## 1. Introducción: presentación de la serie a analizar

Para el presente trabajo se analizará la serie “Afluencia turística nacional mensual al Parque Arqueológico de Machu Picchu, 2004-2021” tomada de la página web del Instituto Nacional de Estadística e Informática de Perú (INEI) y consta de 216 observaciones.
Originalmente, esta era la estructura de los datos:

| Mes | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ene | 6 385 | 7 401 | 8 558 | 8 426 | 11 354 | 12 798 | 11 234 | 13 560 | 20 890 | 23 121 | 23 840 | 33 254 | 38 948 | 36 948 | 31 783 | 44 092 | 33 490 | 21 985 |
| Feb | 3 867 | 5 706 | 6 017 | 7 512 | 7 201 | 10 001 | - | 9 163 | 12 877 | 15 356 | 14 749 | 24 443 | 29 596 | 25 479 | 28 071 | 24 673 | 29 479 | - |
| Mar | 3 755 | 5 782 | 4 779 | 6 098 | 6 887 | 7 495 | - | 7 083 | 11 093 | 16 753 | 14 562 | 17 028 | 25 995 | 21 792 | 19 014 | 17 092 | 18 234 | 9 065 |
| Abr | 3 719 | 3 608 | 5 465 | 7 497 | 6 195 | 9 365 | 5 243 | 9 838 | 16 930 | 13 446 | 17 277 | 18 324 | 20 040 | 18 699 | 16 761 | 18 538 | - | 8 505 |
| May | 4 743 | 5 310 | 5 852 | 8 515 | 9 013 | 10 511 | 13 375 | 12 448 | 14 755 | 18 380 | 19 056 | 22 298 | 30 393 | 18 801 | 21 592 | 23 229 | - | 18 263 |
| Jun | 5 121 | 4 830 | 5 741 | 7 889 | 7 784 | 8 450 | 10 036 | 12 771 | 16 846 | 19 564 | 20 621 | 22 826 | 26 990 | 20 979 | 20 658 | 24 211 | - | 19 709 |
| Jul | 8 967 | 10 668 | 8 289 | 8 364 | 14 415 | 14 690 | 15 905 | 18 500 | 25 994 | 24 492 | 31 254 | 49 350 | 45 464 | 34 146 | 33 932 | 36 365 | - | 38 785 |
| Ago | 12 203 | 13 599 | 15 172 | 18 611 | 18 707 | 14 644 | 20 788 | 33 905 | 33 354 | 39 647 | 41 468 | 36 577 | 48 229 | 43 904 | 54 551 | 50 104 | - | 46 800 |
| Set | 6 394 | 9 773 | 10 985 | 14 294 | 15 478 | 13 250 | 16 007 | 21 030 | 26 190 | 26 819 | 26 291 | 36 232 | 36 724 | 28 950 | 31 715 | 32 961 | - | 37 829 |
| Oct | 23 102 | 23 204 | 23 432 | 31 567 | 25 940 | 21 117 | 31 064 | 37 456 | 38 775 | 40 794 | 35 264 | 51 502 | 51 626 | 38 905 | 37 365 | 38 630 | - | 45 619 |
| Nov | 18 933 | 23 026 | 19 999 | 32 452 | 30 096 | 25 264 | 29 295 | 31 118 | 34 411 | 37 967 | 32 291 | 32 259 | 44 752 | 30 083 | 31 246 | 35 701 | 9 499 | 33 344 |
| Dic | 17 863 | 15 688 | 18 549 | 22 772 | 10 190 | 13 237 | 11 191 | 16 374 | 19 184 | 18 597 | 21 420 | 7 151 | 23 773 | 23 130 | 21 373 | 22 612 | 13 409 | 24 938 |

Por lo que se procedió primero, a convertir los meses en números y los años que estaban en formato “chr” (2004, 2010, 2020 y 2021) se cambiaron a “numeric”.

```
meses <- c("Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Se
t", "Oct", "Nov", "Dic")
datos <- datos %>%
  mutate(Mes = sprintf("%02d", match(Mes, meses)))
datos$Mes <- as.numeric(datos$Mes)
datos$`2004` <- replace_na(as.numeric(datos$`2004`), 0)
datos$`2010` <- replace_na(as.numeric(datos$`2010`), 0)

## Warning in replace_na(as.numeric(datos$`2010`), 0): NAs introducido
s por
## coerción

datos$`2020` <- replace_na(as.numeric(datos$`2020`), 0)

## Warning in replace_na(as.numeric(datos$`2020`), 0): NAs introducido
s por
## coerción

datos$`2021` <- replace_na(as.numeric(datos$`2021`), 0)

## Warning in replace_na(as.numeric(datos$`2021`), 0): NAs introducido
s por
## coerción
```

Se observa que aparecen mensajes de warning debido a que en el conjunto de datos tenemos el símbolo” –“que denota que no hubo visitantes en ese mes.

| 2020 | 2021 |
|---|---|
| 33 490 | 21 985 |
| 29 479 | - |
| 18 234 | 9 065 |
| - | 8 505 |
| - | 18 263 |

Luego, usando la librería tidyverse, se crean las columnas “Fecha” en dónde colocaremos el año y mes y “Visitantes” la cantidad de visitantes para dicho año y mes. Eliminamos la columna “Mes”, formateamos la columna “Fecha” y ordenamos de manera ascendente.
Así tenemos:

```
datos_modificados <- datos %>%
  pivot_longer(cols = starts_with("20"), names_to = "fecha", values_to
= "visitante") %>%
  mutate(fecha = paste(fecha, Mes, sep = "-")) %>%
  select(-Mes) %>%
  mutate(fecha = as.Date(paste0(fecha, "-01"), format = "%Y-%m-%d")) %
>%
  mutate(fecha = format(fecha, "%Y-%m")) %>%
  arrange(fecha)
datos_modificados
```

Finalmente, nuestros datos cuentan con la siguiente estructura:

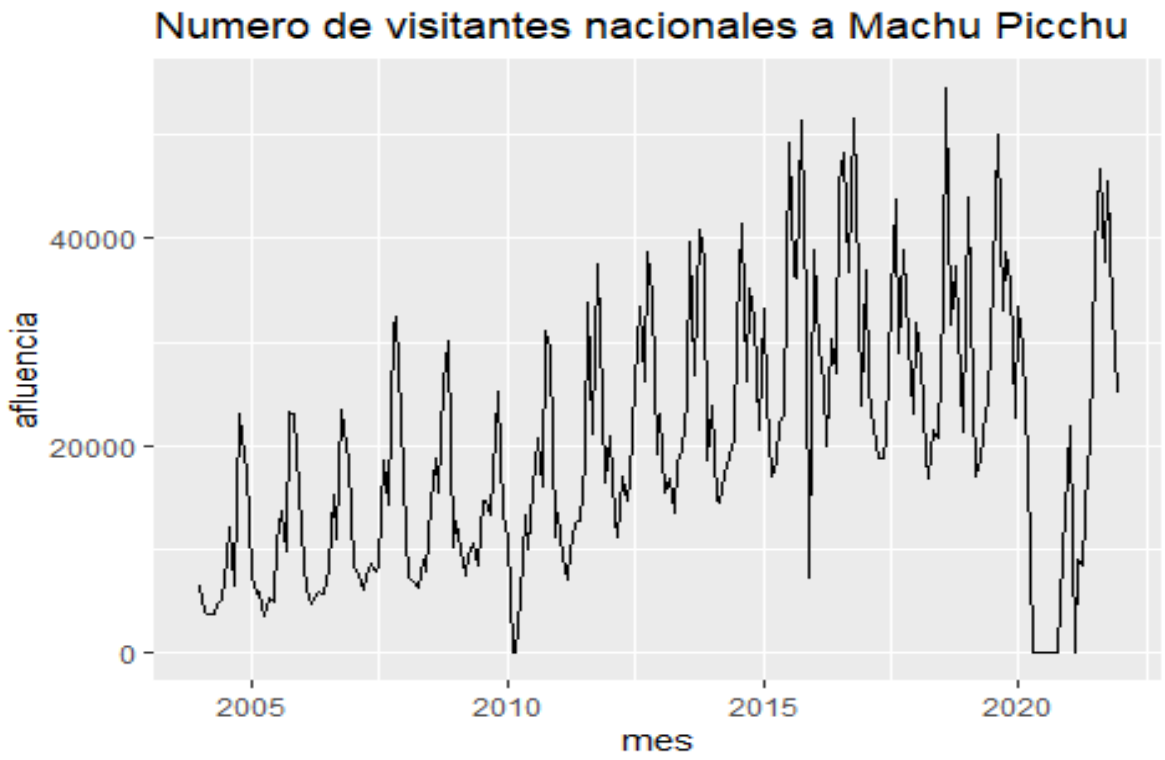
```
knitr::kable(head(datos_modificados, 10))
```

| fecha | visitante |
|---|---|
| 2004-01 | 6385 |
| 2004-02 | 3867 |
| 2004-03 | 3755 |
| 2004-04 | 3719 |
| 2004-05 | 4743 |
| 2004-06 | 5121 |
| 2004-07 | 8967 |
| 2004-08 | 12203 |
| 2004-09 | 6394 |
| 2004-10 | 23102 |

## 2. Representación gráfica y descomposición estacional (si tuviera comportamiento estacional)

Creamos un objeto time series “afluencia”, reemplazamos los NAs generados para los años 2010, 2020 y 2021 por 0 y lo graficamos:

```
afluencia <- ts(datos_modificados[,-1], start=c(2004,1), frequency=12)
#Reemplazando NAs por 0
afluencia[is.na(afluencia)] <- 0
autoplot(afluencia)+ ggtitle("Numero de visitantes nacionales a Machu
Picchu ") + xlab("mes") + ylab("afluencia")
```




Se observa que la serie presenta una tendencia ascendente no estacionaria (no presenta media constante) y es de comportamiento estacional: hay una disminución de visitantes para los meses de lluvias (enero - marzo) en Cuzco y un incremento para los meses de temporada seca (abril – octubre).

