Package ‘ls’

July 14, 2021

Type Package

Title Maximum Likelihood Latent Stratification

Version 0.1.0

Author Ron Berman, Elea McDonnell Feit, and Zhen Huang

Maintainer Elea McDonnell Feit <eleafeit@gmail.com>

Description The package incorporates the latent stratification model via maximum likelihood to better estimate ATE.

License Apache License, Version 2.0

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

ATEd *Difference in means estimate of the ATE*

Description

Computes the ATE by taking the difference in the means.

Usage

```
ATEd(data)
```

Arguments

`data`	data frame containing cols y (positive outcome with zeros) and z (treatment).

Details

For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.
t.test() from base R can be used as an alternative.

Value

difference in means estimate of the ATE

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
ATEd(sim$data)
```

ATEo *Oracle model estimate of the ATE*

Description

Computes the ATE under the assumption that all strata information are known and true.

Usage

```
ATEo(data)
```

Arguments

`data`	data frame containing cols y (positive outcome with zeros) and z (treatment).

Details

For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.
This serves as a benchmark result under the ideal scenario where all strata are known, and can used to compare with other analysis results.

Value

oracle estimate of ATE

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
ATEo(data)
```

`bounds_ls` *Bounds*

Description

Computes parameter bounds to be used in L-BFGS-B mle optimization.

Usage

```
bounds_ls(data)
```

Arguments

`data`	data frame containing cols y (positive outcome with zeros) and z (treatment).

Details

For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.
Each strata must have at least three observations.
According to RDocumentation, the L-BFGS-B method for optimization is taken from Byrd et. al. (1995), allowing box constraints with upper and lower bound.

Value

The maximum and lowest possible values for segment proportions, mean, and variance.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
bounds_ls(sim$data)
```

gr_ll_ls *Gradient of the log-likelihood*

Description

Computes the first order partial derivative of the log-likelihood for the latent stratification model, with respect to every variable in the vector *par*.

Usage

```
gr_ll_ls(par, data, trans = FALSE)
```

Arguments

`par`	vector c(piA, piB, muA1, muA0, muB1, sigma), c(piA, piB/(1-piA), muA1, muA0, muB1, sigma) if trans=TRUE.
`data`	data frame containing columns y (positive outcome with zeros) and z (treatment).
`trans`	boolean signifying if piB has been transformed.

Details

For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.
Sometimes piB is transformed to relative proportions from absolute proportions. This transformation allows the reparameterization of the piA and piB to allow constraint bounds between 0 and 1 in the optimization procedure.
The output vector is named, each representing the gradient taken with respect to that variable in the parameter.

Value

Gradient of the log-likilihood for the latent stratification model as a named vector.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
gr_ll_ls(sim$par, sim$data)

# if the strata proportions are in relative sizes
gr_ll_ls(sim$par, sim$data, trans=TRUE)
```

hes_ll_ls *Hessian Matrix*

Description

Computes the second order partial derivative with respect to each of the *par* variables, resulting in a Hessian matrix.

Usage

```
hes_ll_ls(par, data, trans = FALSE)
```

Arguments

par	vector c(piA, piB, muA1, muA0, muB1, sigma), c(piA, piB/(1-piA), muA1, muA0, muB1, sigma) if trans=TRUE.
data	data frame containing columns y (positive outcome with zeros) and z (treatment).
trans	boolean signifying if piB has been transformed.

Details

For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.
Sometimes piB is transformed to relative proportions from absolute proportions. This transformation allows the reparameterization of the piA and piB to allow constraint bounds between 0 and 1 in the optimization procedure.
The returned Hessian is the second order derivative with respect to θ where θ is in the order of piA, piB, muA1, muA0, muB1, and sigma.

Value

Hessian matrix for the latent stratification model.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
hes_ll_ls(sim$par, sim$data)

# if the strata proportions are in relative sizes
hes_ll_ls(sim$par, sim$data, trans=TRUE)
```

`ll_ls` *Log-likelihood*

Description

Computes the log-likelihood for each of the four observational groups under the assumption of three strata and common variance.

Usage

```
ll_ls(par, data, trans = FALSE)
```

Arguments

`par` vector c(piA, piB, muA1, muA0, muB1, sigma), c(piA, piB/(1-piA), muA1, muA0, muB1, sigma) if trans=TRUE.

`data` data frame containing columns y (positive outcome with zeros) and z (treatment).

`trans` boolean signifying if piB has been transformed.

Details

For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.
Sometimes piB is transformed to relative proportions from absolute proportions. This transformation allows the reparameterization of the piA and piB to allow constraint bounds between 0 and 1 in the optimization procedure.
The log likelihoods are calculated based on equation 11-14 in the paper. Note that the equations presented in the paper are for normal likelihood and for a single individual, thus corresponding adjustments have been made in the code to calculate the log likelihood for the group.
If you receive the warning *Error in ll_ls(): Numeric overruns*, this means that the log likelihood has exceeded R's value storage capacity, therefore storing it as infinity values. In this case, the value will be negative infinity, as likelihood does not exceed 1.

Value

Log-likelihood for the latent stratification model.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
ll_ls(sim$par, sim$data)

# if the strata proportions are in relative sizes
ll_ls(sim$par, sim$data, trans=TRUE)
```

ls_vcv *Variance-Covariance Matrix*

Description

Variance-Covariance Matrix

Usage

```
ls_vcv(par, data, method)
```

Arguments

par	vector c(piA, piB, muA1, muA0, muB1, sigma).
data	data frame containing cols y (positive outcome with zeros) and z (treatment).
method	method used for computation: score, hessian, robust, and bootstrap.

Details

Computes the variance-covariance matrix:
if method = "hessian" then the standard errors are computed by the numeric hessian
if method = "score" then standard errors are computed from the gradient
if method = "robust" then white robust standard errors are computed
if method = "bootstrap" then the standard errors are computed by bootstrap
For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.

