Package: rmsfuns (via r-universe)

August 31, 2024

Title Quickly View Data Frames in 'Excel', Build Folder Paths and Create Date Vectors

Version 1.0.0.1

Description Contains several useful navigation helper functions, including easily building folder paths, quick viewing dataframes in 'Excel', creating date vectors and changing the console prompt to reflect time.

Maintainer Nico Katzke <nfkatzke@gmail.com>

BugReports https://github.com/nicktz/rmsfuns/issues

Imports utils, readr, purrr, magrittr, dplyr, tbl2xts, PerformanceAnalytics, xts, zoo

Depends R (>= 3.2.1)

License GPL-3

URL https://rmsfuns.nfkatzke.com

LazyData TRUE

RoxygenNote 6.1.1

Suggests lubridate, knitr, rmarkdown

NeedsCompilation yes

ByteCompile yes

VignetteBuilder knitr

Repository https://nicktz.r-universe.dev

RemoteUrl https://github.com/nicktz/rmsfuns

RemoteRef HEAD

RemoteSha 5b8d91c4dc209a9595118ec88ecb1cf07390269e

Contents

`build_path` *build_path*

Description

build_path builds the entire folder FilePath provided. If the FilePath does not exist, it builds it without error. It is effectively an extension to the base function dir.create.

Usage

```
build_path(FilePath, Silent = TRUE)
```

Arguments

`FilePath`	A character string with the target folder path. This can be a vector of multiple paths, e.g.: FilePath <- paste0("C:/TestFolder/", c("Subfolder1","Subfolder2"))
`Silent`	True by default, if set to FALSE it shows the address of the folder just created. This is, however, saved if used as: Path <- build_path(FilePath), making the message largely redundant.

Value

Path address just built.

Examples

```
## Not run:
PathLoc <- tempdir()
Path <- build_path(PathLoc)
Pathmultiplecreate <- build_path(file.path(PathLoc, c("XXX", "YYY")))

## End(Not run)
```

dateconverter *dateconverter*

Description

dateconverter makes it easy to create a date vector in R. It offers a simple wrapper using xts functionality to create a vector of dates between a given Start and End date, and then correcting for the chosen frequency transformation.

Usage

```
dateconverter(StartDate, EndDate, Transform)
```

Arguments

StartDate	A valid as.Date object. This can be given as ymd("2000-01-01") or as.Date("2000-01-01")
EndDate	A valid as.Date object. This can be given as ymd("2000-01-01") or as.Date("2000-01-01")
Transform	This is the days that you want returned. Options include: alldays: All calendar days between the start and end date calendarEOM: Last calendar day of each month between the start and end date weekdays: All weekdays between the start and end date (mon - fri) weekdayEOW: All last weekdays between the start and end date weekdayEOM: All last weekdays of the month between the start and end date weekdayEOQ: All last weekdays of the quarter between the start and end date weekdayEOY: All last weekdays of the year between the start and end date

Value

Path address just built.

Examples

```
## Not run:
dateconverter(as.Date("2000-01-01"),
as.Date("2017-01-01"), "weekdays")
dateconverter(as.Date("2000-01-01"),
as.Date("2017-01-01"), "calendarEOM")
dateconverter(as.Date("2000-01-01"),
as.Date("2017-01-01"), "weekdayEOW")
dateconverter(as.Date("2000-01-01"),
as.Date("2017-01-01"), "weekdayEOM")
dateconverter(as.Date("2000-01-01"),
as.Date("2017-01-01"), "weekdayEOQ")
dateconverter(as.Date("2000-01-01"),
as.Date("2017-01-01"), "weekdayEOY")
dateconverter(as.Date("2000-01-01"),
```

```
as.Date("2017-01-01"), "alldays")

## End(Not run)
```

---

load_pkg *load_pkg*

Description

load_pkg Loads a list of packages. If a package requires installation, the function will install it from CRAN. Function is a CRAN only wrapper.

Usage

```
load_pkg(packagelist)
```

Arguments

packagelist	Vector of packages to load into R

Value

Packages loaded into R

Examples

```
## Not run: packagelist <- c("purrr", "readr")
load_pkg(packagelist)
## End(Not run)
```

PromptAsTime *PromptAsTime*

Description

This changes Rstudio's prompt at the bottom to reflect time. Particularly useful for timing functions when executing long scripts.

Usage

```
PromptAsTime(On)
```

Arguments

On	set On to TRUE (Add time to prompter) or FALSE (use default prompter).

Value

The Prompter in Rstudio will now include the time.

Examples

```
## Not run:
PromptAsTime(TRUE)
x <- 100
Sys.sleep(3)
#' x*x
print(x)

## End(Not run)
```

`Safe_Return.portfolio` *Safe_Return.portfolio*

Description

This provides a safe way to do portfolio return calculations. It ensures the returns and weights are explicitly mapped. It is thus a simple wrapper to PerformanceAnalytics::Return.portfolio making it safer. See the following gist for a discussion on why this safety feature is essential: https://gist.github.com/Nicktz/a24ba1775d

Usage

```
Safe_Return.portfolio(R, weights, lag_weights = TRUE, ...)
```

Arguments

`R`	xts of returns.
`weights`	xts of weights.
`lag_weights`	This function makes weights effective on the day it is given. The Return.Portfolio function defaults to having weights become effective only the following day after its specification. E.g. from the vignette: Rebalancing periods can be thought of as taking effect immediately after the close of the bar. So, a March 31 rebalancing date will actually be in effect for April 1.
`...`	parameter inputs from PerformanceAnalytics::Return.portfolio.

Examples

```
## Not run:
library(PerformanceAnalytics)
data(edhec)
data(weights) # rebalance at the beginning of the year to various weights through time
x <- Safe_Return.portfolio(edhec[,1:11], weights=weights, lag_weights = TRUE, verbose=TRUE)

## End(Not run)
```

ViewXL *ViewXL*

Description

Views a data.frame or tbl_df object in excel, by saving it in R's temporary file directory (see: tempdir()). Works on Windows or Mac OS - not linux It will automatically open the excel sheet. User has the choice too of overriding the file location by setting the FilePath directly. It is recommended to save the output and use 'unlink' to delete afterwards. See ?ViewXL

Usage

```
ViewXL(DataFrame, FilePath, FileName, ViewTempFile = TRUE, mac = FALSE)
```

Arguments

DataFrame	This is the dataframe or tbl_df that will be displayed in excel
FilePath	If left blank, tempfile will be used. If specified, the excel files will be saved in specified location.
FileName	If specified to save csv file, this would be the name. If left blank and a FilePath has been specified, it would prompt the user to add a FileName.
ViewTempFile	True by default, if False it will not open the excel file, but merely save it. Only useful if provided with a FilePath.
mac	FALSE by default, set to TRUE if using a Mac, else the base::shell.exec will not work.

Value

File location in promt. Chosen data frame or tbl_df opened directly in excel.

Examples

```
## Not run:
df <- data.frame( date = seq(
as.Date("2012-01-01"),
as.Date("2015-08-18"),"day"),
x = rnorm(1326, 10,2))
x <- ViewXL(df)
# After viewing, it is recommended to delete the temporary file created using:
unlink(x)

## End(Not run)
```

Index