Package 'ccfa'

September 18, 2018

Type Package

Title Continuous Counterfactual Analysis

Version 1.1.0

Description
Contains methods for computing counterfactuals with a continuous treatment variable as in Callaway and Huang (2017) <https://ssrn.com/abstract=3078187>. In particular, the package can be used to calculate the expected value, the variance, the interquantile range, the fraction of observations below or above a particular cutoff, or other user-supplied functions of an outcome of interest conditional on a continuous treatment. The package can also be used for computing these same functionals after adjusting for differences in covariates at different values of the treatment. Further, one can use the package to conduct uniform inference for each parameter of interest across all values of the treatment, uniformly test whether adjusting for covariates makes a difference at any value of the treatment, and test whether a parameter of interest is different from its average value at an value of the treatment.

Depends R (>= 2.1.0)

License GPL-2

Encoding UTF-8

LazyData true

Imports quantreg, stats, pbapply, BMisc, tidyr, ggplot2, TempleMetrics, doParallel, foreach, formula.tools

RoxygenNote 6.1.0

NeedsCompilation no

Author Weige Huang [aut, cre], Brantly Callaway [aut]

Maintainer Weige Huang <weige.huang@temple.edu>

Repository CRAN

Date/Publication 2018-09-18 08:20:08 UTC

R topics documented:

`cfa` *cfa*

Description

compute counterfactuals using distribution regression with a continuous treatment

Usage

```
cfa(formla, xformla = NULL, tvals, yvals, data, method = "dr",
  link = "logit", tau = seq(0.01, 0.99, 0.01), condDistobj = NULL,
  se = TRUE, iters = 100, cl = 1)
```

Arguments

`formla`	a formula y ~ treatment
`xformla`	one sided formula for x variables to include, e.g. ~x1 + x2
`tvals`	the values of the "treatment" to compute parameters of interest for
`yvals`	the values to compute the counterfactual distribution for
`data`	the data.frame where y, t, and x are

method	either "dr" or "qr" for distribution regression or quantile regression
link	if using distribution regression, any link function that works with the binomial family (e.g. logit (the default), probit, cloglog)
tau	if using quantile regression, which values of tau to estimate the conditional quantiles
condDistobj	optional conditional distribution object that has been previously computed
se	whether or not to compute standard errors using the bootstrap
iters	how many bootstrap iterations to use
cl	how many clusters to use for parallel computation of standard errors

Value

CFA object

Examples

```
data(igm)
tvals <- seq(10,12,length.out=8)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95), length.out=50)
## This line doesn't adjust for any covariates
cfa(lcfincome ~ lfincome, tvals=tvals, yvals=yvals, data=igm,
 se=FALSE)

## This line adjusts for differences in education
cfa(lcfincome ~ lfincome, ~HEDUC, tvals=tvals, yvals=yvals, data=igm,
 se=FALSE)
```

CFA.OBJ *CFA.OBJ*

Description

CFA objects

Usage

```
CFA.OBJ(tvals, distcondt, bootiterlist = NULL, tvallist = NULL,
  coef = NULL)
```

Arguments

tvals	the values of the "treatment" to compute parameters of interest for
distcondt	an ecdf object for a particular value of the treatment
bootiterlist	a list of bootstrapped CFA objects that can be used for computing standard errors
tvallist	the values of the treatment used in each bootstrap iteration
coef	the coefficients from a distribution regression

Value

CFA object

`cfa2` *cfa2*

Description

the same as cfa method except it computes two results at the same time which allows one to conduct inference on their difference

Usage

```
cfa2(formla, tvals, yvals, data, xformla1 = NULL, method1 = "dr",
  link1 = "logit", tau1 = seq(0.01, 0.99, 0.01), condDistobj1 = NULL,
  xformla2 = NULL, method2 = "dr", link2 = "logit",
  tau2 = seq(0.01, 0.99, 0.01), condDistobj2 = NULL, se = TRUE,
  iters = 100, cl = 1)
```

