Package 'airutilities'

July 5, 2022

Title Support routines for fractional airline decomposition routines in the rjd3highfreq package

Version 3.1

Description Utilities that support the use of the fractional airline decomposition routines from Demetra version 3 from the rjd3highfreq package, including routines to generate outlier and holiday regressors, assign outlier and calendar regression based factors to different components (as in X-13ARIMA-SEATS), summarize estimated model coefficients, and generate factor plots and boxplots by week. Note that for some routines, series are assumed to be formatted by the tis package, and were developed for use with weekly time series. Plotting routines for weekly series are included, with a high density plotting function for components. Routines have been added that use xts time series objects.

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Imports checkmate, lubridate, rjd3highfreq, timeDate, tis, xts, zoo, RCurl, RProtoBuf, mathjaxr, rJava, rjd3modelling, rjd3sa, rjd3sts, rjd3toolkit

Depends R (>= 2.10)

NeedsCompilation no

Author Brian Monsell [aut, cre]

Maintainer Brian Monsell <Monsell.Brian@bls.gov>

R topics documented:

`cc_day_xts` *Day vector for Continued Unemployment Claims, weekly (cc), as an xts time series object*

Description

A xts time series object containing the day for every observation of the weekly continued unemployment claims released by the Department of Labor

Usage

```
cc_day_xts
```

Format

An xts time series object for CC from January 31, 2004 to January 29, 2022

Source

https://www.dol.gov/ui/data.pdf

`cc_month_xts` *Month vector for Continued Unemployment Claims, weekly (cc), as an xts time series object*

Description

A xts time series object containing the month number for every observation of the weekly continued unemployment claims released by the Department of Labor

Usage

```
cc_month_xts
```

Format

An xts time series object for CC from January 31, 2004 to January 29, 2022

Source

https://www.dol.gov/ui/data.pdf

`cc_sa_tc_log` *Seasonally Adjusted Series for Continued Unemployment Claims, weekly (automatic outlier identification, TC outlier specified, log)*

Description

A time series object with seasonal adjustment generated for weekly continued unemployment claims from a model fit to the logged series that includes automatically identified outliers with a temporary change outlier (TC) specified for the 13th week of 2020

Usage

```
cc_sa_tc_log
```

Format

An xts time series object for seasonally adjusted CC from January 31, 2004 to January 29, 2022

`cc_sa_tc_nolog` *Seasonally Adjusted Series for Continued Unemployment Claims, weekly (automatic outlier identification, TC outlier specified, no log)*

Description

A time series object with seasonal adjustment generated for weekly continued unemployment claims from a model fit to the original series that includes automatically identified outliers with a temporary change outlier (TC) specified for the 13th week of 2020

Usage

```
cc_sa_tc_nolog
```

Format

An xts time series object for seasonally adjusted CC from January 31, 2004 to January 29, 2022

`cc_week_xts` *Week vector for Continued Unemployment Claims, weekly (cc), as an xts time series object*

Description

A xts time series object containing the week number for every observation of the weekly continued unemployment claims released by the Department of Labor

Usage

```
cc_week_xts
```

Format

An xts time series object for CC from January 31, 2004 to January 29, 2022

Source

https://www.dol.gov/ui/data.pdf

`cc_year_xts` *Year vector for Continued Unemployment Claims, weekly (cc), as an xts time series object*

Description

A xts time series object containing the year for every observation of the weekly continued unemployment claims released by the Department of Labor

Usage

```
cc_year_xts
```

Format

An xts time series object for CC from January 31, 2004 to January 29, 2022

Source

https://www.dol.gov/ui/data.pdf

gen_air_components *Generate signal extraction components*

Description

Generates components from the decomposition of the fractional airline model R routines, partioning regression effects to different components as in X-13ARIMA-SEATS

Usage

```
gen_air_components(
  this_est,
  this_decomp,
  this_xtype,
  this_x = NULL,
  this_log = TRUE,
  this_stde = FALSE,
  tc2trend = FALSE,
  remove_user = FALSE
)
```

Arguments

this_est	A list object generated by the estimation procedure of the fractional airline model
this_decomp	A list object generated by the decomposition procedure of the fractional airline model
this_xtype	Character vector; Type of regressor used for including regression effects into components ('hol' for holiday, 'ao' for point outliers, 'ls' for level change outliers).
this_x	Character vector; regression. Default: use the regression matrix in this_est$model$X.
this_log	Logical scalar; set to TRUE if the log transformation is used in the decomposition, FALSE otherwise. Default: TRUE
this_stde	Logical scalar; component standard error included in this_decomp. Default: FALSE.
tc2trend	Logical scalar; TC outliers included in trend component if TRUE. Default: FALSE, TC outliers included in irregular.
remove_user	Logical scalar; remove user-defined regression factors from seasonally adjusted series if TRUE. Default: FALSE, user defined regression factors included in seasonally adjusted series.

Value

List of components from the fractional airline decomposition, partioning regression effects to different components as in X-13ARIMA-SEATS

Examples

```
ic_est <-
   rjd3highfreq::fractionalAirlineEstimation(ic_obs, periods=c(365.25/7),
                                        x=ic_default_matrix)
ic_decomp <-
   rjd3highfreq::fractionalAirlineDecomposition(ic_est$model$linearized,
                                          365.25/7, stde = TRUE)
ic_xtype <- c(rep('hol', 13), rep('ao', 40))
ic_comp <-
   gen_air_components(ic_est, ic_decomp,
                      this_xtype = ic_xtype,
                      this_log = FALSE,
                      this_stde = TRUE)
ic_comp_tis <- lapply(ic_comp, function(x)
   try(tis::tis(x, start = ic_start, tif = 'wsaturday')))
```

---

gen_air_model_matrix *Fractional airline model summary*

Description

Generates a matrix with coefficient estimates from the fractional airline model, generated by the JDemtetra+ high dimension modeling R routines

Usage

```
gen_air_model_matrix(
  this_est,
  xreg_names = NULL,
  this_week = NULL,
  this_year = NULL
)
```

Arguments

this_est	A list object generated by the estimation procedure of the fractional airline model
xreg_names	Character vector; Names of the user-defined regression variables.
this_week	Numeric vector; Week of each observation. Default: NULL
this_year	Numeric vector; Year of each observation. Default: NULL

