

# Limpeza de Dados para Séries Temporais

```
# Importando libs
require(zoo, quietly = TRUE)
require(data.table, quietly = TRUE)
```

## Limpe seu dados

### Lidando com dados ausentes

Os dados ausentes são ainda mais comuns na análise de séries temporais do que na análise de dados transversais, por que a carga da amostragem longitudinal é bastante pesada:

- Séries temporais incompletas são bastante comuns

Métodos para lidar com dados ausentes mais comuns são:

1. imputação
2. interpolação
3. Exclusão dos períodos de tempos afetados

### Preparando um conj. dados para testar metodologias de imputação

```
path = 'C:\\Users\\mateu\\Documents\\MEGA\\Projetos-git\\analise-pratica-series-temporais_predicao-statistics\\data\\unemp.txt'
unemp = fread(input = path, sep = ',')
head(x = unemp, n = 5)
```

```
##          DATE UNRATE
##      <IDat>  <num>
## 1: 1948-01-01    3.4
## 2: 1948-02-01    3.8
## 3: 1948-03-01    4.0
## 4: 1948-04-01    3.9
## 5: 1948-05-01    3.5
```

```
# Convertendo para date
unemp[, DATE := as.Date(DATE)]
```

```
# definindo o index
setkey(unemp, DATE)
```

```
head(x = unemp, n = 4)
```

```
## Key: <DATE>
##      DATE UNRATE
##      <Date> <num>
## 1: 1948-01-01   3.4
## 2: 1948-02-01   3.8
## 3: 1948-03-01   4.0
## 4: 1948-04-01   3.9
```

Gerando um conj. de dados onde os dados estão aleatoriamente ausentes.

```
set.seed(10)
rand.unemp.idx = sample(1:nrow(unemp), .1*nrow(unemp))

rand.unemp = unemp[-rand.unemp.idx]
```

Gerando um conj. de dados onde os dados possuem maior prob. de ausencia quando o desemprego é alto

```
high.unemp.idx = which(unemp$DATE > 8)
num.to.select  = .2 * length(high.unemp.idx)

high.unemp.idx = sample(high.unemp.idx, )
bias.unemp = unemp[-high.unemp.idx]
```

Como excluimos as linhas da nossa tabela de dados para criar um conj. de dados com dados ausentes, precisamos ler as datas ausentes e o valores NA.

```
todas.datas = seq(
  from = unemp$DATE[1], to = tail(unemp$DATE, 1),
  by = "months"
)
todas.datas[1:4]
```

```
## [1] "1948-01-01" "1948-02-01" "1948-03-01" "1948-04-01"
```

```
rand.unemp = rand.unemp[J(todas.datas), roll = 0]
bias.unemp = bias.unemp[J(todas.datas), roll = 0]
```

```
rand.unemp[, rtp := is.na(UNRATE)]
```

```
head(
  x = rand.unemp[rtp == TRUE], n = 3
)
```

```
##      DATE UNRATE  rtp
##      <Date> <num> <lgcl>
## 1: 1949-01-01    NA  TRUE
## 2: 1950-02-01    NA  TRUE
## 3: 1950-09-01    NA  TRUE
```

**Forward Fill** É transferir o último valor conhecido para o valor ausente anterior.

```
na.locf(coluna, na.rm = FALSE)
```

Imputando no R :

```
# Para aleatorio
rand.unemp[, impute.ff := na.locf(UNRATE, na.rm = FALSE)]

# Para o com maior prob. de desemprego
bias.unemp[, impute.ff := na.locf(UNRATE, na.rm = FALSE)]
```