Arguments

`formla`	a formula y ~ treatment
`tvals`	the values of the "treatment" to compute parameters of interest for
`yvals`	the values to compute the counterfactual distribution for
`data`	the data.frame where y, t, and x are
`xformla1`	an optional formula for the first set of x variables
`method1`	the first method for estimating the conditional distribution it can be "dr" for distribution regression or "qr" for quantile regression
`link1`	if using distribution regression, set the link variable. It can be any link function accepted by glm, e.g. logit, probit, cloglog
`tau1`	if using quantile regression, the values of tau to use, the default is seq(.01,.99,.01)
`condDistobj1`	if have already calculated a conditional distribution object outside of the model, can set it here
`xformla2`	an optional formula for the second set of x variables
`method2`	the second method for estimating the conditional distribution it can be "dr" for distribution regression or "qr" for quantile regression
`link2`	if using distribution regression, set the link variable. It can be any link function accepted by glm, e.g. logit, probit, cloglog
`tau2`	if using quantile regression, the values of tau to use, the default is seq(.01,.99,.01)
`condDistobj2`	if have already calculated a conditional distribution object outside of the model, can set it here
`se`	whether or not to compute standard errors using the bootstrap
`iters`	how many bootstrap iterations to use
`cl`	how many clusters to use for parallel computation of standard errors

Value

list of two CFA objects

Examples

```
#' data(igm)
tvals <- seq(10,12,length.out=5)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95), length.out=50)

## obtain counterfactual results using quantile regression with
## no covariates and adjusting for education
cfa2(lcfincome ~ lfincome, tvals, yvals, igm, method1="qr", xformla2=~HEDUC,
method2="qr", se=FALSE, tau1=seq(.1,.9,.1), tau2=seq(.1,.9,.1))
```

CFASE *CFASE*

Description

creates object of class CFASE

Usage

```
CFASE(tvals, est, se = NULL, c = NULL)
```

Arguments

tvals	the values of the "treatment" to compute parameters of interest for
est	the estimate of the parameter at each value of t
se	whether or not to compute standard errors using the bootstrap
c	optional critical value for uniform inference

Value

CFASE object

Diff.lige *Diff.lige*

Description

compute the difference between two estimates of the LIGE

Usage

```
Diff.lige(cfaobj1, cfaobj2, se = T, h)
```

Arguments

cfaobj1	the first CFA object
cfaobj2	the second CFA object
se	boolean whether or not to compute standard errors
h	a bandwidth

Value

a CFASE object

Examples

```
## Not run:
data(igm)
tvals <- seq(10,12,length.out=8)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95), length.out=50)

## obtain counterfactual results
out <- cfa2(lcfincome ~ lfincome, tvals, yvals, igm, method1="qr",
xformla2=~HEDUC, method2="qr", iters=10, tau1=seq(.05,.95,.05),
tau2=seq(.05,.95,.05))
Diff.lige(out$cfa1, out$cfa2, h=0.5)

## End(Not run)
```

E *E*

Description

compute expectations from ecdf object

Usage

```
E(edf)
```

Arguments

edf	an ecdf

Value

scalar expected value

getCoef.CFA *getCoef.CFA*

Description

get a particular parameter of interest from a cfa object

Usage

```
getCoef.CFA(cfaobj, yvals, se = T, ...)
```

Arguments

cfaobj	a CFA object
yvals	the y values that the cfa object is computed for
se	whether or not to compute standard errors
...	can pass additional arguments to fun using this argument

Value

CFASE object

getRes.CFA *getRes.CFA*

Description

get a particular parameter of interest from a cfa object

Usage

```
getRes.CFA(cfaobj, fun, se = T, ...)
```

Arguments

cfaobj	a CFA object
fun	a function to apply for every value of the treatment in the cfaobj. The ccfa package provides several built-in functions: E (for expected value as a function of the treatment variable), Var (for the variance as a function of the treatment variable), IQR (the interquantile range as a function of the treatment variable), pov (the fraction of observations with outcomes below some threshold, as a function of the treatment variable), rich (the fraction of observations with outcomes above some threshold, as a function of the treatment variable), but other user-defined functions can be written. The requirement is that they need to take in an ecdf object and output a scalar result.
se	whether or not to compute standard errors
...	can pass additional arguments to fun using this argument