Value

Generate matrix of model parameters estimated for the fractional airline model

Examples

```
ic_default_est <-
  rjd3highfreq::fractionalAirlineEstimation(ic_obs, periods=c(365.25/7), x=ic_default_matrix)
ic_default_model <-
   gen_air_model_matrix(ic_default_est,
                        xreg_names = colnames(ic_default_matrix),
                        this_week = ic_week, this_year = ic_year)
```

`gen_air_projected_factors` *Generate projected seasonal (and holiday) adjustment factors*

Description

Generates projected adjustment factors from the decomposition of the fractional airline model R routines

Usage

```
gen_air_projected_factors(
  this_est,
  this_decomp,
  this_xtype,
  this_x = NULL,
  this_log = TRUE,
  nfcasts = 104,
  return_series = "combined",
  return_xts = FALSE
)
```

Arguments

`this_est`	A list object generated by the estimation procedure of the fractional airline model; the regression matrix should not be extended with forecasts.
`this_decomp`	A list object generated by the decomposition procedure of the fractional airline model; the `nfcasts` option should have been specified.
`this_xtype`	Character vector; Type of regressor used for including regression effects into components (`'hol'` for holiday, `'ao'` for point outliers, `'ls'` for level change outliers).
`this_x`	Character vector; regression matrix.
`this_log`	Logical scalar; set to TRUE if the log transformation is used in the decomposition, FALSE otherwise. Default: TRUE
`nfcasts`	Integer scalar; number of projected adjustment factors; should match the `nfcasts` argument used to generate the decomposition.
`return_series`	Character scalar; component returned by the routine. Default is "combined"; other possible choices are "seasonal" and "holiday".
`return_xts`	Logical scalar; return projected factors of type `return_series` as an `xts` time series object. Default is FALSE. If TRUE, `this_x` must be an `xts` time series object.

Value

Array of projected adjustment factors from the fractional airline decomposition

Examples

```
ic_est <-
   rjd3highfreq::fractionalAirlineEstimation(ic_obs, periods=c(365.25/7),
                                             x=ic_default_matrix)
ic_decomp <-
   rjd3highfreq::fractionalAirlineDecomposition(ic_est$model$linearized, 365.25/7,
                                                stde = TRUE, nfcasts = 104)
ic_xtype <- c(rep('hol', 13), rep('ao', 40))
ic_proj_seasonal <-
   gen_air_projected_factors(ic_est, ic_decomp,
                      this_xtype = ic_xtype,
                      this_x     = ic_holiday_matrix_fcst,
                      this_log = FALSE, nfcasts = 104)
```

---

gen_hybrid_sa_xts *Generate a hybrid seasonal adjustment of an xts time series object*

Description

Generates a "hybrid" seasonal adjustment by replacing a span of a multiplicative seasonal adjustment with an additive adjustment

Usage

```
gen_hybrid_sa_xts(
  this_mult_sa,
  this_add_sa,
  this_start_hybrid,
  this_end_hybrid,
  this_week_xts,
  this_year_xts
)
```

Arguments

`this_mult_sa`	xts time series object of a multiplicative seasonal adjustment
`this_add_sa`	xts time series object of an additive seasonal adjustment
`this_start_hybrid`	integer vector of length 2, start of the span where additive adjustments replace multiplicative adjustment, defined as `c(year,week)`
`this_end_hybrid`	integer vector of length 2, end of the span where additive adjustments replace multiplicative adjustment, defined as `c(year,week)`
`this_week_xts`	Numeric vector; Week of the year for each observation, stored as an `xts` time series object.
`this_year_xts`	Numeric vector; Year of each observation, stored as an `xts` time series object.

Value

xts series object with hybrid seasonal adjustment

Examples

```
cc_hybrid_sa <- gen_hybrid_sa_xts(cc_sa_tc_log, cc_sa_tc_nolog, c(2020,12), c(2020, 52),
                                  cc_week_xts, cc_year_xts)
```

`gen_level_outlier_matrix`

Generate level change regression matrix

Description

Generates a regression matrix of LS (level change) outliers to be included in weekly modeling routines

Usage

```
gen_level_outlier_matrix(
  outlier_dates,
  this_week,
  this_year,
  forecast = 0,
  air_name = TRUE,
  x13type = TRUE,
  return_xts = FALSE
)
```

Arguments

`outlier_dates`	Integer matrix - matrix of dates for LS outliers
`this_week`	Numeric vector; Week of the year for each observation.
`this_year`	Numeric vector; Year of each observation.
`forecast`	Numeric scalar; Number of forecasts. Default: 0.
`air_name`	Logical scalar; If TRUE, names are formatted as they are in the fractional airline routine; otherwise format them as in movereg. Default: TRUE
`x13type`	Logical scalar; Indicates if level change outlier is defined as in X-13ARIMA-SEATS. Default: TRUE.
`return_xts`	Logical scalar; return matrix as an `xts` time series object. Default is FALSE. If TRUE, `this_week` and `this_year` should be `xts` time series objects.

Value

Generate matrix of level change outlier regressors, with column names that describe the individual regressors

Examples

```
ic_level_dates <-
   matrix(c(12, 2020, 13, 2020), ncol=2, byrow=TRUE)
ic_level_matrix <-
   gen_level_outlier_matrix(ic_level_dates, ic_week, ic_year, 0,
                            return_xts = FALSE)
cc_level_dates <-
   matrix(c(12, 2020, 13, 2020), ncol=2, byrow=TRUE)
cc_level_matrix_xts <-
   gen_level_outlier_matrix(cc_level_dates, cc_week_xts, cc_year_xts, 0,
                            return_xts = TRUE)
```

---

gen_likelihood_stats *Generate likelihood statistics (AIC, AICC, BIC, etc.)*

Description

Generate likelihood statistics that include the Jacobian adjustment for transformations as done in the X-13ARIMA-SEATS program

Usage

```
gen_likelihood_stats(
  Y,
  Lnlkhd,
  Nintvl = 2,
  Np = 2,
  Adj = NULL,
  Logit = FALSE,
  Trans = TRUE,
  Lam = 0,
  Eick = NULL
)
```

Arguments

Y	numeric vector, original time series.
Lnlkhd	numeric scalar, maximized log likelihood of model.
Nintvl	Integer scalar, Number of differences in model used. Default is 2.
Np	Integer scalar, Number of parameters in model used. Default is 2.
Adj	numeric vector, prior adjustment factor time series. Default is NULL, which indicates there is no prior adjustment.
Logit	logical scalar, if TRUE the logit transformation is used. Default is FALSE.
Trans	logical scalar, if TRUE a Box-Cox transform is used. Default is TRUE
Lam	numeric scalar, Box-Cox transformation parameter. Default is 0 (Log transform).
Eick	numeric scalar, weighting paramter for Empiracle Information Criterion. Default is NULL (EIC not computed).