Plotando

```
# plot de um gráfico de amostra que mostra as partes achatadas

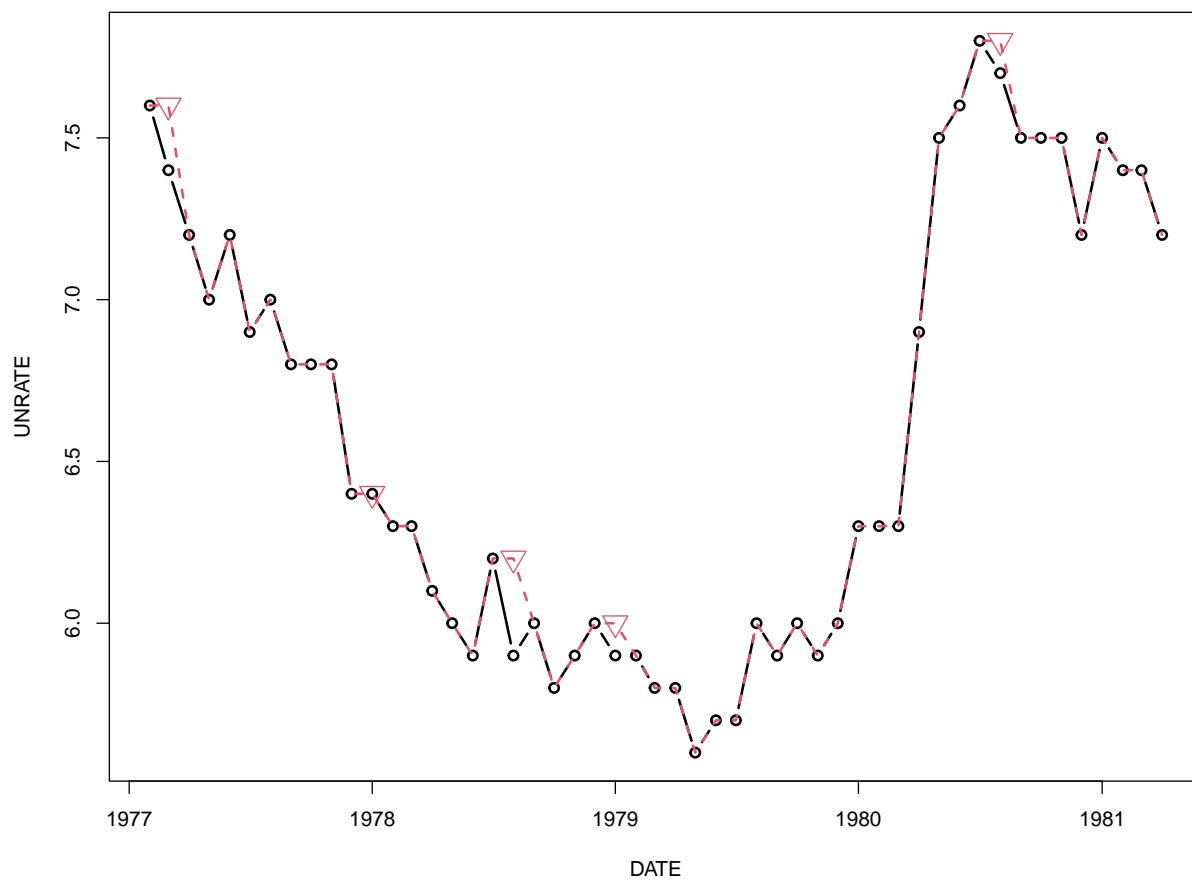
unemp[
  350:400,
  plot(UNRATE, col = 1, lwd = 2, type = 'b')
]
```

