

Procesos Estocásticos y Series Temporales

Entregable Práctica 2 y 3 (Modelo B)

Francisco Javier Mercader Martínez

```
set.seed(1234)
```

En el fichero **Consumo_gas_modelo_B.csv** se encuentran los datos correspondientes al consumo de gas natural mensual en España (en gigavatios hora), desde enero de 2004 hasta diciembre de 2014.

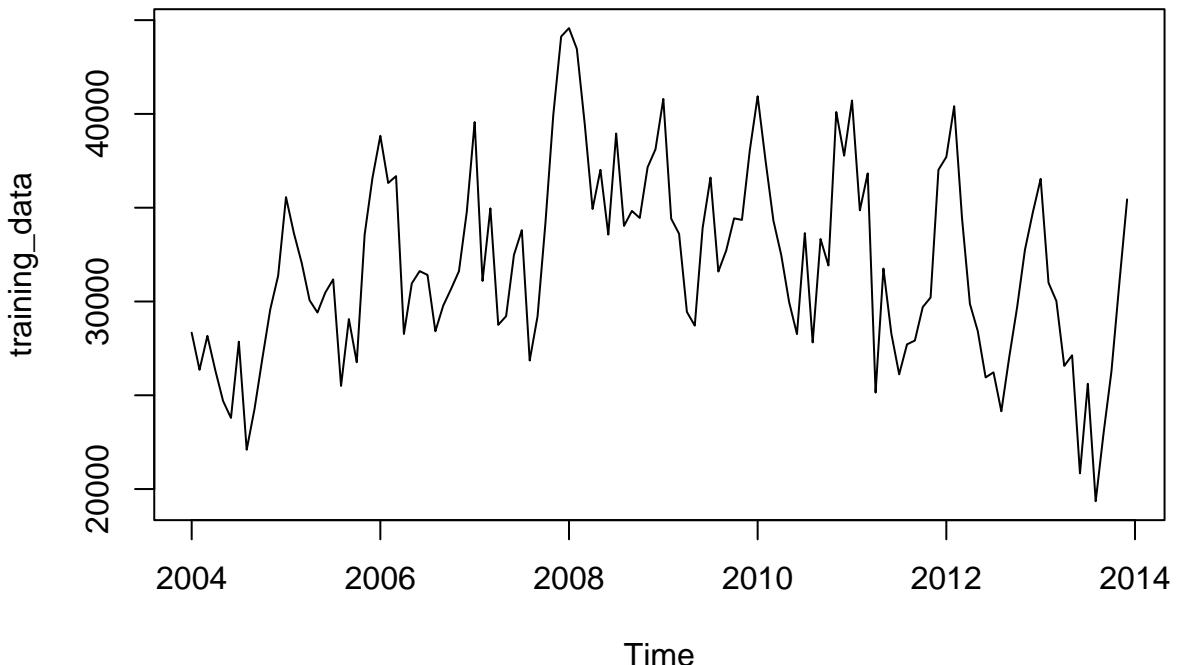
1. **DIVIDE la serie temporal en datos de entrenamiento** (hasta diciembre de 2013) y **datos de test** (desde enero a diciembre de 2014). Representa los **datos de entrenamiento** del consumo de gas en un gráfico temporal. ¿La serie presenta Estacionalidad? Justifica si se trata de un esquema aditivo o multiplicativo realizando un gráfico de desviaciones típicas frente a medias.

```
data <- read.csv("Consumo_gas_modelo_B.csv", sep = "\t", dec = ",")
```

```
# Convertir los datos a una serie temporal  
datos_ts <- ts(data$consumo, frequency = 12, start = c(2004, 1))  
datos_ts
```

