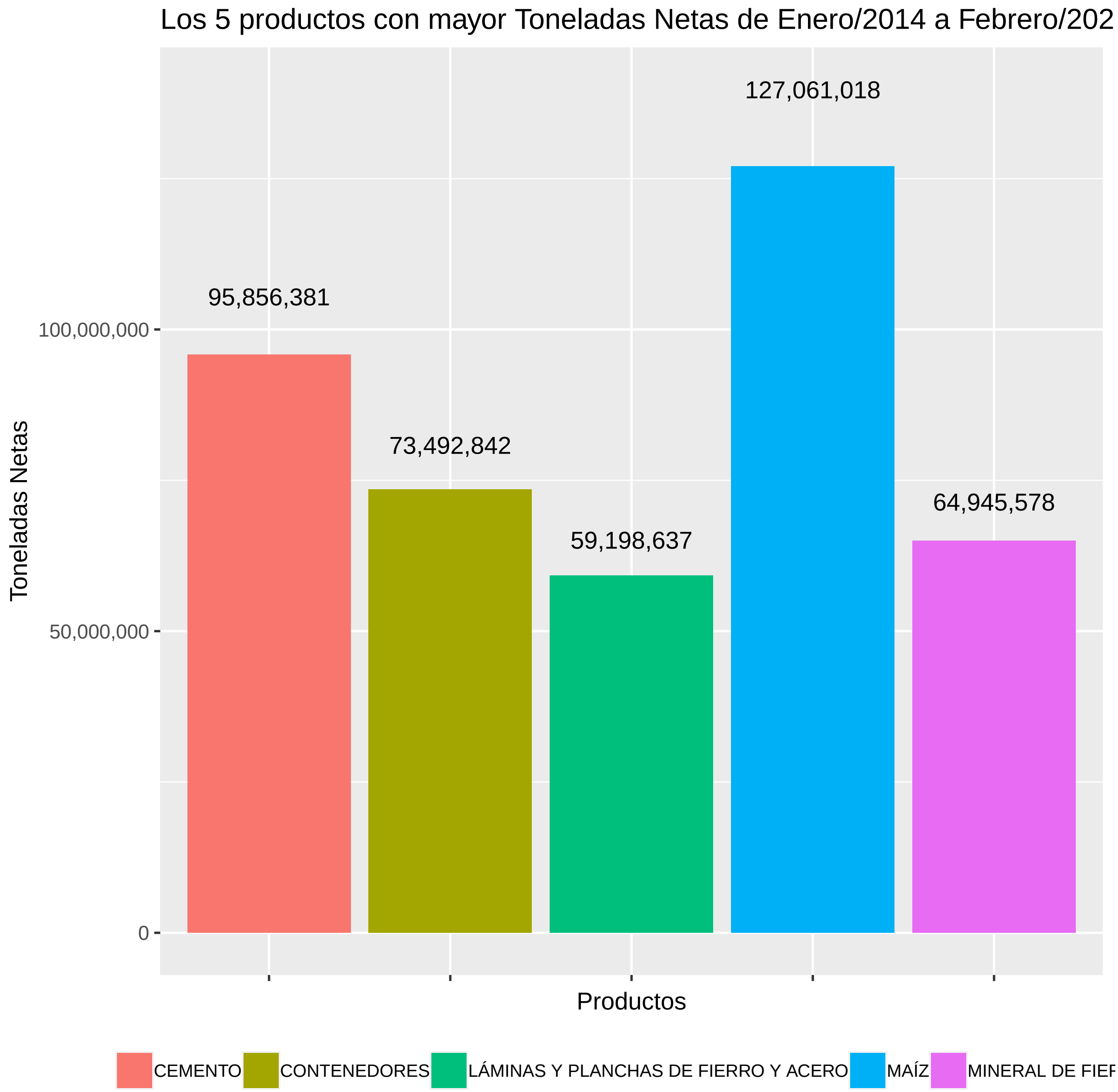
CONTENIDO

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: TTR
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
##
## Attaching package: 'dplyr
## The following objects are masked from 'package:xts':
##
##     first, last
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

1. Análisis

Realizando un histograma de frecuencias absolutas (La tabla dada ya pasó por un filtrado en donde se quedan con los 5 productos con mayor volumen de carga) para ver cuales son los 5 productos con mayor volumen de carga.

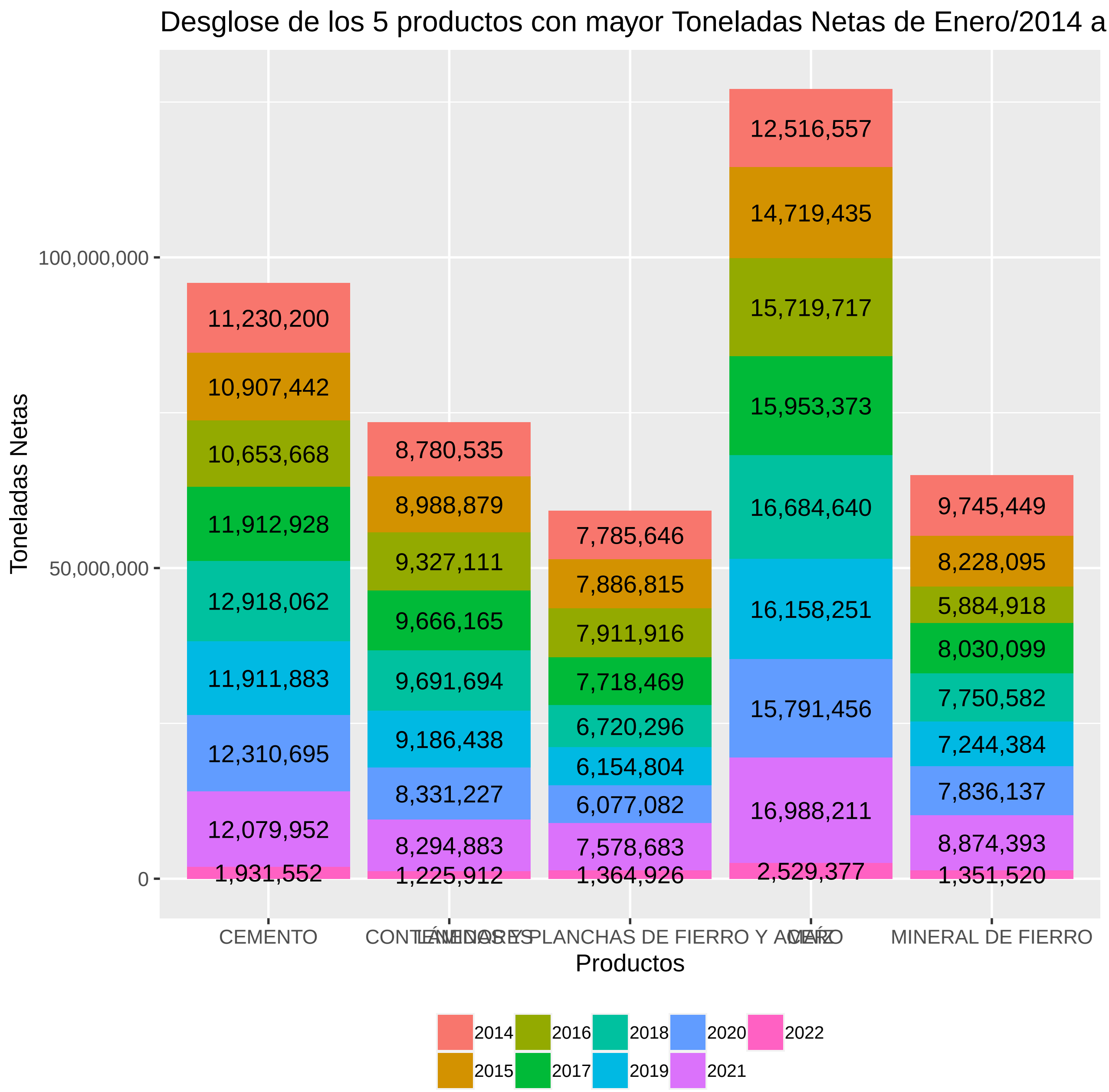


En la cual podemos observar que los productos con mayor volumen de carga ordenados de forma descendente quedan de la siguiente manera:

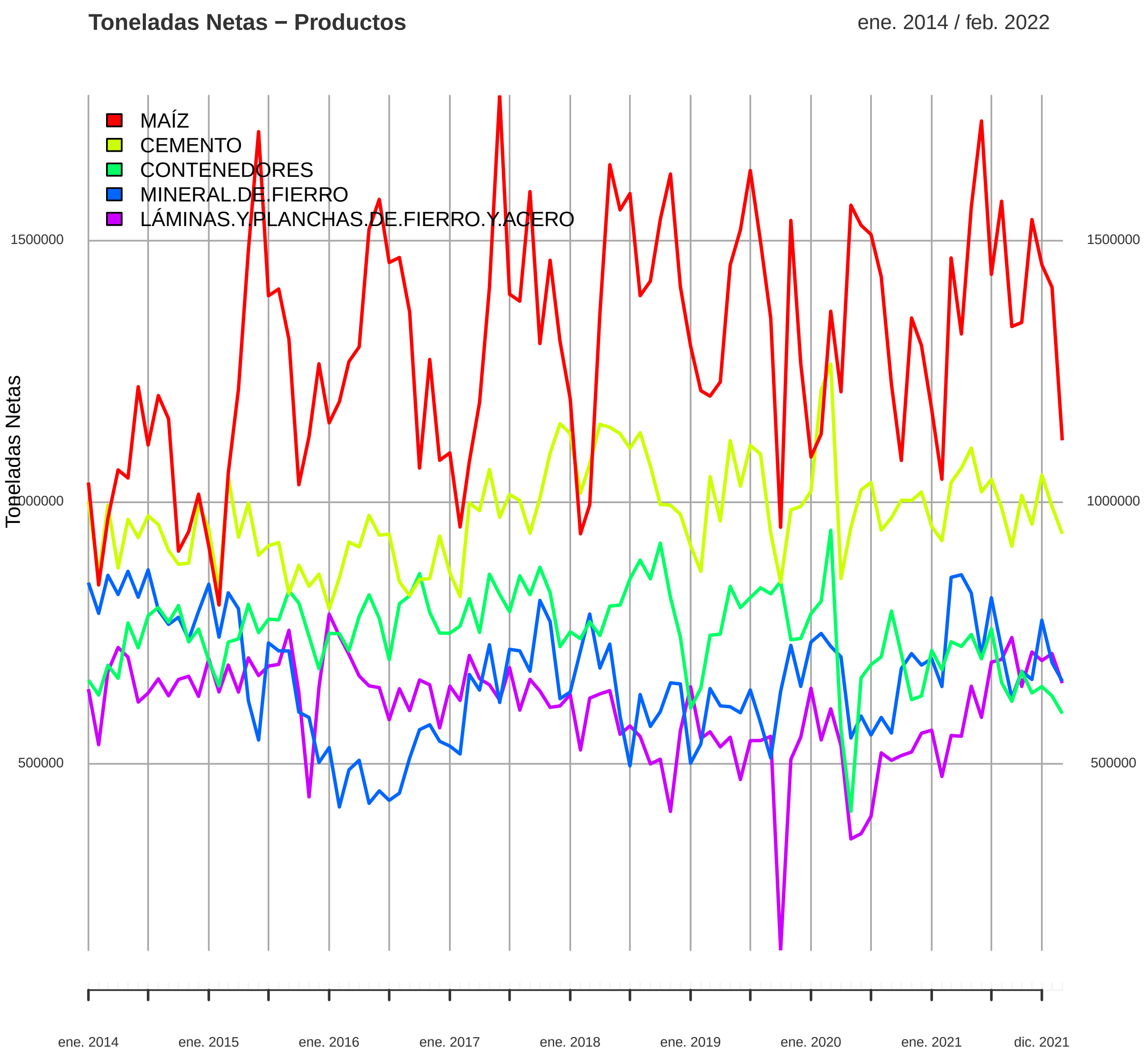
- Maíz con 127,061,018
- Cemento con 95,856,381
- Contenedores con 73,492,892
- Mineral de Fierro con 64,945,578
- Láminas, Planchas de Fierro y Acero con 59,198,637

Haciendo el mismo histograma de frecuencias absolutas con la diferencia de que ahora vamos a poner en el histograma el comportamiento del producto respecto a las toneladas netas transportadas año con año.

```
## [1] "MAÍZ"
## [2] "CEMENTO"
## [3] "CONTENEDORES"
## [4] "MINERAL.DE.FIERRO"
## [5] "LÁMINAS.Y.PLANCHAS.DE.FIERRO.Y.ACERO"

## `summarise()` has grouped output by 'PRODUCTOS'. You can override using the
## `.groups` argument.
```



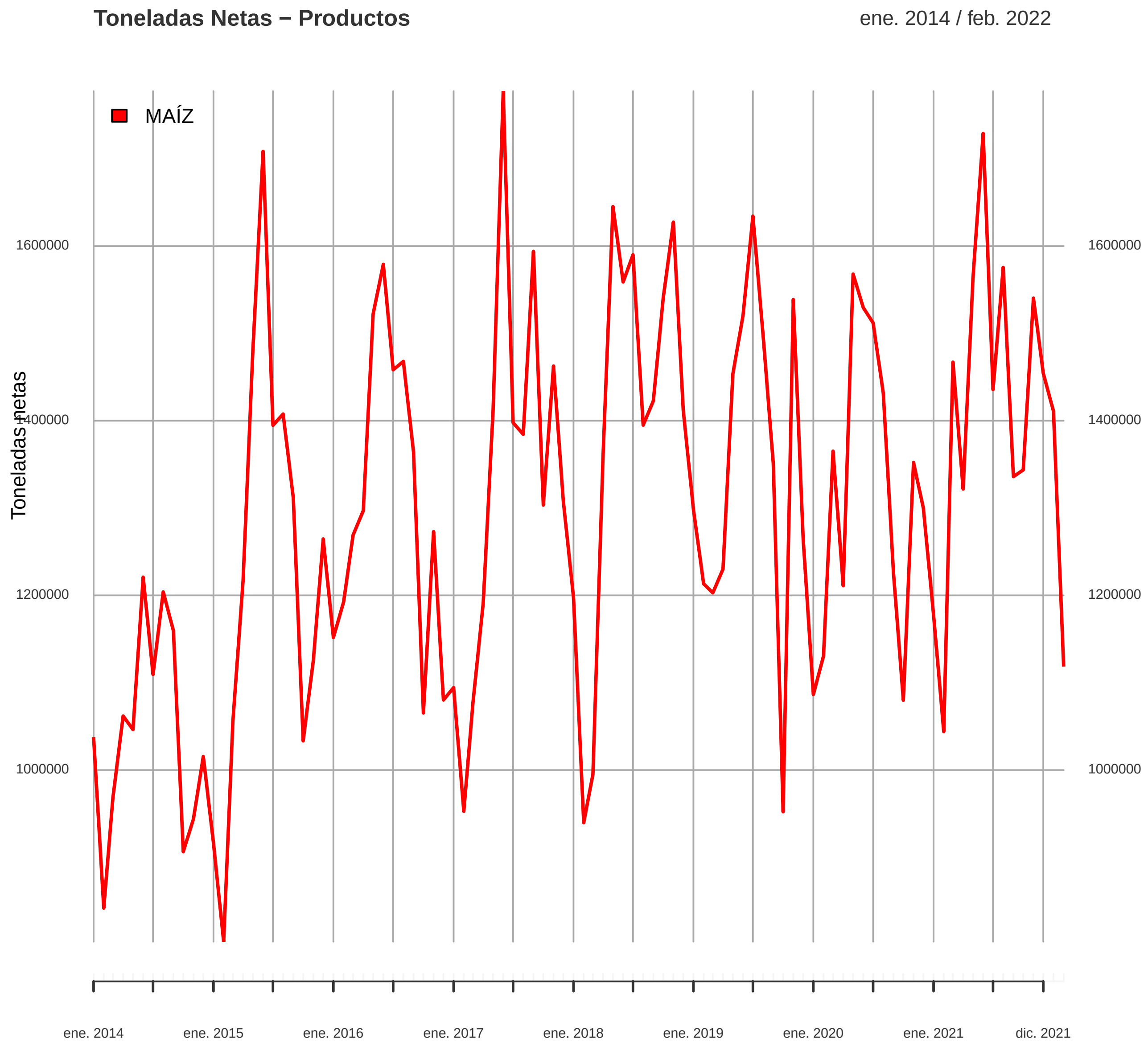
Dado un análisis previo acerca del comportamiento de los 5 productos con mayor volumen, procedemos a realizar el gráfico de la serie temporal de dichos productos.



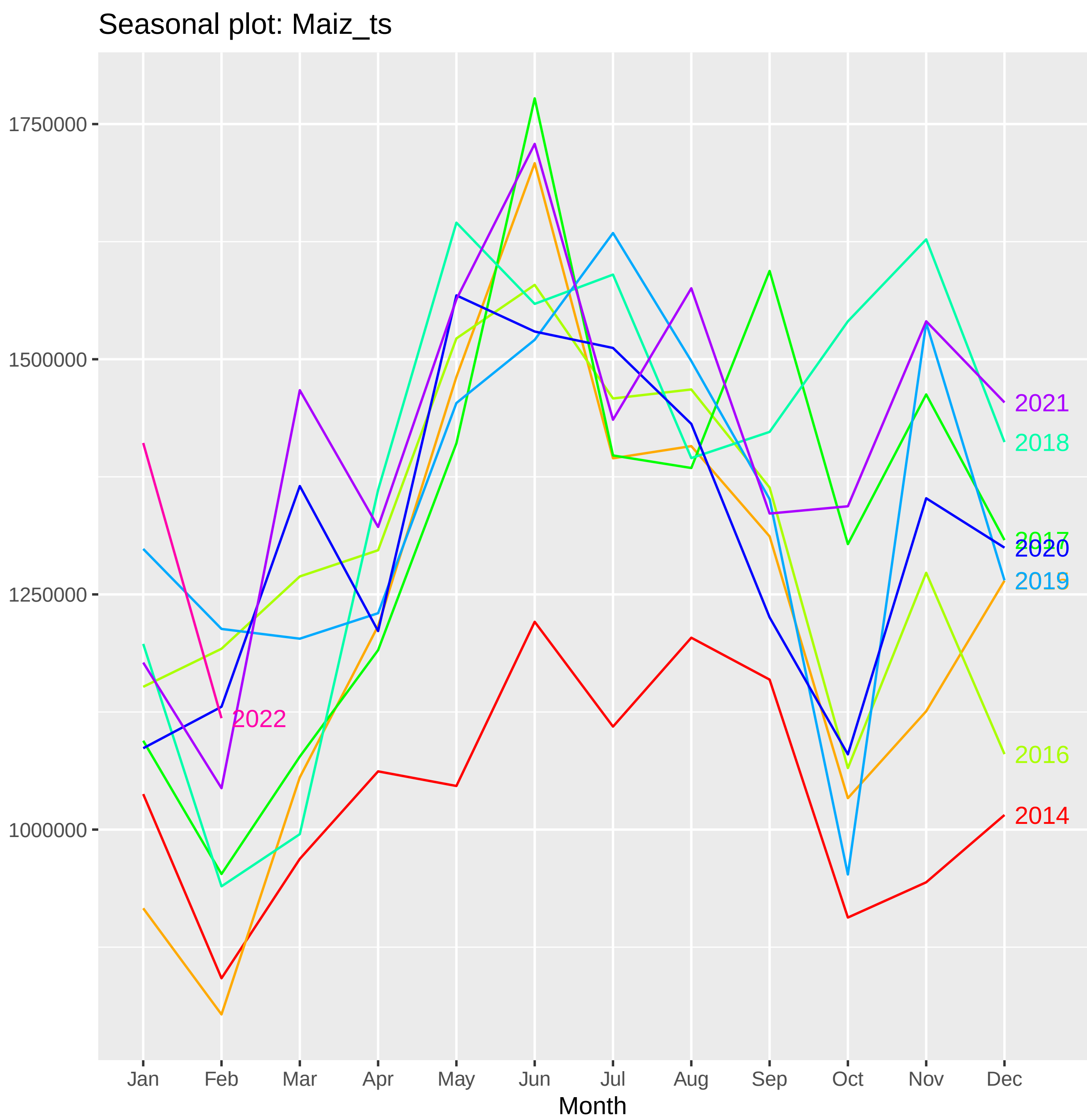
2. MAÍZ

2.1. Parcelas Estacionales

Una **gráfica estacional** es similar a una gráfica de tiempo, excepto que los datos se grafican contra las " estaciones " individuales en las que se observaron los datos. Analicemos la siguiente serie de tiempo.



