

# Evaluación análisis y predicción de series temporales

Ivonne V. Yáñez Mendoza

Profesora: Juana Maria Alonso Revenga

Master big data & business analytics, Universidad Complutense de Madrid

28 de julio de 2022

---

## 1. Introducción

Para este ejercicio de análisis y predicción de series temporales, se ha decidido utilizar la serie descrita a continuación:

El set de datos *Monthly figures on aviation* contiene información relevante al movimiento aéreo, tráfico de pasajeros, transporte de carga y correos de los aeropuertos de los Países Bajos.

Como lo indica su nombre, este set de datos contiene información de tipo mensual desde 1999 en adelante de los movimientos en distintos aeropuertos del país.

Para esta practica en particular, se ha decidido estudiar la serie temporal sobre el trafico de vuelos de tipo cargo (expresado en toneladas), para el aeropuerto de Schiphol en Amsterdam.

Se define como vuelo de tipo cargo, los que transportan mercancías exceptuando el traslado de maletas y correos.

Este set de datos ha sido extraído del repositorio de datos abiertos del instituto de estadística de los Países Bajos (Centraal bureau voor statistiek o CBS en sus siglas)

El set original de datos ha sido filtrado para extraer únicamente la información sobre las toneladas transportadas por vuelo de tipo cargo para Schiphol. El set original permite filtros por tipo de vuelo, aeropuertos y otros datos de interés (ver links)

Fuentes

1. ID del repositorio: 37478eng
2. Explicación de los datos: [https://opendata.cbs.nl/statline/portal.html?\\_la=en&\\_catalog=CBS&tableId=37478eng&\\_theme=1153](https://opendata.cbs.nl/statline/portal.html?_la=en&_catalog=CBS&tableId=37478eng&_theme=1153)
3. Acceso directo al set: <https://opendata.cbs.nl/statline/#/CBS/en/dataset/37478eng/table?ts=1659020962463>
4. Sitio oficial CBS, acceso a datos abiertos: <https://www.cbs.nl/en-gb/our-services/open-data>

## 1.1 Exploracion de datos

Se realiza una exploración previa de los datos a utilizar.

```
head(read_excel("aviation.xlsx"))
```

```
# A tibble: 6 x 2
  Period      Ton
  <chr>      <dbl>
1 2012 January 108877
2 2012 February 118016
3 2012 March   131325
4 2012 April   123613
5 2012 May     124957
6 2012 June    122386
```

Aunque la serie original comienza en 1999 se ha decidido para esta practica usar datos desde el 2012 en adelante.

El set de datos contiene dos columnas *Period* y *Ton*

*Period*: Mes y año

*Ton*: Toneladas transportadas por el sistema cargo

Creación de la serie de tiempo:

```
datos <- read_excel("aviation.xlsx", range = cell_cols("B"))
cargo <- ts(datos, start = c(2012, 1), end = c(2022, 5), frequency = 12)
# Comprobar inicio y fin de la serie
start(cargo)
```

```
[1] 2012    1
```

```
end(cargo)
```

```
[1] 2022    5
```

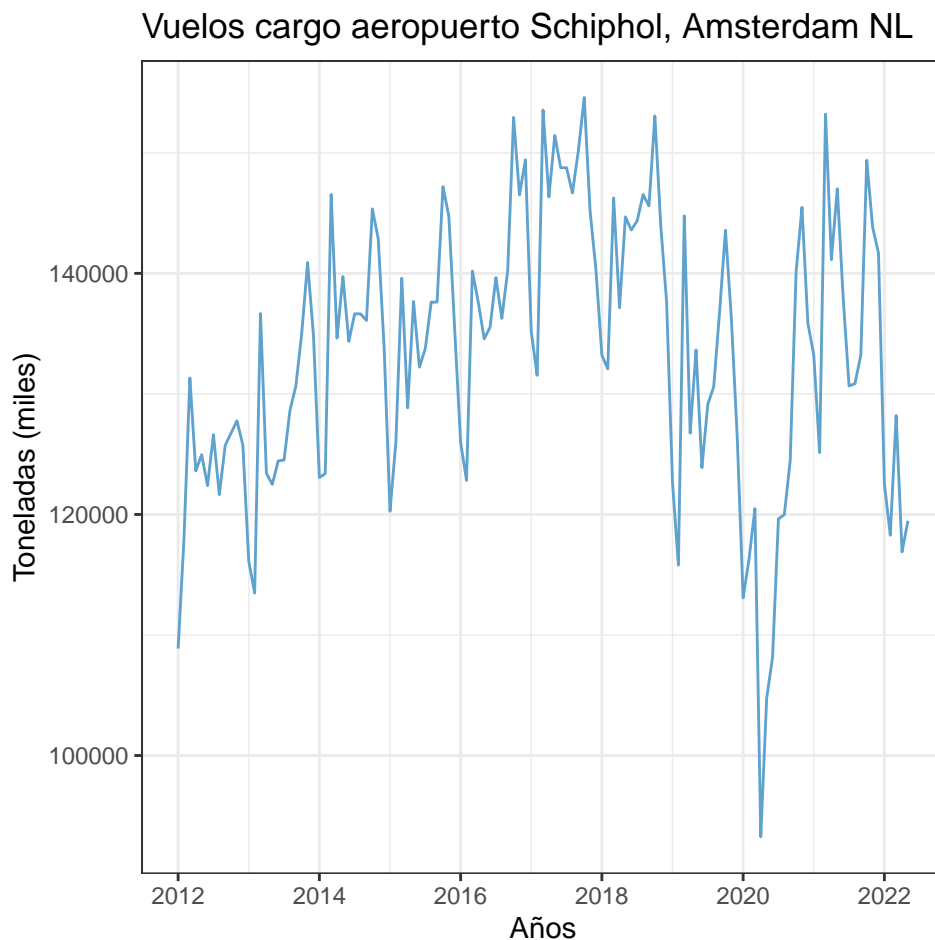
```
str(cargo)
```

```
Time-Series [1:125, 1] from 2012 to 2022: 108877 118016 131325 123613 124957 ...
- attr(*, "dimnames")=List of 2
..$ : NULL
..$ : chr "Ton"
```

Con str se obtiene el total de observaciones para esta serie, 125 registros.

## 2. Representación gráfica de la serie de tiempo y decomposición estacional

```
autoplot(cargo, colour = "#5fa2ce")+  
  ggtitle("Vuelos cargo aeropuerto Schiphol, Amsterdam NL") +  
  xlab("Años") + ylab("Toneladas (miles)") +  
  theme_bw()
```



En el gráfico se observa la tendencia al alza y picos por posible efecto de estacionalidad, en cuanto a la caída dramática en la línea del gráfico entre 2020-2021 se puede atribuir a múltiples factores, efecto COVID, crisis en aeropuertos europeos, guerras, traslado de aviones cargo desde Schiphol a otros recintos etc, el set de datos indica además que estos datos no son definitivos y pueden sufrir ajustes en un futuro. Al no tener media constante se puede inferir que no es estacionaria.

La serie de tiempo aunque va al alza no muestra variación en la magnitud del patrón con el paso del tiempo. Se podría decir que es una serie de tipo aditiva.

Solo como un recurso adicional, la función `mseastest()` de la librería `tsutils` es una comprobación rápida de si la serie es aditiva o multiplicativa.

Para la serie cargo el resultado es que es una serie de tipo aditiva y se realiza el gráfico de descomposición estacional.

