

Series Temporales

Tomas Lemus

28/11/2020

Proyecto series temporales

Primera Parte:

Analisis de serie temporal de fabricación de un producto electrónico.

Importar el dataset Datos.xlsx

```
df = read_excel("Datos.xlsx")
head(df)
```

```
## # A tibble: 1 x 192
##   '2002M01' '2002M02' '2002M03' '2002M04' '2002M05' '2002M06' '2002M07'
##   <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
## 1      124.      121.      142.      128.      137.      160.      109.
## # ... with 185 more variables: '2002M08' <dbl>, '2002M09' <dbl>,
## #   '2002M10' <dbl>, '2002M11' <dbl>, '2002M12' <dbl>, '2003M01' <dbl>,
## #   '2003M02' <dbl>, '2003M03' <dbl>, '2003M04' <dbl>, '2003M05' <dbl>,
## #   '2003M06' <dbl>, '2003M07' <dbl>, '2003M08' <dbl>, '2003M09' <dbl>,
## #   '2003M10' <dbl>, '2003M11' <dbl>, '2003M12' <dbl>, '2004M01' <dbl>,
## #   '2004M02' <dbl>, '2004M03' <dbl>, '2004M04' <dbl>, '2004M05' <dbl>,
## #   '2004M06' <dbl>, '2004M07' <dbl>, '2004M08' <dbl>, '2004M09' <dbl>,
## #   '2004M10' <dbl>, '2004M11' <dbl>, '2004M12' <dbl>, '2005M01' <dbl>,
## #   '2005M02' <dbl>, '2005M03' <dbl>, '2005M04' <dbl>, '2005M05' <dbl>,
## #   '2005M06' <dbl>, '2005M07' <dbl>, '2005M08' <dbl>, '2005M09' <dbl>,
## #   '2005M10' <dbl>, '2005M11' <dbl>, '2005M12' <dbl>, '2006M01' <dbl>,
## #   '2006M02' <dbl>, '2006M03' <dbl>, '2006M04' <dbl>, '2006M05' <dbl>,
## #   '2006M06' <dbl>, '2006M07' <dbl>, '2006M08' <dbl>, '2006M09' <dbl>,
## #   '2006M10' <dbl>, '2006M11' <dbl>, '2006M12' <dbl>, '2007M01' <dbl>,
## #   '2007M02' <dbl>, '2007M03' <dbl>, '2007M04' <dbl>, '2007M05' <dbl>,
## #   '2007M06' <dbl>, '2007M07' <dbl>, '2007M08' <dbl>, '2007M09' <dbl>,
## #   '2007M10' <dbl>, '2007M11' <dbl>, '2007M12' <dbl>, '2008M01' <dbl>,
## #   '2008M02' <dbl>, '2008M03' <dbl>, '2008M04' <dbl>, '2008M05' <dbl>,
## #   '2008M06' <dbl>, '2008M07' <dbl>, '2008M08' <dbl>, '2008M09' <dbl>,
## #   '2008M10' <dbl>, '2008M11' <dbl>, '2008M12' <dbl>, '2009M01' <dbl>,
## #   '2009M02' <dbl>, '2009M03' <dbl>, '2009M04' <dbl>, '2009M05' <dbl>,
## #   '2009M06' <dbl>, '2009M07' <dbl>, '2009M08' <dbl>, '2009M09' <dbl>,
## #   '2009M10' <dbl>, '2009M11' <dbl>, '2009M12' <dbl>, '2010M01' <dbl>,
## #   '2010M02' <dbl>, '2010M03' <dbl>, '2010M04' <dbl>, '2010M05' <dbl>,
## #   '2010M06' <dbl>, '2010M07' <dbl>, '2010M08' <dbl>, '2010M09' <dbl>,
## #   '2010M10' <dbl>, '2010M11' <dbl>, ...
```

```
dato= df$"2002M02"
class(dato)
```

```
## [1] "numeric"
```

```
data <- as.data.frame(t(df))
ts.n1 <- ts(data, frequency = 12, start = c(2002,1))
print(class(ts.n1))
```

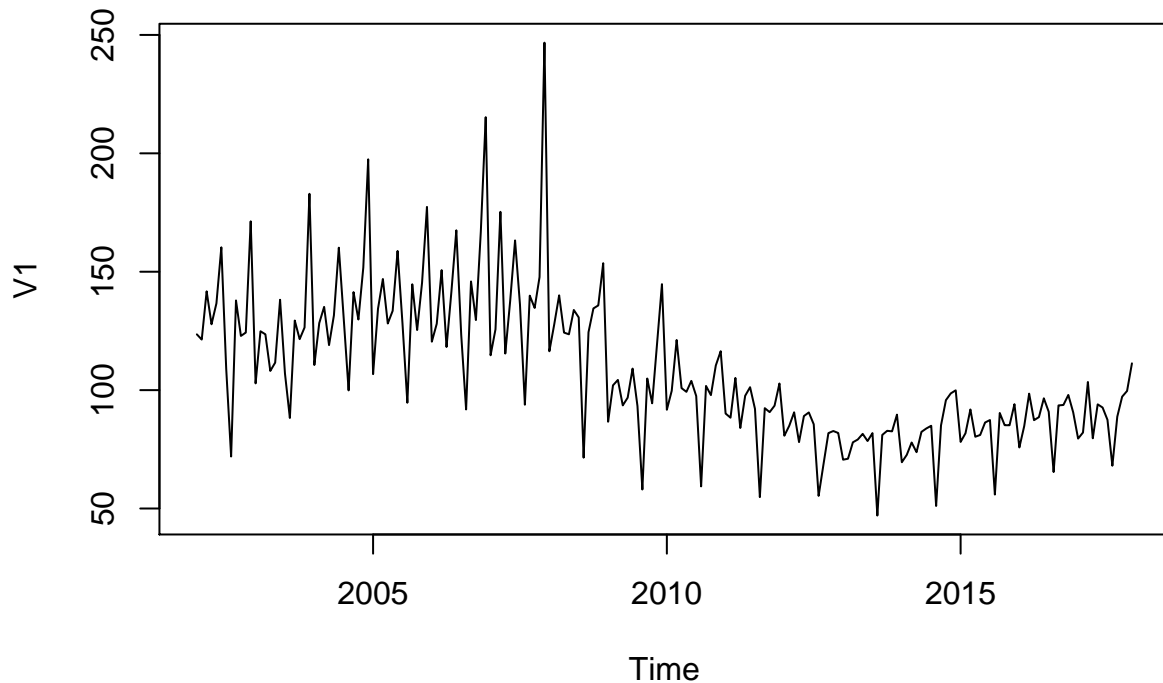
```
## [1] "ts"
```

```
print(ts.n1)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep
## 2002 123.552 121.354 141.716 127.818 136.644 160.312 109.010  71.952 137.886
## 2003 102.850 124.859 123.567 108.103 111.609 138.168 107.047  88.228 129.389
## 2004 110.591 128.314 135.152 119.008 131.597 160.153 130.140  99.880 141.391
## 2005 106.753 134.370 146.936 128.142 133.487 158.779 128.200  94.653 144.709
## 2006 120.456 128.028 150.683 118.204 141.743 167.502 123.301  91.829 145.896
## 2007 114.717 125.687 175.305 115.439 137.688 163.237 136.139  93.829 139.900
## 2008 116.403 127.773 140.036 124.287 123.579 133.845 130.708  71.467 124.460
## 2009  86.645 102.028 104.309  93.552  96.810 109.099  93.278  58.023 104.908
## 2010  91.709  99.728 121.180 100.836  99.277 103.899  97.513  59.339 101.770
## 2011  90.112  88.356 105.158  84.001  97.607 101.240  92.072  54.812  92.386
## 2012  80.750  84.941  90.625  78.052  89.053  90.576  85.579  55.327  68.747
## 2013  70.623  71.025  77.959  79.140  81.526  78.496  81.840  47.025  81.076
## 2014  69.548  72.646  77.878  73.786  82.361  83.807  84.950  51.060  85.109
## 2015  78.131  81.789  91.833  80.310  81.053  86.310  87.376  55.885  90.346
## 2016  75.786  84.847  98.512  87.310  88.604  96.539  90.852  65.388  93.542
## 2017  79.539  82.088 103.447  79.663  93.998  92.653  87.459  68.072  88.796
##           Oct      Nov      Dec
## 2002 122.885 124.307 171.297
## 2003 121.531 126.449 182.917
## 2004 129.817 151.791 197.474
## 2005 125.372 145.349 177.376
## 2006 129.587 166.820 215.284
## 2007 134.705 147.767 246.734
## 2008 134.508 135.779 153.619
## 2009  94.429 119.649 144.760
## 2010  97.894 110.416 116.440
## 2011  90.702  93.429 102.798
## 2012  81.842  82.732  81.903
## 2013  82.821  82.582  89.651
## 2014  95.764  98.624  99.915
## 2015  85.190  85.162  94.057
## 2016  93.724  97.955  90.602
## 2017  97.236  99.637 111.320
```

Con la observación anterior es posible notar como los tipos de datos ya no son numericos, ahora son de serie temporal y se encuentran más estructurados (Años y meses). Procederemos a un analisis a partir de la visualización de la serie temporal.

```
plot.ts(ts.n1)
```



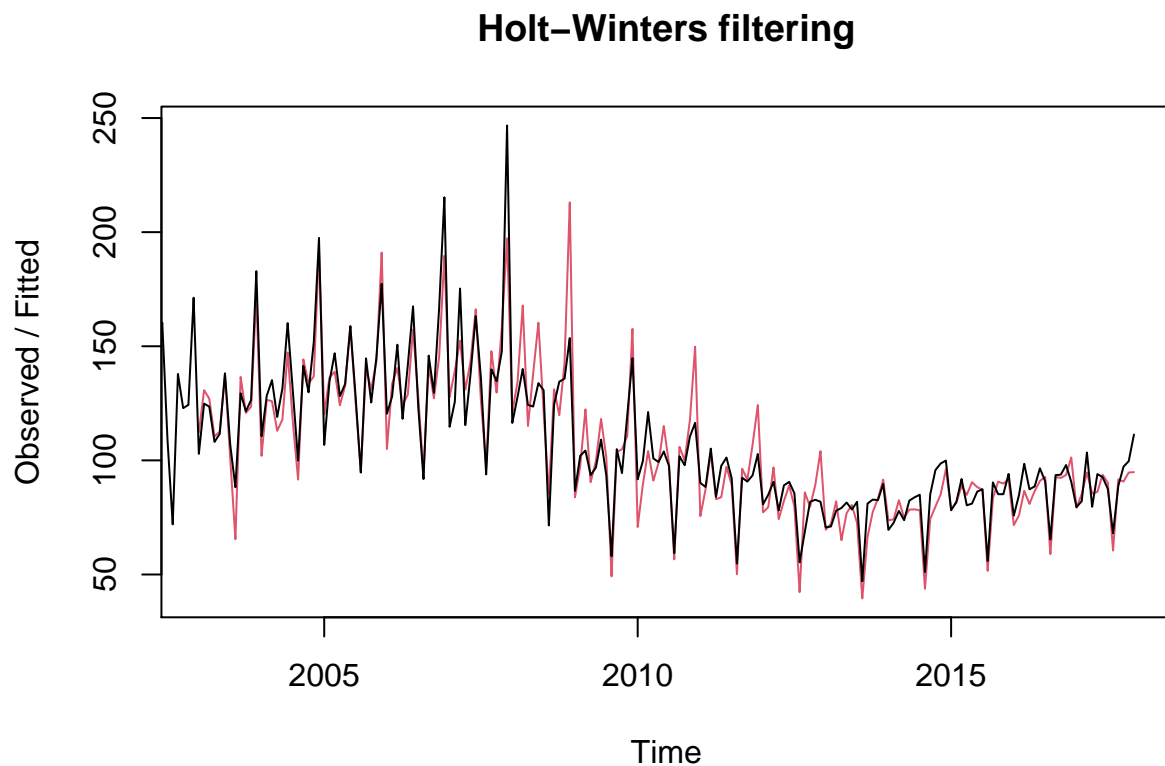
Suavizado: Holt Winters. Esta técnica de suavizado utiliza un conjunto de estimaciones recursivas a partir de la serie histórica. Estas estimaciones utilizan una constante de nivel, una constante de tendencia, y una constante estacional multiplicativa. Como veremos esta tecnica ofrece un ajuste estacional ($s=12$ en el caso de datos mensuales).

```
suavi = HoltWinters(ts.n1)
suavi
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = ts.n1)
##
## Smoothing parameters:
##  alpha: 0.2297538
##  beta : 0.01217016
##  gamma: 0.6635159
##
## Coefficients:
##           [,1]
## a      86.4531200
## b     -0.1801035
## s1    -1.5042434
## s2     2.9180549
```

```
## s3 19.4953243
## s4 1.1051776
## s5 10.3725281
## s6 11.8430185
## s7 7.9783810
## s8 -15.5678831
## s9 8.7700406
## s10 13.5436694
## s11 15.4616769
## s12 20.5999930
```

```
plot(suavi)
```



Analisis inicial: Inicialmente podemos observar algunos patrones que no deberiamos apresurarnos a determinar estacionalidad, ademas, la tendencia puede ser confusa porque incialmente parece ser creciente con unos picos notablemente altos, pero desde el 2010 los picos dejan de ser tan notables y la tendencia parece cambiar de dirección (decreciente), aunque finalmente, a partir del 2015 parece elevarse un poco.

