CONTENIDO

0.1. Gráficas

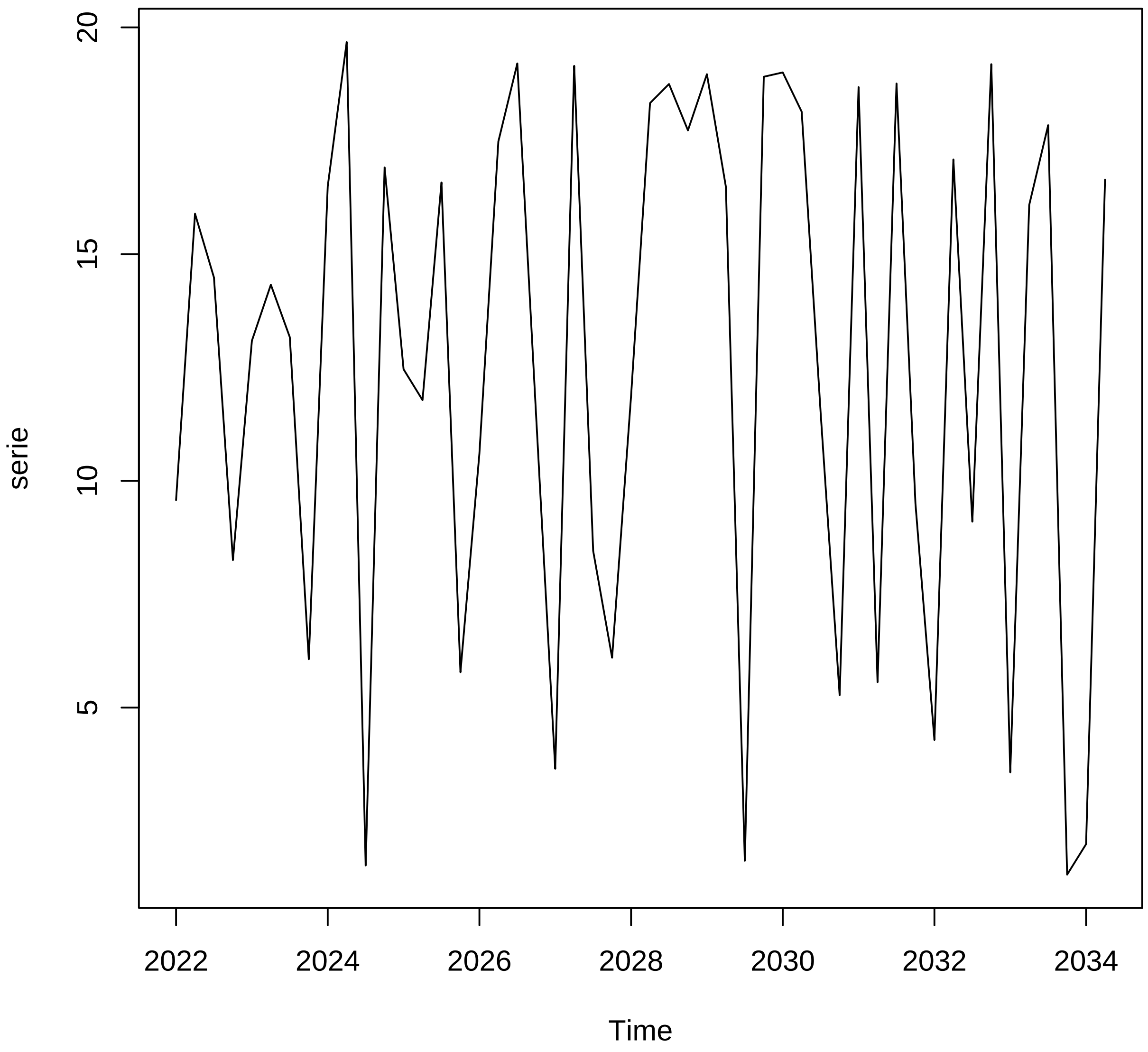
```
set.seed(2021) #Semilla (2021) hace que ya no sea tan aleatorio
datos <- runif(50,1,20) #Me da 50 no. aleatorios del 1 al 20
serie <- ts(datos,start = 2022,frequency = 4) #Trimestres, indico el año donde empiezo
serie
```

```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2022  9.574080 15.891816 14.483962  8.253141
## 2023 13.090152 14.325574 13.168339  6.066915
## 2024 16.493009 19.676750  1.518074 16.912318
## 2025 12.461574 11.781614 16.581003  5.779854
## 2026 10.604387 17.483224 19.205450 11.368256
## 2027  3.652034 19.151447  8.457382  6.101219
## 2028 11.871930 18.330731 18.751549 17.729236
## 2029 18.968201 16.484798  1.622743 18.911478
## 2030 19.007030 18.139734 11.493058  5.272838
## 2031 18.683548  5.561244 18.763111  9.499343
## 2032  4.286675 17.086492  9.100572 19.188800
## 2033  3.573003 16.083291 17.843472  1.317005
## 2034  1.989669 16.643254
```

```
unclass(serie) #Indica desde donde empieza, hasta donde termina y la frecuencia
```

```
##  [1]  9.574080 15.891816 14.483962  8.253141 13.090152 14.325574 13.168339
##  [8]  6.066915 16.493009 19.676750  1.518074 16.912318 12.461574 11.781614
## [15] 16.581003  5.779854 10.604387 17.483224 19.205450 11.368256  3.652034
## [22] 19.151447  8.457382  6.101219 11.871930 18.330731 18.751549 17.729236
## [29] 18.968201 16.484798  1.622743 18.911478 19.007030 18.139734 11.493058
## [36]  5.272838 18.683548  5.561244 18.763111  9.499343  4.286675 17.086492
## [43]  9.100572 19.188800  3.573003 16.083291 17.843472  1.317005  1.989669
## [50] 16.643254
## attr(,"tsp")
## [1] 2022.00 2034.25    4.00
```

```
plot(serie) #La gráfico
```



```
time(serie) #tiempos
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 2022 2022.00 2022.25 2022.50 2022.75
## 2023 2023.00 2023.25 2023.50 2023.75
## 2024 2024.00 2024.25 2024.50 2024.75
## 2025 2025.00 2025.25 2025.50 2025.75
## 2026 2026.00 2026.25 2026.50 2026.75
## 2027 2027.00 2027.25 2027.50 2027.75
## 2028 2028.00 2028.25 2028.50 2028.75
## 2029 2029.00 2029.25 2029.50 2029.75
## 2030 2030.00 2030.25 2030.50 2030.75
## 2031 2031.00 2031.25 2031.50 2031.75
## 2032 2032.00 2032.25 2032.50 2032.75
## 2033 2033.00 2033.25 2033.50 2033.75
## 2034 2034.00 2034.25
```

```
serie2 <- ts(datos,start = c(2021,2),frequency = 4) #Indico el mes y año donde empiezo
time(serie2)
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 2021         2021.25 2021.50 2021.75
## 2022 2022.00 2022.25 2022.50 2022.75
## 2023 2023.00 2023.25 2023.50 2023.75
## 2024 2024.00 2024.25 2024.50 2024.75
## 2025 2025.00 2025.25 2025.50 2025.75
## 2026 2026.00 2026.25 2026.50 2026.75
## 2027 2027.00 2027.25 2027.50 2027.75
## 2028 2028.00 2028.25 2028.50 2028.75
## 2029 2029.00 2029.25 2029.50 2029.75
## 2030 2030.00 2030.25 2030.50 2030.75
## 2031 2031.00 2031.25 2031.50 2031.75
## 2032 2032.00 2032.25 2032.50 2032.75
## 2033 2033.00 2033.25 2033.50
```

```
library(readr)
setwd("C:\\Users\\81799\\Downloads\\Pronosticos_y_series_de_tiempo\\data")
mis_datos <- read.csv("Rmissing.csv") #Leyendo mi csv
serie3 <- ts(mis_datos$mydata, start = 2022, frequency = 12 )
serie3
```

```
