Modelos para datos temporales y espaciales

Trabajo evaluación MDTE (Temas 1 a 4)

Julio 2017

El alumno debe aplicar los modelos ARIMA a un conjunto de datos reales. El conjunto de datos puede seleccionarse de cualquier fuente disponible en la web, por ejemplo www.ine.es.

1 Introducción

1.1 Información del alumno

- **Nombre**: Inmaculada
- **Apellidos**: Perea Fernández

1.2 Carga de librerías necesarias

```r
if (!require('tseries')) install.packages('tseries'); library('tseries')
if (!require('forecast')) install.packages('forecast'); library('forecast')
```

1.3 Datos usados

1.3.1 Fuente

Los datos utilizados para este ejercicio de evaluación se han obtenido del *Instituto Nacional de Estadística* (INE).

Se pueden descargar:

1) Accediendo al siguiente enlace http://www.ine.es/jaxiT3/Tabla.htm?t=20239
2) Navegando por las siguientes secciones: *INEbase > Servicios > Transporte > Estadística de transporte de viajeros > Total de viajeros por tipo, medio de transporte (terrestre, aéreo y maritimo) y distancia*

1.3.2 Descripción

La Estadística de transporte de viajeros (TV) tiene como objetivo proporcionar información mensual sobre el número de viajeros transportados en transporte urbano (autobús y metro), interurbano (autobús, ferrocarril, avión y barco) y especial y discrecional por autobús. El transporte por autobús se investiga mediante una encuesta por muestreo.

Para el transporte por ferrocarril el número de viajeros se calcula a partir de la información suministrada por los operadores ferroviarios (RENFE y otras empresas autonómicas). La información para el transporte aéreo es suministrada por Aviación Civil y para el transporte marítimo el número de pasajeros desembarcados se elabora a partir de la información de Puertos del Estado.

- Tipo de encuesta: continua de periodicidad mensual.
- Ámbito poblacional: empresas que se dedican al transporte de viajeros con independencia de su actividad principal.

- Ámbito geográfico: todo el territorio nacional.
- Período de referencia de la información: mes
- Tamaño muestral: aproximadamente 1.500 empresas
- Tipo de muestreo: muestreo aleatorio estratificado de empresas, según CCAA, número de asalariados y tipo de transporte.
- Método de recogida: cumplimentación del cuestionario por parte del informante usando alguna de las siguientes vías: internet (sistema IRIA), correo electrónico, fax, teléfono o vía postal.

1.3.3 Transformaciones previas

Se han eliminado los datos correspondientes al año en curso (2017), ya que sólo había datos disponibles hasta mayo, y estamos trabajando con datos de años completos.

2 Análisis mediante modelos ARIMA

Determinar el modelo ARIMA(p, d, q)$\times$ (P,D,Q)s (siendo s la estacionalidad) que se ajusta mejor a los datos.

- Debe tenerse en cuenta si la varianza es constante o no (no transformar o transformación logarítmica) y si existe tendencia o no.
- Deben representarse la fas y la fap en cada paso.
- Deben presentarse al menos los valores del coeficiente de información de Akaike (AIC) y los valores de los parámetros del modelo en los pasos seguidos.
- Se debe comprobar que los residuos del modelo seleccionado siguen un ruido blanco.
- Estudiar si se puede simplificar el modelo

2.1 Adquirir los datos de la web en formato csv y realizar su lectura desde R

```
viajeros_ini <- read.csv("total_viajeros.csv", header=F, dec=".", sep=";")
viajeros_ini <- ts(viajeros_ini[,2], start=2005, freq=12)
str(viajeros_ini)
```

```
##  Time-Series [1:144] from 2005 to 2017: 412654 407750 433435 454532 463789 ...
```

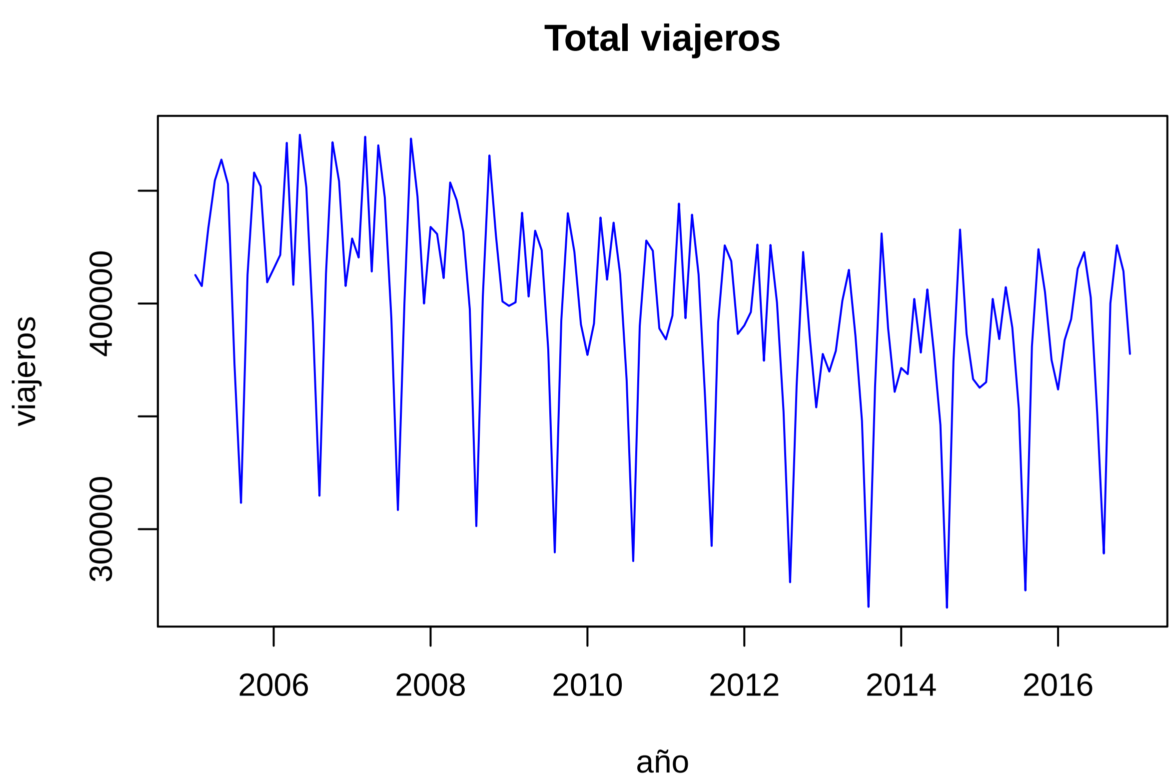
```
viajeros_ini
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2005 412654 407750 433435 454532 463789 452967 373076 311675 412730 458055
## 2006 415427 421543 471214 408305 474787 451410 391155 314838 413224 471477
## 2007 428813 420408 473913 414211 470124 446976 393562 308481 400318 473103
## 2008 433948 430825 411298 453642 445771 431864 397508 301354 403246 465615
## 2009 398959 400562 440223 403121 432272 423634 379613 289705 392418 440032
## 2010 377222 391132 438090 410607 435841 412897 365945 285863 390030 427903
## 2011 384145 394639 444285 393527 439370 412854 358103 292575 391815 425777
## 2012 390293 396218 426110 374663 425948 400374 352022 276474 363500 422843
## 2013 377654 369868 379029 401276 414932 385704 347942 265569 363267 431048
## 2014 371433 368718 402037 378265 406246 378493 346198 265202 374527 432776
## 2015 362698 365159 402041 384260 407243 389351 353208 272852 380950 424072
## 2016 361910 383757 393063 415426 422798 402641 350687 289228 400076 425825
##         Nov    Dec
## 2005 451961 409424
```

```
## 2006 454167 407801
## 2007 447449 400014
## 2008 430151 400925
## 2009 422923 390794
## 2010 423383 388958
## 2011 418821 386528
## 2012 385639 353987
## 2013 388986 360875
## 2014 386548 366513
## 2015 405089 375002
## 2016 414340 377671
```

2.2 Representar gráficamente de la serie

```r
plot(viajeros_ini,
     xlab="año",
     ylab="viajeros",
     main="Total viajeros",
     col="blue")
```

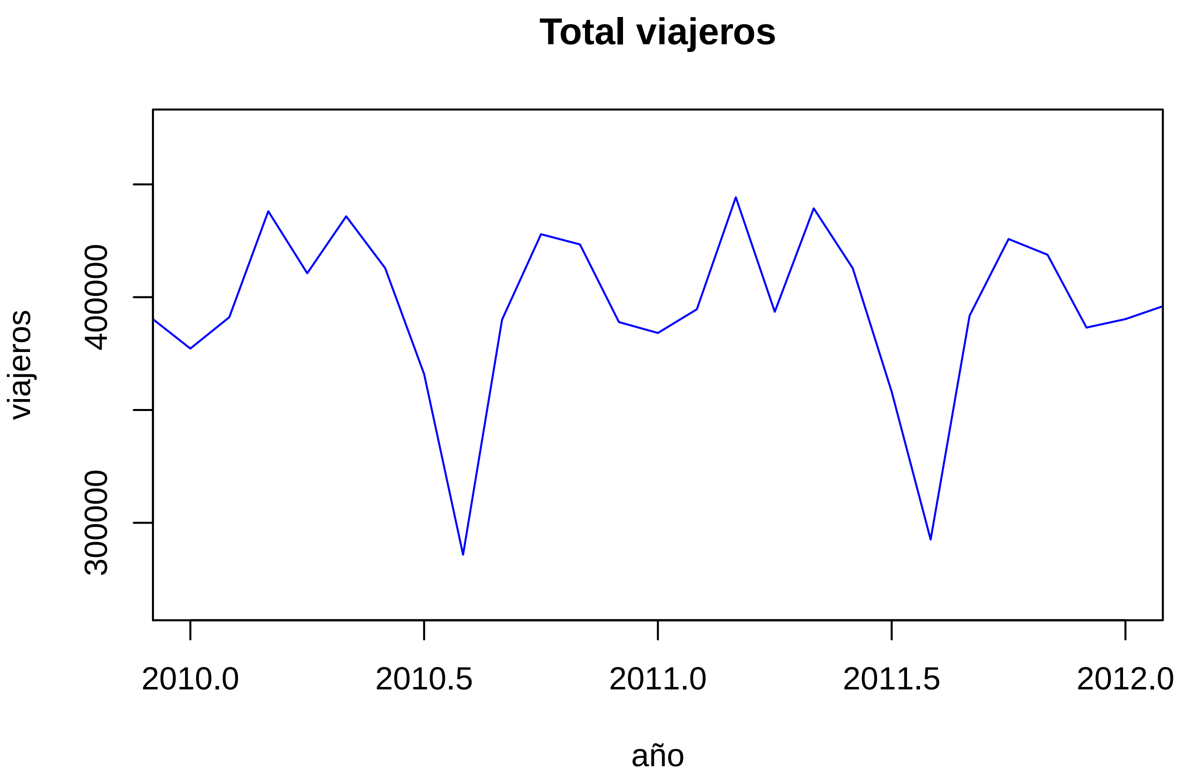


Observamos que la varianza no es constante, va cambiando, hay subidas y bajadas, aunque parece bastante homogénea.