Al describir series de tiempo, utilizamos palabras como “tendencia” y “estacionalidad”, que deben definirse con más cuidado. Basado en los estudios y materiales de clase podemos decir lo siguiente:

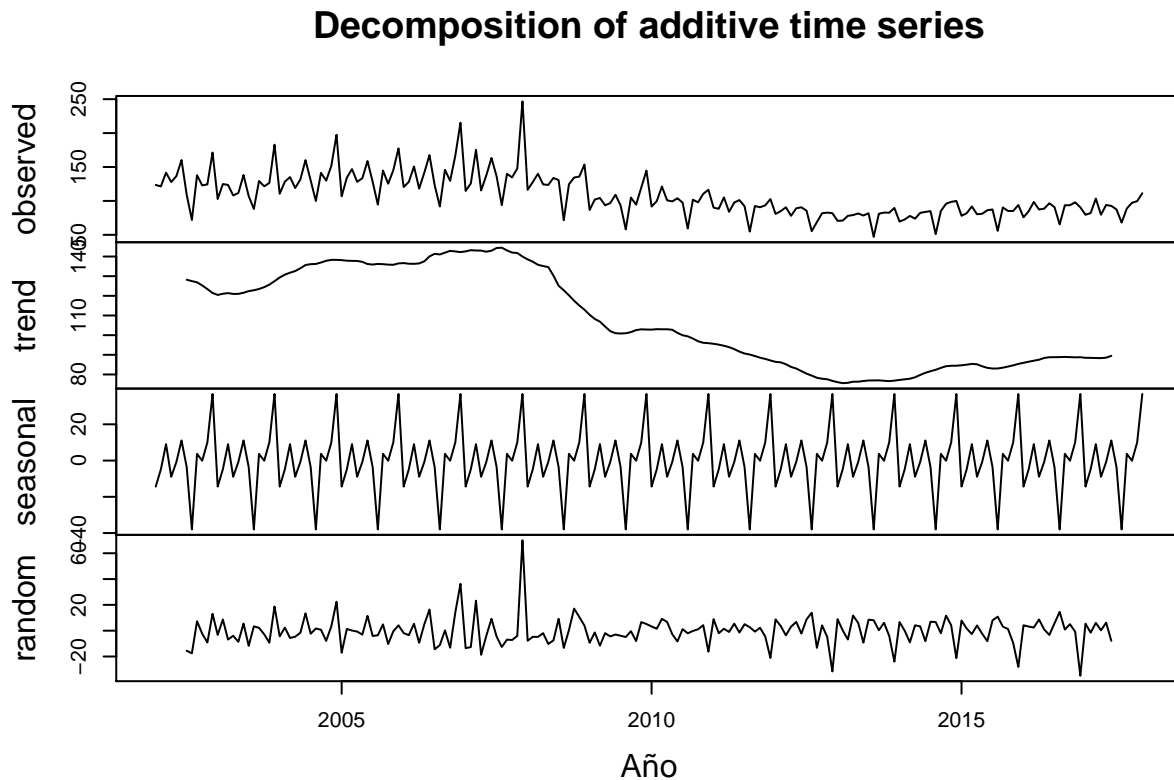
Tendencia Existe una tendencia cuando hay un aumento o disminución a largo plazo en los datos. No tiene por qué ser lineal. A veces nos referiremos a una tendencia como “cambio de dirección”, cuando puede pasar de una tendencia creciente a una tendencia decreciente.

Estacional Un patrón estacional ocurre cuando una serie de tiempo se ve afectada por factores estacionales como la época del año o el día de la semana. (frecuencia fija)

Cíclico Un ciclo se produce cuando la observación de los datos suben y bajan sin una frecuencia fija. Estas fluctuaciones suelen deberse a condiciones económicas o relacionadas con el “ciclo económico”. La duración de estas fluctuaciones suele ser al menos 2 años.

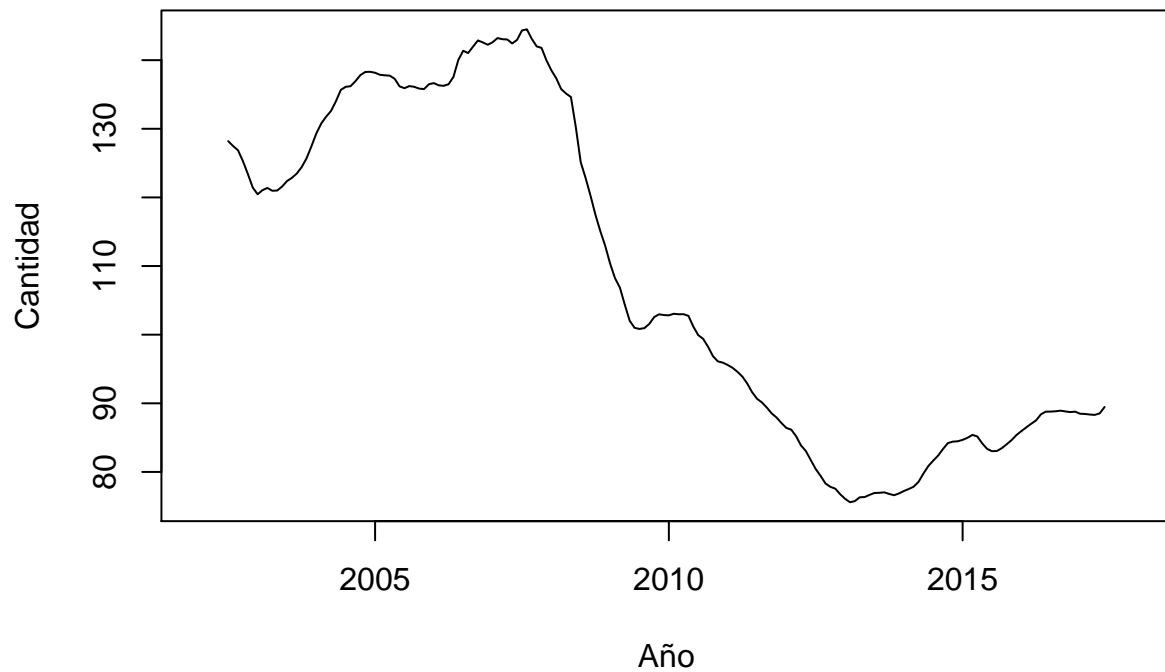
A partir de la descomposición de la serie temporal se puede obtener mayor claridad sobre la tendencia, estacional y el factor de aleatoriedad que puede presentar la fabricación del producto.

```
n1.ts.desc = decompose(ts.n1)
plot(n1.ts.desc, xlab='Año')
```



```
plot(n1.ts.desc$trend, xlab='Año', ylab='Cantidad') +
title("Tendencia para la serie temporal de fabricación")
```

Tendencia para la serie temporal de fabricación

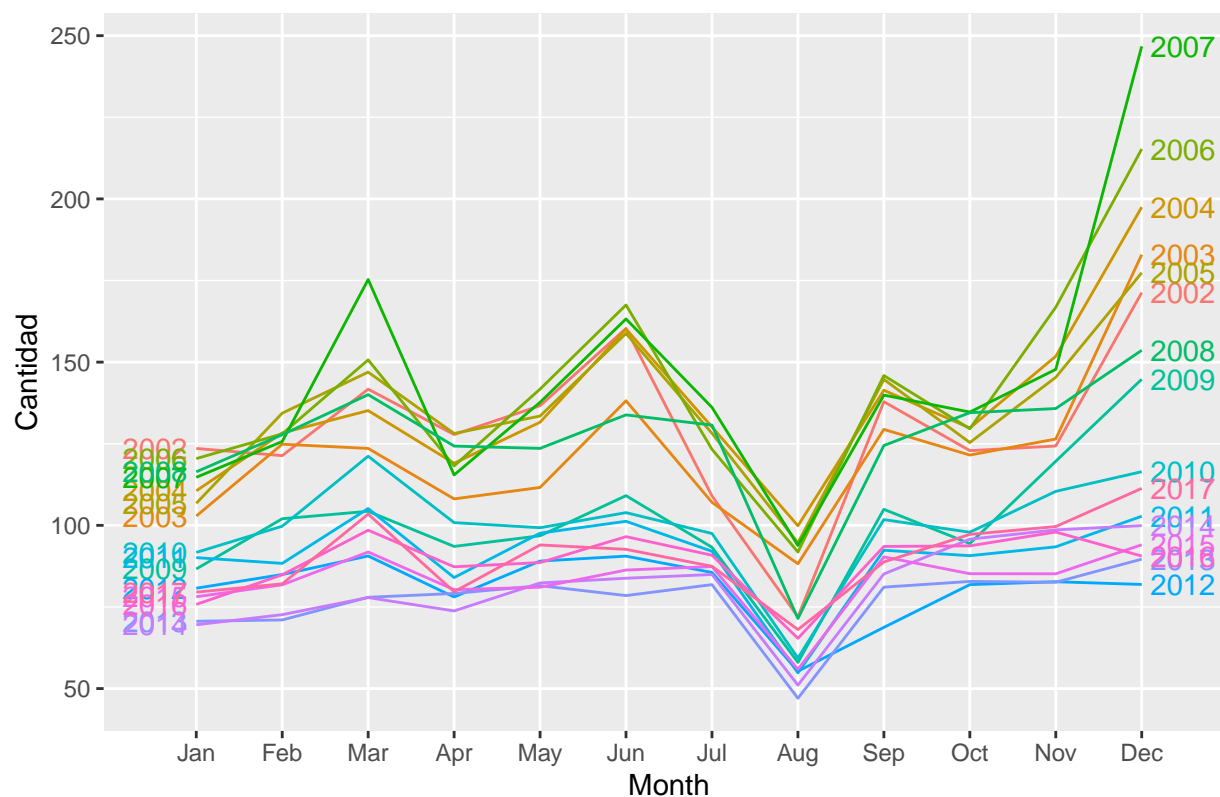


```
## integer(0)
```

A través del plot estacional se pueden observar estaciones de la serie por cada mes en los diferentes años, lo cual permite mejor visualización sobre el comportamiento en la fabricación.

```
ggseasonplot(ts.n1, year.labels=TRUE, year.labels.left=TRUE) +  
  ylab("Cantidad") +  
  ggtitle("Plot Estacional de fabricación de un producto electrónico")+  
  theme(plot.title = element_text(hjust = 0.5))
```