##             Jan        Feb        Mar        Apr        May        Jun
## 2022  32.801464  42.465485         NA  32.204058  55.557647  33.050864
## 2023  36.544163  26.668135  41.325626  28.913199  38.595417  31.341447
## 2024  19.439525  41.892407  30.321857  32.899878  17.686235  10.332791
## 2025  23.408770  42.785841  31.919674  37.571226  33.907485  17.698917
## 2026  33.961104  40.826518  34.389579  27.210322  41.815827         NA
## 2027  33.941547  28.582326  17.945402  40.335543  32.103075  15.609346
## 2028  23.682860  49.014566  59.160903  24.994359  37.321672  11.830421
## 2029  45.001326  36.082827  38.969588  24.260726   8.619401  33.933167
## 2030  48.560701 999.000000  31.011612  33.565184  41.850476  45.780926
## 2031  47.089342  33.955275  36.563838  18.773382  28.077605  40.483324
## 2032  44.662468  43.125045  25.800244  22.833920  51.397357  34.775922
## 2033  20.619643  24.895206   4.682543  17.049461  45.618543  28.288209
## 2034  16.457915  36.157602  35.773314  23.368300  34.220736  39.443674
## 2035  36.949438  17.647508  22.991044  40.388899  30.654440  49.261182
## 2036  41.690112  46.226476  40.741721 999.000000  26.455048  12.766929
## 2037  53.072688  51.889866  43.700888  33.639492  24.988901  27.019376
## 2038  45.478245  34.152629  50.573869  43.564419  57.644084  20.785408
## 2039  33.952911  35.372668  38.059841  40.818440  45.768335  39.272128
## 2040  39.873144  51.126396   2.683472  51.667975  41.336229  43.090450
## 2041  33.557160  33.851597         NA  36.149460  32.985037  31.422766
## 2042  23.581530  36.968395  50.747174  37.981389  28.693203  32.396009
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2022  43.401620  37.768318  22.844180  36.428877  28.496485  59.037881
## 2023  34.547023         NA  30.499324  49.391323  43.976004  22.162741
## 2024  31.612958  40.011275  35.378517  46.167222  26.903207  36.304821
## 2025  19.931775  23.971169 999.000000  32.853670  33.012320  47.893249
## 2026  49.711080  37.246486  34.472507  27.554913  37.976930  24.503481
## 2027  38.637130  58.877558  42.178769  34.075469  29.208206  20.409934
## 2028  49.907975  33.288427  25.900307  34.661099  38.170951  30.246685
## 2029  30.158056  32.211135  46.688584  36.399098  27.266510  39.706101
## 2030  21.679404  32.340497  55.904896  17.349895  32.994516  36.155426
## 2031  41.341771  32.907839  59.604911  20.989279  46.886734  53.931163
## 2032  50.922532  36.430258  32.975690  37.659017  48.006323  49.901919
## 2033  50.446258  34.983971  38.847283  32.301493  46.044574  21.739473
## 2034  26.074044  28.599269  45.410516         NA  30.585004  23.405284
## 2035  31.215505  30.462442  41.294423  28.046393  24.925970  23.934094
## 2036  38.133315  61.653241  11.474054  41.835335  34.572898  59.921615
## 2037  12.774882  21.001551  18.133113  48.563807  64.633075  29.362668
## 2038  39.811707  51.854335  33.351763  23.242107  34.351879  28.113792
## 2039 999.000000  36.665537  50.282893  34.561040  46.040830  39.936308
## 2040  24.686842  52.300908  37.379943  19.043254  43.512121  54.236360
## 2041  36.574851  29.648483  18.290954  38.154075  34.446452  12.037743
## 2042  41.325484  30.017571  14.818111  45.403854
```

