

# Package ‘MECfda’

January 20, 2025

**Type** Package

**Title** Scalar-on-Function Regression with Measurement Error Correction

**Version** 0.1.0

**Maintainer** Heyang Ji <jihx1015@outlook.com>

**Description** Solve scalar-on-function linear models, including generalized linear mixed effect model and quantile linear regression model, and bias correction estimation methods due to measurement error. Details about the measurement error bias correction methods, see Luan et al. (2023) <[doi:10.48550/arXiv.2305.12624](https://doi.org/10.48550/arXiv.2305.12624)>, Tekwe et al. (2022) <[doi:10.1093/biostatistics/kxac017](https://doi.org/10.1093/biostatistics/kxac017)>, Zhang et al. (2023) <[doi:10.5705/ss.202021.0246](https://doi.org/10.5705/ss.202021.0246)>, Tekwe et al. (2019) <[doi:10.1002/sim.8179](https://doi.org/10.1002/sim.8179)>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Depends** R (>= 2.10),

**Imports** stats, stringr, lme4, quantreg, splines, dplyr, MASS, Matrix, gss, corpcor, fda, magrittr, methods, nlme, glme, mgcv

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Heyang Ji [aut, cre, ctb, dtc]

(<<https://orcid.org/0009-0001-7494-7227>>),

Ufuk Beyaztas [ctb, rev] (<<https://orcid.org/0000-0002-5208-4950>>),

Nicolas Escobar-Velasquez [com]

(<<https://orcid.org/0009-0006-0800-5692>>),

Yuanyuan Luan [ctb],

Xiwei Chen [ctb],

Mengli Zhang [ctb],

Roger Zoh [ths],

Lan Xue [ths],

Carmen Tekwe [ths] (<<https://orcid.org/0000-0002-1857-2416>>)

**Repository** CRAN

**Date/Publication** 2024-10-21 12:01:07 UTC

## Contents

basis2fun . . . . .	2
bsplineSeries2fun . . . . .	3
bspline_basis-class . . . . .	4
bspline_basis_expansion . . . . .	5
bspline_series-class . . . . .	6
dim,functional_variable-method . . . . .	6
extractCoef . . . . .	7
fc.beta . . . . .	8
fcQR . . . . .	9
fcRegression . . . . .	10
FourierSeries2fun . . . . .	12
fourier_basis_expansion . . . . .	13
Fourier_series-class . . . . .	14
functional_variable-class . . . . .	15
ME.fcLR_IV . . . . .	15
ME.fcQR_CLS . . . . .	17
ME.fcQR_IV.SIMEX . . . . .	18
ME.fcRegression_MEM . . . . .	20
MECfda.data.sim.0.0 . . . . .	22
MECfda.data.sim.0.1 . . . . .	22
MECfda.data.sim.0.2 . . . . .	23
MECfda.data.sim.0.3 . . . . .	23
MECfda.data.sim.1.0 . . . . .	23
MECfda.data.sim.1.1 . . . . .	23
MECfda.data.sim.1.2 . . . . .	23
MECfda.data.sim.1.3 . . . . .	24
plot,bspline_series-method . . . . .	24
plot,Fourier_series-method . . . . .	25
predict.fcQR . . . . .	25
predict.fcRegression . . . . .	26
<b>Index</b>	<b>28</b>

---

basis2fun

*From the summation series of a functional basis to function value*

---

## Description

Generic function to compute function value from summation series of a functional basis.

**Usage**

```
basis2fun(object, x)

## S4 method for signature 'bspline_series,numeric'
basis2fun(object, x)

## S4 method for signature 'Fourier_series,numeric'
basis2fun(object, x)
```

**Arguments**

object	An object that represents a functional basis.
x	point(s) to take value.

**Details**

When applied to [bspline\\_series](#) object, equivalent to [bsplineSeries2fun](#).  
When applied to [Fourier\\_series](#) object, equivalent to [FourierSeries2fun](#).

**Value**

A numeric atomic vector. See [bsplineSeries2fun](#) and [FourierSeries2fun](#).

**Author(s)**

Heyang Ji

---

bsplineSeries2fun	<i>Compute the value of the Fourier summation series at certain points.</i>
-------------------	-----------------------------------------------------------------------------

---

**Description**

Compute the value of the Fourier summation series at certain points.

**Usage**

```
bsplineSeries2fun(object, x)

## S4 method for signature 'bspline_series,numeric'
bsplineSeries2fun(object, x)
```

**Arguments**

object	an object of <a href="#">bspline_series</a> class.
x	Value of \$x\$.

**Value**

A numeric atomic vector

**Author(s)**

Heyang Ji

**Examples**

```
bsb = bspline_basis(
  Boundary.knots = c(0,24),
  intercept      = TRUE,
  df             = NULL,
  degree        = 3
)
bss = bspline_series(
  coef = c(2,1,1.5,0.5),
  bspline_basis = bsb
)
bsplineSeries2fun(bss,(1:239)/10)
```

---

bspline\_basis-class    *b-spline basis*

---

**Description**

A `s4` class that represents a b-spline basis  $\{B_{i,p}(x)\}_{i=-p}^k$  on the interval  $[t_0, t_{k+1}]$ , where  $B_{i,p}(x)$  is defined as

$$B_{i,0}(x) = \begin{cases} I_{(t_i, t_{i+1}]}(x), & i = 0, 1, \dots, k \\ 0, & i < 0 \text{ or } i > k \end{cases}$$

$$B_{i,r}(x) = \frac{x - t_i}{t_{i+r} - t_i} B_{i,r-1}(x) + \frac{t_{i+r+1} - x}{t_{i+r+1} - t_{i+1}} B_{i+1,r-1}(x)$$

For all the discontinuity points of  $B_{i,r}$  ( $r > 0$ ) in the interval  $(t_0, t_k)$ , let the value equals its limit, which means

$$B_{i,r}(x) = \lim_{t \rightarrow x} B_{i,r}(t)$$

**Slots**

`Boundary.knots` boundary of the domain of the splines (start and end), which is  $t_0$  and  $t_{k+1}$ . Default is  $[0, 1]$ . See `Boundary.knots` in `bs`.