Plot Estacional de fabricación de un producto electrónico

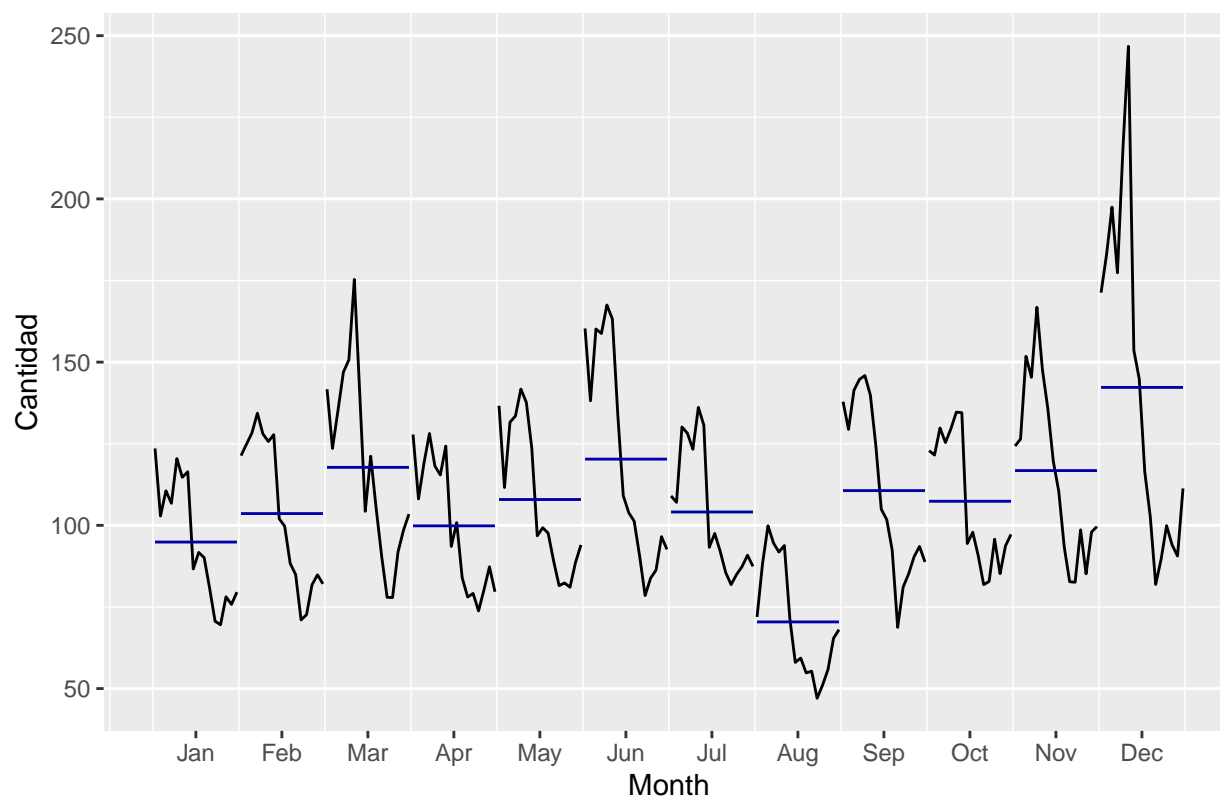


Subseries estacional:

El siguiente gráfico enfatiza los patrones estacionales es donde los datos de cada temporada se recopilan juntos en mini gráficos de tiempo separados, en los cuales podemos observar la cantidad media de fabricación del producto por cada mes.

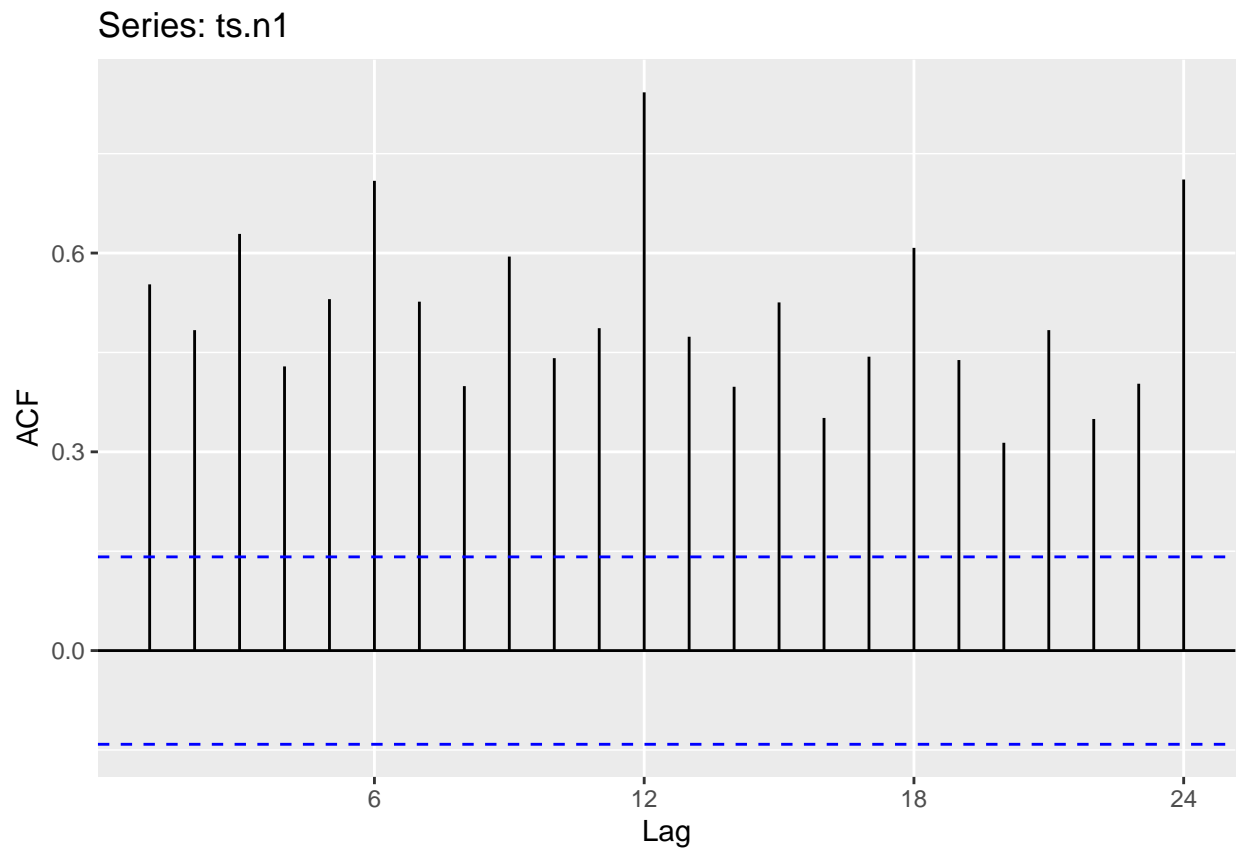
```
ggsubseriesplot(ts.n1) +
  ylab("Cantidad") +
  ggtitle("Subseries Estacional: Fabricación de un producto electrónico")+
  theme(plot.title = element_text(hjust = 0.5))
```

Subseries Estacional: Fabricación de un producto electrónico



Autocorrelograma El siguiente gráfico nos permite representar la función de autocorrelación a través de los picos, debido al patrón estacional de los datos. La línea azul indica una correlación significativamente diferente de cero.

```
ggAcf(ts.n1)
```

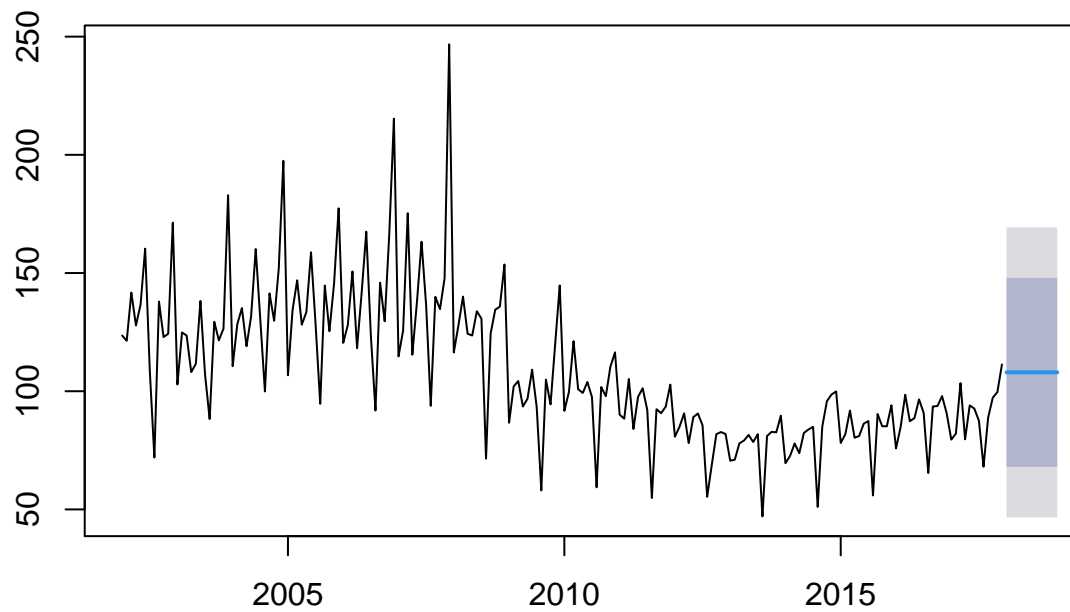



```
library(forecast)
```

Predicción para instantes futuros (un año)

```
mdata.ts.p1 <- meanf(ts.n1,12)  
plot(mdata.ts.p1)
```

Forecasts from Mean



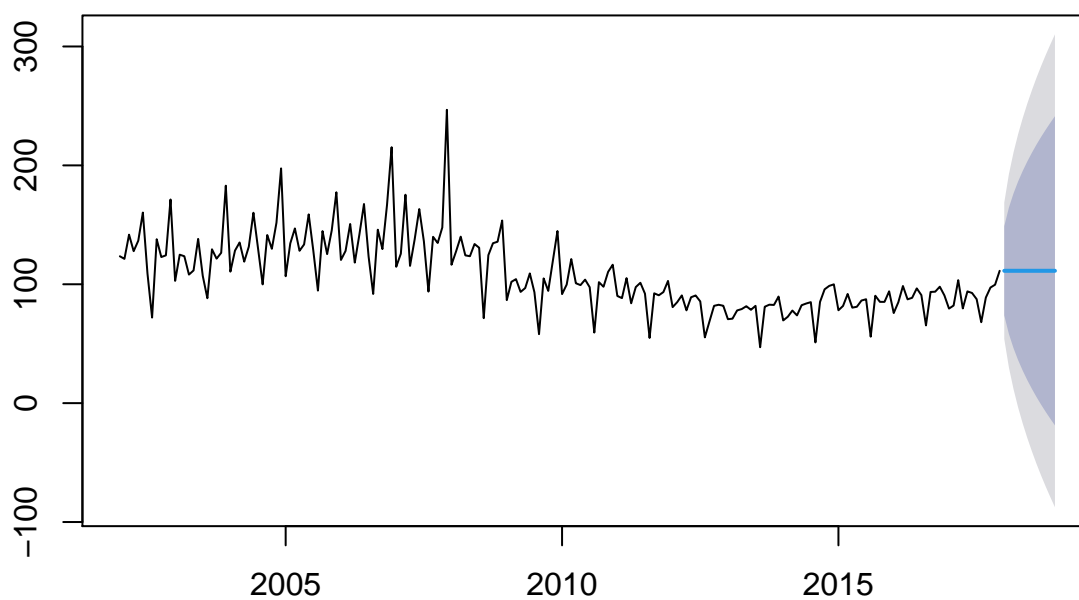
Método de la media

Metodo Naive.