Value

CFASE object

Examples

```
data(igm)
tvals <- seq(10,12,length.out=8)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95),
  length.out=50)

## obtain counterfactual results
cfaresults <- cfa(lcfincome ~ lfincome, tvals=tvals, yvals=yvals, data=igm,
 se=FALSE)

## get the average outcome (lfincome) as a function of the treatment
## variable (lfincome)
getRes.CFA(cfaresults, E, se=FALSE)

## get the variance of the outcomes as a function of the treatment
## variable
getRes.CFA(cfaresults, Var, se=FALSE)
```

```
## get the inter-quantile range of outcomes as a function of the
## treatment variable
getRes.CFA(cfaresults, IQR, se=FALSE, t1=0.9, t2=0.1)
```

---

getResDiff.CFA *getResDiff.CFA*

Description

Get the difference between two CFA objects

Usage

```
getResDiff.CFA(cfaobj1, cfaobj2, fun, se = T, ...)
```

Arguments

cfaobj1	the first CFA object
cfaobj2	the second CFA object
fun	a function to apply for every value of the treatment in the cfaobj
se	whether or not to compute standard errors
...	can pass additional arguments to fun using this argument

Value

CFASE object

Examples

```
## Not run:
data(igm)
tvals <- seq(10,12,length.out=8)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95), length.out=50)

## obtain counterfactual results
out <- cfa2(lcfincome ~ lfincome, tvals, yvals, igm, method1="qr",
xformla2=~HEDUC, method2="qr", iters=10, tau1=seq(.05,.95,.05),
tau2=seq(.05,.95,.05))

## get the difference between the average that adjusts for covariates and
## the one that does not
getResDiff.CFA(out$cfa1, out$cfa2, E)

## End(Not run)
```

ggplot2.CFA *ggplot2.CFA*

Description

function for plotting results from counterfactual analysis using ggplot2

Usage

```
ggplot2.CFA(cfaseobj, setype = "pointwise", ylim = NULL,
  xlabel = NULL, ylabel = NULL, legend = FALSE)
```

Arguments

cfaseobj	a CFASE object to plot
setype	whether to plot pointwise, uniform, or both standard errors
ylim	optional y limits on the plot
xlabel	optional x axis labels
ylabel	optional y axis labels
legend	boolean for whether or not to plot a legend (tends to look better with this option set to FALSE)

Value

ggplot2 object

Examples

```
## Not run:
data(igm)
tvals <- seq(10,12,length.out=8)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95), length.out=50)

## obtain counterfactual results
out <- cfa2(lcfincome ~ lfincome, tvals, yvals, igm, method1="qr",
xformla2=~HEDUC, method2="qr", iters=10, tau1=seq(.05,.95,.05),
tau2=seq(.05,.95,.05))

## get the difference between the average that adjusts for covariates and
## the one that does not
ggplot2.CFA(getResDiff.CFA(out$cfa1, out$cfa2, E), setype="uniform")

## End(Not run)
```

igm *Intergenerational Mobility data from the PSID*

Description

A dataset with 500 observations of matched parent's family income and child's family income data that also contains information on the education level of the family head (this is primary earner in the family)

Usage

```
igm
```

Format

A data frame with 500 rows and 3 columns

lcfincome log of child's family income

lfincome log of parent's family income

HEDUC head of family's education; less than HS, HS, or COLLEGE

Source

subset of PSID data used in Callaway and Huang (2017)

IQR *IQR*

Description

compute interquantile range from ecdf object

Usage

```
IQR(edf, t1, t2)
```

Arguments

edf	an ecdf
t1	upper quantile
t2	lower quantile

Value

scalar interquantile range

lige *lige*

Description

compute the local intergenerational elasticity

Usage

```
lige(cfaobj, h, se = T)
```

Arguments

cfaobj	a CFA object
h	a bandwidth
se	boolean whether or not to compute standard errors

Value

a CFASE object

Examples

```
## Not run:
data(igm)
tvals <- seq(10,12,length.out=8)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95), length.out=50)
## obtain counterfactual results
out <- cfa2(lcfincome ~ lfincome, tvals, yvals, igm, method1="qr",
xformla2=~HEDUC, method2="qr", iters=10, tau1=seq(.05,.95,.05),
tau2=seq(.05,.95,.05))
lige(out$cfa1, h=0.5)

## End(Not run)
```