`knots` knots of the splines, which is  $(t_1, \dots, t_k)$ , equally spaced sequence is chosen by the function automatically with equal space ( $t_j = t_0 + j \cdot \frac{t_{k+1} - t_0}{k+1}$ ) when not assigned. See `knots` in `bs`.

`intercept` Whether an intercept is included in the basis, default value is `TRUE`, and must be `TRUE`. See `intercept` in `bs`.

`df` degree of freedom of the basis, which is the number of the splines, equal to  $p + k + 1$ . By default  $k = 0$ , and  $df = p + 1$ . See `df` in `bs`.

degree degree of the splines, which is the degree of piecewise polynomials  $p$ , default value is 3.  
See degree in [bs](#).

### Author(s)

Heyang Ji

### Examples

```
bsb = bspline_basis(
  Boundary.knots = c(0,24),
  intercept      = TRUE,
  df             = NULL,
  degree        = 3
)
```

---

bspline\_basis\_expansion

*B-splines basis expansion for functional variable data*

---

### Description

For a function  $f(t)$ ,  $t \in \Omega$ , and a basis function sequence  $\{\rho_k\}_{k \in \kappa}$ , basis expansion is to compute  $\int_{\Omega} f(t)\rho_k(t)dt$ . Here we do basis expansion for all  $f_i(t)$ ,  $t \in \Omega = [t_0, t_0 + T]$  in functional variable data,  $i = 1, \dots, n$ . We compute a matrix  $(b_{ik})_{n \times p}$ , where  $b_{ik} = \int_{\Omega} f_i(t)\rho_k(t)dt$ . The basis used here is the b-splines basis,  $\{B_{i,p}(x)\}_{i=-p}^k$ ,  $x \in [t_0, t_{k+1}]$ , where  $t_{k+1} = t_0 + T$  and  $B_{i,p}(x)$  is defined as

$$B_{i,0}(x) = \begin{cases} I_{(t_i, t_{i+1}]}(x), & i = 0, 1, \dots, k \\ 0, & i < 0 \text{ or } i > k \end{cases}$$

$$B_{i,r}(x) = \frac{x - t_i}{t_{i+r} - t_i} B_{i,r-1}(x) + \frac{t_{i+r+1} - x}{t_{i+r+1} - t_{i+1}} B_{i+1,r-1}(x)$$

### Usage

```
bspline_basis_expansion(object, n_splines, bs_degree)
```

```
## S4 method for signature 'functional_variable,integer'
```

```
bspline_basis_expansion(object, n_splines, bs_degree)
```

### Arguments

object            a [functional\\_variable](#) class object.  
n\_splines        the number of splines, equal to  $k + p + 1$ . See df in [bs](#).  
bs\_degree        the degree of the piecewise polynomial of the b-splines. See degree in [bs](#).

### Value

Returns a numeric matrix,  $(b_{ik})_{n \times p}$ , where  $b_{ik} = \int_{\Omega} f_i(t)\rho_k(t)dt$

**Author(s)**

Heyang Ji

---

 bspline\_series-class *b-splines summation series.*


---

**Description**

A S4 class that represents the summation  $\sum_{i=0}^k b_i B_{i,p}(x)$  by a `bspline_basis` object and coefficients  $b_i$  ( $i = 0, \dots, k$ ).

**Slots**

`coef` coefficients of the b-splines,  $b_i$  ( $i = 0, \dots, k$ ).

`bspline_basis` a `bspline_basis` object, represents the b-splines basis used,  $\{B_{i,p}(x)\}_{i=-p}^k$ .

**Author(s)**

Heyang Ji

**Examples**

```
bsb = bspline_basis(
  Boundary.knots = c(0,24),
  intercept      = TRUE,
  df             = NULL,
  degree        = 3
)
bss = bspline_series(
  coef = c(2,1,1.5,0.5),
  bspline_basis = bsb
)
```

---

 dim,functional\_variable-method

*Extract dimensionality of functional data.*

---

**Description**

Extract the dimensionality of slot X of `functional_variable` object.

**Usage**

```
## S4 method for signature 'functional_variable'
dim(x)
```

**Arguments**

x                    a `functional_variable` object.

**Value**

Retruns a 2-element numeric vector.

**Author(s)**

Heyang Ji

**Examples**

```
fv = functional_variable(X=array(rnorm(12),dim = 4:3),period = 3)
dim(fv)
```

---

extractCoef                    *Method of class Fourier\_series to extract Fourier coefficients*

---

**Description**

Method of class Fourier\_series to extract Fourier coefficients

**Usage**

```
extractCoef(object)

## S4 method for signature 'Fourier_series'
extractCoef(object)
```

**Arguments**

object                    an object of `Fourier_series` class.

**Value**

A list that contains the coefficients.

**Author(s)**

Heyang Ji

**Examples**

```
fsc = Fourier_series(
  double_constant = 0.5,
  cos = c(0,0.3),
  sin = c(1,0.7),
  k_cos = 1:2,
)
extractCoef(fsc)
```

---

fc.beta

---

*Extract the value of coefficient parameter function*


---

**Description**

Generic function to extract the value of coefficient parameter function of the covariates from linear model with functional covariates at some certain points.