Value

List of likelihood diagnostics and related estimates (ll = log likelihood, jacadj = jacobian transformation adjustment, lladj = adjusted likelihood, nobs = number of observations, neffective = effective number of observations, nparams = number of parameters, df = Nspobs - Nintvl - Np, aic = AIC, aicc = AICC, hannanquinn = Hannan Quinn, bic = BIC, bic2 = BIC2, EIC = Empiracle Information Criterion)

Examples

```
ic_default_log_est <-
   rjd3highfreq::fractionalAirlineEstimation(log(ic_obs), periods=c(365.25/7),
                                             x=ic_default_matrix)
ic_default_log_lkhd_stat <-
   gen_likelihood_stats(ic_obs,
                        Lnlkhd = ic_default_log_est$likelihood$ll,
                        Nintvl = 2,
                        Np     = ic_default_log_est$likelihood$nparams)
ic_default_log_aicc <- ic_default_log_lkhd_stat$aicc
```

---

gen_movereg_holiday *Generate Movreg holiday regression matrix*

Description

Generates a matrix with holiday regressors as defined within the BLS weekly adjustment program Movereg

Usage

```
gen_movereg_holiday(hol_n, hol_index, hol_wt, hol_type, this_week, this_year)
```

Arguments

hol_n	Numeric scalar; number of weights for this particular holiday
hol_index	Numeric scalar; position of holiday within the weight vector
hol_wt	Numeric vector; weight vector for holiday effect
hol_type	Character scalar; Type of holiday ('newyear' for New Years Day, 'mlk' for MLK Observance, 'president' for President's Day, 'easter' for Easter, 'memorial' for Memorial Day, 'july4' for Independence Day, 'labor' for Labor Day, 'columbus' for Columbus Day, 'veteran' for Veterans Day, 'thanks' for Thanksgiving).
this_week	Numeric vector; Week of the year for each observation.
this_year	Numeric vector; Year of each observation.

Value

Generate matrix of holiday regressors as defined within the BLS weekly adjustment program Movereg

Examples

```
ic_easter  <-
  gen_movereg_holiday(hol_n = 8,
                      hol_index = 8,
                      hol_wt = c(1, 0, 0, 0, 0, 0, 0, 0),
                      hol_type = 'easter',
                      this_week = ic_week,
                      this_year = ic_year)
ic_labor  <-
  gen_movereg_holiday(hol_n = 2,
                      hol_index = 2,
                      hol_wt = c(0, 1),
                      hol_type = 'labor',
                      this_week = ic_week,
                      this_year = ic_year)
ic_thanksgiving <-
  gen_movereg_holiday(hol_n = 1,
                      hol_index = 1,
                      hol_wt = array(1, dim=1),
                      hol_type = 'thanksgiving',
                      this_week = ic_week,
                      this_year = ic_year)
ic_holiday_matrix_reduced <- cbind(ic_easter, ic_labor, ic_thanksgiving)
colnames(ic_holiday_matrix_reduced) <- c('easter', 'labor', 'thanksgiving')
```

---

gen_movereg_holiday_xts

Generate Movreg holiday regression matrix, using xts series as input

---

Description

Generates a matrix with holiday regressors as defined within the BLS weekly adjustment program Movereg

Usage

```
gen_movereg_holiday_xts(
  hol_n,
  hol_index,
  hol_wt,
  hol_type,
  this_week,
  this_year,
  return_xts = FALSE
)
```

Arguments

hol_n	Numeric scalar; number of weights for this particular holiday
hol_index	Numeric scalar; position of holiday within the weight vector
hol_wt	Numeric vector; weight vector for holiday effect

hol_type	Character scalar; Type of holiday ('newyear' for New Years Day, 'mlk' for MLK Observance, 'president' for President's Day, 'easter' for Easter, 'memorial' for Memorial Day, 'july4' for Independence Day, 'labor' for Labor Day, 'columbus' for Columbus Day, 'veteran' for Veterans Day, 'thanks' for Thanksgiving).
this_week	Numeric vector; Week of the year for each observation, stored as an xts time series object.
this_year	Numeric vector; Year of each observation, stored as an xts time series object.
return_xts	Logical scalar; return matrix as an xts time series object. Default is FALSE.

Value

Generate matrix of holiday regressors as defined within the BLS weekly adjustment program Movereg

Examples

```
ic_easter_xts  <-
  gen_movereg_holiday_xts(hol_n = 8,
                          hol_index = 8,
                          hol_wt = c(1, 0, 0, 0, 0, 0, 0, 0),
                          hol_type = 'easter',
                          this_week = ic_week_xts,
                          this_year = ic_year_xts)
ic_labor_xts  <-
  gen_movereg_holiday_xts(hol_n = 2,
                          hol_index = 2,
                          hol_wt = c(0, 1),
                          hol_type = 'labor',
                          this_week = ic_week_xts,
                          this_year = ic_year_xts)
ic_thanksgiving_xts <-
  gen_movereg_holiday_xts(hol_n = 1,
                          hol_index = 1,
                          hol_wt = array(1, dim=1),
                          hol_type = 'thanksgiving',
                          this_week = ic_week_xts,
                          this_year = ic_year_xts)
ic_holiday_matrix_reduced_xts <-
   cbind(ic_easter_xts, ic_labor_xts, ic_thanksgiving_xts)
colnames(ic_holiday_matrix_reduced_xts) <- c('easter', 'labor', 'thanksgiving')
```

---

gen_outlier_matrix *Generate point outlier regression matrix*

Description

Generates a regression matrix of AO (point) outliers to be included in weekly modeling routines

Usage

```
gen_outlier_matrix(
  outlier_dates,
  this_week,
  this_year,
  forecast = 0,
  air_name = TRUE,
  return_xts = FALSE
)
```

Arguments

outlier_dates	Integer matrix - matrix of dates for point outliers
this_week	Numeric vector; Week of the year for each observation.
this_year	Numeric vector; Year of each observation.
forecast	Numeric scalar; Number of forecasts. Default: 0.
air_name	Logical scalar; If TRUE, names are formatted as they are in the fractional airline routine; otherwise format them as in movereg. Default: TRUE
return_xts	Logical scalar; return matrix as an xts time series object. Default is FALSE. If TRUE, this_week and this_year should be xts time series objects.