Value

the variance-covariance matrix based on the specificed method.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
ls_vcv(sim$par, sim$data, "hessian")
```

mle_ls *Maximum Likeihood Estimate*

Description

Computes the maximum likelihood estimate for the latent stratification model, optionally takes starting values in original parameter space.

Usage

```
mle_ls(data, start = NULL, starts = 1, vcv = "hessian", quiet = FALSE)
```

Arguments

`data`	data frame containing cols y (positive outcome with zeros) and z (treatment).
`start`	vector starting values for parameters c(piA, piB, muA1, muA0, muB1, sigma).
`starts`	number of starting values.
`vcv`	the variance-covariance matrix of the data, can be calculated using ls_vcv().
`quiet`	boolean controlling if the computation time should be printed after execution.

Details

If starts=1, then the optimization is run once from the starting values. If starts>1, then mle optimization is done with multiple starting values.
The output is an object, which can be called using summary() to show the ATE parameters, their standard errors, and the maximum likelihood. To access the variance-covariance matrix, use $vcv.
For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.
For the \empthvcv parameter:
if vcv = "hessian" then the standard errors are computed by the numeric hessian
if vcv = "score" then standard errors are computed from the gradient
if vcv = "robust" then white robust standard errors are computed
if vcv = "bootstrap" then the standard errors are computed by bootstrap
The function uses L-BFGS-B method for optimization. According to RDocumentation, it is taken from Byrd et. al. (1995), allowing box constraints with upper and lower bound.
If quiet=TRUE, then the time to optimize and calculate the variance covariance matrix will be displayed along with the results.

Value

object containing the MLE results.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
mle_ls(sim$data)

# if you wish to start the optimization in 3 places
mle_ls(sim$data, starts=3)

# if you wish to identify the starting values yourself
startv = c(0.2, 0.1, 5, 4.5, 3, 0.3)
mle_ls(sim$data, start=startv)
```

`sim_latent_strat` *Simulate Data*

Description

Simulates the data from the latent stratification model with four strata.

Usage

```
sim_latent_strat(
  n = 1e+05,
  p = 0.5,
  piA = 0.05,
  piB = 0.1,
  muA1 = 5,
  muA0 = 4,
  muB1 = 5,
  sigma = 1
)
```

Arguments

`n`	total sample size, the default value is 100000.
`p`	treatment proportions, the default value is 0.5.
`piA`	proportion of strata A, the default value is 0.05.
`piB`	proportion of strata B, the default value is 0.10.
`muA1`	mean for strata A that received treatment, Z = 1, the default value is 5.
`muA0`	mean for strata A that did not received treatment, Z = 0, the default value is 4.
`muB1`	mean for strata B that received treatment, Z = 1, the default value is 5.
`sigma`	variance for all strata, the default value is 1.

Details

The four strata are defined as:
A = positive under treatment and control, or always buyer.
B = positive under treatment only, or influenced buyer.
C = never positive, or never buyer.
The model assumes that those who are positive under control, also known as defiers only doesn't exist. The model also assumes that all strata share the same variance.
The outcome y of strata A and B are generated through mixture models of normal distributions. Strata A is generated using two normal distributions, one with mean muA1 and the other muA0. Strata B is generated using a normal distribution with mean muB1 and 0, representing those in strata B won't be positive without treatment. Strata C is 0 at all times.
For the data frame in the output, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.

Value

A list containing a data frame, a numeric value, and two vectors.
The data frame *data* contains the outcome variable y, treatment dummy z, and strata, and the mean-centered effects-coded dummies for strata.
The numeric value *ATE* is the true average treatment effect.
The vector *par* consists of the true parameters of which the data is simulated.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
sim$par
```

```
# a vector piA 0.2 piB 0.1 muA1 5 muA0 4.5 mUB 3 sigma 0.3
sim$data
# a data frame containing outcome variable y, treatment dummy z, and strata. The first 5000 rows have z=1.
sim$ATE
# 4.0
```

start_ls *Starting Values*

Description

Computes the starting values for proportion, mean, and variance for optimization. The proportions *pi* are transformed to relative sizes.

Usage

```
start_ls(data, rand = FALSE)
```

Arguments

`data`	data frame containing cols y (positive outcome with zeros) and z (treatment).
`rand`	boolean controlling whether random adjusted starting mean values are produced.

Details

For the input data frame, column z is the dummy variable for treatment. If z = 1, then the observation has received treatment. If z = 0, then the observation has not received treatment.
By allowing rand=TRUE, the optim() in mle_ls() can be started in multiple places, providing random adj values while remaining within reasonable bounds.

Value

starting values used for maximum-likelihood optimization.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
start_ls(sim$data)

# if wish to have multiple random starting values
start_ls(sim$data, rand=TRUE)
```

varATEldelta *Standard Error of ATE*

Description

Estimates the standard error and the exponential of the standard error of ATE via the delta method.

Usage

```
varATEldelta(par, vcv)
```

Arguments

`par`	vector c(piA, piB, muA1, muA0, muB1, sigma).
`vcv`	variance-covariance matrix of the parameters, can be calculated using ls_vcv().

Details

The delta method estimates the variance by expanding the function of a random variable through Taylor approximation, which can be expanded to vectorized calculations. For a more detailed explanation, see https://www.stata.com/support/faqs/statistics/delta-method/.

Value

standard error of the average treatment effect.

Examples

```
sim = sim_latent_strat(n=10000, piA=0.2, piB=0.1, muA1=5, muA0=4.5, muB1=3, sigma=0.3)
varcovar = ls_vcv(sim$par, sim$data, "hessian")
varATEldelta(sim$par, varcovar)
```

Index