**Usage**

```
fc.beta(object, ...)

## S4 method for signature 'fcRegression'
fc.beta(object, FC = 1, t_points = NULL)

## S4 method for signature 'fcQR'
fc.beta(object, FC = 1, t_points = NULL)
```

**Arguments**

object	An object that represents a functional covariates linear model.
...	More arguments.
FC	An integer, represent the ordinal number of the functional covariate. Default is 1, which is take the first functional covariate.
t_points	Sequence of the measurement (time) points.

**Value**

A numeric atomic vector

**Author(s)**

Heyang Ji



## Description

Fit a quantile regression models below

$$Q_{Y_i|X_i,Z_i}(\tau) = \sum_{l=1}^L \int_{\Omega} \beta_l(\tau, t) X_{li}(t) dt + (1, Z_i^T) \gamma$$

where  $Q_{Y_i}(\tau) = F_{Y_i|X_i,Z_i}^{-1}(\tau)$  is the  $\tau$ -th quantile of  $Y_i$  given  $X_i(t)$  and  $Z_i$ ,  $\tau \in (0, 1)$ .

Model allows one or multiple functional covariate(s) as fixed effect(s), and zero, one, or multiple scalar-valued covariate(s).

## Usage

```
fcQR(
  Y,
  FC,
  Z,
  formula.Z,
  tau = 0.5,
  basis.type = c("Fourier", "Bspline"),
  basis.order = 6L,
  bs_degree = 3
)
```

## Arguments

Y	Response variable, can be an atomic vector, a one-column matrix or data frame, recommended form is a one-column data frame with column name
FC	Functional covariate(s), can be a "functional_variable" object or a matrix or a data frame or a list of these object(s)
Z	Scalar covariate(s), can be NULL or not input (when there's no scalar covariate), an atomic vector (when only one scalar covariate), a matrix or data frame, recommended form is a data frame with column name(s)
formula.Z	A formula without the response variable, contains only scalar covariate(s). If not assigned, include all scalar covariates and intercept term.
tau	Quantile $\tau \in (0, 1)$ , default is 0.5. See <a href="#">rq</a> .
basis.type	Type of function basis. Can only be assigned as one type even if there is more than one functional covariates. Available options: 'Fourier' or 'Bspline', represent Fourier basis and b-spline basis respectively. For the detailed form for Fourier and b-splines basis, see <a href="#">fourier_basis_expansion</a> and <a href="#">bspline_basis_expansion</a> .

basis.order	Indicate number of the function basis. When using Fourier basis $\frac{1}{2}, \sin kt, \cos kt, k = 1, \dots, K$ , basis.order is the number $K$ . When using b-splines basis $\{B_{i,p}(x)\}_{i=-p}^k$ , basis.order is the number of splines, equal to $k + p + 1$ . (same as argument df in <a href="#">bs</a> .) May set a individual number for each functional covariate. When the element of this argument is less than the number of functional covariates, it will be used recursively.
bs_degree	Degree of the piecewise polynomials if use b-splines basis, default is 3. See degree in <a href="#">bs</a> .

**Value**

fcQR returns an object of class "fcQR". It is a list that contains the following elements.

regression_result	Result of the regression.
FC.BasisCoefficient	A list of Fourier_series or bspline_series object(s), represents the functional linear coefficient(s) of the functional covariates.
function.basis.type	Type of function basis used.
basis.order	Same as in the arguments.
data	Original data.
bs_degree	Degree of the splines, returned only if b-splines basis is used.

**Author(s)**

Heyang Ji

**Examples**

```
data(MECfda.data.sim.0.0)
res = fcQR(FC = MECfda.data.sim.0.0$FC, Y=MECfda.data.sim.0.0$Y, Z=MECfda.data.sim.0.0$Z,
           basis.order = 5, basis.type = c('Bspline'))
```

---

fcRegression

*Solve linear models with functional covariate(s)*

---

**Description**

Function to fit (generalized) linear model with functional covariate(s). Model allows one or multiple functional covariate(s) as fixed effect(s), and zero, one, or multiple scalar-valued fixed or random effect(s).

**Usage**

```
fcRegression(
  Y,
  FC,
  Z,
  formula.Z,
  family = gaussian(link = "identity"),
  basis.type = c("Fourier", "Bspline"),
  basis.order = 6L,
  bs_degree = 3
)
```

**Arguments**

Y	Response variable, can be an atomic vector, a one-column matrix or data frame, recommended form is a one-column data frame with column name.
FC	Functional covariate(s), can be a "functional_variable" object or a matrix or a data frame or a list of these object(s).
Z	Scalar covariate(s), can be NULL or not input (when there's no scalar covariate), an atomic vector (when only one scalar covariate), a matrix or data frame, recommended form is a data frame with column name(s).
formula.Z	A formula without the response variable, contains only scalar covariate(s) (or intercept), use the format of lme4 package if random effects exist. e.g. ~ Z_1 + (1 Z_2). (See <a href="#">lmer</a> and <a href="#">glmer</a> ) If not assigned, include all scalar covariates and intercept term as fixed effects.
family	A description of the error distribution and link function to be used in the model, see <a href="#">family</a> .
basis.type	Type of function basis. Can only be assigned as one type even if there is more than one functional covariates. Available options: 'Fourier' or 'Bspline', represent Fourier basis and b-spline basis respectively. For the detailed form for Fourier and b-splines basis, see <a href="#">fourier_basis_expansion</a> and <a href="#">bspline_basis_expansion</a> .
basis.order	Indicate number of the function basis. When using Fourier basis $\frac{1}{2}, \sin kt, \cos kt, k = 1, \dots, p_f$ , basis.order is the number $p_f$ . When using b-splines basis $\{B_{i,p}(x)\}_{i=-p}^k$ , basis.order is the number of splines, equal to $k + p + 1$ . (same as argument df in <a href="#">bs</a> .) May set a individual number for each functional covariate. When the element of this argument is less than the number of functional covariates, it will be used recursively.
bs_degree	Degree of the piecewise polynomials if use b-splines basis, default is 3. See degree in <a href="#">bs</a> .

