

# Package ‘TempleMetrics’

November 23, 2017

**Title** Estimating Conditional Distributions

**Version** 1.1.0

**Description** Estimates conditional distributions and conditional quantiles. The versions of the methods in this package are primarily for use in multiple step procedures where the first step is to estimate a conditional distribution. In particular, there are functions for implementing distribution regression, quantile regression, and versions of local linear distribution regression; all in a unified framework. Distribution regression provides a way to flexibly model the distribution of some outcome  $Y$  conditional on covariates  $X$  without imposing parametric assumptions on the conditional distribution but providing more structure than fully nonparametric estimation (See Foresi and Peracchi (1995) <doi:10.2307/2291056> and Chernozhukov, Fernandez-Val, and Melly (2013) <doi:10.3982/ECTA10582>).

**Depends** R (>= 2.1.0)

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Imports** stats, utils, BMisc, pbapply

**RoxygenNote** 6.0.1

**NeedsCompilation** no

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

**Maintainer** Brantly Callaway <brantly.callaway@temple.edu>

**Repository** CRAN

**Date/Publication** 2017-11-23 00:44:13 UTC

## R topics documented:

distreg . . . . .	2
DR . . . . .	3
Fycondx . . . . .	3
Fycondx.DR . . . . .	4
Fycondx.lldRlist . . . . .	4

Fycondx.rqs . . . . .	5
igm . . . . .	5
lldistreg . . . . .	6
lIDR . . . . .	7
lldr.inner . . . . .	7
lldrs.inner . . . . .	8
TempleMetrics . . . . .	8
<b>Index</b>	<b>9</b>

---

distreg	<i>distreg</i>
---------	----------------

---

## Description

the main function for running distribution regressions

## Usage

```
distreg(formla, data, yvals, link = "logit")
```

## Arguments

formla	$y \sim x$
data	the dataset
yvals	all the values of y to compute $F(y x)$
link	which link function to use, it can be anything accepted by glm (for example, logit, probit, or cloglog), the default is "logit"

## Value

DR object

## Examples

```
data(igm)
y0 <- median(igm$lcfincome)
distreg(lcfincome ~ lfincome + HEDUC, igm, y0)
```

---

 DR

*DR*


---

**Description**

DR (distribution regression) objects

**Usage**

DR(yvals, glmlist)

**Arguments**

yvals	the values of the y of F(y x)
glmlist	an estimated model for each y value for F(y x)

---

 Fycondx

*Fycondx*


---

**Description**

a generic method for computing conditional distributions

**Usage**

Fycondx(object, yvals, xdf)

**Arguments**

object	either a distribution regression or quantile regression object
yvals	the values to compute the ecdf for
xdf	a dataframe (can contain multiple rows) with x values

**Value**

a list of conditional distributions

---

 Fycondx.DR

*Fycondx.DR*


---

**Description**

take a particular value of y and predict  $F(y|x)$

**Usage**

```
## S3 method for class 'DR'
Fycondx(object, yvals, xdf)
```

**Arguments**

object	a distribution regression object
yvals	the values to compute the ecdf for
xdf	a dataframe (can contain multiple rows) with x values

**Value**

an ecdf for each value of the x's

**Examples**

```
data(igm)
yvals <- seq(quantile(igm$lcfincome,.05,type=1),
  quantile(igm$lcfincome,.95, type=1), length.out=100)
dres <- distreg(lcfincome ~ lfincome + HEDUC, igm, yvals)
xdf <- data.frame(lfincome=10, HEDUC=c("LessHS","HS"))
d <- Fycondx(dres, yvals, xdf)
d
y0 <- yvals[50]
d[[1]](y0)
```

---

 Fycondx.l1DRlist

*Fycondx.l1DRlist*


---

**Description**

take a particular value of y and predict  $F(y|x)$

**Usage**

```
## S3 method for class 'l1DRlist'
Fycondx(object, yvals, xdf)
```

**Arguments**

object	either a distribution regression or quantile regression object
yvals	the values to compute the ecdf for
xdf	a dataframe (can contain multiple rows) with x values

**Value**

a list of ecdfs for each row in xdf

---

Fycondx.rqs

*Fycondx.rqs*


---

**Description**

compute the conditional distribution of y conditional on x using quantile regression

**Usage**

```
## S3 method for class 'rqs'
Fycondx(object, yvals, xdf)
```

**Arguments**

object	a quantile regression object that has been estimated in a first step
yvals	the values to compute the ecdf for
xdf	a dataframe (can contain multiple rows) with x values

**Value**

a list of conditional distributions

---

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)

---

`lldistreg`

*lldistreg*

---

**Description**

the main function for running "local" distribution regressions. This function runs a local regression that is local for a single (scalar) continuous treatment variable. It also allows for other variables but it does not smooth over these variables.

