'Roxygen documentation for file util_funs.R '

January 3, 2022

R topics documented:

`AtoQ` *Linear interpolation based on aremos command reference page 292*

Description

Linear interpolation based on aremos command reference page 292

Usage

```
AtoQ(ser_in, aggr = "mean")
```

Arguments

`ser_in`	the xts series to be interpolated (freq = a)
`aggr`	interpolation method: aggregate via mean (default) or sum

Value

interpolated xts series (freq = q)

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2021"] <- c(323127513, 325511184, 327891911, 330268840, 332639102, 334998398)
test1 <- AtoQ(`ncen@us.sola`)
```

---

explode_xts *Splitting of xts matrix to individual xts vectors (don't use, pollutes global environment)*

Description

Splitting of xts matrix to individual xts vectors (don't use, pollutes global environment)

Usage

```
explode_xts(xts_in)
```

Arguments

xts_in	the xts matrix to be split into individual xts vectors

Value

nothing (silently store split series in global environment)

Examples

```
get_series_exp(74, save_loc = NULL) %>%
  ts_long() %>%
  ts_xts() %>%
  explode_xts()
rm(list = ls(pattern = glob2rx("*@HI.Q")))
```

---

find_end *Find the date of the last observation (NAs are dropped)*

Description

Find the date of the last observation (NAs are dropped)

Usage

```
find_end(ser_in)
```

Arguments

ser_in	an xts series

Value

date associated with last observation

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2060, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2018"] <- c(323127513, 325511184, 327891911)
find_end(`ncen@us.sola`)
```

---

find_start *Find the date of the first observation (NAs are dropped)*

Description

Find the date of the first observation (NAs are dropped)

Usage

```
find_start(ser_in)
```

Arguments

ser_in	an xts series

Value

date associated with first observation

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2017/2021"] <- c(325511184, 327891911, 330268840, 332639102, 334998398)
find_start(`ncen@us.sola`)
```

---

get_series *Download a set of series from udaman using series names*

Description

Download a set of series from udaman using series names

Usage

```
get_series(ser_id_vec)
```

Arguments

ser_id_vec	vector of series names

Value

time and data for all series combined in a tibble

Examples

```
get_series(c("VISNS@HI.M", "VAPNS@HI.M"))
```

---

get_series1 *Download a single series from udaman using series name*

Description

Download a single series from udaman using series name

Usage

```
get_series1(ser_id)
```

Arguments

ser_id	udaman series name

Value

time and data for a single series combined in a tibble

Examples

```
get_series("VISNS@HI.M")
```

---

get_series_exp *Download series listed in an export table from udaman*

Description

Download series listed in an export table from udaman

Usage

```
get_series_exp(exp_id, save_loc = "data/raw")
```

Arguments

exp_id	export id
save_loc	location to save the csv of the retrieved data, set to NULL to avoid saving

Value

time and data for all series combined in a tibble

Examples

```
get_series_exp(74)
get_series_exp(74, save_loc = NULL)
```

---

get_var *Construct a series name from variable components and retrieve the series*

Description

Construct a series name from variable components and retrieve the series

Usage

```
get_var(ser_in, env = parent.frame())
```

Arguments

ser_in	a variable name (string with substituted expressions)
env	environment where the expression should be evaluated

Value

variable

Examples

```
ser_i <- "_NF"
cnty_i <- "HI"
get_series_exp(74, save_loc = NULL) %>%
  ts_long() %>%
  ts_xts() %$% get_var("E{ser_i}@{cnty_i}.Q")
```

---

make_xts *Create xts and fill with values*

Description

Create xts and fill with values

Usage

```
make_xts(start = bnk_start, end = bnk_end, per = "year", val = NA)
```

Arguments

start	date of series start (string: "yyyy-mm-dd")
end	date of series end (string: "yyyy-mm-dd")
per	periodicity of series (string: "quarter", "year")
val	values to fill in (scalar or vector)

Value

an xts series

Examples

```
make_xts()
make_xts(start = ymd("2010-01-01"), per = "quarter", val = 0)
```

---

p *Concatenate dates to obtain period*

Description

Concatenate dates to obtain period

Usage

```
p(dat1, dat2)
```

Arguments

dat1	date of period start (string: yyyy-mm-dd)
dat2	date of period end (string: yyyy-mm-dd)

Value

string containing date range

Examples

```
p("2010-01-01", "2020-01-01")
```

---

pca_to_pc *Convert annualized growth to quarterly growth*

Description

Convert annualized growth to quarterly growth

Usage

```
pca_to_pc(ser_in)
```

Arguments

ser_in	the series containing annualized growth (in percent)

Value

series containing quarterly growth (in percent)

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2021"] <- c(323127513, 325511184, 327891911, 330268840, 332639102, 334998398)
test1 <- AtoQ(`ncen@us.sola`)
ts_c(test1 %>% ts_pca() %>% pca_to_pc(), test1 %>% ts_pc())
```

---

pchmy *Calculate multi-period average growth*

Description

Calculate multi-period average growth

Usage

```
pchmy(ser_in, lag_in = 1)
```

Arguments

ser_in	name of xts series for which growth is calculated
lag_in	length of period over which growth is calculated