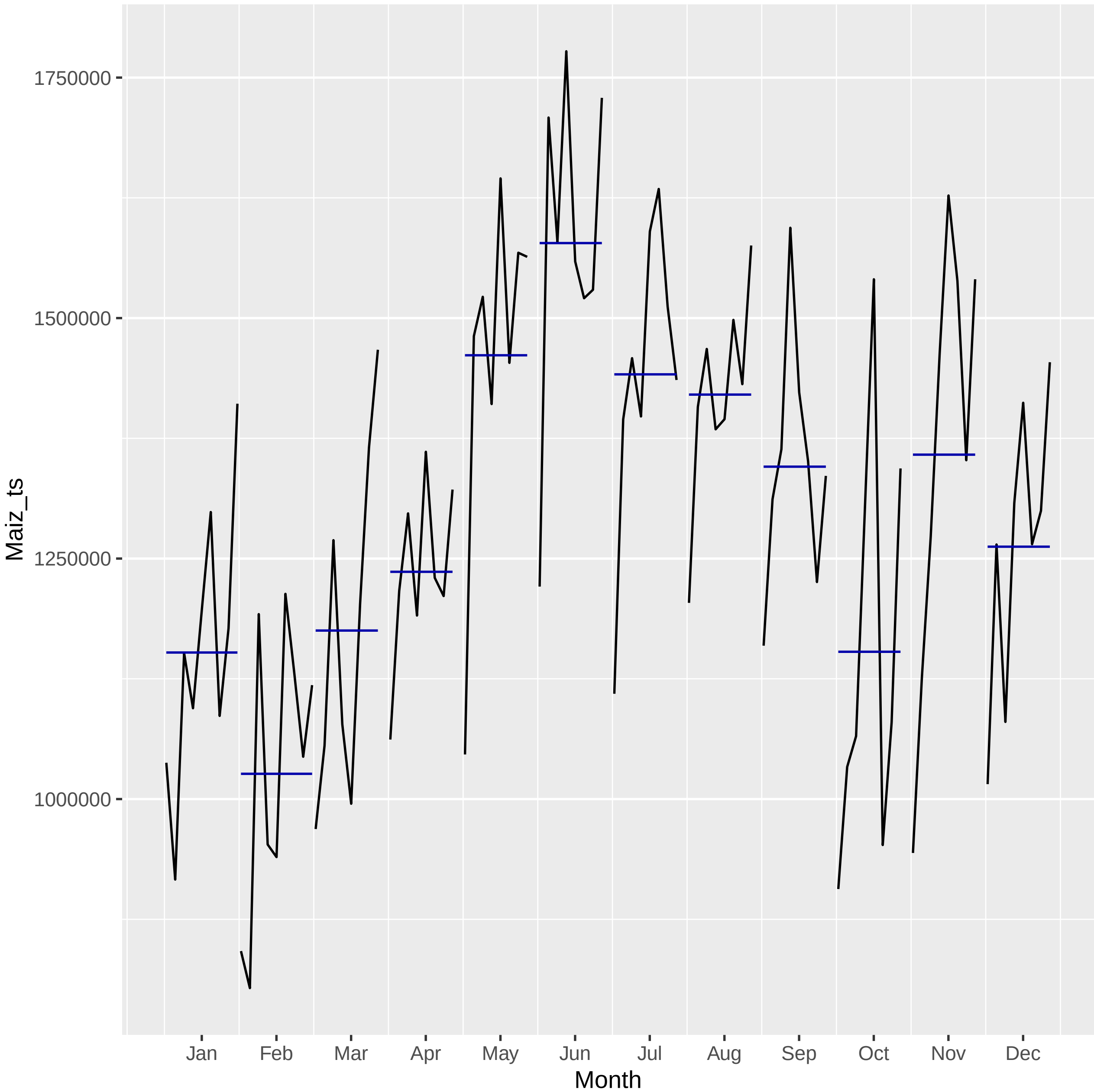
Ahora de su gráfica estacional, tenemos lo siguiente:



Estos son los mismos datos que se mostraron anteriormente, pero los datos de cada temporada se superponen. Un **gráfico estacional** permite ver más claramente el patrón estacional subyacente y es especialmente útil para identificar los años en los que cambia el patrón.

2.2. Gráfica de subseries estacionales

Una gráfica alternativa que enfatiza los patrones estacionales es donde los datos de cada temporada se recopilan juntos en mini gráficas de tiempo serparadas.
Una gráfica de sub-serie estacional faceta la serie temporal por cada estación en el período estacional. Esta faceta forman gráficos de series de tiempo más pequeños que consisten en datos solo de esa temporada. Si tuviera varios años de datos (mensuales, trimestrales, diarios, anuales, etc), el gráfico resultante mostraría un gráfico de serie de tiempo separado por cada período (ya sea mes, año, días) que se este trabajando la serie de tiempo.



Las lineas horizontales azules indican las medias de cada mes. Esta forma de gráfico permite ver claramente el patrón estacional subyacente y también muestra los cambios en la estacionalidad a lo largo del tiempo. Es especialmente útil para identificar cambios dentro de estaciones particulares. Notemos que la serie de tiempo esta dada en forma mensual, es por eso que hace un gráfico estacional separado por mes.

2.3. Componente generales de la serie temporal

Al describir las series temporales, podremos encontrar palabras como "tendencia" y "estacionalidad", las cuales se definen de la siguiente manera:

- **Tendencia:** Una *tendencia* existe cuando hay un aumento o disminución a largo plazo en los datos. No tiene que ser lineal. A veces nos referiremos a una tendencia como "cambio de dirección", cuando podría 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. La estacionalidad es siempre de un período fijo y conocido.

- **Cíclico:** Un *ciclo* ocurre cuando los datos exhiben subidas y bajadas que no tienen una frecuencia fija. Estas fluctuaciones generalmente se deben a las condiciones económicas y, a menudo, están relacionadas con el ”ciclo económico”.

Mucha gente confunde el comportamiento cíclico con el comportamiento estacional, pero en realidad son bastantes diferentes. Si las fluctuaciones no son de una frecuencia fija entonces son cíclicas; si la frecuencia no cambia y está asociada con algún aspecto del calendario, entonces el ”patrón es estacional”. En general, la duración media de los ciclos es mayor que la duración de un patrón estacional. y las magnitudes de los ciclos tienden a ser más variables que las magnitudes de los patrones estacionales. Muchas series de tiempo incluyen tendencia, ciclos y estacionalidad. Al elegir un método de pronóstico, primero necesitaremos identificar los patrones de series de tiempo en los datos y luego elegir un método que pueda capturar los patrones correctamente.

Si asumimos una **descomposición aditiva**, entonces podemos escribir:

$$y_t = S_t + T_t + R_t$$

Donde:

- y_t son los datos.
- T_t es el componente de ciclo de tendencia.
- R_t es el comportamiento restante.

todo en el período t. Alternativamente, una **descomposición multiplicativa** se escribiría como:

$$y_t = S_t \cdot T_t \cdot R_t$$

.

La **descomposición aditiva** es la más adecuada si la magnitud de las fluctuaciones estacionales, o la variación en torno a la tendencia-ciclo, no varía con el nivel de la serie temporal. Cuando la variación alrededor del ciclo tendencia, parece ser proporcionada al nivel de la serie temporal, entonces es más apropiada una **descomposición multiplicativa** . Las descomposiciones multiplicativas son comunes con las series temporales económicas.

Una alternativa al uso de una descomposición multiplicativa es primero transformar los datos hasta que la variación en la serie parezca estable a lo largo del tiempo y luego usar una descomposición aditiva. Cuando se ha usado una transformación logarítmica, esto es equivalente a usar una descomposición multiplicativa porque:

$y_t = S_t \cdot T_t \cdot R_t$ es equivalente a $log\ y_t = log\ S_t + log\ T_t + log\ R_t$

Visualizando de mejor manera las diferencias entre los componentes que integran la serie de tiempo (MAIZ), hacemos lo siguiente:

2.3.1. Descomposición por la función decompose() de R

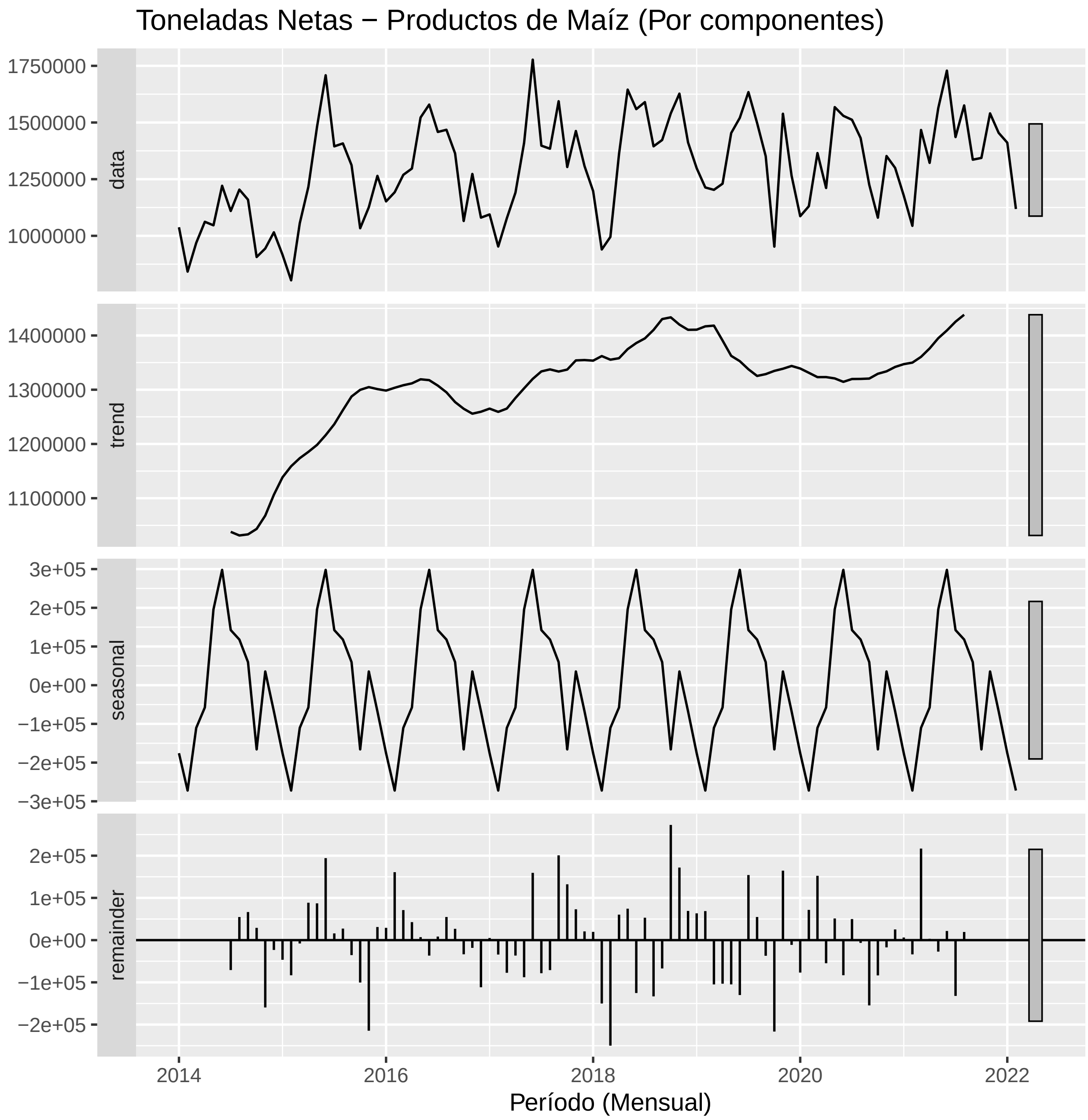
Utilizando la función *decompose()* en tipo aditiva de R, tenemos lo siguiente:

```
## $x
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 2014 1037733.6  841936.6  968867.5 1061894.8 1046361.4 1220890.4 1109470.9
## 2015  916324.4  803495.1 1055547.1 1216488.3 1481237.2 1708447.4 1394690.7
## 2016 1151765.0 1192223.4 1269109.9 1296962.5 1522123.0 1579058.8 1458302.9
## 2017 1094419.1  952764.9 1077635.7 1190780.7 1410637.8 1777352.4 1397730.8
## 2018 1197400.4  939691.1  995145.6 1361005.3 1645195.0 1558841.7 1589994.9
## 2019 1298361.2 1213312.8 1202920.1 1230037.4 1453447.6 1520640.7 1634203.5
## 2020 1086484.4 1130574.9 1365206.0 1211035.2 1567916.6 1529451.7 1511977.6
## 2021 1177533.1 1043996.2 1467083.3 1321736.0 1563662.3 1728951.7 1435664.5
## 2022 1410967.6 1118409.3
```

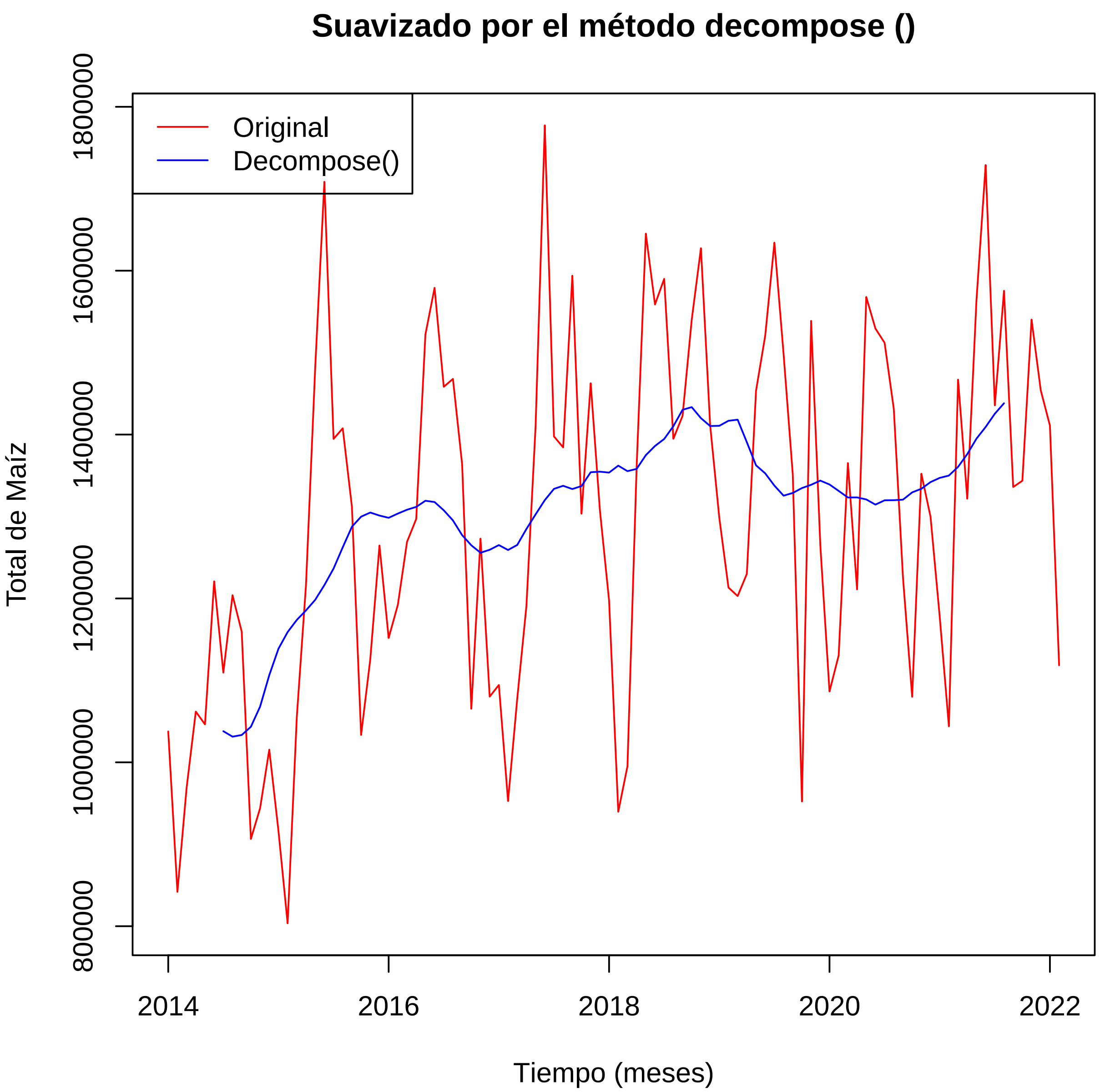
```
##            Aug       Sep       Oct       Nov       Dec
## 2014 1203979.7 1159457.2  906492.2  943968.5 1015504.7
## 2015 1407658.0 1311686.6 1033417.1 1125823.7 1264619.4
## 2016 1467896.1 1363641.1 1065388.9 1273013.8 1080231.7
## 2017 1384385.5 1593865.6 1303444.0 1462605.5 1307751.5
## 2018 1394899.8 1422756.9 1540333.0 1627415.7 1411960.7
## 2019 1498119.5 1351274.6  952244.0 1538732.2 1264958.0
## 2020 1431310.0 1225618.7 1079972.6 1352228.9 1299678.9
## 2021 1575463.4 1335967.8 1343687.9 1540361.6 1454103.5
## 2022
##
## $seasonal
##             Jan        Feb        Mar        Apr        May        Jun
## 2014 -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
## 2015 -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
## 2016 -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
## 2017 -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
## 2018 -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
## 2019 -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
## 2020 -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
## 2021 -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
## 2022 -175772.15 -272243.42
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2014  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
## 2015  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
## 2016  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
## 2017  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
## 2018  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
## 2019  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
## 2020  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
## 2021  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
## 2022
##
## $trend
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 2014      NA      NA      NA      NA      NA      NA 1037988 1031327 1033337
## 2015 1138585 1158956 1173785 1185417 1198283 1216240 1236430 1262437 1287532
## 2016 1298412 1303572 1308247 1311744 1319209 1317659 1307587 1295220 1277265
## 2017 1265148 1259145 1265258 1284770 1302588 1319968 1333739 1337485 1333503
## 2018 1353600 1362049 1355357 1358098 1374835 1386045 1394593 1410201 1430259
## 2019 1410682 1416825 1418147 1390665 1362466 1352646 1337693 1325417 1328732
## 2020 1339091 1331214 1323195 1323281 1320832 1314508 1319748 1319935 1320572
## 2021 1347133 1349959 1360564 1376150 1394977 1409250 1425411 1438238      NA
## 2022      NA      NA
##          Oct     Nov     Dec
## 2014 1043390 1067951 1106386
## 2015 1299784 1304840 1301153
## 2016 1264862 1255793 1259410
## 2017 1337159 1354025 1354693
## 2018 1433459 1420013 1410432
## 2019 1334702 1338680 1343816
## 2020 1329429 1333865 1342000
## 2021      NA      NA      NA
## 2022
```

```
##
## $random
##              Jan         Feb         Mar         Apr         May         Jun
## 2014          NA          NA          NA          NA          NA          NA
## 2015  -46488.712  -83217.463   -7960.125   88517.295   87099.295  194152.331
## 2016   29125.142  160894.418   71141.048   42664.294    7058.591  -36655.739
## 2017    5042.958  -34136.454  -77343.886  -36542.956  -87805.617  159329.321
## 2018   19573.031 -150114.041 -249933.313   60353.131   74504.307 -125258.223
## 2019   63451.243   68731.266 -104948.981 -103181.887 -104873.962 -130060.674
## 2020  -76834.059   71604.191  152289.435  -54800.023   51229.127  -83111.463
## 2021    6172.488  -33719.826  216797.913    3032.237  -27169.648   21646.538
## 2022          NA          NA
##              Jul         Aug         Sep         Oct         Nov         Dec
## 2014  -70866.764   54726.886   66540.144   29035.643 -159494.405  -23277.636
## 2015   15911.180   27295.777  -35425.393 -100432.875 -214528.018   31070.598
## 2016    8365.942   54750.344   26796.640  -33539.726  -18290.449 -111574.280
## 2017  -78357.812  -71024.904  200782.761  132219.060   73069.603   20662.268
## 2018   53051.591 -133226.783  -67082.052  272807.279  171891.352   69132.774
## 2019  154160.800   54776.752  -37036.863 -216524.154  164541.241  -11254.446
## 2020   49879.360   -6550.114 -154533.146  -83523.135  -17147.233   25282.814
## 2021 -132096.192   19300.145          NA          NA          NA          NA
## 2022
##
## $figure
##  [1] -175772.15 -272243.42 -110278.25  -57445.89  195855.23  298055.28
##  [7]  142349.92  117925.57   59579.84 -165933.68   35511.41  -67603.86
##
## $type
## [1] "additive"
##
## attr(,"class")
## [1] "decomposed.ts"
```

