

Package: tramvs (via r-universe)

September 20, 2026

Type Package

Title Optimal Subset Selection for Transformation Models

Version 0.0-9

Description Greedy optimal subset selection for transformation models (Hothorn et al., 2018, <[doi:10.1111/sjos.12291](https://doi.org/10.1111/sjos.12291)>) based on the abess algorithm (Zhu et al., 2020, <[doi:10.1073/pnas.2014241117](https://doi.org/10.1073/pnas.2014241117)>). Applicable to models from packages 'tram' and 'cotram'. Application to shift-scale transformation models are described in Siegfried et al. (2024, <[doi:10.1080/00031305.2023.2203177](https://doi.org/10.1080/00031305.2023.2203177)>).

Depends R (>= 4.2.0), tram (>= 1.4-1)

Imports stats, variables, methods, cotram, future, future.apply, mvtnorm

Suggests abess, tramnet (>= 0.0.991), colorspace, knitr, mlt, TH.data, survival, ordinal, testthat (>= 3.0.0)

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

URL <http://ctm.R-forge.R-project.org>

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs cmake

Repository <https://r-forge.r-universe.dev>

Date/Publication 2026-05-22 09:46:49 UTC

RemoteUrl <https://github.com/r-forge/ctm>

RemoteRef HEAD

RemoteSha 6cd3bd0ce8570d012a6fdd3955b10c8140794b84

RemoteSubdir pkg/tramvs

Contents

abess_mmlt	2
abess_tram	3
AIC.tramvs	5
BoxCoxVS	5
coef.abess_tram	6
coef.mmltvs	7
coef.tramvs	7
ColrVS	8
cor_init	9
cor_init.default	9
cor_init.mmlt	10
cor_init.stram	10
cotramVS	11
CoxphVS	12
LehmannVS	13
LmVS	14
logLik.tramvs	15
mmltVS	15
plot.tramvs	16
PolrVS	17
predict.tramvs	18
print.tramvs	18
residuals.tramvs	19
SIC	19
SIC.tramvs	20
simulate.tramvs	20
summary.tramvs	21
support.tramvs	21
SurvregVS	22
tramvs	23
Index	25

abess_mmlt

Optimal subset selection for multivariate transformation models

Description

Optimal subset selection for multivariate transformation models

Usage

```
abess_mmlt(
  mltargs,
  supp,
  k_max = supp,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  m0 = NULL,
  ...
)
```

Arguments

<code>mltargs</code>	Arguments passed to <code>mmlt</code>
<code>supp</code>	support size of the coefficient vector
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>m0</code>	Transformation model for initialization
<code>...</code>	Currently ignored

Value

List containing the fitted model via `mmlt`, active set A and inactive set I .

abess_tram

Optimal subset selection for transformation models

Description

Optimal subset selection for transformation models

Usage

```
abess_tram(
  formula,
  data,
  modFUN,
  supp,
```

```

    mandatory = NULL,
    k_max = supp,
    thresh = NULL,
    init = TRUE,
    m_max = 10,
    m0 = NULL,
    ...
  )

```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>modFUN</code>	function for fitting a transformation model, e.g., <code>BoxCox()</code> .
<code>supp</code>	support size of the coefficient vector
<code>mandatory</code>	formula of mandatory covariates, which will always be included and estimated in the model. Note that this also changes the initialization of the active set. The active set is then computed with regards to the model residuals of <code>modFUN(mandatory, ...)</code> instead of the unconditional model.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n\$$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those covariates that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>m0</code>	Transformation model for initialization
<code>...</code>	additional arguments supplied to <code>modFUN</code> .

Value

List containing the fitted model via `modFUN`, active set A and inactive set I .

References

Siegfried, S., Kook, L., & Hothorn, T. (2023). Distribution-free location-scale regression. *The American Statistician*, 77(4), 345-356. doi:[10.1080/00031305.2023.2203177](https://doi.org/10.1080/00031305.2023.2203177)

Examples

```

set.seed(24101968)
library(tramvs)

N <- 1e2
P <- 5
nz <- 3

```

```

beta <- rep(c(1, 0), c(nz, P - nz))
X <- matrix(rnorm(N * P), nrow = N, ncol = P)
Y <- 1 + X %*% beta + rnorm(N)

dat <- data.frame(y = Y, x = X)

abess_tram(y ~ ., dat, modFUN = Lm, supp = 3)

```

AIC.tramvs

*AIC "tramvs"***Description**

AIC "tramvs"

Usage

```

## S3 method for class 'tramvs'
AIC(object, ...)