## NULL

```
rand.unemp[
  350:400,
  lines(
    UNRATE, impute.ff, col = 2, lwd = 2, lty = 2
  )
]
```

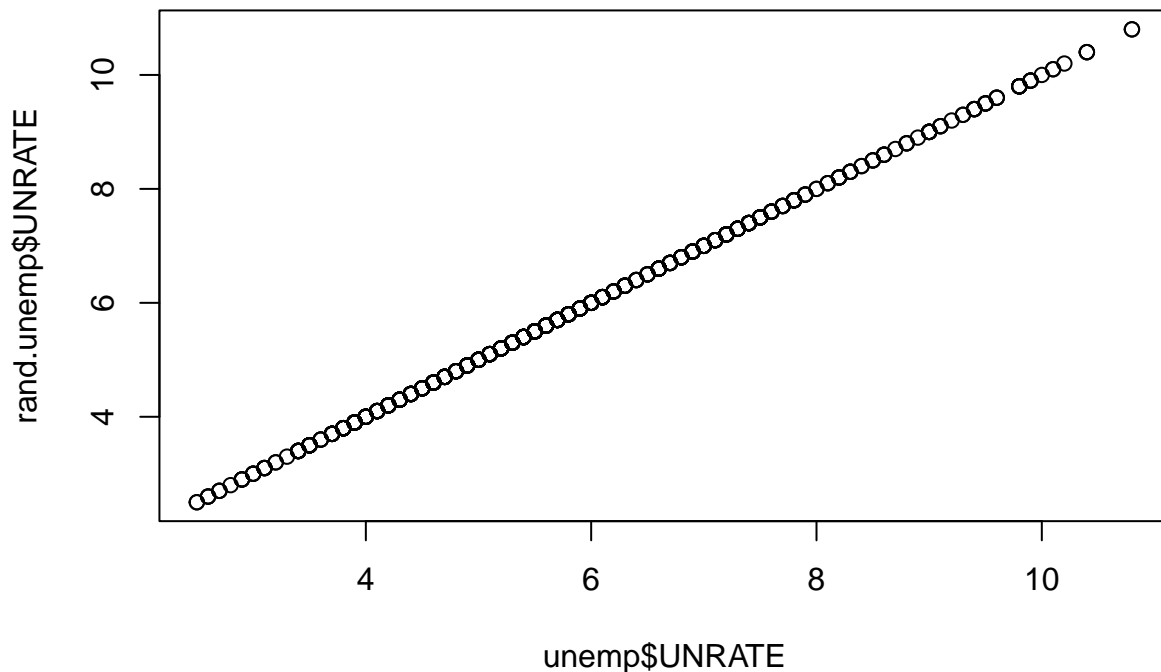
## NULL

```
rand.unemp[350:400][rtp == TRUE,
  points(
    UNRATE, impute.ff,
    col = 2, pch = 6, cex = 2
  )]
```



```
## NULL
```

```
plot(
  unemp$UNRATE, rand.unemp$UNRATE
)
```



Pode imputar os valor de trás para frente , contudo, se quiser treinar um modelo isso trata de um lookahead.

Em alguns forward fill é melhor maneira de preencher valores ausentes, mesmo que métodos “mais sofisticados” sejam possíveis.

- Não é exigente em termos computacionais, pode ser facilmente aplicados a dados em tempo real.

**Média Móvel** Semelhante ao forward fill mas imputa dados com uma média model ou mediana. Voce usa entradas provenientes de multiplos tempos recentes no passado.

Situaçõess em que a média móvel melhor se adéqua á tarefa em questão do que um forward fill :

1. Se os dados forem ruidosos e voce tem razões para duvidar do valor de qualquer ponto de dados individual em relação a um média geral, recomenda-se usar uma média móvel.
  - Forward Fill pode incluir mais ruídos aleatório do que a métrica “verdadeira” que lhe interessa. Média movel pode remover parte desse ruído.

```
rand.unemp[,
  impute.rm.nolookahead := rollapply(
    c(NA, NA, UNRATE), 3,
    function(x){
      if (!is.na(x[3])) x[3] else mean(x, na.rm = TRUE)
    }
  )]
```

```

bias.unemp[,
  impute.rm.nolookahead := rollapply(
    c(NA, NA, UNRATE), 3,
    function(x){
      if (!is.na(x[3])) x[3] else mean(x, na.rm = TRUE)
    }
  )]

```

*# plot de um gráfico de amostra que mostra as partes achatadas*

```

unemp[
  350:400,
  plot(UNRATE, col = 1, lwd = 2, type = 'b')
]