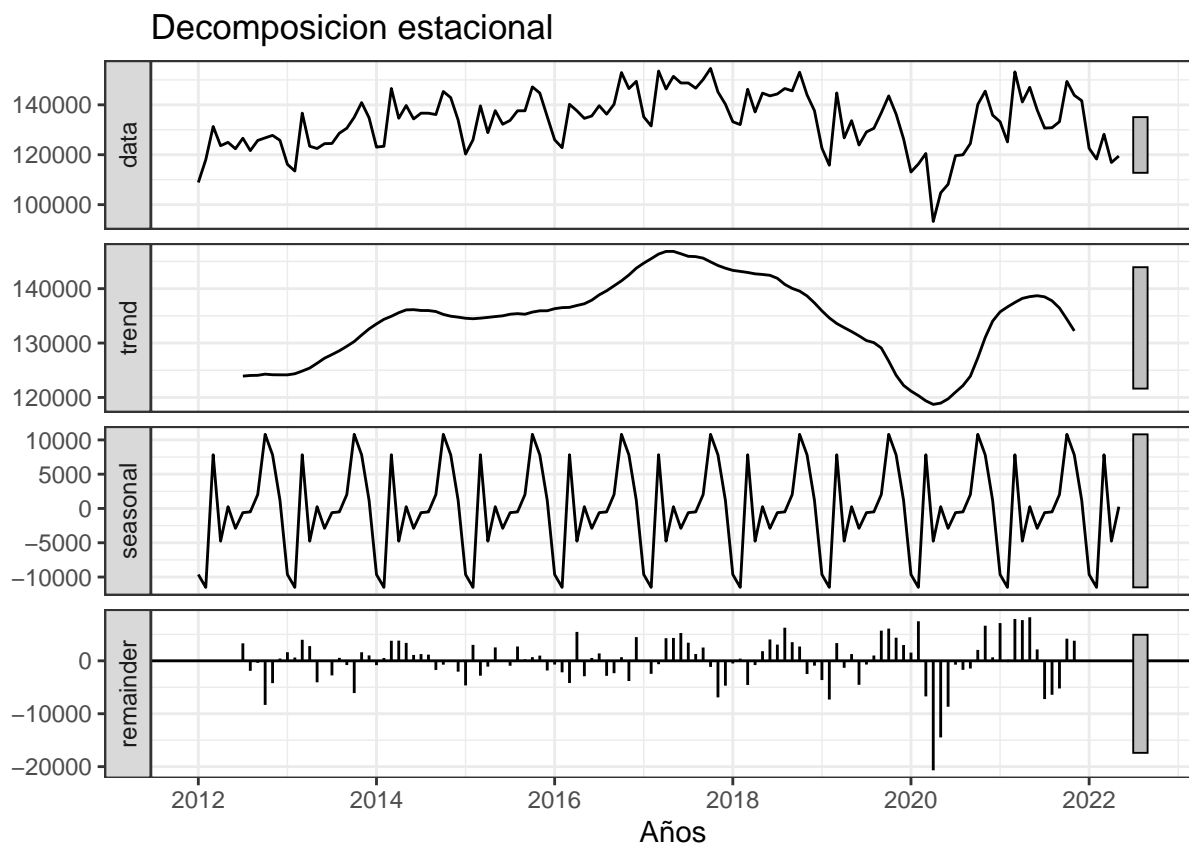
```
# Calcular coeficientes de estacionalidad
tsutils::mseastest(cargo)

$is.multiplicative
[1] FALSE

$statistic
[1] 0.201758

$pvalue
[1] 0.2880927

cargo_decomp <- decompose(cargo, type = c("additive"))
p1 <- autoplot(cargo_decomp, colour = "#5fa2ce") +
  ggtitle("Decomposicion estacional") +
  xlab("Años") +
  theme_bw()
print(p1)
```



Con el gráfico en la imagen anterior se observa algo similar a lo concluido en el análisis inicial de la serie de datos (figura 1) y que en el caso de la estacionalidad esta se repite de forma constante, por lo que se confirma el comportamiento.

## 2.2 Coeficientes de estacionalidad

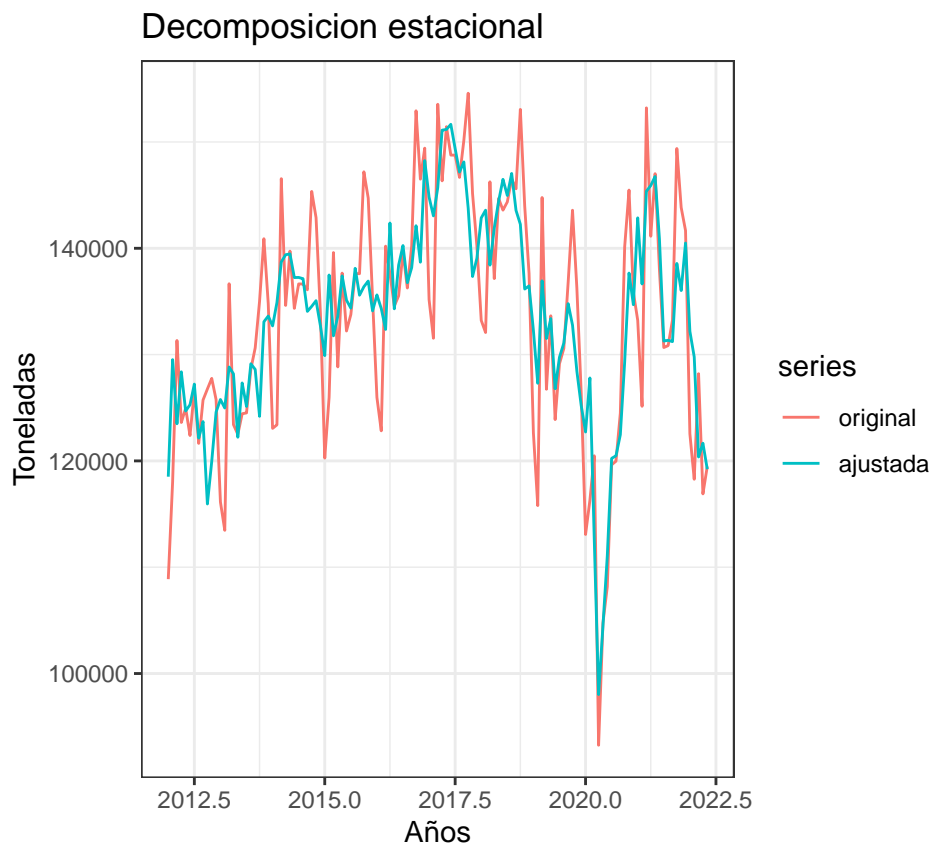
Cuadro 1: Coeficientes de estacionalidad

ene	feb	mar	abr	may	jun	jul	ago	sep	oct	nov	dic
-9638.03	-11506.7	7832.8	-4767.2	262.01	-2896.57	-607.38	-501.04	2028.52	10811.04	7807.86	1174.68

La tabla muestra el componente estacional de la serie de datos, en este caso un aumento del transporte de carga en toneladas sobre todo en los meses de marzo, octubre y noviembre. Esto coincidiría con los peaks de carga por cambios de temporada en el caso de marzo y en el caso de octubre y noviembre por celebraciones de fin de año.