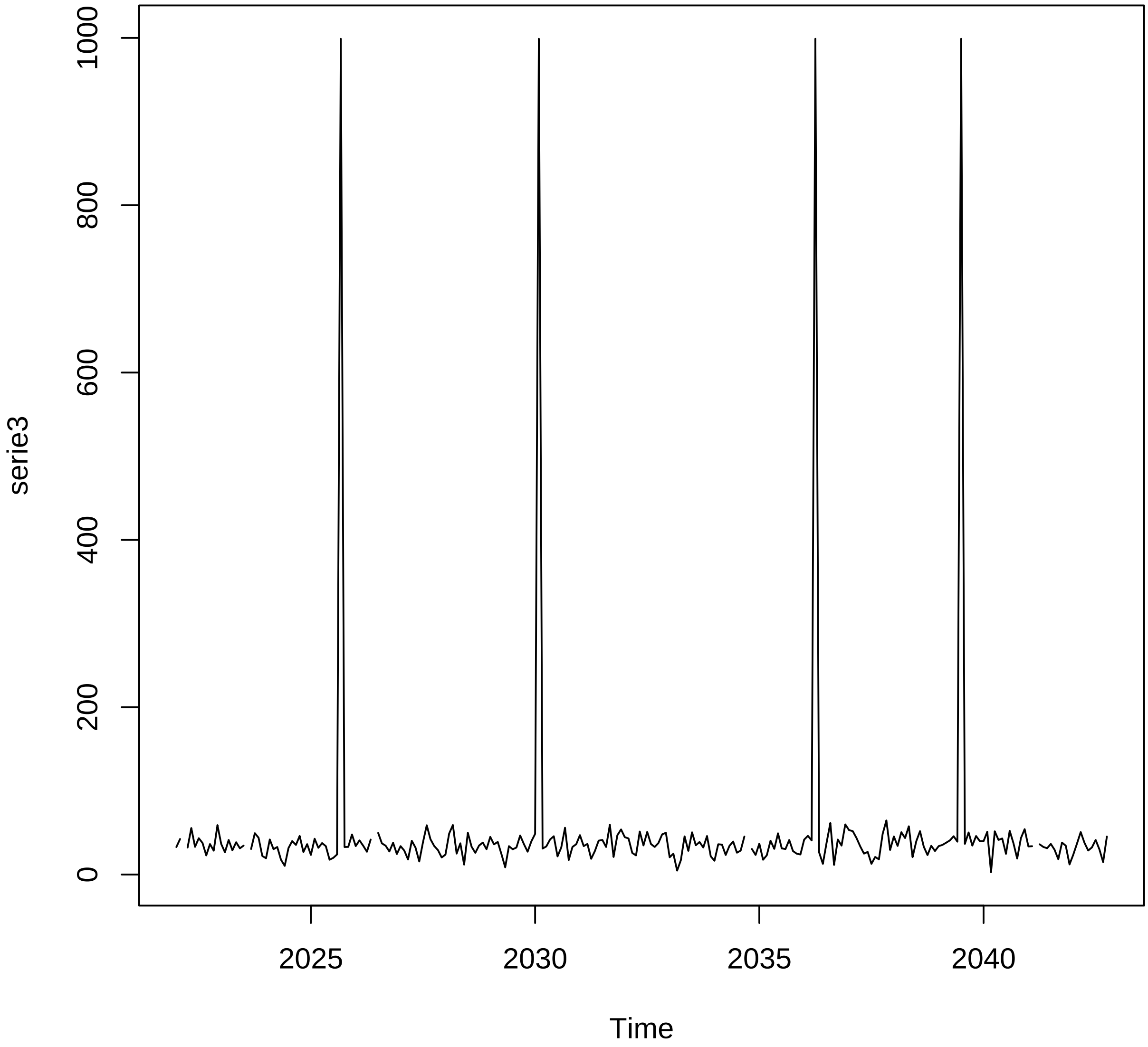
```
class(serie3)
```

```
## [1] "ts"
```

```
summary(serie3) #Nos da la información de serie3
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   2.683  28.078  34.573  50.710  42.465 999.000       5
```

```
plot(serie3)
```