La tendencia es creciente hasta el 2008, entre los años 2008 y 2014 presenta tendencia descendente, y a partir de 2014 tendencia creciente.

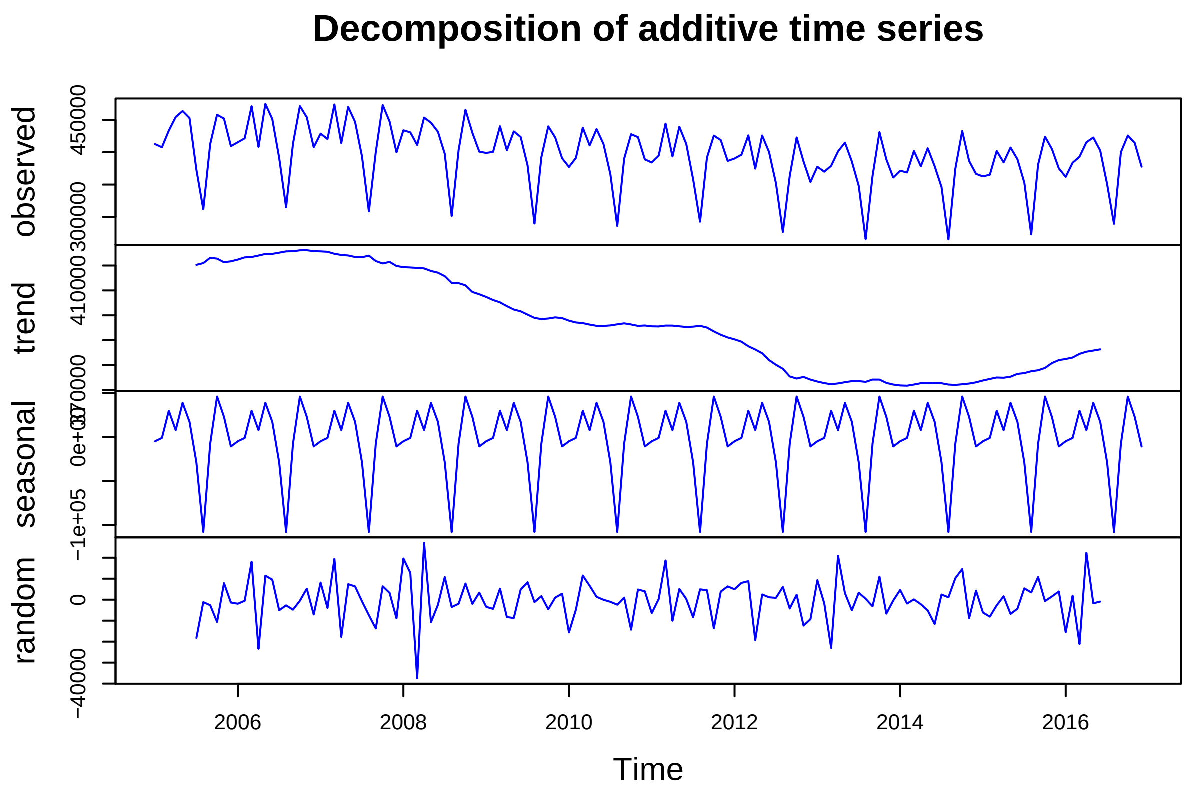
Se observa estacionalidad, pero vamos a hacer zoom entre los años centrales 2010 a 2012 para observar mejor la estacionalidad de la serie.

```
plot(viajeros_ini,
     xlab="año",
     ylab="viajeros",
     main="Total viajeros",
     col="blue",
     xlim=c(2010, 2012))
```



A continuación utilizaremos la funcion *decompose* con el objetivo de descomponer la serie temporal en las componentes estacional, tendencia e irregular, usando las medias móviles. y el modelo aditivo.

```
componentes_viajeros=decompose(viajeros_ini)
plot(componentes_viajeros, type="l", col="blue")
```




```
# Componente estacional
componentes_viajeros$seasonal
```

```
##              Jan         Feb         Mar         Apr         May
## 2005   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2006   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2007   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2008   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2009   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2010   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2011   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2012   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2013   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2014   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2015   -5027.082   -1191.158   29712.164    7700.361   38694.857
## 2016   -5027.082   -1191.158   29712.164    7700.361   38694.857
##              Jun         Jul         Aug         Sep         Oct
## 2005   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2006   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2007   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2008   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2009   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2010   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2011   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2012   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2013   17218.183  -28977.040 -108125.218   -7750.775   45884.460
```

```
## 2014   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2015   17218.183  -28977.040 -108125.218   -7750.775   45884.460
## 2016   17218.183  -28977.040 -108125.218   -7750.775   45884.460
##              Nov         Dec
## 2005   22771.130  -10909.881
## 2006   22771.130  -10909.881
## 2007   22771.130  -10909.881
## 2008   22771.130  -10909.881
## 2009   22771.130  -10909.881
## 2010   22771.130  -10909.881
## 2011   22771.130  -10909.881
## 2012   22771.130  -10909.881
## 2013   22771.130  -10909.881
## 2014   22771.130  -10909.881
## 2015   22771.130  -10909.881
## 2016   22771.130  -10909.881
```

```
# Componente tendencia
componentes_viajeros$trend
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 2005       NA       NA       NA       NA       NA       NA 420286.2
## 2006 422387.2 423272.3 423424.7 424004.5 424655.7 424680.0 425170.1
## 2007 425692.5 425528.0 424725.3 424255.3 424043.2 423438.8 423328.3
## 2008 419354.0 419221.5 419046.5 418856.5 417823.7 417141.0 415721.0
## 2009 407385.2 406154.2 405217.7 403700.5 402333.4 401610.1 400282.3
## 2010 397870.0 397140.4 396880.8 396276.0 395789.8 395732.4 395944.4
## 2011 395581.8 395534.7 395888.7 395874.5 395595.8 395304.5 395459.4
## 2012 390348.7 389424.5 387573.8 386271.8 384766.9 382028.5 380146.0
## 2013 373407.3 372783.0 372318.9 372651.0 373132.4 373558.8 373586.6
## 2014 371833.9 371746.0 372199.8 372741.0 372711.4 372844.7 372715.7
## 2015 373835.1 374445.9 375032.3 374937.2 375347.1 376473.4 376794.3
## 2016 382459.0 383036.2 384515.5 385385.5 385844.0 386340.6       NA
##           Aug      Sep      Oct      Nov      Dec
## 2005 420976.5 423125.3 422773.3 421305.4 421698.8
## 2006 425680.5 425745.7 426104.2 426156.0 425777.0
## 2007 423976.3 421801.4 420835.4 421463.6 419819.2
## 2008 413002.2 412946.5 412046.6 409379.1 408473.7
## 2009 398983.7 398501.9 398724.9 399185.5 398886.9
## 2010 396379.0 396783.2 396329.7 395765.0 395910.3
## 2011 395781.4 395089.9 393546.6 392201.3 391122.1
## 2012 378521.4 375461.8 374609.0 375258.8 374188.6
## 2013 373279.5 374190.2 374190.1 372869.4 372207.0
## 2014 372203.5 372055.3 372305.3 372596.6 373090.6
## 2015 377536.3 377937.2 378861.7 380808.4 382010.2
## 2016       NA       NA       NA       NA       NA
```

```
# Componente aleatoria
componentes_viajeros$random
```

```
##               Jan          Feb          Mar          Apr          May
## 2005           NA           NA           NA           NA           NA
## 2006  -1933.12626   -538.13384  18077.16919 -23399.86111  11436.47601
## 2007   8147.54040  -3928.80051  19475.50253 -17744.69444   7385.97601
## 2008  19621.08207  12794.69949 -37460.66414  27085.13889 -10747.60732
## 2009  -3399.12626  -4401.05051   5293.16919  -8279.90278  -8756.27399
```

```
## 2010 -15620.91793  -4817.25884  11497.00253   6630.68056   1356.39268
## 2011  -6409.66793    295.49116  18684.12753 -10047.86111   5079.30934
## 2012   4971.37374   7984.69949   8824.04419 -19309.11111   2486.22601
## 2013   9273.74874  -1723.80051 -23002.03914  20924.59722   3104.76768
## 2014   4626.16540  -1836.80051    125.00253  -2176.36111  -5160.27399
## 2015  -6110.00126  -8095.75884  -2703.45581   1622.38889  -6798.98232
## 2016 -15521.87626   1911.90783 -21164.66414  22340.18056  -1740.81566
##               Jun          Jul          Aug          Sep          Oct
## 2005           NA -18233.16793  -1176.23990  -2644.51641 -10602.75126
## 2006   9511.85859  -5038.04293  -2717.32323  -4770.93308   -511.70960
## 2007   6319.02525   -789.25126  -7370.07323 -13732.59975   6383.16540
## 2008  -2495.14141  10763.99874  -3522.98990  -1949.68308   7683.91540
## 2009   4805.69192   8307.74874  -1153.44823   1666.90025  -4577.37626
## 2010    -53.59975  -1022.33460  -2390.73990    997.56692 -14311.12626
## 2011    331.31692  -8379.37626   4918.84343   4475.90025 -13654.04293
## 2012   1127.35859    853.08207   6077.80177  -4211.01641   2349.58207
## 2013  -5073.01641   3332.41540    414.71843  -3172.47475  10973.41540
## 2014 -11569.93308   2459.33207   1123.76010  10222.44192  14586.24874
## 2015  -4340.55808   5390.79040   3440.88510  10763.60859   -674.12626
## 2016   -917.80808           NA           NA           NA           NA
##               Nov          Dec
## 2005   7884.45328  -1364.91035
## 2006   5239.82828  -7066.11869
## 2007   3214.24495  -8895.36869
## 2008  -1999.25505   3361.13131
## 2009    966.32828   2817.00631
## 2010   4846.82828   3957.58965
## 2011   3848.53662   6315.79798
## 2012 -12390.96338  -9291.70202
## 2013  -6654.54672   -422.16035
## 2014  -8819.75505   4332.29798
## 2015   1509.49495   3901.63131
## 2016           NA           NA
```