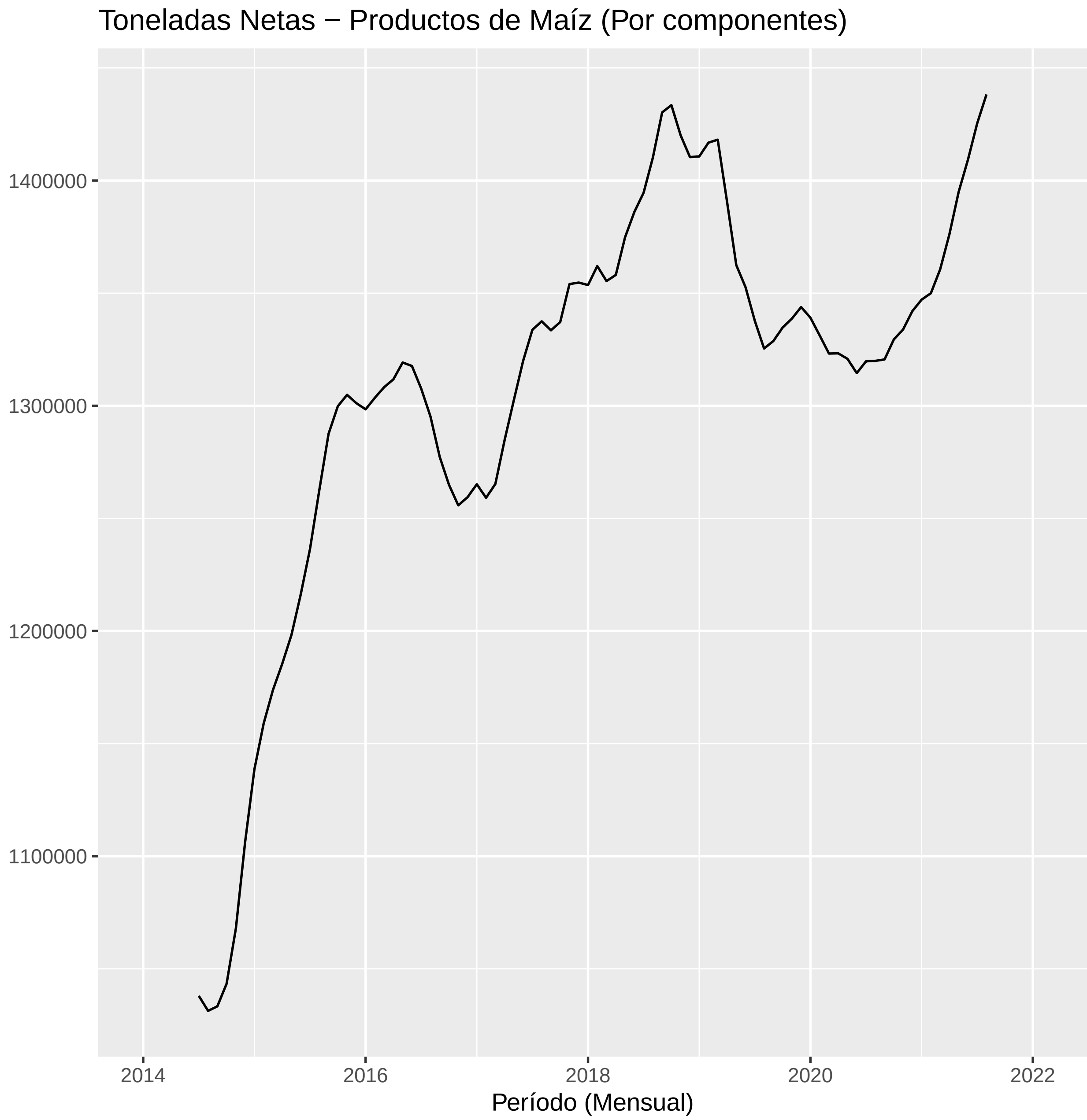
El resultado anterior muestra los componentes de una descomposición dada la función *decompose()* de *R*.
Podemos graficar todos los componentes en una sola figura de la siguiente manera:



La columna **trend** (que contiene la tendencia-ciclo T_t) que sigue el movimiento general de la serie, ignorando cualquier estacionalidad y fluctuaciones aleatorias. Como se muestra en la siguiente gráfica



Podemos comprender a la **tendencia** de la serie de tiempo del maiz como el comportamiento a largo plazo de la serie. Visualizando únicamente la tendencia, tendremos lo siguiente:



Podemos observar que la tendencia tuvo un comportamiento sostenido en la gran parte del período, esta también tuvo un importante cambio alrededor de Julio-2016 a Diciembre-2016 y Febrero-2019 a Octubre-2019, donde prácticamente paso de tener un comportamiento creciente a uno decreciente.

2.4. Análisis de correlograma

El análisis del correlograma es una opción que nos ayudará bastante a analizar la interdependencia de los valores de observación. La idea con la que debemos conservar al manejar correlogramas es que sirven para describir la presencia o ausencia de correlación en los datos de las series temporales, indicando si las observaciones pasadas influyen en las actuales.

Ahora veremos la correlación que tienen los datos , es decir, una correlación mide el nivel de semejanza entre una secuencia de hace varios períodos y datos actuales. La secuencia de hace varios períodos atrás se llama retraso.º "lag.en inglés, porque es una versión retrasada de la original.

Dada una serie de tiempo Y_t, se define su **función de auto-correlación de retraso k** como:

$$\varphi_k(t) = corr(Y_t, Y_{t-k})$$

Así como la **correlación** mide el alcance de una relación lineal entre dos variables, la **auto-correlación** mide la relación lineal entre los *valores rezagados* de una serie de tiempo. Hay varios coeficiente de autocorrelación, correspondientes a cada panel en el gráfico de retardo, es decir, $\varphi_k(t) = corr(y_t, y_{t-k})$.

Por ejemplo:

- $\varphi_1(t)$ mide la relación entre y_1 y y_{t-1}.
- $\varphi_2(t)$ mide la relación entre y_t y y_{t-2}, y así sucesivamente.

El valor de $\varphi_k(t)$ se puede escribir como

$$\varphi_k = \frac{\sum\limits_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum\limits_{t=1}^{T} (y_t - \bar{y})^2}$$

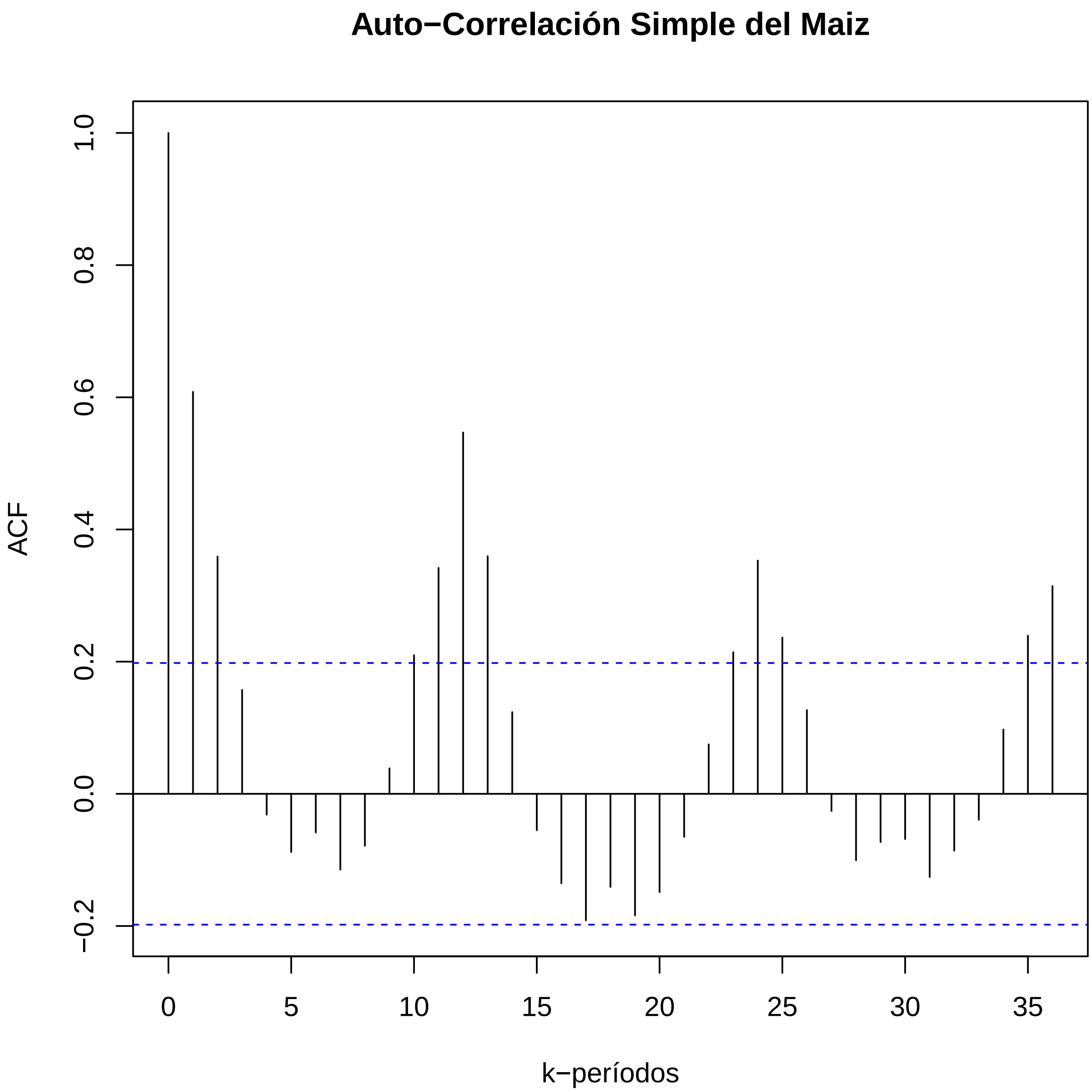
donde T es la duración de la serie temporal. Los coeficientes de autocorrelación componen la *función de autocorrelación* o *ACF*

Las dos funciones de **auto-correlación** que podemos manejar en los pronósticos de Series de Tiempo son:

1. **Función de Auto-correlación simple:** Mide la autocorrelación entre dos variables separadas por k períodos, o , en otras palabras, el grado de asociación lineal que existe entre dos variables del mismo proceso aleatorio.
2. **Función de Auto-Correlación parcial:** Mide la auto-correlación entre dos variables separadas k períodos cuando no se considera la precedencia de la dependencia creada por los retardos intermedios que hay entre ambas. En otras palabras, esta encuentra la auto-correlación que existe entre dos variables separadas k períodos descontando los posibles efectos debidos a las variables intermedias.

2.4.1. Autocorrelación simple

En seguida se darán los valores y el gráfico de las auto-correlaciones simples del Maiz




```
##
## Autocorrelations of series 'ts(Maiz_ts)', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.608  0.359  0.157 -0.032 -0.088 -0.058 -0.115 -0.079  0.039  0.210
##     11     12     13     14     15     16     17     18     19     20     21
##  0.342  0.547  0.360  0.124 -0.055 -0.135 -0.191 -0.141 -0.184 -0.149 -0.065
##     22     23     24     25     26     27     28     29     30     31     32
##  0.075  0.214  0.353  0.236  0.127 -0.026 -0.101 -0.073 -0.068 -0.126 -0.086
##     33     34     35     36
## -0.039  0.097  0.239  0.314
```

Dado los valores y la gráfica de auto-correlación simple, tenemos que $\varphi_1 = 0.608$ y $\varphi_{12} = 0.547$, es decir , es conveniente ”predecir” el futuro ya sea con 1 o 12 retrasos (Entre más cercano a 1 es mejor). Lo anterior lo podremos expresar de la siguiente manera:

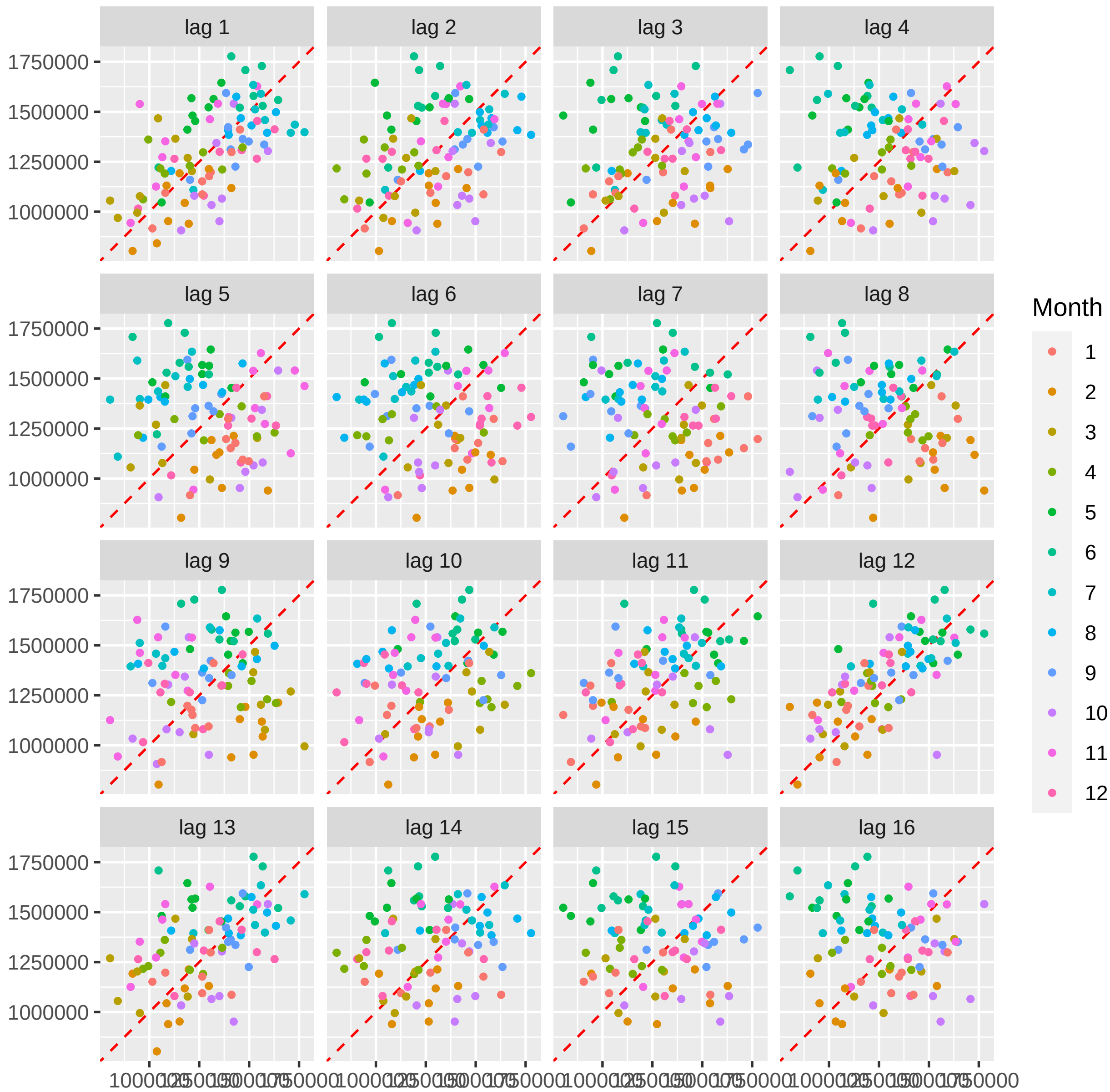
$$\varphi_1(t) = corr(X_t, X_{t-1}) = 0.608$$
$$\varphi_{12}(t) = corr(X_t, X_{t-12}) = 0.547$$

Tendencia y estacionalidad en parcelas ACF

- Cuando los **datos tienen una tendencia**, las autocorrelaciones para pequeños retrasos tienden a ser grandes y positivas porque las observaciones cercanas en el tiempo también tienen un valor cercano. Entonces, el ACF de una serie de tiempo con tendencia tiende a tener valores positivos que disminuyen lentamente a medida que aumentan los retrasos.
- Cuando los **datos son estacionales**, las auto-correlaciones serán mayores para los retrasos estacionales (en múltiplos de período estacional) que para otros retrasos.
- Cuando los **datos son de tendencia y estacionales**, se ve una combinación de estos efectos.

2.4.2. Gráficas de Retraso

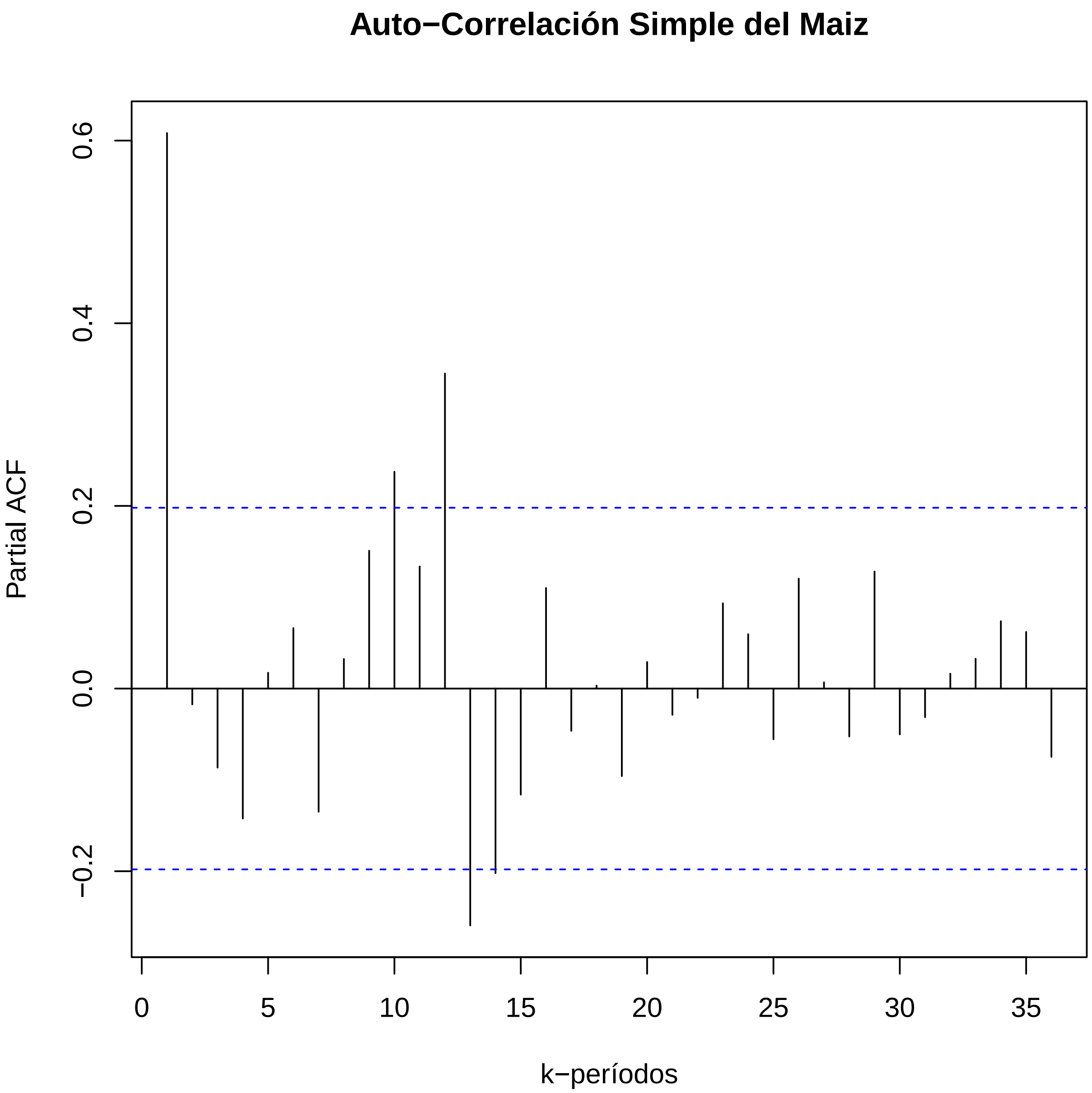
Un gráfico de retraso es un tipo especial de gráfico de dispersión en el que el eje x representa el conjunto de datos con algunas unidades de tiempo atrás o adelante en comparación con el eje y. La diferencia entre estas unidades de tiempo se llama retraso y se representa por k



La relación es fuertemente positiva en el rezago 1 y 12 (φ_1 y φ_{12}) lo que refleja la fuerte estacionalidad de los datos.

Auto-correlación parcial

En seguida se darán los valores y la gráfica de *auto-correlación parcial* del Maiz.




```
##
## Partial autocorrelations of series 'ts(Maiz_ts)', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.608 -0.017 -0.087 -0.142  0.017  0.066 -0.135  0.032  0.151  0.237  0.134
##     12     13     14     15     16     17     18     19     20     21     22
##  0.345 -0.259 -0.202 -0.116  0.110 -0.046  0.003 -0.096  0.029 -0.029 -0.010
##     23     24     25     26     27     28     29     30     31     32     33
##  0.093  0.060 -0.056  0.120  0.007 -0.053  0.128 -0.050 -0.031  0.016  0.033
##     34     35     36
##  0.074  0.062 -0.075
```

En este caso, es solamente conveniente tomar $k = 1$ retraso para poder predecir mi futuro, ya que vemos que $\varphi_1 = corr(Y_t, Y_{t-1}) = 0.608$, donde los demás valores ya están muy alejados de este valor, teniendo en cuenta que este tipo de auto-correlación no toma en cuenta los efectos intermedios que hay entre el intervalo $(t, t-k)$

2.5. Técnicas de suavizado

2.5.1. Suavizado de la media móvil centrado

El método clásico de descomposición de series de tiempo se originó en la década de 1920 y fue amplia-mente utilizado hasta la década de 1950. Todavía constituye la base de muchos métodos de descomposición de series temporales, por lo que es importante entender como funcionan. El primer paso en una descomposición clásica es usar un método de promedio móvil para estimar el ciclo de tendencia, por lo que comenzaremos analizando los promedios móviles.

Una media móvil de orden m se puede escribir como:

$$\hat{T}_t = \frac{1}{m}\sum_{j=-k}^{k} y_{t+j}$$

donde $m = 2k+1$. Es decir, la estimación de la tendencia-ciclo en el momento t se obtiene promediando los valores de la serie temporal dentro de k períodos de t. También es probable que las observaciones que están cercanas en el tiempo tengan un valor cercano. Por lo tanto, el promedio elimina parte de la aleatoriedad en los datos, dejando un componente suave de ciclo-tendencia. A esto le llamamos m-**MA**, lo que significa un promedio móvil de orden m.