0.2. Tratamiento de valores faltantes

ZOO

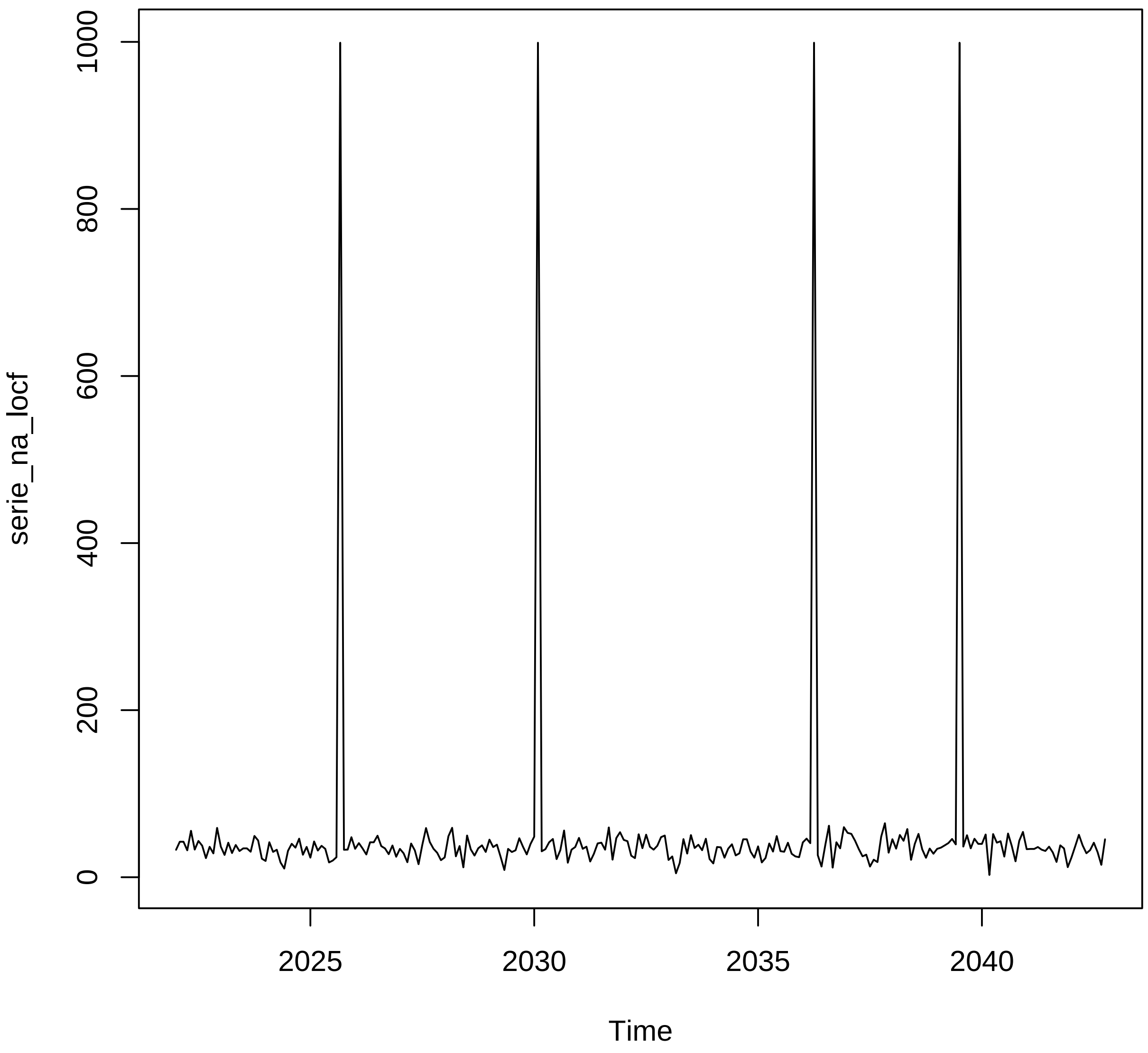
```
library(lubridate)

##
## Attaching package: 'lubridate
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tseries)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(zoo)

##
## Attaching package: ’zoo
## The following objects are masked from ’package:base’:
##
##     as.Date, as.Date.numeric

library(ggplot2)
library(forecast)
```

