Minería de Datos y Modelización Predictiva (IV)

Fernández Hernández, Alberto. 54003003S

1/29/2021

1. Introducción. Presentación de la serie a analizar

El objetivo del presente proyecto consiste en el análisis y modelado predictivo de una serie temporal con la **estimación de ventas mensuales en tiendas de ropa en Estados Unidos**, conocido como *Monthly Retail Sales*. Los datos han sido obtenidos del repositorio de Investigación Económica de la Reserva Federal del Banco de Saint Louis. [1]

```
ventas.ropa <- read_excel("retail_sales.xls")
```

El fichero de datos contiene un total de dos variables: *observation_date*, con la fecha de estimación, así como las ventas o *sales* (en millones de dólares). Dicho conjunto abarca un total de 168 observaciones mensuales, **desde enero del año 2006 hasta diciembre del año 2019**:

```
min(ventas.ropa$observation_date) # Fecha min: Enero 2006
```

```
## [1] "2006-01"
```

```
max(ventas.ropa$observation_date) # Fecha max: Diciembre 2019
```

```
## [1] "2019-12"
```

```
# Analizamos las 6 primeras filas
head(ventas.ropa)
```

```
## # A tibble: 6 x 2
##   observation_date sales
##   <chr>            <dbl>
## 1 2006-01          12893
## 2 2006-02          14474
## 3 2006-03          16386
## 4 2006-04          16848
## 5 2006-05          17103
## 6 2006-06          16505
```

Por otro lado, analizando brevemente las estadísticas de ventas podemos comprobar que **existe un cierto contraste en los valores de ventas mínimo y máximo**. A modo de ejemplo, el valor de la mediana nos indica la presencia de meses en los que las ventas se sitúan por debajo de los 20 mil millones de dólares, situación contraria en otros meses, donde la estimación de ventas aumenta considerablemente, hasta llegar incluso a los 34 mil millones (valor máximo). No obstante, con tan solo el *summary* no podemos aventurarnos a asegurar que la componente presenta estacionalidad, con valores mínimos y máximos de ventas anuales, por lo que necesitaremos la representación gráfica para comprobarlo.

```
summary(ventas.ropa$sales)
```