```

Arguments

object	object of class "tramvs"
...	additional arguments to AIC()

Value

Numeric vector containing AIC of best model

BoxCoxVS

*Optimal subset selection in a BoxCox-type transformation model***Description**

Optimal subset selection in a BoxCox-type transformation model

Usage

```

BoxCoxVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,

```

```
parallel = FALSE,  
future_args = list(strategy = "multisession", workers = supp_max),  
...  
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
supp_max	maximum support which to call abess_tram with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to supp.
thresh	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to TRUE and initializes the active set with those co-variates that are most correlated with score residuals of an unconditional modFUN(update(formula, . ~ 1)).
m_max	maximum number of iterating the splicing algorithm.
parallel	toggle for parallel computing via future_lapply
future_args	arguments passed to plan ; defaults to a "multisession" with supp_max workers
...	Additional arguments supplied to BoxCox

Value

See [tramvs](#)

coef.abess_tram	<i>Coef "abess_tram"</i>
-----------------	--------------------------

Description

Coef "abess_tram"

Usage

```
## S3 method for class 'abess_tram'  
coef(object, ...)
```

Arguments

object	object of class "tramvs"
...	additional arguments to coef()

Value

Named numeric vector containing coefficient estimates see [coef.tram](#)

coef.mmltvs	<i>Coef "mmltvs"</i>
-------------	----------------------

Description

Coef "mmltvs"

Usage

```
## S3 method for class 'mmltvs'  
coef(object, best_only = FALSE, ...)
```

Arguments

object	Object of class "tramvs"
best_only	Wether to return the coefficients of the best model only (default: FALSE)
...	additional arguments to coef()

Value

Vector (best_only = TRUE) or matrix (best_only = FALSE) of coefficients

coef.tramvs	<i>Coef "tramvs"</i>
-------------	----------------------

Description

Coef "tramvs"

Usage

```
## S3 method for class 'tramvs'  
coef(object, best_only = FALSE, ...)
```

Arguments

object	Object of class "tramvs"
best_only	Wether to return the coefficients of the best model only (default: FALSE)
...	additional arguments to coef()

Value

Vector (best_only = TRUE) or matrix (best_only = FALSE) of coefficients

ColrVS

*Optimal subset selection in a Colr-type transformation model***Description**

Optimal subset selection in a Colr-type transformation model

Usage

```
ColrVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
supp_max	maximum support which to call <code>abess_tram</code> with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
thresh	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
m_max	maximum number of iterating the splicing algorithm.
parallel	toggle for parallel computing via future_lapply
future_args	arguments passed to plan ; defaults to a "multisession" with <code>supp_max</code> workers
...	Additional arguments supplied to Colr

Value

See [tramvs](#)

cor_init	<i>Compute correlation for initializing the active set</i>
----------	--

Description

Compute correlation for initializing the active set

Usage

```
cor_init(m0, mb)
```

Arguments

m0	modFUN(formula, data)
mb	modFUN(mandatory, data)

Value

Vector of correlations for initializing the active set, depends on type of model (see e.g. [cor_init.default](#))

cor_init.default	<i>Default method for computing correlation</i>
------------------	---

Description

Default method for computing correlation

Usage

```
## Default S3 method:  
cor_init(m0, mb)
```

Arguments

m0	modFUN(formula, data)
mb	modFUN(mandatory, data)

Value

Vector of correlation for initializing the active set

cor_init.mmlt	<i>Method for computing correlations in mmlts</i>
---------------	---

Description

Method for computing correlations in mmlts

Usage

```
## S3 method for class 'mmlt'
cor_init(m0, mb)
```

Arguments

m0	modFUN(formula, data)
mb	modFUN(mandatory, data)

Value

Vector of correlation for initializing the active set

cor_init.stram	<i>Shit-scale tram method for computing correlation</i>
----------------	---