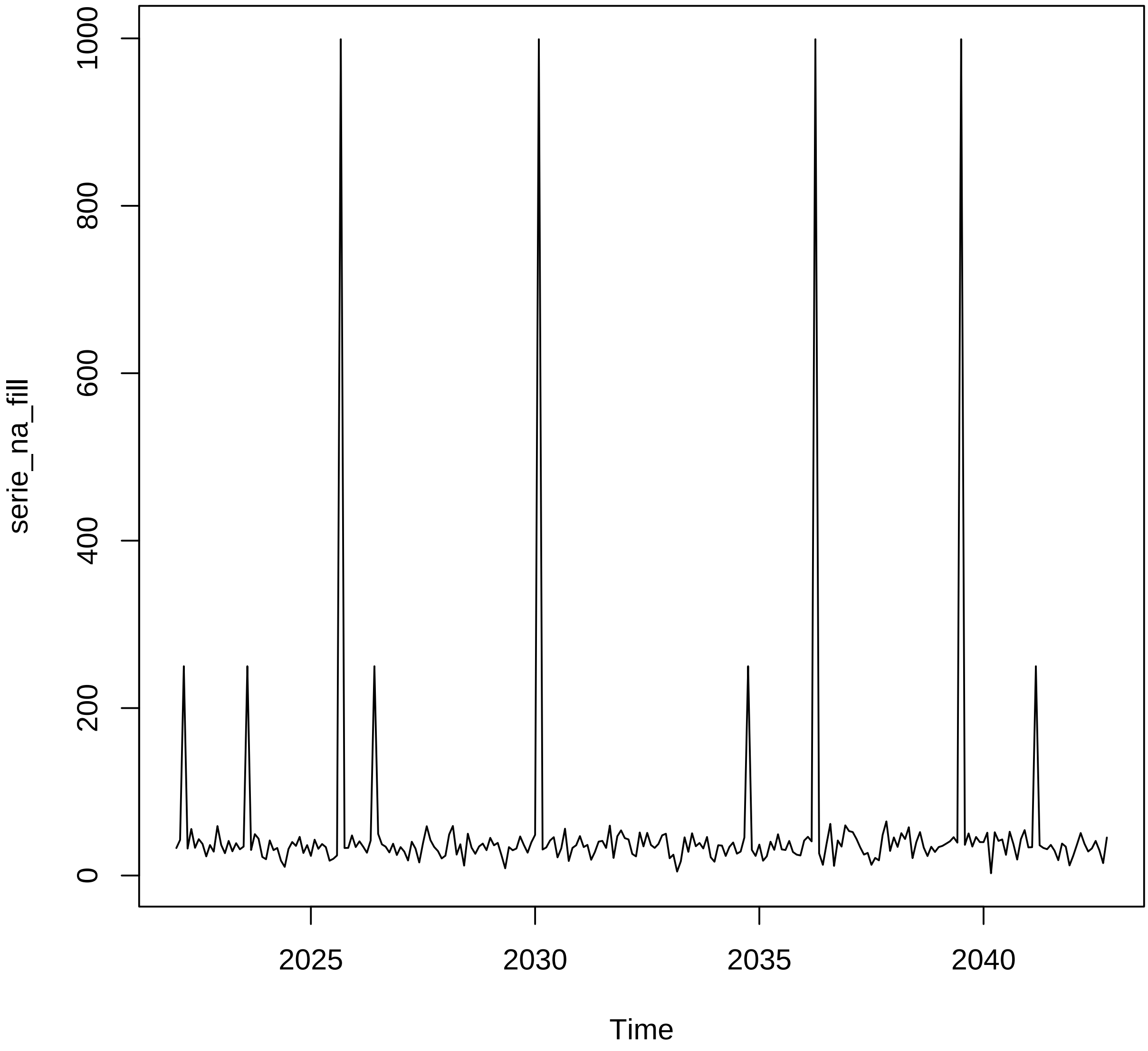
NA con el último valor

```
serie_na_locf <- na.locf(serie3) #rellena los NA con el último valor previo
plot(serie_na_locf)
```