Tanto la componente estacional como la aleatoria son despreciables con respecto a los datos observados, puesto que estos últimos son varios órdenes de magnitud mayores. Por tanto, la influencia de estas dos componentes será pequeña.

2.3 Determinar si es estacionaria o necesita alguna transformación previa

2.3.1 Homogeneidad de varianzas

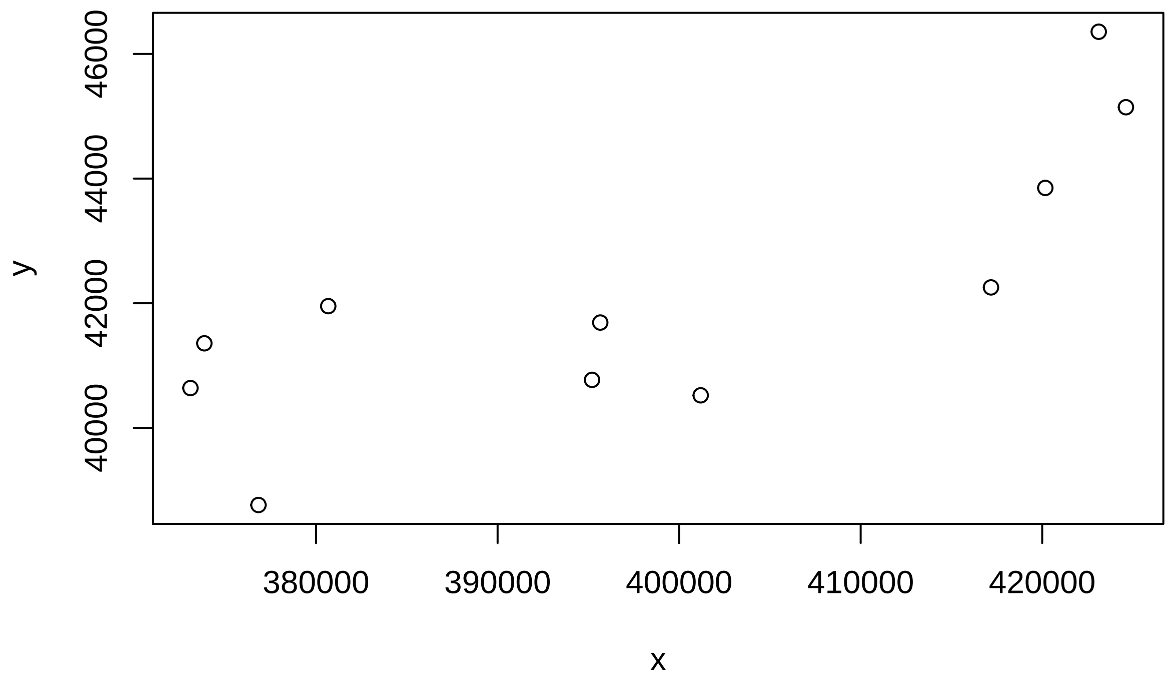
Respecto a la homogeneidad de la varianza considerar tan sólo la transformación logarítmica (no buscar otra transformación dentro de la familia Box-Cox)

Dado que de la representación gráfica del apartado anterior observamos que la varianza no era constante, en primer lugar realizaremos un estudio de la homogeneidad de varianzas:

```
# Instante de inicio de la serie
start=start(viajeros_ini)

# Instante de fin de la serie
end=end(viajeros_ini)

cat("Número de años en la serie= ", end-start)
```

```
## Número de años en la serie=  11 11
```

```r
anuales <- matrix(viajeros_ini, nr=12, byrow=F)
x <- c(rep(0, 11))
y <- c(rep(0, 11))
for(i in 1:11){
  x[i] <- mean(anuales[,i])
  y[i] <- sd(anuales[,i])
}
plot(x,y)
```



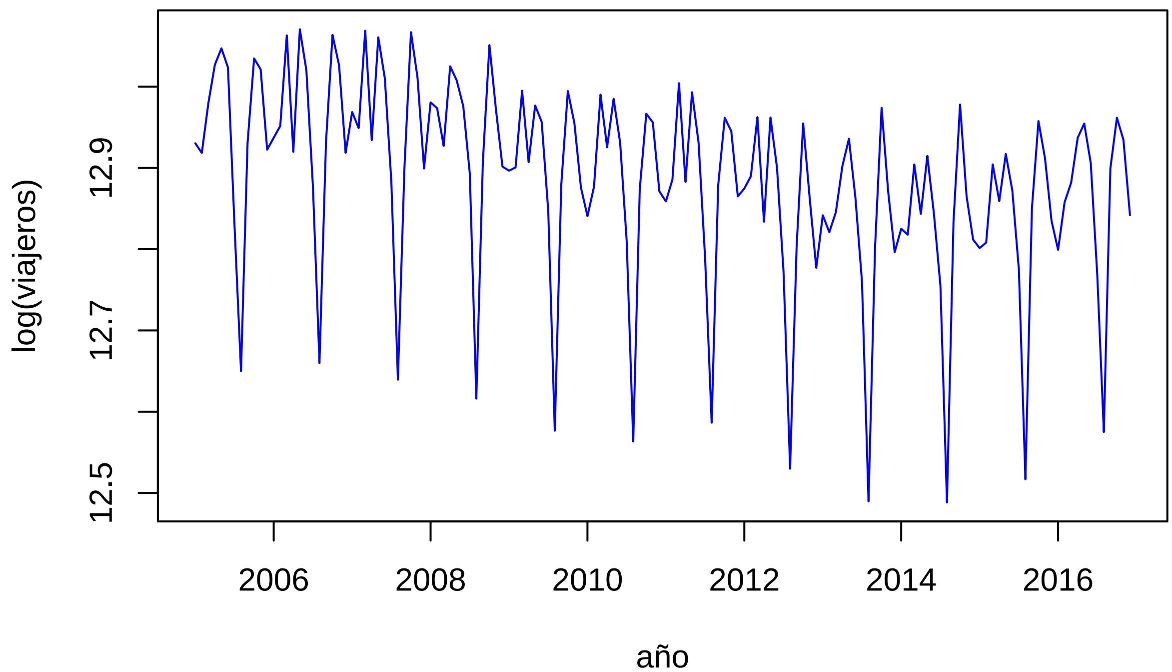
Se aplica la transformación logarítmica a los datos y se calcula el modelo lineal

```r
x <- log(x)
y <- log(y)
regresion <- lm(y ~ x)
regresion
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##      0.4336       0.7921
```

1-lambda = 0.7921, por tanto basta tomar *lambda* igual a 0, transformación logarítmica, en la familia Box-Cox.

```r
viajeros_log <- log(viajeros_ini)
plot(viajeros_log, col="blue", xlab="año", ylab="log(viajeros)")
```

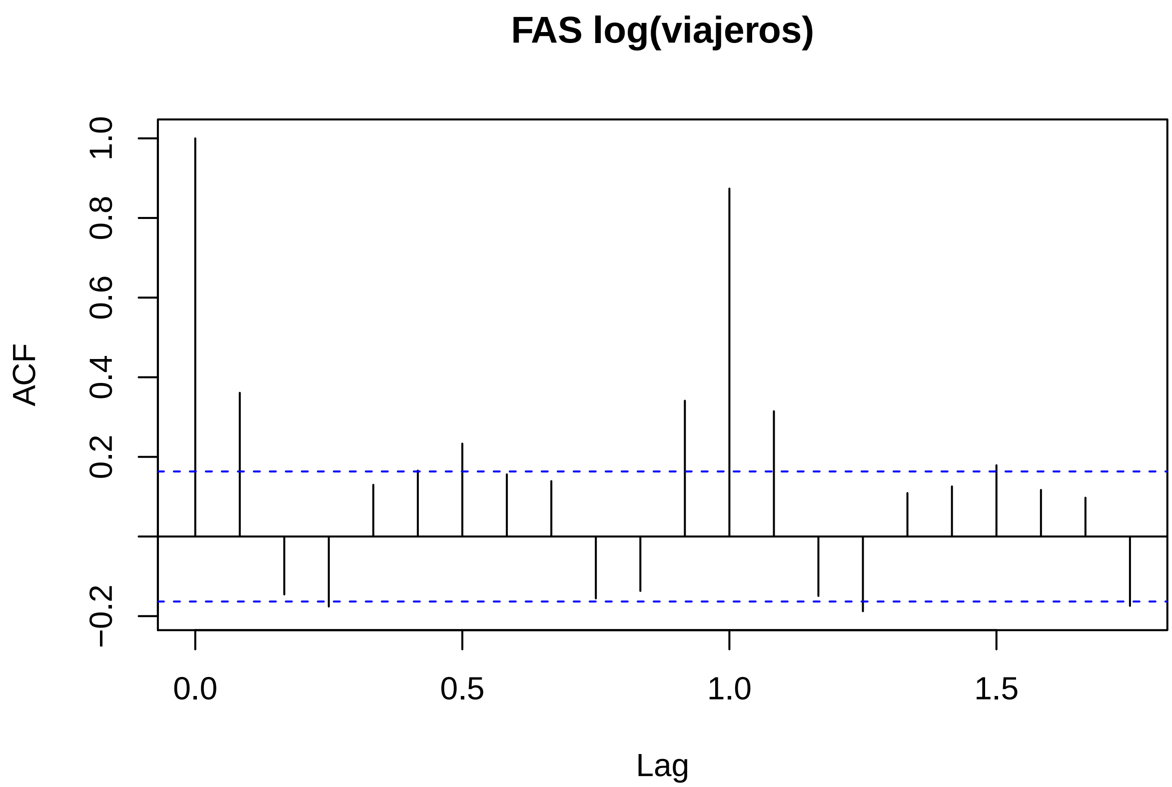


Observamos que no hemos ganado mucho con la transformación logarítmica.