**Descomposición estacional**

Cálculo de coeficientes de estacionalidad

```
afluencia_comp <- decompose(afluencia, type=c("additive"))

knitr::kable(afluencia_comp$figure, digits =2,caption = "Coef Estacionalidad")
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2359.13 | -5525.75 | -8132.94 | -9031.12 | -5823.98 | -5987.58 | 2586.59 | 9303.01 | 924.77 | 12467.67 | 9402.16 | -2541.97 |

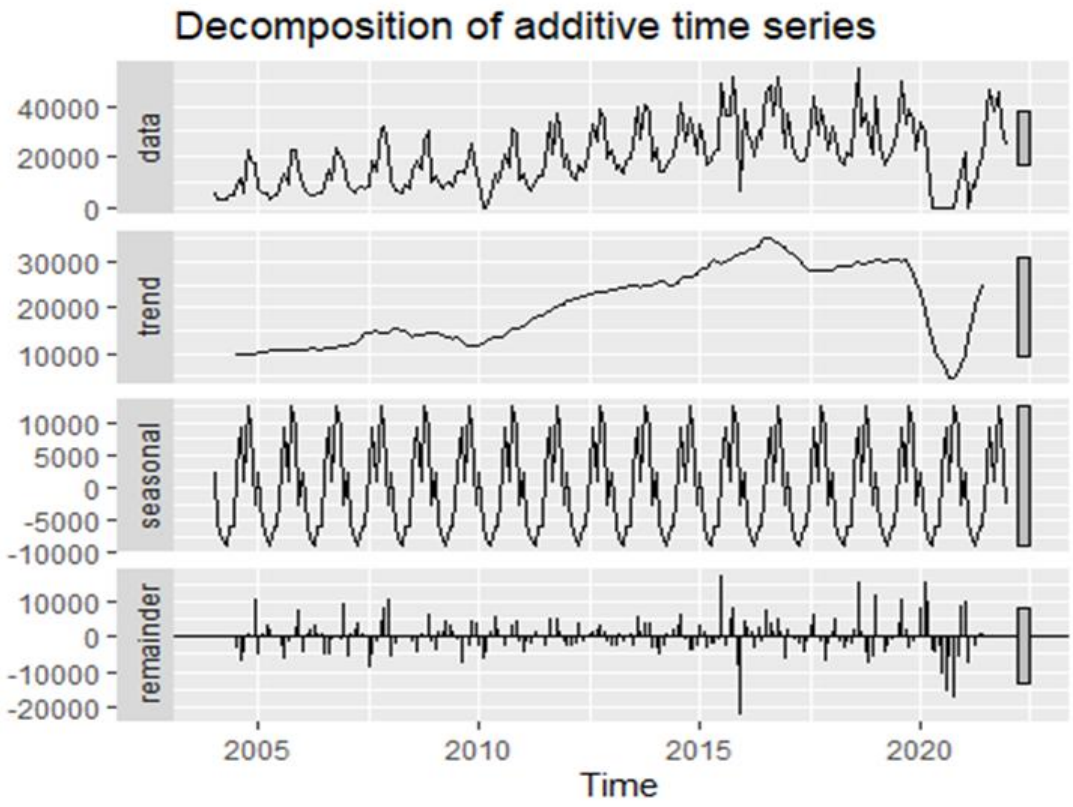
Se observa que el mes que tiene un coeficiente mayor es octubre lo que quiere decir que en este mes es en dónde hay una cantidad mayor de turistas nacionales que visitan el sitio arqueológico de Machu Picchu. Esto puede deberse al buen clima y que en este mes se da el fin de la temporada seca por lo cual los turistas tendrían que esperar el fin de la temporada de lluvia (noviembre a marzo) para poder visitar Machu Picchu.

El mes que presenta un menor coeficiente es abril lo que indica que en este mes hay la menor afluencia de turistas nacionales a Machu Picchu. Si bien en este mes se celebra Semana Santa y se da el feriado largo en el país, los precios de

los alojamientos y los pasajes aéreos nacionales se elevan por lo cual los turistas nacionales se decantan por viajar a otras provincias.

Gráfico de la serie, la componente estacional, la estimación de la tendencia y el error.

```
autoplot(afluencia_comp)
```




Se observa que la tendencia es al alza a partir del año 2007 esto podría deberse a que en junio de ese año Machu Picchu fue elegida como una de las “Siete Maravillas del Mundo Moderno” y por tanto el interés de los turistas nacionales aumentó. También, se visualiza que hubo un descenso en la afluencia en el año de la pandemia 2020.

Sobre la estacionalidad, se observa un patrón recurrente lo cual nos indica que existen meses del año en el que número de turistas nacionales aumenta o disminuye.

Gráfico de la serie original, la tendencia calculada con la descomposición y la serie ajustada estacionalmente.

```
autoplot(afluencia, series="Datos") +
  autolayer(trendcycle(afluencia_comp), series="Tendencia")+
  autolayer(seasadj(afluencia_comp), series="Estacionalmente
ajustada")+
  xlab("Year") + ylab("Afluencia") +
  ggtitle("Serie de afluencia") +
  scale_colour_manual(values=c("gray","blue","red"),
                      breaks=c("Datos","Estacionalmente
ajustada","Tendencia"))
```

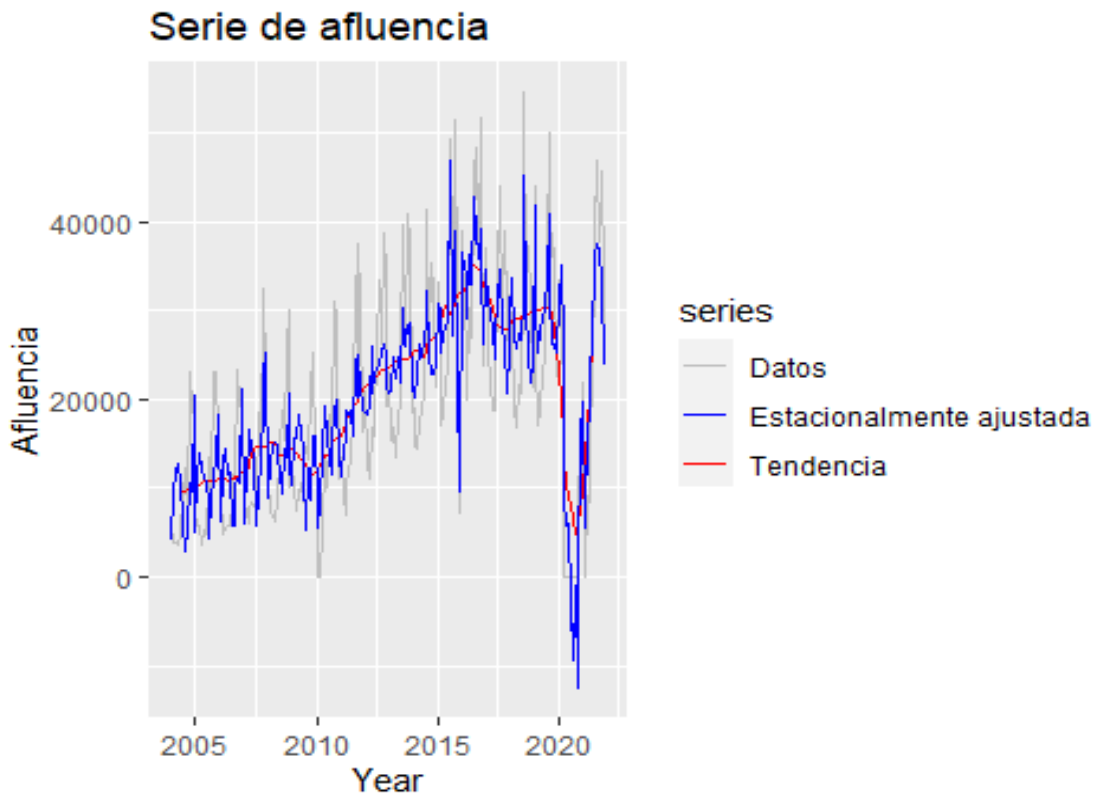
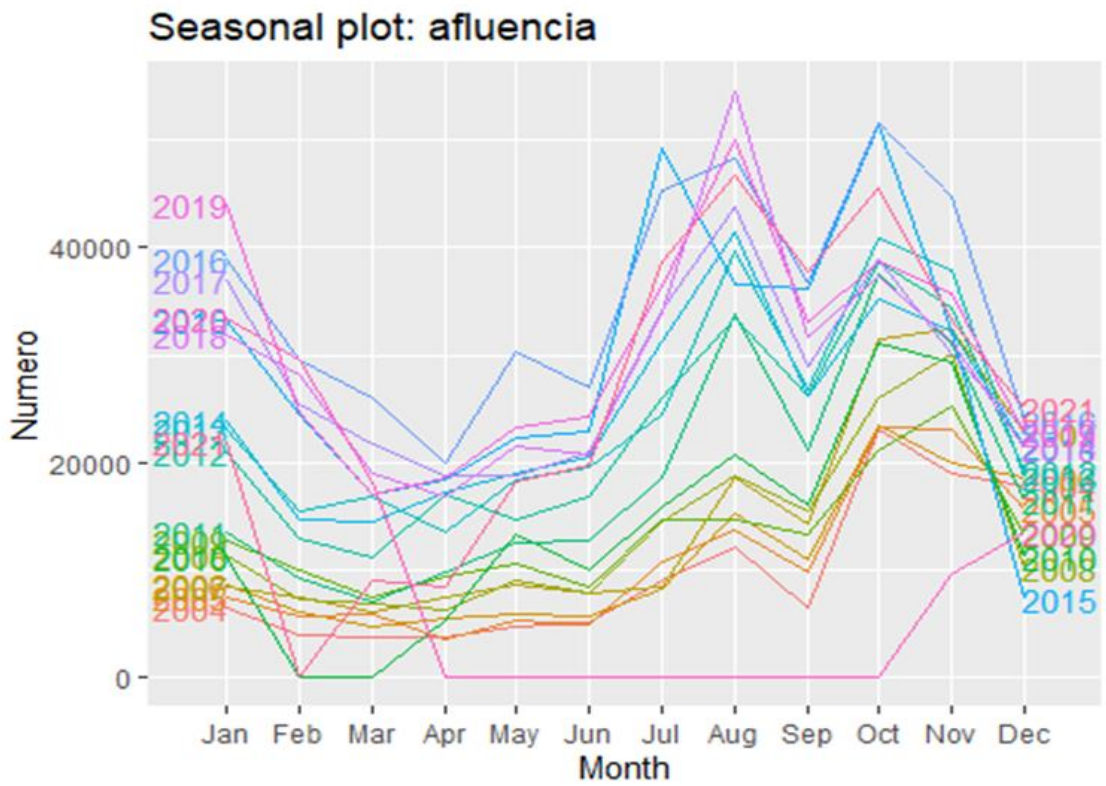


Gráfico series de cada año