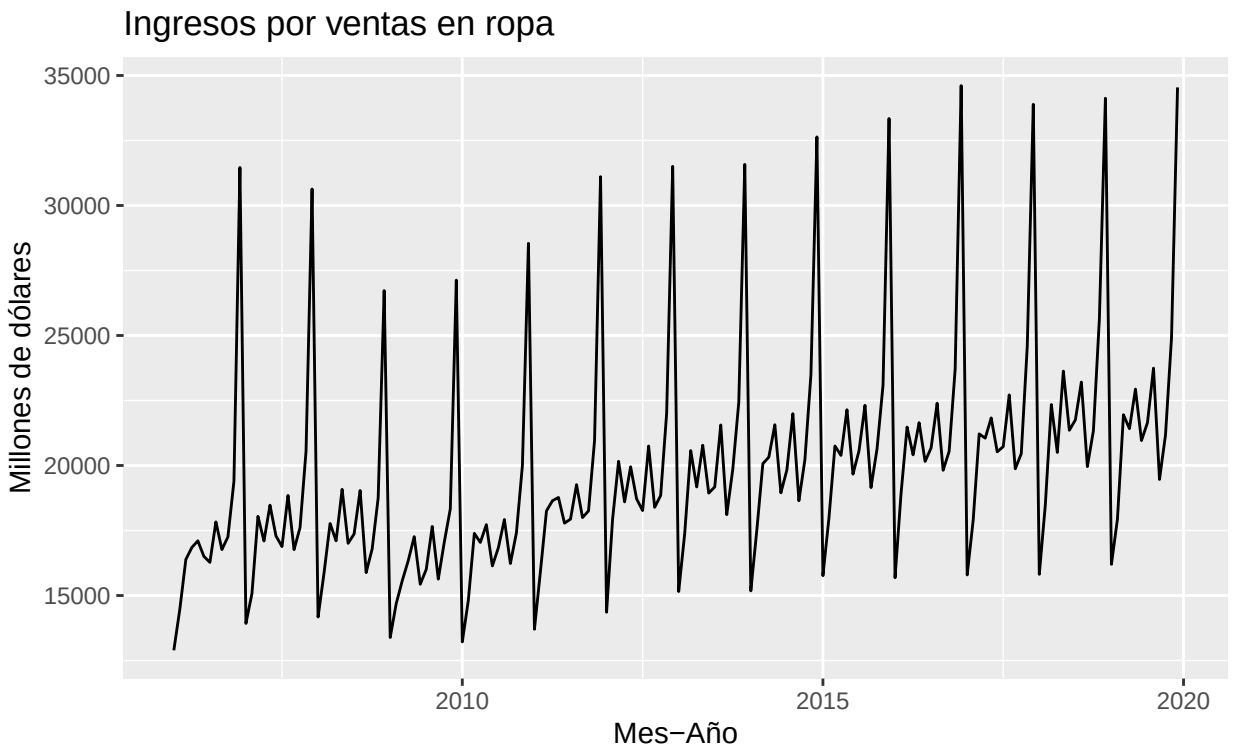
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12893   17103   18994   19896   21341   34611
```

[1] https://fred.stlouisfed.org/series/MRTSSM4481USN

2. Representación gráfica y descomposición de la serie

De forma previa a los modelos predictivos, debemos representar gráficamente la serie temporal con el objetivo de estudiar su características como estacionalidad, tendencia y estacionariedad:

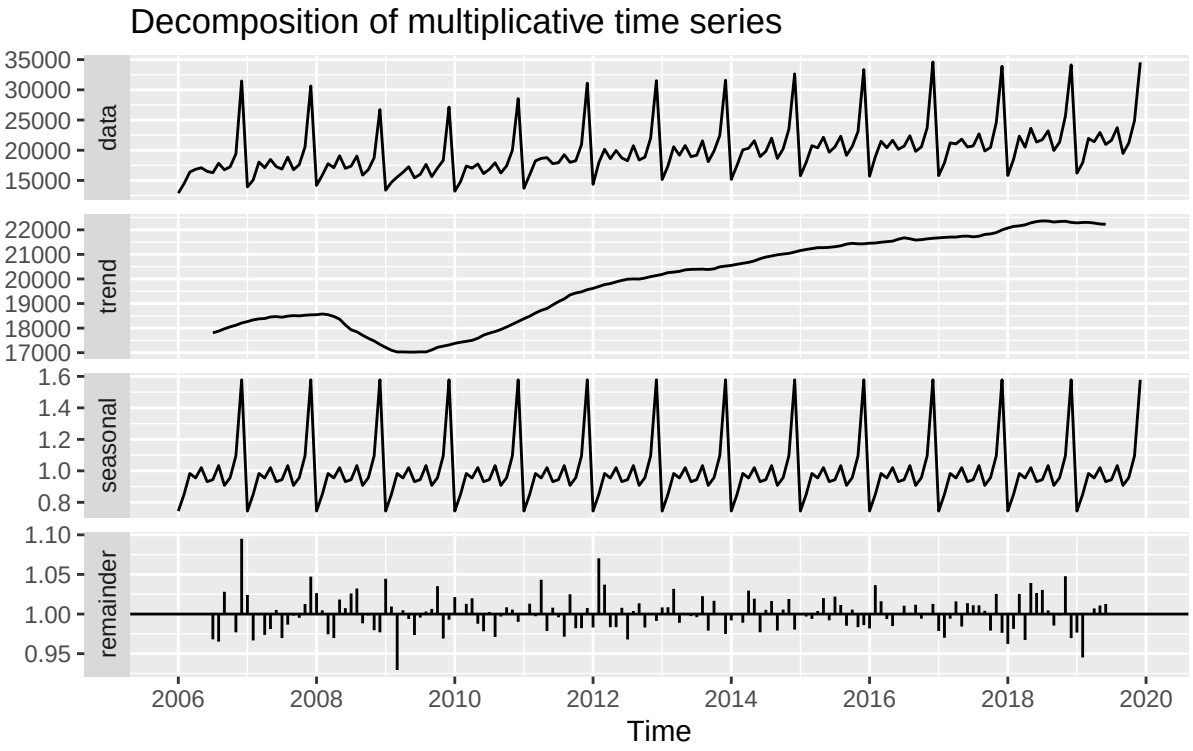
```
ventas.ropa.ts <- ts(ventas.ropa[,-1], start=c(2006,1), frequency=12)
ventas.ropa.test <- window(ventas.ropa.ts,start=c(2019,1), end=c(2019,12))
autoplot(ventas.ropa.ts) +  ggtitle("Ingresos por ventas en ropa") +
  xlab("Mes-Año") +  ylab("Millones de dólares")
```



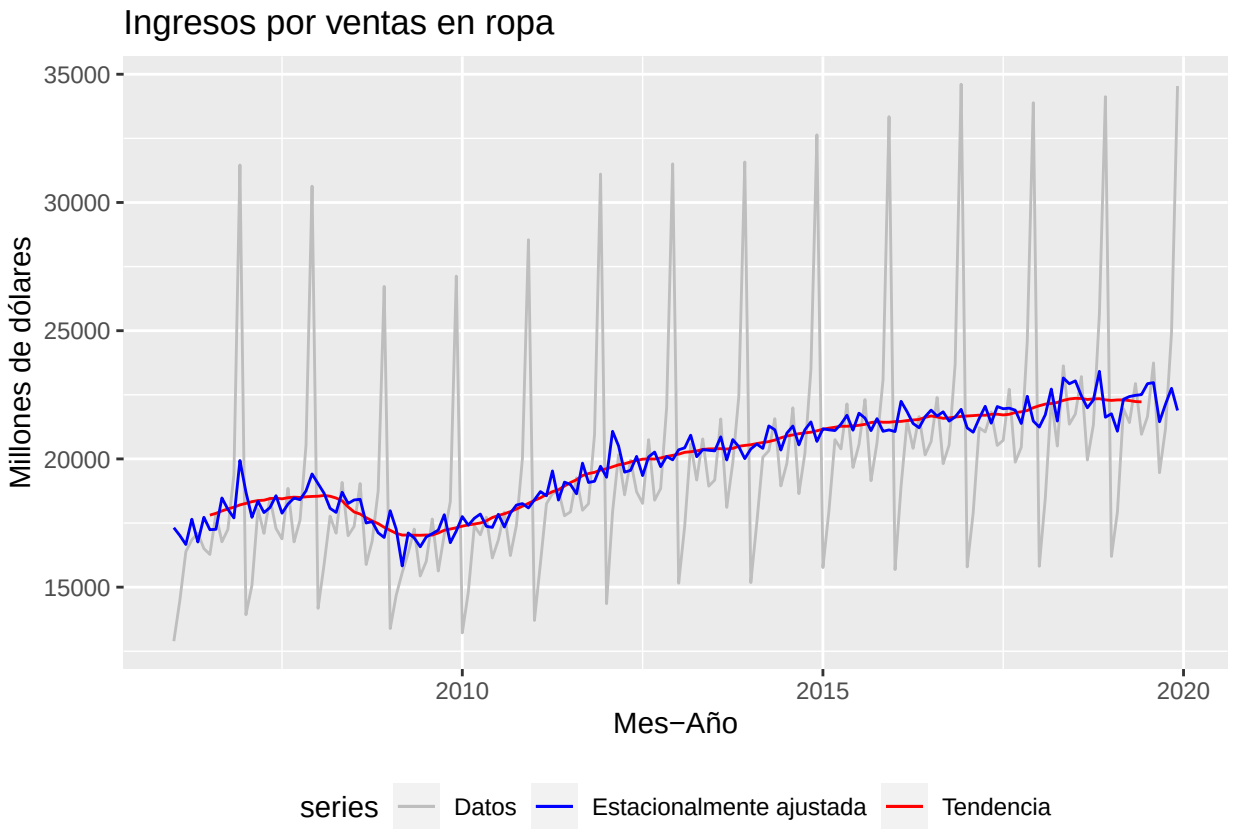
Analizando la serie temporal, podemos extraer varias características: en primer lugar, la serie comienza con un decrecimiento en los ingresos desde el año 2006 hasta el año 2010, aproximadamente. A continuación, **la tendencia es prácticamente ascendente (media no constante)** hasta prácticamente el año 2019, momento en el que parece estabilizarse la serie. En relación con la varianza, tampoco es constante, presentando un aumento en la variabilidad de las ventas a lo largo de los años, tal y como podemos comprobar en el gráfico de la serie, donde la amplitud entre los valores de venta mínimo y máximo aumentan con el transcurso de los años, es decir, desde el año 2009-2010 existe cada vez un mayor constraste entre periodos donde se acentúan las ventas y momentos en los que se reduce al mínimo. Por tanto, dando que la varianza aumenta con el tiempo, **la serie presenta un esquema multiplicativo**.

Por otro lado, si analizamos la descomposición de la serie, podemos evidenciar tanto la tendencia ascendente desde el año 2009-2010 como el aumento de la varianza con el paso de los años:

```
ventas.ropa.comp<- decompose(ventas.ropa.ts,type=c("multiplicative"))
autoplot(ventas.ropa.comp)
```



Así como la tendencia claramente ascendente en el número de ventas, pasando de una media de más de 15.000 millones de dólares en el año 2010 a más de 20.000 millones en el año 2019:



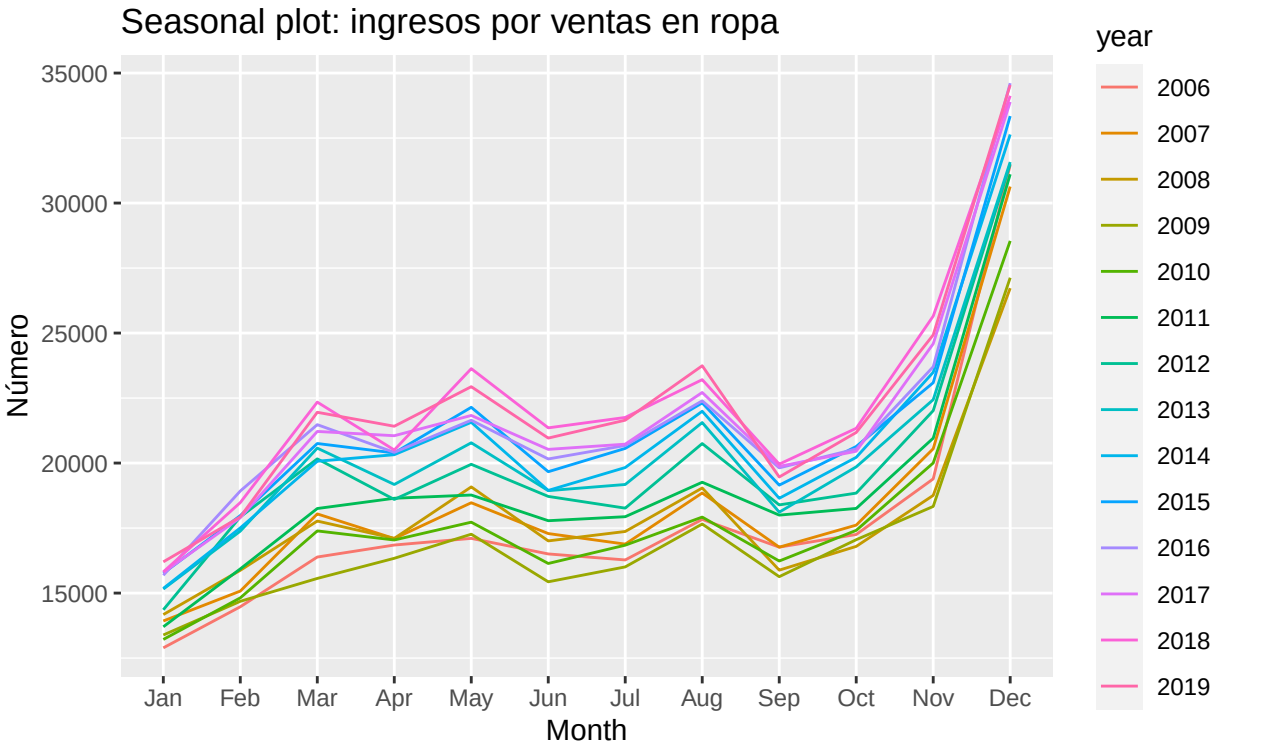
Por otro lado, cabe destacar la estacionalidad que se produce anualmente, donde un mes (probablemente diciembre) concentra el mayor número de ventas, mientras que en el mes siguiente (enero) los ingresos se reducen al mínimo. De

hecho, si analizamos los valores de la componente estacional:

```
comp.est <- data.frame(t(ventas.ropa.comp$seasonal[c(1:12)]))
```

```
##    Jan   Feb   Mar   Apr  May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
##  0.745 0.851 0.983 0.955 1.02 0.931 0.944 1.033 0.907 0.957 1.096 1.578
```

Efectivamente, observamos que **los ingresos por ventas en ropa son un 57.8 % superior a la media anual**, mientras que **los meses de enero y febrero concentran los porcentajes más bajos de ventas, con un 25.5 y un 14.9 % inferior en relación a la media anual, respectivamente**. Por otro lado, si analizamos las tendencias anuales:



Podemos comprobar, nuevamente, la tendencia ascendente de los ingresos por ventas en ropa con el transcurso del tiempo, siendo los últimos años, 2018 y 2019, los que presentan el mayor número de ingresos.

Una vez realizado el primer análisis de la serie, de cara a los modelos tanto de suavizado exponencial como ARIMA **reservaremos los últimos datos observados (dado que la estacionalidad es anual escogemos los ingresos del año 2019)** como conjunto de prueba para comprobar la eficacia de los métodos de predicción:

```
ventas.ropa.ts.transformado <- window(ventas.ropa.ts,start=c(2006,1),end=c(2018,12))
ventas.ropa.test <- window(ventas.ropa.ts,start=c(2019,1), end=c(2019,12))
```

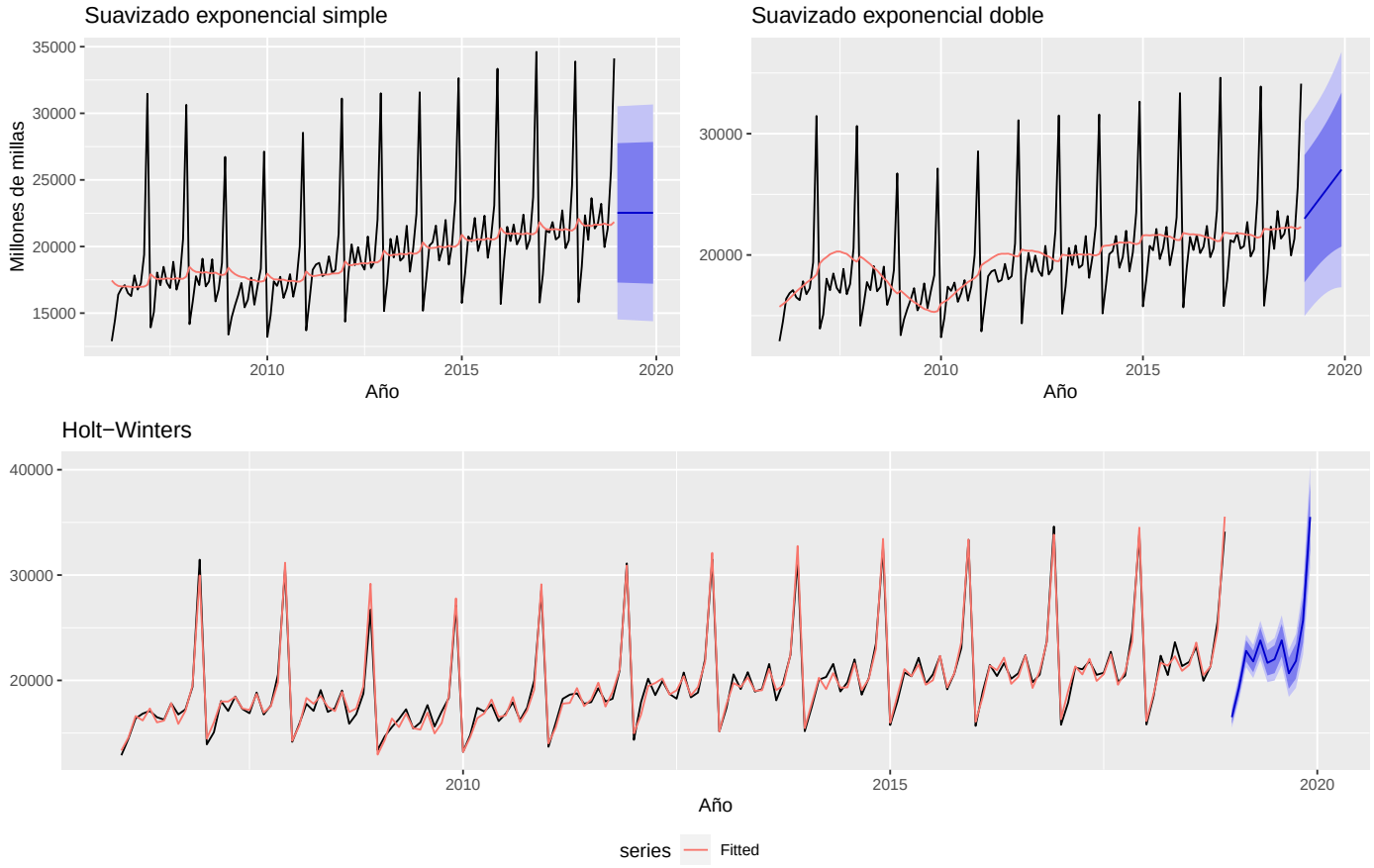
3. Modelo de suavizado exponencial

Para determinar el mejor modelo de suavizado exponencial, realizamos una comparación de la precisión entre los diferentes modelos, tanto de alisado simple, doble, como de *Holt-Winters*. Dado que el conjunto de prueba empleado contiene los ingresos por venta del año 2019, la predicción calculada será para los siguientes 12 meses (h = 12). En el caso del modelo de *Holt-Winters*, dado que la serie es multiplicativa, debemos indicarlo a través del parámetro *seasonal*:

```
ventas.ropa.ss <- ses(ventas.ropa.ts.transformado, h=12) # Alisado simple
ventas.ropa.holt <- holt(ventas.ropa.ts.transformado, h=12) # Alisado doble (Holt)
ventas.ropa.hw <- hw(ventas.ropa.ts.transformado, h=12, seasonal="multiplicative") # Alisado Holt-Winters
estadisticas.suavizado <- rbind(round(accuracy(ventas.ropa.ss),3), round(accuracy(ventas.ropa.holt),3),
                          round(accuracy(ventas.ropa.hw),3))
```

Table 1: Precisión de los modelos de suavizado

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	AIC	BIC
Alisado Simple	581.130	4054.864	2392.677	-0.270	11.176	3.232	-0.077	3385.771	3394.921
Alisado Doble	0.774	4042.258	2604.200	-3.115	12.490	3.517	-0.053	3388.800	3404.049
Holt-Winters	-12.558	585.568	460.540	-0.074	2.347	0.622	-0.070	2796.921	2848.768



Analizando tanto la tabla como la salida gráfica, sin duda alguna **el método *Holt-Winters* ofrece un mejor modelo prácticamente en todos los sentidos**, desde los errores medios más bajos hasta valores AIC y BIC significativamente menores en comparación con los modelos de alisado simple y doble (2796 y 2848, respectivamente). Por otro lado, la salida gráfica evidencia la clara ventaja del modelo *Holt-Winters*: mientras que el modelo de alisado simple devuelve la misma predicción para los siguientes 12 meses y el modelo de alisado doble realiza una predicción meramente lineal, el método de *Holt-Winters* se aproxima en mejor medida a los valores de la serie original, lo que se traduce además en intervalos de confianza más cerrados, tal y como podemos comprobar en los gráficos anteriores.

Por tanto, dado el menor error, AIC y BIC obtenido, así como una mejor aproximación a la serie original, **elegimos como modelo ganador al obtenido por el método de *Holt-Winters***:

```
ventas.ropa.hw
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2019       16513.22 15863.89 17162.54 15520.16 17506.28
## Feb 2019       19312.14 18511.94 20112.35 18088.33 20535.95
## Mar 2019       22804.55 21798.67 23810.43 21266.18 24342.91
## Apr 2019       21802.18 20770.20 22834.16 20223.90 23380.45
## May 2019       23799.02 22583.25 25014.80 21939.65 25658.39
## Jun 2019       21664.49 20465.84 22863.15 19831.31 23497.68
```

```
## Jul 2019       22010.00 20688.86 23331.15 19989.49 24030.52
## Aug 2019       23808.24 22257.44 25359.04 21436.50 26179.99
## Sep 2019       20668.68 19208.86 22128.50 18436.07 22901.28
## Oct 2019       21844.03 20173.60 23514.46 19289.33 24398.73
## Nov 2019       25698.13 23574.63 27821.63 22450.51 28945.74
## Dec 2019       35518.77 32354.38 38683.16 30679.26 40358.28
```

```
ventas.ropa.hw$model$par[1:3] # Obtenemos los parametros alpha, beta y gamma
```