Value

Generate matrix of point outlier regressors, with column names that describe the individual regressors

Examples

```
ic_outlier_date <-
   matrix(c(30, 1992, 30, 1993, 52, 1993,  5, 1994,  3, 1996,
            38, 2001, 39, 2001, 40, 2001, 41, 2001, 42, 2001,
            43, 2001, 47, 2001, 48, 2001, 37, 2005, 38, 2005,
            39, 2005, 40, 2005, 41, 2005,  1, 2006,  2, 2007,
             4, 2008, 45, 2012, 35, 2017, 12, 2020, 13, 2020,
            14, 2020, 15, 2020, 16, 2020), ncol=2, byrow=TRUE)
ic_outlier_matrix_1992 <-
   gen_outlier_matrix(ic_outlier_date, ic_week, ic_year, 0,
                      return_xts = FALSE)
cc_outlier_dates_1992 <-
   matrix(c(30, 1992, 30, 1993, 52, 1993,  5, 1994,  3, 1996,
            38, 2001, 39, 2001, 40, 2001, 41, 2001, 42, 2001,
            43, 2001, 47, 2001, 48, 2001, 37, 2005, 38, 2005,
            39, 2005, 40, 2005, 41, 2005,  1, 2006,  2, 2007,
             4, 2008, 45, 2012, 35, 2017, 12, 2020, 13, 2020), ncol=2, byrow=TRUE)
cc_level_matrix_xts_1992 <-
   gen_level_outlier_matrix(cc_outlier_dates_1992, cc_week_xts, cc_year_xts, 0,
                            return_xts = TRUE)
```

gen_tc_outlier_matrix *Generate temporary change outlier regression matrix*

Description

Generates a regression matrix of TC (temporary change) outliers to be included in weekly modeling routines

Usage

```
gen_tc_outlier_matrix(
  outlier_dates,
  this_week,
  this_year,
  forecast = 0,
  air_name = TRUE,
  this_freq = NULL,
  tc_alpha = NULL,
  return_xts = FALSE
)
```

Arguments

outlier_dates	Integer matrix - matrix of dates for TC outliers
this_week	Numeric vector; Week of the year for each observation.
this_year	Numeric vector; Year of each observation.
forecast	Numeric scalar; Number of forecasts. Default: 0.
air_name	Logical scalar; If TRUE, names are formatted as they are in the fractional airline routine; otherwise format them as in movereg. Default: TRUE
this_freq	Numeric scalar; frequency of time series. Default: 365.25/7, for a weekly series
tc_alpha	Numeric scalar; Rate of decay for the TC outlier. Default: will be computed as in X-13ARIMA-SEATS for a weekly series
return_xts	Logical scalar; return matrix as an xts time series object. Default is FALSE. If TRUE, this_week and this_year should be xts time series objects.

Value

Generate matrix of temporary change outlier regressors, with column names that describe the individual regressors

Examples

```
ic_tc_dates <-
   matrix(c(13, 2020), ncol=2, byrow=TRUE)
ic_tc_matrix <-
   gen_level_outlier_matrix(ic_tc_dates, ic_week, ic_year, 0,
                            return_xts = FALSE)
cc_tc_dates <-
   matrix(c(13, 2020), ncol=2, byrow=TRUE)
cc_tc_matrix_xts <-
```

```
gen_level_outlier_matrix(cc_tc_dates, cc_week_xts, cc_year_xts, 0,
                         return_xts = TRUE)
```

`ic_day` *Day vector for Initial Unemployment Claims, weekly (ic)*

Description

A tis time series object the day for every observation of the weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_day
```

Format

A tis time series object for the day of IC from February 1, 2003 to January 30, 2021

`ic_default_matrix` *Regression matrix with default outlier and holiday regressors for IC*

Description

A matrix object with the default regression matrix used for the default model of IC in the factional airline model fit to the weekly initial unemployment claims data

Usage

```
ic_default_matrix
```

Format

A 940 x 69 matrix with regressors in each column. The first 10 columns are regular holidays:

ny New Years Day Holiday

mlk MLK Holiday

president Presidents Day Holiday

easter Easter Holiday

memorial Memorial Day Holiday

july4 July 4th Holiday

labor Labor Day Holiday

columbus Columbus Day Holiday

veteran Veteran's Day Holiday

thanksgiving Thanksgiving Holiday

The next 3 columns are special holidays:

july4_wed July 4th falls on a Wednesday

xmas_w53 Christams falls in the 53rd week

xmas_fri Christmas falls on a Friday

The remaining columns are AO outliers in different weeks. Every week in the pandemic has an AO outlier. The outlier list is given below: AO(week 37, 2005) AO(week 38, 2005) AO(week 39, 2005) AO(week 40, 2005) AO(week 41, 2005) AO(week 1, 2006) AO(week 2, 2007) AO(week 4, 2008) AO(week 45, 2012) AO(week 35, 2017) AO(week 12, 2020) AO(week 13, 2020) AO(week 14, 2020) AO(week 15, 2020) AO(week 16, 2020) AO(week 17, 2020) AO(week 18, 2020) AO(week 19, 2020) AO(week 20, 2020) AO(week 21, 2020) AO(week 22, 2020) AO(week 23, 2020) AO(week 24, 2020) AO(week 25, 2020) AO(week 26, 2020) AO(week 27, 2020) AO(week 28, 2020) AO(week 29, 2020) AO(week 30, 2020) AO(week 31, 2020) AO(week 32, 2020) AO(week 33, 2020) AO(week 34, 2020) AO(week 35, 2020) AO(week 36, 2020) AO(week 37, 2020) AO(week 38, 2020) AO(week 39, 2020) AO(week 40, 2020) AO(week 41, 2020) AO(week 42, 2020) AO(week 43, 2020) AO(week 44, 2020) AO(week 45, 2020) AO(week 46, 2020) AO(week 47, 2020) AO(week 48, 2020) AO(week 49, 2020) AO(week 50, 2020) AO(week 51, 2020) AO(week 52, 2020) AO(week 1, 2021) AO(week 2, 2021) AO(week 3, 2021) AO(week 4, 2021) AO(week 5, 2021)

`ic_end` *Ending date for Initial Unemployment Claims, weekly (ic)*

Description

A numeric vector of length 2 with the end date of the weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_end
```

Format

A numeric vector of length 2.

first element starting year

second element starting week

Source

https://www.dol.gov/ui/data.pdf

ic_holiday_matrix *Regression matrix with default holiday regressors for IC*

Description

A matrix object with the default holiday regression matrix used for the default model of IC in the factional airline model fit to the weekly initial unemployment claims data