El orden de la media móvil determina la suavidad de la estimación del ciclo de tendencia. En general, un pedido más grande significa una curva más suave.

```
##                MAÍZ      5-MA
## ene. 2014 1037733.6        NA
## feb. 2014  841936.6        NA
## mar. 2014  968867.5  991358.8
## abr. 2014 1061894.8 1027990.1
## may. 2014 1046361.4 1081497.0
## jun. 2014 1220890.4 1128519.4
## jul. 2014 1109470.9 1148031.9
## ago. 2014 1203979.7 1120058.1
## sep. 2014 1159457.2 1064673.7
## oct. 2014  906492.2 1045880.5
## nov. 2014  943968.5  988349.4
## dic. 2014 1015504.7  917157.0
## ene. 2015  916324.4  946967.9
## feb. 2015  803495.1 1001471.9
## mar. 2015 1055547.1 1094618.4
## abr. 2015 1216488.3 1253043.0
## may. 2015 1481237.2 1371282.1
## jun. 2015 1708447.4 1441704.3
## jul. 2015 1394690.7 1460744.0
## ago. 2015 1407658.0 1371180.0
## sep. 2015 1311686.6 1254655.2
## oct. 2015 1033417.1 1228641.0
## nov. 2015 1125823.7 1177462.4
## dic. 2015 1264619.4 1153569.7
## ene. 2016 1151765.0 1200708.3
## feb. 2016 1192223.4 1234936.1
## mar. 2016 1269109.9 1286436.8
## abr. 2016 1296962.5 1371895.5
## may. 2016 1522123.0 1425111.4
## jun. 2016 1579058.8 1464868.6
## jul. 2016 1458302.9 1478204.4
## ago. 2016 1467896.1 1386857.5
```

```
## sep. 2016 1363641.1 1325648.5
## oct. 2016 1065388.9 1250034.3
## nov. 2016 1273013.8 1175338.9
## dic. 2016 1080231.7 1093163.7
## ene. 2017 1094419.1 1095613.0
## feb. 2017  952764.9 1079166.4
## mar. 2017 1077635.7 1145247.6
## abr. 2017 1190780.7 1281834.3
## may. 2017 1410637.8 1370827.5
## jun. 2017 1777352.4 1432177.4
## jul. 2017 1397730.8 1512794.4
## ago. 2017 1384385.5 1491355.6
## sep. 2017 1593865.6 1428406.3
## oct. 2017 1303444.0 1410410.4
## nov. 2017 1462605.5 1373013.4
## dic. 2017 1307751.5 1242178.5
## ene. 2018 1197400.4 1180518.8
## feb. 2018  939691.1 1160198.8
## mar. 2018  995145.6 1227687.5
## abr. 2018 1361005.3 1299975.7
## may. 2018 1645195.0 1430036.5
## jun. 2018 1558841.7 1509987.3
## jul. 2018 1589994.9 1522337.7
## ago. 2018 1394899.8 1501365.3
## sep. 2018 1422756.9 1515080.1
## oct. 2018 1540333.0 1479473.2
## nov. 2018 1627415.7 1460165.5
## dic. 2018 1411960.7 1418276.7
## ene. 2019 1298361.2 1350794.1
## feb. 2019 1213312.8 1271318.4
## mar. 2019 1202920.1 1279615.8
## abr. 2019 1230037.4 1324071.7
## may. 2019 1453447.6 1408249.8
## jun. 2019 1520640.7 1467289.7
## jul. 2019 1634203.5 1491537.2
## ago. 2019 1498119.5 1391296.4
## sep. 2019 1351274.6 1394914.8
## oct. 2019  952244.0 1321065.7
## nov. 2019 1538732.2 1238738.6
## dic. 2019 1264958.0 1194598.7
## ene. 2020 1086484.4 1277191.1
## feb. 2020 1130574.9 1211651.7
## mar. 2020 1365206.0 1272243.4
## abr. 2020 1211035.2 1360836.9
## may. 2020 1567916.6 1437117.4
## jun. 2020 1529451.7 1450338.2
## jul. 2020 1511977.6 1453254.9
## ago. 2020 1431310.0 1355666.1
## sep. 2020 1225618.7 1320221.6
## oct. 2020 1079972.6 1277761.8
## nov. 2020 1352228.9 1227006.4
## dic. 2020 1299678.9 1190681.9
## ene. 2021 1177533.1 1268104.1
## feb. 2021 1043996.2 1262005.5
```

```
## mar. 2021 1467083.3 1314802.2
## abr. 2021 1321736.0 1425085.9
## may. 2021 1563662.3 1503419.6
## jun. 2021 1728951.7 1525095.6
## jul. 2021 1435664.5 1527941.9
## ago. 2021 1575463.4 1483947.1
## sep. 2021 1335967.8 1446229.0
## oct. 2021 1343687.9 1449916.9
## nov. 2021 1540361.6 1417017.7
## dic. 2021 1454103.5 1373506.0
## ene. 2022 1410967.6        NA
## feb. 2022 1118409.3        NA
```

En la última columna de esta tabla, se muestra una media móvil de orden $m = 5$, que proporciona una estimación de la tendencia-ciclo.

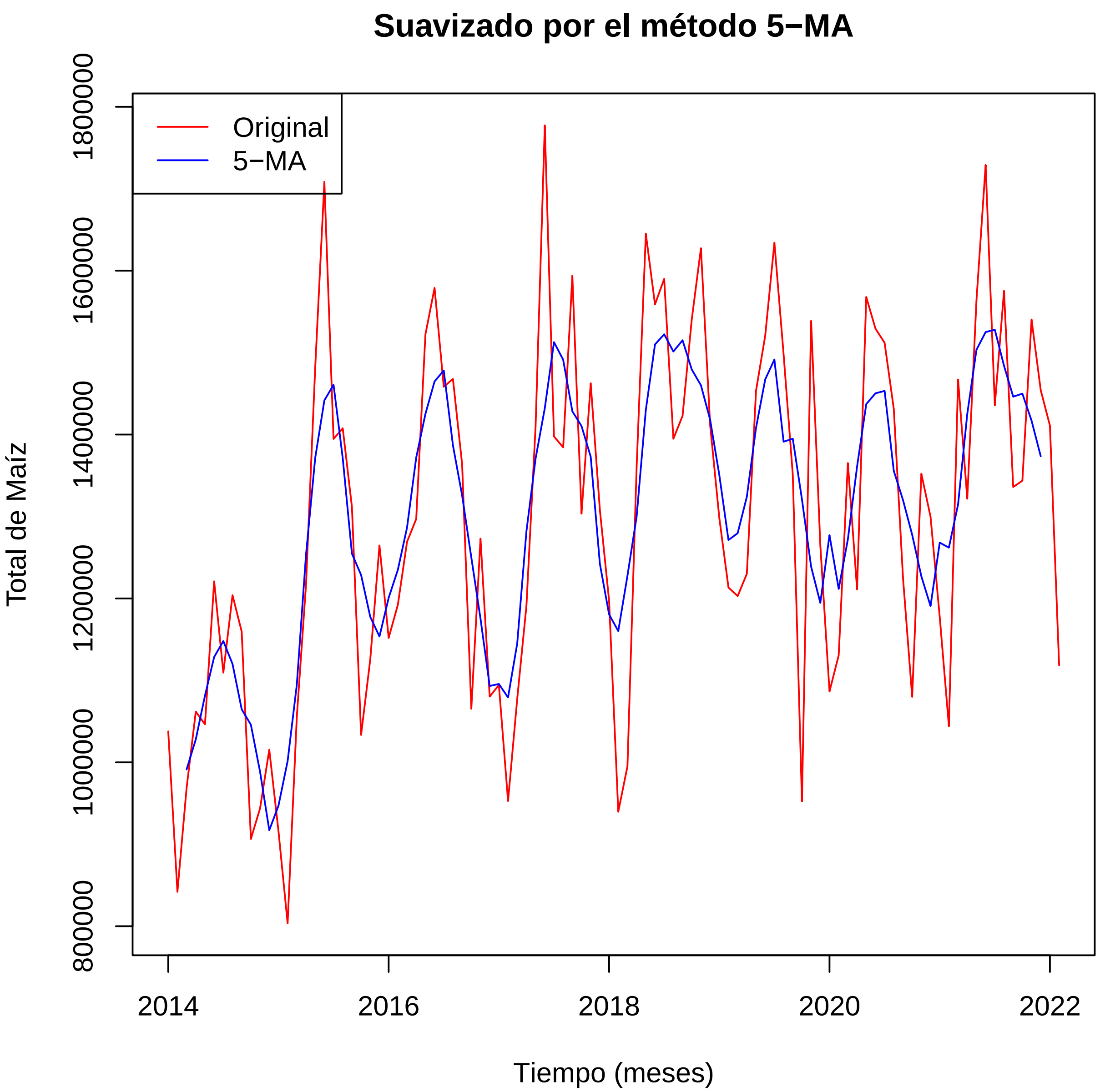
- El primer valor de esta columna es el promedio de las primero 5 observaciones, $ene.2014 - mayo.2014$, es decir:

$$\frac{y_a + y_2 + \cdots + y_5}{5}$$

- El segundo valor en la columna **5-MA** es el promedio de los valores para $feb.2014 - jun.2014$ y así sucesivamente.

Cada valor en la columna **5-MA** es el promedio de las observaciones en la ventana de 5 meses centrada en el mes correspondiente.

La columna **5-MA** contiene los valores de $\hat{T}_t$ con $m = 2k + 1 = 5$ y $k = 2$. No hay valores ni para los primeros dos años ni para los dos últimos años, porque no tenemos observaciones en cada lado.



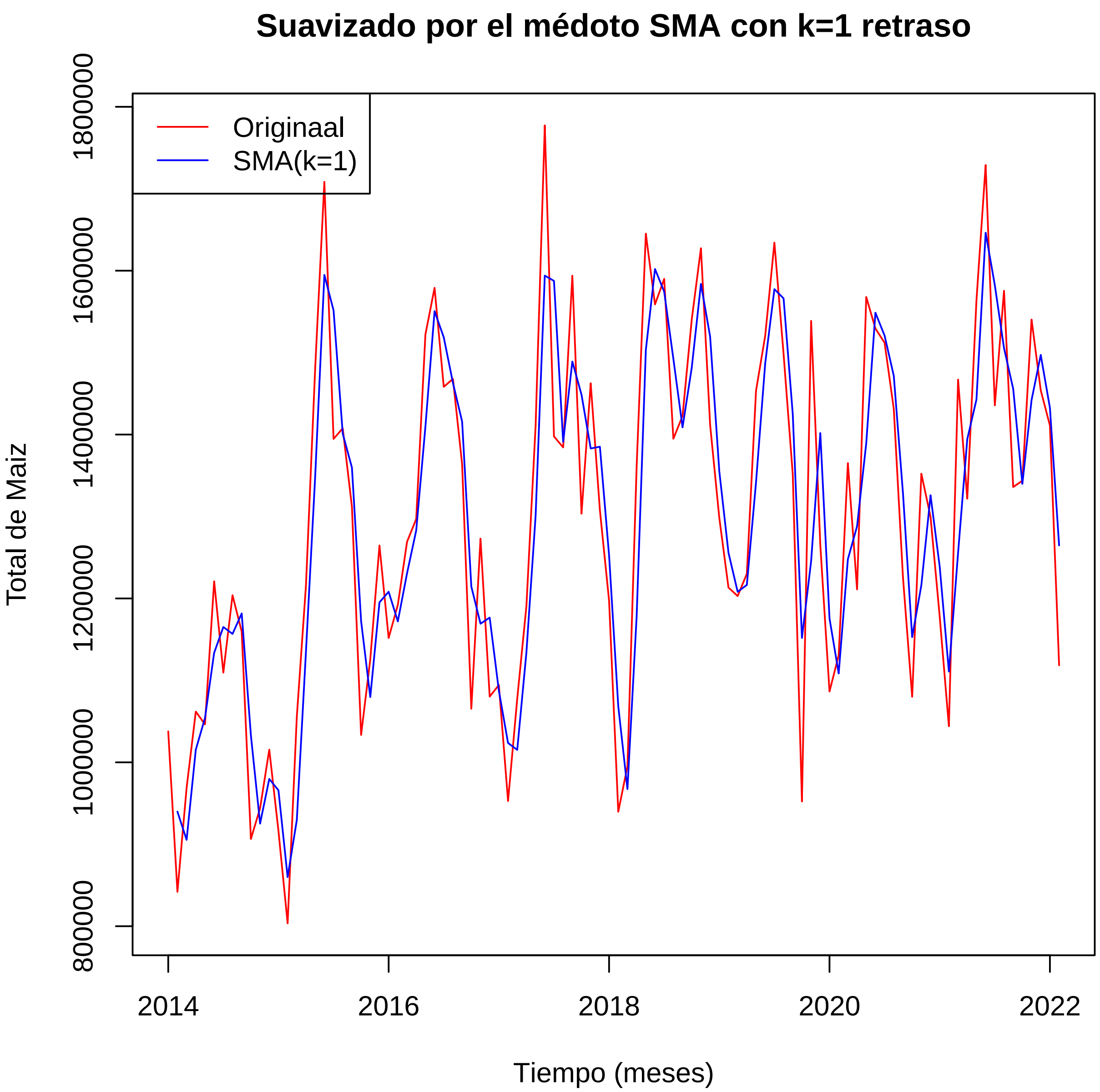
2.5.2. Suavizado de la media móvil hacía atrás

El método de **Promedio Móvil Simple** utiliza información histórica del desempeño de la variable que se desea pronosticar para poder generar un pronóstico de la misma a futuro. Es decir, se considera válida la premisa que el *pasado* es de utilidad para predecir el *futuro*. Su notación matemática es de la siguiente forma:

$$SMA = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

donde x_i son los valores de la variable, por tanto SMA es un promedio aritmético de k períodos atrás.

Dado que en la auto-correlación simple como en la parcial φ_1 dio casi cercano a 1, entonces para aplicar el método de medias móviles hacemos $k = 1$. Es decir: $Y_t = \frac{X_t + X_{t-1}}{2}$ para $t > 1$



Tenemos que $X_1 = 1037733.6$ (Valor de toneladas netas en Enero/2014) y $X_2 = 841936.6$ (Valor de toneladas netas en Febrero/2014), entonces dado lo anterior, tendremos que: $Y_2 = \frac{X_2+X_1}{2}$, es decir, El valor predictor para Febrero/2014 (Y_2) es $939, 835.1$

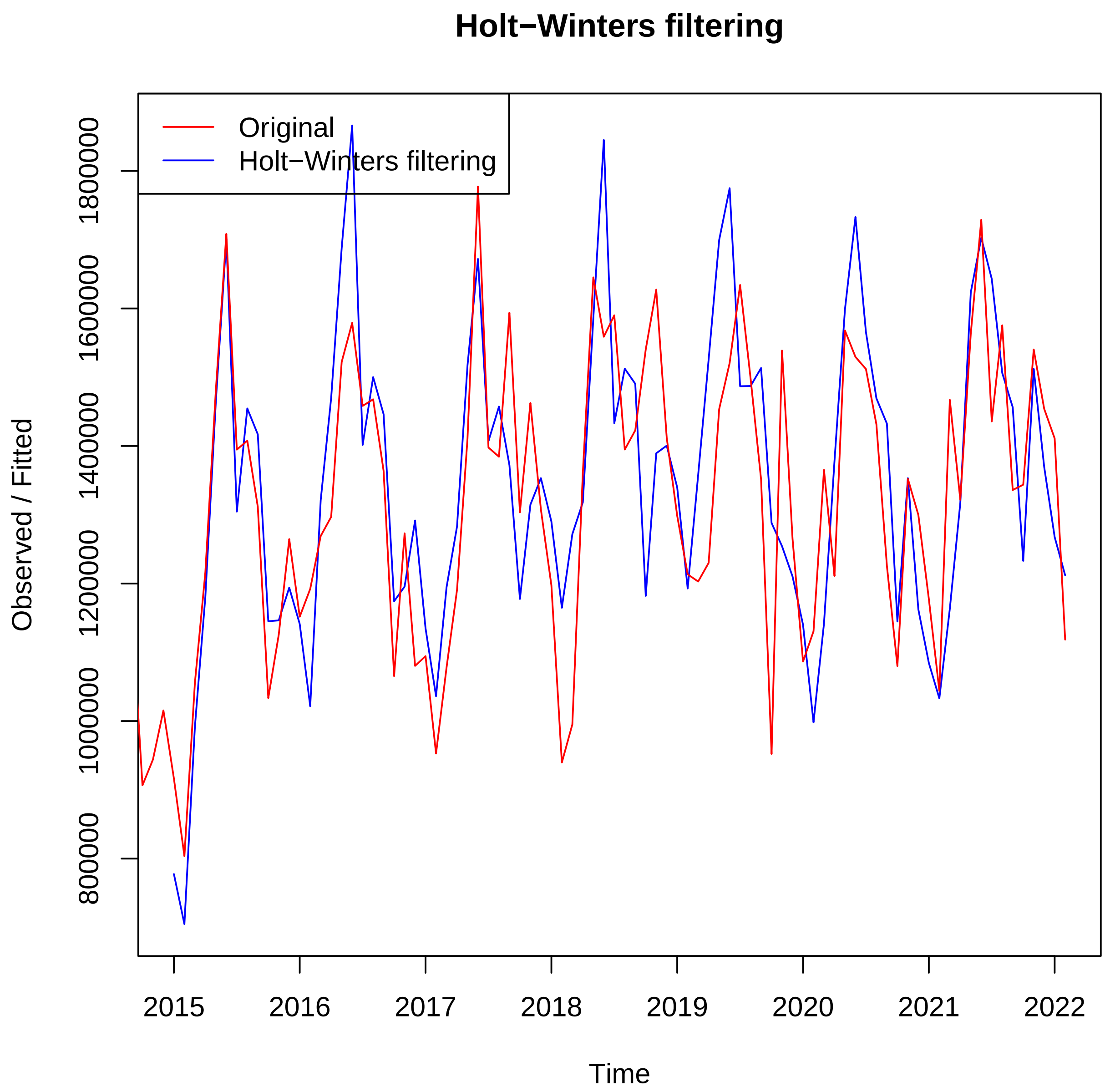
2.5.3. Suavizado por Método de Holt-Winters

En este método se toma en cuenta que contribuyen, como la **tendencia** y la **estacionalidad**, es decir, este método lleva a cabo un suavizado exponencial en la presencia de tendencias y estacionalidad. Utilizando el **packages forecast** para llevar a cabo este método.