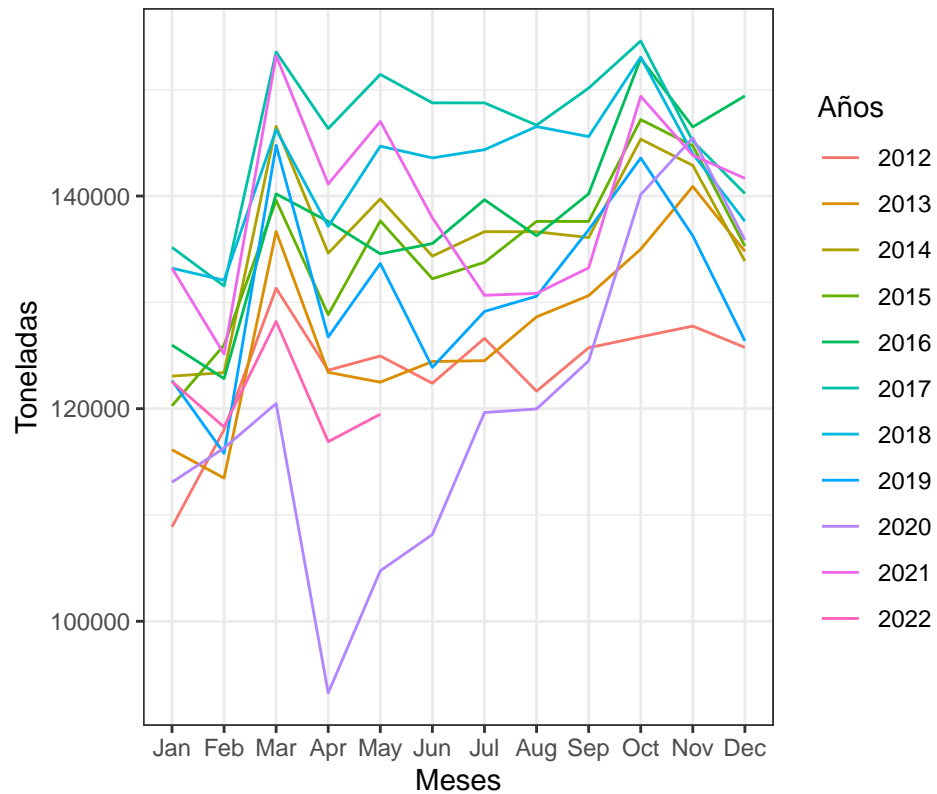
*# Representar la tendencia*

```
autoplot(cbind(cargo, seasadj(cargo_decomp))) +
  xlab("Años") + ylab("Toneladas") + ggtitle("Decomposicion estacional") +
  scale_color_discrete(labels = c("original", "ajustada")) +
  theme_bw()
```



```
ggseasonplot(cargo) + ggtitle("Grafico estacional: Vuelos cargo, Schiphol, NL") +
  xlab("Meses") + ylab("Toneladas") +
  scale_colour_discrete("Años") +
  theme_bw()
```

Grafico estacional: Vuelos cargo, Schiphol, NL



El gráfico estacional muestra los peaks de vuelos cargo para los meses de marzo, octubre y noviembre. Mientras que en los meses de enero, febrero y agosto se observan una baja en las toneladas transportadas. Se ve un efecto importante del COVID 19 para el mes de abril de 2020, representando una caída importante en las toneladas transportadas.

### 3. Partición de datos en train y test

Se realiza la partición de datos en train y test para contrastar la eficacia de los métodos de predicción. En el caso de train, se deja la serie de tiempo hasta diciembre de 2019 (pre pandemia) y los datos test desde enero de 2020.

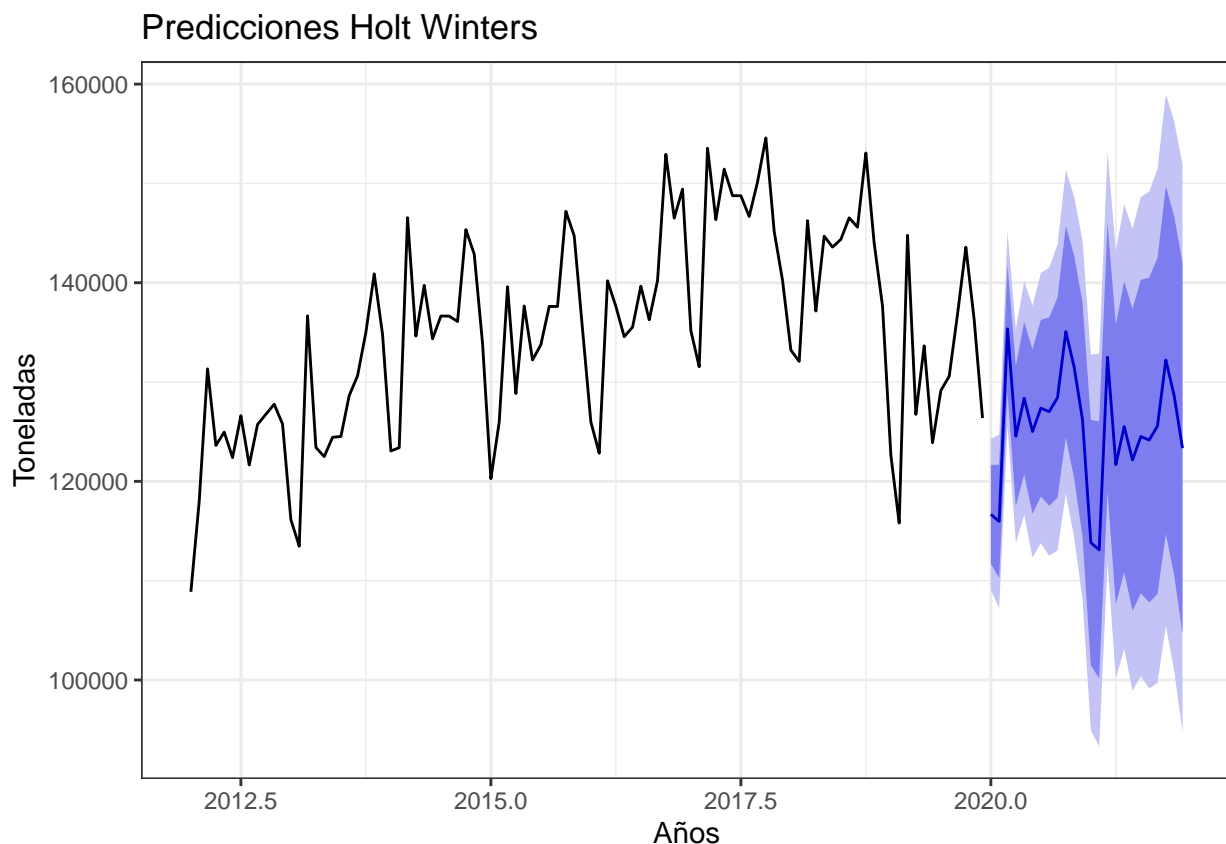
```
# Partición train/test
cargo_train <- window(cargo, end=c(2019, 12))
cargo_test  <- window(cargo, start = c(2020, 1))
```

### 4. Metodo de suavizado adecuado para la serie temporal

Una vez realizada la partición se prueba a realizar predicciones con el método de Holt-Winters que es el adecuado para este tipo de series con comportamiento estacional.

```
# Suavizado Holt Winters para series de tiempo con estacionalidad
cargo_hw <- hw(cargo_train, h=24, seasonal = 'additive', level = c(80,95))

autoplot(cargo_hw, colour = "#5fa2ce") + ggtitle("Predicciones Holt Winters ") +
  xlab("Años") +
  ylab("Toneladas") +
  theme_bw()
```



El gráfico en sus predicciones indica que la tendencia al alza en las toneladas transportadas se

mantendría. A continuación se muestran los valores de las predicciones además de sus intervalos de predicción para el nivel de confianza de 80 y 95.