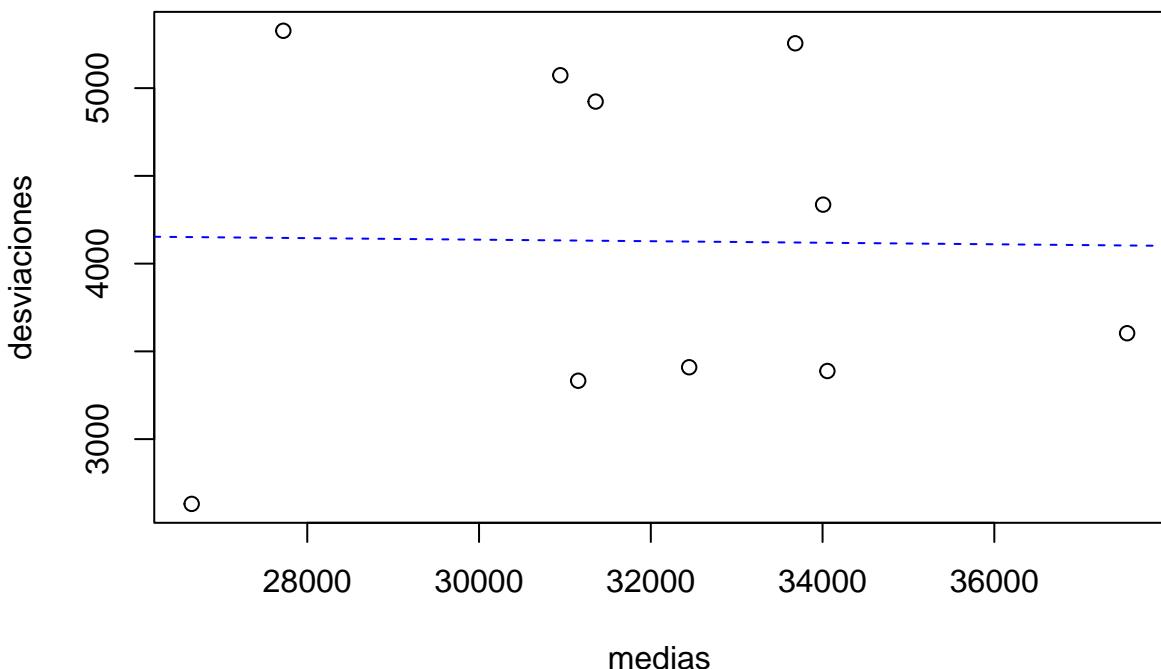
```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug  
## 2004 28332.94 26356.30 28167.68 26340.88 24691.32 23792.17 27855.36 22094.89  
## 2005 35564.12 33650.69 32052.25 30064.49 29410.96 30474.40 31177.07 25497.68  
## 2006 38834.51 36317.30 36683.55 28267.67 30962.82 31620.18 31416.70 28411.95  
## 2007 39566.13 31094.47 34963.70 28755.11 29217.46 32485.82 33802.84 26851.44  
## 2008 44574.45 43464.16 39463.35 34932.22 37020.02 33565.06 38954.58 34029.32  
## 2009 40800.76 34415.46 33611.36 29443.97 28712.06 33926.23 36607.73 31595.52  
## 2010 40942.23 37523.43 34318.03 32470.02 29974.08 28260.58 33646.27 27816.72  
## 2011 40717.21 34860.22 36830.10 25143.67 31754.86 28301.32 26116.84 27713.95  
## 2012 37706.67 40413.12 34427.31 29871.92 28423.73 25950.75 26221.28 24145.22  
## 2013 36538.94 30995.42 30023.76 26567.58 27130.33 20835.74 25610.79 19350.57  
## 2014 31292.43 29416.54 25077.20 23542.06 23165.35 21523.45 20860.68 20846.69  
##           Sep      Oct      Nov      Dec  
## 2004 24280.57 26982.45 29580.68 31372.95  
## 2005 29062.20 26758.70 33555.24 36589.52  
## 2006 29775.98 30670.04 31614.83 34785.09  
## 2007 29206.31 34168.97 39944.10 44125.81  
## 2008 34828.99 34452.89 37156.55 38113.80  
## 2009 32720.47 34432.01 34348.70 38059.66  
## 2010 33330.58 31918.27 40101.53 37774.71  
## 2011 27917.36 29701.26 30213.00 37021.56  
## 2012 27025.86 29676.26 32755.10 34739.76  
## 2013 22965.45 26254.78 30935.42 35443.71  
## 2014 23909.11 23377.34 24657.49 30447.22
```

```
# Dividir la serie en datos de entrenamiento y datos de test  
training_data <- window(datos_ts, start = c(2004, 1), end = c(2013, 12))  
test_data <- window(datos_ts, start = c(2014, 1))  
  
ts.plot(training_data)
```