LLIGE *LLIGE*

Description

Computes local intergenerational elasticities and its Standard Deviations

Usage

```
LLIGE(B, formla, xformla, data, tvals, h, cl)
```

Arguments

B	number of bootstrap iterations
formla	a formula y ~ treatment
xformla	one sided formula for x variables to include, e.g. ~x1 + x2
data	the data.frame where y, t, and x are
tvals	a grid of values of treatment variable
h	bandwidth
cl	the number of clusters to use, default is 1

Value

LLIGE AND its SD

Examples

```
data(igm)
igm$hs=ifelse(igm$HEDUC=="HS",1,0)
igm$col=ifelse(igm$HEDUC=="COL",1,0)
formla=lcfincome~lfincome
xformla=~hs+col
tvals=seq(quantile(igm$lfincome,probs = 0.1),quantile(igm$lfincome,probs = 0.9),length.out = 10)
h=1.2
data=igm
B=7
cl=1
LLIGE(B,formla=formla, xformla=xformla, data=data,tvals=tvals,h=h,cl=cl)
```

---

localIGE *localIGE*

Description

Computes local intergenerational elasticities

Usage

```
localIGE(formla, xformla = NULL, data, tvals, h = NULL, cl = 1)
```

Arguments

formla	a formula y ~ treatment
xformla	one sided formula for x variables to include, e.g. ~x1 + x2
data	the data.frame where y, t, and x are
tvals	a grid of values of treatment variable
h	bandwidth
cl	the number of clusters to use, default is 1

Value

lige

Examples

```
data(igm)
igm$hs=ifelse(igm$HEDUC=="HS",1,0)
igm$col=ifelse(igm$HEDUC=="COL",1,0)
formla=lcfincome~lfincome
xformla=~hs+col
tvals=seq(quantile(igm$lfincome,probs = 0.1),quantile(igm$lfincome,probs = 0.9),length.out = 10)
h=1.2
cl=1
data=igm
localIGE(formla=formla, xformla=xformla, data=data,tvals=tvals,h=h,cl=cl)
```

Plot_ALIGE *Plot_ALIGE*

Description

plot ALIGE with 95% confidence intervals

Usage

```
Plot_ALIGE(tvals, ALIGE, sd_ALIGE, xlab, ylab, ylim = c(0, 1))
```

Arguments

tvals	a grid of values of treatment variable
ALIGE	lige from
sd_ALIGE	SD of ALIGE
xlab	name of x axis
ylab	name of y axis
ylim	ranges of values for y axis

Value

Plot of ALIGE

Examples

```
data(igm)
igm$hs=ifelse(igm$HEDUC=="HS",1,0)
igm$col=ifelse(igm$HEDUC=="COL",1,0)
formla=lcfincome~lfincome
xformla=~hs+col
tvals=seq(quantile(igm$lfincome,probs = 0.1),quantile(igm$lfincome,probs = 0.9),length.out = 10)
h=1.2
data=igm
cl=1
B=7
ALIGE=localIGE(formla=formla, xformla=xformla, data=data,tvals=tvals,h=h,cl=cl)
sd_ALIGE=sdF(B,formla=formla, xformla=xformla, data=data,tvals=tvals,h=h)
Plot_ALIGE(tvals,ALIGE,sd_ALIGE,xlab="t",ylab="ALIGE",ylim=c(0,1))
```

---

pov *pov*

Description

compute fraction below the poverty line from ecdf

Usage

```
pov(edf, povline)
```

Arguments

edf	an ecdf
povline	the poverty line

Value

scalar fraction below poverty line

---

rich *rich*

Description

compute fraction of "rich"

Usage

```
rich(edf, richline)
```

Arguments

edf	an ecdf
richline	the cutoff for being rich

Value

scalar fraction that are "rich"

---

sdF *sdF*

Description

using wild bootstrap to obtain standard deviation

Usage

```
sdF(B, formla, xformla, data, tvals, h, cl = 1)
```

Arguments

B	number of bootstrap iterations
formla	a formula y ~ treatment
xformla	one sided formula for x variables to include, e.g. ~x1 + x2
data	the data.frame where y, t, and x are
tvals	a grid of values of treatment variable
h	bandwidth
cl	the number of clusters to use, default is 1