Revisando los datos entregados en la tabla de predicciones, se puede indicar por ejemplo, que con un nivel de confianza de un 95% para el mes de enero de 2021 las toneladas a transportar vía cargo aéreo en el aeropuerto de Schiphol en Amsterdam estarán entre 94.913 y 132.729 toneladas.

```
knitr::kable(cargo_hw, digits = 0, format = "pipe",
              caption = "Predicciones e intervalos 80-95 suavizado Holt-Winters")
```

Cuadro 2: Predicciones e intervalos 80-95 suavizado Holt-Winters

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2020	116681	111708	121655	109076	124287
Feb 2020	115967	110259	121674	107238	124695
Mar 2020	135356	128962	141750	125577	145135
Apr 2020	124546	117498	131595	113767	135326
May 2020	128371	120692	136050	116626	140115
Jun 2020	125013	116720	133306	112330	137696
Jul 2020	127371	118477	136264	113769	140972
Aug 2020	127019	117534	136504	112513	141525
Sep 2020	128431	118363	138500	113032	143831
Oct 2020	135075	124428	145723	118791	151359
Nov 2020	131481	120259	142702	114318	148643
Dec 2020	126201	114408	137994	108164	144238
Jan 2021	113821	101457	126184	94913	132729
Feb 2021	113106	100174	126038	93328	132884
Mar 2021	132495	118995	145995	111849	153142
Apr 2021	121685	107617	135754	100170	143201
May 2021	125510	110873	140147	103125	147895
Jun 2021	122153	106946	137359	98896	145409
Jul 2021	124510	108732	140287	100380	148640
Aug 2021	124158	107808	140508	99153	149164
Sep 2021	125571	108646	142495	99687	151454
Oct 2021	132214	114714	149715	105449	158979
Nov 2021	128620	110541	146699	100970	156269
Dec 2021	123340	104680	142000	94802	151878

Los parámetros del modelo ajustado son:

```
knitr::kable(cargo_hw$model$par, digits = 4, format = "pipe",
              caption = "Estimadores de los parámetros")
```



Cuadro 3: Estimadores de los parámetros

	x
alpha	0.5464
beta	0.0166
gamma	0.0001
l	125845.9670
b	429.5093
s0	720.0282
s1	5760.8404
s2	9116.5943
s3	2234.2529
s4	583.7539
s5	697.3324
s6	-1898.5834
s7	1220.5284
s8	-2842.7458
s9	7729.1748
s10	-11898.8512

$$L_\tau = 0.5464(x_\tau - S_{\tau-s}) + (1 - 0.5464)(L_{\tau-1} + b_{\tau-1})$$

$$B_\tau = 0.0166(L_\tau - L_{\tau-1}) + (1 - 0.0166)b_{\tau-1}$$

$$S_\tau = 0.0001(x_\tau - L_\tau) + (1 - 0.0001)S_{\tau-s}$$

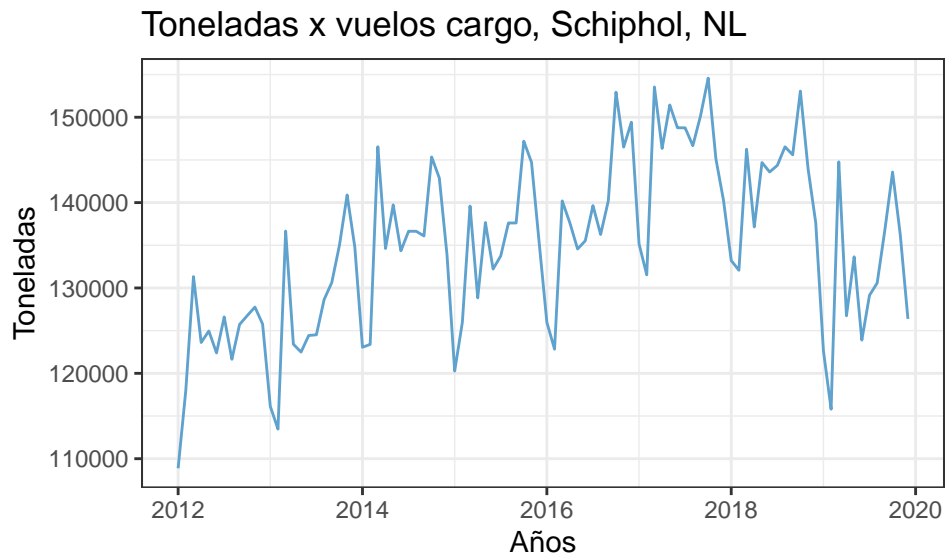
$$\hat{x}_{\tau+1} = (L_\tau + b_\tau) + S_{\tau-s+1}$$

## 5. Correlogramas

Se muestra el plot y correlogramas de la serie original

```
#Autoplot serie original
```

```
autoplot(cargo_train, colour = "#5fa2ce") +  
  ggtitle("Toneladas x vuelos cargo, Schiphol, NL") +  
  xlab("Años") + ylab("Toneladas") + theme_bw()
```

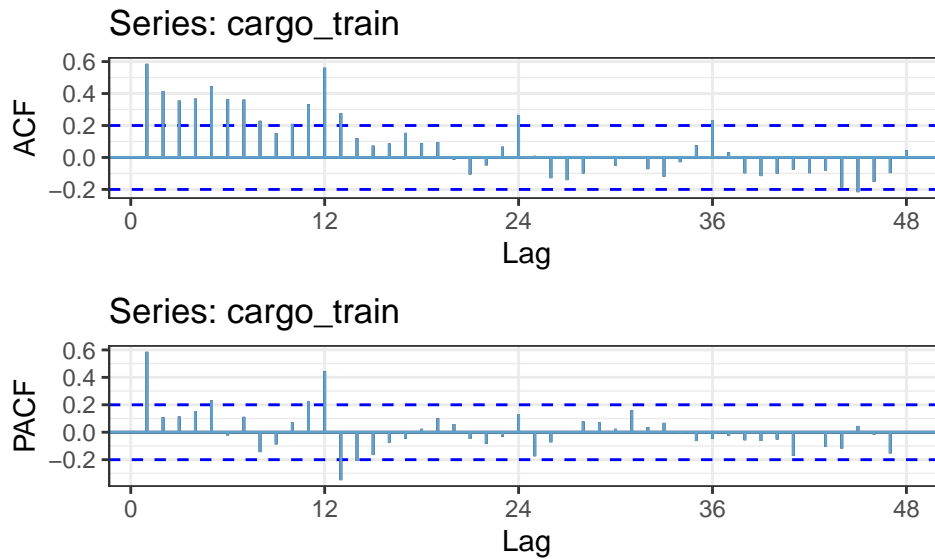


```
# ACF y PACF
```

```
a <- ggAcf(cargo_train, lag= 48) +  
  geom_segment(lineend = "butt", color = "#5fa2ce") +  
  geom_hline(yintercept = 0, color = "#5fa2ce") +  
  theme_bw()
```

```
b <- ggPacf(cargo_train, lag = 48) +  
  geom_segment(lineend = "butt", color = "#5fa2ce") +  
  geom_hline(yintercept = 0, color = "#5fa2ce") +  
  theme_bw()
```

```
grid.arrange(a, b, ncol=1)
```



```
adf <- adf.test(cargo_train, alternative = "stationary")
adf$p.value
```

```
[1] 0.5558598
```

En el primer plot es la serie original que muestra picos de estacionalidad que se han descrito anteriormente y tendencia al alza con el paso del tiempo.