Al representar los datos del conjunto de entrenamiento podemos observar analíticamente que la serie temporal presenta componente estacional ya que muestra un comportamiento que se ha repetido su comportamiento con un periodo de 12 meses.

```
medias <- as.vector(aggregate(training_data, FUN = mean))
desviaciones <- as.vector(aggregate(training_data, FUN = sd))
plot(medias, desviaciones)
abline(lm(desviaciones ~ medias), col = "blue", lty = 2)
```



Observando el gráfico resultante vemos que la desviación típica es relativamente constante por lo que podemos afirmar que se trata de un **modelo aditivo**

2. ¿Qué modelo de alisado exponencial te parece más adecuado? Razona tu respuesta. Aplica dicho método de alisado exponencial a la serie, **indica los parámetros de alisado óptimos e interprétales**.

El método de alisado más adecuado sería el método de Holt-Winters aditivo ya que la serie temporal presenta todas las componentes, es decir, tenemos componente Tendencia-Ciclo, componente Estacionalidad y componente Irregular.

```
HW_aditivo <- HoltWinters(training_data, seasonal = "additive")
HW_aditivo
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
```

```

## 
## Call:
## HoltWinters(x = training_data, seasonal = "additive")
##
## Smoothing parameters:
##   alpha: 0.566104
##   beta : 0.01523691
##   gamma: 0.1791379
##
## Coefficients:
##              [,1]
## a     31106.56187
## b      67.12385
## s1    5359.55824
## s2    2531.53157
## s3    1198.39197
## s4   -2443.35731
## s5   -1756.53754
## s6   -2007.97730
## s7    327.46892
## s8   -4773.94180
## s9   -3179.45024
## s10  -1765.49801
## s11  1039.91318
## s12  2904.82559

```

Los parámetros de alisado óptimos que nos marca el modelo mediante ajuste por mínimos cuadrados son:

$$\alpha = 0.566104, \quad \beta = 0.01523691, \quad \gamma = 0.1791379$$

Podemos observar que α toma un valor más cercano a 1 que ha cero por lo que tiene algo de importancia para la observación anterior (X_t/S_{t-L}) y toma menos peso en la suma del nivel de la serie y la pendiente de la tendencia. En cuanto a β podemos observar que es casi cero, esto quiere decir que toman mayor peso los valores anteriores de la pendiente de la tendencia (b_t) y como es un proceso iterativo

3. Determina las predicciones mensuales del consumo de gas para 2014, usando el **método de alisado exponencial** seleccionado. ¿Cuántos gigavatios hora estaba previsto consumir en junio de 2014?

```
predict(HW_aditivo, n.ahead = 12)
```

```

##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2014 36533.24 33772.34 32506.33 28931.70 29685.64 29501.33 31903.90 26869.61
##           Sep      Oct      Nov      Dec
## 2014 28531.23 30012.30 32884.84 34816.87

```

Para el mes de junio de 2014 estaba previsto consumir ≈ 29501 gigavatios hora.