Usage

```
ic_holiday_matrix
```

Format

A 940 x 13 matrix with regressors in each column. The first 10 columns are regular holidays:

ny New Years Day Holiday

mlk MLK Holiday

president Presidents Day Holiday

easter Easter Holiday

memorial Memorial Day Holiday

july4 July 4th Holiday

labor Labor Day Holiday

columbus Columbus Day Holiday

veteran Veteran's Day Holiday

thanksgiving Thanksgiving Holiday

The next 3 columns are special holidays:

july4_wed July 4th falls on a Wednesday

xmas_w53 Christams falls in the 53rd week

xmas_fri Christmas falls on a Friday

ic_holiday_matrix_fcst *Regression matrix with default holiday regressors for IC, with forecasts*

Description

A matrix object with the default holiday regression matrix used for the default model of IC in the factional airline model fit to the weekly initial unemployment claims data, extended with 104 weeks of forecasts

Usage

```
ic_holiday_matrix_fcst
```

Format

A 1044 x 13 matrix with regressors in each column. The first 10 columns are regular holidays:

ny New Years Day Holiday

mlk MLK Holiday

president Presidents Day Holiday

easter Easter Holiday

memorial Memorial Day Holiday

july4 July 4th Holiday

labor Labor Day Holiday

columbus Columbus Day Holiday

veteran Veteran's Day Holiday

thanksgiving Thanksgiving Holiday

The next 3 columns are special holidays:

july4_wed July 4th falls on a Wednesday

xmas_w53 Christams falls in the 53rd week

xmas_fri Christmas falls on a Friday

---

ic_month *Month vector for Initial Unemployment Claims, weekly (ic)*

Description

A tis time series object the month for every observation of the weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_month
```

Format

A tis time series object for the month of IC from February 1, 2003 to January 30, 2021

ic_obs — *Initial Unemployment Claims, weekly (ic)*

Description

A time series object with weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_obs
```

Format

A tis time series object for IC from February 1, 2003 to January 30, 2021

Source

https://www.dol.gov/ui/data.pdf

ic_outlier — *Seasonal Factors for Initial Unemployment Claims, weekly (ic)*

Description

A time series object with seasonal factors generated for weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_outlier
```

Format

A tis time series object from February 1, 2003 to January 30, 2021

ic_outlier_auto — *Outlier Factors for Initial Unemployment Claims, weekly (ic, automatic outlier identification)*

Description

A time series object with outlier factors generated for weekly initial unemployment claims from a model that includes automatically identified outliers

Usage

```
ic_outlier_auto
```

Format

A tis time series object from February 1, 2003 to January 30, 2021

ic_outlier_auto_ljung *Outlier Factors for Initial Unemployment Claims, weekly (ic, automatic outlier identification, ljung)*

Description

A time series object with outlier factors generated for weekly initial unemployment claims from a model that includes automatically identified outliers with a critical value determined by an algorithm developed by Greta Ljung

Usage

```
ic_outlier_auto_ljung
```

Format

A tis time series object from February 1, 2003 to January 30, 2021

ic_outlier_tc *Outlier Factors for Initial Unemployment Claims, weekly (automatic outlier identification, TC outlier specified)*

Description

A time series object with outlier factors generated for weekly initial unemployment claims from a model that includes automatically identified outliers with a temporary change outlier (TC) specified for the 13th week of 2020

Usage

```
ic_outlier_tc
```

Format

A tis time series object from February 1, 2003 to January 30, 2021

ic_sf — *Seasonal Factors for Initial Unemployment Claims, weekly (ic)*

Description

A time series object with seasonal factors generated for weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_sf
```

Format

A tis time series object from February 1, 2003 to January 30, 2021

Source

https://www.dol.gov/ui/data.pdf

ic_sf_auto — *Seasonal Factors for Initial Unemployment Claims, weekly (ic, automatic outlier identification)*

Description

A time series object with seasonal factors generated for weekly initial unemployment claims from a model that includes automatically identified outliers

Usage

```
ic_sf_auto
```

Format

A tis time series object from February 1, 2003 to January 30, 2021

ic_sf_auto_ljung — *Seasonal Factors for Initial Unemployment Claims, weekly (ic, automatic outlier identification, ljung)*

Description

A time series object with seasonal factors generated for weekly initial unemployment claims from a model that includes automatically identified outliers with a critical value determined by an algorithm developed by Greta Ljung

Usage

```
ic_sf_auto_ljung
```

Format

A tis time series object from February 1, 2003 to January 30, 2021

ic_sf_tc — *Seasonal Factors for Initial Unemployment Claims, weekly (automatic outlier identification, TC outlier specified)*

Description

A time series object with seasonal factors generated for weekly initial unemployment claims from a model that includes automatically identified outliers with a temporary change outlier (TC) specified for the 13th week of 2020

Usage

```
ic_sf_tc
```

Format

A tis time series object from February 1, 2003 to January 30, 2021

ic_start *Starting date for Initial Unemployment Claims, weekly (ic)*

Description

A numeric vector of length 2 with the start of the weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_start
```

Format

A numeric vector of length 2.

first element starting year

second element starting week

Source

https://www.dol.gov/ui/data.pdf

ic_week *Week vector for Initial Unemployment Claims, weekly (ic)*

Description

A tis time series object the week number for every observation of the weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_week
```

Format

A tis time series object for IC from February 1, 2003 to January 30, 2021

Source

https://www.dol.gov/ui/data.pdf

ic_week_xts — *Week vector for Initial Unemployment Claims, weekly (ic), as an xts time series object*

Description

A xts time series object the week number for every observation of the weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_week_xts
```

Format

An xts time series object for IC from February 1, 2003 to January 30, 2021

Source

https://www.dol.gov/ui/data.pdf

ic_xts — *Initial Unemployment Claims, weekly (ic) in xts format*

Description

An xts time series object with weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_xts
```

Format

An xts time series object for IC from February 1, 2003 to January 30, 2021

Source

https://www.dol.gov/ui/data.pdf

ic_xts_dates — *Date Index used in the xts object for Initial Unemployment Claims, weekly (ic)*

Description

A vector of dates used as an index for the xts object of weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_xts_dates
```

Format

An vector of weekly dates from February 1, 2003 to January 30, 2021

Source

https://www.dol.gov/ui/data.pdf

ic_year — *Year vector for Initial Unemployment Claims, weekly (ic)*

Description

A tis time series object the year for every observation of the weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_year
```