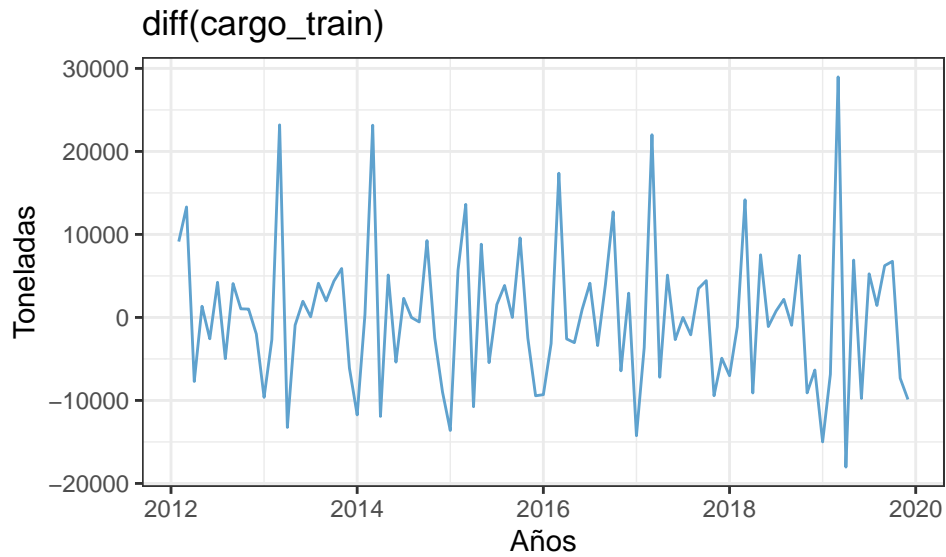
En los correlogramas ACF y PACF se observa que la serie original tiende a decrecer de forma lenta lo que podría indicar que no es estacionaria por esto que se decide a aplicar una primera diferencia con el fin de convertir la serie a estacionaria.

Como recurso adicional se utiliza el `adf.test` para saber si es estacionaria o no. Observando el p value se puede concluir que no lo es.

## 5.1 Primera diferencia aplicada

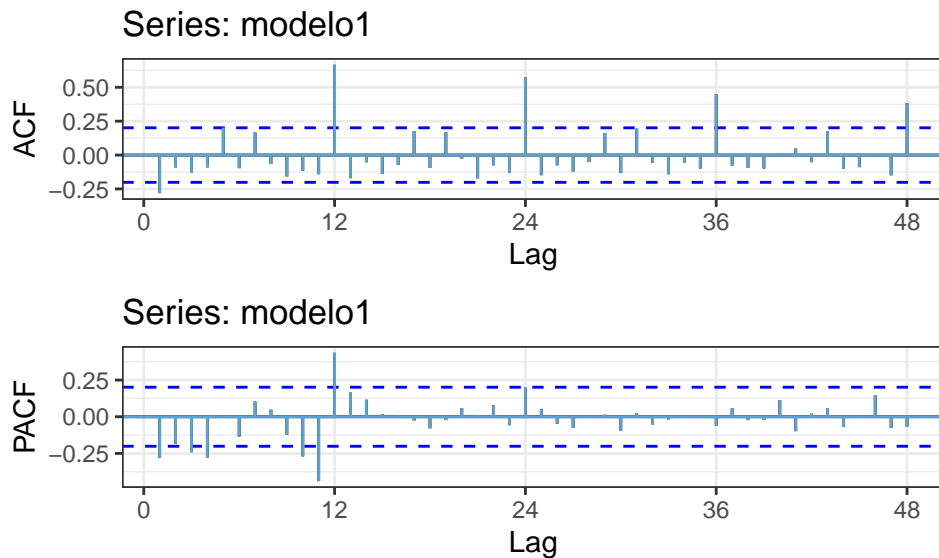
Plot y correlogramas de la serie con diferencia aplicada

```
# Se aplica la primera diferencia
modelo1 <- diff(cargo_train)
autoplot(modelo1, colour = "#5fa2ce") + ggtitle('diff(cargo_train)') +
  xlab("Años") + ylab("Toneladas") +
  theme_bw()
```



```
# ACF y PACF
c <-ggAcf(modelo1, lag= 48) +
  geom_segment(lineend = "butt", color = "#5fa2ce") +
  geom_hline(yintercept = 0, color = "#5fa2ce") +
  theme_bw()

d <-ggPacf(modelo1, lag= 48) +
  geom_segment(lineend = "butt", color = "#5fa2ce") +
  geom_hline(yintercept = 0, color = "#5fa2ce") +
  theme_bw()
grid.arrange(c, d, ncol=1)
```



Se observa en el primer gráfico con autoplot que con esta diferencia la serie se va a semejando a una serie con ruido blanco. Se mantienen picos significativos en los lag 12, 24 por lo que es necesario aplicar una segunda diferencia considerando la estacionalidad.

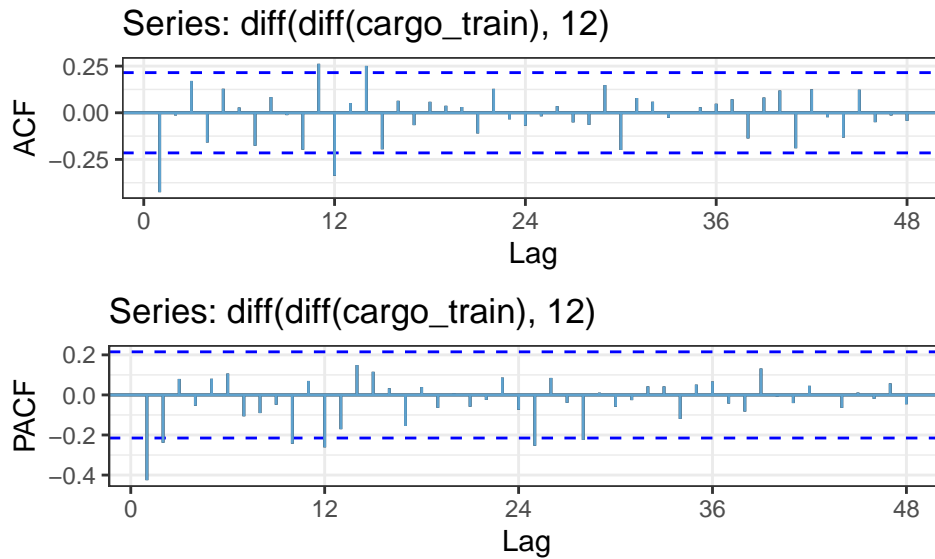
## 5.2 Segunda diferencia aplicada

Plot y correlogramas de la serie con diferencia aplicada en estacionalidad

```
# Se aplica diferencia con estacionalidad  
autoplot(diff(diff(cargo_train), 12), lag = 48, colour = "#5fa2ce") +  
  ggtitle('Diferencia con estacionalidad') +  
  xlab("Años") + ylab("Toneladas") +  
  theme_bw()
```



```
e <- ggAcf(diff(diff(cargo_train), 12), lag = 48) +  
  geom_segment(lineend = "butt", color = "#5fa2ce") +  
  geom_hline(yintercept = 0, color = "#5fa2ce") +  
  theme_bw()  
  
f <- ggPacf(diff(diff(cargo_train), 12), lag = 48) +  
  geom_segment(lineend = "butt", color = "#5fa2ce") +  
  geom_hline(yintercept = 0, color = "#5fa2ce") +  
  theme_bw()  
  
grid.arrange(e, f, ncol=1)
```



Se observa una mejora en los gráficos ACF y PACF, se han “suavizado” los picos en los lags 12, 24 etc.