2.3.2 Estacionariedad en medias

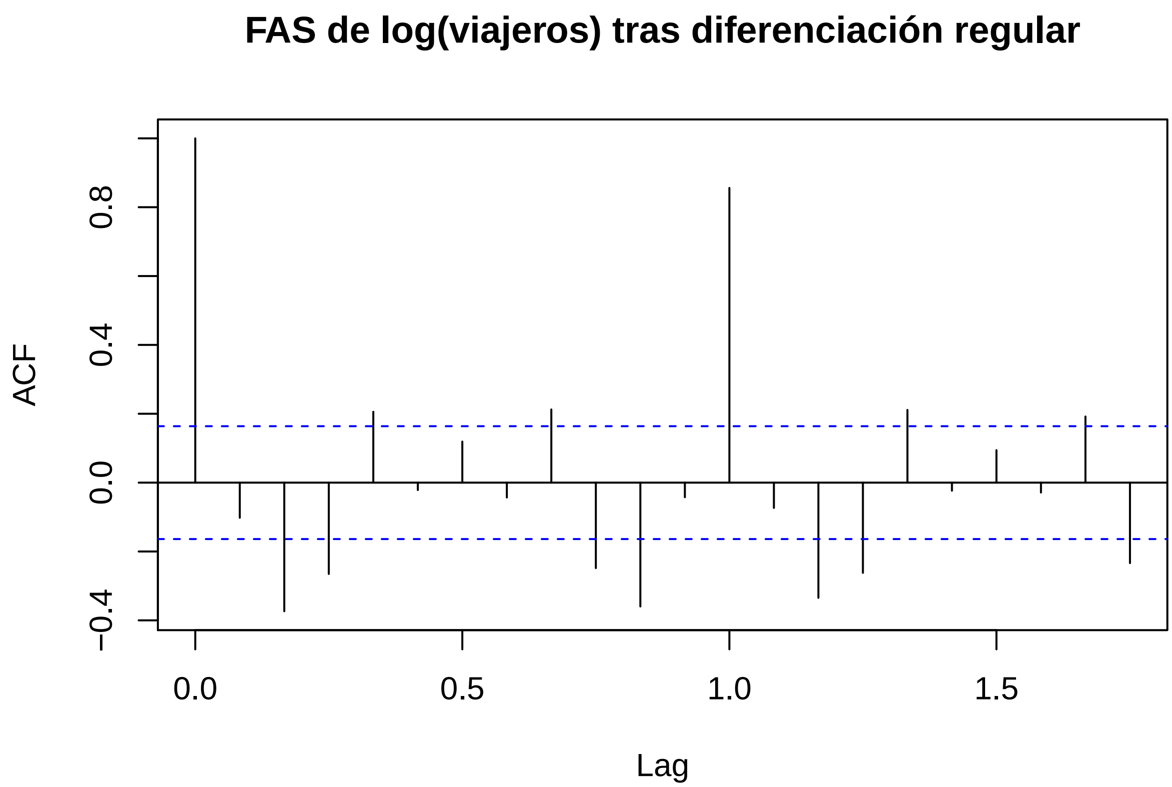
A continuación representaremos la funcion autocorrelación estimada de la serie para realizar un análisis visual, y determinar si es necesario realizar algún tipo de diferenciación de la serie para asegurar la estacionariedad en la media

```r
acf(viajeros_log, main="FAS log(viajeros)")
```



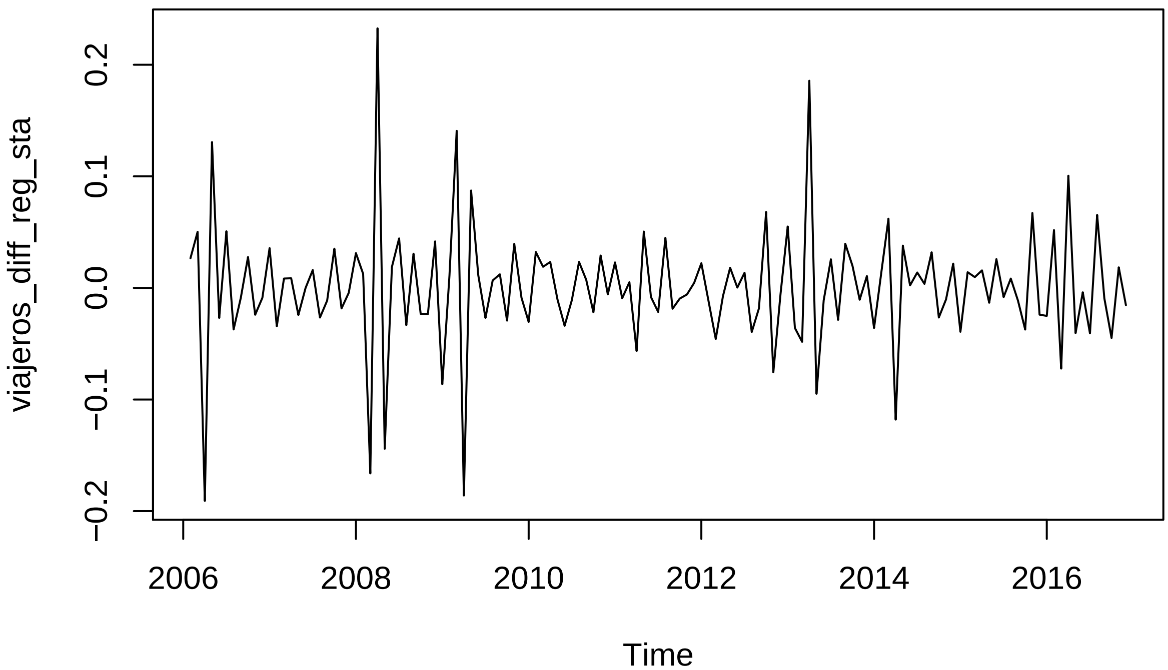
Aplicaremos una diferenciación de orden 1 a la serie sucesivamente hasta que exista estacionariedad en media.

```
viajeros_diff_regular <- diff(viajeros_log, lag = 1, differences = 1)
acf(viajeros_diff_regular, main="FAS de log(viajeros) tras diferenciación regular")
```

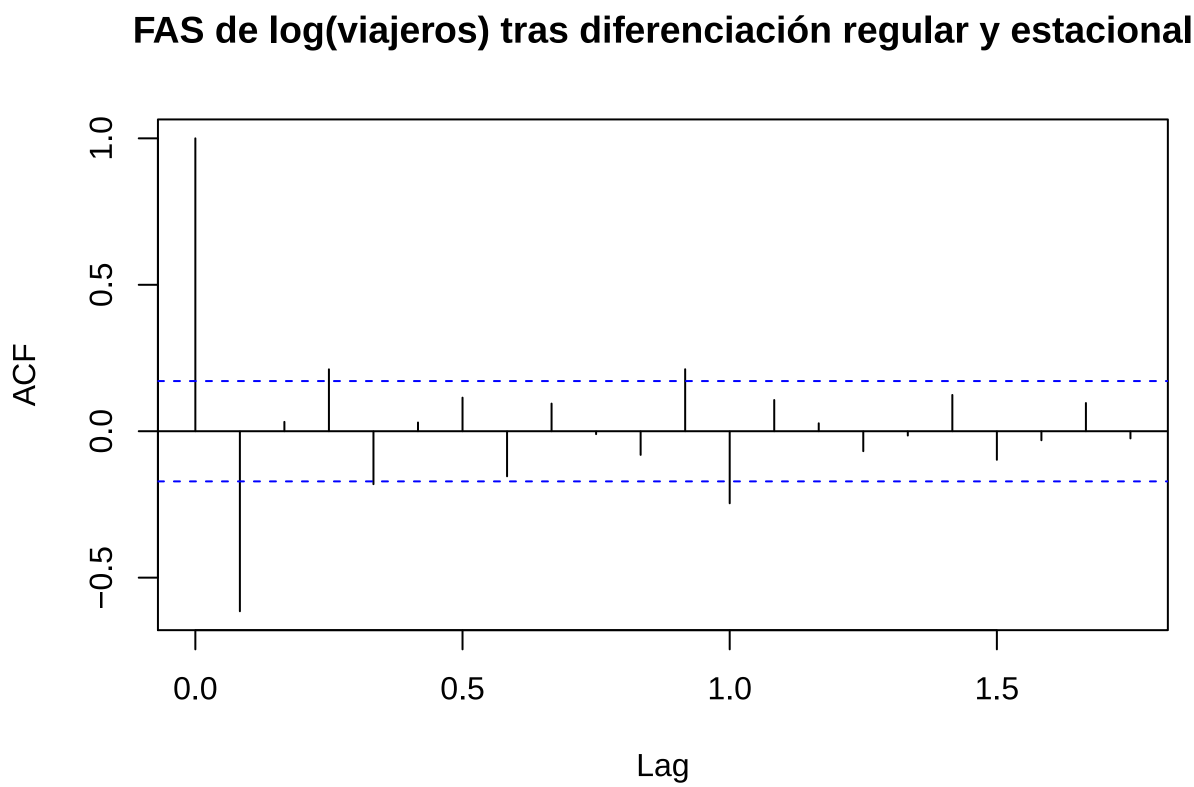


Observando la función de autocorrelación vemos que existe estacionalidad de orden 12, por lo que es necesario aplicar diferenciaciones de orden 12 a la serie para eliminar la componente estacional.

```r
viajeros_diff_reg_sta <- diff(viajeros_diff_regular, lag = 12, differences = 1)
plot(viajeros_diff_reg_sta)
```




```r
acf(viajeros_diff_reg_sta, main="FAS de log(viajeros) tras diferenciación regular y estacional")
```



2.4 Contrastar si la serie transformada puede considerarse estacionaria

Para determinar si la serie diferenciada ya es estacionaria se suele aplicar un contraste de raíz unitaria. Este contraste es conocido como el contraste de raíz unitaria ampliado de *Dickey-Fuller* (ADF).