Este método sencilla nos muestra un instante futuro que utiliza como predictor el último instante. El intervalo de color morado nos muestra la incertidumbre del valor que a menor amplitud nos da mayor probabilidad de que el valor se encuentre en esta zona.

```
mdata.ts.p1 <- naive(ts.n1, 12)
plot(mdata.ts.p1)
```

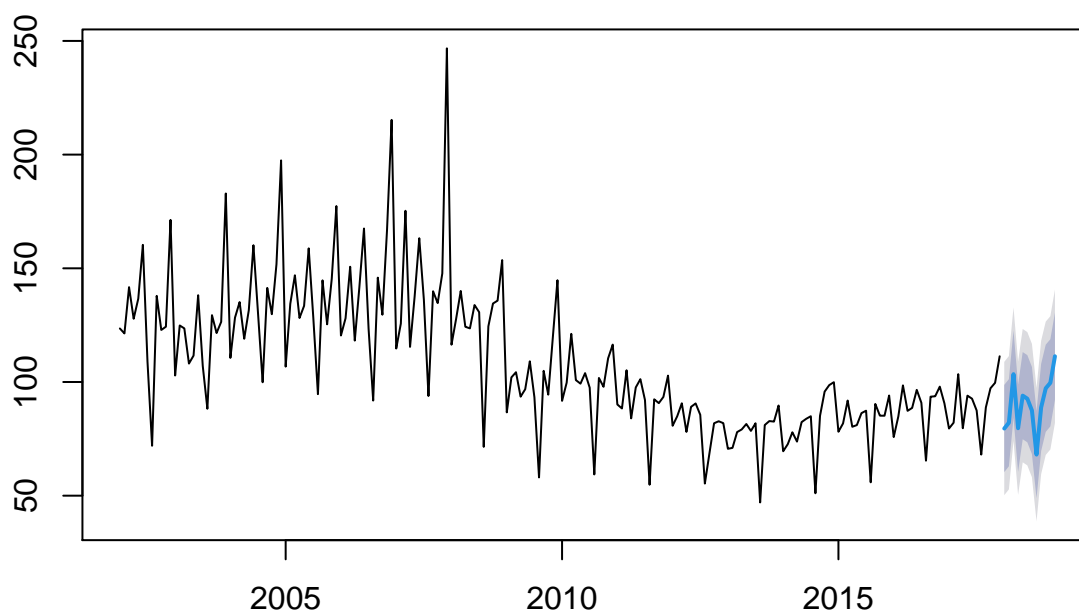
Forecasts from Naive method



Seasonal naïve Este método utiliza el último año para predecir según estaciones observadas en el último año.

```
mdata.ts.p1 <- snaive(ts.n1, 12)  
plot(mdata.ts.p1)
```

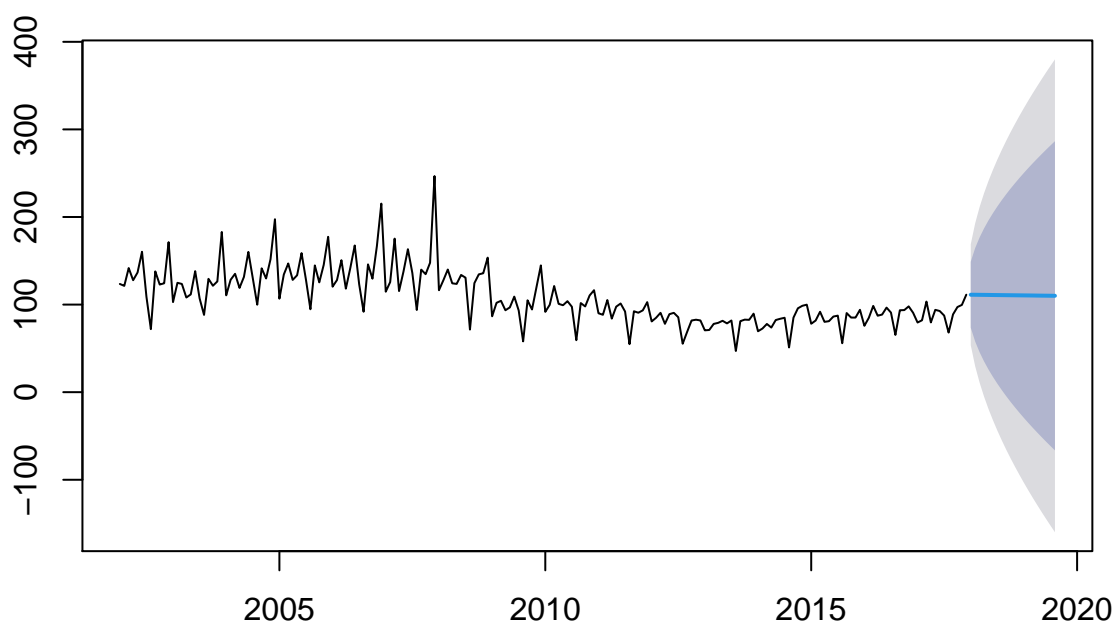
Forecasts from Seasonal naive method



Drift method Este método se basa en el anterior y que la cantidad de cambio en el tiempo se establece como el cambio medio en los datos históricos.

```
mdata.ts.p1 <- rwf(ts.n1, 20, drift=TRUE)
plot(mdata.ts.p1)
```

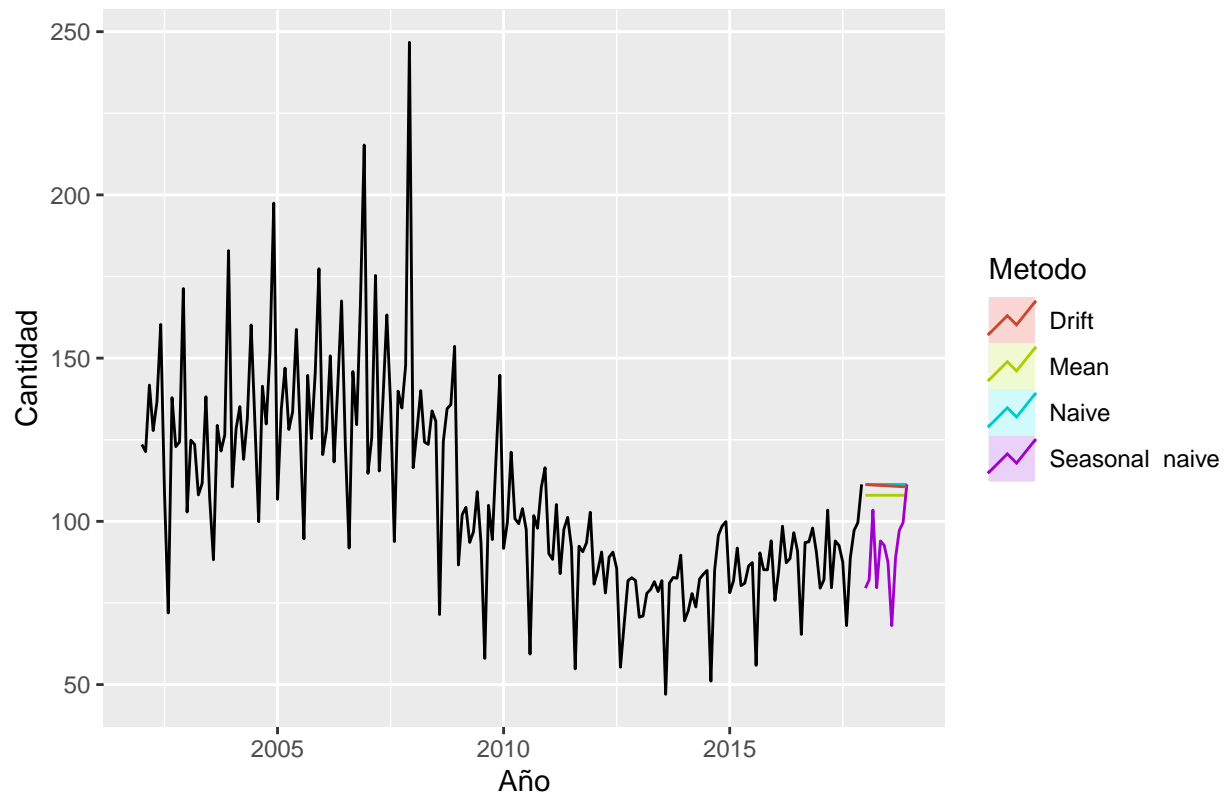
Forecasts from Random walk with drift



Para realizar un mejor análisis vamos a integrar los metodos utilizados en un gráfica para predecir valores futuros a un año.

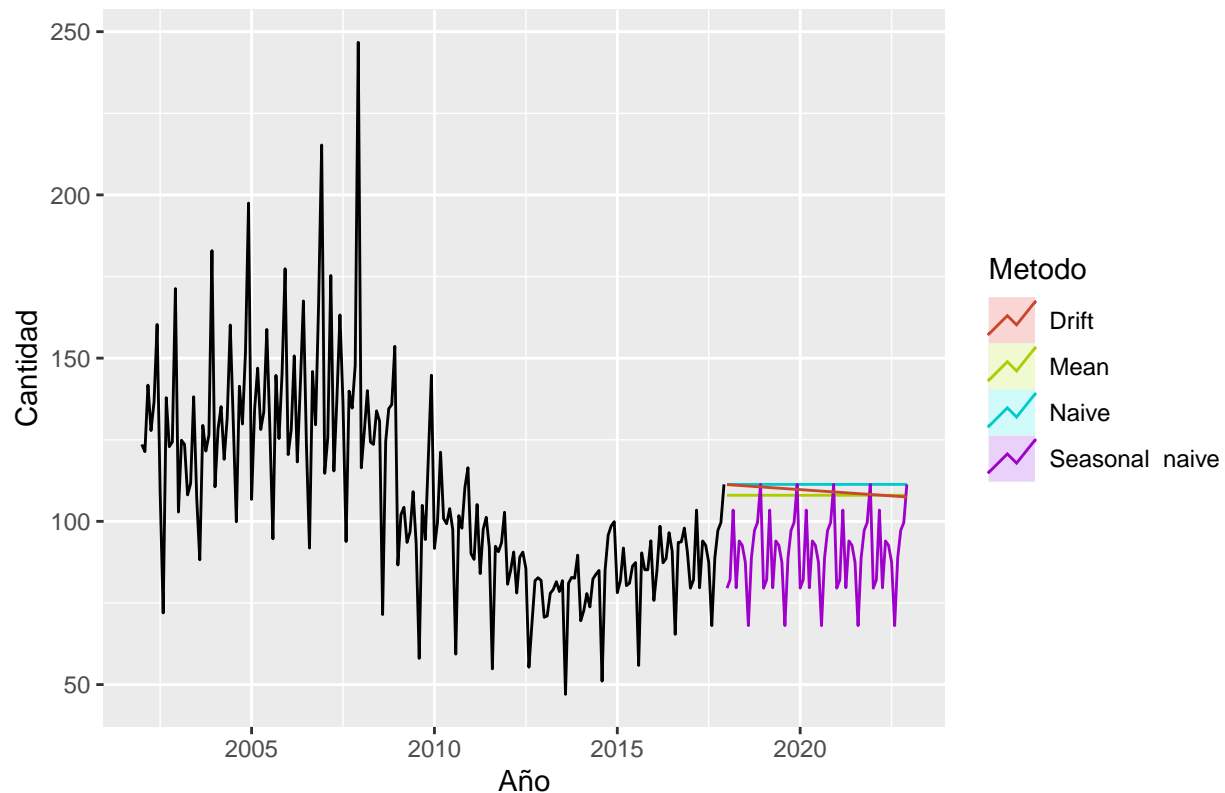
```
autoplot(ts.n1)+forecast::autolayer(meanf(ts.n1,h=12),PI=FALSE,series="Mean") + forecast::autolayer(naivef(ts.n1,h=12),PI=FALSE,series="Naive") +  
  forecast::autolayer(snaive(ts.n1,h=12),PI=FALSE,series="Seasonal naive") +  
  forecast::autolayer(rwf(ts.n1,drift=TRUE,h=12),PI=FALSE,series="Drift") +  
  ggtitle("Pronóstico para la fabricación de un producto electrónico para un año")+xlab("Año") + ylab("Valor")
```

Pronóstico para la fabricación de un producto electrónico para un año



```
autoplot(ts.n1)+forecast::autolayer(meanf(ts.n1,h=60),PI=FALSE,series="Mean") + forecast::autolayer(naivef(ts.n1,h=60),PI=FALSE,series="Naive") +
forecast::autolayer(snaive(ts.n1,h=60),PI=FALSE,series="Seasonal naive") +
forecast::autolayer(rwf(ts.n1,drift=TRUE,h=60),PI=FALSE,series="Drift") +
ggtitle("Pronóstico para la fabricación de un producto electrónico para 5 años")+xlab("Año") + ylab("Cantidad")
```

Pronóstico para la fabricación de un producto electrónico para 5 años



Conclusión Los métodos de predicciones que se utilizaron anteriormente pueden ser interesantes, sin embargo pueden ser muy simples para largas estimaciones de tiempo.