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = Maiz_ts)
##
## Smoothing parameters:
##  alpha: 0.2890066
##  beta : 0.02799386
```

```
##  gamma: 0.374021
##
## Coefficients:
##            [,1]
## a   1517860.955
## b      5992.591
## s1  -131543.289
## s2  -151243.472
## s3   130452.879
## s4   244259.239
## s5   108864.905
## s6   101431.949
## s7   -23646.523
## s8  -154553.453
## s9    65965.856
## s10  -74598.762
## s11 -190997.612
## s12 -357811.540
```

Con lo anterior, tenemos que la gráfica a través de este método nos queda de la siguiente manera:



2.5.4. Suavizado por la función ETS() de R

El suavizado en R hace este procedimiento de una manera más sencilla con la función *ets()* de la paquetería TTR. El código nos quedaría de la siguiente manera:

```
Suavizado_Maiz_R <- ets(Maiz_ts, model = "ZZZ") #Suavizado por la función ets() en forma automat
```

Cuando se pone **model="ZZZ"** le indicamos a R que nos calcule cual es el método más conveniente. En model = "Errores, Tendencia, Estacionalida", los cuales podremos utilizar parámetros de la siguiente manera:

- N = No hacer caso.
- A = Tipo aditiva.
- M = Tipo Multiplicativa.
- Z = Que R calcule el tipo que se va a requerir.

Coeficientes del suavizado

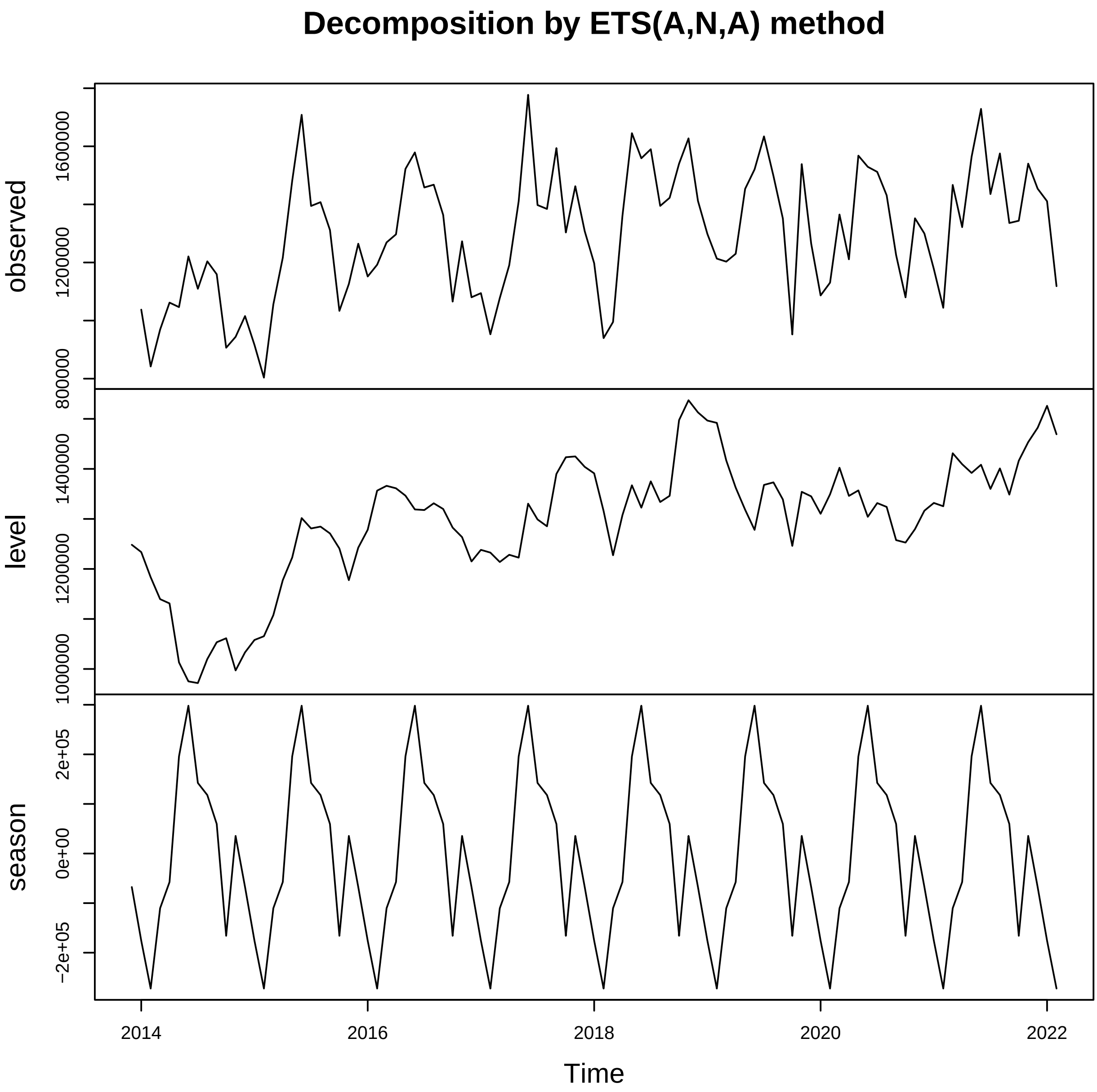
```
## ETS(A,N,A)
##
## Call:
##  ets(y = Maiz_ts, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 0.4203
##     gamma = 1e-04
##
##   Initial states:
##     l = 1248315.1426
##     s = -67603.11 35512 -165933.2 59579.95 117925.9 142350.2
##            298054.5 195854.2 -57446.23 -110278.5 -272243.4 -175772.1
##
##   sigma:  127565.4
##
##      AIC     AICc      BIC
## 2768.471 2774.325 2807.246
```

Lo anterior, tenemos que R nos da un modelo de la forma:

- Error de tipo aditivo.
- Tendencia no hacer caso.
- Estacionalidad de tipo aditivo

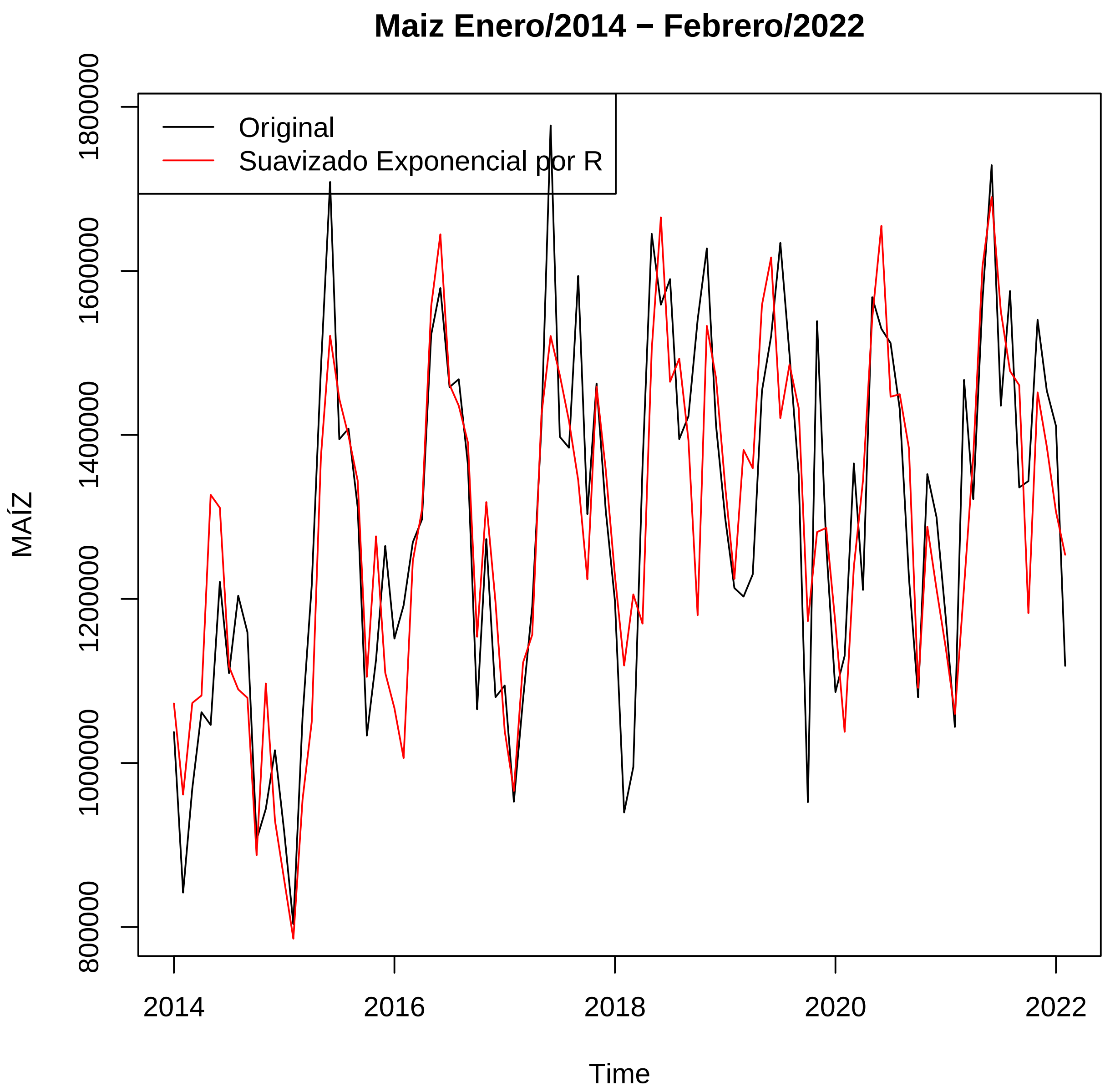
Descomposición de la serie por el método ETS()

La descomposición del modelo anterior queda de la siguiente manera:



Gráfica del suavizado ETS() vs original

Haciendo una gráfica de la serie maíz con el suavizado exponencial por R, tendremos lo siguiente:



2.6. Pruebas de Estacionariedad

Recordando...

P-valor

Con base en los datos proporcionados es que se puede realizar la prueba, y así determinar si se puede rechazar o no la hipótesis nula; la forma de determinar esto es con base en los diferentes estadísticos, o bien, usando el p-valor de la prueba. La forma más sencilla de saber que es el p-valor, es verlo como la probabilidad de rechazar la hipótesis nula cuando esta sea verdadera. Por lo tanto:

- Si p-valor $< \alpha$, entonces se puede rechazar la hipótesis nula (Se acepta H_a).
- Si p-valor$> \alpha$, hay evidencia para que **NO** se rechace la hipótesis nula (Se acepta H_0)

Prueba de Dickey-Fuller Aumentada para raíz unitaria

$$H_0 = \textit{No estacionaria (Raíz Unitaria).}$$
$$H_a = \textit{Es estacionaria.}$$

Prueba de Phillips-Perron

Es una modificación de la prueba de Dickey-Fuller. Esta prueba corrige la auto-correlación y heterocedasticidad (Varianza NO constante) en los errores para comprobar la existencia de la serie.

$$H_0 = \textit{No estacionaria (Raíz Unitaria).}$$
$$H_a = \textit{Es estacionaria.}$$

Una manera de corregir la NO estacionariedad es utilizando diferencias

Prueba Ljung Box

$$H_0 = \textit{Ruido Blanco).}$$
$$H_a = \textit{NO hay Ruido Blanco.}$$

Ruido Blanco

El ruido blanco cumple las siguientes características:

$$E[\varepsilon_t] = 0, \textit{es decir, media igual a} 0.$$
$$Var[\varepsilon_t] = c, \textit{ es decir, varianza constante.}$$
$$Cov(\varepsilon_t, \varepsilon_{t-k}) = 0, \textit{ es decir, no hay correlación}$$

Siguiendo la **metodología de Box-Jenkins**, lo primero sería realizar las pruebas de estacionariedad (*Prueba de Dickey-Fuller aumentada* y *Phillips-Perron*) para comprobar que la serie sea o no estacionaria.

2.6.1. Prueba de Dickey-Fuller aumentada

Para realizar la prueba de Dickey-Fuller aumentada sobre la serie que estamos ocupando (MAÍZ), utilizamos el siguiente código:

```
## Warning in adf.test(Maiz_ts, alternative = "stationary", k = 0): p-value smaller than
printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  Maiz_ts
## Dickey-Fuller = -5.1793, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

Con un valor del p-valor del 0.01, menor al $\alpha = 0.05$ nivel de significancia de la prueba, se acepta la hipótesis alternativa (H_a) y **rechazamos la hipótesis nula** (H_0), lo que contrasta que la serie presenta un**comportamiento Estacionario** ($p - valor < \alpha$) .
Para comparar los resultados de la prueba de Dickey-Fuller aumentada utilizaremos la *Prueba Phillips-Perron* y comprobaremos si esta igual señala la existencia de un comportamiento ESTACIONARIO.

2.6.2. Prueba de Phillips-Perron

.

```
## Warning in pp.test(Maiz_ts, alternative = "stationary"): p-value smaller than printed
p-value

##
##  Phillips-Perron Unit Root Test
##
## data:  Maiz_ts
## Dickey-Fuller Z(alpha) = -46.443, Truncation lag parameter = 3, p-value
## = 0.01
## alternative hypothesis: stationary
```

Con un p-valor menor al 0.05 nivel de significancia, al igual que la *prueba Dickey-Fuller Aumentada*, la prueba de Phillips-Perron contrasta que la serie es una **serie estacionaria**.

Conclusión

Dada estas dos pruebas, tenemos como conclusión que $d = 0$, entonces tendremos que encontrar un proceso de la forma $ARMA(p, q) = ARIMA(p, 0, q)$

2.7. ARIMA(p,d,q)

Dada la prueba de estacionariedad, podemos confirmar que la serie *Maiz_ ts* se puede considerar como un **proceso estacionario**, y que, a su vez este se puede modelar a partir de un proceso $AR(p)$ y $MA(q)$; es decir, un **proceso** $ARMA(p, q)$ **estacionario**.
Dentro de la metodología estándar de estimación de modelos ARMA, el proceso se puede realizar a partir de candidatos posibles que puedan modelar adecuadamente el *comportamiento estocástico de la serie*. Por lo cual propongo 5 candidatos a modelos $ARIMA(p, d, q)$, los cuales son:

- $ARIMA(1, 0, 1)$
- $ARIMA(1, 0, 0)$
- $ARIMA(2, 0, 1)$
- $ARIMA(1, 0, 2)$
- $ARIMA(0, 0, 1)$

El objetivo de esto será comparar y ver cual de estos candidatos tiene el menor **criterio de información**.
El código que ocuparemos para estimar los procesos ARIMA seleccionados sobre la serie es el siguiente:

```
ModA_ARIMA =arima(Maiz_ts , order = c(1,0,1))
ModB_ARIMA =arima(Maiz_ts , order = c(1,0,0))
ModC_ARIMA =arima(Maiz_ts , order = c(2,0,1))
ModD_ARIMA =arima(Maiz_ts , order = c(1,0,2))
ModE_ARIMA =arima(Maiz_ts , order = c(0,0,1))
```

2.7.1. Criterios de Información.

El *Criterio de Información de Akaike(AIC)* son medidas de la calidad relativa de un modelo que representan el ajuste y el número de términos en el modelo. Este criterio se utiliza para comparar diferentes modelos. Sin embargo, el modelo con el valor más pequeño para un conjunto de predictores **NO necesariamente ajusta los datos adecuadamente.** Utilice también pruebas y gráficas de residuos para evaluar que tan bien se ajusta el modelo a los datos.

El código que ocuparemos para comparar los *criterios de información* de los modelos propuestos es:

```
ModA_ARIMA$aic

## [1] 2650.104

ModB_ARIMA$aic

## [1] 2648.111

ModC_ARIMA$aic

## [1] 2651.426

ModD_ARIMA$aic

## [1] 2651.483

ModE_ARIMA$aic

## [1] 2660.408
```

Podemos observar que al hacer la comparación de modelos utilizando el **Criterio de Información de Akaike**, el modelo que cuenta con el menor valor es el modelo $ARIMA(1, 0, 0)$, por lo cual ese es el modelo que ocuparemos para realizar las pruebas sobre los supuestos y, en el caso de que los supuestos se cumplan, el modelo que podremos ocupar para realizar el pronóstico de la serie. Podemos ocupar la función *summary()* sobre el modelo seleccionado para observar las estimaciones de los parámetros del modelo:

```
##
## Call:
## arima(x = Maiz_ts, order = c(1, 0, 0))
##
## Coefficients:
##          ar1   intercept
##       0.6152  1289609.65
## s.e.  0.0796    44640.69
##
## sigma^2 estimated as 2.979e+10:  log likelihood = -1321.06,  aic = 2648.11
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set 2136.598 172610.3 139227.9 -1.721672 11.07217 0.8852303 0.00743361
```

Con lo anterior, calculamos los parámetros a través de *mínimos cuadrados*.Con la ayuda de R obtuvimos la estimación de esos parámetros así que tenemos que el modelo matemáticamente de $ARIMA(1, 0, 0) = AR(1)$ esta dado de la siguiente manera:

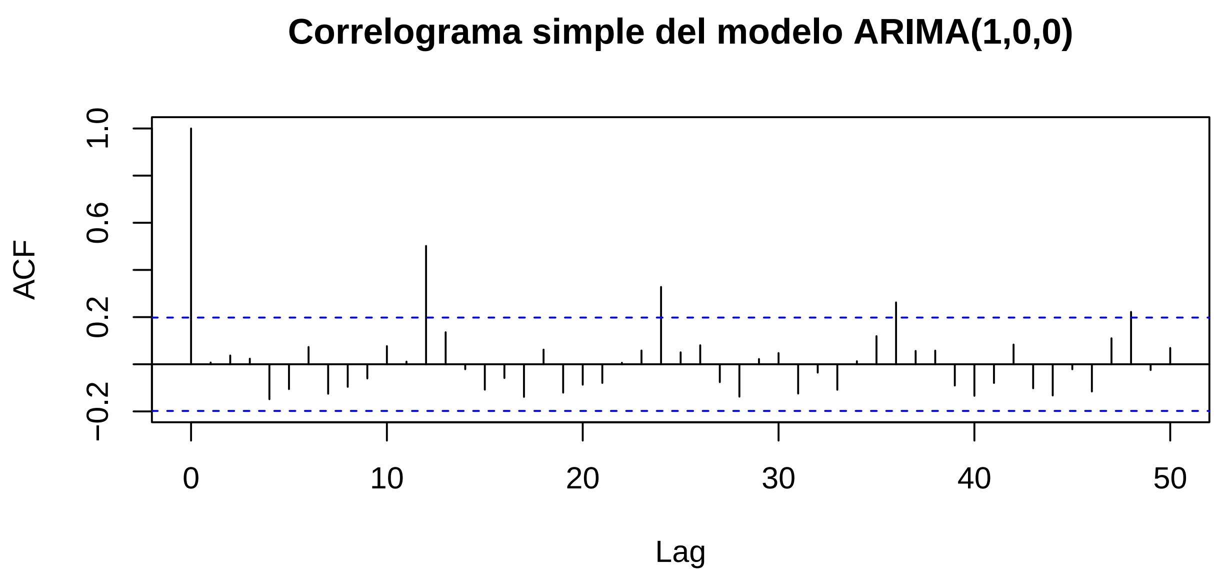
$$Y_t = \Phi_0 + \Phi_1 Y_{t-1} + \varepsilon_t$$
$$Y_t = 1289609.65 + 0.6152 Y_{t-1} + \varepsilon_t$$

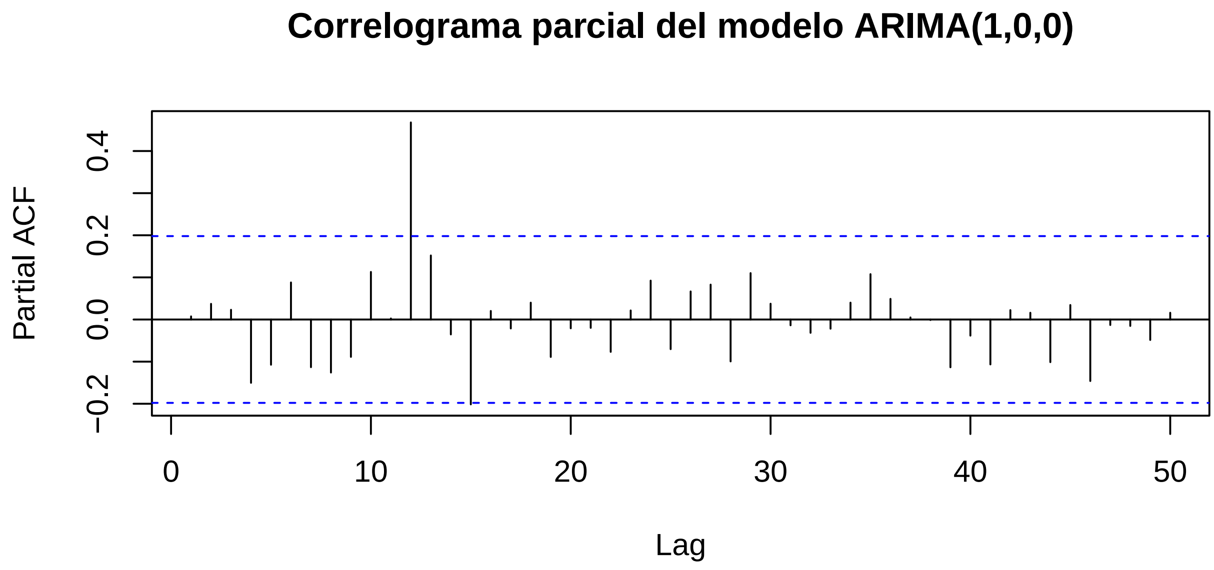
2.8. SARIMA(p,d,q)(P,D,Q)[m]

Una manera de poder obtener de forma automática un modelo adecuado es usando la función *auto.arima()*, con lo cual obtenemos lo siguiente:

```
## Series: Maiz_ts
## ARIMA(1,0,1)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1      ma1     sma1     drift
##       0.7384  -0.3824  -0.7926  3268.075
## s.e.  0.1457   0.1997   0.2065  1071.698
##
## sigma^2 = 1.704e+10:  log likelihood = -1138.93
## AIC=2287.86   AICc=2288.61   BIC=2300.13
```

2.8.1. Correlograma ACF y $PACF$ del modelo ARIMA(1,0,0)





Tanto en el Correlograma Simple, como el en Correlograma Parcial existen valores signficativo φ. Este

tipo de comportamiento en los correlogramas indican que aún hay factores que no están considerando en los modelos y afectan de manera importante el comportamiento de la serie temporal.
Es importante observar que, de acuerdo con el correlograma simple, los valores fuera del intervalo representan un comportamiento similar, ya que se repiten cada determinado tiempo de acuerdo al rezago de la serie, en este caso, cada múltiplo del rezago 12. **Este tipo de comportamiento es bastante común en las series temporales que presentan un comportamiento estacional;** es decir, que tiene un efecto temporal que se repite cada determinado período.
Para poder estimar modelos que consideren el comportamiento estacional de las series podemos ocupar los modelos $SARIMA$. Para incorporar el comportamiento estacional de la serie, podemos anexar un factor de diferenciación adicional al de integración, con el fin de diferencial de acuerdo con el componente estacional. Para empezar a estimar modelos $SARIMA$, propongo realizar 1 diferencia estacional que se le deberá aplicar a la serie (lo que significa que debemos de diferenciar sobre 12 unidades, ya que la serie es una serie mensual).
De acuerdo a lo anterior, propongo 9 propuestas de modelos SARIMA en donde únicamente variaremos el número de valores Autorregresivos y Medias Móviles, así como el orden de integración normal del modelo.

```
ndiffs(Maiz_ts) # Nos da el número de diferenciación ordinaria

## [1] 1

nsdiffs(Maiz_ts) #Nos da el número de diferenciación estacional

## [1] 1
```

Modelos SARIMA(p,d,q)(P,D,Q) propuestos

```
SARIMA_1<- arima(Maiz_ts,order=c(1,1,1), seasonal = list(order=c(0,1,1)))
SARIMA_2<- arima(Maiz_ts,order=c(0,1,1), seasonal = list(order=c(0,1,1)))
SARIMA_3<- arima(Maiz_ts,order=c(1,1,0), seasonal = list(order=c(0,1,1)))
SARIMA_4<- arima(Maiz_ts,order=c(1,1,1), seasonal = list(order=c(1,1,1)))
SARIMA_5<- arima(Maiz_ts,order=c(0,1,1), seasonal = list(order=c(1,1,1)))
SARIMA_6<- arima(Maiz_ts,order=c(1,1,0), seasonal = list(order=c(1,1,1)))
SARIMA_7<- arima(Maiz_ts,order=c(1,1,1), seasonal = list(order=c(1,1,0)))
SARIMA_8<- arima(Maiz_ts,order=c(0,1,1), seasonal = list(order=c(1,1,0)))
SARIMA_9<- arima(Maiz_ts,order=c(1,1,0), seasonal = list(order=c(1,1,0)))
```

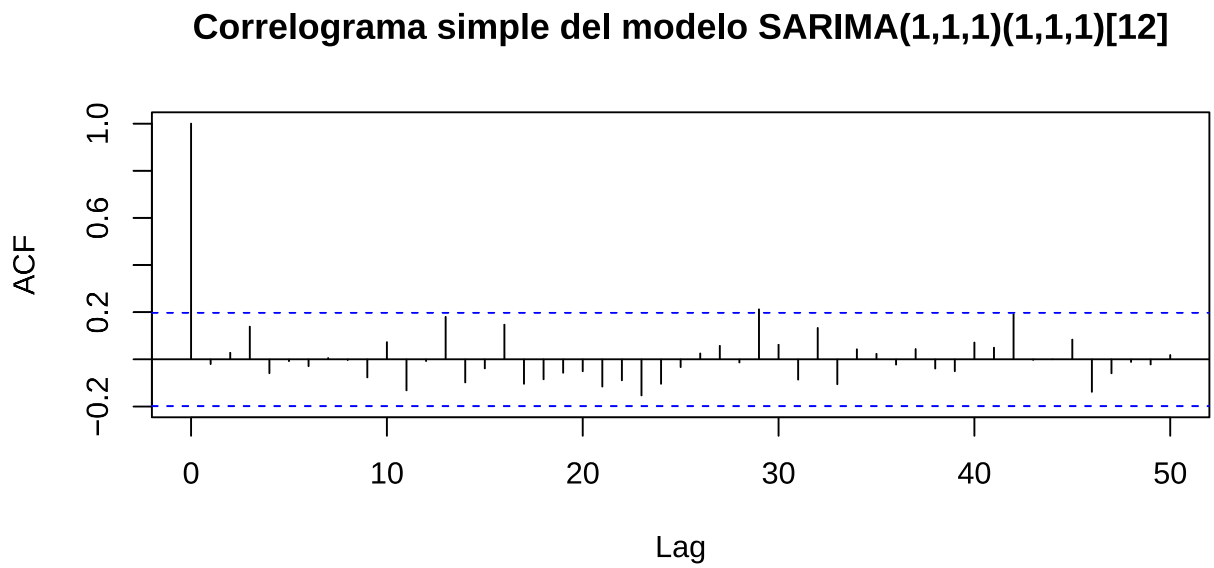
Vamos a utilizar el mejor modelo de acuerdo con los **Criterios de Información.**

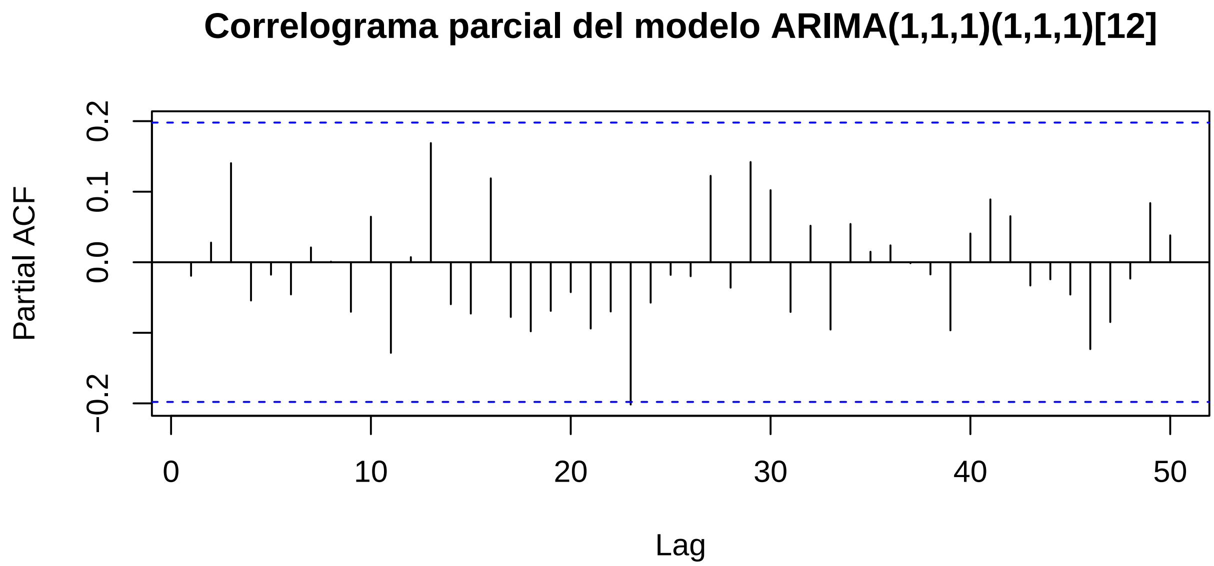
2.8.2. Criterios de Información

```
SARIMA_1$aic

## [1] 2265.639

SARIMA_2$aic

## [1] 2266.262

SARIMA_3$aic

## [1] 2274.933

SARIMA_4$aic
```

```
## [1] 2265.459

SARIMA_5$aic

## [1] 2265.818

SARIMA_6$aic

## [1] 2274.692

SARIMA_7$aic

## [1] 2275.807

SARIMA_8$aic

## [1] 2277.278

SARIMA_9$aic

## [1] 2285.34
```

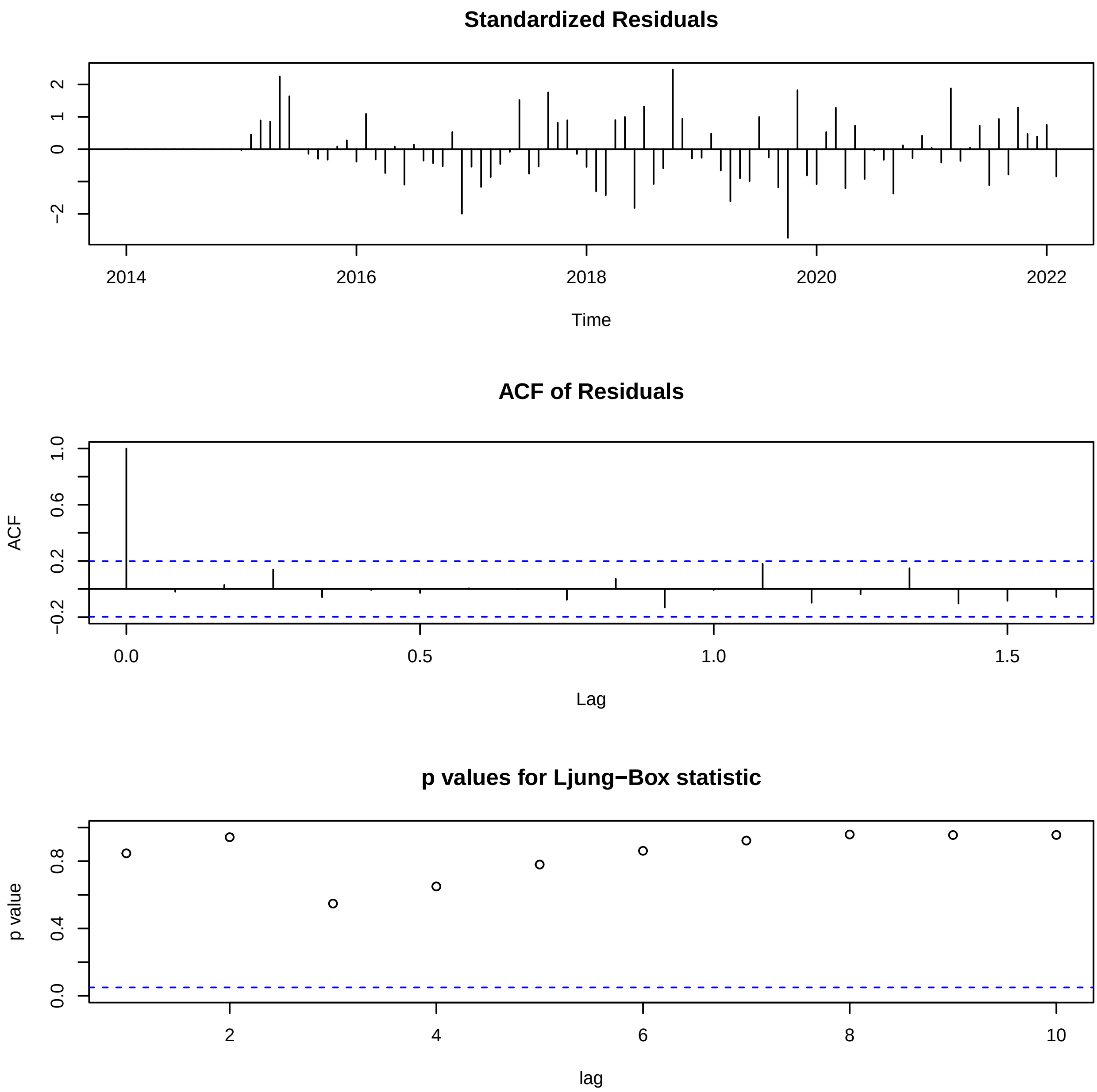
Conclusión: De acuerdo a los resultados de cada modelo, el mejor modelo obtenido es el modelo $SARIMA(1,1,1)(1,1,1)[12]$.





En esta ocasión observamos que en el Correlograma Simple, como el en Correlograma Parcial solamente hay un valor φ significativo, pero es muy muy mínimo. Así que en esta ocasión nuestro modelo se ajusta muy bien.

2.8.3. Diagnostico del modelo SARIMA(1,1,1)(1,1,1)[12]



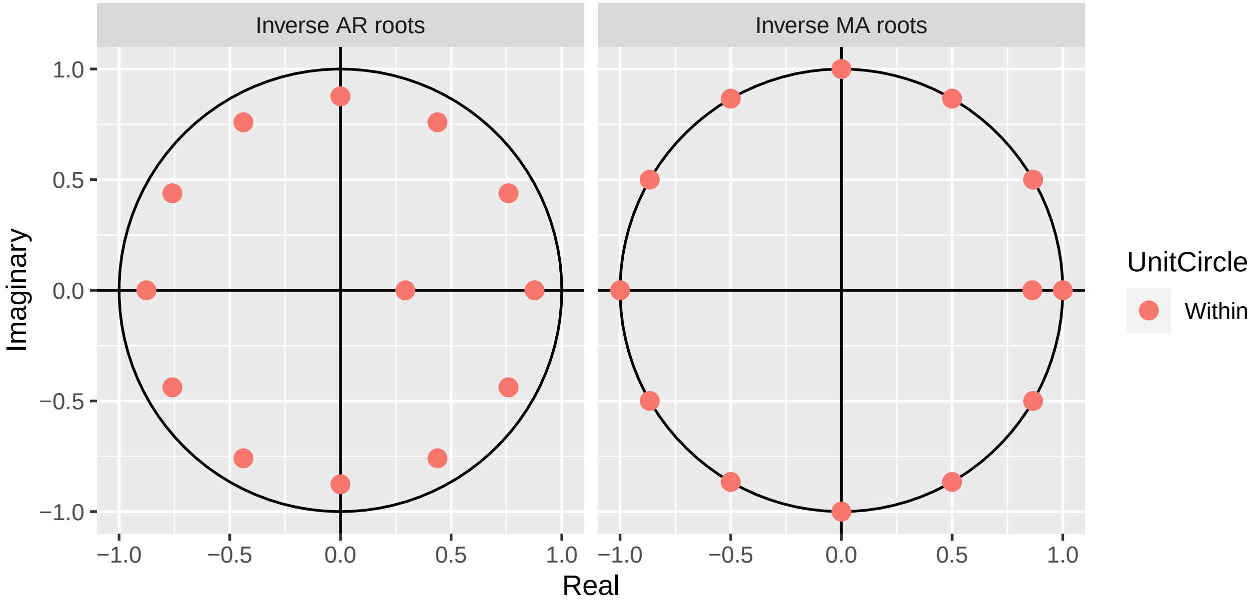
En la gráfica podemos ver los p-valor de **Ljung Box**, en donde recordemos que si el $P-valor > 0,05$, entonces hay **Ruido Blanco.**

Dado lo anterior, vemos que el modelo dado tiene *Ruido Blanco*, con esto concluimos que se ajusta bien. También podemos notar que la gráfica de ACF no hay valores de significancia (afuera de la franja azul), esto quiere decir que no hay correlación en los errores, es decir:

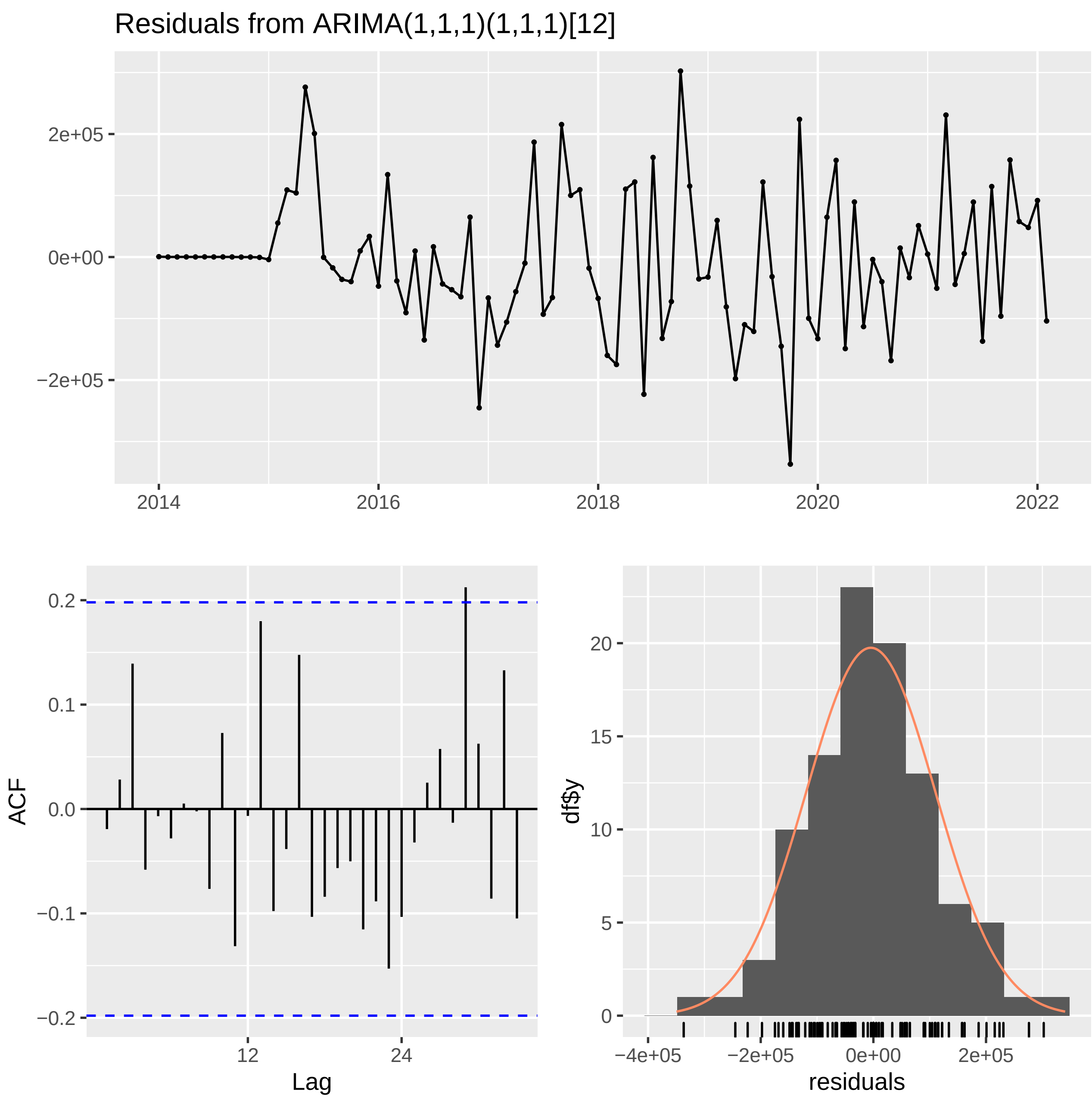
$$corr(\varepsilon_t, \varepsilon_{t-k}) = 0$$

Para todo $k \geq 1$ que pertenezca a los no. Naturales.

2.8.4. Gráfico de Raíces Unitarias



Las raíces del proceso $SARIMA(1,1,1)(1,1,1)$ en el modelo propuesto se encuentran dentro del *círculos unitario* en el caso de AR, pero en las del modelo MA están justo en la raya del círculo. Una vez comprobado este supuesto, podremos continuar con realizar los supuestos sobre los errores de las estimaciones del modelo. El cumplimiento de los supuestos básicos sobre los residuos es de suma importancia para el uso de pronósticos con modelos de series de tiempo, debido a que lo que se busca es que no exista (o que al menos no de manera importante) ninguna variable determinista que pueda afectar al modelo, dejando "limpia" esa variable, y por lo tanto, lo que sobre sea una variable **puramente aleatoria**. Para esto, tendremos que contrastar con unas últimas pruebas si estos errores se distribuyen o no de manera **normal**. Como una primera impresión, podemos ocupar la función *checkresiduals()* .




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(1,1,1)[12]
## Q* = 16.244, df = 16, p-value = 0.436
##
## Model df: 4.   Total lags used: 20
```

Al utilizar la función, podemos apreciar una imagen que cuenta con 3 gráficas útiles:

1. . Gráfica de los residuos: Podemos contrastar que, en efecto, los errores se mantienen en una media constate igual a cero, y dentro de unos intervalos de varianza entre 250000 y 250000.

2. Histograma junto con una gráfica de distribución: Observamos que la serie tiene un comportamiento ligeramente normal, ya que no presenta un importante sesgo ni una fuerte curtosis.

Para contrastar el cumplimiento del supuesto de **NO autocorrelación**, realizamos la prueba **Ljung-Box** sobre los errores.

Prueba Ljung-Box

$$H_0 : \textit{Ruido Blanco.}$$
$$H_a : \textit{No hay Ruido Blanco.}$$

```
##
##  Box-Ljung test
##
## data:  SARIMA_4$residuals
## X-squared = 0.037373, df = 1, p-value = 0.8467
```

Al realizar la prueba Ljung-Box sobre los residuos del modelo, con un p-valor mucho mayor al 0.05 nivel de significancia, podemos **aceptar la hipótesis nula** H_0 de que la serie no tiene problemas de auto-correlación.

2.8.5. Pruebas de Normalidad

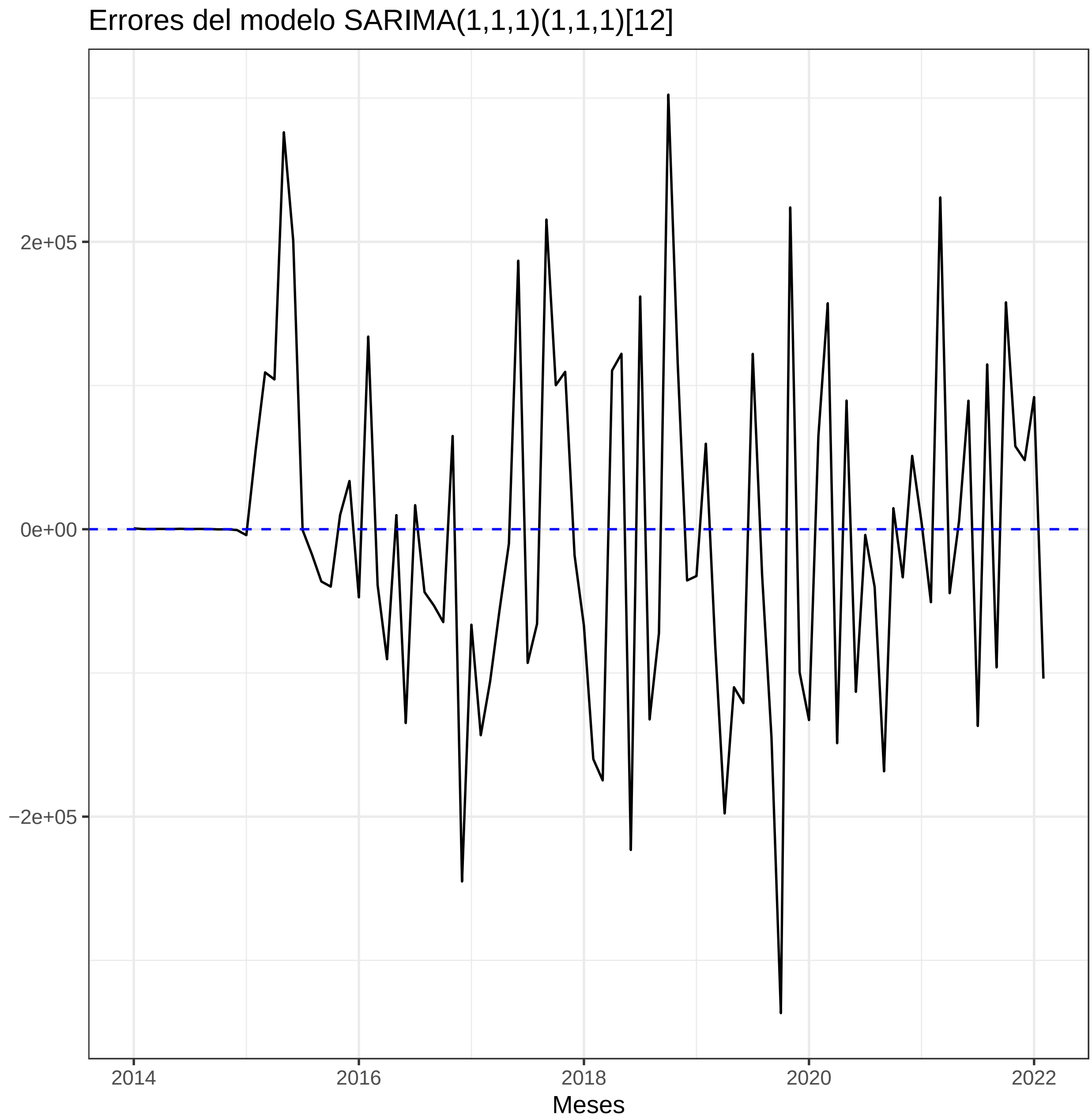
Para realizar la prueba de normalidad, vamos a proponer dos pruebas estadísticas:

1. Prueba de Jarque-Bera.
2. Prueba de Shapiro.

Las pruebas están dadas de la siguiente manera:

- H_0 : Los datos se comportan como una distribución normal.
- H_a : Los datos NO se comportan como una distribución normal.

El uso de las pruebas, al igual que el uso de las pruebas de raíces unitarias, servirán para confirmar de mejor manera el cumplimiento o el incumplimiento del supuesto de normalidad de la serie. Lo anterior lo hacemos de la siguiente manera:



Prueba de Jarque-Bera

En estadística, **la prueba de Jarque-Bera** es una prueba de bondad de ajuste para comprobar si una muestra de datos tienen la asimetría y la curtosis de una distribución normal