**Usage**

```
lldistreg(formla, xformla = NULL, data, yvals, tvals, link = "logit",
          cl = 1)
```

**Arguments**

<code>formla</code>	<code>y ~ t</code> , <code>t</code> must be a single continuous variable
<code>xformla</code>	<code>~x</code> , <code>x</code> are other (non-smoothed) variables included in the model
<code>data</code>	the dataset
<code>yvals</code>	all the values of <code>y</code> to compute $F(y t,x)$
<code>tvals</code>	the values of the continuous treatment to compute $F(y t,x)$
<code>link</code>	which link function to use, it can be anything accepted by <code>glm</code> (for example, <code>logit</code> , <code>probit</code> , or <code>cloglog</code> ), the default is "logit"
<code>cl</code>	the number of clusters to use, default is 1

**Value**

a list of IIDR objects that are indexed by the values in `yvals` and `tvals`

**Examples**

```
data(igm)
lldistreg(lcfincome ~ lfincome, ~HEDUC, igm, 10, 10)
```

---

 IIDR

---

*IIDR*


---

**Description**

IIDR (local linear distribution regression) object. It contains a value for y, a value for t, and a value for the parameters

**Usage**

```
IIDR(y, t, thet)
```

**Arguments**

y	the value of y for which F(y t,x) was computed
t	the value of t for which F(y t,x) was computed
thet	the local parameters from the local linear distribution regression

---

 lldr.inner

---

*lldr.inner*


---

**Description**

This calculates a single distribution regression for one value of y and one value of xmain

**Usage**

```
lldr.inner(xmain, y, Y, XMain, XOther = NULL, h = NULL, method = "level")
```

**Arguments**

xmain	a particular value for the "main" continuous x variable
y	a particular value of y to compute local linear distribution regression for
Y	a vector containing the data for the outcome
XMain	a vector containing the data for the "main" x variable
XOther	a matrix or data.frame containing the data for the "other" x variables
h	optional bandwidth
method	"level" or "rank" determining whether method should be used conditional on y <sub>tmin1</sub> or the rank of y <sub>tmin1</sub>

**Value**

an IIDR object

**Examples**

```

data(igm)
lcinc <- 10
Y <- igm$lcfincome
XMain <- igm$lfincome
XOther <- data.frame(COL=1*(igm$HEDUC=="COL"))
lldr.inner(lcinc, 10, Y, XMain, XOther)

```

---

lldrs.inner

*lldrs.inner*


---

**Description**

internal function for running local linear distribution regression for a vector of y and x

**Usage**

```

lldrs.inner(y.seq, xmain.seq, Y, XMain, XOther = NULL, h = NULL,
  method = "level", cl = 1)

```

**Arguments**

y.seq	a vector of y values to compute $F(y x)$ for using local linear distribution regression
xmain.seq	a vector of x values to compute $F(y x)$ using local linear distribution regression
Y	a vector containing the data for the outcome
XMain	a vector containing the data for the "main" x variable
XOther	a matrix or data.frame containing the data for the "other" x variables
h	optional bandwidth
method	"level" or "rank" determining whether method should be used conditional on <code>ytmin1</code> or the rank of <code>ytmin1</code>
cl	The number of clusters to use, default is 1

**Value**

a list of lldr objects that are indexed by y and t separately

---

TempleMetrics

*TempleMetrics: Functions from the Temple Econometrics Reading Group*


---

**Description**

TempleMetrics: Functions from the Temple Econometrics Reading Group



# Index

## \*Topic **datasets**

igm, [5](#)

distreg, [2](#)

DR, [3](#)

Fycondx, [3](#)

Fycondx.DR, [4](#)

Fycondx.l1DRlist, [4](#)

Fycondx.rqs, [5](#)

igm, [5](#)

l1distreg, [6](#)

l1DR, [7](#)

l1dr.inner, [7](#)

l1drs.inner, [8](#)

TempleMetrics, [8](#)

TempleMetrics-package (TempleMetrics), [8](#)