```

## NULL

```

rand.unemp[
  350:400,
  lines(
    DATE, impute.rm.nolookahead, col = 2, lwd = 2, lty = 2
  )
]

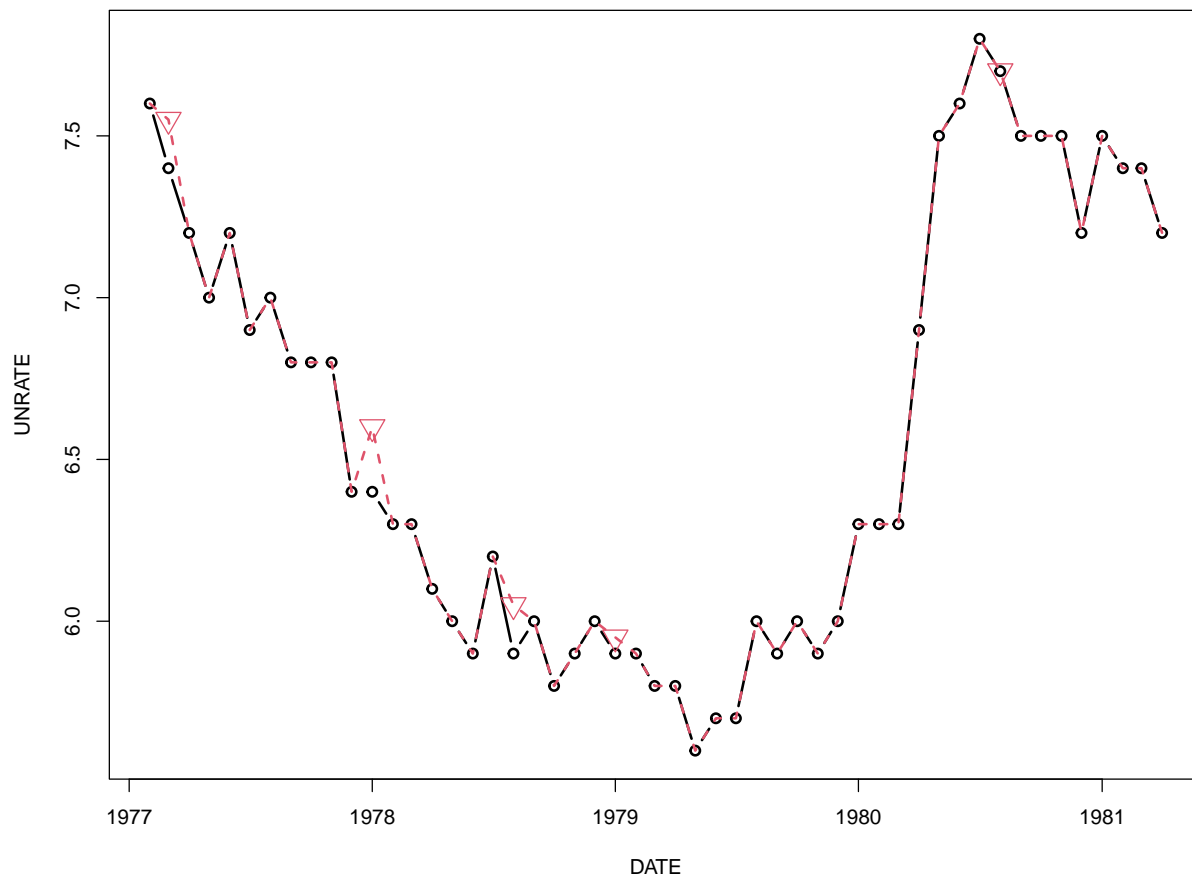
```

## NULL

```

rand.unemp[350:400][rtp == TRUE,
  points(
    DATE, impute.rm.nolookahead,
    col = 2, pch = 6, cex = 2
  )]

```



```
## NULL
```

Exemplo real do rollapply:

```
a = rollapply( c(NA, NA, c(1, 2, NA, NA, 1, 2, 2, NA, 10) ), 3,
              function(x){
                if (!is.na(x[3])) x[3] else mean(x, na.rm = TRUE) }
              )
a
```

```
## [1] 1.0 2.0 1.5 2.0 1.0 2.0 2.0 2.0 10.0
```

Retornando...

Caso não esteja preocupado em fornecer seus dados a um modelo e se sente a vontade de construir um lookahead. Pode usar a media movel incluindo dados do passado e futuro.

```
rand.unemp[,
  complete.rm := rollapply(
    c(NA, UNRATE, NA), 3,
    function(x){
```

```

        if (!is.na(x[2])) x[2]
        else mean(x, na.rm = TRUE)
    }
)
]

```

*# plot de um gráfico de amostra que mostra as partes achatadas*

```

unemp[
  350:400,
  plot(DATE, UNRATE, col = 1, lwd = 2, type = 'b')
]