Contrasta la hipótesis nula de existencia de una raíz unitaria contra la alternativa de que no existen raíces unitarias

H0: beta=1 existe raíz unitaria ==> No estacionaria

H1: beta<1 no existe raíz unitaria

```r
adf.test(viajeros_diff_reg_sta)
```

```
## Warning in adf.test(viajeros_diff_reg_sta): p-value smaller than printed p-
## value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  viajeros_diff_reg_sta
## Dickey-Fuller = -6.7522, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

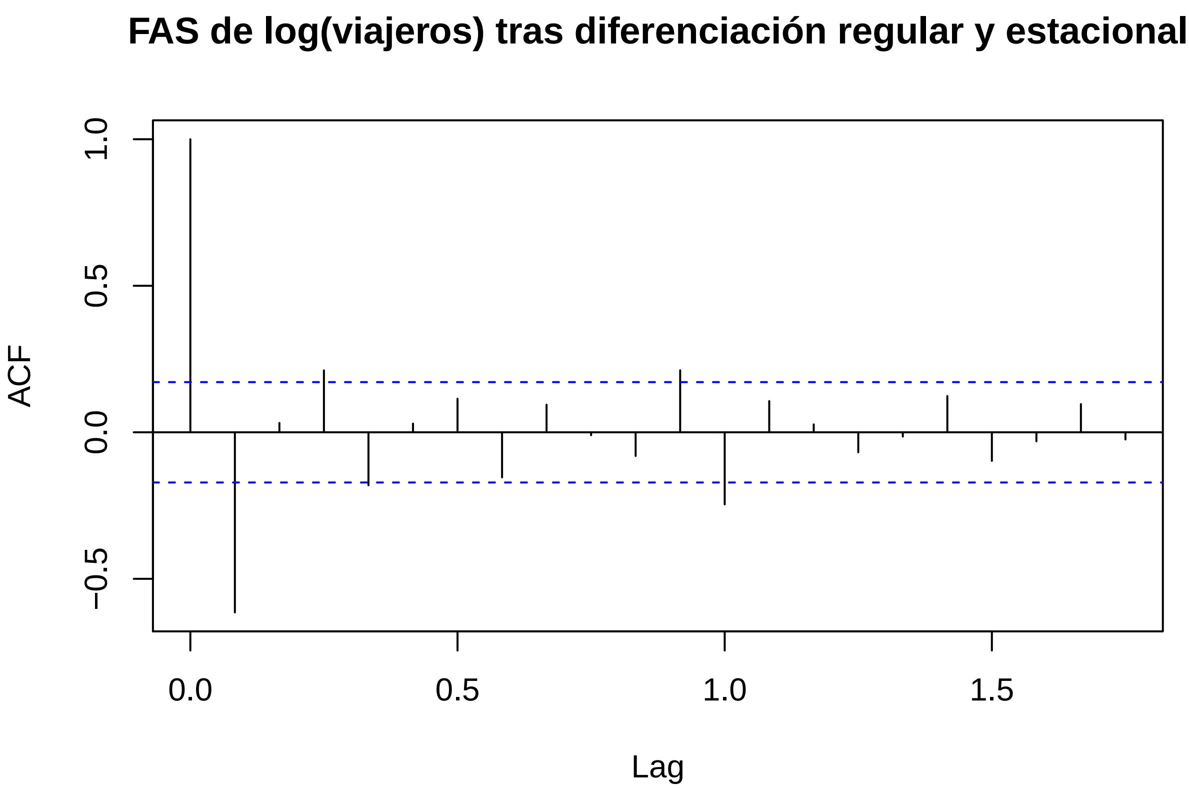
pvalor=0.01 < 0.05, con lo que se rechaza la hipótesis nula, y por tanto se admite que **la serie transformada es estacionaria**.

2.5 Identificar la estructura ARIMA de los datos

Las funciones autocorrelacion estimadas (FAS) y autocorrelación parcial muestral (FAP) se suelen utilizar para identificar el modelo *ARIMA(p, d, q)×(P,D,Q)s* que mejor se ajusta a los datos.

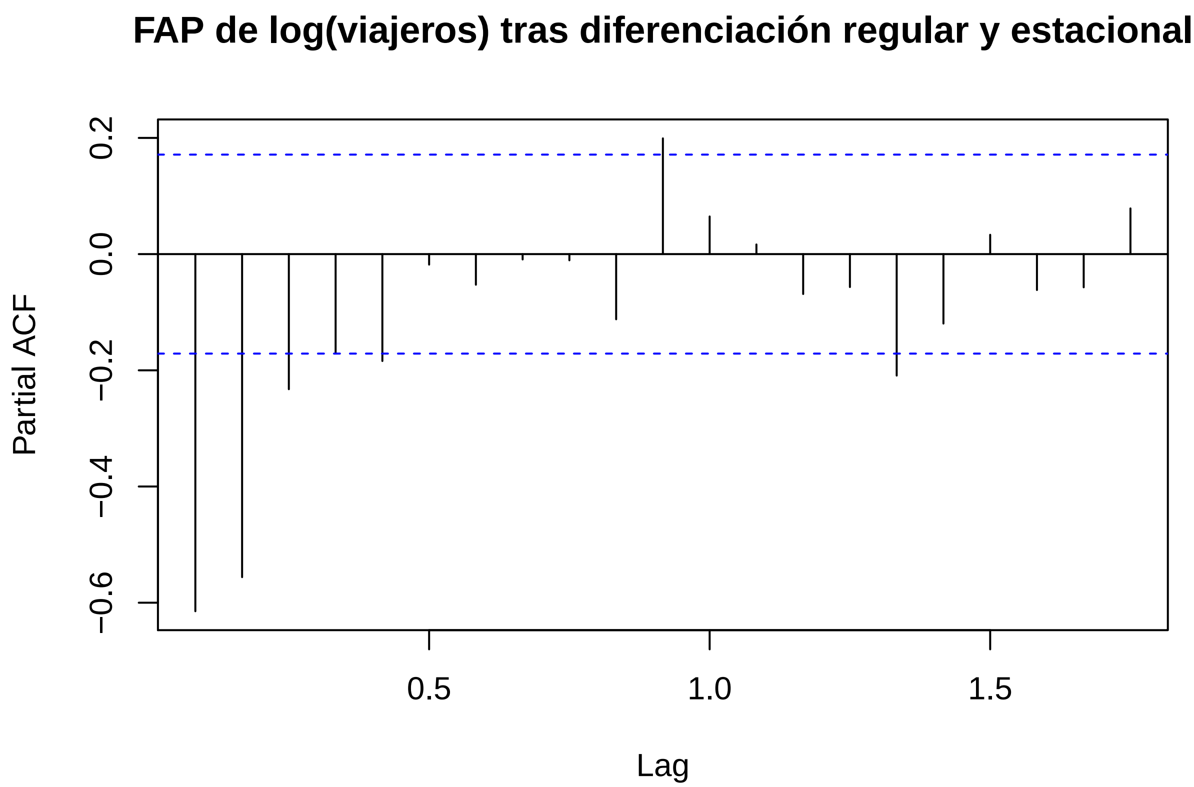
Se utilizará como herramienta el análisis visual de las funciones *FAS* y *FAP* para determinar los parámetros p, d y q de la parte no estacional y los parámetros P, D, Q de la parte estacional del modelo.

```
# FAS
acf(viajeros_diff_reg_sta, main="FAS de log(viajeros) tras diferenciación regular y estacional")
```



Esta es la parte correpondiente al modelo *MA* medias móviles, hay 6 componentes significativas, pero nos quedaremos en MA(1) para simplificar al máximo ya que vamos a añadir un modelo *AR* también en la parte estacional