```
##
##  Jarque Bera Test
##
## data:  SARIMA_4$residuals
## X-squared = 0.94851, df = 2, p-value = 0.6223
```

El resultado de la prueba, al igual que otros contrastes que hemos mencionado, se puede confirmar observando el $p-valor$ obtenido. De acuerdo con los resultados de la prueba, con un p-valor mayor al 0.05, **Se Acepta la hipótesis nula de normalidad** (H_0) sobre los residuos del modelo seleccionado. Confirmamos que **el modelo tiene residuos que se distribuyen de manera normal.**
Por otra parte, procedemos a confirmar el supuesto de normalidad sobre los residuos utilizando de manera paralela la **prueba de Shapiro.**

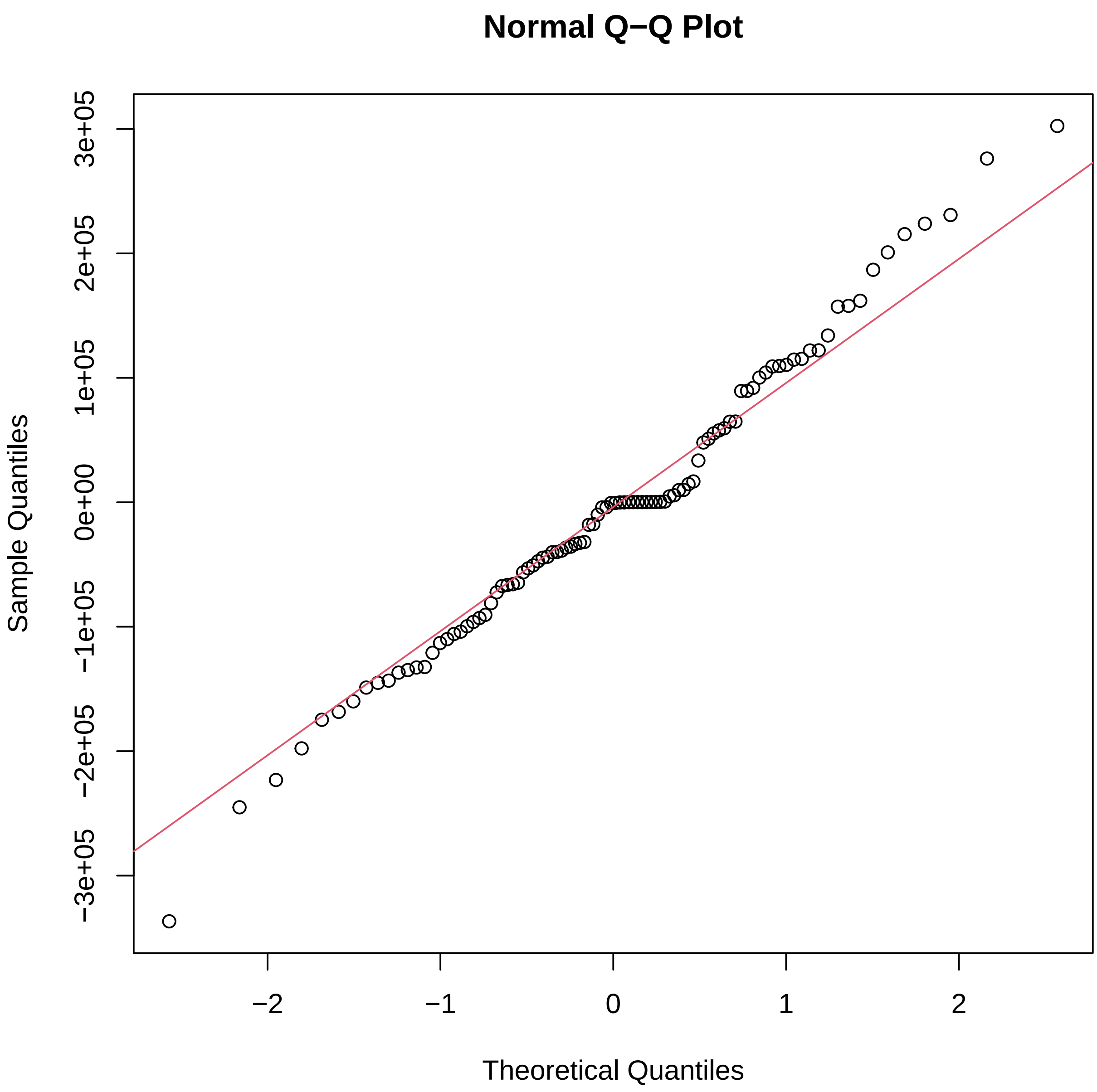
Prueba de Shapiro-Wilk

La Prueba de Shapiro-Wilk se usa para contrastar la normalidad de un conjunto de datos.

```
##
##  Shapiro-Wilk normality test
##
## data:  SARIMA_4$residuals
## W = 0.98651, p-value = 0.4201
```

El p-valor de la prueba, con un valor de 0.4401, mucho mayor al nivel de significancia del 0.05, **se acepta la hipótesis nula (H_0) de normalidad sobre los residuos del modelo** $SARIMA(1,1,1)(1,1,1)[12]$.

2.8.6. Contraste de normalidad con Q-Q plot



Prueba de Dickey-Fuller Aumentada a residuos de SARIMA(1,1,1)(1,1,1)[12]

```
##
##  Augmented Dickey-Fuller Test
##
## data:  SARIMA_4$residuals
## Dickey-Fuller = -3.9979, Lag order = 4, p-value = 0.01234
## alternative hypothesis: stationary
```

Podemos observar que la gran mayoría de puntos se alinean dentro de los datos simulados normales, de tal manera que se confirmar que los residuos de la serie presentan una distribución normal, con una media igual a cero $E[\varepsilon = 0]$ y una varianza constante $Var[\varepsilon] = c$. Con estas pruebas podemos confirmar que los pronósticos que arrojara el modelo serán los más eficientes posibles.

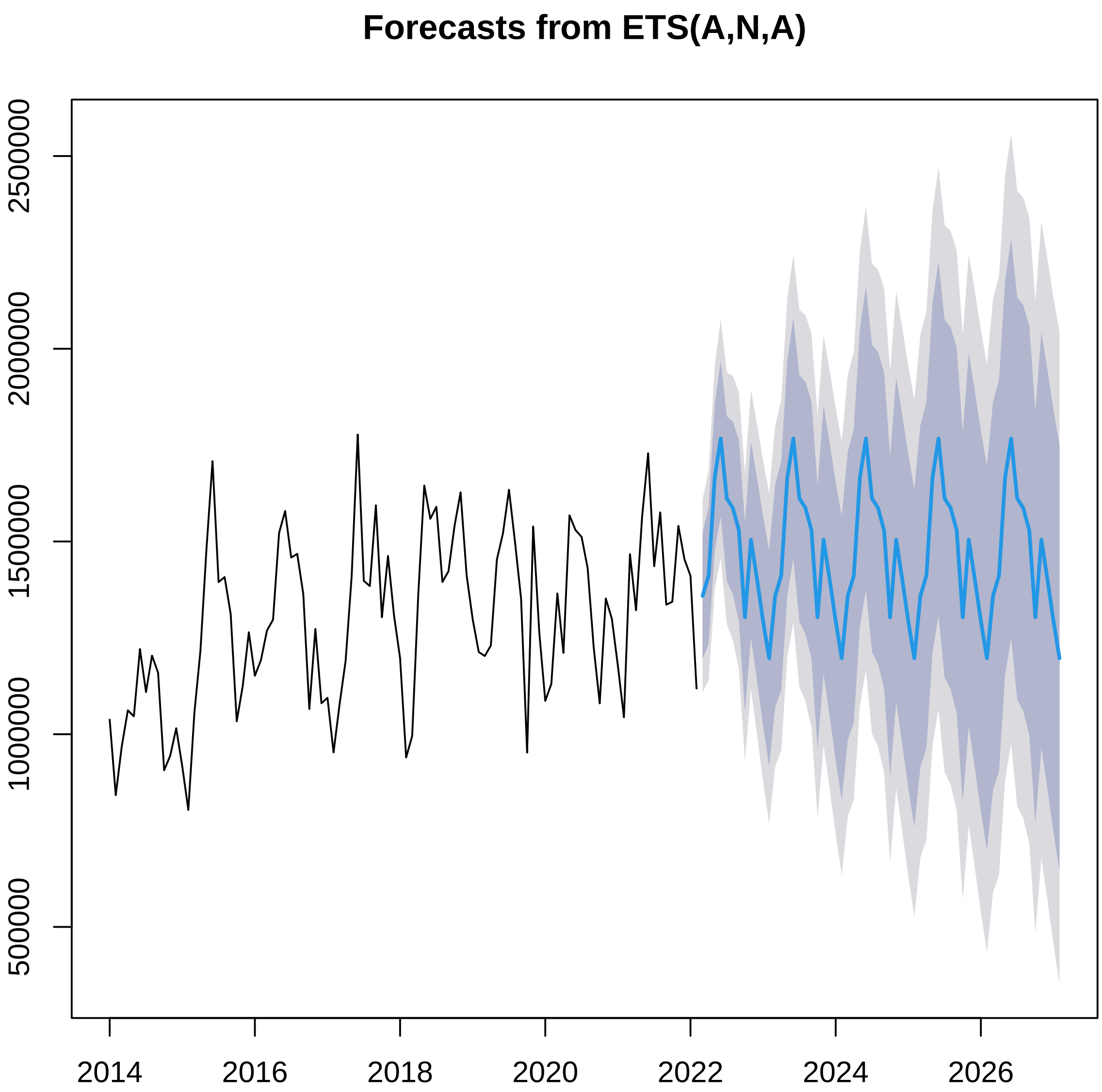
Conclusión: El modelo $AR(1, 1, 1)(1, 1, 1)[12]$ es eficiente para poder hacer un pronóstico.

2.9. Pronósticos

2.9.1. Predicción usando la función ETS() de R

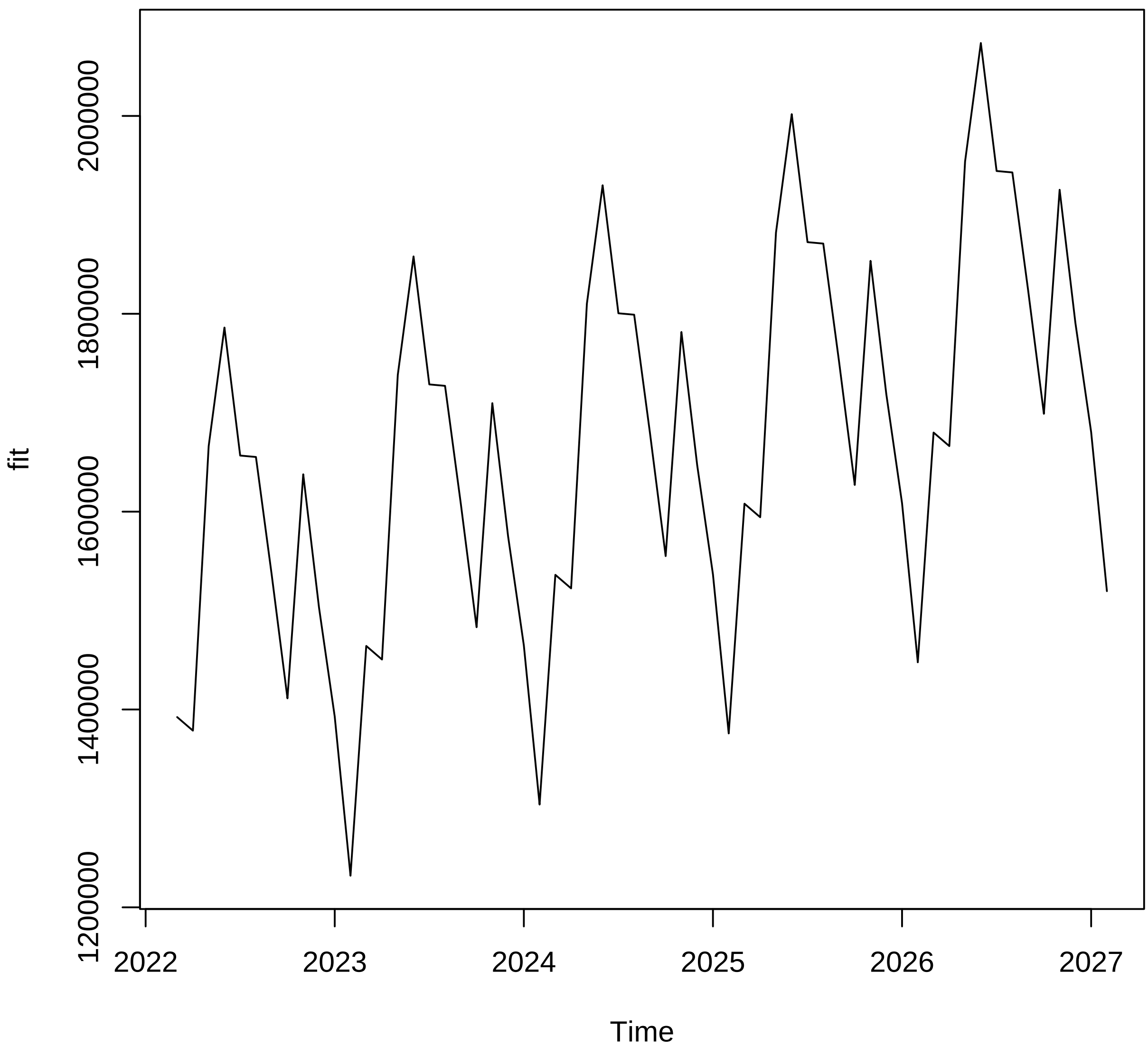
```
##          Point Forecast     Lo 80   Hi 80     Lo 95   Hi 95
## Mar 2022        1358901 1195419.1 1522382 1108877.1 1608924
## Apr 2022        1411741 1234408.2 1589074 1140533.8 1682948
## May 2022        1665017 1474839.0 1855195 1374164.8 1955869
## Jun 2022        1767237 1565028.4 1969446 1457985.6 2076489
## Jul 2022        1611548 1397985.5 1825111 1284932.1 1938165
## Aug 2022        1587120 1362777.2 1811463 1244017.1 1930224
## Sep 2022        1528756 1294127.7 1763385 1169922.8 1887590
## Oct 2022        1303272 1058790.7 1547754  929369.9 1677175
## Nov 2022        1504711 1250758.0 1758664 1116323.5 1893098
## Dec 2022        1401595 1138511.8 1664678  999243.9 1803946
## Jan 2023        1293426 1021518.9 1565333  877580.0 1709272
## Feb 2023        1196922  916464.1 1477379  767998.8 1625844
## Mar 2023        1358901 1070149.6 1647652  917294.1 1800507
## Apr 2023        1411741 1114928.2 1708554  957804.9 1865677
## May 2023        1665017 1360355.5 1969678 1199077.5 2130957
## Jun 2023        1767237 1454924.4 2079550 1289595.9 2244879
## Jul 2023        1611548 1291767.1 1931330 1122485.2 2100612
## Aug 2023        1587120 1260041.1 1914199 1086895.9 2087345
## Sep 2023        1528756 1194538.5 1862974 1017614.3 2039898
## Oct 2023        1303272  962065.3 1644479  781441.3 1825103
## Nov 2023        1504711 1156654.8 1852767  972405.1 2037016
## Dec 2023        1401595 1046822.2 1756368  859016.9 1944173
## Jan 2024        1293426  932061.4 1654791  740766.5 1846086
## Feb 2024        1196922  829080.2 1564763  634356.6 1759487
## Mar 2024        1358901  984697.4 1733104  786606.1 1931195
## Apr 2024        1411741 1031282.4 1792200  829879.7 1993602
## May 2024        1665017 1278404.1 2051630 1073743.6 2256290
## Jun 2024        1767237 1374566.6 2159908 1166699.3 2367775
## Jul 2024        1611548 1212912.0 2010185 1001886.6 2221210
## Aug 2024        1587120 1182606.1 1991634  968469.3 2205771
## Sep 2024        1528756 1118448.5 1939064  901244.7 2156268
## Oct 2024        1303272  887251.7 1719293  667023.7 1939521
## Nov 2024        1504711 1083054.7 1926367  859843.4 2149578
## Dec 2024        1401595  974377.6 1828812  748222.5 2054967
## Jan 2025        1293426  860719.0 1726133  631657.8 1955194
## Feb 2025        1196922  758791.2 1635052  526858.9 1866984
```