Format

A tis time series object for the year of IC from February 1, 2003 to January 30, 2021

ic_year_xts — *Year vector for Initial Unemployment Claims, weekly (ic), as an xts time series object*

Description

A xts time series object the year for every observation of the weekly initial unemployment claims released by the Department of Labor

Usage

```
ic_year_xts
```

Format

An xts time series object for IC from February 1, 2003 to January 30, 2021

Source

https://www.dol.gov/ui/data.pdf

jacobian_trans_adj — *Jacobian transformation adjustment*

Description

compute the Jacobian adjustment for transformations as done in the X-13ARIMA-SEATS program

Usage

```
jacobian_trans_adj(
  Y,
  Nspobs,
  Nintvl = 2,
  Adj = NULL,
  Logit = FALSE,
  Trans = TRUE,
  Lam = 0
)
```

Arguments

Y	numeric vector, original time series.
Nspobs	Integer scalar, Length of Y.
Nintvl	Integer scalar, Number of differences in model used. Default is 2.
Adj	numeric vector, prior adjustment factor time series. Default is NULL, which indicates there is no prior adjustment.
Logit	logical scalar, if TRUE the logit transformation is used. Default is FALSE.
Trans	logical scalar, if TRUE a Box-Cox transform is used. Default is TRUE
Lam	numeric scalar, Box-Cox transformation parameter. Default is 0 (Log transform).

Value

Jacobean adjustment to the likelihood of estimated models to allow AIC and other likelihood statistics to be compared for different transformations.

Examples

```
Nspobs <- length(ic_obs)
ic_jacadj_log   <- jacobian_trans_adj(ic_obs, Nspobs, 2, Trans = TRUE, Lam = 0.0)
```

---

match_date *Match a specific date*

Description

Create an indicator variables for a specific date (year, month, day) for a weekly time series as defined in `tis`.

Usage

```
match_date(x, date_string)
```

Arguments

x	Numeric tis object.
date_string	Character string; Year, month and day of the date (Example - January 1, 1990 is '19900101').

Value

An indicator variable where the week contains the date entered = 1, 0 otherwise.

Examples

```
pandemic_start <- match_date(ic_week, '20200317')
```

---

match_month_day *Match a specific month and day*

Description

Create an indicator variables for a specific month and day.

Usage

```
match_month_day(x, date_string)
```

Arguments

x	Numeric tis object.
date_string	Character string; Month and day of the date (Example - January 1 is '0101').

Value

An indicator variable where the week that contains the date entered = 1, 0 otherwise.

Examples

```
july4_wed <- match_month_day(ic_week, '0707')
```

---

match_month_day_xts *Match a specific month and day*

Description

Create an indicator variables for a specific month and day.

Usage

```
match_month_day_xts(x, date_string, return_xts = FALSE)
```

Arguments

x	Numeric xts object.
date_string	Character string; Month and day of the date (Example - January 1 is '0101').
return_xts	Logical scalar; return matrix as an xts time series object. Default is FALSE.

Value

An indicator variable where the week that contains the date entered = 1, 0 otherwise.

Examples

```
july4_wed_xts <- match_month_day_xts(ic_week_xts, '0707')
```

---

match_week *Match a specific week*

Description

Create an indicator variable for a specific week.

Usage

```
match_week(x_week, week_number)
```

Arguments

x_week	Numeric tis object.
week_number	Numeric scalar; Week of the year to match.

Value

An indicator variable where the observation that matches the week entered = 1, 0 otherwise.

Examples

```
xmas_w53 <- match_week(ic_week, 53)
```

`match_week_xts` *Match a specific week*

Description

Create an indicator variable for a specific week.

Usage

```
match_week_xts(x_week, week_number, return_xts = FALSE)
```

Arguments

`x_week`	Numeric xts object with week of each observation.
`week_number`	Numeric scalar; Week of the year to match.
`return_xts`	Logical scalar; return matrix as an `xts` time series object. Default is FALSE.

Value

An indicator variable where the observation that matches the week entered = 1, 0 otherwise.

Examples

```
xmas_w53_xts <- match_week_xts(ic_week_xts, 53)
```

`plot_boxplot_single` *Generate one set of weekly boxplots*

Description

Generate a single set of boxplots of factors by week for a range of weeks

Usage

```
plot_boxplot_single(
  this_factor,
  this_week,
  first_week,
  last_week,
  forecast = 0,
  this_range = NULL
)
```

Arguments

this_factor	Numeric vector of factors from weekly seasonal adjustment.
this_week	Numeric vector; Week of the year to match.
first_week	Numeric scalar; First week to be included in the plot.
last_week	Numeric scalar; Last week to be included in the plot.
forecast	Numeric scalar; Number of forecasts. Default: 0.
this_range	Numeric scalar; Range of the seasonal factors to be plotted. Default: range taken from the observations being plotted.

Value

Generate a plot of seasonal factors by week. User can specify limits for the weeks plotted in order to generate a multi-panel plot.

Examples

```
par(mfrow = c(3, 2), mar = c(3.1, 2.1, 0.5, 0.5), oma = c(0, 0, 3, 0))
sf_range <- range(ic_sf)
plot_boxplot_single(ic_sf, ic_week,  1,  9, this_range = sf_range)
plot_boxplot_single(ic_sf, ic_week, 10, 18, this_range = sf_range)
plot_boxplot_single(ic_sf, ic_week, 19, 27, this_range = sf_range)
plot_boxplot_single(ic_sf, ic_week, 28, 36, this_range = sf_range)
plot_boxplot_single(ic_sf, ic_week, 37, 45, this_range = sf_range)
plot_boxplot_single(ic_sf, ic_week, 46, 53, this_range = sf_range)
```

---

plot_boxplot_weekly *Generate weekly boxplots*

Description

Generate a multi-panel plot of boxplots by week

Usage

```
plot_boxplot_weekly(
  this_factor,
  this_week,
  forecast = 0,
  this_range = NULL,
  this_type = "Irregular",
  this_title
)
```

Arguments

this_factor	Numeric vector of factors from a weekly seasonal adjustment.
this_week	Numeric vector; Week of the year to match.
forecast	Numeric scalar; Number of forecasts. Default: 0.
this_range	Numeric scalar; Range of the seasonal factors to be plotted. Default: range taken from the observations being plotted.
this_type	Character string; type of factor being plotted. Default: 'Irregular'
this_title	Character string; Main title for plot.