0.2.1. NA con valor dado

```
serie_na_fill <- na.fill(serie3,250) #rellena NA con el valor 250
plot(serie_na_fill)
```

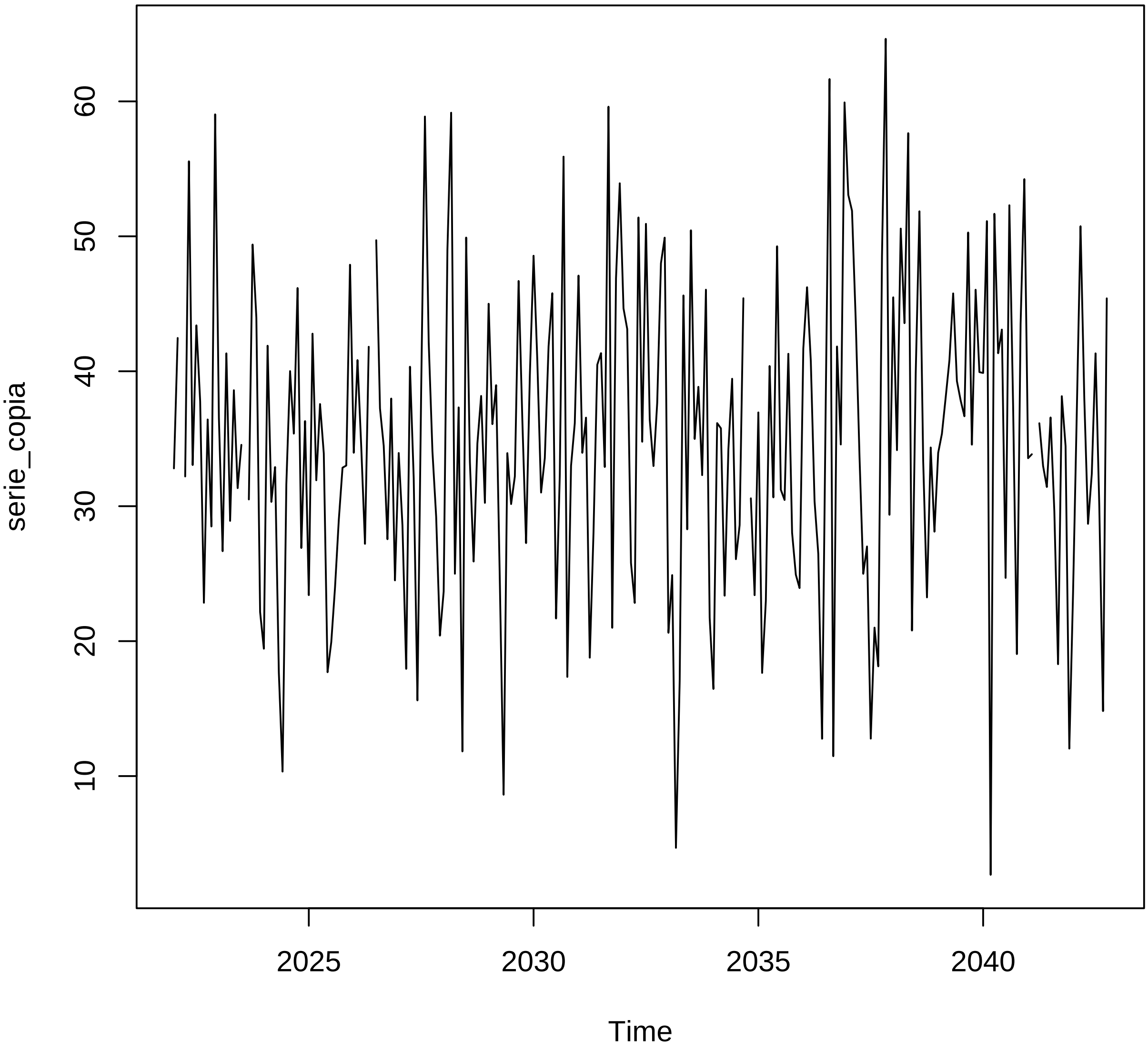


0.2.2. Tratamiento de valores atípicps con FORECAST

FORECAST corrige los valores atípicos.