```
# FAP
pacf(viajeros_diff_reg_sta, main="FAP de log(viajeros) tras diferenciación regular y estacional")
```



Probaremos con:

- parte regular: *MA(1)*
- parte estacional: *AR(2)* o *ARMA(1,1)*
- El parámetro d es igual a 0 porque con las transformaciones hemos eliminado la parte estacional.

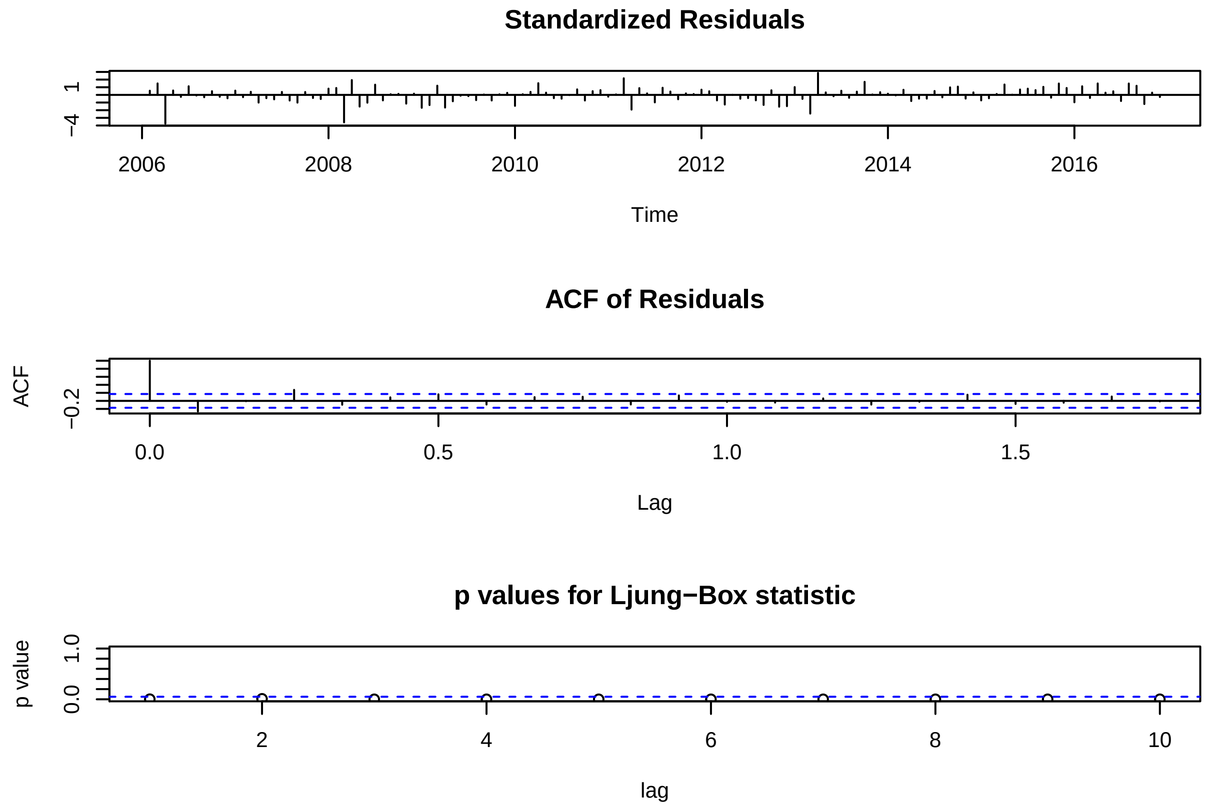
Probaremos por tanto con los siguientes modelos:

- *ARIMA(0, 0, 1)× (2,0,0)12*
- *ARIMA(0, 0, 1)× (1,0,1)12*

A continuación verficiaremos ambos modelos utilizando el estadístico de *Ljung y Box*

Modelo ARIMA(0, 0, 1)× (2,0,0)12

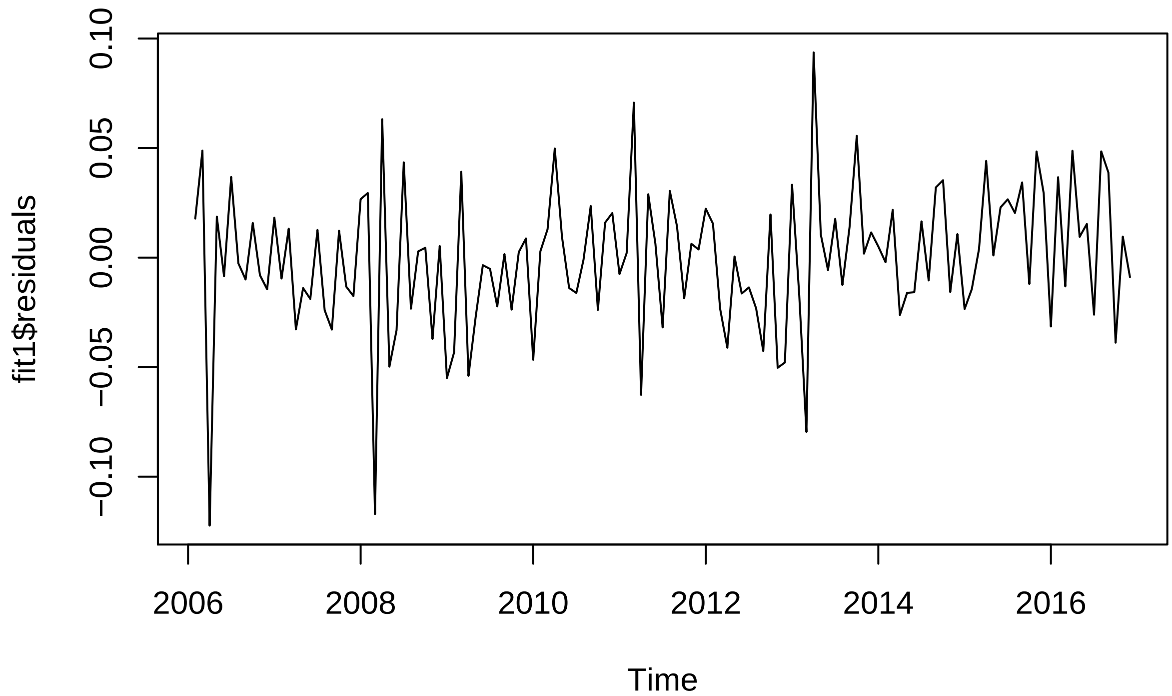
```
fit1 <- arima(viajeros_diff_reg_sta,
              order=c(0,0,1),
              seasonal = list(order = c(2, 0, 0), period = 12))

tsdiag(fit1)
```




```
Box.test (fit1$residuals, lag = 1, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  fit1$residuals
## X-squared = 9.2939, df = 1, p-value = 0.002299
```