```
ggseasonplot(afluencia, year.labels=TRUE, year.labels.left=TRUE) +
  ylab("Numero") +
  ggtitle("Seasonal plot: afluencia")
```




Se observan picos grandes de afluencia para los meses de julio, agosto y octubre. También, menor afluencia para los meses de febrero, marzo y abril.

**3. Para comprobar la eficacia de los métodos de predicción que vamos a hacer en los siguientes apartados reservamos los últimos datos observados (un periodo en las series estacionales o aproximadamente 10 observaciones) para comparar con las predicciones realizadas por cada uno de los métodos. Luego ajustamos los modelos sobre la serie sin esos últimos datos en los siguientes apartados.**

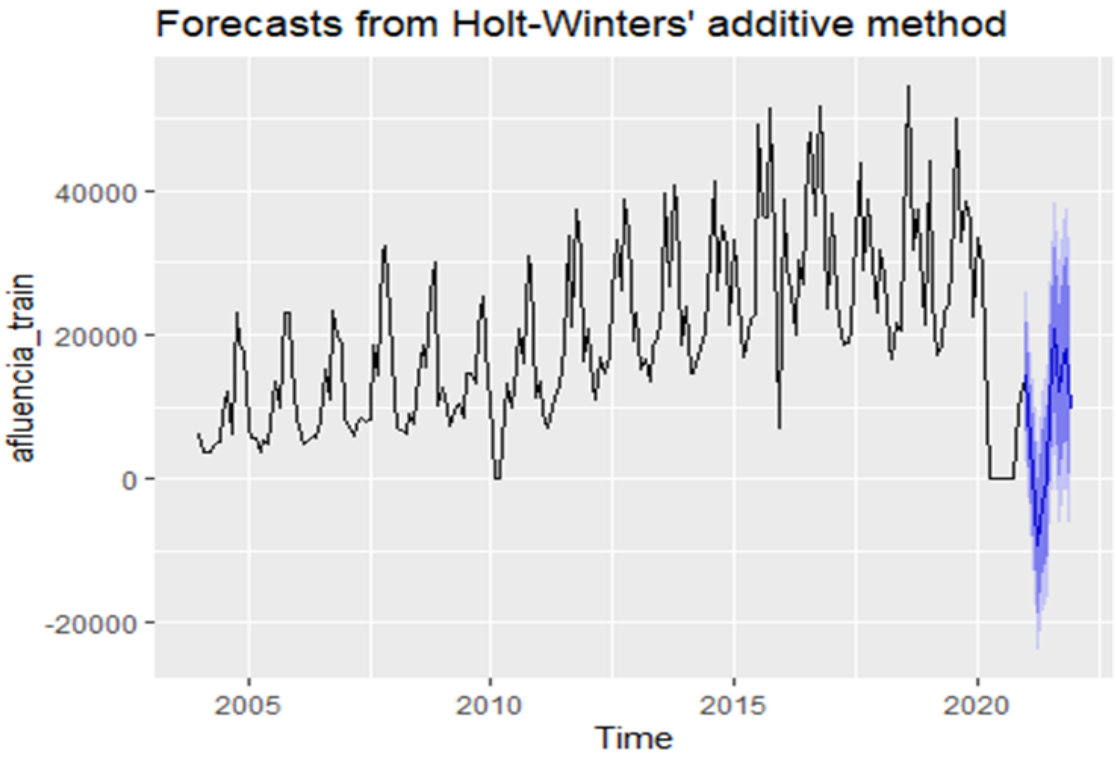
Reservamos el último año para comparar las predicciones:

```
afluencia_train <- window(afluencia, end=c(2020,12))
```

**4. Encontrar el modelo de suavizado exponencial más adecuado, mostrando una tabla con los estimadores de los parámetros del modelo elegido. Para dicho modelo, representar gráficamente la serie observada y la suavizada con las predicciones para un periodo que se considere adecuado. Mostrar una tabla con las predicciones.**

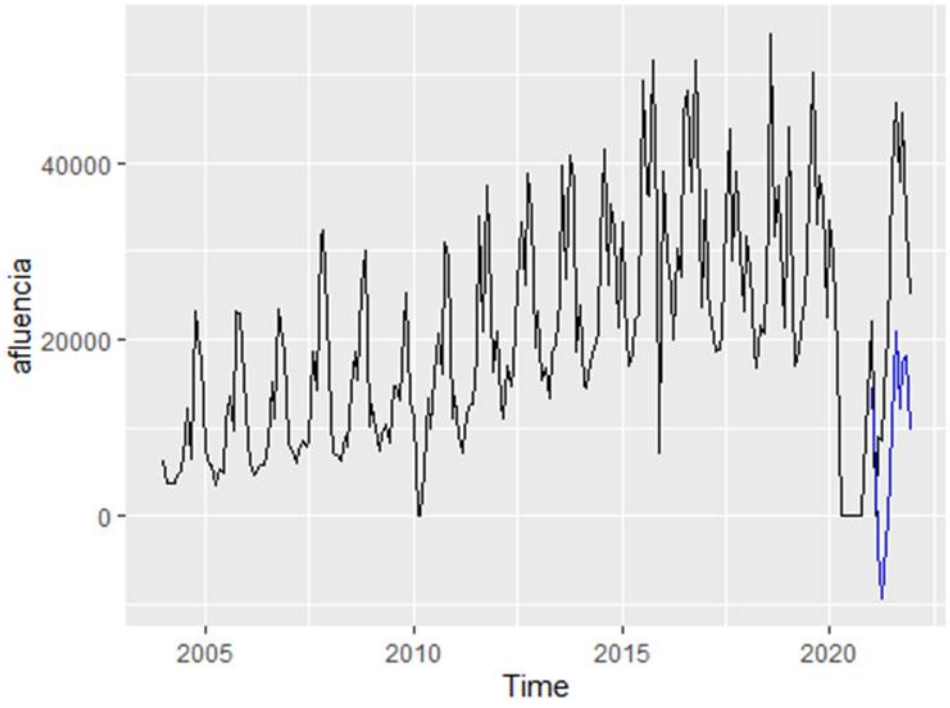
En este caso usaremos el **modelo de suavizado Holt-Winters** ya que nuestra serie presenta comportamiento estacional.