```
## Mar 2025        1358901  915415.6 1802386  680648.7 2037153
## Apr 2025        1411741  962965.2 1860517  725397.6 2098084
## May 2025        1665017 1211012.1 2119022  970676.4 2359358
## Jun 2025        1767237 1308062.8 2226412 1064990.5 2469484
## Jul 2025        1611548 1147261.9 2075835  901483.5 2321613
## Aug 2025        1587120 1117777.5 2056463  869322.5 2304918
## Sep 2025        1528756 1054411.1 2003101  803307.9 2254205
## Oct 2025        1303272  823976.9 1782568  570253.3 2036291
## Nov 2025        1504711 1020515.8 1988906  764198.4 2245223
## Dec 2025        1401595  912549.4 1890640  653664.3 2149526
## Jan 2026        1293426  799577.6 1787274  538150.0 2048702
## Feb 2026        1196922  698314.4 1695529  434367.6 1959476
## Mar 2026        1358901  855581.7 1862220  589140.8 2128661
## Apr 2026        1411741  903754.2 1919728  634842.1 2188640
## May 2026        1665017 1152404.7 2177629  881044.1 2448990
## Jun 2026        1767237 1250040.8 2284434  976253.6 2558221
## Jul 2026        1611548 1089808.2 2133289  813615.6 2409481
## Aug 2026        1587120 1060875.5 2113365  782298.4 2391942
## Sep 2026        1528756  998045.1 2059467  717103.6 2340409
## Oct 2026        1303272  768132.2 1838413  484846.1 2121699
## Nov 2026        1504711  965177.9 2044244  679566.4 2329855
## Dec 2026        1401595  857704.8 1945485  569786.7 2233403
## Jan 2027        1293426  745213.3 1841639  455007.0 2131845
## Feb 2027        1196922  644418.1 1749425  351940.4 2041903
```



2.9.2. Predicción usando Holt Winter

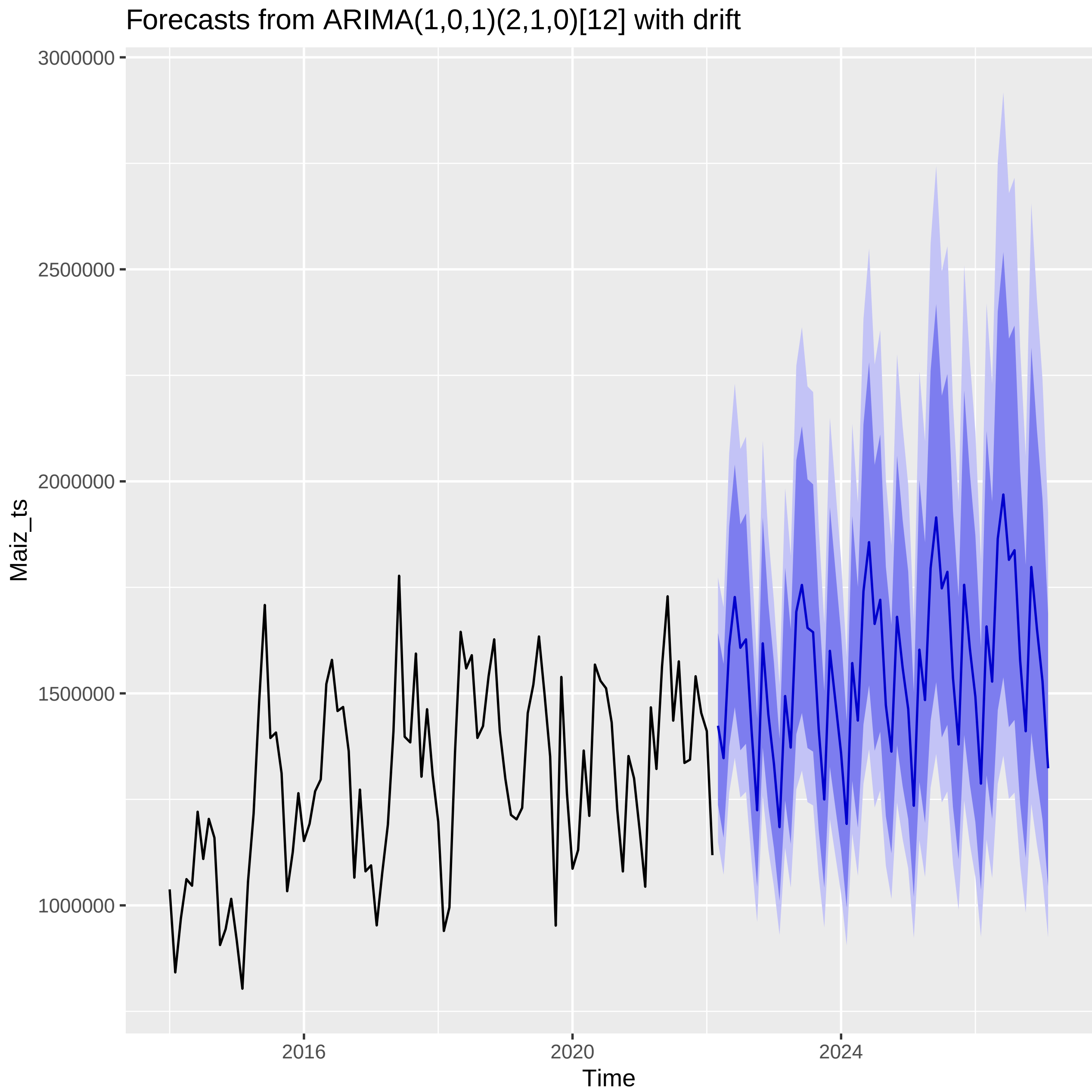
```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 2022                 1392310 1378603 1666292 1786091 1656689 1655248 1536163
## 2023 1392782 1231961 1464221 1450514 1738203 1858002 1728600 1727160 1608074
## 2024 1464693 1303872 1536132 1522425 1810114 1929913 1800511 1799071 1679985
## 2025 1536604 1375783 1608044 1594336 1882025 2001824 1872422 1870982 1751896
## 2026 1608515 1447694 1679955 1666247 1953936 2073735 1944333 1942893 1823807
## 2027 1680426 1519605
##          Oct     Nov     Dec
## 2022 1411248 1637760 1503188
## 2023 1483159 1709671 1575099
## 2024 1555070 1781582 1647010
## 2025 1626981 1853493 1718921
## 2026 1698893 1925404 1790832
## 2027
```



2.9.3. Predicción usando Auto.Arima()

```
##          Point Forecast     Lo 80   Hi 80     Lo 95   Hi 95
## Mar 2022        1423619 1237049.2 1642395 1149516.3 1773338
## Apr 2022        1347140 1159642.2 1569382 1072382.0 1703518
## May 2022        1610709 1374438.1 1893586 1265304.3 2065659
## Jun 2022        1727214 1467578.8 2039589 1348092.2 2230333
## Jul 2022        1607550 1365737.4 1898524 1254464.3 2076218
## Aug 2022        1627400 1381110.0 1924130 1267881.5 2105514
## Sep 2022        1413268 1202219.1 1666850 1104995.4 1821532
## Oct 2022        1224287 1044146.5 1440094  960977.0 1571431
## Nov 2022        1618009 1372159.4 1914449 1259203.1 2095773
## Dec 2022        1450403 1232626.6 1712356 1132386.4 1872281
## Jan 2023        1333857 1135417.0 1572109 1043949.6 1717355
## Feb 2023        1184494 1010597.9 1392728  930284.7 1519413
## Mar 2023        1493725 1247296.4 1796199 1135571.5 1983801
```

```
## Apr 2023        1372176 1144983.9 1651266 1042043.7 1824475
## May 2023        1692395 1403579.2 2049691 1273403.4 2272659
## Jun 2023        1755726 1453759.2 2129985 1317844.4 2363877
## Jul 2023        1654394 1371156.9 2005054 1243568.3 2224011
## Aug 2023        1644273 1362654.3 1992964 1235803.8 2210708
## Sep 2023        1415615 1177015.8 1709926 1069239.0 1893170
## Oct 2023        1249748 1041938.9 1505279  947849.8 1663988
## Nov 2023        1600349 1326778.5 1938920 1203511.2 2150269
## Dec 2023        1479456 1228723.7 1789132 1115574.9 1982136
## Jan 2024        1358424 1130365.3 1639476 1027278.8 1814339
## Feb 2024        1192241  994962.1 1434551  905566.0 1584918
## Mar 2024        1571503 1293666.6 1918173 1169237.5 2135963
## Apr 2024        1435830 1183128.4 1750784 1069861.5 1948477
## May 2024        1740677 1426823.7 2134220 1286779.9 2382412
## Jun 2024        1856671 1518981.6 2281041 1368550.9 2549140
## Jul 2024        1663751 1364528.8 2038705 1230949.8 2275055
## Aug 2024        1720707 1409946.2 2110533 1271325.5 2356461
## Sep 2024        1472273 1210848.8 1798815 1093863.9 2004134
## Oct 2024        1362601 1122678.7 1661662 1015148.5 1849395
## Nov 2024        1680545 1377686.2 2060254 1242535.3 2299698
## Dec 2024        1560807 1281799.3 1909898 1157101.8 2129681
## Jan 2025        1462968 1203306.0 1787274 1087100.7 1991170
## Feb 2025        1234952 1019826.9 1502400  923221.4 1669944
## Mar 2025        1603006 1290787.1 2002632 1153590.3 2258754
## Apr 2025        1484223 1194189.6 1855794 1066828.8 2094107
## May 2025        1795247 1434865.1 2260509 1277515.1 2560720
## Jun 2025        1914947 1526541.3 2417907 1357336.6 2743215
## Jul 2025        1747605 1396009.0 2201816 1242568.8 2495045
## Aug 2025        1786661 1425991.8 2253053 1268707.2 2554379
## Sep 2025        1536459 1231179.7 1929415 1097594.1 2182381
## Oct 2025        1379542 1108558.3 1727217  989692.5 1950461
## Nov 2025        1756073 1401897.3 2213947 1247414.4 2509709
## Dec 2025        1607299 1286204.5 2021258 1145861.9 2288072
## Jan 2026        1492011 1196337.6 1872316 1066883.5 2116994
## Feb 2026        1287149 1036139.0 1608542  925867.4 1814580
## Mar 2026        1657897 1306756.3 2118605 1155267.8 2419672
## Apr 2026        1527649 1204316.3 1951782 1064802.6 2228900
## May 2026        1864814 1459803.1 2400446 1286102.3 2752672
## Jun 2026        1968969 1537828.8 2540668 1353284.0 2917396
## Jul 2026        1814849 1420415.6 2336610 1251279.2 2679776
## Aug 2026        1837606 1437416.3 2367326 1265895.1 2715906
## Sep 2026        1578388 1240093.0 2023900 1094549.8 2315892
## Oct 2026        1410701 1111919.2 1802710  983019.2 2058881
## Nov 2026        1798056 1407124.4 2315248 1239504.9 2655440
## Dec 2026        1653114 1296903.6 2123018 1143845.5 2431408
## Jan 2027        1528563 1201943.0 1958288 1061322.2 2239719
## Feb 2027        1323357 1044937.9 1687896  924637.9 1925729
```

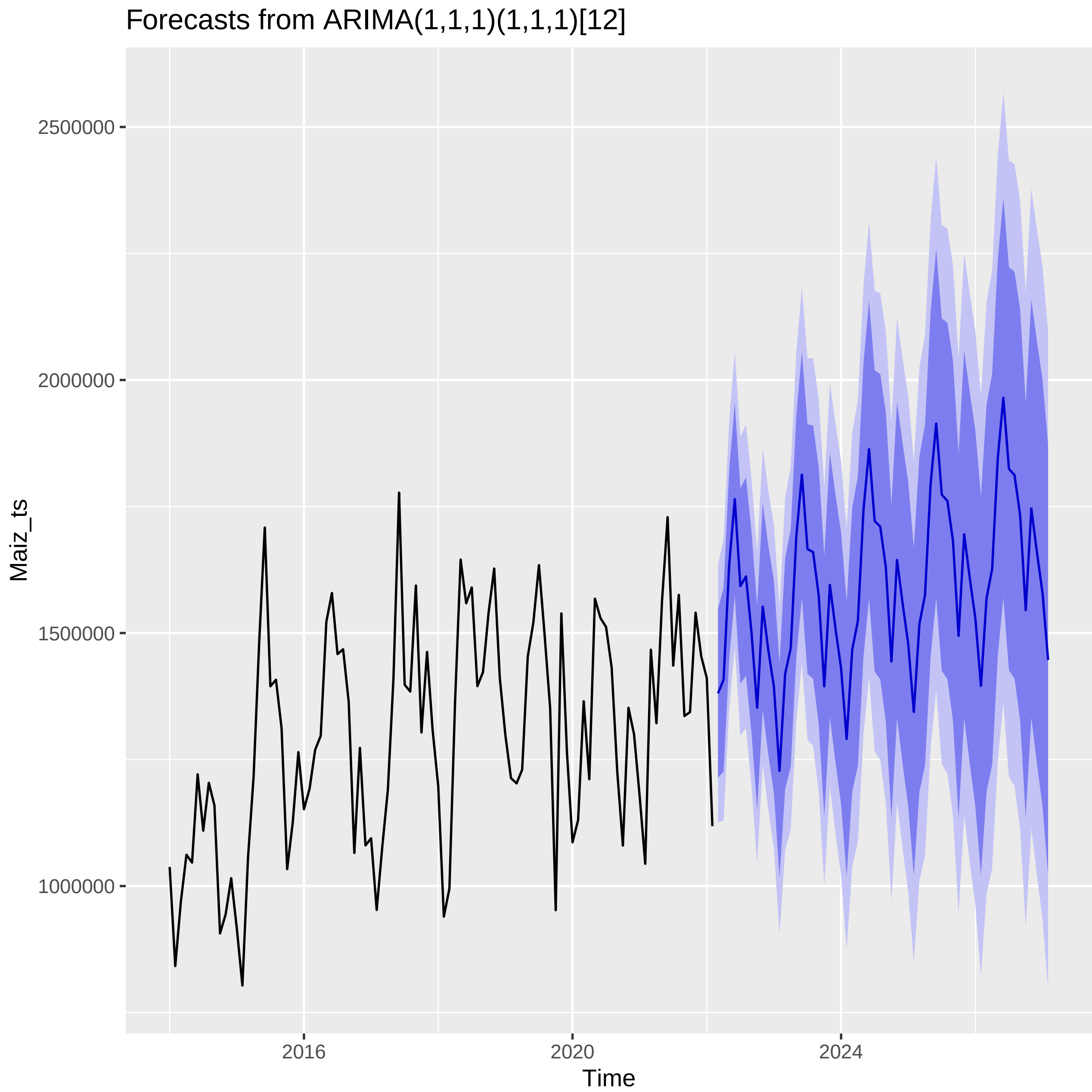


2.9.4. Predicción usando SARIMA(1,1,1)(1,1,1)[12]

```
SARIMA_4<- arima(Maiz_ts,order=c(1,1,1), seasonal = list(order=c(1,1,1)))
Pronostico_SARIMA <- forecast(SARIMA_4, h=60)
Pronostico_SARIMA

##          Point Forecast   Lo 80   Hi 80     Lo 95   Hi 95
## Mar 2022        1380818 1214060 1547577 1125783.0 1635854
## Apr 2022        1407499 1225857 1589140 1129701.4 1685296
## May 2022        1634268 1447355 1821182 1348408.3 1920128
## Jun 2022        1764574 1574256 1954891 1473508.7 2055638
## Jul 2022        1592926 1399700 1786151 1297412.0 1888439
## Aug 2022        1611840 1415868 1807812 1312126.3 1911553
## Sep 2022        1499850 1301202 1698499 1196043.2 1803657
## Oct 2022        1352351 1151062 1553639 1044506.2 1660195
## Nov 2022        1552243 1348325 1756161 1240376.7 1864109
```

```
## Dec 2022         1465990 1259412 1672568 1150056.6 1781924
## Jan 2023         1393245 1184130 1602360 1073430.9 1713059
## Feb 2023         1227747 1016166 1439328  904161.1 1551333
## Mar 2023         1419191 1191212 1647171 1070526.3 1767856
## Apr 2023         1470241 1235731 1704752 1111588.3 1828894
## May 2023         1690637 1451647 1929628 1325133.3 2056142
## Jun 2023         1812779 1569876 2055683 1441290.0 2184269
## Jul 2023         1665925 1419304 1912546 1288750.8 2043099
## Aug 2023         1659841 1409595 1910086 1277122.8 2042558
## Sep 2023         1574109 1320297 1827921 1185936.9 1962281
## Oct 2023         1394608 1137267 1651949 1001039.4 1788176
## Nov 2023         1595162 1334294 1856030 1196198.6 1994125
## Dec 2023         1508910 1244433 1773387 1104426.9 1913393
## Jan 2024         1430061 1161934 1698187 1019996.1 1840125
## Feb 2024         1290753 1019387 1562118  875734.8 1705771
## Mar 2024         1467570 1187047 1748092 1038547.9 1896592
## Apr 2024         1523643 1237892 1809393 1086625.5 1960660
## May 2024         1742725 1452730 2032720 1299216.1 2186234
## Jun 2024         1863185 1569249 2157120 1413648.7 2312721
## Jul 2024         1721441 1423685 2019196 1266062.9 2176819
## Aug 2024         1710204 1408695 2011712 1249085.5 2171322
## Sep 2024         1629884 1324669 1935100 1163097.1 2096671
## Oct 2024         1443787 1134892 1752682  971373.5 1916201
## Nov 2024         1644477 1331888 1957066 1166413.3 2122541
## Dec 2024         1558225 1241812 1874638 1074313.1 2042137
## Jan 2025         1478118 1157654 1798582  988011.0 1968225
## Feb 2025         1344209 1020391 1668026  848971.7 1839445
## Mar 2025         1518010 1186447 1849574 1010927.2 2025094
## Apr 2025         1575119 1238668 1911570 1060561.4 2089676
## May 2025         1793931 1453344 2134517 1273048.3 2314813
## Jun 2025         1914043 1569560 2258527 1387201.1 2440885
## Jul 2025         1773353 1425070 2121636 1240699.6 2306006
## Aug 2025         1761053 1409025 2113081 1222673.0 2299434
## Sep 2025         1681850 1326115 2037585 1137800.0 2225899
## Oct 2025         1494393 1134970 1853816  944702.9 2044083
## Nov 2025         1695111 1331971 2058251 1139735.9 2250486
## Dec 2025         1608859 1241836 1975882 1047545.8 2170172
## Jan 2026         1528493 1157213 1899773  960669.2 2096316
## Feb 2026         1395696 1021068 1770324  822752.0 1968640
## Mar 2026         1568876 1186973 1950780  984805.0 2152948
## Apr 2026         1626198 1239549 2012847 1034869.0 2217527
## May 2026         1844954 1454235 2235673 1247400.9 2442507
## Jun 2026         1964995 1570421 2359569 1361546.2 2568444
## Jul 2026         1824522 1426179 2222864 1215309.2 2433734
## Aug 2026         1812003 1409940 2214067 1197100.1 2426907
## Sep 2026         1733030 1327277 2138783 1112484.1 2353575
## Oct 2026         1545293 1135863 1954723  919123.6 2171462
## Nov 2026         1746017 1332870 2159163 1114162.9 2377870
## Dec 2026         1659765 1242708 2076821 1021931.5 2297598
## Jan 2027         1579345 1157890 2000799  934786.0 2223904
## Feb 2027         1446777 1021996 1871559  797130.0 2096425
```

```r
autoplot(Pronostico_SARIMA )
```




```r
Pronostico_SARIMA <- data.frame(Pronostico_SARIMA)
Pronostico_SARIMA_ts <- ts(Pronostico_SARIMA$Point.Forecast, frequency = 12,
                          start =c(2022,3))
```

Unión de Pronósticos

```
## Warning in file(file, "rt"): no fue posible abrir el archivo 'Fechas_prediccion.csv':
No such file or directory
## Error in file(file, "rt"): no se puede abrir la conexión
## Error in eval(expr, envir, enclos): objeto 'Fechaño encontrado
## Error in `[.data.frame`(Pronosticos_tabla, , c(5, 1:4)): undefined columns selected
##    Pronostico_Maiz_ETS_ts Pronostico_auto_Arima_ts Pronostico_H_W
## 1                 1358901                  1423619         1392310
## 2                 1411741                  1347140         1378603
## 3                 1665017                  1610709         1666292
## 4                 1767237                  1727214         1786091
```

```
## 5                    1611548                  1607550        1656689
## 6                    1587120                  1627400        1655248
## 7                    1528756                  1413268        1536163
## 8                    1303272                  1224287        1411248
## 9                    1504711                  1618009        1637760
## 10                   1401595                  1450403        1503188
## 11                   1293426                  1333857        1392782
## 12                   1196922                  1184494        1231961
## 13                   1358901                  1493725        1464221
## 14                   1411741                  1372176        1450514
## 15                   1665017                  1692395        1738203
## 16                   1767237                  1755726        1858002
## 17                   1611548                  1654394        1728600
## 18                   1587120                  1644273        1727160
## 19                   1528756                  1415615        1608074
## 20                   1303272                  1249748        1483159
## 21                   1504711                  1600349        1709671
## 22                   1401595                  1479456        1575099
## 23                   1293426                  1358424        1464693
## 24                   1196922                  1192241        1303872
## 25                   1358901                  1571503        1536132
## 26                   1411741                  1435830        1522425
## 27                   1665017                  1740677        1810114
## 28                   1767237                  1856671        1929913
## 29                   1611548                  1663751        1800511
## 30                   1587120                  1720707        1799071
## 31                   1528756                  1472273        1679985
## 32                   1303272                  1362601        1555070
## 33                   1504711                  1680545        1781582
## 34                   1401595                  1560807        1647010
## 35                   1293426                  1462968        1536604
## 36                   1196922                  1234952        1375783
## 37                   1358901                  1603006        1608044
## 38                   1411741                  1484223        1594336
## 39                   1665017                  1795247        1882025
## 40                   1767237                  1914947        2001824
## 41                   1611548                  1747605        1872422
## 42                   1587120                  1786661        1870982
## 43                   1528756                  1536459        1751896
## 44                   1303272                  1379542        1626981
## 45                   1504711                  1756073        1853493
## 46                   1401595                  1607299        1718921
## 47                   1293426                  1492011        1608515
## 48                   1196922                  1287149        1447694
## 49                   1358901                  1657897        1679955
## 50                   1411741                  1527649        1666247
## 51                   1665017                  1864814        1953936
## 52                   1767237                  1968969        2073735
## 53                   1611548                  1814849        1944333
## 54                   1587120                  1837606        1942893
## 55                   1528756                  1578388        1823807
## 56                   1303272                  1410701        1698893
## 57                   1504711                  1798056        1925404
## 58                   1401595                  1653114        1790832
```

```
## 59               1293426                      1528563        1680426
## 60               1196922                      1323357        1519605
##    Pronostico_SARIMA_ts
## 1               1380818
## 2               1407499
## 3               1634268
## 4               1764574
## 5               1592926
## 6               1611840
## 7               1499850
## 8               1352351
## 9               1552243
## 10              1465990
## 11              1393245
## 12              1227747
## 13              1419191
## 14              1470241
## 15              1690637
## 16              1812779
## 17              1665925
## 18              1659841
## 19              1574109
## 20              1394608
## 21              1595162
## 22              1508910
## 23              1430061
## 24              1290753
## 25              1467570
## 26              1523643
## 27              1742725
## 28              1863185
## 29              1721441
## 30              1710204
## 31              1629884
## 32              1443787
## 33              1644477
## 34              1558225
## 35              1478118
## 36              1344209
## 37              1518010
## 38              1575119
## 39              1793931
## 40              1914043
## 41              1773353
## 42              1761053
## 43              1681850
## 44              1494393
## 45              1695111
## 46              1608859
## 47              1528493
## 48              1395696
## 49              1568876
## 50              1626198
## 51              1844954
```

```
## 52                  1964995
## 53                  1824522
## 54                  1812003
## 55                  1733030
## 56                  1545293
## 57                  1746017
## 58                  1659765
## 59                  1579345
## 60                  1446777
```

2.9.5. Gráfica de los pronósticos