**Details**

Solve linear models with functional covariates below

$$g(E(Y_i|X_i, Z_i)) = \sum_{l=1}^L \int_{\Omega_l} \beta_l(t) X_{li}(t) dt + (1, Z_i^T) \gamma$$

where the scalar-valued covariates can be fixed or random effect or doesn't exist (may do not contain scalar-valued covariates).

### Value

fcRegression returns an object of class "fcRegression". It is a list that contains the following elements.

regression_result	Result of the regression.
FC.BasisCoefficient	A list of Fourier_series or bspline_series object(s), represents the functional linear coefficient(s) of the functional covariates.
function.basis.type	Type of function basis used.
basis.order	Same as in the arguments.
data	Original data.
bs_degree	Degree of the splines, returned only if b-splines basis is used.

### Author(s)

Heyang Ji

### Examples

```
data(MECfda.data.sim.0.0)
res = fcRegression(FC = MECfda.data.sim.0.0$FC, Y=MECFda.data.sim.0.0$Y, Z=MECFda.data.sim.0.0$Z,
  basis.order = 5, basis.type = c('Bspline'),
  formula.Z = ~ Z_1 + (1|Z_2))
```

---

FourierSeries2fun

*Compute the value of the Fourier summation series*

---

### Description

Compute the value of the Fourier summation series

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{p_a} a_k \cos\left(\frac{2\pi}{T}k(x - t_0)\right) + \sum_{k=1}^{p_b} b_k \sin\left(\frac{2\pi}{T}k(x - t_0)\right), \quad x \in [t_0, t_0 + T]$$

at some certain point(s).

### Usage

```
FourierSeries2fun(object, x)
```

```
## S4 method for signature 'Fourier_series,numeric'
FourierSeries2fun(object, x)
```

**Arguments**

object            an object of `Fourier_series` class.  
 x                Value of  $x$ .

**Value**

A numeric atomic vector

**Author(s)**

Heyang Ji

**Examples**

```
fsc = Fourier_series(
  double_constant = 0.5,
  cos = c(0,0.3),
  sin = c(1,0.7),
  k_cos = 1:2,
)
FourierSeries2fun(fsc,1:5)
```

---

fourier\_basis\_expansion

*Fourier basis expansion for functional variable data*

---

**Description**

For a function  $f(x)$ ,  $x \in \Omega$ , and a basis function sequence  $\{\rho_k\}_{k \in \kappa}$ , basis expansion is to compute  $\int_{\Omega} f(t)\rho_k(t)dt$ . Here we do basis expansion for all  $f_i(t)$ ,  $t \in \Omega = [t_0, t_0 + T]$  in functional variable data,  $i = 1, \dots, n$ . We compute a matrix  $(b_{ik})_{n \times p}$ , where  $b_{ik} = \int_{\Omega} f_i(t)\rho_k(t)dt$ . The basis used here is the Fourier basis,

$$\frac{1}{2}, \cos\left(\frac{2\pi}{T}k[x - t_0]\right), \sin\left(\frac{2\pi}{T}k[x - t_0]\right)$$

where  $x \in [t_0, t_0 + T]$  and  $k = 1, \dots, p_f$ .

**Usage**

```
fourier_basis_expansion(object, order_fourier_basis)
```

```
## S4 method for signature 'functional_variable,integer'
fourier_basis_expansion(object, order_fourier_basis)
```

**Arguments**

object            a `functional_variable` class object.  
 order\_fourier\_basis    the order of Fourier basis,  $p_f$ .

**Value**

Returns a numeric matrix,  $(b_{ik})_{n \times p}$ , where  $b_{ik} = \int_{\Omega} f(t) \rho_k(t) dt$

**Author(s)**

Heyang Ji

---

Fourier\_series-class    *s4 class of Fourier summation series*

---

**Description**

A s4 class that represents the linear combination of Fourier basis functions below:

$$\frac{a_0}{2} + \sum_{k=1}^{p_a} a_k \cos\left(\frac{2\pi}{T} k(x - t_0)\right) + \sum_{k=1}^{p_b} b_k \sin\left(\frac{2\pi}{T} k(x - t_0)\right), \quad x \in [t_0, t_0 + T]$$

**Details**

If not assigned,  $t_0 = 0$ ,  $T = 2\pi$ . If not assigned, `k_cos` and `k_sin` equals 1, 2, 3, ...

**Slots**

`double_constant` value of  $a_0$ .  
`cos` values of coefficients of cos waves,  $a_k$ .  
`sin` values of coefficients of sin waves,  $b_k$ .  
`k_cos` values of  $k$  corresponding to the coefficients of cos waves  
`k_sin` values of  $k$  corresponding to the coefficients of sin waves  
`t_0` left end of the domain interval,  $t_0$   
`period` length of the domain interval,  $T$ .

**Author(s)**

Heyang Ji

**Examples**

```
fsc = Fourier_series(
  double_constant = 0.5,
  cos = c(0,0.3),
  sin = c(1,0.7),
  k_cos = 1:2,
)
```

---

functional\_variable-class

*Function-valued variable data.*


---

**Description**

A s4 class that represents data of a function-valued variable. The format is  $f_i(t)$ ,  $t \in \Omega = [t_0, t_0 + T]$  where  $i$  is the observation (subject) index,  $t$  represents the measurement (time) points.