```
afluencia_sh <- hw(afluencia_train,seasonal="additive", h=12, level =
c(80, 95))
autoplot(afluencia_sh)
```




**Autoplot predicción del modelo para el año 2021 sobre la serie original**

```
autoplot(afluencia) + autolayer(afluencia_sh, PI = FALSE)
```



Se observa que el modelo elegido no predice de manera precisa la afluencia de turistas para cada mes del año 2021.En esta serie existen 2 hechos anómalos en ciertos períodos: febrero y marzo 2010 tuvieron 0 turistas nacionales debido a que las lluvias torrenciales dañaron la vía férrea que llega hasta Aguas Calientes, el poblado que da acceso a Machu Picchu, por lo que se tuvieron que realizar reparaciones y prohibir la llegada de turistas.
De abril a octubre de 2020 Machu Picchu cierra sus puertas al turismo al declararse a nivel nacional el confinamiento ante la pandemia de COVID-19.

**Estimadores de los parámetros del modelo elegido**

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

| | x |
|---|---|
| alpha | 0.4206 |
| beta | 0.0001 |
| gamma | 0.3364 |
| l | 9545.1412 |
| b | 151.7326 |
| s0 | -3311.5851 |
| s1 | 9524.0717 |
| s2 | 13305.0484 |
| s3 | 1042.5266 |
| s4 | 10045.0554 |
| s5 | 3015.5178 |
| s6 | -6280.8342 |
| s7 | -6108.7231 |
| s8 | -9075.6200 |
| s9 | -8359.4245 |
| s10 | -5269.2392 |

**Tabla de predicciones**

```
knitr::kable(forecast(afluencia_sh,h=12), digits =4,caption = "Predicc
iones ")
```

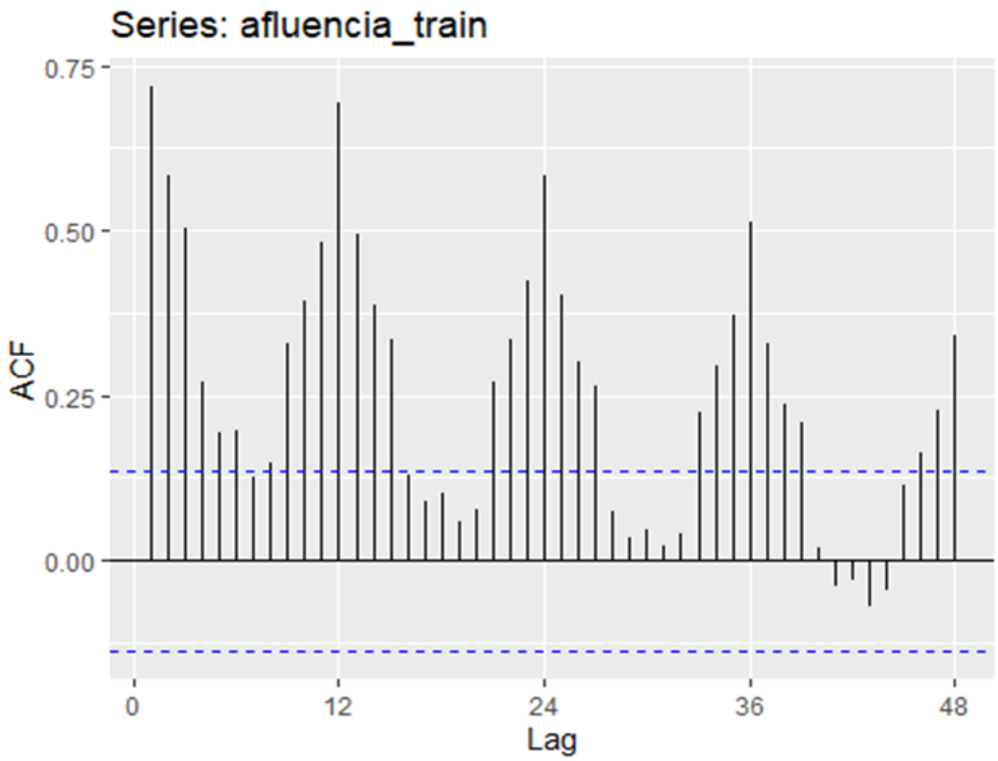
| | Point Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
|---|---|---|---|---|---|
| Jan 2021 | 14409.8702 | 6741.8389 | 22077.9015 | 2682.628 | 26137.112 |
| Feb 2021 | 4984.1626 | -3334.6941 | 13303.0194 | -7738.431 | 17706.756 |
| Mar 2021 | -3571.3268 | -12493.9385 | 5351.2848 | -17217.284 | 10074.630 |

| | Point Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
|---|---|---|---|---|---|
| Apr 2021 | -9288.1116 | -18776.3974 | 200.1742 | -23799.193 | 5222.969 |
| May 2021 | -3182.7662 | -13205.0956 | 6839.5632 | -18510.596 | 12145.064 |
| Jun 2021 | -285.5158 | -10815.0721 | 10244.0406 | -16389.083 | 15818.051 |
| Jul 2021 | 10457.3805 | -556.2919 | 21471.0528 | -6386.578 | 27301.339 |
| Aug 2021 | 20956.4808 | 9478.8785 | 32434.0831 | 3403.003 | 38509.959 |
| Sep 2021 | 12256.8658 | 333.1631 | 24180.5684 | -5978.864 | 30492.595 |
| Oct 2021 | 17233.0604 | 4879.1553 | 29586.9654 | -1660.607 | 36126.728 |
| Nov 2021 | 18172.9378 | 5403.1213 | 30942.7542 | -1356.811 | 37702.687 |
| Dec 2021 | 9611.6409 | -3561.1498 | 22784.4316 | -10534.404 | 29757.686 |

**5. Representar la serie y los correlogramas. Decidir qué modelo puede ser ajustado. Ajustar el modelo adecuado comprobando que sus residuales están incorrelados. (Sintaxis, tablas de los parámetros estimados y gráficos)**

**Autocorrelaciones simples hasta el retardo 48**

```
ggAcf(afluencia_train, lag= 48)
```




Se observa que decrece de forma lenta, que los retardos 12, 24, 36 y 48 presentan una correlación más fuerte y también un patrón que se repite cada 12 meses. Además, se visualizan valores negativos en los retardos 41, 42, 43 y 44 lo cual podría deberse a que no hubo turistas en el período de COVID-19.

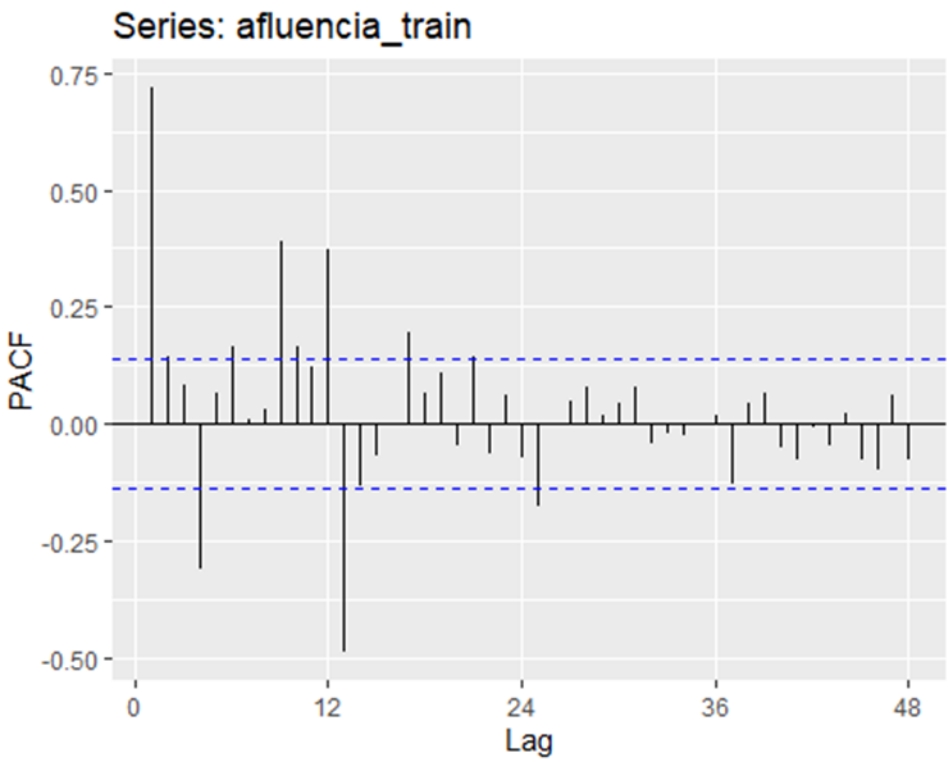
```
corr<-Acf(afluencia_train, lag=12)
```

```
print(corr)

##
## Autocorrelations of series 'afluencia_train', by lag
##
##     0     1     2     3     4     5     6     7     8     9    10
11    12
## 1.000 0.718 0.585 0.502 0.270 0.193 0.196 0.128 0.147 0.330 0.394 0
.484 0.692
```

La tabla muestra que el decrecimiento en los valores no es exponencial por lo cual se procederá con una diferenciación de orden 1.

**Autocorrelaciones parciales hasta el retardo 48**