Value

Generate a mulit-panel plot of boxplots of factors by week. Need to have `plot_boxplot_single` loaded to run.

Examples

```
ic_sf_range <-
   range(ic_sf, ic_sf_auto)
plot_boxplot_weekly(ic_sf, ic_week,
                    forecast = 0,
                    this_range = ic_sf_range,
                    this_type  = "Seasonal",
                    this_title = 'Initial Claims (default)')
plot_boxplot_weekly(ic_sf_auto, ic_week,
                    forecast = 0,
                    this_range = ic_sf_range,
                    this_type  = "Seasonal",
                    this_title = 'Initial Claims (auto outliers)')
```

---

plot_sf_mean *Plot of factor means*

Description

Generates a plot of the means of factors

Usage

```
plot_sf_mean(
  this_factor = NULL,
  this_period = NULL,
  this_col = "green",
  y_limit = range(this_factor),
  this_freq,
  this_trans = TRUE,
  this_title = "Mean of Seasonal Factors",
  forecast = 0,
  this_type = "Seasonal",
  add_line = FALSE
)
```

Arguments

this_factor	`tis` object of the factors from a weekly seasonal adjustment
this_period	Integer scalar; vector with period number of the observations.
this_col	Character scalar; color used for factor plots. Default is `green`.
y_limit	Numeric vector of length 2; Range of values you wish the plot to be plotted over. Default is range of the seasonal factors.
this_freq	integer scalar; time series frequency.
this_trans	Logical scalar; indicates if the adjustment was done with a log transform. Default is TRUE.

this_title	Character string; main title of plot. Default is 'Mean of Seasonal Factors'.
forecast	Integer scalar; Number of forecasts appended to the factors. Default is 0.
this_type	Character string; type of factors plotted. Default is 'seasonal'.
add_line	Logical scalar; indicates if this line is being added to an existing plot. Default is FALSE.

Value

Generate plot of the means of factors by period, or add a line to an existing plot. If factors not specified, print out error message and return NULL.

Examples

```
ic_sf_range <- range(ic_sf, ic_sf_auto, ic_sf_auto_ljung, ic_sf_tc)
this_color <- c("#4682B4", "#7846B4", "#B47846", "#82B446")
plot_sf_mean(ic_sf,
             this_period = ic_week,
             this_col = this_color[1],
             y_limit = ic_sf_range,
             this_freq = 53,
             this_type = 'Combined',
             this_trans = FALSE,
             this_title = 'Initial Claims (Weekly Combined Factors)')
plot_sf_mean(ic_sf_auto,
             this_period = ic_week,
             this_col = this_color[2],
             this_freq = 53,
             this_trans = FALSE,
             add_line = TRUE)
plot_sf_mean(ic_sf_auto_ljung,
             this_period = ic_week,
             this_col = this_color[3],
             this_freq = 53,
             this_trans = FALSE,
             add_line = TRUE)
plot_sf_mean(ic_sf_tc,
             this_period = ic_week,
             this_col = this_color[4],
             this_freq = 53,
             this_trans = FALSE,
             add_line = TRUE)
legend('topright',
     legend=c('ic (default)', 'ic (auto outliers)', 'ic (auto, ljung cv)', 'ic (auto tc)'),
       col=this_color,
       lty=rep(1,4),
       cex=0.75)
```

---

plot_sf_weekly *Plot factors by week*

Description

Generate a multi-panel plot of seasonal factors by week

Usage

```
plot_sf_weekly(
  this_sf,
  this_week,
  forecast = 0,
  this_trans = TRUE,
  this_range = NULL,
  this_type = "Seasonal",
  this_title
)
```

Arguments

this_sf	Numeric vector of seasonal factors.
this_week	Numeric vector; Week of the year to match.
forecast	Numeric scalar; Number of forecasts. Default: 0.
this_trans	Logical scalar; set to TRUE if the log transformation is used in the decomposition, FALSE otherwise. Default: TRUE
this_range	Numeric scalar; Range of the seasonal factors to be plotted. Default: range taken from the observations being plotted.
this_type	Character string; type of factor being plotted. Default: 'Seasonal'
this_title	Character string; Main title for plot.

Value

Generate a mulit-panel plot of seasonal factors by week. Need to have `plot_sf_weekly_single` loaded to run.

Examples

```
ic_sf_range <- range(ic_sf, ic_sf_auto, ic_sf_auto_ljung, ic_sf_tc)
plot_sf_weekly(ic_sf,
               this_week = ic_week,
               forecast = 0,
               this_range = ic_sf_range,
               this_trans = FALSE,
               this_type = 'Combined',
               this_title = 'Initial Claims (default)')
plot_sf_weekly(ic_sf_auto,
               this_week = ic_week,
               forecast=0,
               this_range = ic_sf_range,
               this_trans = FALSE,
               this_type = 'Combined',
               this_title = 'Initial Claims (auto)')
plot_sf_weekly(ic_sf_auto_ljung,
               this_week = ic_week,
               forecast=0,
               this_range = ic_sf_range,
               this_trans = FALSE,
               this_type = 'Combined',
               this_title = 'Initial Claims (auto, ljung cv)')
plot_sf_weekly(ic_sf_tc,
```

```
                this_week = ic_week,
                forecast=0,
                this_range = ic_sf_range,
                this_trans = FALSE,
                this_type = 'Combined',
                this_title = 'Initial Claims (fractional airline)')
```

`plot_sf_weekly_single` *Plot factors by week for a range of weeks*

Description

Generate a single plot of seasonal factors by week for a range of weeks

Usage

```
plot_sf_weekly_single(
  this_sf,
  this_week,
  first_week,
  last_week,
  forecast = 0,
  this_range = NULL,
  this_trans = TRUE
)
```

Arguments

`this_sf`	Numeric vector of seasonal factors.
`this_week`	Numeric vector; Week of the year to match.
`first_week`	Numeric scalar; First week to be included in the plot.
`last_week`	Numeric scalar; Last week to be included in the plot.
`forecast`	Numeric scalar; Number of forecasts. Default: 0.
`this_range`	Numeric scalar; Range of the seasonal factors to be plotted. Default: range taken from the observations being plotted.
`this_trans`	Logical scalar; set to TRUE if the log transformation is used in the decomposition, FALSE otherwise. Default: TRUE

Value

Generate a plot of seasonal factors by week. User can specify limits for the weeks plotted in order to generate a multi-panel plot.