### 5.3 Ajuste del modelo

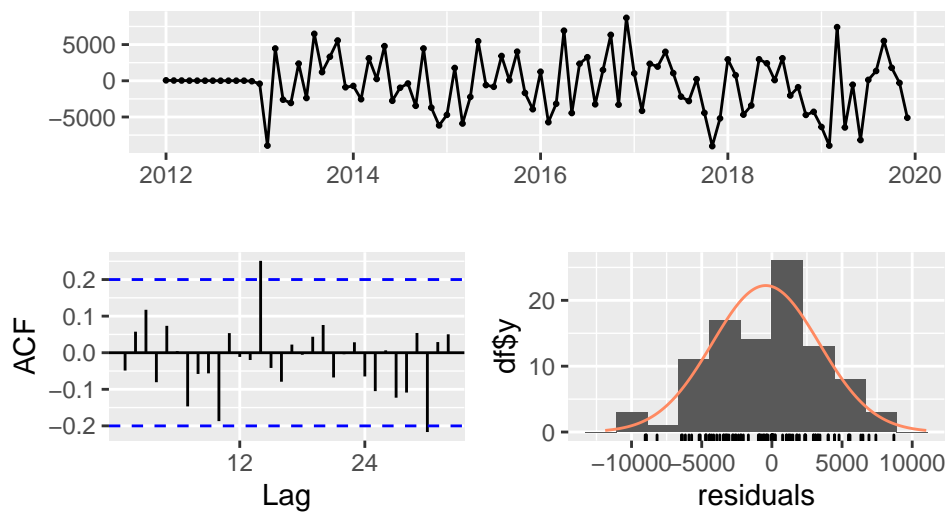
Observando los gráficos ACF y PACF la serie podría ser de tipo MA(1) considerando esto se propone realizar un primer ajuste manual, se propone un modelo (0,1,1) (considerando aplicar una diferencia) y un ajuste *seasonal* de (0,1,1) con su correspondiente diferencia estacional.

Determinar el ajuste correcto de forma manual puede ser de alta dificultad, en el apartado anterior se propone un modelo en base MA, pero revisando los gráficos también podría ser un ARMA. En el segundo modelo propuesto se prueba considerando este modelo.

Se muestran los resultados para ambas pruebas. El p value es mejor en el modelo1 que en el modelo 2.

```
# Arima manual
arima1 <- Arima(cargo_train, order=c(0,1,1), seasonal=c(0,1,1))
checkresiduals(arima1)
```

Residuals from ARIMA(0,1,1)(0,1,1)[12]



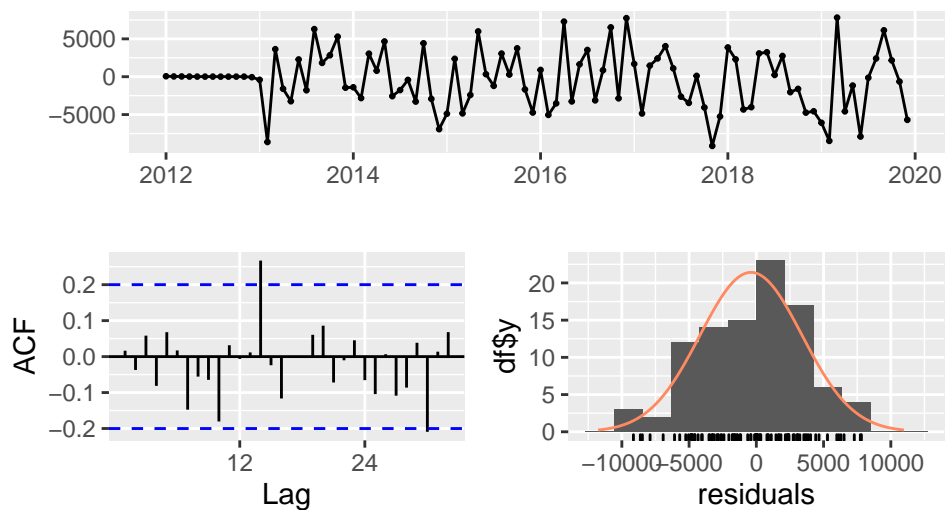
Ljung-Box test

```
data: Residuals from ARIMA(0,1,1)(0,1,1)[12]
Q* = 18.842, df = 17, p-value = 0.3377
```

Model df: 2. Total lags used: 19

```
arima2 <- Arima(cargo_train, order = c(0,1,2), seasonal = c(1,1,1))
checkresiduals(arima2)
```

Residuals from ARIMA(0,1,2)(1,1,1)[12]



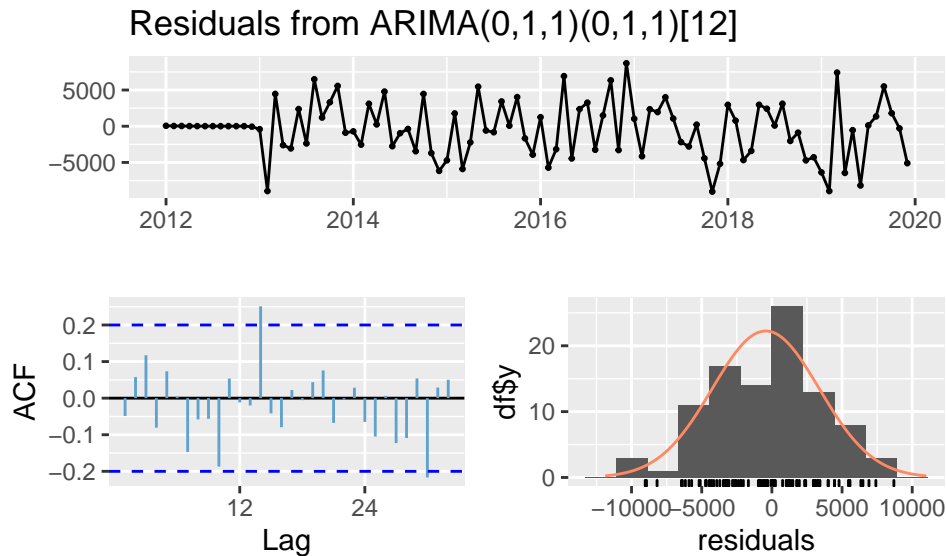
Ljung-Box test

```
data: Residuals from ARIMA(0,1,2)(1,1,1)[12]
Q* = 18.75, df = 15, p-value = 0.2253
```

Model df: 4. Total lags used: 19

A continuación se prueba la función `autoarima` para estimar el mejor ajuste al modelo y comprobar las propuestas realizadas anteriormente.

```
# Arima automatico
arima_auto <- auto.arima(cargo_train, stepwise = TRUE)
checkresiduals(arima_auto, colour = "#5fa2ce")
```



Ljung-Box test

```
data: Residuals from ARIMA(0,1,1)(0,1,1)[12]
Q* = 18.842, df = 17, p-value = 0.3377
```

Model df: 2. Total lags used: 19

El resultado que entrega la función `auto.arima()` es igual a la planteada en el modelo manual.

El p value es de 0.3377 se podría aceptar la hipótesis de los residuos incorrelados y se observa en el autocorrelograma que prácticamente todos los picos están dentro de las líneas de intervalos de confianza. El modelo ARIMA para considerar predicciones es el `arima1`.

## 6. Expresion algebraica modelo ajustado

```
print(arima_auto)
```

```
Series: cargo_train
ARIMA(0,1,1)(0,1,1)[12]
```

Coefficients:

```
      ma1      sma1
      -0.4513 -0.6671
s.e.    0.0880  0.1252
```

```
sigma^2 = 17280775: log likelihood = -811.99
```



AIC=1629.99    AICc=1630.29    BIC=1637.24

Los parámetros estimados son: -0.45 para MA y -0.6625 para medias móviles.