Description

Shit-scale tram method for computing correlation

Usage

```
## S3 method for class 'stram'
cor_init(m0, mb)
```

Arguments

m0	modFUN(formula, data)
mb	modFUN(mandatory, data)

Value

Vector of correlations for initializing the active set, includes both shift and scale residuals

cotramVS

*Optimal subset selection in a cotram model***Description**

Optimal subset selection in a cotram model

Usage

```
cotramVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
supp_max	maximum support which to call <code>abess_tram</code> with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
thresh	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
m_max	maximum number of iterating the splicing algorithm.
parallel	toggle for parallel computing via <code>future_lapply</code>
future_args	arguments passed to <code>plan</code> ; defaults to a "multisession" with <code>supp_max</code> workers
...	Additional arguments supplied to <code>cotram</code>

Value

See `tramvs`

CoxphVS

*Optimal subset selection in a Coxph-type transformation model***Description**

Optimal subset selection in a Coxph-type transformation model

Usage

```
CoxphVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>parallel</code>	toggle for parallel computing via future_lapply
<code>future_args</code>	arguments passed to plan ; defaults to a "multisession" with <code>supp_max</code> workers
<code>...</code>	Additional arguments supplied to Coxph

Value

See [tramvs](#)

LehmannVS

*Optimal subset selection in a Lehmann-type transformation model***Description**

Optimal subset selection in a Lehmann-type transformation model

Usage

```
LehmannVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>parallel</code>	toggle for parallel computing via future_lapply
<code>future_args</code>	arguments passed to plan ; defaults to a "multisession" with <code>supp_max</code> workers
<code>...</code>	Additional arguments supplied to <code>Lehmann</code>

Value

See [tramvs](#)

LmVS

*Optimal subset selection in an Lm-type transformation model***Description**

Optimal subset selection in an Lm-type transformation model

Usage

```
LmVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
supp_max	maximum support which to call <code>abess_tram</code> with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
thresh	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
m_max	maximum number of iterating the splicing algorithm.
parallel	toggle for parallel computing via <code>future_lapply</code>
future_args	arguments passed to <code>plan</code> ; defaults to a "multisession" with <code>supp_max</code> workers
...	Additional arguments supplied to <code>Lm</code>

Value

See `tramvs`

logLik.tramvs	<i>logLik "tramvs"</i>
---------------	------------------------

Description

logLik "tramvs"

Usage

```
## S3 method for class 'tramvs'
logLik(object, ...)
```

Arguments

object	object of class "tramvs"
...	additional arguments to logLik()

Value

Numeric vector containing log-likelihood of best model, see [logLik.tram](#)

mmltVS	<i>Select optimal subset based on high dimensional BIC in mmlts</i>
--------	---

Description

Select optimal subset based on high dimensional BIC in mmlts

Usage

```
mmltVS(
  mltargs,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  verbose = TRUE,
  parallel = FALSE,
  m0 = NULL,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

mltargs	Arguments passed to <code>mmlt</code>
supp_max	maximum support which to call <code>abess_tram</code> with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
thresh	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n\$$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
m_max	maximum number of iterating the splicing algorithm.
verbose	show progress bar (default: <code>TRUE</code>)
parallel	toggle for parallel computing via future_lapply
m0	Transformation model for initialization
future_args	arguments passed to plan ; defaults to a "multisession" with <code>supp_max</code> workers
...	Arguments passed on to abess_mmlt
supp	support size of the coefficient vector

Details

L0-penalized (i.e., best subset selection) multivariate transformation models using the `abess` algorithm.

Value

object of class "mltvs", containing the regularization path (information criterion SIC and coefficients `coefs`), the best fit (`best_fit`) and all other models (`all_fits`)

plot.tramvs	<i>Plot "tramvs" object</i>
-------------	-----------------------------

Description

Plot "tramvs" object

Usage

```
## S3 method for class 'tramvs'
plot(x, which = c("tune", "path"), ...)
```

Arguments

<code>x</code>	object of class "tramvs"
<code>which</code>	plotting either the regularization path ("path") or the information criterion against the support size ("tune", default)
<code>...</code>	additional arguments to <code>plot()</code>

Value

Returns `invisible(NULL)`

PolrVS	<i>Optimal subset selection in a Polr-type transformation model</i>
--------	---

Description

Optimal subset selection in a Polr-type transformation model

Usage

```
PolrVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .

<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>parallel</code>	toggle for parallel computing via future_lapply
<code>future_args</code>	arguments passed to plan ; defaults to a "multisession" with <code>supp_max</code> workers
<code>...</code>	Additional arguments supplied to Polr

Value

See [tramvs](#)

<code>predict.tramvs</code>	<i>Predict "tramvs"</i>
-----------------------------	-------------------------

Description

Predict "tramvs"

Usage

```
## S3 method for class 'tramvs'
predict(object, ...)
```

Arguments

<code>object</code>	object of class "tramvs"
<code>...</code>	additional arguments to <code>predict.tram()</code>

Value

See [predict.tram](#)

<code>print.tramvs</code>	<i>Print "tramvs"</i>
---------------------------	-----------------------

Description

Print "tramvs"

Usage

```
## S3 method for class 'tramvs'
print(x, ...)
```

Arguments

x	object of class "tramvs"
...	ignored

Value

"tramvs" object is returned invisibly

residuals.tramvs	<i>Residuals "tramvs"</i>
------------------	---------------------------

Description

Residuals "tramvs"

Usage

```
## S3 method for class 'tramvs'  
residuals(object, ...)
```

Arguments

object	object of class "tramvs"
...	additional arguments to residuals()

Value

Numeric vector containing residuals of best model, see [residuals.tram](#)

SIC	<i>SIC generic</i>
-----	--------------------

Description

SIC generic

Usage

```
SIC(object, ...)
```

Arguments

object	Model to compute SIC from
...	for methods compatibility only

Value

Numeric vector (best_only = TRUE) or data.frame with SIC values

SIC.tramvs	<i>SIC "tramvs"</i>
------------	---------------------

Description

SIC "tramvs"

Usage

```
## S3 method for class 'tramvs'  
SIC(object, best_only = FALSE, ...)
```

Arguments

object	object of class "tramvs"
best_only	Whether to return the coefficients of the best model only (default: FALSE)
...	for methods compatibility only

Value

Numeric vector (best_only = TRUE) or data.frame with SIC values

simulate.tramvs	<i>Simulate "tramvs"</i>
-----------------	--------------------------

Description

Simulate "tramvs"

Usage

```
## S3 method for class 'tramvs'  
simulate(object, nsim = 1, seed = NULL, ...)
```

Arguments

object	object of class "tramvs"
nsim	number of simulations
seed	random seed for simulation
...	additional arguments to simulate()

Value

See [simulate.mlt](#)

summary.tramvs	Summary "tramvs"
----------------	------------------

Description

Summary "tramvs"

Usage

```
## S3 method for class 'tramvs'
summary(object, ...)
```

Arguments

object	object of class "tramvs"
...	ignored

Value

"tramvs" object is returned invisibly

support.tramvs	Support "tramvs"
----------------	------------------

Description

Support "tramvs"

Usage

```
## S3 method for class 'tramvs'
support(object, ...)
```

Arguments

object	object of class "tramvs"
...	ignored

Value

Character vector containing active set of best fit

SurvregVS

*Optimal subset selection in a Survreg model***Description**

Optimal subset selection in a Survreg model

Usage

```
SurvregVS(
  formula,
  data,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

<code>formula</code>	object of class "formula".
<code>data</code>	data frame containing the variables in the model.
<code>supp_max</code>	maximum support which to call <code>abess_tram</code> with.
<code>k_max</code>	maximum support size to consider during the splicing algorithm. Defaults to <code>supp</code> .
<code>thresh</code>	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
<code>init</code>	initialize active set. Defaults to <code>TRUE</code> and initializes the active set with those co-variables that are most correlated with score residuals of an unconditional <code>modFUN(update(formula, . ~ 1))</code> .
<code>m_max</code>	maximum number of iterating the splicing algorithm.
<code>parallel</code>	toggle for parallel computing via future_lapply
<code>future_args</code>	arguments passed to plan ; defaults to a "multisession" with <code>supp_max</code> workers
<code>...</code>	Additional arguments supplied to Survreg

Value

See [tramvs](#)

tramvs

Select optimal subset based on high dimensional BIC

Description

Select optimal subset based on high dimensional BIC

Usage