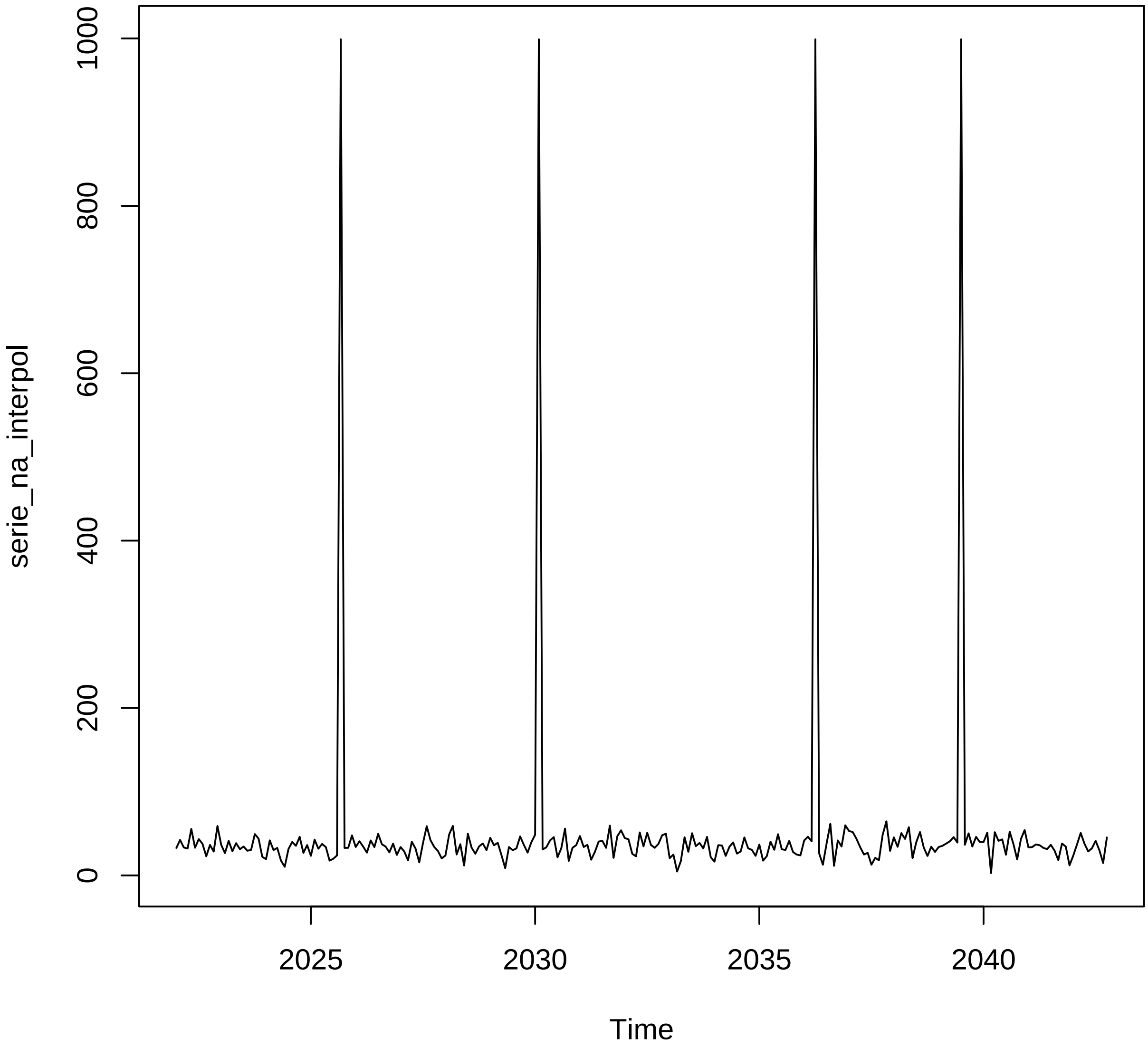
```
tsoutliers(serie3) #Me da el índice de los NA y me da sugerencia del valor en esos lugares

## $index
## [1]  45  98 172 211
##
## $replacements
## [1] 28.96650 40.76541 30.37705 37.83168
```

```
serie_copia <- serie3 #Hago una copia de serie3
# Me dice que tome los index de los atípicos y los cambie por las sugerencias
serie_copia[tsoutliers(serie3)$index] <- tsoutliers(serie3)$replacements
plot(serie_copia)
```



