_standardise\_colnames**

**Package**

GTquantrisk

**Description**

This function takes raw data and the mapping document to produce a dataframe with column names labelled as per the mapping document. Column names in the raw data will change to the names in the mapping document (located at [i,1]) as long as the column names in the raw data exactly match the elements of the mapping document (located at [1,2:ncol(mapping)]).

**Usage**

f\_standardise\_colnames(t\_input, t\_map = mapping)

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| t\_input | Dataframe of raw data. | t\_input = t\_financial\_data |
| t\_map | Mapping document containing both the dataset column names and the financial factor names in standard format. | t\_map = mapping |

**Arguments**

**Function Name**

# **f\_standard\_size**

**Package**

GTquantrisk

**Description**

This function takes a dataframe and the mapping document to produce a dataframe of standard size where the number of column names equals number of mapping variables. This is done by removing the column names in the dataframe that aren't in the first column of the mapping document. Any excess names in the mapping document that aren't in the dataframe are joined to the dataframe as an empty vector (Note: this function should be run on the dataframe produced by the f\_standardise\_colnames() in order to minimise the amount of columns being removed, thus, retaining the maximum possible data).

**Usage**

f\_standard\_size(t\_input, t\_map = mapping)

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| t\_input | Dataframe produced from f\_standardise\_colnames(). | t\_input = t\_financial\_data |
| t\_map | Mapping document containing both the dataset column names and the financial factor names in standard format. | t\_map = mapping |

**Arguments**

**Function Name**

# **f\_standard\_format**

**Package**

GTquantrisk

**Description**

This function takes a dataframe and the mapping document to produce a dataframe of standard format where "symbol" and "date" are the first and second columns followed by all other financial factors (Note: this function should be run on the dataframe produced by the f\_standardise\_colnames() and then f\_standard\_size() in order to ensure that the "symnbol" and "date" columns are in the conventional naming format).

**Usage**

f\_standard\_format(t\_input, t\_map = mapping)

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| t\_input | Dataframe produced from f\_standardise\_colnames() and then further passed through f\_standardise\_colnames(). | t\_input = t\_financial\_data |
| t\_map | Mapping document containing both the dataset column names and the financial factor names in standard format. | t\_map = mapping |

**Argument**

**Function Name**

# **f\_standard\_clean\_final**

**Package**

GTquantrisk

**Description**

This function takes raw data and the mapping document to produce a dataframe with column names labelled as per the mapping document, of standard size where the number of column names equals number of mapping variables and of standard format where "symbol" and "date" are the first and second columns followed by all other financial factors. It utilises f\_standardise\_colnames(), f\_standard\_size(), f\_standard\_format() at once.

**Usage**

f\_standard\_clean\_final(t\_input, t\_map = mapping)

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| t\_input | Dataframe of raw data. | t\_input = t\_financial\_data |
| t\_map | Mapping document containing both the dataset column names and the financial factor names in standard format. | t\_map = mapping |

**Arguments**

**Function Name**

# **f\_server\_build**

**Package**

GTquantrisk

**Description**

Creates the server to be used in the Financial Tool shinyApp. Responds to the inputs from the user interface and produces outputs specified by the user interface.

**Usage**

f\_server\_build(def\_table = t\_definitions, final\_table, joined\_table)

|  |  |  |
| --- | --- | --- |
| Label | Description | Example |
| def\_table | A Dataset, i.e a dataframe. Used to subset the conditional sidepanels based on user choice of variable type. The default setting is t\_definitions. | def\_table = t\_definitions |
| final\_table | A Dataset, i.e a dataframe. Used as a master dataframe from which reactive dataframes are made throughout the server. Comes from the output of f\_reshape() and will therefore be a longer dataframe than joined\_table. | final\_table |
| joined\_table | A Dataset, i.e a dataframe. Used as a master dataframe from which reactive dataframes are made throughout the server. Comes from the output of f\_ratio and will contain a column for the grouping of each observation (company / borrower) in the population. | joined\_table |

**Arguments**