**Slots**

$X$  a matrix  $(x_{ij})_{n \times m}$ , where  $x_{ij} = f_i(t_j)$ , represents the value of  $f_i(t_j)$ , each row represent an observation (subject), each column is corresponding to a measurement (time) point.

$t_0$  start of the domain (time period),  $t_0$ . Default is 0.

period length of the domain (time period),  $T$ . Default is 1.

$t\_points$  sequence of the measurement points,  $(t_1, \dots, t_m)$ . Default is  $t_k = t_0 + \frac{(2k-1)T}{2(m+1)}$ .

**Author(s)**

Heyang Ji

**Examples**

```
X = array(rnorm(12),dim = 4:3)
functional_variable(X=X,period = 3)
```

---

ME.fCLR\_IV

*Bias correction method of applying linear regression to one functional covariate with measurement error using instrumental variable.*


---

**Description**

See detailed model in reference

**Usage**

```
ME.fcLR_IV(
  data.Y,
  data.W,
  data.M,
  t_interval = c(0, 1),
  t_points = NULL,
  CI.bootstrap = FALSE
)
```

**Arguments**

<code>data.Y</code>	Response variable, can be an atomic vector, a one-column matrix or data frame, recommended form is a one-column data frame with column name.
<code>data.W</code>	A dataframe or matrix, represents $W$ , the measurement of $X$ . Each row represents a subject. Each column represent a measurement (time) point.
<code>data.M</code>	A dataframe or matrix, represents $M$ , the instrumental variable. Each row represents a subject. Each column represent a measurement (time) point.
<code>t_interval</code>	A 2-element vector, represents an interval, means the domain of the functional covariate. Default is <code>c(0,1)</code> , represent interval $[0, 1]$ .
<code>t_points</code>	Sequence of the measurement (time) points, default is <code>NULL</code> .
<code>CI.bootstrap</code>	Whether to return the confidence using bootstrap method. Default is <code>FALSE</code> .

**Value**

Returns a `ME.fcLR_IV` class object. It is a list that contains the following elements.

<code>beta_tW</code>	Parameter estimates.
<code>CI</code>	Confidence interval, returned only when <code>CI.bootstrap</code> is <code>TRUE</code> .

**References**

Tekwe, Carmen D., et al. "Instrumental variable approach to estimating the scalar-on-function regression model with measurement error with application to energy expenditure assessment in childhood obesity." *Statistics in medicine* 38.20 (2019): 3764-3781.

**Examples**

```
data(MECfda.data.sim.0.3)
res = ME.fcLR_IV(data.Y = MECfda.data.sim.0.3$Y,
  data.W = MECfda.data.sim.0.3$W,
  data.M = MECfda.data.sim.0.3$M)
```



ME.fcQR\_CLS

*Bias correction method of applying quantile linear regression to dataset with one functional covariate with measurement error using corrected loss score method.*

## Description

Zhang et al. proposed a new corrected loss function for a partially functional linear quantile model with functional measurement error in this manuscript. They established a corrected quantile objective function of the observed measurement that is an unbiased estimator of the quantile objective function that would be obtained if the true measurements were available. The estimators of the regression parameters are obtained by optimizing the resulting corrected loss function. The resulting estimator of the regression parameters is shown to be consistent.

## Usage

```
ME.fcQR_CLS(
  data.Y,
  data.W,
  data.Z,
  tau = 0.5,
  t_interval = c(0, 1),
  t_points = NULL,
  grid_k,
  grid_h,
  degree = 45,
  observed_X = NULL
)
```

## Arguments

data.Y	Response variable, can be an atomic vector, a one-column matrix or data frame, recommended form is a one-column data frame with column name.
data.W	A 3-dimensional array, represents $W$ , the measurement of $X$ . Each row represents a subject. Each column represent a measurement (time) point. Each layer represents an observation.
data.Z	Scalar covariate(s), can be not input or NULL (when there's no scalar covariate), an atomic vector (when only one scalar covariate), a matrix or data frame, recommended form is a data frame with column name(s).
tau	Quantile $\tau \in (0, 1)$ , default is 0.5.
t_interval	A 2-element vector, represents an interval, means the domain of the functional covariate. Default is c(0,1), represent interval $[0, 1]$ .
t_points	Sequence of the measurement (time) points, default is NULL.
grid_k	An atomic vector, of which each element is candidate number of basis.

grid_h	A non-zero-value atomic vector, of which each element is candidate value of tuning parameter.
degree	Used in computation for derivative and integral, default is 45, large enough for most scenario.
observed_X	For estimating parametric variance. Default is NULL.

### Value

Returns a ME.fcQR\_CLS class object. It is a list that contains the following elements.

estimated_beta_hat	Estimated coefficients from corrected loss function (including functional part)
estimated_beta_t	Estimated functional curve
SE_est	Estimated parametric variance. Returned only if observed_X is not NULL.
estimated_Xbasis	The basis matrix we used
res_naive	results of naive method

### References

Zhang, Mengli, et al. "PARTIALLY FUNCTIONAL LINEAR QUANTILE REGRESSION WITH MEASUREMENT ERRORS." *Statistica Sinica* 33 (2023): 2257-2280.

