

# Package ‘fctbases’

November 23, 2020

**Type** Package

**Title** Functional Bases

**Version** 1.0.0

**Date** 2020-11-20

**Author** Niels Olsen

**Maintainer** Niels Olsen <nalo@dtu.dk>

**Description** Easy-to-use, very fast implementation of various functional bases. Easily used together with other packages.

A functional basis is a collection of basis functions  $[\phi_1, \dots, \phi_n]$  that can represent a smooth function, i.e.  $f(t) = \sum c_k \phi_k(t)$ .

As of version 1.0, this package includes B-splines, Fourier bases and polynomials.

**URL** <https://github.com/naolsen/fctbases>

**License** GPL-3

**Imports** Rcpp (>= 0.12.19)

**LinkingTo** Rcpp, RcppArmadillo

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2020-11-23 08:50:11 UTC

## R topics documented:

|                           |          |
|---------------------------|----------|
| fctbases-package          | 2        |
| Functional basis function | 2        |
| make.bspline.basis        | 3        |
| make.fourier.basis        | 3        |
| make.pol.basis            | 4        |
| make.std.bspline.basis    | 5        |
| object.info               | 6        |
| <b>Index</b>              | <b>7</b> |

fctbases-package      *fctbases: Functional bases*

---

### Description

Easy-to-use, very fast implementations of various functional bases

### Details

fctbases is a fast and easy implementation of functional bases in R. Simply initialize the desired basis, which returns function of class fctbases.

Internally, functions are stored as C++ objects are stored in C++.

---

Functional basis function  
*Functional basis function*

---

### Description

A fctbases object is a function of class fctbases which takes three arguments (*t*, *x*, *deriv*)

### Arguments

|              |  |
|--------------|--|
| <i>t</i>     | time points  |
| <i>x</i>     | vector of coefficients (optional)                                |
| <i>deriv</i> | Should the derivative be used and which order? Defaults to FALSE |

### Details

If *deriv* is FALSE or zero, the function itself is evaluated. If *deriv* is one or TRUE, the first derivative is evaluated. If *deriv* is two, the second derivative is evaluated.

### Value

Returns a matrix if *x* is missing, and a vector if *x* is provided.

---

make.bspline.basis      *Make b-spline basis*

---

**Description**

Make B-spline basis

**Usage**

```
make.bspline.basis(knots, order = 4)
```

**Arguments**

|       |   |
|-------|---|
| knots | Knots of the basis, including endpoints |
| order | Spline order. Defaults to 4.            |

**Value**

Function of class "fctbasis"

**See Also**

[Functional basis function](#)

**Examples**

```
## B-spline with equidistant knots with 13 basis function
bf <- make.bspline.basis(knots = 0:10, order = 4)

## B-spline of order 2 (ie. a linear approximation) with some uneven knots
bf <- make.bspline.basis(knots = c(-1.3, 0, 0.5, 0.7, 1.1), order = 2)
```

---

make.fourier.basis      *Make fourier basis*

---

**Description**

Make fourier basis

**Usage**

```
make.fourier.basis(range, order, use.trig.id = FALSE)
```

**Arguments**

|             |  |
|-------------|--|
| range       | Left and right end points.                         |
| order       | Order of harmonics                                 |
| use.trig.id | Use trigonometrical identities with this function? |

**Details**

The number of basis elements (degrees of freedom) is  $2 * \text{order} + 1$ .

The basis functions are ordered [1, sin(t), cos(t), sin(2t), cos(2t), ...]

Using trigonometrical identities is faster, but introduces (negligible) round-off errors.

**Value**

Function of class "fctbasis"

**See Also**

[Functional basis function](#)

**Examples**

```
## A fourier basis with period 1 and 11 basis functions.  
bf <- make.fourier.basis(c(0,1), order = 5)
```

---

|                |                              |
|----------------|------------------------------|
| make.pol.basis | <i>Make polynomial basis</i> |
|----------------|------------------------------|

---

**Description**

Make polynomial basis

**Usage**

```
make.pol.basis(order)
```

**Arguments**

|       |                                    |
|-------|------------------------------------|
| order | Order of polynomial (= degree + 1) |
|-------|------------------------------------|

**Details**

The polynomial basis is ordered [1, t, t<sup>2</sup>, t<sup>3</sup>, ..., t<sup>n</sup>]

**Value**

Function of class "fctbasis"

**See Also**

[Functional basis function](#)

**Examples**

```
## A four-degree polynomial
mypol <- make.pol.basis(order = 5)
```

---

```
make.std.bspline.basis
      'Standard' B-spline basis
```

---

**Description**

This initializes a bspline of order 4 with uniformly places knots.  $df = \text{intervals} + 3$ .

**Usage**

```
make.std.bspline.basis(range = c(0, 1), intervals)
```

**Arguments**

|           |                      |
|-----------|----------------------|
| range     | End points of spline |
| intervals | Number of intervals  |

**Details**

make.std.bspline.basis uses a different internal implementation than make.bspline.basis, but is not conclusively faster.

**Value**

function

**See Also**

[Functional basis function](#), [make.bspline.basis](#)

**Examples**

```
## 16 equidistant knots between 0 and 2 (both included)
bf <- make.std.bspline.basis(range = c(0,2), intervals = 15)
```

---

`object.info`*Functional basis info*

---

**Description**

This function returns details about a functional basis.

**Usage**

```
object.info(fctbasis)
```

**Arguments**

`fctbasis`      object of class `fctbasis`

**Value**

A named list including no. of basis, type of basis, and possibly additional information.

# Index

fctbases (fctbases-package), [2](#)  
fctbases-package, [2](#)  
Functional basis function, [2](#), [3–5](#)

make.bspline.basis, [3](#), [5](#)  
make.fourier.basis, [3](#)  
make.pol.basis, [4](#)  
make.std.bspline.basis, [5](#)

object.info, [6](#)