```
ggPacf(afluencia_train, lag=48)
```

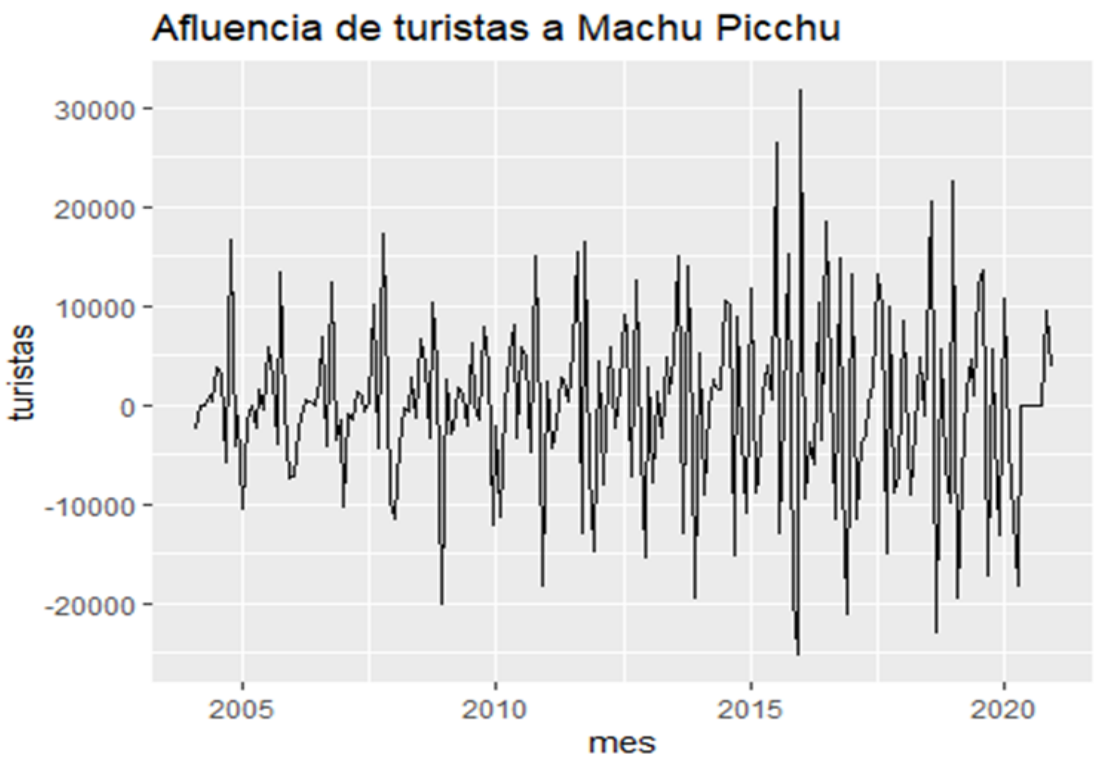



```
corrp<-Pacf(afluencia_train, lag=12)
```

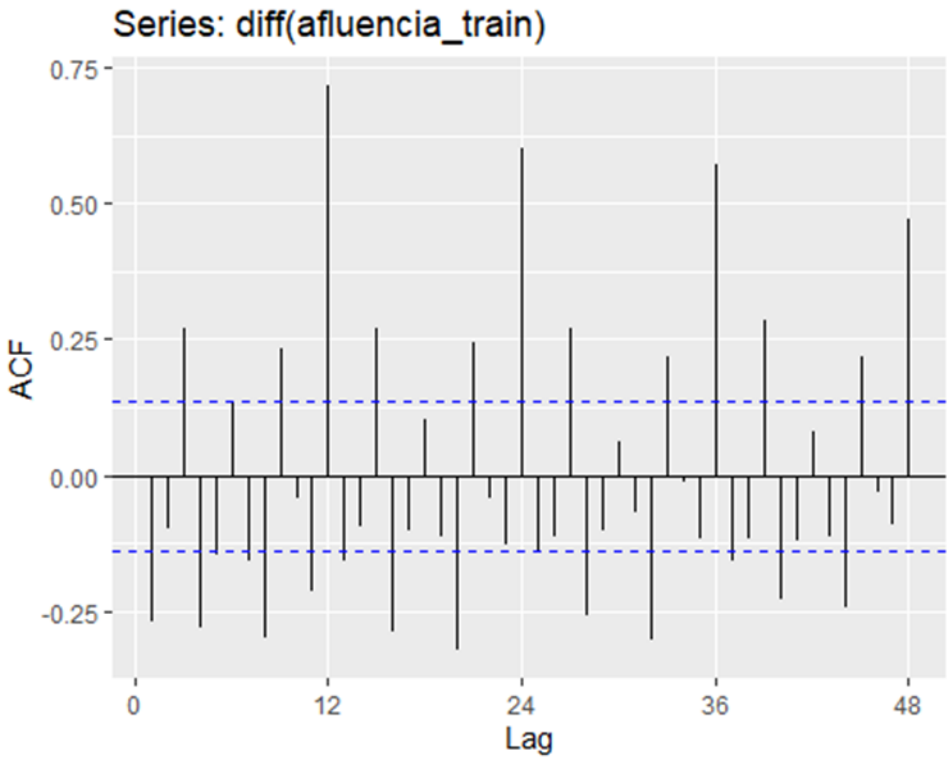
```
print(corrp)

##
## Partial autocorrelations of series 'afluencia_train', by lag
##
##      1      2      3      4      5      6      7      8      9     10
11
##  0.718  0.142  0.085 -0.312  0.065  0.166  0.008  0.033  0.391  0.165
0.121
##     12
##  0.375
```

**Serie diferenciada**

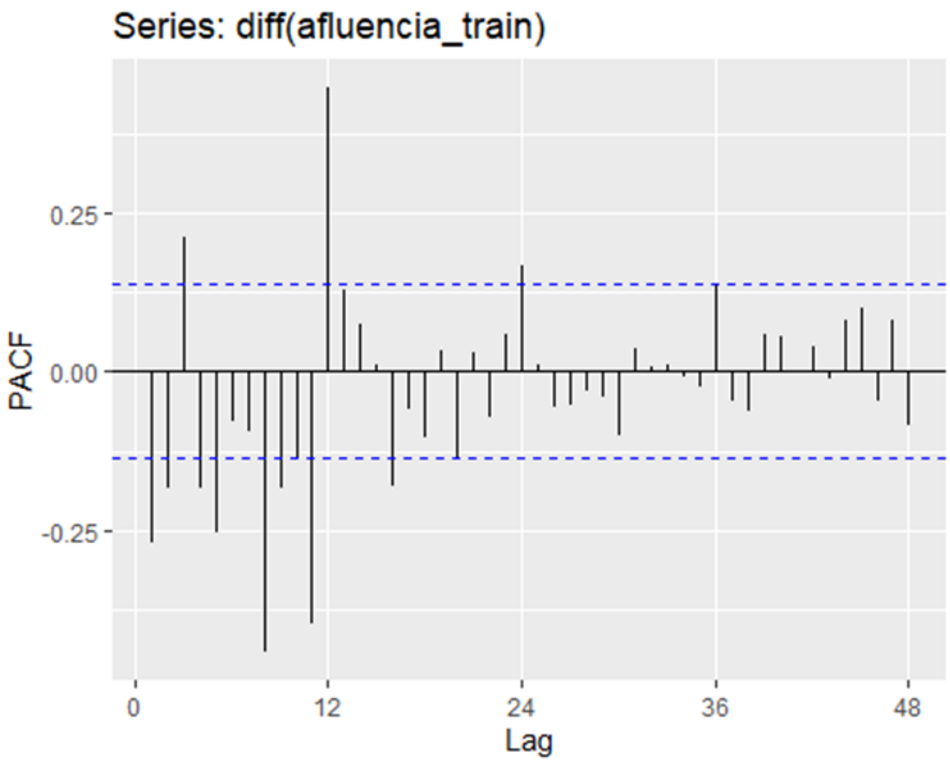
```
autoplot(diff(afluencia_train))+ ggtitle("Afluencia de turistas a Mach
u Picchu") +
  xlab("mes") + ylab("turistas")
```