### Examples

```
data(MECfda.data.sim.0.1)

res = ME.fcQR_CLS(data.Y = MECfda.data.sim.0.1$Y,
  data.W = MECfda.data.sim.0.1$W,
  data.Z = MECfda.data.sim.0.1$Z,
  tau = 0.5,
  grid_k = 4:7,
  grid_h = 1:2)
```

---

ME.fcQR_IV.SIMEX	<i>Bias correction method of applying quantile linear regression to dataset with one functional covariate with measurement error using instrumental variable.</i>
------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

### Description

Perform a two-stage strategy to correct the measurement error of a function-valued covariate and then fit a linear quantile regression model. In the first stage, an instrumental variable is used to estimate the covariance matrix associated with the measurement error. In the second stage, simulation extrapolation (SIMEX) is used to correct for measurement error in the function-valued covariate. See detailed model in the reference.

**Usage**

```
ME.fcQR_IV.SIMEX(
  data.Y,
  data.W,
  data.Z,
  data.M,
  tau = 0.5,
  t_interval = c(0, 1),
  t_points = NULL,
  formula.Z,
  basis.type = c("Fourier", "Bspline"),
  basis.order = NULL,
  bs_degree = 3
)
```

**Arguments**

data.Y	Response variable, can be an atomic vector, a one-column matrix or data frame, recommended form is a one-column data frame with column name.
data.W	A dataframe or matrix, represents $W$ , the measurement of $X$ . Each row represents a subject. Each column represent a measurement (time) point.
data.Z	Scalar covariate(s), can be not input or NULL (when there's no scalar covariate), an atomic vector (when only one scalar covariate), a matrix or data frame, recommended form is a data frame with column name(s).
data.M	A dataframe or matrix, represents $M$ , the instrumental variable. Each row represents a subject. Each column represent a measurement (time) point.
tau	Quantile $\tau \in (0, 1)$ , default is 0.5.
t_interval	A 2-element vector, represents an interval, means the domain of the functional covariate. Default is c(0,1), represent interval $[0, 1]$ .
t_points	Sequence of the measurement (time) points, default is NULL.
formula.Z	A formula without the response variable, contains only scalar covariate(s), no random effects. If not assigned, include all scalar covariates and intercept term.
basis.type	Type of funtion basis. Can only be assigned as one type even if there is more than one functional covariates. Available options: 'Fourier' or 'Bspline', represent Fourier basis and b-spline basis respectively. For the detailed form for Fourier and b-splines basis, see <a href="#">fourier_basis_expansion</a> and <a href="#">bspline_basis_expansion</a> .
basis.order	Indicate number of the function basis. When using Fourier basis $\frac{1}{2}, \sin kt, \cos kt, k = 1, \dots, K$ , basis.order is the number $K$ . When using b-splines basis $\{B_{i,p}(x)\}_{i=-p}^k$ , basis.order is the number of splines, equal to $k + p + 1$ . (same as arguement df in <a href="#">bs</a> .) May set a individual number for each functional covariate. When the element of this argument is less than the number of functional covariates, it will be used recursively.
bs_degree	Degree of the piecewise polynomials if use b-splines basis, default is 3. See degree in <a href="#">bs</a> .

**Value**

Returns a `ME.fcQR_IV.SIMEX` class object. It is a list that contains the following elements.

<code>coef.X</code>	A <code>Fourier_series</code> or <code>bspline_series</code> object, represents the functional coefficient parameter of the functional covariate.
<code>coef.Z</code>	The estimate of the linear coefficients of the scalar covariates.
<code>coef.all</code>	Original estimate of linear coefficients.
<code>function.basis.type</code>	Type of function basis used.
<code>basis.order</code>	Same as in the input arguments.
<code>t_interval</code>	A 2-element vector, represents an interval, means the domain of the functional covariate.
<code>t_points</code>	Sequence of the measurement (time) points.
<code>formula</code>	Regression model.
<code>formula.Z</code>	formula object contains only the scalar covariate(s).
<code>zlevels</code>	levels of the non-continuous scalar covariate(s).

**References**

Tekwe, Carmen D., et al. "Estimation of sparse functional quantile regression with measurement error: a SIMEX approach." *Biostatistics* 23.4 (2022): 1218-1241.

**Examples**

```
data(MECfda.data.sim.0.2)

res = ME.fcQR_IV.SIMEX(data.Y = MECfda.data.sim.0.2$Y,
                       data.W = MECfda.data.sim.0.2$W,
                       data.Z = MECfda.data.sim.0.2$Z,
                       data.M = MECfda.data.sim.0.2$M,
                       tau = 0.5,
                       basis.type = 'Bspline')
```

---

`ME.fcRegression_MEM`     *Use `UP_MEM` or `MP_MEM` substitution to apply (generalized) linear regression with one functional covariate with measurement error.*

---

**Description**

The Mixed-effect model (MEM) approach is a two-stage-based method that employs functional mixed-effects models. It allows us to delve into the nonlinear measurement error model, where the relationship between the true and observed measurements is not constrained to be linear, and the distribution assumption on the observed measurement is relaxed to encompass the exponential family rather than being limited to the Gaussian distribution. The MEM approach employs point-wise (`UP_MEM`) and multi-point-wise (`MP_MEM`) estimation procedures to avoid potential computational complexities caused by analyses of multi-level functional data and computations of potentially intractable and complex integrals.