```
plot(fit1$residuals)
```

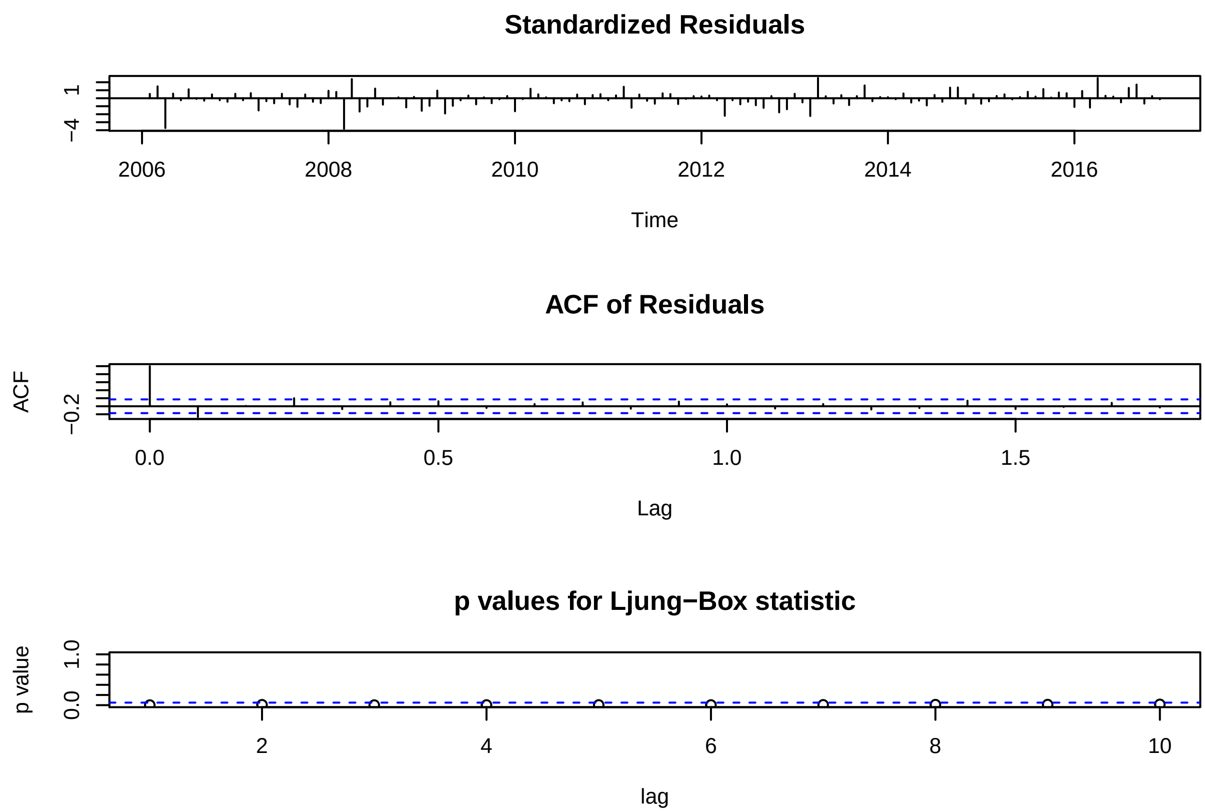



```r
fit1
```

```
##
## Call:
## arima(x = viajeros_diff_reg_sta, order = c(0, 0, 1), seasonal = list(order = c(2,
##     0, 0), period = 12))
##
## Coefficients:
##           ma1     sar1     sar2  intercept
##       -0.8318  -0.4137  -0.3997      1e-04
## s.e.   0.0398   0.0893   0.0823      3e-04
##
## sigma^2 estimated as 0.00106:  log likelihood = 259.47,  aic = -508.94
```

Modelo ARIMA(0, 0, 1)× (1,0,1)12

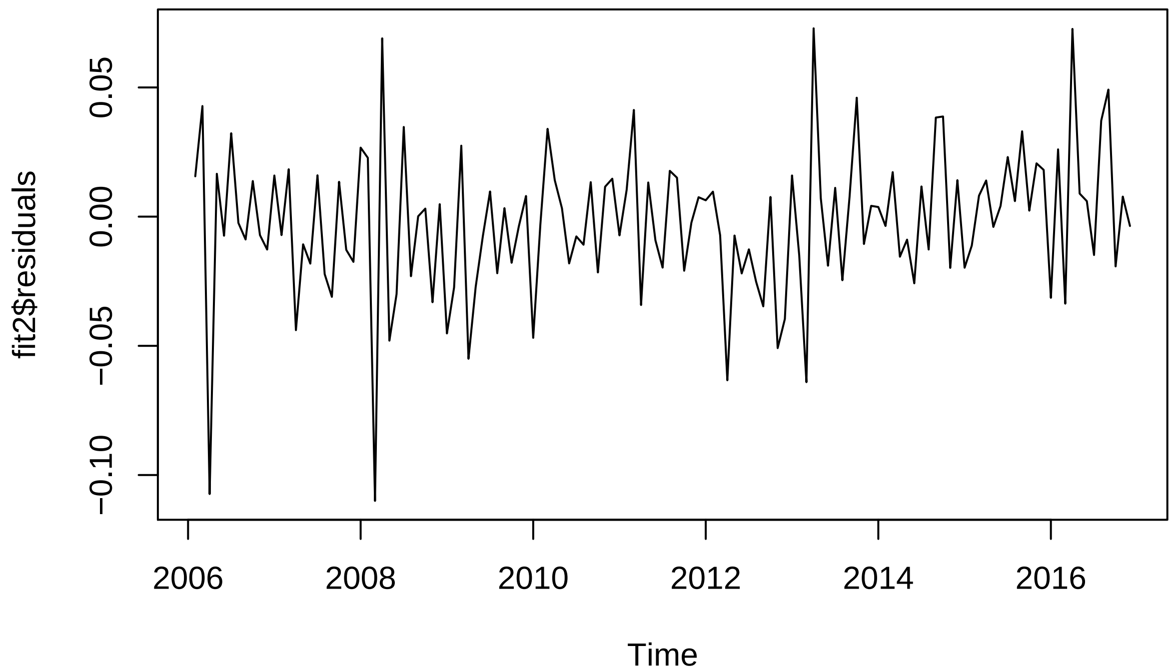
```r
fit2 <- arima(viajeros_diff_reg_sta,
              order=c(0,0,1),
              seasonal = list(order = c(1, 0, 1), period = 12))
tsdiag(fit2)
```




```
Box.test (fit2$residuals, lag = 1, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  fit2$residuals
## X-squared = 9.5412, df = 1, p-value = 0.002009
```

```
plot(fit2$residuals)
```




```r
fit2
```

```
##
## Call:
## arima(x = viajeros_diff_reg_sta, order = c(0, 0, 1), seasonal = list(order = c(1,
##     0, 1), period = 12))
##
## Coefficients:
##           ma1    sar1     sma1  intercept
##       -0.8246  0.1721  -0.9997      1e-04
## s.e.   0.0396  0.1050   0.1222      2e-04
##
## sigma^2 estimated as 0.0008282:  log likelihood = 265.31,  aic = -520.62
```

Para ninguno de los dos modelos obtenidos se acepta la hipótesis de que los residuos del modelo se pueden considerar que provienen de un ruido blanco. Entre ambos modelos el que presenta menor AIC y por tanto se ajusta mejor es el modelo *fit2* (ARIMA(0, 0, 1)× (1,0,1)12).

Puede que los resultados obtenidos se deban a que hemos simplificado en exceso el modelo. En el siguiente apartado se realizará un mejor ajuste de los parámetros del modelo con ayuda de la función *auto.arima*.

2.6 Estimar los parámetros del modelo

Utilizaremos la función *auto.arima* de la librería *forecast* para obtener los parámetros del modelo ARIMA que mejor se ajustan a los datos. Esta función evalúa entre todos los posibles modelos, considerando diversos criterios (estacionariedad, estacionalidad, diferencias), y devuelve el que presente menor AIC (o el criterio de información especificado en la llamada)

```r
auto.fit = auto.arima(viajeros_diff_reg_sta, approximation=FALSE, trace=FALSE)
```

```r
summary(auto.fit)
```

```
## Series: viajeros_diff_reg_sta
## ARIMA(0,0,2)(2,0,1)[12] with zero mean
##
## Coefficients:
##           ma1     ma2    sar1     sar2     sma1
##       -1.1398  0.3751  0.1347  -0.2243  -0.8393
## s.e.   0.0905  0.0851  0.1366   0.1124   0.1794
##
## sigma^2 estimated as 0.0007858:  log likelihood=275.76
## AIC=-539.53   AICc=-538.85   BIC=-522.28
##
## Training set error measures:
##                          ME       RMSE        MAE      MPE     MAPE
## Training set -0.0005326919 0.02749277 0.02098188 12352.47 12567.29
##                   MASE       ACF1
## Training set 0.3915424 0.03231667
```

El método auto.arima nos devuelve que el modelo que mejor se ajusta es el *ARIMA(0,0,2)(2,0,1)[12]*

2.7 Diagnosticar y seleccionar el modelo final

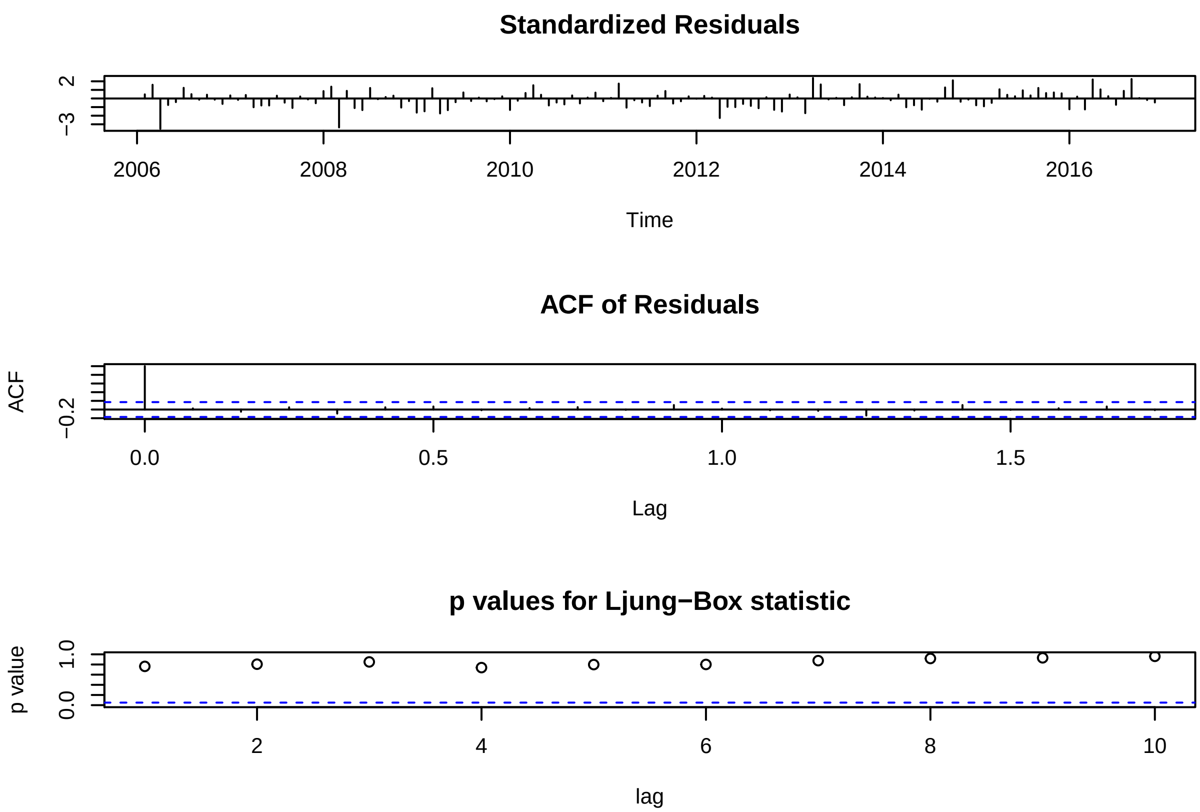
A continuación construiremos el modelo final devuelto por la función de ajuste auto.arima y que presenta mejor ajuste a los datos

```r
best.fit <- arima(viajeros_diff_reg_sta,
                  order=c(0,0,2),
                  seasonal = list(order = c(2,0,1), period = 12))
```

```r
summary(best.fit)
```

```
##
## Call:
## arima(x = viajeros_diff_reg_sta, order = c(0, 0, 2), seasonal = list(order = c(2,
##     0, 1), period = 12))
##
## Coefficients:
##           ma1     ma2    sar1     sar2     sma1  intercept
##       -1.1445  0.3723  0.1426  -0.2218  -0.8698      1e-04
## s.e.   0.0909  0.0863  0.1367   0.1130   0.2107      2e-04
##
## sigma^2 estimated as 0.0007402:  log likelihood = 276.12,  aic = -538.24
##
## Training set error measures:
##                         ME       RMSE        MAE      MPE     MAPE     MASE
## Training set -0.002423155 0.02720699 0.02076106 12703.38 12933.76 0.314055
##                    ACF1
## Training set 0.02605552
```