```
ggAcf(diff(afluencia_train), lag=48)
```



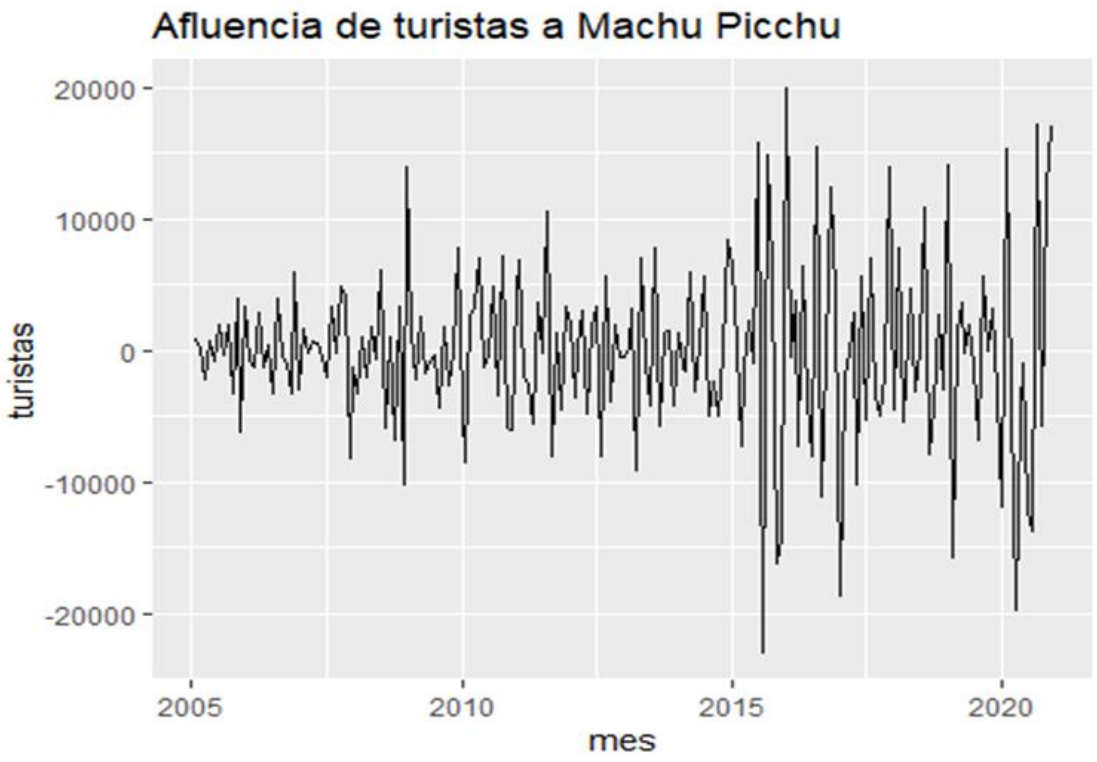

```
ggPacf(diff(afluencia_train), lag=48)
```



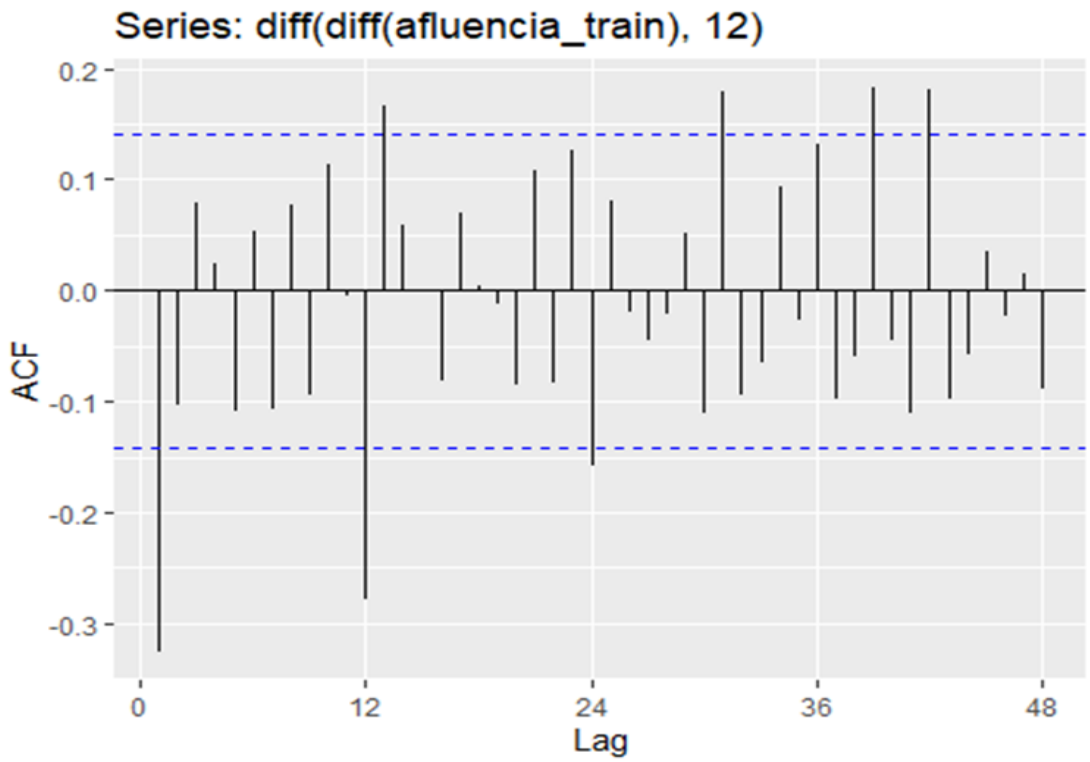
Se observa que las correlaciones han disminuido pero los retardos 12, 24, 36 y 48 siguen presentando una correlación fuerte. Así procederemos a realizar una diferenciación de orden 12 (estacional).

**Diferenciación estacional**

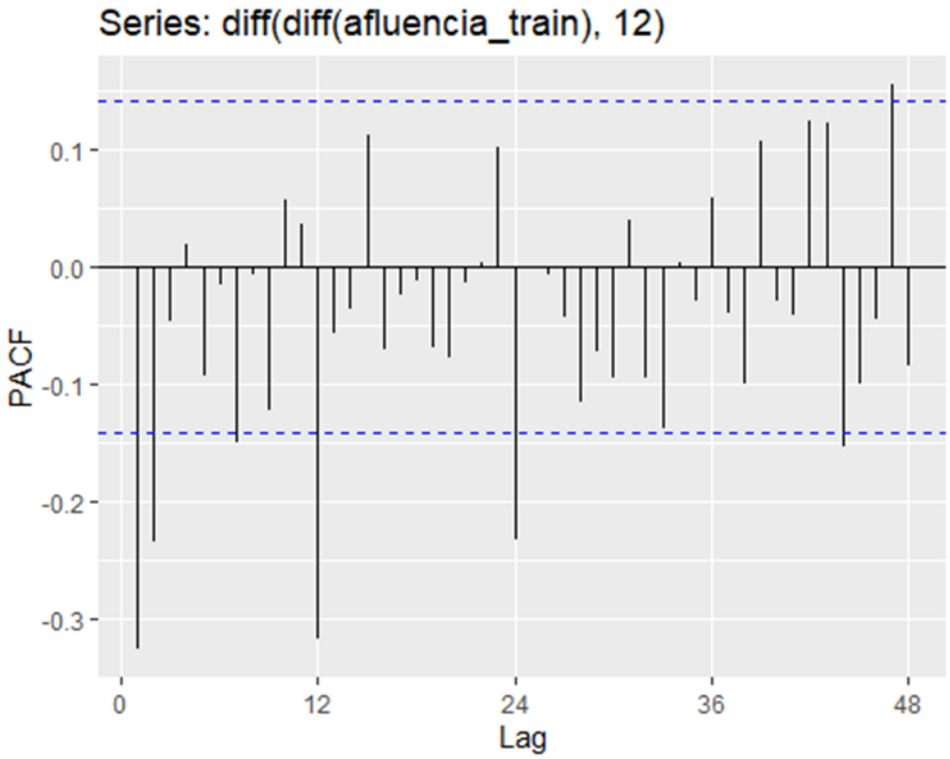
```
autoplot(diff(diff(afluencia_train),12))+ ggtitle("Afluencia de turistas a Machu Picchu") +
  xlab("mes") + ylab("turistas")
```



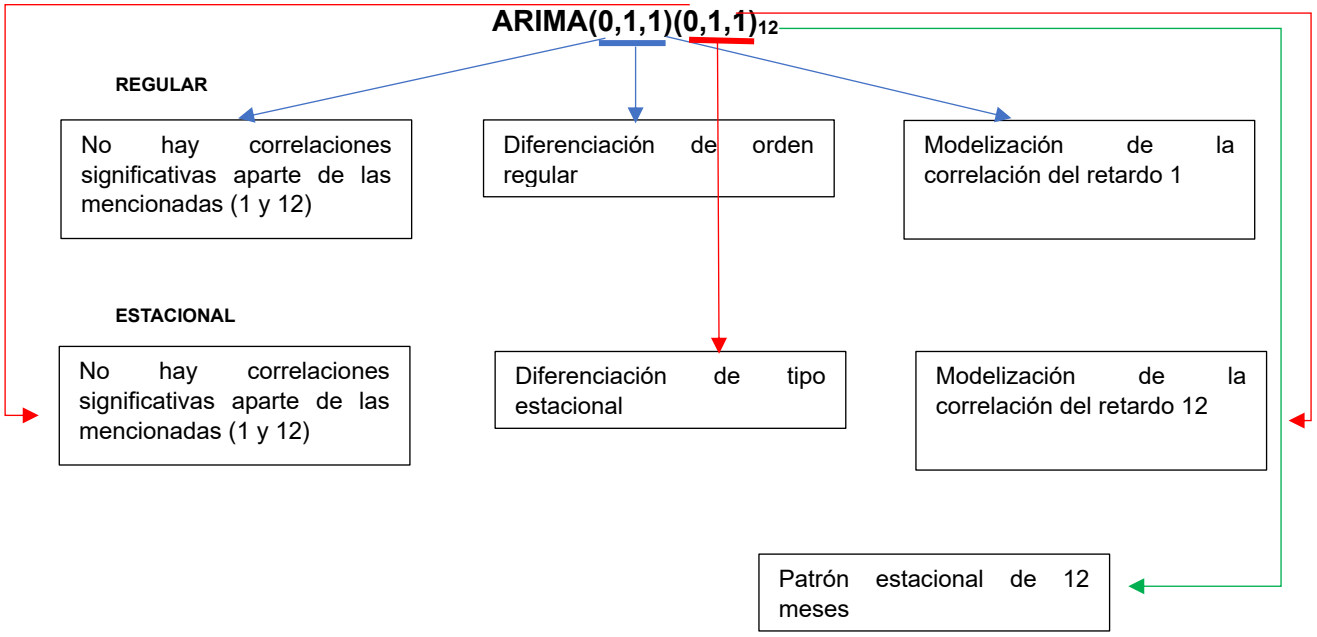
```
ggAcf(diff(diff(afluencia_train),12), lag=48)
```



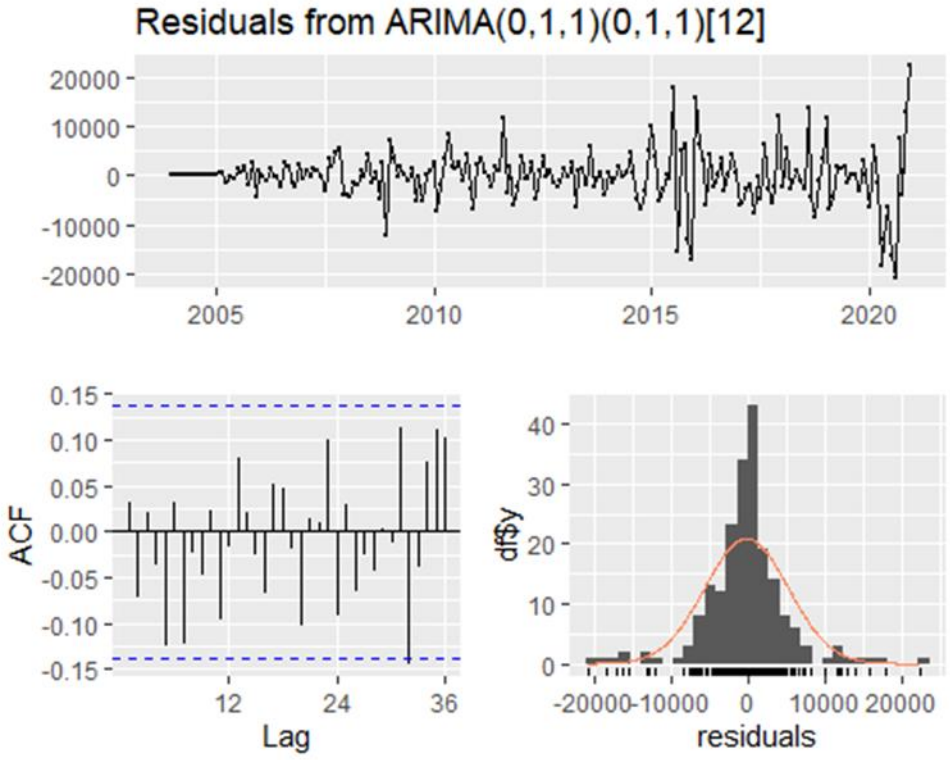