4. Determina los **valores finales** que estimó R para nivel, pendiente de la tendencia y factores estacionales (indica sólo el factor estacional de junio). Calcula usando dichos valores finales las predicciones mensuales del consumo de gas para 2014. ¿Se obtiene lo mismo que en el apartado anterior?

```

print("Valores finales de R")

## [1] "Valores finales de R"
cat("- Nivel: ", HW_aditivo$coefficients[1], "\n")
## - Nivel: 31106.56
cat("- Pendiente de la tendencia: ", HW_aditivo$coefficients[2], "\n")
## - Pendiente de la tendencia: 67.12385
cat("- Factor estacional (junio): ", HW_aditivo$coefficients["s6"], "\n")
## - Factor estacional (junio): -2007.977

```

```

print("Predicciones manuales")

## [1] "Predicciones manuales"

horizonte <- 1:12
valores_estacionales <- HW_aditivo$coefficients[3:14]
a <- HW_aditivo$coefficients[1]
b <- HW_aditivo$coefficients[2]
(a + b * horizonte) + valores_estacionales

##      s1      s2      s3      s4      s5      s6      s7      s8
## 36533.24 33772.34 32506.33 28931.70 29685.64 29501.33 31903.90 26869.61
##      s9      s10     s11     s12
## 28531.23 30012.30 32884.84 34816.87

```

Si se obtienen los mismos valores que en el apartado anterior.

5. Usando sólo los datos reales disponibles para 2014, calcula las medidas de error MAE y RMSE que se obtienen con el método de alisado.

```

prediccion_2014 <- predict(HW_aditivo, n.ahead=12)
residuos <- test_data - prediccion_2014
MAE <- mean(abs(residuos))
RMSE <- sqrt(mean(residuos^2))
MAE

```

```
## [1] 6486.147
```

```
RMSE
```

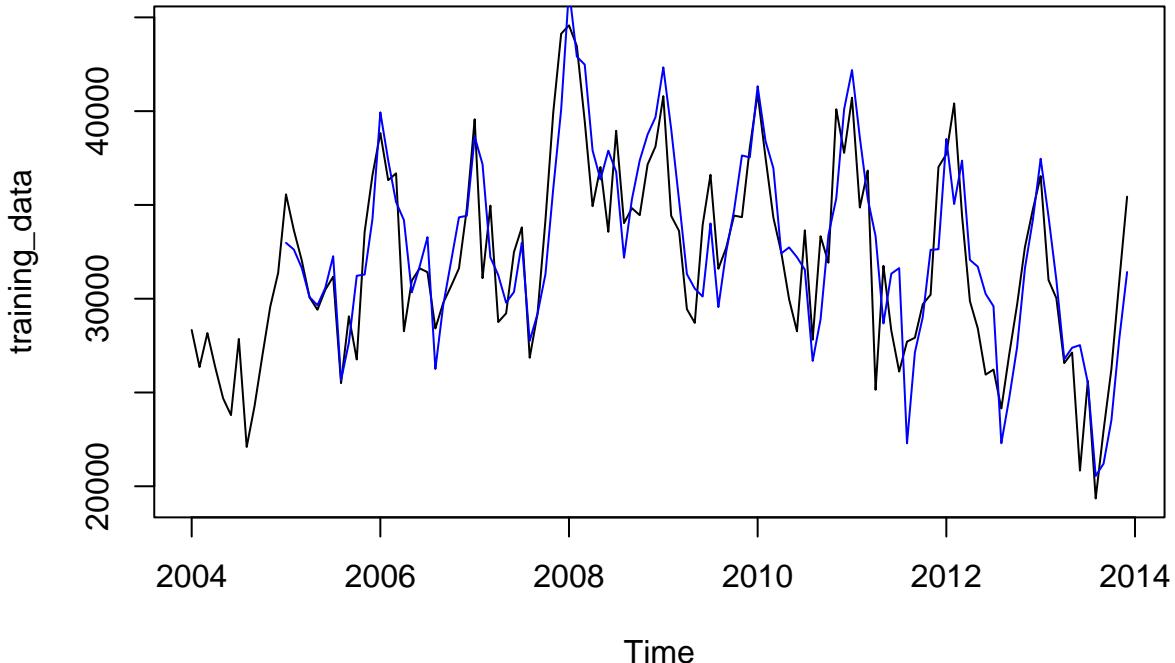
```
## [1] 6753.012
```

6. Para los resultados obtenidos con el método de alisado, representa en un mismo gráfico la serie original (en negro) y la serie ajustada con predicciones a un año (en azul). ¿A qué se debe el error tan grande en la zona test?

```

ts.plot(training_data, col = "black")
lines(HW_aditivo$fitted[, 1], col = "blue")

```



7. Atendiendo al método de alisado propuesto en el apartado (8), ¿qué **modelo(s)** ETS sería más adecuado para la serie en estudio? Realiza la selección del modelo ETS de forma automática con R y comprueba si coincide con tu respuesta.