**Usage**

```
ME.fcRegression_MEM(
  data.Y,
  data.W,
  data.Z,
  method = c("UP_MEM", "MP_MEM", "average"),
  t_interval = c(0, 1),
  t_points = NULL,
  d = 3,
  family.W = c("gaussian", "poisson"),
  family.Y = "gaussian",
  formula.Z,
  basis.type = c("Fourier", "Bspline"),
  basis.order = NULL,
  bs_degree = 3,
  smooth = FALSE,
  silent = TRUE
)
```

**Arguments**

data.Y	Response variable, can be an atomic vector, a one-column matrix or data frame, recommended form is a one-column data frame with column name.
data.W	A 3-dimensional array, represents $W$ , the measurement of $X$ . Each row represents a subject. Each column represent a measurement (time) point. Each layer represents an observation.
data.Z	Scalar covariate(s), can be not input or NULL (when there's no scalar covariate), an atomic vector (when only one scalar covariate), a matrix or data frame, recommended form is a data frame with column name(s).
method	The method to construct the substitution $X$ . Available options: 'UP_MEM', 'MP_MEM', 'average'.
t_interval	A 2-element vector, represents an interval, means the domain of the functional covariate. Default is c(0,1), represent interval $[0, 1]$ .
t_points	Sequence of the measurement (time) points, default is NULL.
d	The number of time points involved for MP_MEM (default and miniumn is 3).
family.W	Distribution of $W$ given $X$ , Available options: "gaussian", "poisson".
family.Y	A description of the error distribution and link function to be used in the model, see <a href="#">family</a> .
formula.Z	A formula without the response variable, contains only scalar covariate(s), use the format of lme4 package if random effects exist. e.g. $\sim Z_1 + (1 Z_2)$ . If not assigned, include all scalar covariates and intercept term as fixed effects.
basis.type	Type of function basis. Can only be assigned as one type even if there is more than one functional covariates. Available options: 'Fourier' or 'Bspline', represent Fourier basis and b-spline basis respectively. For the detailed form for Fourier and b-splines basis, see <a href="#">fourier_basis_expansion</a> and <a href="#">bspline_basis_expansion</a> .

basis.order	Indicate number of the function basis. When using Fourier basis $\frac{1}{2}, \sin kt, \cos kt, k = 1, \dots, K$ , basis.order is the number $K$ . When using b-splines basis $\{B_{i,p}(x)\}_{i=-p}^k$ , basis.order is the number of splines, equal to $k + p + 1$ . (same as argument df in <a href="#">bs</a> .) May set a individual number for each functional covariate. When the element of this argument is less than the number of functional covariates, it will be used recursively.
bs_degree	Degree of the piecewise polynomials if use b-splines basis, default is 3. See degree in <a href="#">bs</a> .
smooth	Whether to smooth the substitution of $X$ . Default is FALSE.
silent	Whether not to show the state of the running of the function. Default is TRUE.

**Value**

Returns a `fcRegression` object. See [fcRegression](#).

**References**

Luan, Yuanyuan, et al. "Scalable regression calibration approaches to correcting measurement error in multi-level generalized functional linear regression models with heteroscedastic measurement errors." arXiv preprint arXiv:2305.12624 (2023).

**Examples**

```
data(MECfda.data.sim.0.1)
res = ME.fcRegression_MEM(data.Y = MECfda.data.sim.0.1$Y,
                           data.W = MECfda.data.sim.0.1$W,
                           data.Z = MECfda.data.sim.0.1$Z,
                           method = 'UP_MEM',
                           family.W = "gaussian",
                           basis.type = 'Bspline')
```

---

MECfda.data.sim.0.0    *Simulated data*

---

**Description**

Simulated data

---

MECfda.data.sim.0.1    *Simulated data*

---

**Description**

Simulated data

---

MECfda.data.sim.0.2    *Simulated data*

---

**Description**

Simulated data

---

MECfda.data.sim.0.3    *Simulated data*

---

**Description**

Simulated data

---

MECfda.data.sim.1.0    *Simulated data*

---

**Description**

Simulated data

---

MECfda.data.sim.1.1    *Simulated data*

---

**Description**

Simulated data

---

MECfda.data.sim.1.2    *Simulated data*

---

**Description**

Simulated data

---

MECFda.data.sim.1.3    *Simulated data*

---

**Description**

Simulated data

---

plot,bspline\_series-method

*Plot b-splines baisi summation series.*

---

**Description**

Plot b-splines baisi summation series.

**Usage**

```
## S4 method for signature 'bspline_series'  
plot(x)
```

**Arguments**

x                    A [bspline\\_series](#) object.

**Value**

No return value. Generate a scatter plot.

**Author(s)**

Heyang Ji

**Examples**

```
bsb = bspline_basis(  
  Boundary.knots = c(0,24),  
  intercept      = TRUE,  
  df             = NULL,  
  degree        = 3  
)  
bss = bspline_series(  
  coef = c(2,1,1.5,3),  
  bspline_basis = bsb  
)  
plot(bss)
```



---

plot,Fourier\_series-method  
*Plot Fourier basis summation series.*

---

**Description**

Plot Fourier basis summation series.

**Usage**

```
## S4 method for signature 'Fourier_series'  
plot(x)
```

**Arguments**

x                    A [Fourier\\_series](#) object.

**Value**

No return value. Generate a scatter plot.

**Author(s)**

Heyang Ji

**Examples**

```
fsc = Fourier_series(  
  double_constant = 0.5,  
  cos = c(0,0.3),  
  sin = c(1,0.7),  
  k_cos = 1:2,  
  )  
plot(fsc)
```

---

predict.fcQR                    *Predicted values based on fcQR object*

---

**Description**

Predicted values based on the Quantile linear model with functional covariates represented by a "fcQR" class object.

**Usage**

```
## S3 method for class 'fcQR'  
predict(object, newData.FC, newData.Z = NULL, ...)
```

**Arguments**

object	A fcQR class object produced by <a href="#">fcQR</a> .
newData.FC	A atomic vector or a matrix or a dataframe or a functional_variable class object or a list of objects above. See argument FC in <a href="#">fcRegression</a> .
newData.Z	A dataframe or a matrix or a atomic vector. See argument Z in <a href="#">fcRegression</a> .
...	Further arguments passed to or from other methods <a href="#">predict.rq</a> .