Value

series containing the average growth of ser_in (in

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2021"] <- c(323127513, 325511184, 327891911, 330268840, 332639102, 334998398)
test1 <- AtoQ(`ncen@us.sola`)
ts_c(pchmy(`ncen@us.sola`, lag_in = 3), ts_pc(`ncen@us.sola`))
ts_c(pchmy(test1, lag_in = 4), ts_pcy(test1), ts_pca(test1), ts_pc(test1))
```

---

plot_1 *Interactive plot of a single variable with level and growth rate*

Description

Interactive plot of a single variable with level and growth rate

Usage

```
plot_1(
  ser,
  rng_start = as.character(Sys.Date() - years(15)),
  height = 300,
  width = 900
)
```

Arguments

ser	time series to plot
rng_start	start of zoom range ("YYYY-MM-DD")
height	height of a single panel (px)
width	width of a single panel (px)

Value

a dygraph plot

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2021"] <- c(323127513, 325511184, 327891911, 330268840, 332639102, 334998398)
test1 <- AtoQ(`ncen@us.sola`)
plot_1(`ncen@us.sola`, rng_start = "2017-01-01")
plot_1(test1, rng_start = "2017-01-01")
```

---

plot_comp *Three-panel plot of levels, index, and growth rates*

Description

Three-panel plot of levels, index, and growth rates

Usage

```
plot_comp(
  sers,
  rng_start = as.character(Sys.Date() - years(15)),
  rng_end = as.character(Sys.Date()),
  height = 300,
  width = 900
)
```

Arguments

sers	a vector of series to plot
rng_start	start of the zoom range ("YYYY-MM-DD")
rng_end	end of the zoom range ("YYYY-MM-DD")
height	height of a single panel (px)
width	width of a single panel (px)

Value

a list with three dygraph plots (level, index, growth)

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2021"] <- c(323127513, 325511184, 327891911, 330268840, 332639102, 334998398)
test1 <- AtoQ(`ncen@us.sola`)
plot_comp(ts_c(`ncen@us.sola`, test1), rng_start = "2017-01-01")
get_series_exp(74, save_loc = NULL) %>%
  ts_long() %>%
  ts_xts() %>%
  extract(, c("E_NF@HI.Q", "ECT@HI.Q", "EMN@HI.Q")) %>%
  plot_comp()
```

---

plot_comp_2 *Two-panel plot of levels, index, and growth rates*

Description

Two-panel plot of levels, index, and growth rates

Usage

```
plot_comp_2(
  sers,
  rng_start = as.character(Sys.Date() - years(15)),
  rng_end = as.character(Sys.Date()),
  height = 300,
  width = 900
)
```

Arguments

sers	a vector of series to plot
rng_start	start of the zoom range ("YYYY-MM-DD")
rng_end	end of the zoom range ("YYYY-MM-DD")
height	height of a single panel (px)
width	width of a single panel (px)

Value

a list with two dygraph plots (level, index, growth)

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2021"] <- c(323127513, 325511184, 327891911, 330268840, 332639102, 334998398)
test1 <- AtoQ(`ncen@us.sola`)
plot_comp_2(ts_c(`ncen@us.sola`, test1), rng_start = "2017-01-01")
get_series_exp(74, save_loc = NULL) %>%
  ts_long() %>%
  ts_xts() %>%
  extract(, c("E_NF@HI.Q", "ECT@HI.Q", "EMN@HI.Q")) %>%
  plot_comp_2()
```

QtoA *Conversion from quarterly to annual frequency*

Description

Conversion from quarterly to annual frequency

Usage

```
QtoA(ser_in, aggr = "mean")
```

Arguments

ser_in	the xts series to be converted (freq = q)
aggr	aggregate via mean (default) or sum

Value

converted xts series (freq = a)

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2021"] <- c(323127513, 325511184, 327891911, 330268840, 332639102, 334998398)
test1 <- AtoQ(`ncen@us.sola`)
test2 <- QtoA(test1) # for stock type variables mean, for flow type variables sum
print(test1)
print(cbind(`ncen@us.sola`, test2))
```

QtoM *Interpolate a tibble of series from quaterly to monthly freq*

Description

Interpolate a tibble of series from quaterly to monthly freq

Usage

```
QtoM(data_q, conv_type)
```

Arguments

data_q	tibble containing variables at quarterly freq
conv_type	match the quarterly value via "first", "last", "sum", "average"

Value

tibble containing variables at monthly freq

Examples

```
`ncen@us.sola` <- ts(NA, start = 2016, end = 2021, freq = 1) %>% ts_xts()
`ncen@us.sola`["2016/2021"] <- c(323127513, 325511184, 327891911, 330268840, 332639102, 334998398)
test1 <- AtoQ(`ncen@us.sola`)
QtoM(ts_tbl(test1), "average")
ts_frequency(QtoM(ts_tbl(test1), "average") %>% ts_xts())
```

QtoM1 *Interpolate a single series from quarterly to monthly freq*

Description

Interpolate a single series from quarterly to monthly freq

Usage

```
QtoM1(var_q, ts_start, conv_type)
```

Arguments

var_q	vector containing a single variable at quarterly freq
ts_start	starting period as c(year, quarter) e.g. c(2001, 1)
conv_type	match the quarterly value via "first", "last", "sum", "average"

Value

vector containing a single variable at monthly freq

Examples

```
QtoM1(test1, c(2010, 1), "average")
```

---

qtrs *Convert period in quarters to period months*

Description

Convert period in quarters to period months

Usage

```
qtrs(nr_quarters)
```

Arguments

nr_quarters	number of quarters in period (integer)

Value

number of months in period

Examples

```
qtrs(3)
ymd("2020-01-01") + qtrs(3)
```