Ya que en el problema anterior se ha determinado que la serie presenta un comportamiento aditivo, un modelo ETS(A,A,A) podría ser adecuado para capturar tanto la tendencia como la estacionalidad de manera aditiva.

```

library(forecast)

# Modelo determinado manualmente
modelo_ets <- ets(training_data, model = "AAA")

# Modelo obtenido automáticamente
modelo_ets_auto <- ets(datos_ts)

# Comprobación de que mi elección es correcta
modelo_ets_auto$method == modelo_ets$method

## [1] TRUE

```

8. Usando el modelo automático de R, calcula los intervalos de predicción para el consumo de gas en 2014. Usa nivel de confianza del 99% e indica el intervalo obtenido para junio de 2014. ¿Podemos decir que el consumo de gas será inferior a los 20000 gigavatios hora?

```

forecast(modelo_ets, h = 12, level = 99)

##          Point Forecast    Lo 99    Hi 99
## Jan 2014      35985.26 29728.31 42242.21
## Feb 2014      32640.18 25711.67 39568.70
## Mar 2014      31511.24 23970.51 39051.97
## Apr 2014      26401.81 18294.76 34508.86
## May 2014      27102.69 18466.20 35739.18
## Jun 2014      26279.72 17144.26 35415.19
## Jul 2014      28606.12 18997.40 38214.84
## Aug 2014      24368.93 14309.06 34428.80
## Sep 2014      26564.45 16072.68 37056.22
## Oct 2014      27732.13 16825.43 38638.83
## Nov 2014      31112.12 19805.60 42418.64
## Dec 2014      33858.72 22165.94 45551.49

```

El intervalo de junio de 2014 es de $x \in (17144.26, 35415.19)$, por lo que no podemos asegurar con certeza que el consumo vaya a ser menor de 20000 gigavatios hora.

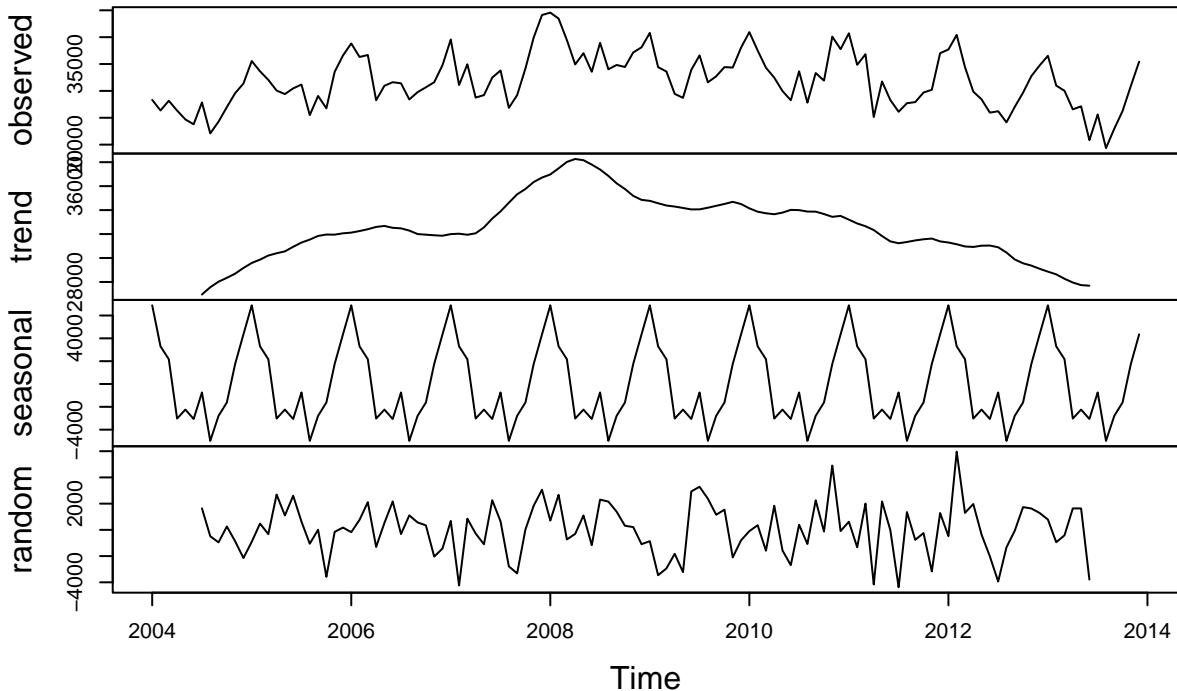
9. Extrae, usando **Análisis Clásico**, las componentes de la **serie de entrenamiento** (Tendencia-Ciclo, Estacionalidad e Irregular). Indica el valor de cada componente para junio de 2013 e **interpreta** el valor de la componente estacional en dicho mes.

```

library(zoo)
descomposicion_clasica <- decompose(training_data, type = "additive")
plot(descomposicion_clasica)