Examples

```
par(mfrow = c(3, 2), mar = c(3.1, 2.1, 0.5, 0.5), oma = c(0, 0, 3, 0))
sf_range <- range(ic_sf)
plot_sf_weekly_single(ic_sf, this_week = ic_week,
                      first_week = 1,  last_week = 9,
                      forecast = 0, this_range = sf_range,
                      this_trans = FALSE)
```

```
plot_sf_weekly_single(ic_sf, this_week = ic_week,
                      first_week = 10, last_week = 18,
                      forecast = 0, this_range = sf_range,
                      this_trans = FALSE)
plot_sf_weekly_single(ic_sf, this_week = ic_week,
                      first_week = 19, last_week = 27,
                      forecast = 0, this_range = sf_range,
                      this_trans = FALSE)
plot_sf_weekly_single(ic_sf, this_week = ic_week,
                      first_week = 28, last_week = 36,
                      forecast = 0, this_range = sf_range,
                      this_trans = FALSE)
plot_sf_weekly_single(ic_sf, this_week = ic_week,
                      first_week = 37, last_week = 45,
                      forecast = 0, this_range = sf_range,
                      this_trans = FALSE)
plot_sf_weekly_single(ic_sf, this_week = ic_week,
                      first_week = 46, last_week = 53,
                      forecast = 0, this_range = sf_range,
                      this_trans = FALSE)
```

`set_critical_value` *Set outlier critical value*

Description

Set outlier critical value using the Ljung algorithm as given in Ljung, G. M. (1993). On outlier detection in time series. Journal of Royal Statistical Society B 55, 559-567.

Usage

```
set_critical_value(number_observations, cv_alpha = 0.01)
```

Arguments

`number_observations`	number of observations tested for outliers
`cv_alpha`	alpha for critical value

Value

outlier critical value generated by the algorithm given in Ljung (1993). The critical value in X-13 is different as it is adjusted to allow for smaller values to approximate the normal distribution.

Examples

```
this_critical_value <- set_critical_value(12, 0.025)
```

t_values_air *Generate t-statistics*

Description

Generate t-statistics from estimates generated from the fractional airline model

Usage

```
t_values_air(this_beta, this_cov)
```

Arguments

this_beta	Numeric vector; estimated model parameters from the fractional airline model.
this_cov	Numeric matrix; estimated covariance matrx from the fractional airline model.

Value

Numeric vector of t-values

Examples

```
ic_air_est <-
  rjd3highfreq::fractionalAirlineEstimation(ic_obs, periods=c(365.25/7),
                                    x=ic_holiday_matrix,
                                    outliers=c('ao', 'ls'),
                                    criticalValue = 6)
ic_air_t_reg   <- t_values_air(ic_air_est$model$b, ic_air_est$model$bcov)
ic_air_t_arima <- t_values_air(ic_air_est$estimation$parameters,
                               ic_air_est$estimation$covariance)
```

weekly_high_density_plot *High density plots for weekly series*

Description

Generates high density plots for weekly time series around a given mean - current supported time series objects are xts and tis

Usage

```
weekly_high_density_plot(
  x,
  this_start = NULL,
  this_end = NULL,
  this_main = NULL,
  this_range = NULL,
  this_ylab = " ",
  this_col = "grey",
```

```
  this_mu = 0,
  this_cex = 0.75
)
```

Arguments

x	numeric vector, original time series object to be plotted.
this_start	Integer array of length 2, beginning date for plot. Default is NULL (reset to starting date of series).
this_end	Integer array of length 2, ending date for plot. Default is NULL (reset to ending date of series).
this_main	Character string, Main title of plot. Default is "High Density Plot".
this_range	numeric vector of length 2, range of data displayed in plot. Default is NULL, the range of x will be used.
this_ylab	character string, label for Y-axis on plot. Default is " ", or no label.
this_col	character string, color of lines in plot. Default is "grey".
this_mu	numeric scalar, mean value that the high density plot will be plotted around. Default is 0.0
this_cex	numeric scalar, value for plotting parameter cex which controls the amount by which plotting text and symbols should be scaled relative to the default. Default is 0.75.

Value

High density plot for x

Examples

```
this_color <- c("#4682B4", "#7846B4", "#B47846", "#82B446")
par(mfrow = c(2,2), mar= c(2,4,4.25,1.0), oma=c(0,0,3,0))
ic_outlier_range     <-
   range(ic_outlier, ic_outlier_auto, ic_outlier_auto_ljung, ic_outlier_tc)
weekly_high_density_plot(ic_outlier, this_start = c(2018,1),
                         this_main = "default model",
                         this_range = ic_outlier_range,
                         this_ylab = "IC", this_col = this_color[1])
weekly_high_density_plot(ic_outlier_auto, this_start = c(2018,1),
                         this_main = "auto outlier",
                         this_range = ic_outlier_range,
                         this_ylab = "IC", this_col = this_color[2])
weekly_high_density_plot(ic_outlier_auto_ljung, this_start = c(2018,1),
                         this_main = "auto, ljung cv",
                         this_range = ic_outlier_range,
                         this_ylab = "IC", this_col = this_color[3])
weekly_high_density_plot(ic_outlier_tc, this_start = c(2018,1),
                         this_main = "auto with tc",
                         this_range = ic_outlier_range,
                         this_ylab = "IC", this_col = this_color[4])
mtext("Initial Claims, outlier, from 2018", 3, 1.5, outer=TRUE)
```

`within_month_regressors` *Generate within-month effect regression matrix*

Description

Generates a regression matrix of within month effects from Cleveland and Scott to be included in weekly modeling routines

Usage

```
within_month_regressors(
  this_year,
  this_month,
  this_day,
  l = 30,
  return_xts = TRUE
)
```

Arguments

`this_year`	Numeric vector; Year of each observation.
`this_month`	Numeric vector; Month for each observation.
`this_day`	Numeric vector; Day of the month for each observation.
`l`	Numeric scalar; Number of sin-cos terms to generate, with the total number of regressors being `2*l`. Default: 30.
`return_xts`	Logical scalar; return matrix as an `xts` time series object. Default is FALSE.

Value

Generate matrix of within month effects regressors, with column names that describe the individual regressors

Examples

```
ic_wm_matrix_1992 <-
   within_month_regressors(ic_year, ic_month, ic_day, return_xts = FALSE)
cc_wm_matrix_xts <-
   within_month_regressors(cc_year_xts, cc_month_xts, cc_day_xts,
                           return_xts = TRUE)
```

Index