Value

sd

Examples

```
data(igm)
igm$hs=ifelse(igm$HEDUC=="HS",1,0)
igm$col=ifelse(igm$HEDUC=="COL",1,0)
formla=lcfincome~lfincome
xformla=~hs+col
tvals=seq(quantile(igm$lfincome,probs = 0.1),quantile(igm$lfincome,probs = 0.9),length.out = 10)
h=1.2
data=igm
B=7
sdF(B,formla=formla, xformla=xformla, data=data,tvals=tvals,h=h)
```

summary_LLIGE *summary_LLIGE*

Description

prints a summary of a LLIGE object

Usage

```
summary_LLIGE(object, ...)
```

Arguments

object	an LLIGE object
...	extra arguments

Value

summary of LLIGE

Examples

```
data(igm)
igm$hs=ifelse(igm$HEDUC=="HS",1,0)
igm$col=ifelse(igm$HEDUC=="COL",1,0)
formla=lcfincome~lfincome
xformla=~hs+col
tvals=seq(quantile(igm$lfincome,probs = 0.1),quantile(igm$lfincome,probs = 0.9),length.out = 10)
h=1.2
data=igm
B=7
cl=1
object=LLIGE(B,formla=formla, xformla=xformla, data=data,tvals=tvals,h=h,cl=cl)
summary_LLIGE(object)
```

summary_localIGE *summary_localIGE*

Description

prints a summary of a localIGE object

Usage

```
summary_localIGE(object, ...)
```

Arguments

object	an localIGE object
...	extra arguments

Examples

```
data(igm)
igm$hs=ifelse(igm$HEDUC=="HS",1,0)
igm$col=ifelse(igm$HEDUC=="COL",1,0)
formla=lcfincome~lfincome
xformla=~hs+col
tvals=seq(quantile(igm$lfincome,probs = 0.1),quantile(igm$lfincome,probs = 0.9),length.out = 10)
h=1.2
cl=1
data=igm
object=localIGE(formla=formla, xformla=xformla, data=data,tvals=tvals,h=h,cl=cl)
summary_localIGE (object)
```

---

test.CFA *test.CFA*

Description

test if a counterfactual distribution is equal to its average for all values of the treatment

Usage

```
test.CFA(cfaobj, fun, allt, se = T, ...)
```

Arguments

cfaobj	a CFA object
fun	which function to use
allt	all values of t in the dataset
se	whether or not to compute standard errors
...	additional parameters for the function fun

Value

CFASE object

Examples

```
## Not run:
data(igm)
tvals <- seq(10,12,length.out=8)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95), length.out=50)

## obtain counterfactual results
out <- cfa2(lcfincome ~ lfincome, tvals, yvals, igm, method1="qr",
xformla2=~HEDUC, method2="qr", iters=10, tau1=seq(.05,.95,.05),
tau2=seq(.05,.95,.05))
test.CFA(out$cfa1, Var, igm$lfincome)

## End(Not run)
```

`test.lige` *test.lige*

Description

test if the local intergnerational elasticity is the same across all values of the treatment variable

Usage

```
test.lige(cfaobj, allt, se = T, h)
```

Arguments

`cfaobj`	a CFA object
`allt`	all the values of the treatment variable in the dataset
`se`	boolean whether or not to compute standard errors
`h`	a bandwidth

Value

a CFASE object

Examples

```
## Not run:
data(igm)
tvals <- seq(10,12,length.out=8)
yvals <- seq(quantile(igm$lcfincome, .05), quantile(igm$lcfincome, .95), length.out=50)

## obtain counterfactual results
out <- cfa2(lcfincome ~ lfincome, tvals, yvals, igm, method1="qr",
xformla2=~HEDUC, method2="qr", iters=10, tau1=seq(.05,.95,.05),
tau2=seq(.05,.95,.05))
test.lige(out$cfa1, allt=igm$lfincome, h=0.5)
```

```
## End(Not run)
```

`Var` *Var*

Description

compute variance from ecdf object

Usage

```
Var(edf)
```

Arguments

`edf`	an ecdf

Value

scalar variance

Index