```
##      alpha       beta      gamma
## 0.27962247 0.05314254 0.32077280
```

En base a los parámetros alfa, beta y gamma obtenidos, y dado que se trata de un modelo multiplicativo, la expresión del modelo final es la siguiente:

$$L_t = 0.2796\frac{x_t}{S_{t-s}} + (1-0.2796)(L_{t-1}+b_{t-1})$$
$$b_t = 0.0531(L_t - L_{t-1}) + (1-0.0531)b_{t-1}$$
$$S_t = 0.3207\frac{x_t}{L_t} + (1-0.3207)S_{t-s}$$
$$\hat{x}_{t+1} = (L_t + b_t)S_{t+1-s}$$

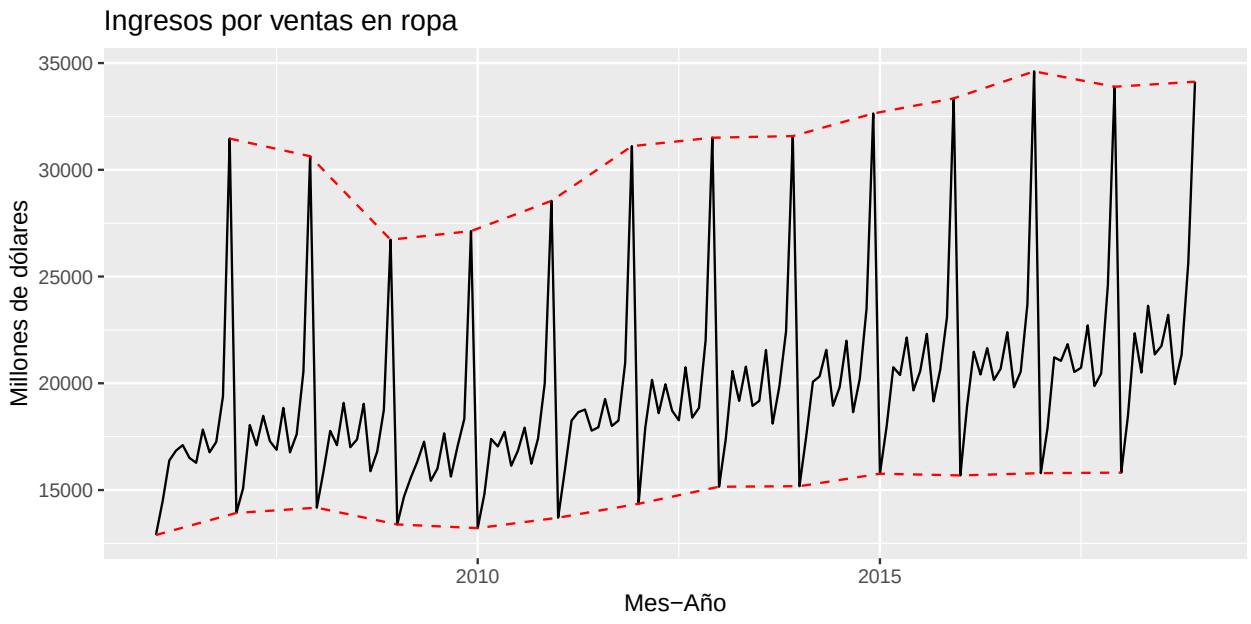
Donde L_t representa la componente de nivel de la serie temporal, b_t la tendencia y S_t la estacionalidad.

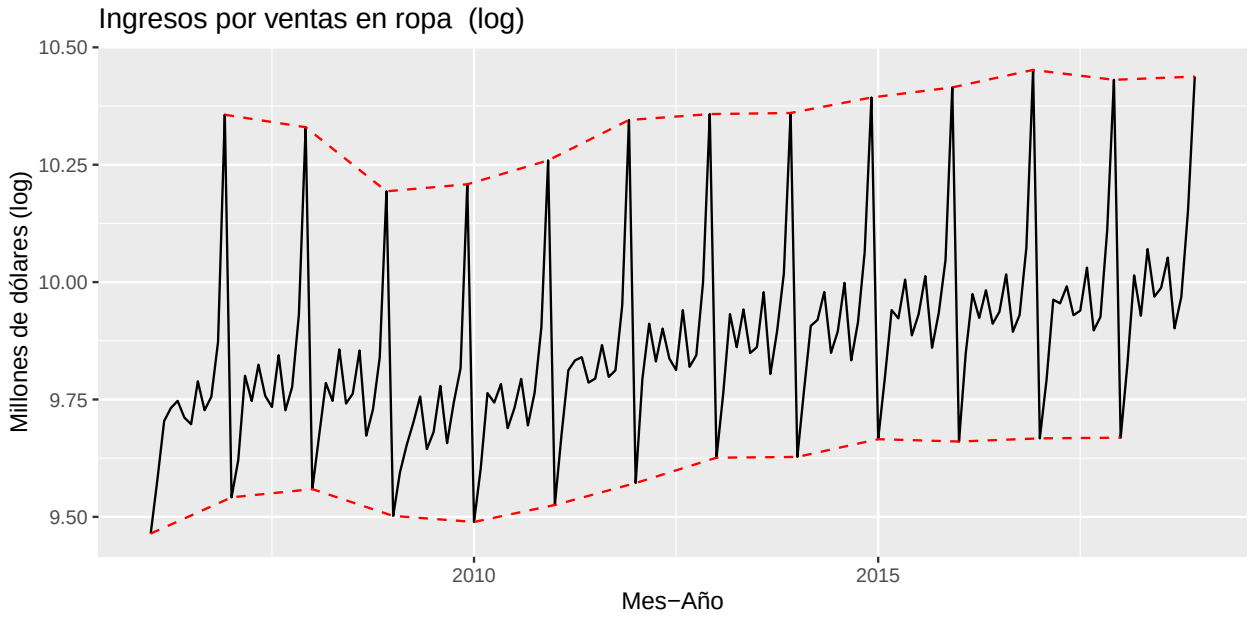
4. Modelo ARIMA

4.1 Transformaciones de la serie temporal

De forma previa a la elaboración del modelo ARIMA, así como a las funciones de autocorrelación y autocorrelación parcial, cabe recordar que la serie original **no es estacionaria en cuanto a varianza se refiere**, por lo que debemos comprobar si es posible, por medio de transformaciones Box-Cox, estabilizar dicha variabilidad. Como primera opción, realizamos una de las más comunes: la logarítmica (es decir, $\lambda = 0$).

```
serie.temp.original <- autoplot(ventas.ropa.ts.transformado)
transf.box.cox <-  log(ventas.ropa.ts.transformado)
serie.temp.log <- autoplot(transf.box.cox)
```





Como podemos observar en ambos gráfico, la transformación logarítmica **parece estabilizar la variabilidad de la serie, especialmente a partir del año 2010**, donde el crecimiento de la varianza ya no es tan significativo en comparación con la serie original, aunque conservando su tendencia ascendente. Por otro lado, la librería *forecast* dispone de una función denominada *BoxCox.lambda* que permite obtener el coeficiente *lambda* óptimo para la transformación de la serie. Dispone de dos métodos: *loglik*, que elige el valor de *lambda* que maximice la verosimilitud de la transformada con respecto a un modelo lineal; y *guerrero*, que escoge el parámetro *lambda* que minimice el coeficiente de variación para cada una de las sub-series del conjunto de datos. En caso de que λ sea 1, implicaría que la transformación no es necesaría:

```
BoxCox.lambda(ventas.ropa.ts.transformado, method = "guerrero")
```

```
## [1] 0.3614745
```

```
BoxCox.lambda(ventas.ropa.ts.transformado, method = "loglik")
```

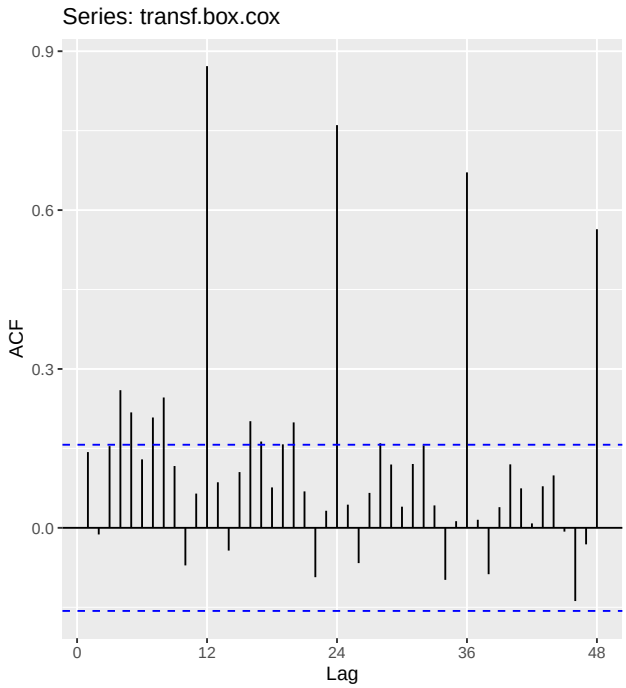
```
## [1] 0.05
```

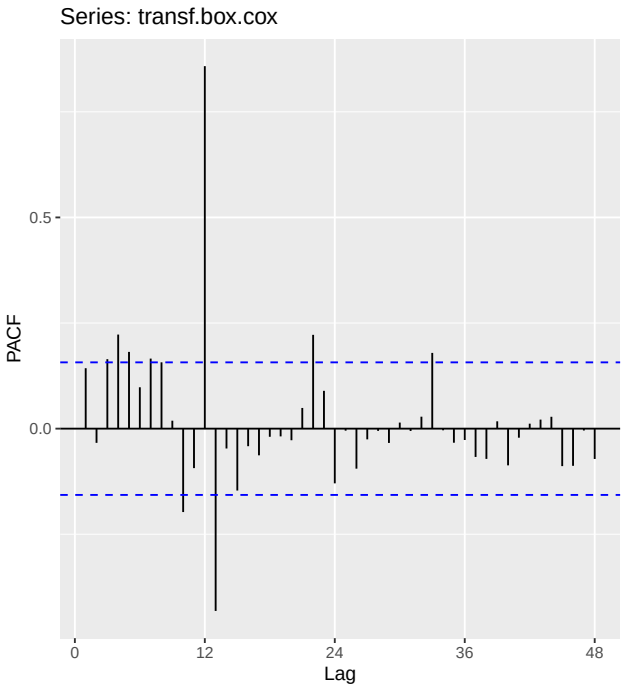
Como podemos observar, el valor de *lambda* en ambos casos es más cercano a 0 (0.36 y 0.05, respectivamente), **lo que evidencia nuevamente la necesidad de transformar la serie original**. No obstante, se han comparado gráficamente la serie transformada logarítmica con las series obtenidas a partir de los valores *lambda* anteriores, pero la diferencia no es muy significativa, por lo que en adelante se ha decido trabajar con la serie logarítimica.

4.2 Funciones de autocorrelación y autocorrelación parcial

Una vez transformada la serie temporal, representamos gráficamente tanto la función de autocorrelación como de autocorrelación parcial, con el objetivo no solo de comprobar si la serie es o no estacionaria, sino además para determinar el tipo de modelo en función de los retardos que son significativamente distintos de cero. Dado que la serie presenta estacionalidad anual, se ha decido calcular las autocorrelaciones hasta el retardo 48, es decir, hasta 4 años:

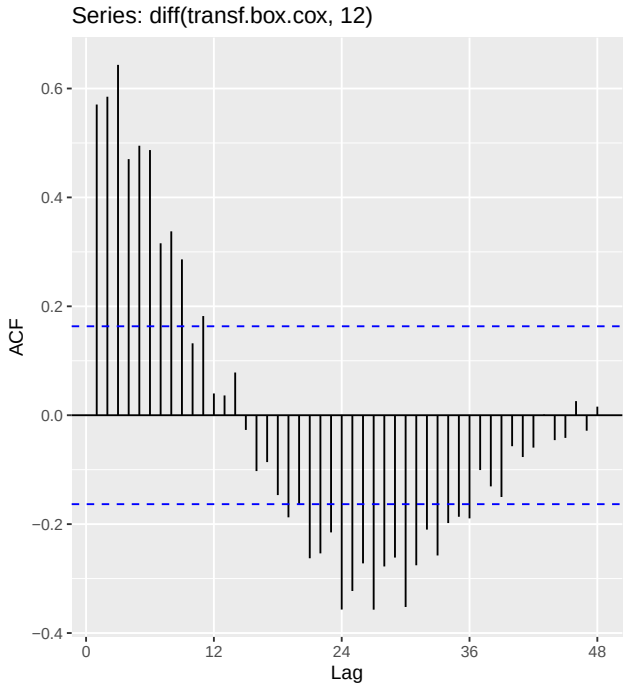
```
ggAcf(transf.box.cox, lag = 48) # Funcion de Autocorrelacion
ggPacf(transf.box.cox, lag = 48) # Funcion de Autocorrelacion Parcial
```

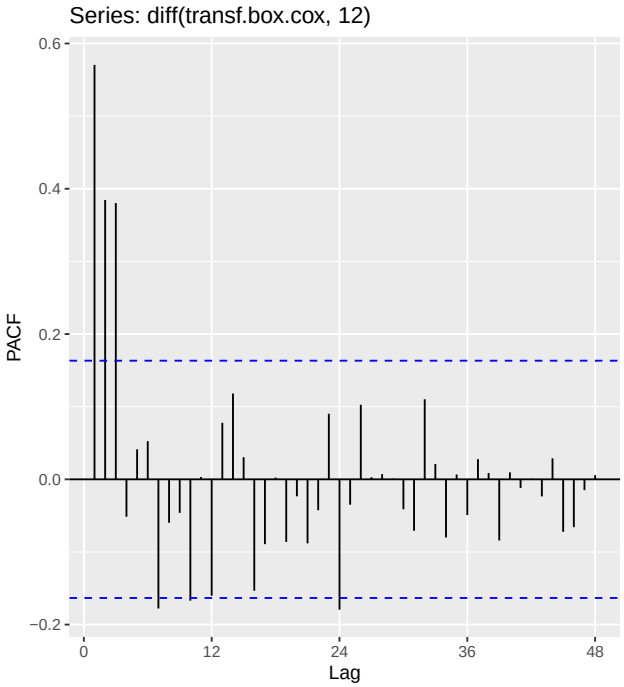




En los primeros pasos, debemos fijarnos sobretodo en la Función de Autocorrelación. En ella, detectamos la estacionalidad mencionada en los primeros apartados: por un lado, existe un patrón de autocorrelación que se repite anualmente y disminuye conforme aumentan los retardos. De hecho, el decrecimiento lento se aprecia mejor en los **retardos múltiplos de la estacionalidad**: 12, 24, 36 y 48, indicativo de que la serie presenta estacionalidad y no es estacionaria. Por tanto, para eliminar dicha estacionalidad **debemos aplicar una diferenciación de orden 12 (anual)**:

