

# Package ‘ffstream’

May 30, 2023

**Title** Forgetting Factor Methods for Change Detection in Streaming Data

**Version** 0.1.7.2

**Date** 2023-05-25

**Author** Dean Bodenham

**Maintainer** Dean Bodenham <deanbodenhampkgs@gmail.com>

**Description** An implementation of the adaptive forgetting factor scheme described in Bodenham and Adams (2016) <doi:10.1007/s11222-016-9684-8> which adaptively estimates the mean and variance of a stream in order to detect multiple changepoints in streaming data. The implementation is in 'C++' and uses 'Rcpp'. Additionally, implementations of the fixed forgetting factor scheme from the same paper, as well as the classic cumulative sum ('CUSUM') and exponentially weighted moving average ('EWMA') methods, are included.

**Depends** R (>= 4.1.0), Rcpp (>= 1.0.0)

**License** GPL-2 | GPL-3

**LinkingTo** Rcpp

**Imports** methods

**Suggests** testthat (>= 2.0.0), knitr, rmarkdown

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2023-05-30 10:20:02 UTC

## R topics documented:

computeAFFMean . . . . .	2
computeFFFMean . . . . .	3
demo_ffstream . . . . .	3
detectAFFMean . . . . .	5

detectCUSUMMean . . . . .	7
detectEWMAMean . . . . .	9
detectFFFMean . . . . .	11
ffstream . . . . .	12
ffstream_vignette . . . . .	13
initAFFMean . . . . .	13
initAFFMeanCD . . . . .	14
initCUSUMMeanCD . . . . .	15
initEWMAMeanCD . . . . .	15
initFFFMean . . . . .	16
initFFFMeanCD . . . . .	17

<b>Index</b>	<b>18</b>
--------------	-----------

---

computeAFFMean	<i>Quick computation of AFF mean of a given vector</i>
----------------	--

---

### Description

Given a vector  $x$  and a value  $\eta$  for step-size in the stochastic gradient descent for the adaptive forgetting factor, this returns the value of the fixed forgetting factor mean  $\bar{x}_{N, \vec{\lambda}}$ , where  $N$  is the length of  $x$ . Algorithm is implemented in 'C++'.

### Usage

```
computeAFFMean(x = c(0), eta = 0.01)
```

### Arguments

$x$	Vector of numeric values values. Default is $c(0)$ , a vector of one element (zero)
$\eta$	Value for the step size in the gradient descent step. Default is $\eta=0.01$ .

### Value

The adaptive forgetting factor mean (scalar).

### Author

Dean Bodenham

### References

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation*. Statistics and Computing doi:10.1007/s11222-016-9684-8

### See Also

[computeFFFMean](#)

---

computeFFFMean	<i>Quick computation of FFF mean of a given vector</i>
----------------	--

---

**Description**

Given a vector  $x$  and a value  $\lambda$  for a fixed forgetting factor, returns the value of the fixed forgetting factor mean  $\bar{x}_{N,\lambda}$ , where  $N$  is the length of  $x$ . Algorithm is implemented in 'C++'.

**Usage**

```
computeFFFMean(x = c(0), lambda = 0.99)
```

**Arguments**

$x$	Vector of numeric values values. Default is $c(0)$ , a vector of one element (zero)
$\lambda$	Value for the fixed forgetting factor in $[0, 1]$ . Default is $\lambda=0.99$ .

**Value**

The fixed forgetting factor mean (scalar).

**Author**

Dean Bodenham

**References**

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation*. Statistics and Computing doi:10.1007/s11222-016-9684-8

**See Also**

[computeAFFMean](#)

---

demo_ffstream	<i>Demo for ffstream</i>
---------------	--------------------------

---

**Description**

Provides a demonstration of the AFF method detecting changepoints in a stream.

**Usage**

```
demo_ffstream(showPlot = FALSE, returnStream = FALSE, plotSmall = FALSE)
```

## Arguments

showPlot	Boolean flag; if TRUE, then a plot is generated. Default is FALSE.
returnStream	Boolean flag; if TRUE, then return the stream as part of the list returned by the demo. Default is FALSE.
plotSmall	Boolean flag; if TRUE, creates a small plot, as needed for the vignette. Default is FALSE.

## Details

This method generates a stream with three changepoints, and finds the changepoints with AFF. Also creates a plot of the data and the changepoints if the showPlot flag is set to TRUE. The observations are shown in black, the true changepoints are shown as red dotted vertical lines, and the detected (estimated) changepoints are shown as blue dashed lines. The following is returned in a list:

**tau** The location of the true changepoints.

**tauhat** The detected (estimated) changepoints.

**method** The method used, in this case AFF.

**param** The data frame with the parameters used in the AFF method, in this case,

**alpha** The significance level,

**eta** The step size in the gradient descent, whose value is not particularly important,

**BL** The length of the burn-in period.

## Value

Print output to screen, and plot (optional).

## Author

Dean Bodenham

## References

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation*. Statistics and Computing doi:10.1007/s11222-016-9684-8

## Examples

```
df <- demo_ffstream()
```

```
demo_ffstream(showPlot=TRUE)
```

---

detectAFFMean	<i>Detect a change/changes in a vector using AFF method</i>
---------------	---

---

### Description

Given a vector  $x$ , use the fff method to sequentially detect changes (or a single change) in the MEAN of the vector.

### Usage

```
detectAFFMean(
  x,
  alpha = 0.01,
  eta = 0.01,
  BL = 50,
  multiple = TRUE,
  single = !multiple,
  usePrechange = FALSE,
  prechangeMean = NULL,
  prechangeSigma = NULL,
  prechangeVar = NULL,
  skipCheck = FALSE
)
```

### Arguments

<code>x</code>	The vector (stream) in which to detect change(s).
<code>alpha</code>	The value for the threshold. Default is <code>alpha=0.01</code> .
<code>eta</code>	The value for of the step size in the gradient descent step. Results show that values of 0.1, 0.01, 0.001 all produce similar results. Default is <code>eta=0.01</code> .
<code>BL</code>	The burn-in length. Default is <code>BL=50</code> .
<code>multiple</code>	Boolean to use to decide whether to detect multiple changes or only a single change. Default is <code>TRUE</code> (i.e. detect multiple changes).
<code>single</code>	Boolean to use to decide whether to detect only a single change or multiple changes. Set to <code>!multiple</code> , i.e. default is <code>FALSE</code> . If both <code>single</code> and <code>multiple</code> are set to <code>TRUE</code> , then only a single change will be detected; if both set to <code>FALSE</code> then multiple changes will be detected (i.e. <code>single</code> dominates).
<code>usePrechange</code>	Boolean indicating whether prechange parameters (mean and variance) are known and will be used (or not). Default is <code>FALSE</code> . If <code>TRUE</code> , then prechange mean and standard deviation variance must be specified. See parameters <code>prechangeMean</code> , <code>prechangeSigma</code> and <code>prechangeVar</code> .
<code>prechangeMean</code>	Value to be used for the prechange mean. Default is <code>NULL</code> . If <code>prechangeKnown = TRUE</code> and value is <code>NULL</code> , this will result in an error.



---

detectCUSUMMean	<i>Detect a change/changes in a vector using 'CUSUM' method</i>
-----------------	---

---

### Description

Given a vector  $x$ , use the 'CUSUM' method to sequentially detect changes (or a single change) in the MEAN of the vector.

### Usage

```
detectCUSUMMean(
  x,
  k = 0.25,
  h = 8,
  BL = 50,
  multiple = TRUE,
  single = !multiple,
  usePrechange = FALSE,
  prechangeMean = NULL,
  prechangeSigma = NULL,
  prechangeVar = NULL,
  skipCheck = FALSE
)
```

### Arguments

<code>x</code>	The vector (stream) in which to detect change(s).
<code>k</code>	control parameter for 'CUSUM'. Default is 0.25.
<code>h</code>	control parameter for 'CUSUM'. Default is 8.00.
<code>BL</code>	The burn-in length. Default is BL=50.
<code>multiple</code>	Boolean to use to decide whether to detect multiple changes or only a single change. Default is TRUE (i.e. detect multiple changes).
<code>single</code>	Boolean to use to decide whether to detect only a single change or multiple changes. Set to !multiple, i.e. default is FALSE. If both <code>single</code> and <code>multiple</code> are set to TRUE, then only a single change will be detected; if both set to FALSE then multiple changes will be detected (i.e. single dominates).
<code>usePrechange</code>	Boolean indicating whether prechange parameters (mean and variance) are known and will be used (or not). Default is FALSE. If TRUE, then prechange mean and standard deviation variance must be specified. See parameters <code>prechangeMean</code> , <code>prechangeSigma</code> and <code>prechangeVar</code> .
<code>prechangeMean</code>	Value to be used for the prechange mean. Default is NULL. If <code>prechangeKnown = TRUE</code> and value is NULL, this will result in an error.
<code>prechangeSigma</code>	Value to be used for the prechange standard deviation. Default is NULL. If <code>prechangeKnown = TRUE</code> and value is NULL, this will result in an error, unless <code>prechangeVar</code> is not NULL.





---

detectEWMAMean	<i>Detect a change/changes in a vector using 'EWMA' method</i>
----------------	--

---

### Description

Given a vector  $x$ , use the 'EWMA' method to sequentially detect changes (or a single change) in the MEAN of the vector.

### Usage

```
detectEWMAMean(
  x,
  r = 0.25,
  L = 3,
  BL = 50,
  multiple = TRUE,
  single = !multiple,
  usePrechange = FALSE,
  prechangeMean = NULL,
  prechangeSigma = NULL,
  prechangeVar = NULL,
  skipCheck = FALSE
)
```

### Arguments

$x$	The vector (stream) in which to detect change(s).
$r$	Control parameter for EWMA. Must be in range $[0, 1]$ . Default is $r=0.25$ .
$L$	Control parameter for EWMA. Default is $L=3.00$
$BL$	The burn-in length. Default is $BL=50$ .
<code>multiple</code>	Boolean to use to decide whether to detect multiple changes or only a single change. Default is TRUE (i.e. detect multiple changes).
<code>single</code>	Boolean to use to decide whether to detect only a single change or multiple changes. Set to <code>!multiple</code> , i.e. default is FALSE. If both <code>single</code> and <code>multiple</code> are set to TRUE, then only a single change will be detected; if both set to FALSE then multiple changes will be detected (i.e. <code>single</code> dominates).
<code>usePrechange</code>	Boolean indicating whether prechange parameters (mean and variance) are known and will be used (or not). Default is FALSE. If TRUE, then prechange mean and standard deviation variance must be specified. See parameters <code>prechangeMean</code> , <code>prechangeSigma</code> and <code>prechangeVar</code> .
<code>prechangeMean</code>	Value to be used for the prechange mean. Default is NULL. If <code>prechangeKnown = TRUE</code> and value is NULL, this will result in an error.
<code>prechangeSigma</code>	Value to be used for the prechange standard deviation. Default is NULL. If <code>prechangeKnown = TRUE</code> and value is NULL, this will result in an error, unless <code>prechangeVar</code> is not NULL.



---

detectFFFMean	<i>Detect a change/changes in a vector using FFF method</i>
---------------	---

---

### Description

Given a vector  $x$ , use the FFF method to sequentially detect changes (or a single change) in the MEAN of the vector.

### Usage

```
detectFFFMean(
  x,
  lambda = 0.95,
  alpha = 0.01,
  BL = 50,
  multiple = TRUE,
  single = !multiple,
  usePrechange = FALSE,
  prechangeMean = NULL,
  prechangeSigma = NULL,
  prechangeVar = NULL,
  skipCheck = FALSE
)
```

### Arguments

<code>x</code>	The vector (stream) in which to detect change(s).
<code>lambda</code>	The value for the forgetting factor. Default is <code>lambda=0.95</code> .
<code>alpha</code>	The value for the threshold. Default is <code>alpha=0.01</code> .
<code>BL</code>	The burn-in length. Default is <code>BL=50</code> .
<code>multiple</code>	Boolean to use to decide whether to detect multiple changes or only a single change. Default is <code>TRUE</code> (i.e. detect multiple changes).
<code>single</code>	Boolean to use to decide whether to detect only a single change or multiple changes. Set to <code>!multiple</code> , i.e. default is <code>FALSE</code> . If both <code>single</code> and <code>multiple</code> are set to <code>TRUE</code> , then only a single change will be detected; if both set to <code>FALSE</code> then multiple changes will be detected (i.e. <code>single</code> dominates).
<code>usePrechange</code>	Boolean indicating whether prechange parameters (mean and variance) are known and will be used (or not). Default is <code>FALSE</code> . If <code>TRUE</code> , then prechange mean and standard deviation variance must be specified. See parameters <code>prechangeMean</code> , <code>prechangeSigma</code> and <code>prechangeVar</code> .
<code>prechangeMean</code>	Value to be used for the prechange mean. Default is <code>NULL</code> . If <code>prechangeKnown = TRUE</code> and value is <code>NULL</code> , this will result in an error.
<code>prechangeSigma</code>	Value to be used for the prechange standard deviation. Default is <code>NULL</code> . If <code>prechangeKnown = TRUE</code> and value is <code>NULL</code> , this will result in an error, unless <code>prechangeVar</code> is not <code>NULL</code> .

prechangeVar	Value to be used for the prechange variance. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeSigma is not NULL. prechangeVar is set to $\sqrt{\text{prechangeSigma}}$ .
skipCheck	A boolean which allows the function to skip the check of the stream. Default is FALSE.

**Value**

A list with the following elements:

tauhat A vector of the changepoints found.

**Author**

Dean Bodenham

**References**

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation*. Statistics and Computing doi:10.1007/s11222-016-9684-8

**Examples**

```
# create a stream with three changepoints
set.seed(8)
x <- rnorm(400, 5, 1) + rep(c(0:3), each=100) # mean is 5 and s.d. is 1

# multiple changepoints
list_fff <- detectFFFMean(x, alpha=0.01, lambda=0.95, BL=50, multiple=TRUE)

# now only a single (the first) changepoint
list_fff2 <- detectFFFMean(x, alpha=0.01, lambda=0.95, BL=50, single=TRUE)

# now only a single (the first) changepoint, but with the prechange
# mean and variance known
list_fff3 <- detectFFFMean(x, alpha=0.01, lambda=0.95, BL=50, single=TRUE,
                           prechangeMean=5, prechangeSigma=1)
```

---

ffstream

*ffstream package 0.1.7*

---

**Description**

This function gives information on the package and how to see the vignette.

**Usage**

```
ffstream()
```

---

ffstream\_vignette      *ffstream vignette 0.1.7*

---

**Description**

This function opens the fstream vignette.

**Usage**

```
ffstream_vignette()
```

**Value**

Does not return anything.

**Examples**

```
ffstream_vignette()
```

---

initAFFMean      *Initialisation of AFF mean*

---

**Description**

This function makes it simple to initialise an AFF object.

**Usage**

```
initAFFMean(eta = 0.1)
```

**Arguments**

eta                      The value of the step size in the gradient descent. Default is eta=0.01.

**Value**

An AFF object.

**Examples**

```
library(Rcpp)
aff1 <- initAFFMean()      # initialises with eta=0.01

aff2 <- initAFFMean(0.1) # initialises with eta=0.1
```

---

initAFFMeanCD	<i>Initialisation of AFF change detector</i>
---------------	--

---

### Description

This function makes it simple to initialise an FFF object.

### Usage

```
initAFFMeanCD(alpha = 0.01, eta = 0.01, BL = 50)
```

### Arguments

alpha	The value of the significance level. Default value is 0.01, although it is recommended that the user set this parameter.
eta	The value of the step-size in the gradient descent. Default is eta=0.01.
BL	The length of the burn-in region. Default value is BL=50. Must be at least greater than or equal to 2. No maximum. However, there is an exception: BL=0 also works, but in this case the user needs to specify the streamEstMean and streamEstSigma; see

### Value

An AFF mean change detector object.

### Examples

```
library(Rcpp)
affmeancd1 <- initAFFMeanCD()           # initialises with alpha=0.01,
                                         # eta=0.01 and BL=50

affmeancd2 <- initAFFMeanCD(alpha=0.005, eta=0.1, BL=100)

affmeancd3 <- initAFFMeanCD(alpha=0.005, eta=0.1, BL=0)   # Example 3
affmeancd3$streamEstMean <- 0
affmeancd3$streamEstSigma <- 1
```

---

initCUSUMMeanCD      *Initialisation of 'CUSUM'*

---

### Description

This function makes it simple to initialise a 'CUSUM' object.

### Usage

```
initCUSUMMeanCD(k = 0.25, h = 8, BL = 50)
```

### Arguments

k                      One of the 'CUSUM' control parameters. Default value is k=0.25.  
h                      One of the 'CUSUM' control parameters. Default value is h=8.00.  
BL                     The burn-in length to be used with a 'CUSUM' change detector. Default value is BL=50.

### Value

A CUSUM mean change detector object.

### Examples

```
library(Rcpp)
c1 <- initCUSUMMeanCD()      # initialises with k=0.25, h=8.00, BL=50

c2 <- initCUSUMMeanCD(k=0.5, h=4.00, BL=30)
```

---

initEWMAMeanCD      *Initialisation of EWMA*

---

### Description

This function makes it simple to initialise a 'EWMA' object.

### Usage

```
initEWMAMeanCD(r = 0.2, L = 3, BL = 50)
```

### Arguments

r                      One of the 'EWMA' control parameters. Default value is r=0.20.  
L                      One of the 'EWMA' control parameters. Default value is L=3.00.  
BL                     The burn-in length to be used with a 'EWMA' change detector. Default value is BL=50.

**Value**

A EWMA mean change detector object.

**Examples**

```
library(Rcpp)
e1 <- initEWMAMeanCD()           #initialises with r=0.20, L=3.00

e1 <- initEWMAMeanCD(r=0.05, L=0.275) #initialises with r=0.20, L=3.00
```

---

initFFFMean	<i>Initialisation of FFF</i>
-------------	------------------------------

---

**Description**

This function makes it simple to initialise an FFF mean object.

**Usage**

```
initFFFMean(lambda = 1)
```

**Arguments**

lambda            The value of the fixed forgetting factor. Default is lambda=1.

**Value**

A FFF object.

**Examples**

```
library(Rcpp)
fff1 <- initFFFMean()    # initialises with lambda=1

fff2 <- initFFFMean(0.9) # initialises with lambda=0.9
```





# Index

[computeAFFMean](#), [2](#), [3](#)

[computeFFFMean](#), [2](#), [3](#)

[demo\\_ffstream](#), [3](#)

[detectAFFMean](#), [5](#)

[detectCUSUMMean](#), [7](#)

[detectEWMAMean](#), [9](#)

[detectFFFMean](#), [11](#)

[ffstream](#), [12](#)

[ffstream\\_vignette](#), [13](#)

[initAFFMean](#), [13](#)

[initAFFMeanCD](#), [14](#)

[initCUSUMMeanCD](#), [15](#)

[initEWMAMeanCD](#), [15](#)

[initFFFMean](#), [16](#)

[initFFFMeanCD](#), [17](#)