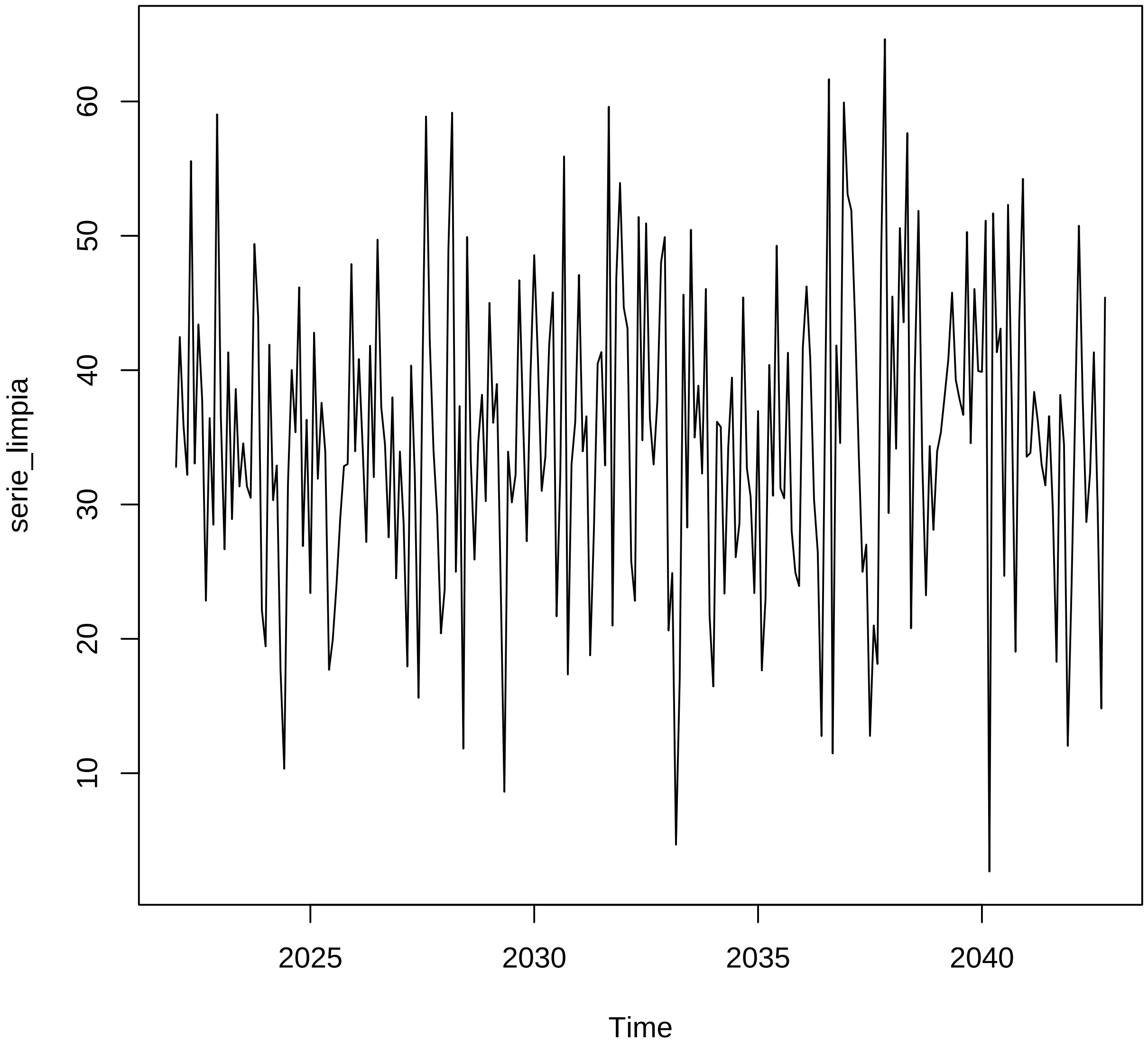
0.2.3. INTERPOLACIÓN

```
serie_na_interpol <- na.interp(serie3)  # rellenar NA con interpolaciones
plot(serie_na_interpol)
```



0.3. Valores atípicos y faltantes con tsclean()

```
serie_limpia <- tsclean(serie3)

plot(serie_limpia)
```


serie_limpia
10
20
30
40
50
60
2025
2030
2035
2040
Time