```
ggPacf(diff(diff(afluencia_train),12), lag=48)
```



Se observa que en los correlogramas diferenciados sobresalen significativamente de las barras de confianza los retardos del orden 1 y 12 por lo cual el modelo ARIMA a ajustar sería:



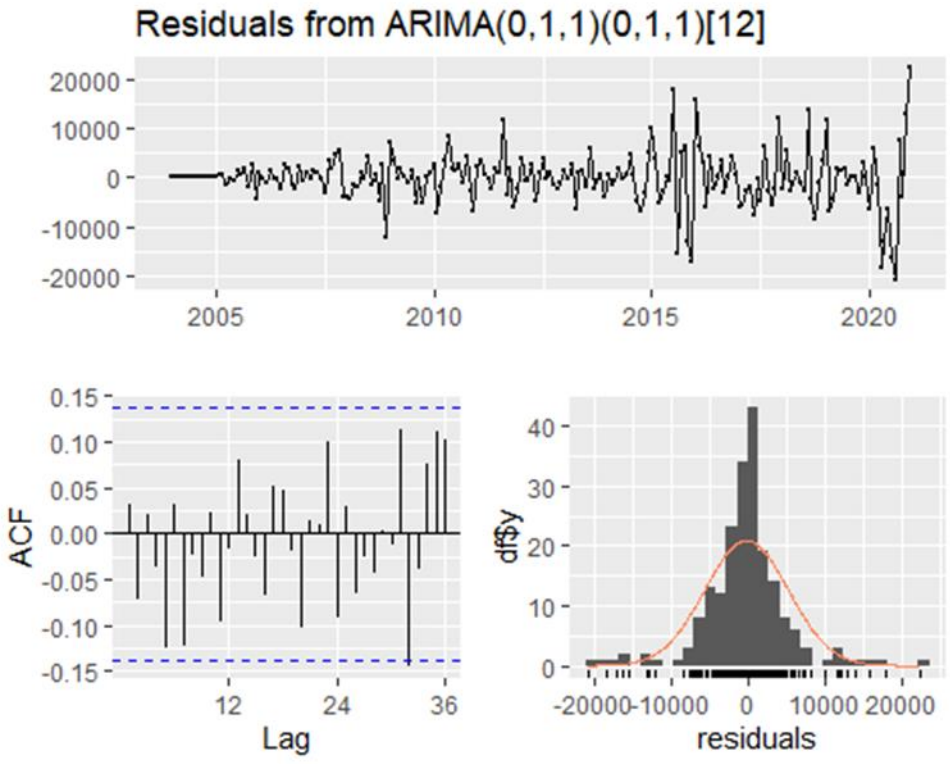

```
fitafluencia1 <- Arima((afluencia_train),c(0,1,1),seasonal=c(0,1,1))
checkresiduals(fitafluencia1)
```



```
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 21.94, df = 22, p-value = 0.4635
##
## Model df: 2.   Total lags used: 24

print(fitafluencia1)

## Series: (afluencia_train)
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.4658  -0.5087
## s.e.   0.0779   0.0757
##
## sigma^2 = 31861782:  log likelihood = -1921.88
## AIC=3849.75   AICc=3849.88   BIC=3859.51
```

**Auto ARIMA**

```
fitafluencia2 <- auto.arima(afluencia_train)
checkresiduals(fitafluencia2)
```



```
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 21.94, df = 22, p-value = 0.4635
##
## Model df: 2.   Total lags used: 24
```

```
print(fitafluencia2)
```

```
## Series: afluencia_train
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.4658  -0.5087
## s.e.   0.0779   0.0757
##
## sigma^2 = 31861782:  log likelihood = -1921.88
## AIC=3849.75   AICc=3849.88   BIC=3859.51
```

La función auto.arima seleccionó como mejor modelo el mismo que señalamos en el ajuste manual:

**ARIMA(0,1,1)(0,1,1)$_{12}$**

## 6. Escribir la expresión algebraica del modelo ajustado con los parámetros estimados.

$(1 - B)(1 - B^{12})(1 - B)(X_t) = (1 + 0.4658B)(1 + 0.5087B)(Z_t)$

$(X_t - X_{t-1})(1 - B^{12}) = (Z_t + 0.4658Z_{t-1})(1 + 0.5087B)$

$X_t - X_{t-1} - X_{t-13} + X_{t-14} = Zt + 0.4658Z_{t-1} + 0.5087Z_{t-1} + 0.4658Z_{t-13} + 0.5087Z_{t-13}$

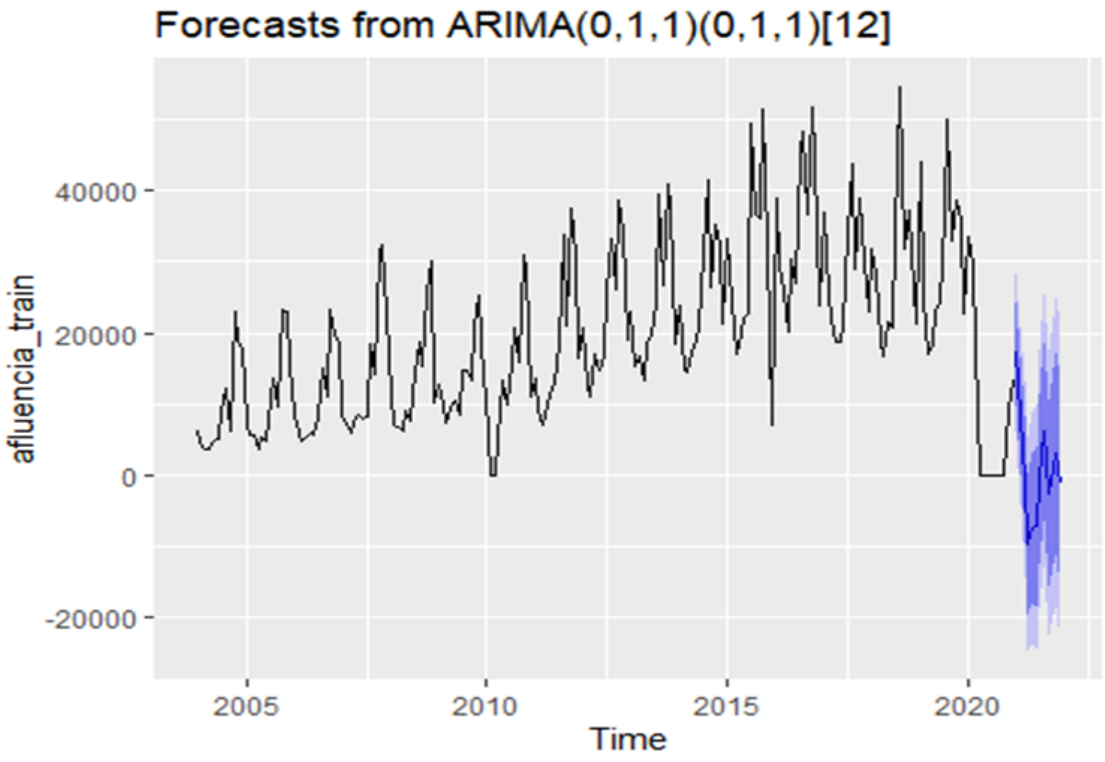
$Xt = X_{t-1} + X_{t-13} - X_{t-14} + Z_t + 0.4658Z_{t-1} + 0.5087Z_{t-1} + 0.4658Z_{t-13} + 0.5087Z_{t-13}$

**7. Calcular las predicciones y los intervalos de confianza para las unidades de tiempo que se considere oportuno, dependiendo de la serie, siguientes al último valor observado. Representarlas gráficamente.**