*Arima*

$$(1 + 0.4513B^{12})(1 - B^{12})X_{\tau} = (1 - 0.6671B^{12})Z_{\tau}$$

$$(1 + 0.45136B^{12})(1 - B^{12})(X_{\tau} - X_{\tau-1}) = -0.6671Z_{\tau-12} + Z_{\tau}$$

$$(1 + 0.4513B^{12})(X_{\tau} - X_{\tau-12} - X_{\tau-1} + X_{\tau-13}) = -0.6671Z_{\tau-12} + Z_{\tau}$$

$$X_{\tau} - X_{\tau-12} - X_{\tau-1} + X_{\tau-13} + 0.57X_{\tau-12} - 0.45X_{\tau-24} - 0.45X_{\tau-13} + 0.45X_{\tau-25} = 0.66Z_{\tau-12} + Z_{\tau}$$

$$X_{\tau} - 1.45X_{\tau-12} - X_{\tau-1} + 1.45X_{\tau-13} - 1.457X_{\tau-24} + 1.45X_{\tau-25} = -0.66Z_{\tau-12} + Z_{\tau}$$

$$X_{\tau} = X_{\tau-1} + 0.45X_{\tau-12} - 0.45X_{\tau-13} + 0.457X_{\tau-24} - 0.45X_{\tau-25} - 0.66Z_{\tau-12} + Z_{\tau}$$

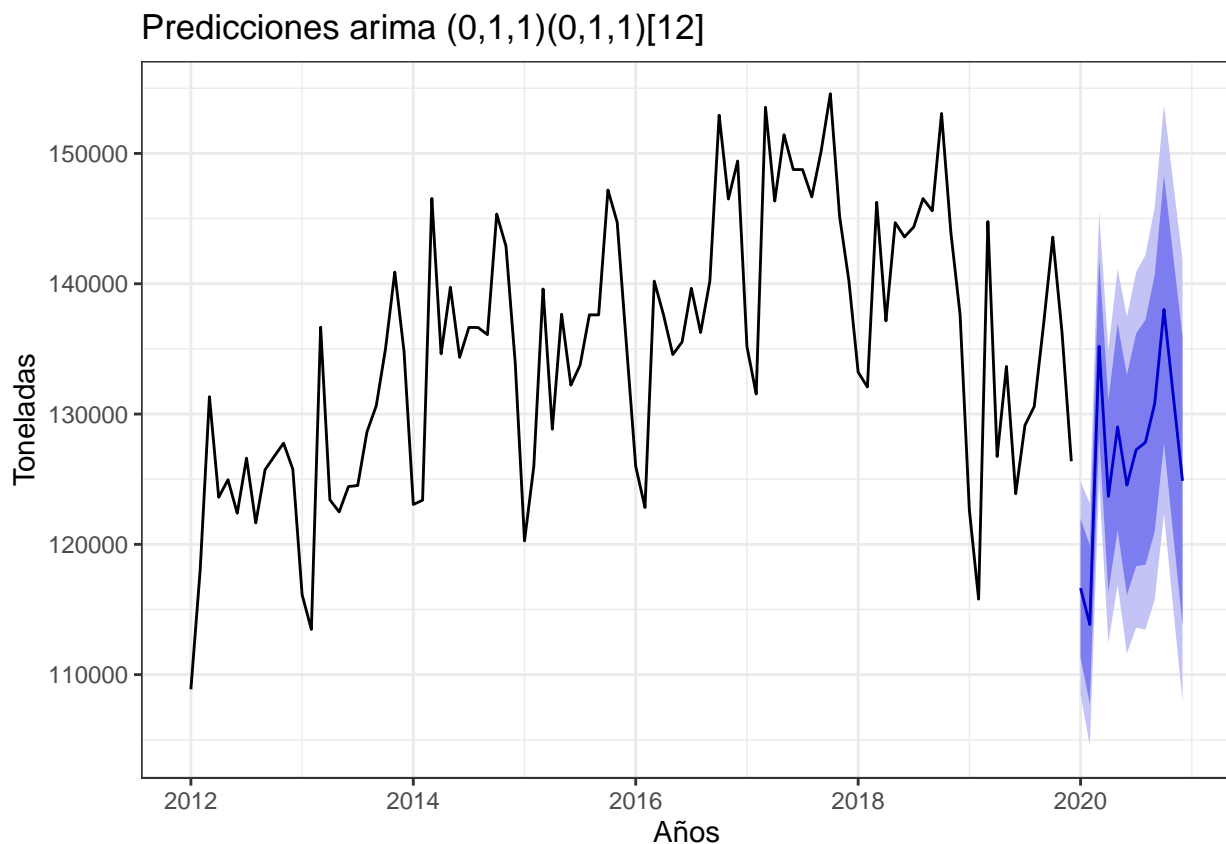
A continuación se generaran las predicciones de los modelos.

## 7. Predicciones

Se realiza el calculo de las predicciones con sus intervalos de confianza.

En primer lugar se genera un gráfico con las predicciones basadas en el modelo arima (0,1,1)(0,1,1)[12]

```
#Plot con predicciones
autoplot(forecast(arima1, h = 12), colour = "#5fa2ce") +
  ggtitle('Predicciones arima (0,1,1)(0,1,1)[12]') +
  xlab("Años") + ylab("Toneladas") +
  theme_bw()
```



Adicionalmente se genera una tabla con las predicciones e intervalos de confianza para el modelo Arima.

```
# Tabla predicciones
knitr::kable(forecast(arima1, h=24), digits = 2, caption="Predicciones")
```

Cuadro 4: Predicciones

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2020	116642.9	111310.89	121974.8	108488.32	124797.4
Feb 2020	113848.0	107766.58	119929.3	104547.29	123148.6
Mar 2020	135193.2	128445.19	141941.3	124872.98	145513.5
Apr 2020	123686.1	116331.50	131040.6	112438.23	134933.9