```
ggAcf(diff(transf.box.cox, 12), lag = 48) # Funcion de Autocorrelacion
ggPacf(diff(transf.box.cox, 12), lag = 48) # Funcion de Autocorrelacion Parcial
```

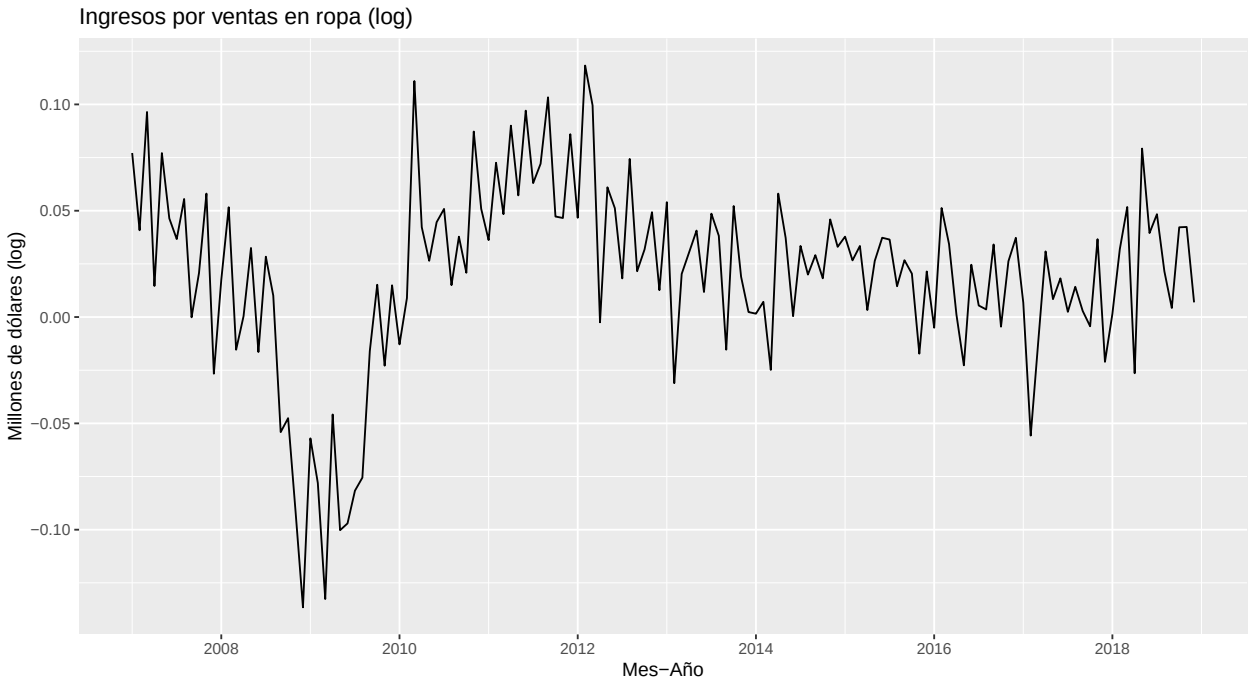




Una vez aplicada la diferenciación estacional, la serie continúa sin ser estacionaria, principalmente por un motivo: **sigue exitiendo un decrecimiento lento de los valores de autocorrelación**, tal y como podemos observar en

la función ACF, es decir, la media no es constante. Una forma de comprobarlo sería gráficamente:

```
autoplot(diff(transf.box.cox, 12)) +  ggtitle("Ingresos por ventas en ropa (log)") +
  xlab("Mes-Año") +  ylab("Millones de dólares (log)")
```



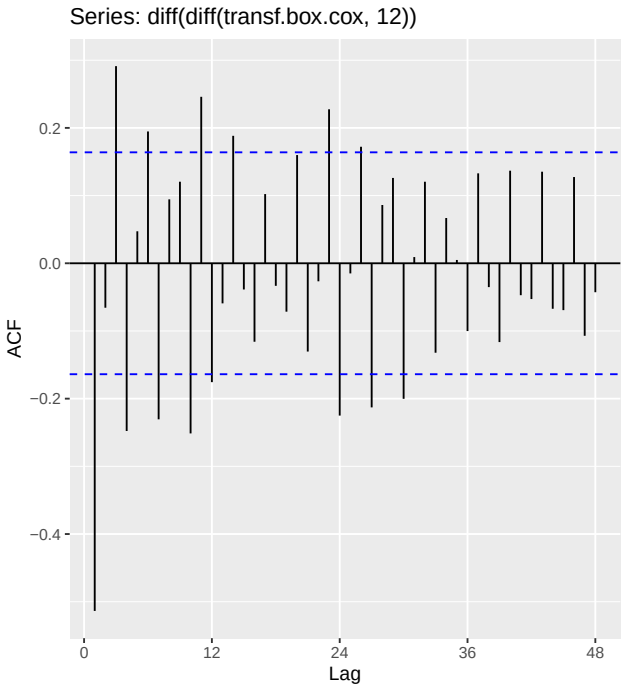
Como podemos comprobar nuevamente, la estacionalidad anual se ha visto reducida considerablemente. No obstante, debido al decrecimiento en el número de ventas ocasionado entre los años 2007-2010 (periodo de recesión económica), provoca que la media no sea constante y, por tanto, **no podemos asegurar que la serie sea estacionaria en sentido débil**: si escojo dos series en cualquier instante de tiempo, sus medias y varianzas deberán ser constantes, pero el decrecimiento producido entre estos años lo impide. Además, incluso contrastes de hipótesis como el *kpss* **rechazan la hipótesis nula de que la serie es estacionaria alrededor de una tendencia determinista**:

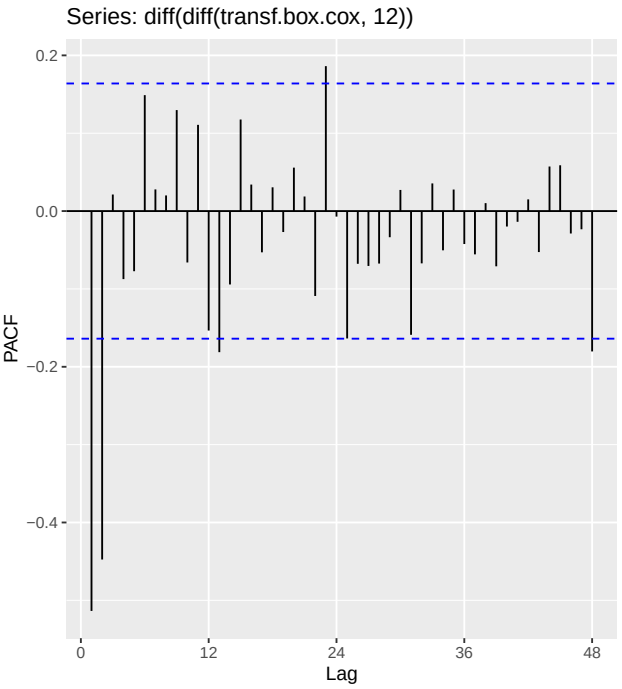
```
tseries::kpss.test(diff(transf.box.cox, 12), null = "Trend") # null = Hipotesis nula
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  diff(transf.box.cox, 12)
## KPSS Trend = 0.17318, Truncation lag parameter = 4, p-value = 0.02735
```

En el caso anterior, el p-valor obtenido es de 0.027, significativamente inferior a 0.05, por lo que rechazamos (al 95 % de confianza) la hipótesis nula de que la serie sea estacionaria en media. Por tanto, debemos aplicar una diferenciación a la componente regular de la serie:

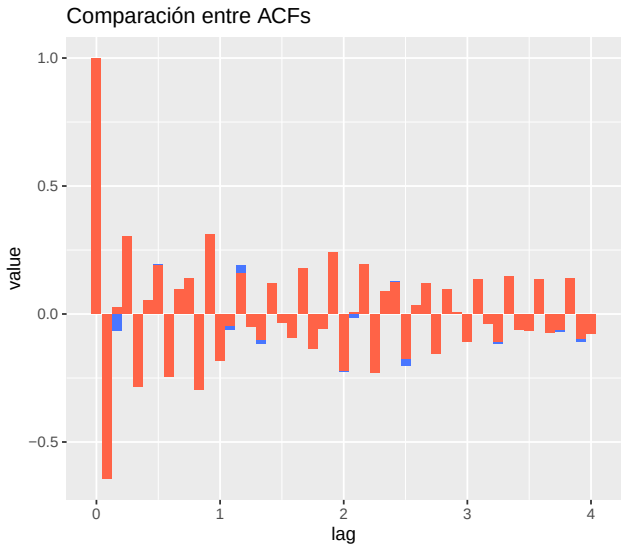
```
ggAcf(diff(diff(transf.box.cox, 12)), lag = 48) # Funcion de Autocorrelacion
ggPacf(diff(diff(transf.box.cox, 12)), lag = 48) # Funcion de Autocorrelacion Parcial
```

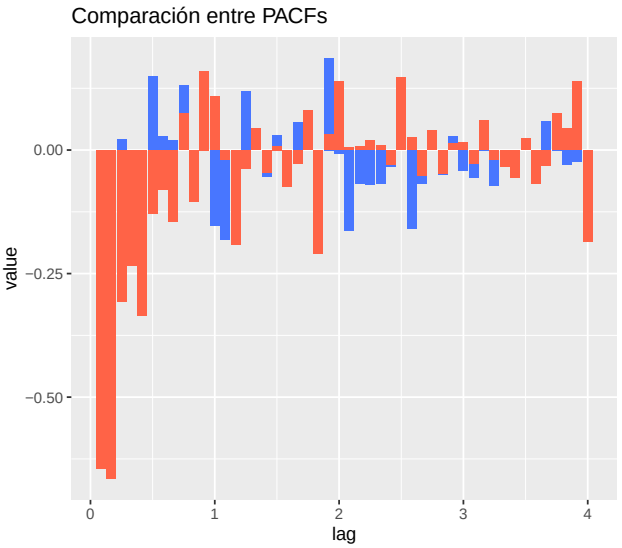


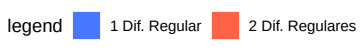


Como podemos comprobar, hemos conseguido eliminar "parcialmente" el decrecimiento de los retardos en la función de Autocorrelación. No obstante, pese a la diferenciación aplicada, continúa existiendo un decaimiento lento en los valores de autocorrelación. Por tanto, ¿Y si aplicamos una diferenciación de orden 2 a la parte regular para eliminar dicho decrecimiento? Analicemos el resultado, **comparando ambas funciones de autocorrelación, tanto con una diferenciación de orden 1 como de orden 2 en la parte regular**. Para ello, se ha elaborado una función denominada *comparar.autocorrelaciones*, que recibe como parámetros las series a utilizar, el tipo de función de autocorrelación a mostrar (ACF o PACF), así como el título del gráfico (empleando las funciones de *ggplot2*):