```r
tsdiag(best.fit)
```



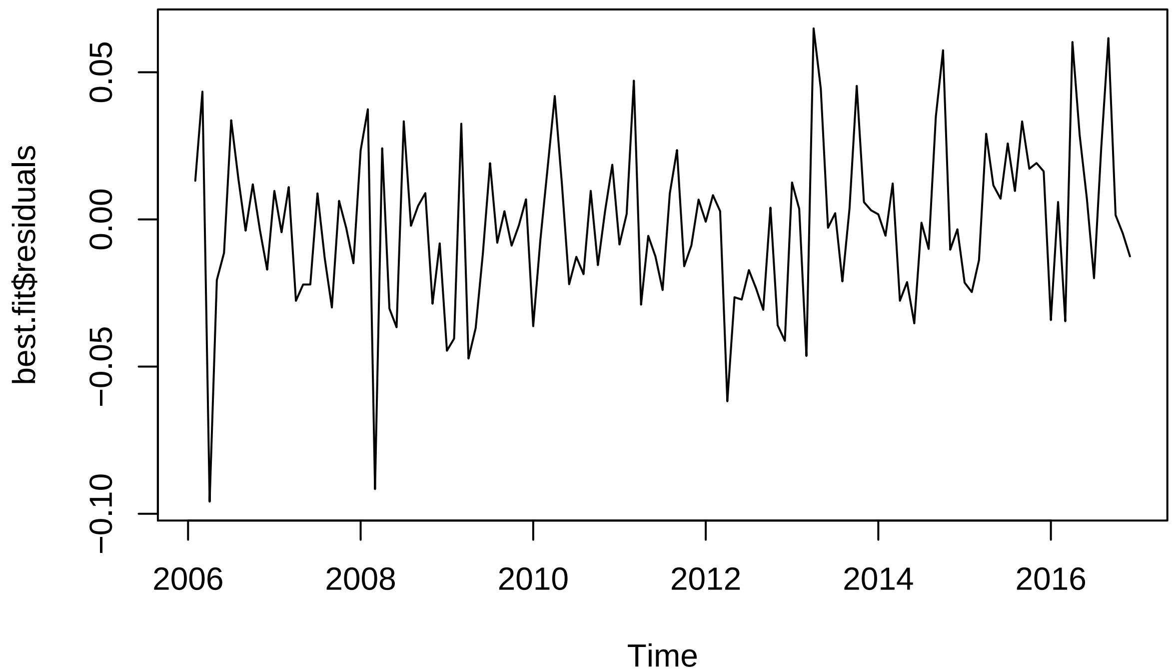
Por último, vamos a verificar el modelo mediante el estadístico de *Ljung y Box*

```
Box.test(best.fit$residuals, lag = 1, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  best.fit$residuals
## X-squared = 0.090987, df = 1, p-value = 0.7629
```

En este caso el $pvalor$=0.7629 > 0.05, por lo que se acepta la hipótesis nula, y por tanto los residuos provienen de un ruido blanco.

```
plot(best.fit$residuals)
```




```
best.fit
```

```
##
## Call:
## arima(x = viajeros_diff_reg_sta, order = c(0, 0, 2), seasonal = list(order = c(2,
##     0, 1), period = 12))
##
## Coefficients:
##           ma1     ma2    sar1     sar2     sma1  intercept
##       -1.1445  0.3723  0.1426  -0.2218  -0.8698      1e-04
## s.e.   0.0909  0.0863  0.1367   0.1130   0.2107      2e-04
##
## sigma^2 estimated as 0.0007402:  log likelihood = 276.12,  aic = -538.24
```

El AIC de este modelo es el menor de todos los modelos obtenidos:

```
cat("AIC del modelo final (best.fit) = ", best.fit$aic)
```

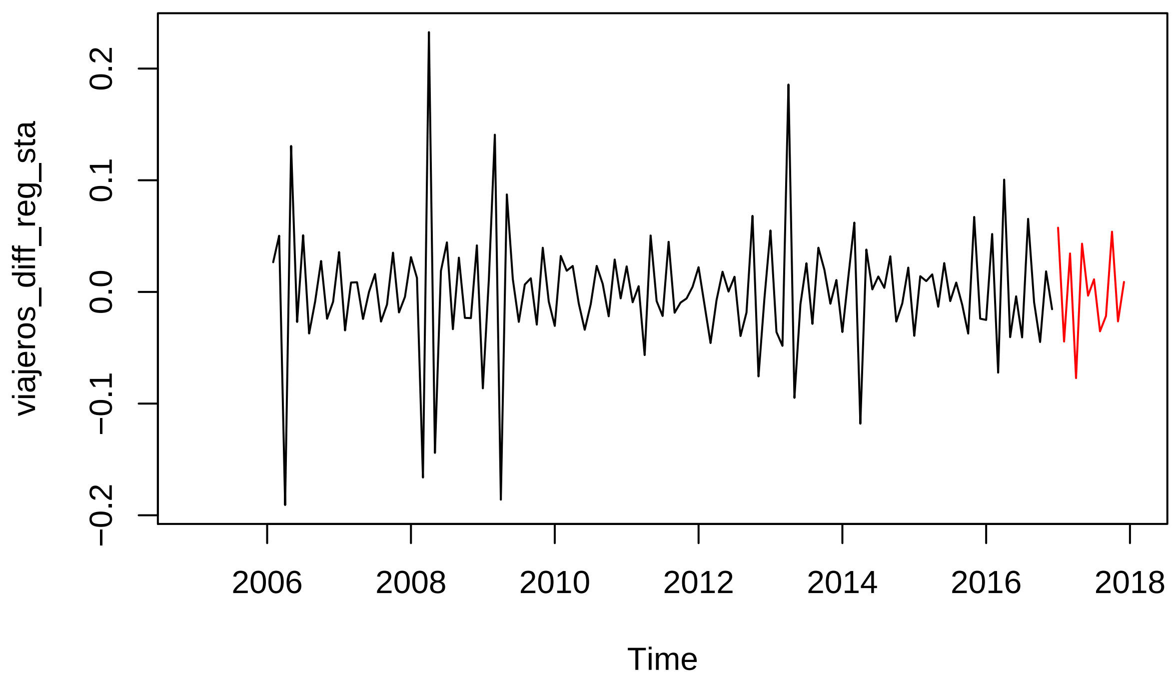
```
## AIC del modelo final (best.fit) =  -538.2354
```

2.8 Predecir la serie temporal para el año siguiente al último dato disponible

En primer lugar representaremos la serie temporal transformada y la predicción de un año

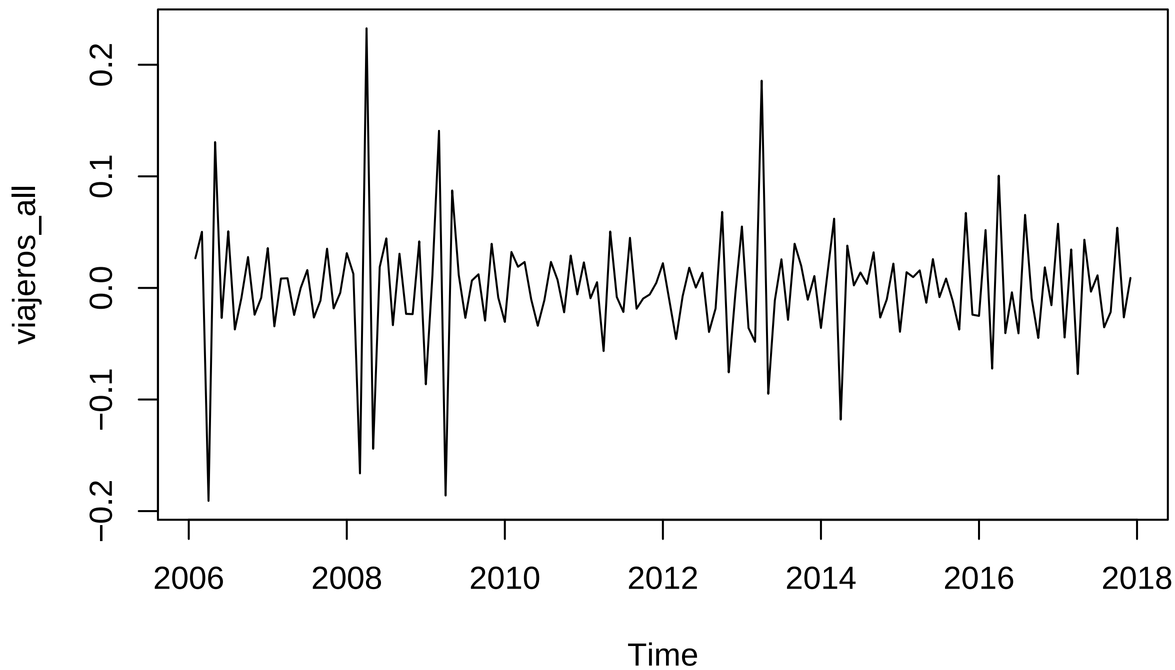
```
plot(viajeros_diff_reg_sta,
     xlim=c(2005, 2018))
```

```r
viajeros.pred<-predict(best.fit, n.ahead=12)

lines(viajeros.pred$pred, col="red")
```



A continuación construiremos la serie *viajeros_all* con la concatenación de las observaciones de la serie original y las predicciones

```r
viajeros_all <- c(viajeros_diff_reg_sta[1L:131L],  # serie original => (2017-2005)*12-13=131
                  viajeros.pred$pred[1L:12L])      # predicción

# Hemos perdido 13 datos porque hicimos una diferenciacion de orden 12 y otra de orden 1.
# Por tanto, los datos comienzan 13 meses despues de la serie original (enero 2005)
# Es decir, febrero de 2006
viajeros_all <- ts(viajeros_all, start=c(2006, 2), freq=12)
plot(viajeros_all, type="l")
```



Deshacemos la diferenciación de orden 12 y a continuación la de orden 1, tomando como datos los originales con los que se realizó la diferenciación correspondiente

```
viajeros_all <- diffinv(viajeros_all,
                        lag = 12,
                        differences = 1,
                        xi = c(viajeros_diff_regular[1],  viajeros_diff_regular[2],
                               viajeros_diff_regular[3],  viajeros_diff_regular[4],
                               viajeros_diff_regular[5],  viajeros_diff_regular[6],
                               viajeros_diff_regular[7],  viajeros_diff_regular[8],
                               viajeros_diff_regular[9],  viajeros_diff_regular[10],
                               viajeros_diff_regular[11], viajeros_diff_regular[12]))

viajeros_all <- diffinv(viajeros_all,
                        lag = 1,
                        differences = 1,
                        xi = viajeros_log[1])
```

Deshacemos la transformacion logarítmica

```
viajeros_all = exp(viajeros_all)
```

Construimos la serie temporal completa con la misma fecha de inicio que la original

```
viajeros_all <- ts(viajeros_all, start=2005, freq=12)
```

Representación gráfica de la serie temporal completa y la predicción

```
plot(viajeros_all, type="l",
     xlab="año",
     ylab="viajeros",
     xlim=c(2005, 2018),
     col="red")

# superposición de la serie original
lines(viajeros_ini, col="blue")
```