```
knitr::kable(forecast(fitafluencia1,h=12))
```

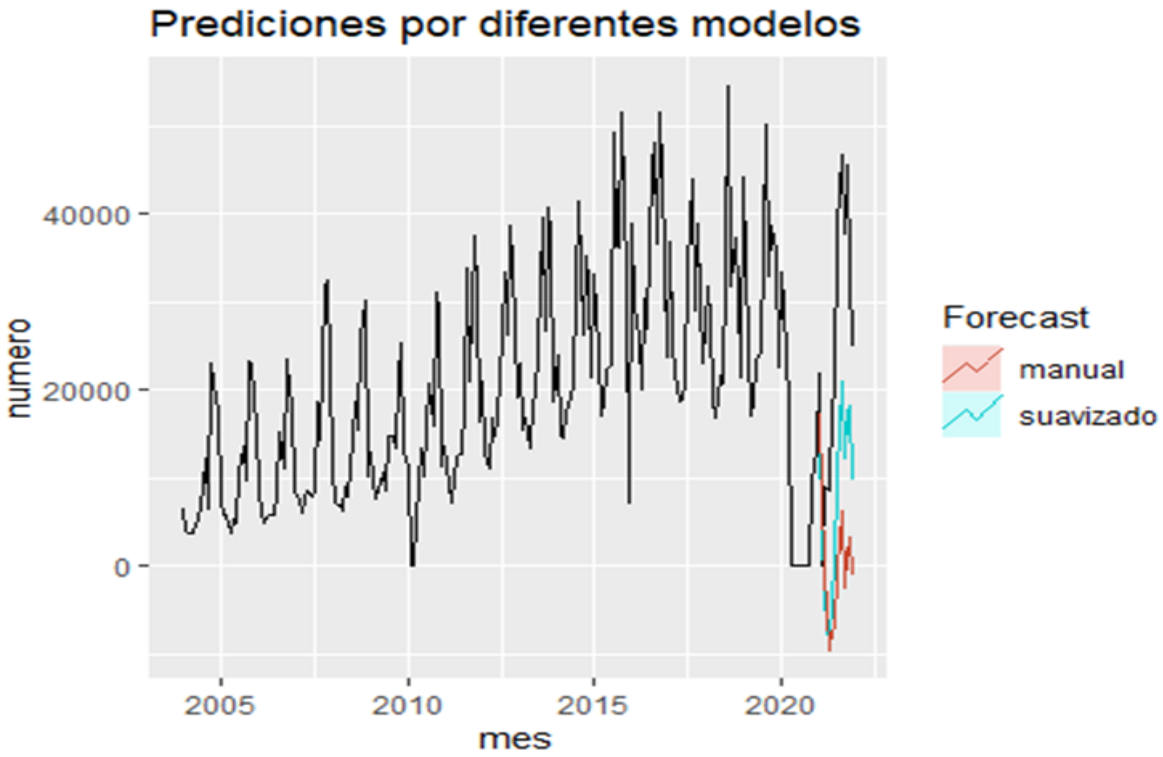
| | Point Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
|---|---|---|---|---|---|
| Jan 2021 | 17260.7357 | 10026.8589 | 24494.6126 | 6197.476 | 28323.996 |
| Feb 2021 | 8626.3075 | 425.1231 | 16827.4919 | -3916.322 | 21168.937 |
| Mar 2021 | -428.2103 | -9494.0734 | 8637.6529 | -14293.252 | 13436.831 |
| Apr 2021 | -9657.4847 | -19512.4492 | 197.4798 | -24729.352 | 5414.383 |
| May 2021 | -7416.9584 | -18002.3622 | 3168.4455 | -23605.937 | 8772.020 |
| Jun 2021 | -7231.8595 | -18500.4541 | 4036.7352 | -24465.688 | 10001.969 |
| Jul 2021 | -458.3601 | -12371.0288 | 11454.3085 | -18677.215 | 17760.494 |
| Aug 2021 | 6295.7829 | -6227.8797 | 18819.4455 | -12857.506 | 25449.072 |
| Sep 2021 | -2462.7640 | -15568.9677 | 10643.4397 | -22506.973 | 17581.445 |
| Oct 2021 | 1259.9600 | -12403.9716 | 14923.8915 | -19637.220 | 22157.140 |
| Nov 2021 | 3244.3184 | -10955.4518 | 17444.0887 | -18472.356 | 24960.993 |
| Dec 2021 | -1165.6781 | -15881.7892 | 13550.4330 | -23672.028 | 21340.672 |

```
autoplot(forecast(fitafluencia1,h=12))
```



**8. Comparar las predicciones obtenidas con cada uno de los métodos (suavizado y ARIMA) con los valores observados que habíamos reservado antes. Conclusiones.**

```
autoplot(afluencia) +
  autolayer(forecast(afluencia_sh,h=12), series="suavizado", PI=FALSE)
+
  autolayer(forecast(fitafluencia1,h=12), series="manual", PI=FALSE) +
  ggtitle("Prediciones por diferentes modelos ") + xlab("mes") +
  ylab("numero") +
  guides(colour=guide_legend(title="Forecast"))
```




En la gráfica se observa que el modelo Holt-Winters (suavizado) presenta predicciones más cercanas a los datos reales para los meses del 2021 a comparación del modelo ARIMA.

En el cuadro resumen podemos constatar lo que se visualiza en la gráfica usando los datos del intervalo de confianza al 95% obtenidos con la función forecast().

Asi tenemos:

| PERIODO | HOLT-WINTERS | ARIMA | AFLUENCIA REAL |
|---|---|---|---|
| Jan 2021 | 26137 | 28324 | 21985 |
| Feb 2021 | 17707 | 21169 | 0 |
| Mar 2021 | 10075 | 13437 | 9065 |
| Apr 2021 | 5223 | 5414 | 8505 |
| May 2021 | 12145 | 8772 | 18263 |
| Jun 2021 | 15818 | 10002 | 19709 |
| Jul 2021 | 27301 | 17760 | 38785 |
| Aug 2021 | 38510 | 25449 | 46800 |
| Set 2021 | 30493 | 17581 | 37829 |
| Oct 2021 | 36127 | 22157 | 45619 |
| Nov 2021 | 37703 | 24961 | 33344 |
| Dec 2021 | 29758 | 21341 | 24938 |

Sin embargo, comparando las medidas de ajuste para ambos modelos

**Modelo Holt-Winters**

```
knitr::kable(accuracy(afluencia_sh), digits = 4, caption = "Medidas de ajuste")
```

| | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
|---|---|---|---|---|---|---|---|
| Training set | -366.8506 | 5743.963 | 3869.794 | NaN | Inf | 0.7644 | 0.1534 |

**Modelo ARIMA(0,1,1)(0,1,1)$_{12}$**

```
knitr::kable(accuracy(fitafluencia1), digits = 4, caption = "Medidas de ajuste")
```

| | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
|---|---|---|---|---|---|---|---|
| Training set | -193.943 | 5433.139 | 3510.316 | NaN | Inf | 0.6934 | 0.0324 |

El modelo ARIMA presenta un mejor desempeño respecto al modelo Holt-Winters.

Ahora bien, como se mencionó antes, la serie presenta 2 hechos anómalos por lo que esto puede influir en las predicciones realizadas por el modelo ARIMA como se observa en el cuadro resumen. Por lo tanto, **el modelo que mejor se adapta a nuestros datos** sería el **modelo Holt-Winters**.