```

## NULL

```

rand.unemp[
  350:400,
  lines(
    DATE, complete.rm, col = 2, lwd = 2, lty = 2
  )
]

```

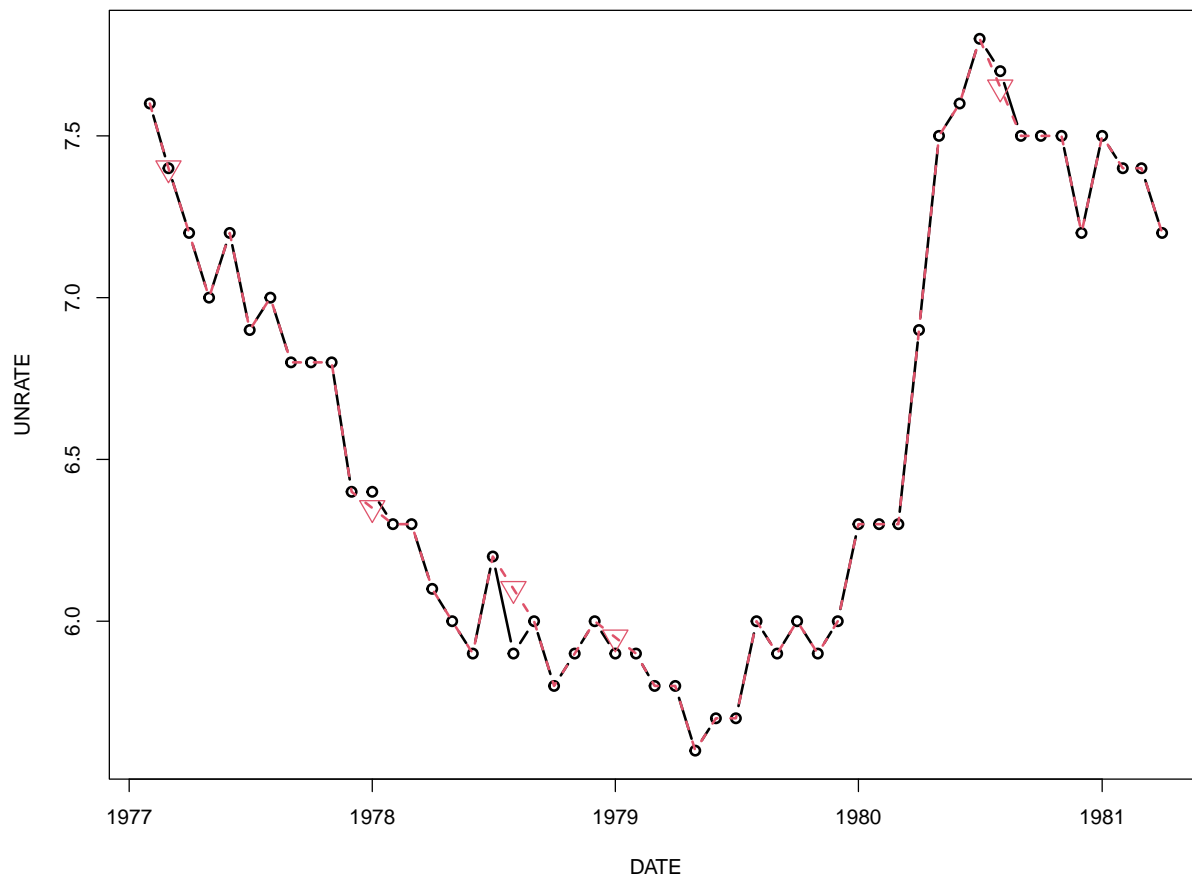
## NULL

```

rand.unemp[350:400][rtp == TRUE,
  points(
    DATE, complete.rm,
    col = 2, pch = 6, cex = 2
  )]

```





```
## NULL
```

Usar a média movel, **reduz a variancia** no conj. de dados.

**Interpolação** Usa para determinar dados de valores ausentes com base em restrições geometricas sobre como queremos que os dados gerais se comportem.

Pode ser feito semelhante a média movel se quiser imputar valores com base em valores passados e futuros(Isso cria um lookahead)

```
rand.unemp[, impute.li.lookahead := NULL]
```

```
## Warning in '[.data.table'(rand.unemp, , ':='(impute.li.lookahead, NULL))':
## Tentado atribuir NULL a coluna 'impute.li.lookahead', mas essa coluna não
## existe para remover
```

```
rand.unemp[, impute.li := NULL]
```

```
## Warning in '[.data.table'(rand.unemp, , ':='(impute.li, NULL))': Tentado
## atribuir NULL a coluna 'impute.li', mas essa coluna não existe para remover
```

```
#interpolacao linear com lookahead
rand.unemp[,impute.li.lookahead := na.approx(UNRATE, rule=2)]
bias.unemp[,impute.li.lookahead := na.approx(UNRATE, rule=2)]
```

```
#interpolacao polinomial com lookahead
rand.unemp[,impute.sp.lookahead := na.spline(UNRATE)]
bias.unemp[,impute.sp.lookahead := na.spline(UNRATE)]
```

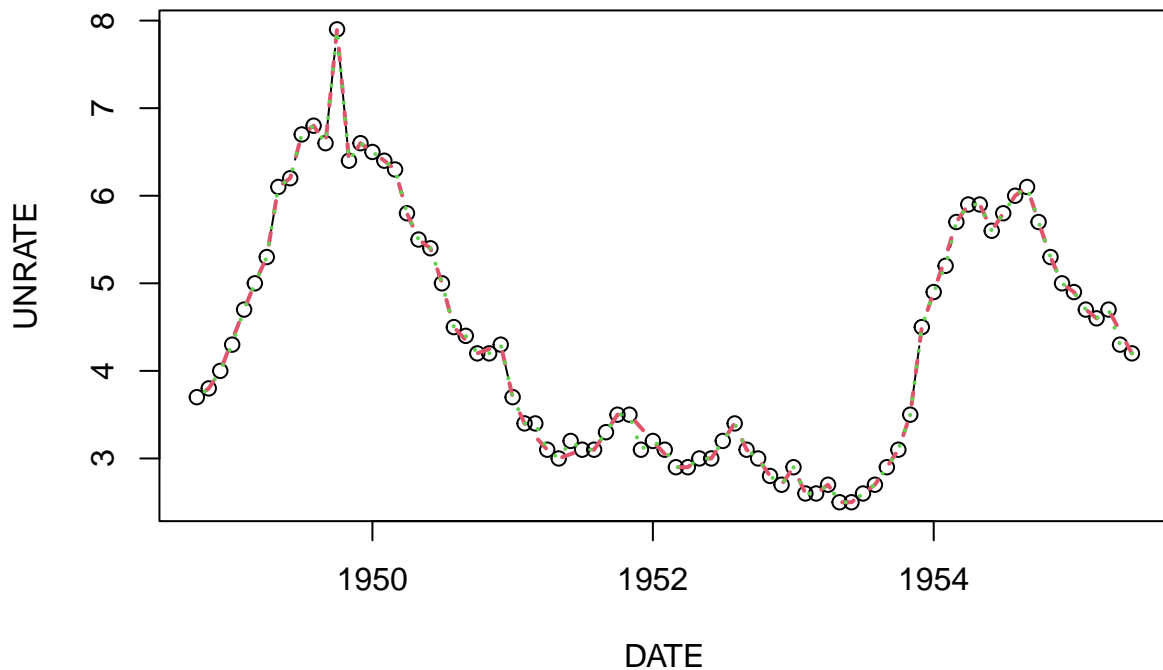
```
use.idx = 90:10
unemp[
  use.idx, plot(UNRATE, col = 1, type = 'b')
]
```

```
## NULL
```

```
rand.unemp[
  use.idx, lines(UNRATE, col = 2, lwd = 2, lty = 2)
]
```

```
## NULL
```

```
bias.unemp[
  use.idx, lines(UNRATE, col = 3, lwd = 2, lty = 3)
]
```



```
## NULL
```

```
# sem lookahead utilizar o rollapply
na_approx_no_lookahead <- function(x) {
  for (i in seq_along(x)) {
    if (is.na(x[i])) {
      # Interpolação apenas até o índice atual
      x[1:i] <- na.approx(x[1:i], na.rm = FALSE)
    }
  }
  return(x)
}
```

Utilizar quando:

1. Há tendências

Quando não utilizar

1. Quando não tem tendências

```
rand.unemp[,
  lapply(.SD,
    function(x){
      return(mean((x - unemp$UNRATE)**2, na.rm = TRUE))
    }),
  .SDcols = c("impute.ff", "impute.rm.nolookahead", "complete.rm", "impute.li.lookahead", "impute.sp.lookahead")
]
```

## Comparação

```
##      impute.ff impute.rm.nolookahead complete.rm impute.li.lookahead
##      <num>          <num>          <num>          <num>
## 1: 0.003613744      0.004869514    0.00133452      0.001324381
##      impute.sp.lookahead
##      <num>
## 1:      0.00239141
```

## Suavização de dados