**Details**

If no new data is input, will return the fitted value.

**Value**

See [predict.rq](#).

**Author(s)**

Heyang Ji

---

predict.fcRegression    *Predicted values based on fcRegression object*

---

**Description**

Predicted values based on the linear model with functional covariates represented by a "fcRegression" class object.

**Usage**

```
## S3 method for class 'fcRegression'
predict(object, newData.FC, newData.Z = NULL, ...)
```

**Arguments**

object	A fcRegression class object produced by <a href="#">fcRegression</a> .
newData.FC	A atomic vector or a matrix or a dataframe or a functional_variable class object or a list of objects above. See argument FC in <a href="#">fcRegression</a> .
newData.Z	A dataframe or a matrix or a atomic vector. See argument Z in <a href="#">fcRegression</a> .
...	Further arguments passed to or from other methods, including <a href="#">predict.lm</a> , <a href="#">predict.glm</a> , <a href="#">predict.merMod</a> .

**Details**

If no new data is input, will return the fitted value.

**Value**

See [predict.lm](#), [predict.glm](#), [predict.merMod](#).

**Author(s)**

Heyang Ji

**Examples**

```
data(MECfda.data.sim.0.0)
res = fcRegression(FC = MECfda.data.sim.0.0$FC, Y=MECfda.data.sim.0.0$Y, Z=MECfda.data.sim.0.0$Z,
                  basis.order = 5, basis.type = c('Bspline'),
                  formula.Z = ~ Z_1 + (1|Z_2))
data(MECfda.data.sim.1.0)
predict(object = res, newData.FC = MECfda.data.sim.1.0$FC, newData.Z = MECfda.data.sim.1.0$Z)
```

# Index

## \* data

- MECfda.data.sim.0.0, [22](#)
  - MECfda.data.sim.0.1, [22](#)
  - MECfda.data.sim.0.2, [23](#)
  - MECfda.data.sim.0.3, [23](#)
  - MECfda.data.sim.1.0, [23](#)
  - MECfda.data.sim.1.1, [23](#)
  - MECfda.data.sim.1.2, [23](#)
  - MECfda.data.sim.1.3, [24](#)
- basis2fun, [2](#)
- basis2fun,bspline\_series,numeric-method  
(basis2fun), [2](#)
- basis2fun,Fourier\_series,numeric-method  
(basis2fun), [2](#)
- bs, [4](#), [5](#), [10](#), [11](#), [19](#), [22](#)
- bspline\_basis, [6](#)
- bspline\_basis (bspline\_basis-class), [4](#)
- bspline\_basis-class, [4](#)
- bspline\_basis\_expansion, [5](#), [9](#), [11](#), [19](#), [21](#)
- bspline\_basis\_expansion,functional\_variable,integer-method  
(bspline\_basis\_expansion), [5](#)
- bspline\_series, [3](#), [24](#)
- bspline\_series (bspline\_series-class), [6](#)
- bspline\_series-class, [6](#)
- bsplineSeries2fun, [3](#), [3](#)
- bsplineSeries2fun,bspline\_series,numeric-method  
(bsplineSeries2fun), [3](#)
- dim,functional\_variable-method, [6](#)
- extractCoef, [7](#)
- extractCoef,Fourier\_series-method  
(extractCoef), [7](#)
- family, [11](#), [21](#)
- fc.beta, [8](#)
- fc.beta,fcQR-method (fc.beta), [8](#)
- fc.beta,fcRegression-method (fc.beta), [8](#)
- fcQR, [9](#), [26](#)
- fcRegression, [10](#), [22](#), [26](#)
- fourier\_basis\_expansion, [9](#), [11](#), [13](#), [19](#), [21](#)
- fourier\_basis\_expansion,functional\_variable,integer-method  
(fourier\_basis\_expansion), [13](#)
- Fourier\_series, [3](#), [7](#), [13](#), [25](#)
- Fourier\_series (Fourier\_series-class),  
[14](#)
- Fourier\_series-class, [14](#)
- FourierSeries2fun, [3](#), [12](#)
- FourierSeries2fun,Fourier\_series,numeric-method  
(FourierSeries2fun), [12](#)
- functional\_variable, [5](#), [7](#), [14](#)
- functional\_variable  
(functional\_variable-class), [15](#)
- functional\_variable-class, [15](#)
- glmer, [11](#)
- lmer, [11](#)
- ME.fcLR\_IV, [15](#)
- ME.fcQR\_CLS, [17](#)
- ME.fcQR\_IV.SIMEX, [18](#)
- ME.fcRegression\_MEM, [20](#)
- MECfda.data.sim.0.0, [22](#)
- MECfda.data.sim.0.1, [22](#)
- MECfda.data.sim.0.2, [23](#)
- MECfda.data.sim.0.3, [23](#)
- MECfda.data.sim.1.0, [23](#)
- MECfda.data.sim.1.1, [23](#)
- MECfda.data.sim.1.2, [23](#)
- MECfda.data.sim.1.3, [24](#)
- plot,bspline\_series-method, [24](#)
- plot,Fourier\_series-method, [25](#)
- predict.fcQR, [25](#)
- predict.fcRegression, [26](#)
- predict.glm, [26](#), [27](#)
- predict.lm, [26](#), [27](#)
- predict.merMod, [26](#), [27](#)

`predict.rq`, [26](#)

`rq`, [9](#)