```
tramvs(
  formula,
  data,
  modFUN,
  mandatory = NULL,
  supp_max = NULL,
  k_max = NULL,
  thresh = NULL,
  init = TRUE,
  m_max = 10,
  m0 = NULL,
  verbose = TRUE,
  parallel = FALSE,
  future_args = list(strategy = "multisession", workers = supp_max),
  ...
)
```

Arguments

formula	object of class "formula".
data	data frame containing the variables in the model.
modFUN	function for fitting a transformation model, e.g., BoxCox().
mandatory	formula of mandatory covariates, which will always be included and estimated in the model. Note that this also changes the initialization of the active set. The active set is then computed with regards to the model residuals of modFUN(mandatory, ...) instead of the unconditional model.
supp_max	maximum support which to call abess_tram with.
k_max	maximum support size to consider during the splicing algorithm. Defaults to supp.
thresh	threshold when to stop splicing. Defaults to $0.01 * \text{supp} * p * \log(\log(n)) / n$, where p denotes the number of predictors and n the sample size.
init	initialize active set. Defaults to TRUE and initializes the active set with those covariates that are most correlated with score residuals of an unconditional modFUN(update(formula, . ~ 1)).
m_max	maximum number of iterating the splicing algorithm.

<code>m0</code>	Transformation model for initialization
<code>verbose</code>	show progress bar (default: TRUE)
<code>parallel</code>	toggle for parallel computing via future_lapply
<code>future_args</code>	arguments passed to plan ; defaults to a "multisession" with <code>supp_max</code> workers
<code>...</code>	Arguments passed on to abess_tram
<code>supp</code>	support size of the coefficient vector

Details

L0-penalized (i.e., best subset selection) transformation models using the abess algorithm.

Value

object of class "tramvs", containing the regularization path (information criterion SIC and coefficients `coefs`), the best fit (`best_fit`) and all other models (`all_fits`)

Examples

```
set.seed(24101968)
library("tramvs")

N <- 1e2
P <- 5
nz <- 3
beta <- rep(c(1, 0), c(nz, P - nz))
X <- matrix(rnorm(N * P), nrow = N, ncol = P)
Y <- 1 + X %*% beta + rnorm(N)

dat <- data.frame(y = Y, x = X)
res <- tramvs(y ~ ., data = dat, modFUN = Lm)
plot(res, type = "b")
plot(res, which = "path")
```

Index

abess_mmlt, [2](#), [16](#)
abess_tram, [3](#), [24](#)
AIC.tramvs, [5](#)

BoxCox, [6](#)
BoxCoxVS, [5](#)

coef.abess_tram, [6](#)
coef.mmltvs, [7](#)
coef.tram, [6](#)
coef.tramvs, [7](#)
Colr, [8](#)
ColrVS, [8](#)
cor_init, [9](#)
cor_init.default, [9](#), [9](#)
cor_init.mmlt, [10](#)
cor_init.stam, [10](#)
cotram, [11](#)
cotramVS, [11](#)
Coxph, [12](#)
CoxphVS, [12](#)

future_lapply, [6](#), [8](#), [11–14](#), [16](#), [18](#), [22](#), [24](#)

LehmannVS, [13](#)
Lm, [14](#)
LmVS, [14](#)
logLik.tram, [15](#)
logLik.tramvs, [15](#)

mmltVS, [15](#)

plan, [6](#), [8](#), [11–14](#), [16](#), [18](#), [22](#), [24](#)
plot.tramvs, [16](#)
Polr, [18](#)
PolrVS, [17](#)
predict.tram, [18](#)
predict.tramvs, [18](#)
print.tramvs, [18](#)

residuals.tram, [19](#)

residuals.tramvs, [19](#)

SIC, [19](#)
SIC.tramvs, [20](#)
simulate.mlt, [20](#)
simulate.tramvs, [20](#)
summary.tramvs, [21](#)
support.tramvs, [21](#)
Survreg, [22](#)
SurvregVS, [22](#)

tramvs, [6](#), [8](#), [11–14](#), [18](#), [22](#), [23](#)