Al graficar el pronóstico, se puede observar que el método de la media (línea roja) no ayuda mucho debido a que nuestra serie tiene tendencia, ni el caso de naive ya que de igual forma solo desplaza una línea sobre la última observación. Finalmente, el método que mejor pronostica para este caso es el Naive estacional, pues este replica el comportamiento del último año de la serie con su estacionalidad.

Segunda Parte:

Gasto Turístico, Resultados por comunidades autónomas (Andalucía): Gasto de los turistas internacionales según comunidad autónoma de destino principal (Andalucía) Observaciones: Gasto total viaje. Tipo de dato: Dato base. Unidades: Millones Euros. fuente: <https://www.ine.es/jaxiT3/Tabla.htm?t=10839&L=0>

```
series=read_excel("Turism.xlsx")
head(series)
```

```
## # A tibble: 1 x 62
##   '2015M10' '2015M11' '2015M12' '2016M01' '2016M02' '2016M03' '2016M04'
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1    1049      530      554      509      487      721      992
## # ... with 55 more variables: '2016M05' <dbl>, '2016M06' <dbl>,
## # '2016M07' <dbl>, '2016M08' <dbl>, '2016M09' <dbl>, '2016M10' <dbl>,
## # '2016M11' <dbl>, '2016M12' <dbl>, '2017M01' <dbl>, '2017M02' <dbl>,
## # '2017M03' <dbl>, '2017M04' <dbl>, '2017M05' <dbl>, '2017M06' <dbl>,
## # '2017M07' <dbl>, '2017M08' <dbl>, '2017M09' <dbl>, '2017M10' <dbl>,
```

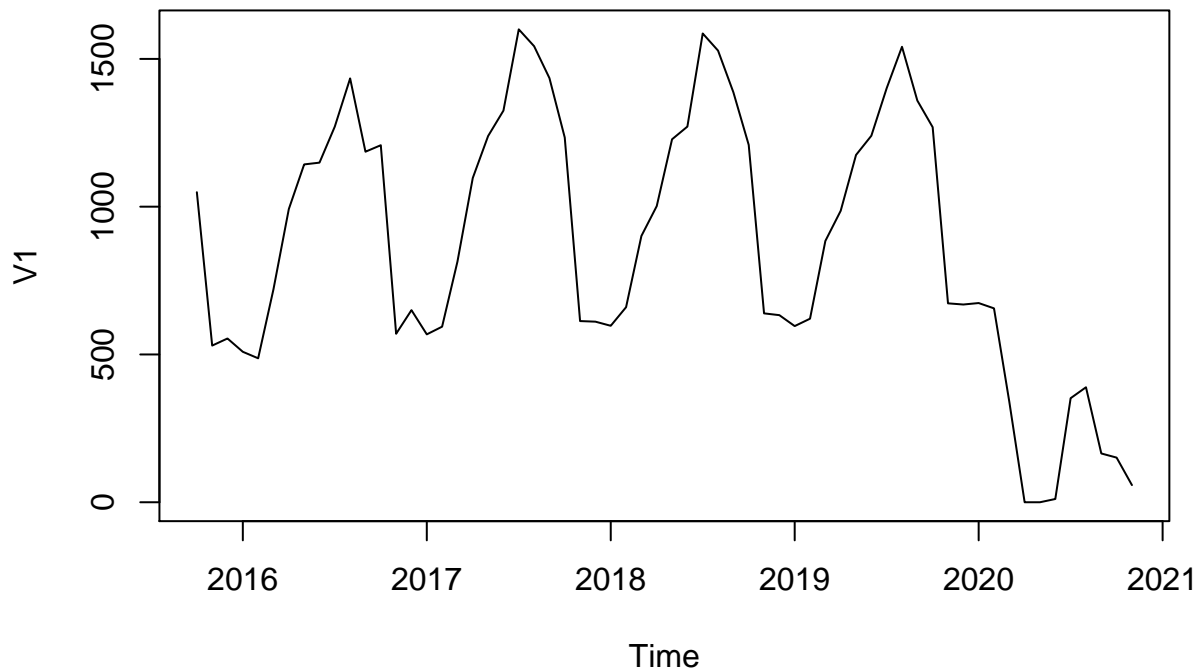
```
## # '2017M11' <dbl>, '2017M12' <dbl>, '2018M01' <dbl>, '2018M02' <dbl>,
## # '2018M03' <dbl>, '2018M04' <dbl>, '2018M05' <dbl>, '2018M06' <dbl>,
## # '2018M07' <dbl>, '2018M08' <dbl>, '2018M09' <dbl>, '2018M10' <dbl>,
## # '2018M11' <dbl>, '2018M12' <dbl>, '2019M01' <dbl>, '2019M02' <dbl>,
## # '2019M03' <dbl>, '2019M04' <dbl>, '2019M05' <dbl>, '2019M06' <dbl>,
## # '2019M07' <dbl>, '2019M08' <dbl>, '2019M09' <dbl>, '2019M10' <dbl>,
## # '2019M11' <dbl>, '2019M12' <dbl>, '2020M01' <dbl>, '2020M02' <dbl>,
## # '2020M03' <dbl>, '2020M04' <dbl>, '2020M05' <dbl>, '2020M06' <dbl>,
## # '2020M07' <dbl>, '2020M08' <dbl>, '2020M09' <dbl>, '2020M10' <dbl>,
## # '2020M11' <dbl>
```

```
datas <- as.data.frame(t(series))
ts <- ts(datas, frequency = 12, start = c(2015,10))
print(ts)
```

```
##      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 2015      1049  530  554
## 2016  509  487  721  992 1143 1149 1271 1434 1186 1208  570  650
## 2017  568  594  814 1097 1239 1325 1600 1543 1435 1234  613  611
## 2018  597  660  901 1002 1228 1271 1586 1528 1387 1209  639  633
## 2019  596  621  884  986 1175 1240 1401 1541 1359 1269  673  669
## 2020  674  656  340    0    0   11  352  389  165  151   58
```

Serie temporal del gasto turistico en Andalucia 2015-2020

```
plot.ts(ts)
```

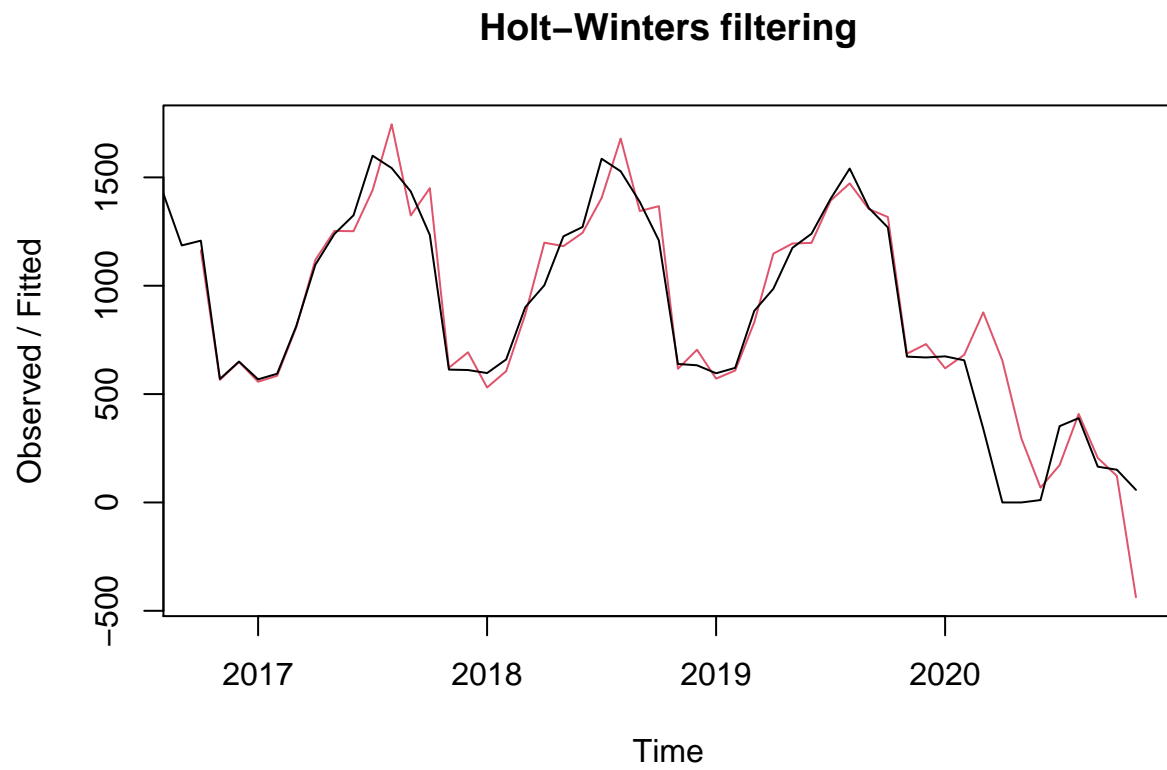



```
su = HoltWinters(ts)
su
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = ts)
##
## Smoothing parameters:
##  alpha: 0.8646753
##  beta : 0
##  gamma: 1
##
## Coefficients:
##           [,1]
## a      390.02374
## b       10.28642
## s1    -359.88546
## s2    -412.74229
## s3    -419.38846
## s4    -281.39726
## s5     -65.52252
## s6     179.03254
## s7     229.24198
## s8     404.72663
## s9     448.22958
```

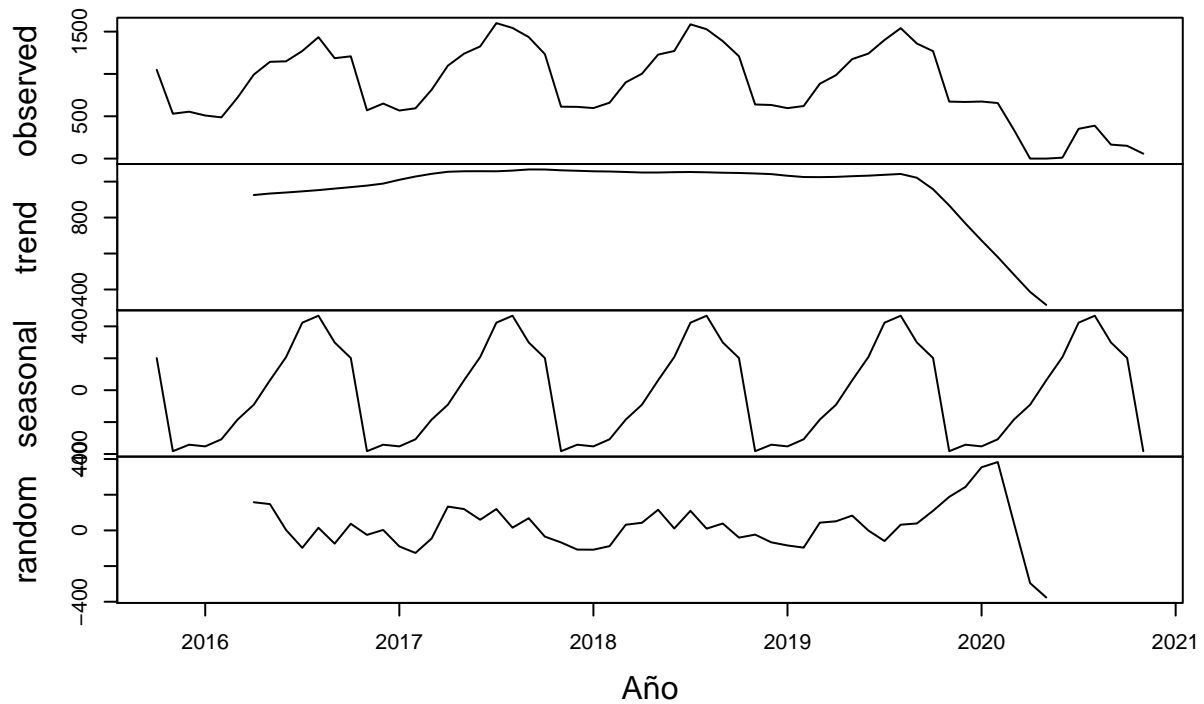
```
## s10 248.89293
## s11 199.57692
## s12 -332.02374
```

```
plot(su)
```



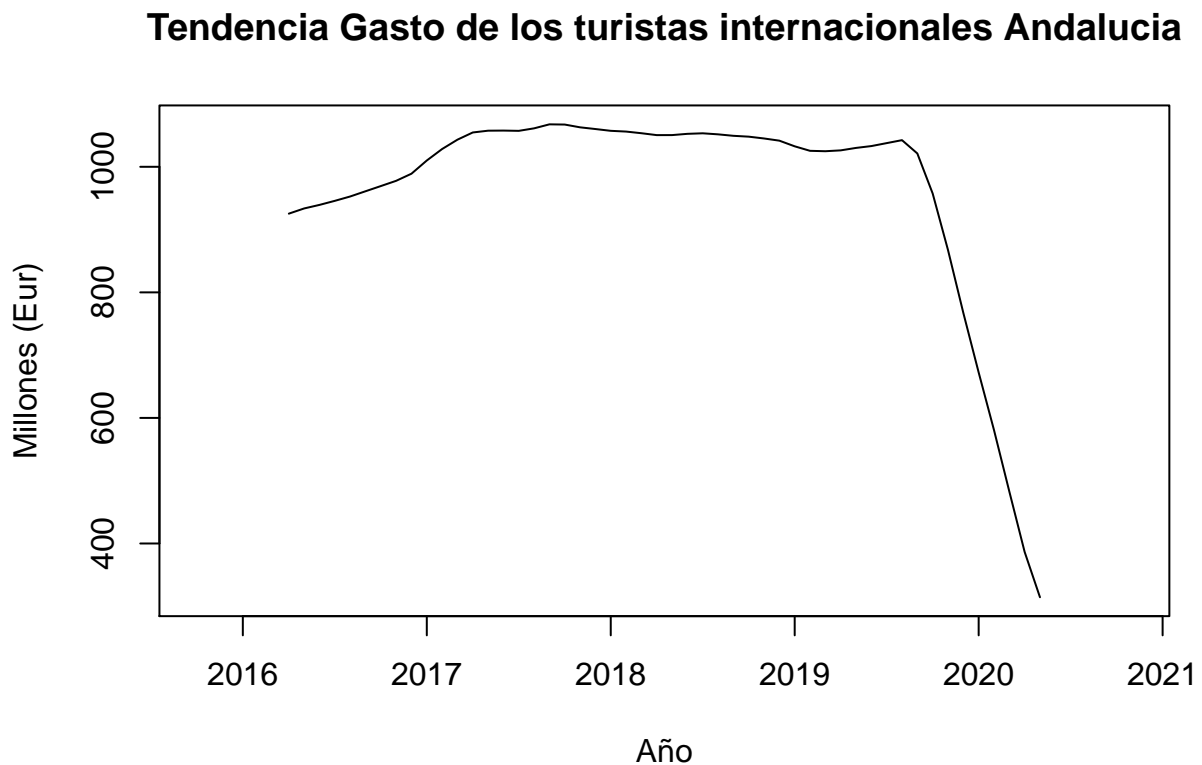
```
ts.desc = decompose(ts)
plot(ts.desc, xlab='Año')
```

Decomposition of additive time series



La descomposición de la serie temporal nos permite visualizaciones por separado para lo observado, la tendencia y estaciones aparentemente claras, además del componente aleatorio; Estos aún son en años.

```
plot(ts.desc$trend, xlab='Año', ylab='Millones (Eur)') +  
title("Tendencia Gasto de los turistas internacionales Andalucía")
```

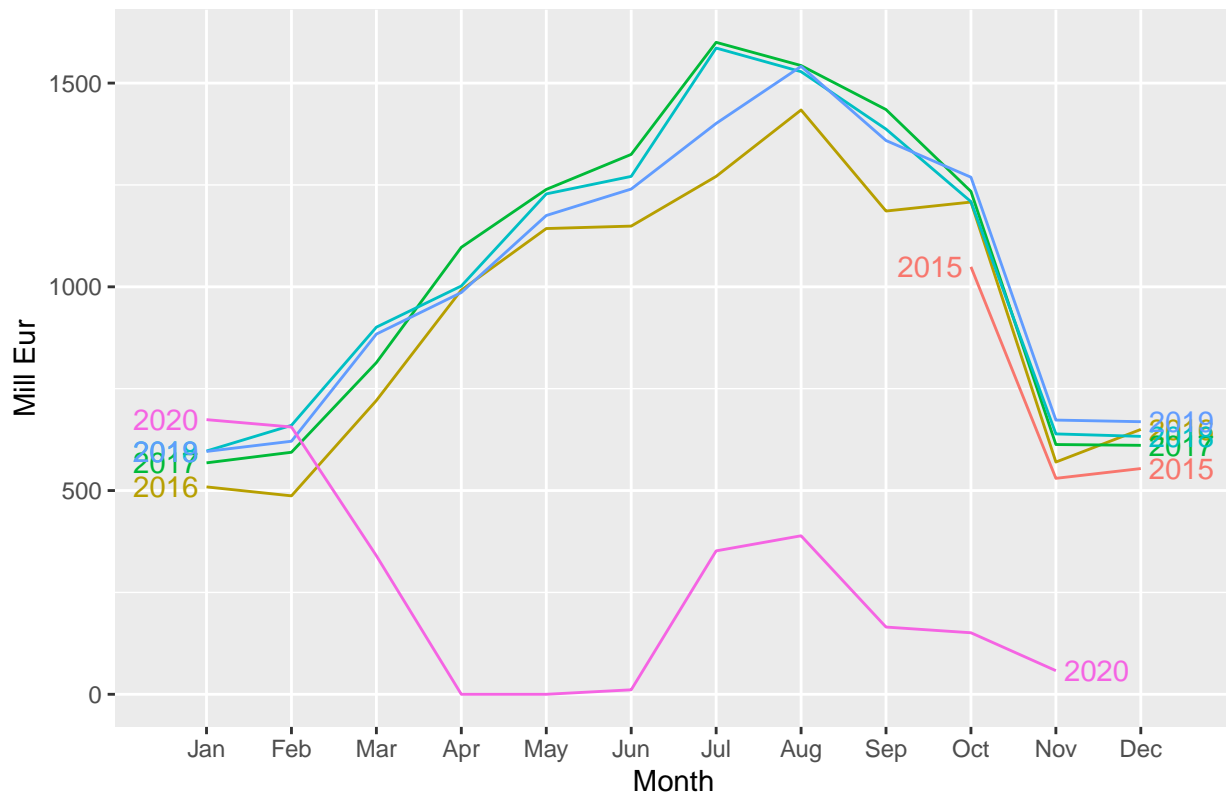


```
## integer(0)
```

En el gráfico anterior se puede hacer una primera observación aparentemente estable hasta que en el año 2020 la tendencia decreciente es determinante. A través del gráfico estacional se puede observar estaciones de la serie según los meses en los diferentes años, que por lo cual, permite mayor visualización sobre el comportamiento del gasto turístico en Andalucía.

```
ggseasonplot(ts, year.labels=TRUE, year.labels.left=TRUE) +  
  ylab("Mill Eur") +  
  ggtitle("Plot Estacional Gasto de los turistas internacionales Andalucía")+  
  theme(plot.title = element_text(hjust = 0.5))
```

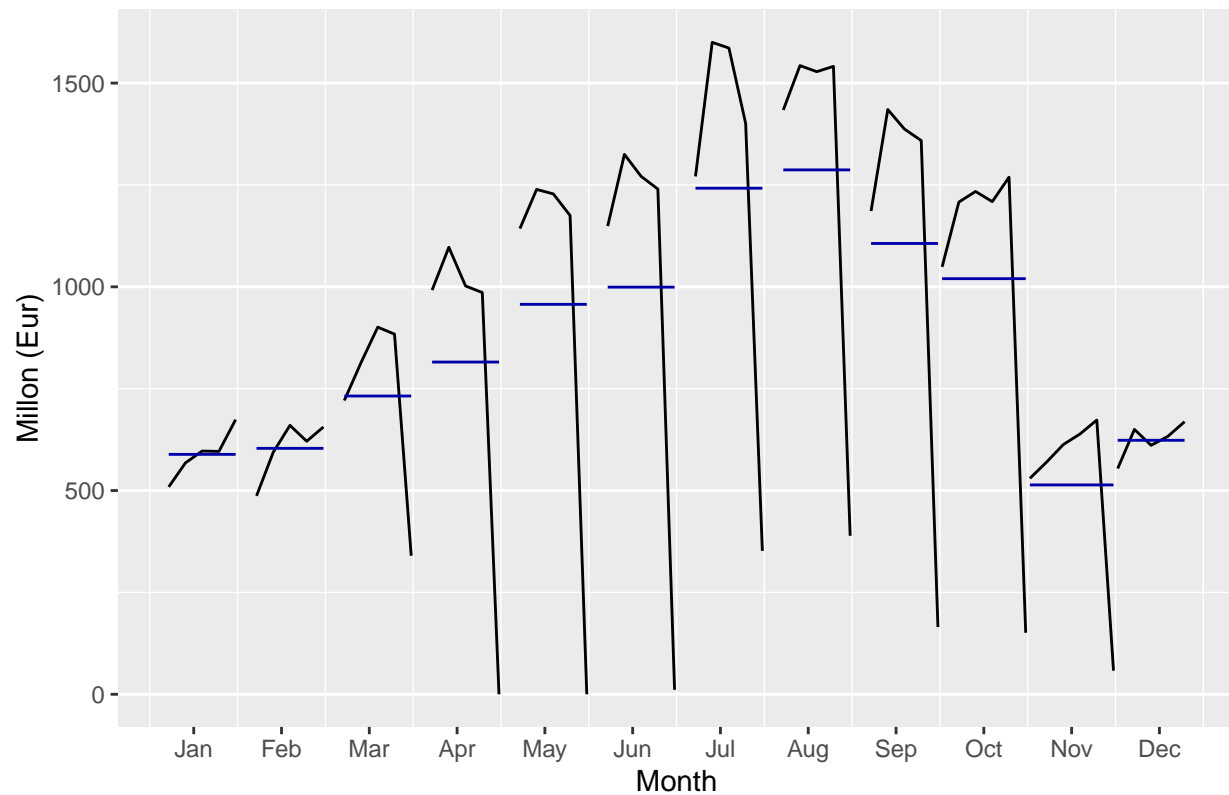
Plot Estacional Gasto de los turistas internacionales Andalucía



Personalmente, elegí ese conjunto de datos por la importancia del sector turismo en Andalucía y porque no es secreto que éste se ha visto afectado en gran medida por la pandemia del covid-19. En el gráfico anterior podemos observar como estaciones con tendencias crecientes en cuando más cerca del verano como Junio a Septiembre (quisiera realizar observaciones sociales sobre lo que representa la Costa del sol para el turismo, pero no quisiera cometer alguna generalización apresurada), los valores más bajos se sitúan al inicio y final del año. Finalmente, el año 2020 muestra un comportamiento contradictorio en cuanto a gráficas que podríamos asociar a los temas de la pandemia y confinamiento, sin embargo en los meses de verano se puede apreciar una creciente nuevamente. Subseries estacional: El siguiente gráfico enfatiza los patrones estacionales es donde los datos de cada temporada se recopilan juntos en mini gráficos de tiempo separados, en los cuales resaltar nuevamente que los meses que corresponden al verano cuentan con un media más elevada y los meses más alejados de verano con un media más baja. Una punto interesante al realizar observaciones a través de la media es que una medida que puede verse fácilmente afectada por datos atípicos (outliers) como lo fue el año 2020 especialmente en el verano.

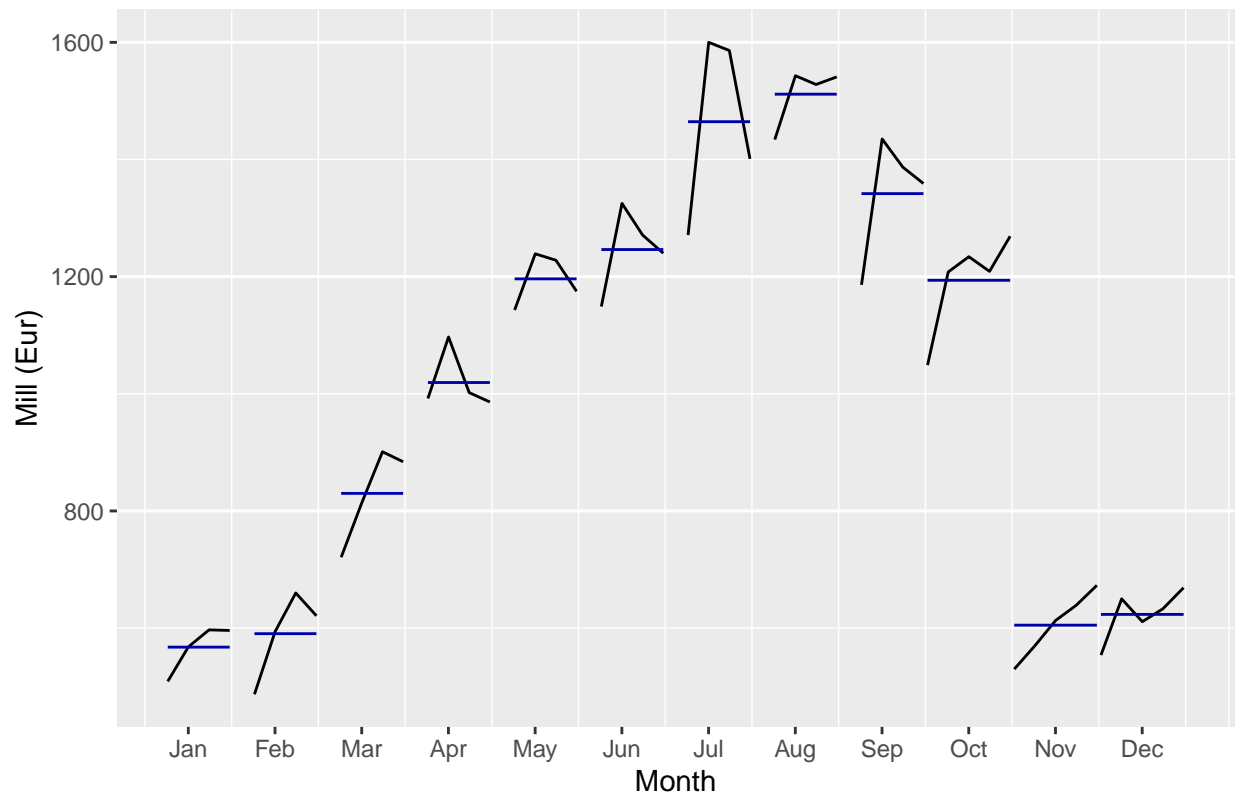
```
ggsubseriesplot(ts) +
  ylab("Millon (Eur)") +
  ggtitle("Subseries Estacional: Gasto turistas internacionales Andalucía (Incluyendo 2020)") +
  theme(plot.title = element_text(hjust = 0.5))
```

Subseries Estacional: Gasto turistas internacionales Andalucía (Incluyendo 20



```
ts.adj = ts(datas, frequency=12, start=c(2015,10), end=c(2019,12))
ggsubseriesplot(ts.adj) +
  ylab("Mill (Eur)") +
  ggtitle("Subseries Estacional: Gasto turismo internacionales Andalucía (hasta 2019)") +
  theme(plot.title = element_text(hjust = 0.5))
```

Subseries Estacional: Gasto turismo internacionales Andalucía (hasta 2019)

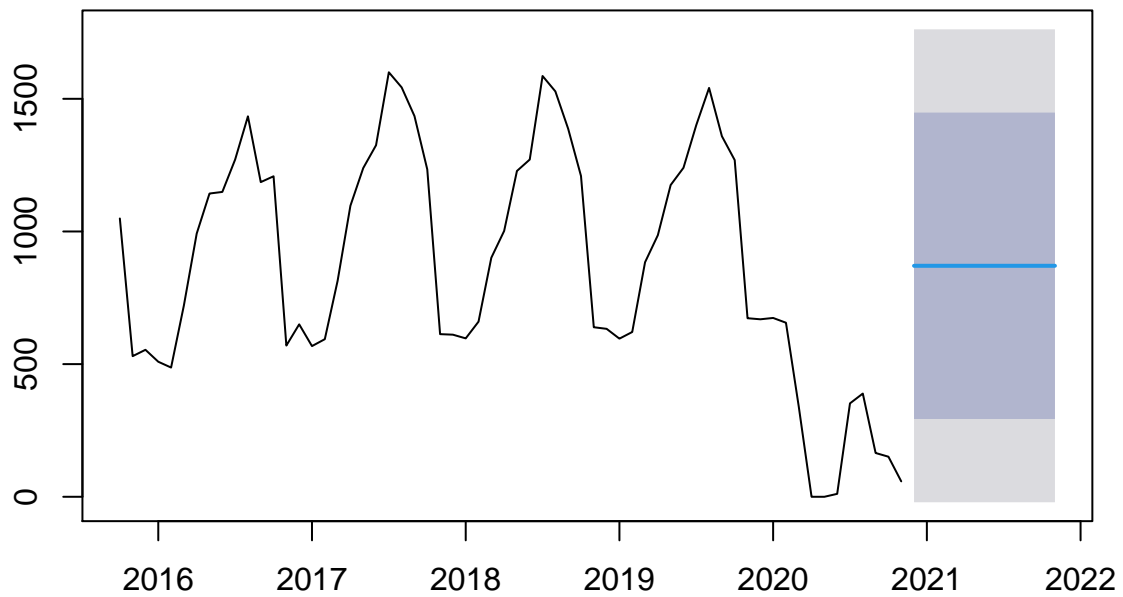


Tanto el gráfico como las subseries han cambiado notoriamente y los valores de las medias son más altos.
Predicción para instantes futuros del gasto de los turistas internacionales en Andalucía (un año)

Método de la media El método de la media muestra un valor al rededor de los 900 debido al histórico de registros .

```
mdata.ts <- meanf(ts,12)
plot(mdata.ts)
```

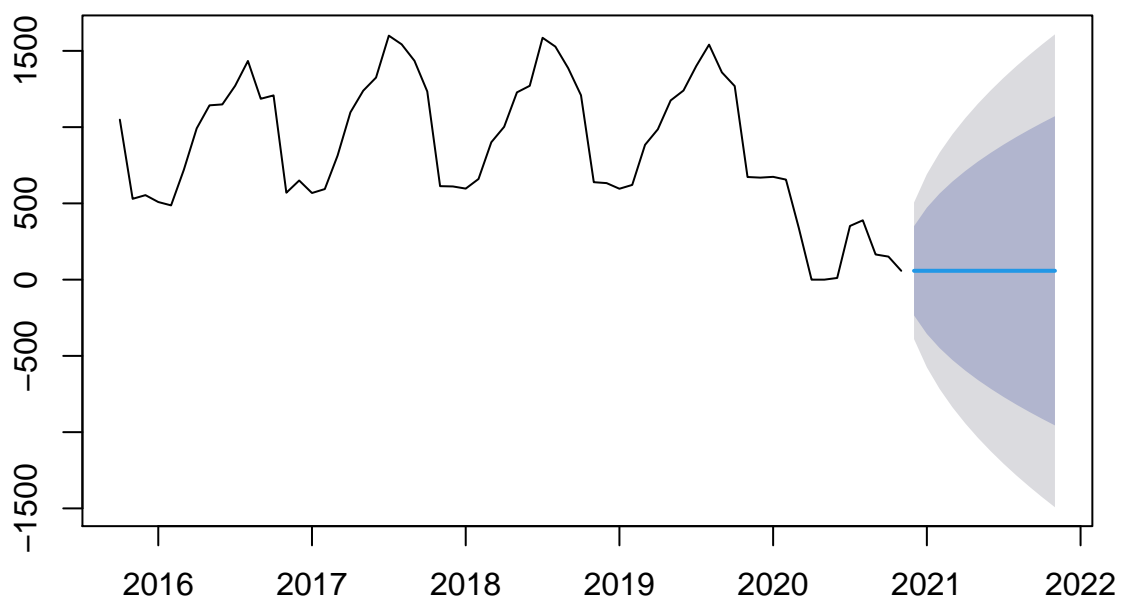
Forecasts from Mean



Metodo Naive Este método muestra el valor que continua el valor observado del 2020 para un instante futuro.

```
mdata.ts <- naive(ts, 12)  
plot(mdata.ts)
```

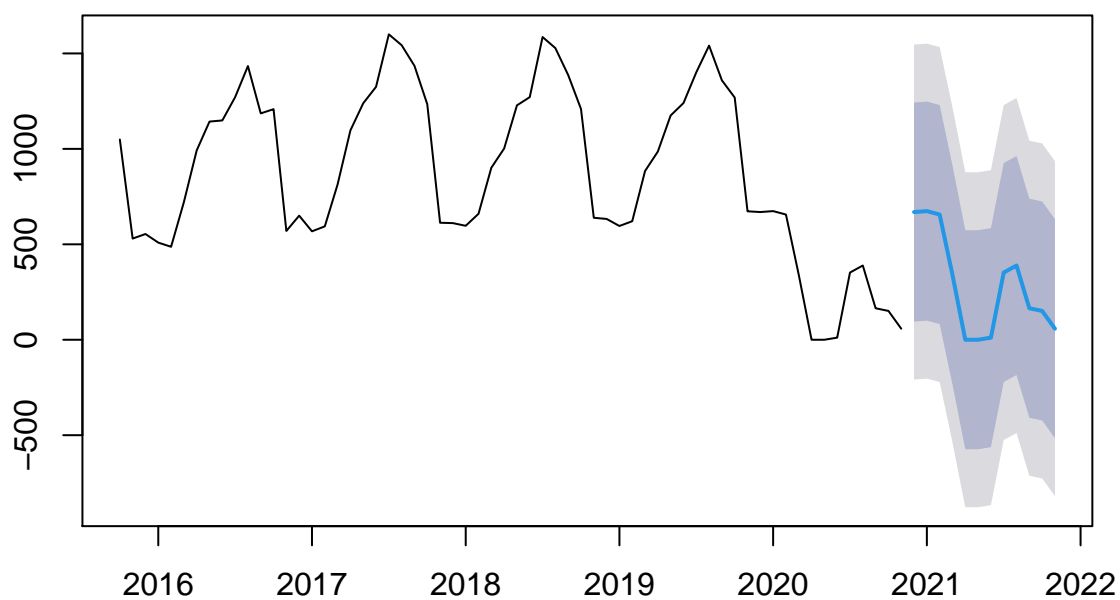

Forecasts from Naive method



Seasonal naïve method Este método está utilizando la estación del año 2020 y sus los valores son más altos.

```
mdata.ts <- snaive(ts, 12)
plot(mdata.ts)
```

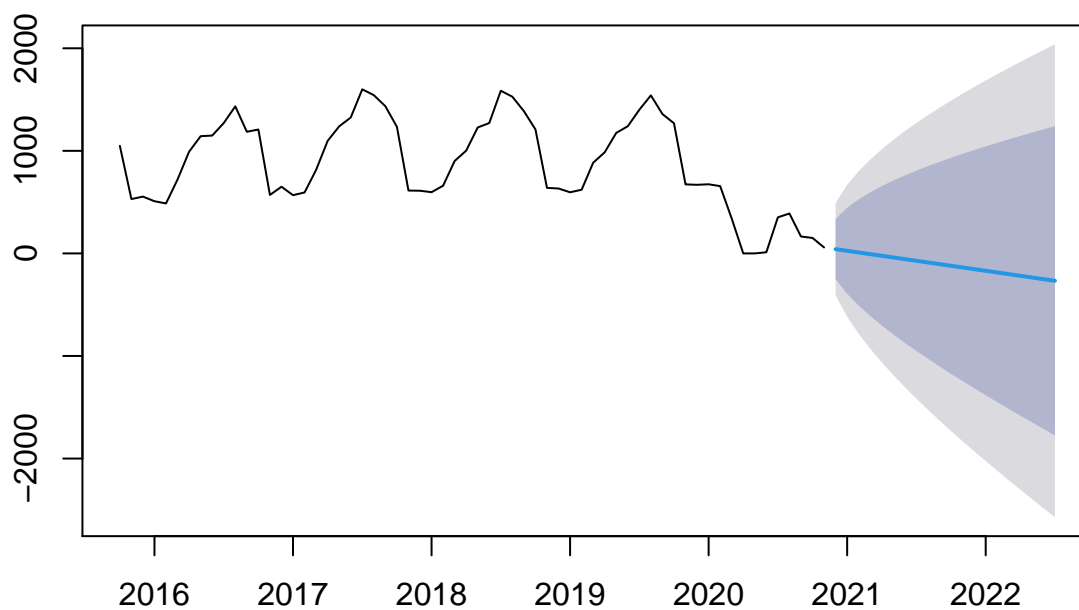
Forecasts from Seasonal naive method



Drift method Según este método que establece el cambio medio en los datos históricos, parece que el gasto del turismo internacional para el 2021 será menor aunque el rango de incertidumbre podría contradecirlo.

```
mdata.ts <- rwf(ts, 20, drift=TRUE)
plot(mdata.ts)
```

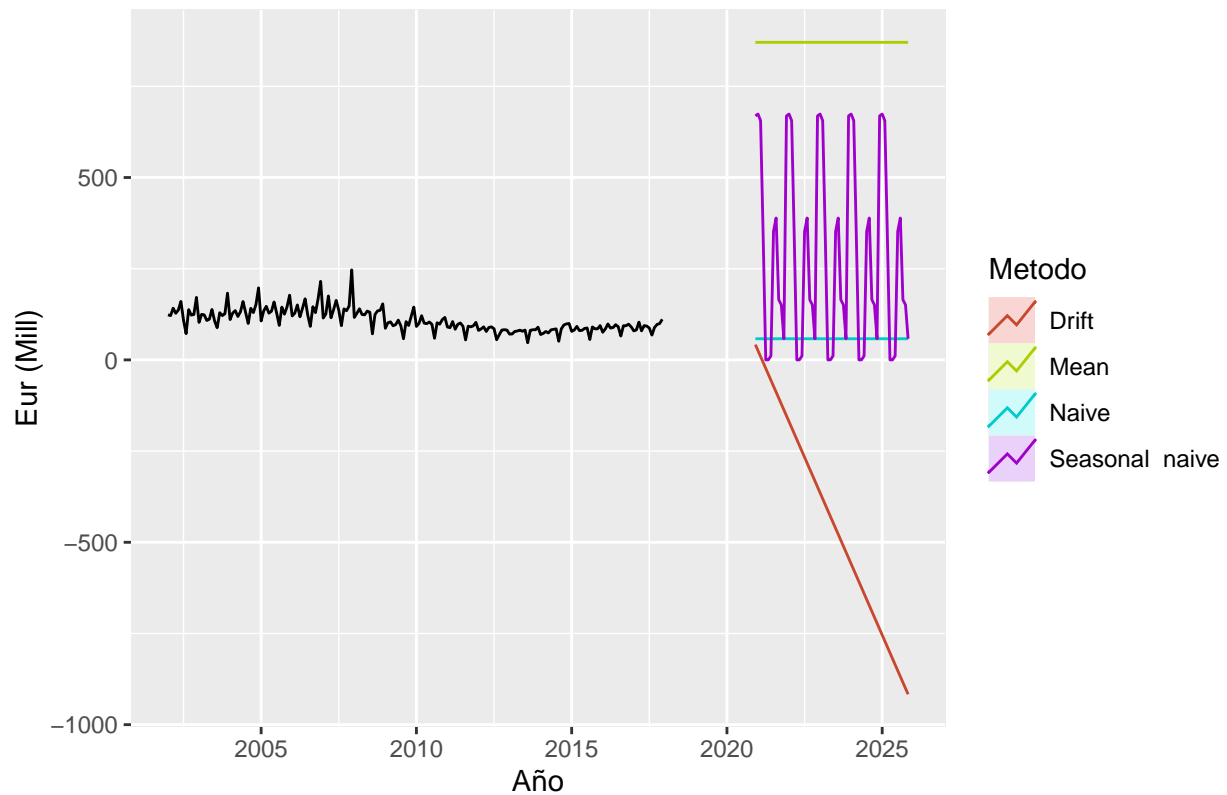
Forecasts from Random walk with drift



Predicción para 5 años

```
autoplot(ts.n1)+forecast::autolayer(meanf(ts,h=60),PI=FALSE,series="Mean") + forecast::autolayer(naive(
  forecast::autolayer(snaive(ts,h=60),PI=FALSE,series="Seasonal naive") +
  forecast::autolayer(rwf(ts,drift=TRUE,h=60),PI=FALSE,series="Drift") +
  ggtitle("Pronóstico del gasto de turismo internacional en Andalucía en 5 años")+xlab("Año") + ylab("E"))
```

Pronóstico del gasto de turismo internacional en Andalucía en 5 años

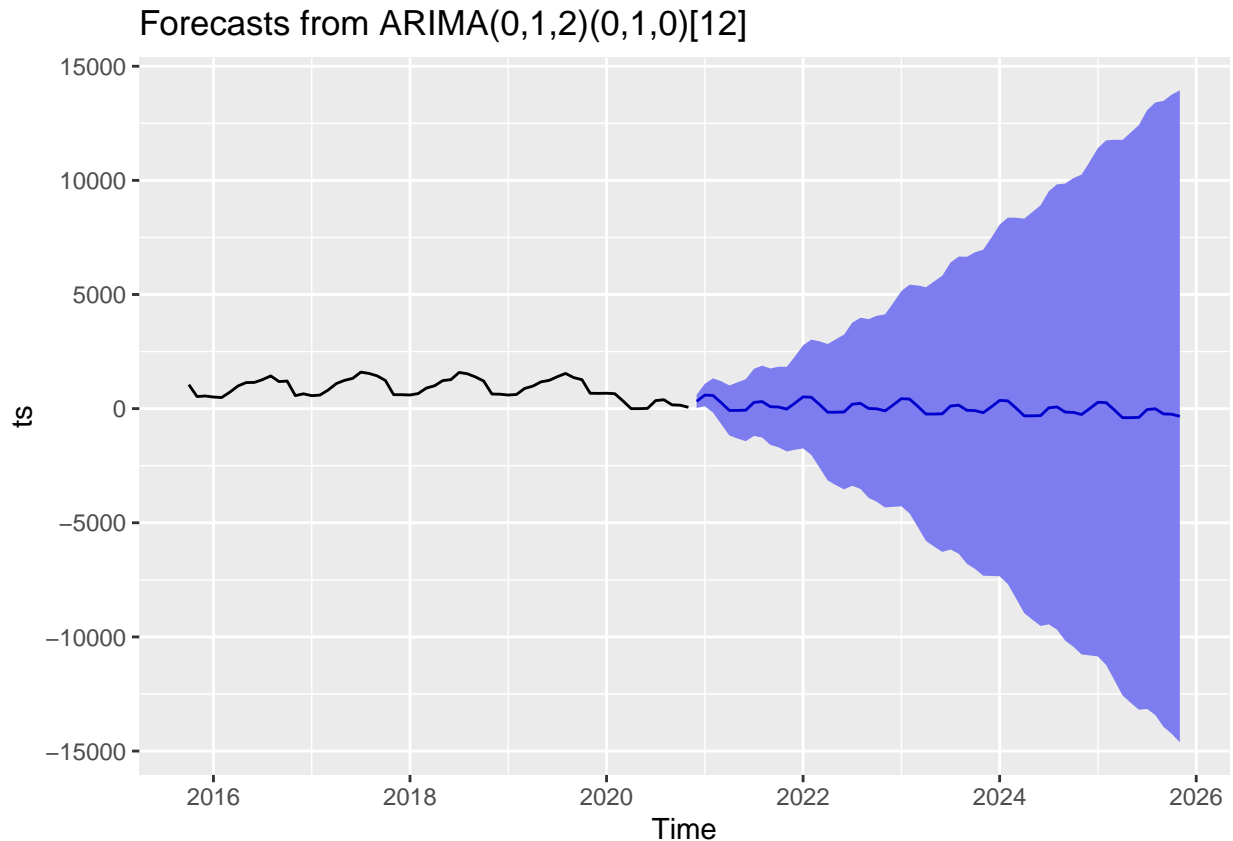


Método ARIMA

```
auto.arima(ts, stepwise = FALSE, approximation = FALSE)
```

```
## Series: ts
## ARIMA(0,1,2)(0,1,0)[12]
##
## Coefficients:
##      ma1      ma2
##      0.3773  0.5802
## s.e.  0.1107  0.2188
##
## sigma^2 estimated as 21714: log likelihood=-313.6
## AIC=633.19  AICc=633.73  BIC=638.87
```

```
ari<- Arima(ts, order=c(0,1,2), seasonal=list(order=c(0,1,0),period=12))
forecast1<-forecast(ari, level = c(95), h = 60)
autoplot(forecast1)
```



Conclusión

Me gustó poder utilizar conjuntos de datos reales, nacionales y del contexto que vivimos actualmente. Aunque el futuro que vivimos es más incierto que lo que hubiese pensado un año atrás, la practica anterior ya nos indicaba que los métodos utilizados para las predicciones hasta ahora no son suficientes para estos tipos de series temporales, el modelo ARIMA finalmente es el que mejor parece ajustarse a los datos que se tenían históricos “normales” con los registrados para el 2020. Un ejercicio muy interesante que se desenvuelve entre el análisis de series temporales y las visualizaciones. ##Bibliografia

<https://otexts.com/fpp2/tspatterns.html> https://rstudio-pubs-static.s3.amazonaws.com/398708_9bff64be426c47b689871d92.html
http://sigma.iimas.unam.mx/jsantibanez/Cursos/Regresion/2018_2/ejemplos/05_cobertura.html
<https://rpubs.com/joser/SeriesTemporalesBasicas> <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/EDescrip/tema7.pdf> <https://bookdown.org/content/2274/series-temporales.html> <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/ts>