```
comparar.autocorrelaciones(diff(diff(transf.box.cox, 12)), # Funcion de Autocorrelacion
                           diff(diff(diff(transf.box.cox, 12))), tipo = "acf", "Comparación entre ACFs")
comparar.autocorrelaciones(diff(diff(transf.box.cox, 12)), # Funcion de Autocorrelacion Parcial
                           diff(diff(diff(transf.box.cox, 12))), tipo = "pacf", "Comparación entre PACFs")
```

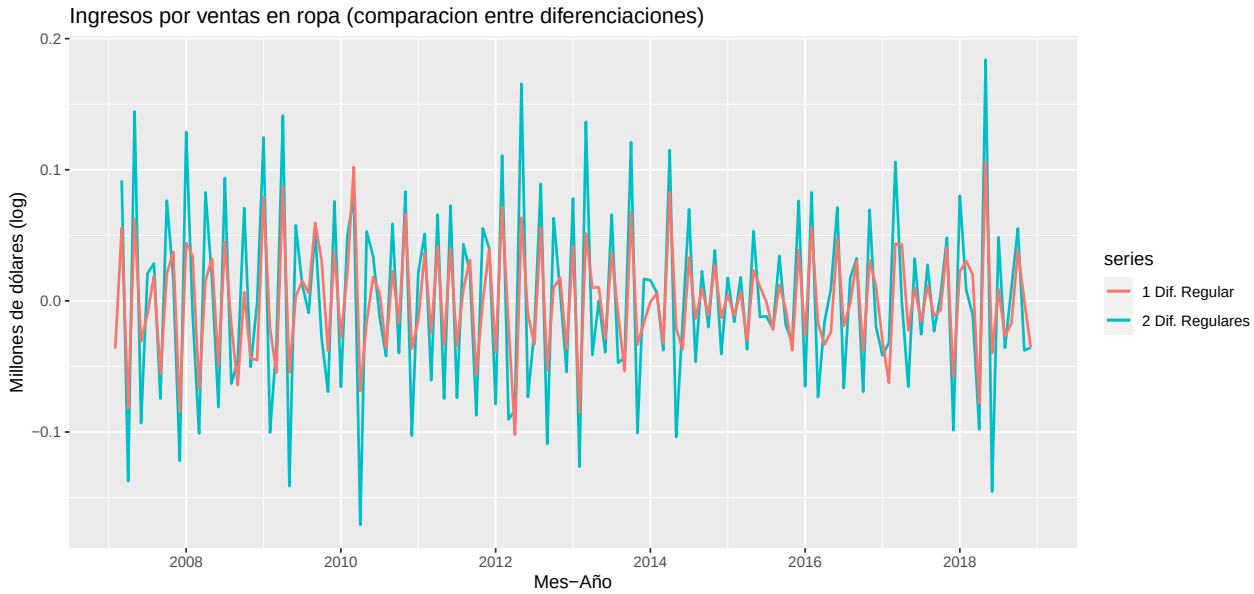






Pese a aplicar una diferenciación adicional, los valores de autocorrelación no se han visto prácticamente afectados:

en el diagrama de barras de la izquierda podemos comprobar que tan solo ve reducida la correlación en un pequeño subconjunto de retardos (donde la barra en rojo es menor a la barra azul). En el resto de casos, la autocorrelación no ha disminuido. Con respecto a la función de autocorrelación parcial, bien es cierto que muchas de las autocorrelaciones se ven reducidas. Sin embargo, debemos destacar una importante diferencia en la función de autocorrelación parcial: **mientras que con una diferenciación regular tan solo son los dos primeros retardos son significativos, con dos diferenciaciones el número aumenta hasta 5**. Además, incluso si comparamos ambas series observamos un gran contraste en la varianza, con una variabilidad mucho mayor empleando dos diferenciaciones:



Por tanto, dado que una diferenciación de orden 2 no reduce el decrecimiento en la función de autocorrelación parcial, además de aumentar incluso la variabilidad en la serie, optamos por una única diferenciación en la parte regular.

4.3 Selección de los parámetros del modelo

Una aplicadas las diferenciaciones, ya tenemos los coeficientes d y D del modelo ARIMA (1). Por tanto, debemos ajustar el resto de parámetros (autoregresivo y media móvil) tanto de la componente regular como estacional. Regresemos nuevamente con las funciones de autocorrelación y autocorrelación parcial: