

# Package ‘edm1.sequence’

July 29, 2024

**Title** Set of tools to manipulate time series  
**Version** 2.0.0.0  
**Description** Provides set of functions to manipulate time series dataset; create variables that designates the value of the individual at n-x , handle missing values...  
**License** GPL (==3)  
**Encoding** UTF-8  
**Roxygen** list(markdown = TRUE)  
**RoxygenNote** 7.3.1  
**Imports** stringr

## Contents

historic_sequence1 . . . . .	1
historic_sequence2 . . . . .	2
sequence_na_mean1 . . . . .	4
sequence_na_mean2 . . . . .	5
sequence_na_med1 . . . . .	6
sequence_na_med2 . . . . .	7
<b>Index</b>	<b>9</b>

---

historic_sequence1	<i>historic_sequence1</i>
--------------------	---------------------------

---

## Description

Allow to perform a pivot wider on a sequential dataset (here the type is dataframe), each variable will be duplicated in a column to show the value to this variable at n - 1 for each individual, see examples.

## Usage

```
historic_sequence1(inpt_datf, bf_ = 1)
```

## Arguments

inpt\_datf is the input dataframe  
bf\_ is the number of previous value of the individual it will search for, see examples

**Examples**

```

set.seed(123)
var1 <- round(runif(n = 14, min = 100, max = 122))
set.seed(123)
var2 <- round(runif(n = 14, min = 14, max = 20))

datf <- data.frame("ids" = c(20, 20, 20, 20, 19, 19, 19, 18, 18, 18, 18,
                             17, 17, 17),
                  "individual" = c("oui", "non", "peut1", "peut2",
                                   "oui", "peut1", "peut2"),
                  "var1" = var1,
                  "var2" = var2)

print(datf)

```

	ids	individual	var1	var2
1	20	oui	106	16
2	20	non	117	19
3	20	peut1	109	16
4	20	peut2	119	19
5	19	oui	121	20
6	19	peut1	101	14
7	19	peut2	112	17
8	18	oui	120	19
9	18	non	112	17
10	18	peut1	110	17
11	18	peut2	121	20
12	17	oui	110	17
13	17	peut1	115	18
14	17	peut2	113	17

```
historic_sequence1(inpt_datf = datf, bf_ = 2)
```

	id_seq	individual	var1-1	var1-2	var2-1	var2-2
1	20	oui	121	120	20	19
2	20	non	NA	112	NA	17
3	20	peut1	101	110	14	17
4	20	peut2	112	121	17	20
5	19	oui	120	110	19	17
6	19	peut1	110	115	17	18
7	19	peut2	121	113	20	17

```
historic_sequence1(inpt_datf = datf, bf_ = 3)
```

	id_seq	individual	var1-1	var1-2	var1-3	var2-1	var2-2	var2-3
1	20	oui	121	120	110	20	19	17
2	20	non	NA	112	NA	NA	17	NA
3	20	peut1	101	110	115	14	17	18
4	20	peut2	112	121	113	17	20	17

## Description

Allow to perform a pivot wider on a sequential dataset (here the type is dataframe), each variable will be duplicated in a column to show the value to this variable at  $n - 1$  for each individual, see examples.

## Usage

```
historic_sequence2(inpt_datf, bf_ = 1)
```

## Arguments

`inpt_datf` is the input dataframe  
`bf_` is the number of previous value of the individual it will search for, see examples

## Examples

```
set.seed(123)
var1 <- round(runif(n = 14, min = 100, max = 122))
set.seed(123)
var2 <- round(runif(n = 14, min = 14, max = 20))

datf <- data.frame("ids" = c(20, 20, 20, 20, 19, 19, 19, 18, 18, 18, 18,
                             17, 17, 17),
                  "individual" = c("oui", "non", "peut1", "peut2",
                                   "oui", "peut1", "peut2"),
                  "var1" = var1,
                  "var2" = var2)

print(datf)
```

	ids	individual	var1	var2
1	20	oui	106	16
2	20	non	117	19
3	20	peut1	109	16
4	20	peut2	119	19
5	19	oui	121	20
6	19	peut1	101	14
7	19	peut2	112	17
8	18	oui	120	19
9	18	non	112	17
10	18	peut1	110	17
11	18	peut2	121	20
12	17	oui	110	17
13	17	peut1	115	18
14	17	peut2	113	17

```
print(historic_sequence2(inpt_datf = datf, bf_ = 2))
```

	id_seq	individual	var1-0	var1-1	var1-2	var2-0	var2-1	var2-2
1	20	oui	106	121	120	16	20	19
2	20	non	117	NA	112	19	NA	17
3	20	peut1	109	101	110	16	14	17
4	20	peut2	119	112	121	19	17	20
5	19	oui	121	120	110	20	19	17
6	19	peut1	101	110	115	14	17	18
7	19	peut2	112	121	113	17	20	17

```
print(historic_sequence2(inpt_datf = datf, bf_ = 3))
```

	id_seq	individual	var1-0	var1-1	var1-2	var1-3	var2-0	var2-1	var2-2	var2-3
1	20	oui	106	121	120	110	16	20	19	17
2	20	non	117	NA	112	NA	19	NA	17	NA
3	20	peut1	109	101	110	115	16	14	17	18
4	20	peut2	119	112	121	113	19	17	20	17

---

```
sequence_na_mean1    sequence_na_mean1
```

---

## Description

In a dataframe generated by the function `historic_sequence1`, convert all NA to the mean of the values at the same variable for the individual at the id where the NA occurs, see examples (only accepts numeric variables)

## Usage

```
sequence_na_mean1(inpt_datf, bf_)
```

## Arguments

`inpt_datf`      is the input dataframe

## Examples

```
set.seed(123)
var1 <- round(runif(n = 14, min = 100, max = 122))
set.seed(123)
var2 <- round(runif(n = 14, min = 14, max = 20))

datf <- data.frame("ids" = c(20, 20, 20, 20, 19, 19, 19, 18, 18, 18, 18,
17, 17, 17),
"individual" = c("oui", "non", "peut1", "peut2",
"oui", "peut1", "peut2"),
"var1" = var1,
"var2" = var2)
datf <- historic_sequence1(inpt_datf = datf, bf_ = 2)
datf[3, 4] <- NA
datf[6, 4] <- NA
datf[1, 3] <- NA
print(datf)
```

	id_seq	individual	var1-1	var1-2	var2-1	var2-2
1	20	oui	NA	120	20	19
2	20	non	NA	112	NA	17
3	20	peut1	101	NA	14	17
4	20	peut2	112	121	17	20
5	19	oui	120	110	19	17
6	19	peut1	110	NA	17	18
7	19	peut2	121	113	20	17

```
print(sequence_na_mean1(inpt_datf = datf, bf_ = 2))
```

	id_seq	individual	var1-1	var1-2	var2-1	var2-2
1	20	oui	115	120.0	20	19
2	20	non	112	112.0	17	17
3	20	peut1	101	105.5	14	17
4	20	peut2	112	121.0	17	20
5	19	oui	120	110.0	19	17
6	19	peut1	110	105.5	17	18
7	19	peut2	121	113.0	20	17

---

```
sequence_na_mean2    sequence_na_mean2
```

---

## Description

In a dataframe generated by the function `historic_sequence1`, convert all NA to the mean of the values at the same variable for the individual at the id where the NA occurs, see examples (only accepts numeric variables)

## Usage

```
sequence_na_mean2(inpt_datf, bf_)
```

## Arguments

<code>inpt_datf</code>	is the input dataframe
<code>bf_</code>	is how at how many n-1 we look for the value of the variables for the individual at time index n

## Examples

```
set.seed(123)
var1 <- round(runif(n = 14, min = 100, max = 122))
set.seed(123)
var2 <- round(runif(n = 14, min = 14, max = 20))

datf <- data.frame("ids" = c(20, 20, 20, 20, 19, 19, 19, 18, 18, 18, 18,
17, 17, 17),
"individual" = c("oui", "non", "peut1", "peut2",
"oui", "peut1", "peut2"),
"var1" = var1,
"var2" = var2)
datf <- historic_sequence2(inpt_datf = datf, bf_ = 2)
datf[3, 4] <- NA
datf[6, 4] <- NA
datf[1, 3] <- NA
print(datf)
```

	id_seq	individual	var1-0	var1-1	var1-2	var2-0	var2-1	var2-2
1	20	oui	NA	121	120	16	NA	19
2	20	non	117	NA	112	19	NA	17
3	20	peut1	109	NA	110	16	14	17

```

4      20      peut2      119      112      121      19      17      20
5      19        oui      121      120      110      20      19      17
6      19      peut1      101      NA      115      14      17      18
7      19      peut2      112      121      113      17      20      17

```

```
print(sequence_na_mean2(inpt_datf = datf, bf_ = 2))
```

```

  id_seq individual var1-0  var1-1 var1-2 var2-0 var2-1 var2-2
1      20        oui    117 121.0000    120     16     18     19
2      20        non    117 114.5000    112     19     18     17
3      20      peut1    109 108.3333    110     16     14     17
4      20      peut2    119 112.0000    121     19     17     20
5      19        oui    121 120.0000    110     20     19     17
6      19      peut1    101 108.3333    115     14     17     18
7      19      peut2    112 121.0000    113     17     20     17

```

---

sequence_na_med1	<i>sequence_na_med1</i>
------------------	-------------------------

---

## Description

In a dataframe generated by the function `historic_sequence1`, convert all NA to the median of the values at the same variable for the individual at the id where the NA occurs, see examples (only accepts numeric variables)

## Usage

```
sequence_na_med1(inpt_datf, bf_)
```

## Arguments

<code>inpt_datf</code>	is the input dataframe
<code>bf_</code>	is how at how many n - 1 we look for the value of the variables for the individual at time index n

## Examples

```

set.seed(123)
var1 <- round(runif(n = 14, min = 100, max = 122))
set.seed(123)
var2 <- round(runif(n = 14, min = 14, max = 20))

datf <- data.frame("ids" = c(20, 20, 20, 20, 19, 19, 19, 18, 18, 18, 18,
17, 17, 17),
"individual" = c("oui", "non", "peut1", "peut2",
"oui", "peut1", "peut2"),
"var1" = var1,
"var2" = var2)
datf <- historic_sequence1(inpt_datf = datf, bf_ = 2)
datf[3, 4] <- NA
datf[6, 4] <- NA
datf[1, 3] <- NA
print(datf)

```

	id_seq	individual	var1-1	var1-2	var2-1	var2-2
1	20	oui	NA	120	20	19
2	20	non	NA	112	NA	17
3	20	peut1	101	NA	14	17
4	20	peut2	112	121	17	20
5	19	oui	120	110	19	17
6	19	peut1	110	NA	17	18
7	19	peut2	121	113	20	17

```
print(sequence_na_med1(inpt_datf = datf, bf_ = 2))
```

	id_seq	individual	var1-1	var1-2	var2-1	var2-2
1	20	oui	115	120.0	20	19
2	20	non	112	112.0	17	17
3	20	peut1	101	105.5	14	17
4	20	peut2	112	121.0	17	20
5	19	oui	120	110.0	19	17
6	19	peut1	110	105.5	17	18
7	19	peut2	121	113.0	20	17

---

sequence_na_med2	sequence_na_med2
------------------	------------------

---

## Description

In a dataframe generated by the function `historic_sequence2`, convert all NA to the median of the values at the same variable for the individual at the id where the NA occurs, see examples (only accepts numeric variables)

## Usage

```
sequence_na_med2(inpt_datf, bf_, step = 1)
```

## Arguments

<code>inpt_datf</code>	is the input dataframe
<code>bf_</code>	is how at how many n -1 we look for the value of the variables for the individual at time index n

## Examples

```
set.seed(123)
var1 <- round(runif(n = 14, min = 100, max = 122))
set.seed(123)
var2 <- round(runif(n = 14, min = 14, max = 20))
datf <- data.frame("ids" = c(20, 20, 20, 20, 19, 19, 19, 18, 18, 18, 18,
17, 17, 17),
"individual" = c("oui", "non", "peut1", "peut2",
"oui", "peut1", "peut2"),
"var1" = var1,
"var2" = var2)
datf <- historic_sequence2(inpt_datf = datf, bf_ = 2)
```

```

datf[3, 4] <- NA
datf[6, 4] <- NA
datf[1, 3] <- NA
print(datf)

```

	id_seq	individual	var1-0	var1-1	var1-2	var2-0	var2-1	var2-2
1	20	oui	NA	121	120	16	20	19
2	20	non	117	NA	112	19	NA	17
3	20	peut1	109	NA	110	16	14	17
4	20	peut2	119	112	121	19	17	20
5	19	oui	121	120	110	20	19	17
6	19	peut1	101	NA	115	14	17	18
7	19	peut2	112	121	113	17	20	17

```

print(sequence_na_med2(inpt_datf = datf, bf_ = 2))

```

	id_seq	individual	var1-0	var1-1	var1-2	var2-0	var2-1	var2-2
1	20	oui	120	121.0	120	16	20	19
2	20	non	117	114.5	112	19	18	17
3	20	peut1	109	109.0	110	16	14	17
4	20	peut2	119	112.0	121	19	17	20
5	19	oui	121	120.0	110	20	19	17
6	19	peut1	101	109.0	115	14	17	18
7	19	peut2	112	121.0	113	17	20	17



# Index

historic\_sequence1, [1](#)  
historic\_sequence2, [2](#)

sequence\_na\_mean1, [4](#)  
sequence\_na\_mean2, [5](#)  
sequence\_na\_med1, [6](#)  
sequence\_na\_med2, [7](#)