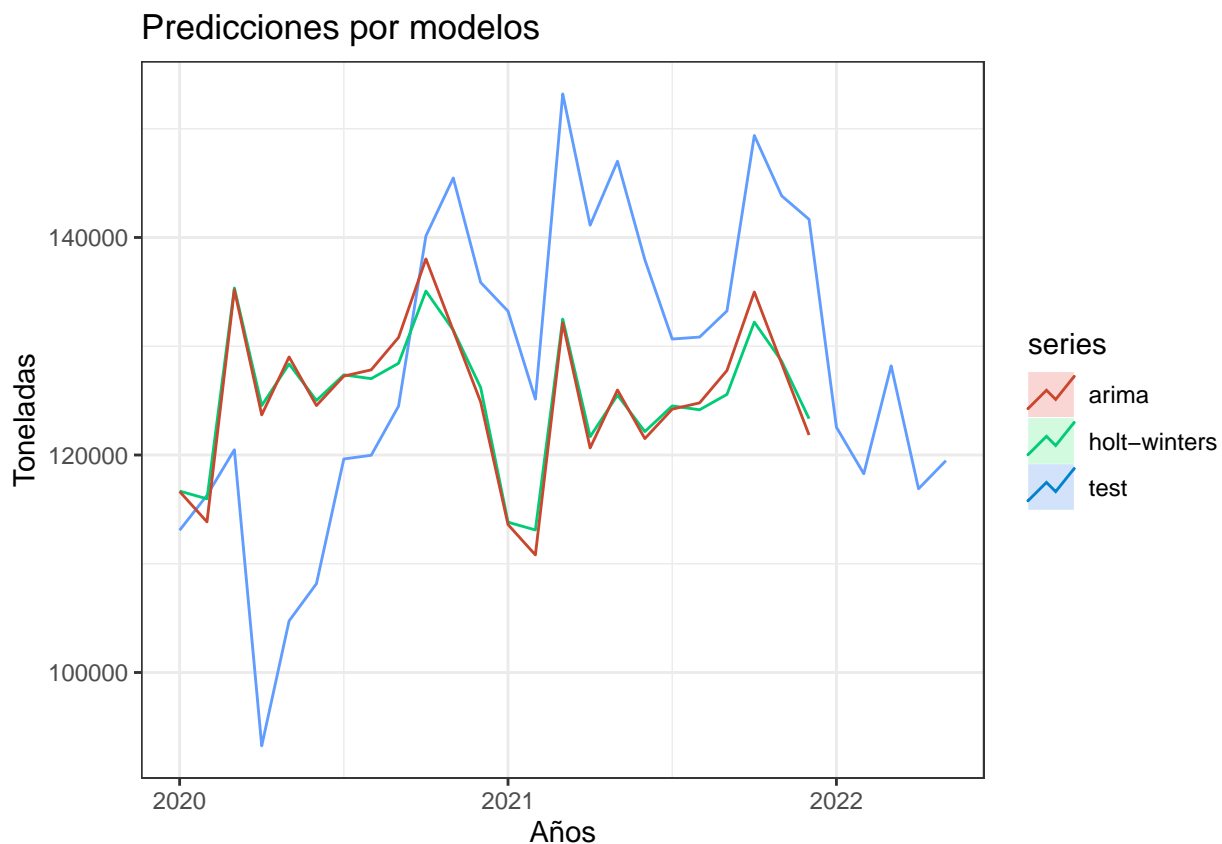
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
May 2020	129014.9	121100.19	136929.6	116910.39	141119.4
Jun 2020	124545.1	116107.28	132982.8	111640.60	137449.5
Jul 2020	127255.3	118325.04	136185.5	113597.66	140912.9
Aug 2020	127827.5	118430.61	137224.5	113456.18	142198.9
Sep 2020	130807.9	120966.35	140649.4	115756.56	145859.2
Oct 2020	138026.2	127759.36	148293.1	122324.41	153728.1
Nov 2020	131439.1	120763.79	142114.4	115112.63	147765.5
Dec 2020	124873.2	113804.54	135941.9	107945.15	141801.2
Jan 2021	113604.6	101576.15	125632.9	95208.70	132000.4
Feb 2021	110809.6	98164.98	123454.3	91471.30	130148.0
Mar 2021	132154.9	118922.68	145387.2	111917.96	152391.9
Apr 2021	120647.7	106852.91	134442.6	99550.37	141745.1
May 2021	125976.6	111641.23	140311.9	104052.56	147900.6
Jun 2021	121506.7	106650.51	136362.9	98786.10	144227.4
Jul 2021	124217.0	108857.52	139576.4	100726.73	147707.2
Aug 2021	124789.2	108942.54	140635.9	100553.82	149024.6
Sep 2021	127769.5	111450.16	144088.9	102811.21	152727.9
Oct 2021	134987.9	118209.14	151766.7	109327.00	160648.8
Nov 2021	128400.8	111174.84	145626.7	102056.00	154745.5
Dec 2021	121834.9	104173.13	139496.6	94823.57	148846.2

Tomando como ejemplo el mes de enero de 2021, se puede predecir que con un intervalo de confianza del 95% las toneladas transportadas vía cargo aéreo desde Schiphol,NL estarán entre 95.208 y 132.000 ton.

## 8. Conclusiones

Considerando ambos modelos (suavizado y Arima) se realiza una comparación con el set de datos `test` que ha sido reservado para estos efectos.

```
autoplot(cargo_test, series = "test") +  
  autolayer(cargo_hw, series = "holt-winters", PI = FALSE) +  
  autolayer(forecast(arima1, h=24), series = "arima", PI = FALSE) +  
  ggtitle('Predicciones por modelos') +  
  xlab("Años") + ylab("Toneladas") +  
  theme_bw()
```



La predicción para este gráfico se ha mostrado para un periodo de dos años, donde se observan diferencias importantes que se pueden clasificar en dos grupos:

La primera diferencia predictiva entre Arima-Hw versus test es al principio de la pandemia COVID-19 donde las predicciones estimadas son mas elevadas que el real.

La segunda diferencia tiene que ver con que a partir de mediados del 2021 las toneladas transportadas crece de forma significativa, superando las expectativas de las predicciones, sin llegar a alcanzar la linea azul de test.

Como nota al margen la serie train presenta un descenso dramático desde el ultimo cuarto del 2021 en adelante y repunta a partir del 2022. Si se extiende la predicción, los datos tanto de holt winters y Arima no coinciden en absoluto. Si se revisan las notas del set de datos original desde CBS indican que los datos de 2020 en adelante son provisionales y están sujetos a revisiones y/o

cambios.

Considerando estas particularidades por la situación excepcional desde el 2020 en adelante en contexto de pandemia, guerra, crisis en los aeropuertos y en especial para Schiphol se infiere que las predicciones para este modelo probablemente no coincidan y puedan mostrar varias inexactitudes.

Si se revisan las predicciones para 24 meses con ambos métodos, el modelo Arima es el que mas se “acercaría” a los datos test por lo que provisionalmente seria el modelo ganador para esta practica.

Como prueba y tomando como base para predecir el mes de enero de 2021 (tal como en las conclusiones del suavizado y Arima) se muestra una tabla resumen con las predicciones versus la cifra real de toneladas transportadas en Schiphol, NL

```
# Transforma a df
prediccion_hw <- as.data.frame(cargo_hw)
prediccion_arima <- as.data.frame(forecast(arima1, h = 24))
datx <- as.xts(cargo_test)
df <- data.frame(date=index(datx), coredata(datx))

#Predicciones enero 2021
knitr::kable(prediccion_hw[13,], caption = "Prediccion hw enero 2021")
```

Cuadro 5: Prediccion hw enero 2021

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2021	113820.6	101457.3	126184	94912.55	132728.7

```
knitr::kable(prediccion_arima[13,], caption = "Prediccion Arima enero 2021")
```

Cuadro 6: Prediccion Arima enero 2021

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2021	113604.5	101576.1	125632.9	95208.7	132000.4

```
knitr::kable(df[13,], caption = "Toneladas transportadas (real)")
```

Cuadro 7: Toneladas transportadas (real)

date	Ton
13 Jan 2021	133222

Como comprobación final se generan las tablas de exactitud (accuracy) y la comparación de los modelos con su AIC.

```
knitr::kable(accuracy(cargo_hw), digits = 2, caption="Accuracy Holt-Winters")
```

Cuadro 8: Accuracy Holt-Winters

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-418.4	3542.45	2901.26	-0.36	2.17	0.44	-0.06

```
knitr::kable(accuracy(arima1), digits = 2, caption="Accuracy Arima")
```

Cuadro 9: Accuracy Arima

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-438.2	3818.46	2946.78	-0.39	2.18	0.45	-0.05

```
print(cargo_hw$model$aic)
```

```
[1] 2041.312
```

```
print(arima1$aic)
```

```
[1] 1629.986
```

En el caso de accuracy, observando los valores por ejemplo el índice mas bajo en error y en MAPE lo obtiene el modelo Arima, al ser un MAPE menor a 10 indica un buen grado de exactitud en las predicciones.

Si se debe elegir un modelo independiente de la excepcionalidad del contexto actual y los resultados de las predicciones vistos anteriormente, si se observan los valores AIC de ambos modelos, el *Arima* muestra un número menor, sumado a los resultados del gráfico de predicciones y tablas, se confirma su elección para realizar las predicciones correspondientes.