```

Decomposition of additive time series



10. Proporciona un modelo determinista que nos permita realizar predicciones, **ajustado un modelo adecuado para representar la tendencia en función del tiempo**. Expresa el modelo determinista de la forma:

$$Consumo(t) = 32859.91 - 5.425 \cdot t$$

```
tiempo <- 1:length(training_data)
tendencia.lm <- lm(descomposicion_clasica$trend ~ tiempo)
tendencia.lm
```

```
##
## Call:
## lm(formula = descomposicion_clasica$trend ~ tiempo)
##
## Coefficients:
## (Intercept)      tiempo
##   32859.910     -5.425
```

11. Determina las **predicciones mensuales** del consumo de gas para 2014. ¿Cuántos gigavatios hora estaba previsto consumir en junio de 2014?

```
tiempo_pred <- 1:(length(training_data) + 12)
estacional_pred <- c(descomposicion_clasica$seasonal,
                      descomposicion_clasica$seasonal[1:12])

tendencia_pred <- predict(tendencia.lm, data.frame(tiempo = tiempo_pred))

datos_pred <- tendencia_pred + estacional_pred

cat("Consumo predicho para junio de 2014:", datos_pred[length(training_data) + 6 - 1],
    "gigavatios hora")
```

Consumo predicho para junio de 2014: 29939.48 gigavatios hora

12. Determina las predicciones mensuales del consumo de gas para 2014, pero usando el **método de descomposición STL** (selecciona un ancho de banda para estimar la componente estacional de 3 años, dejando el ancho de banda para la tendencia de forma automática). Indica el valor de la predicción para junio de 2014.

```
datos_STL <- stl(training_data, s.window = 37) # 3 años * 12 meses = 36 -> usamos 37

# Extraer componentes
```

```

estacional_STL <- datos_STL$time.series[, 1]
tendencia_STL <- datos_STL$time.series[, 2]
irregular_STL <- datos_STL$time.series[, 3]

# Modelo determinista STL
tiempo_STL <- 1:length(training_data)
tendencia_STL.lm <- lm(tendencia_STL ~ tiempo_STL)

# Predicciones con STL
tiempo_pred_STL <- 1:(length(training_data) + 12)
estacional_STL_pred <- c(estacional_STL, estacional_STL[1:12])
tendencia_STL_pred <- predict(tendencia_STL.lm, data.frame(tiempo_STL = tiempo_pred_STL))

datos_STL_pred <- tendencia_STL_pred + estacional_STL_pred
cat("Consumo predicho para junio de 2014 (STL):", datos_STL_pred[length(training_data) + 6 - 1],
    "gigavatios hora")

```

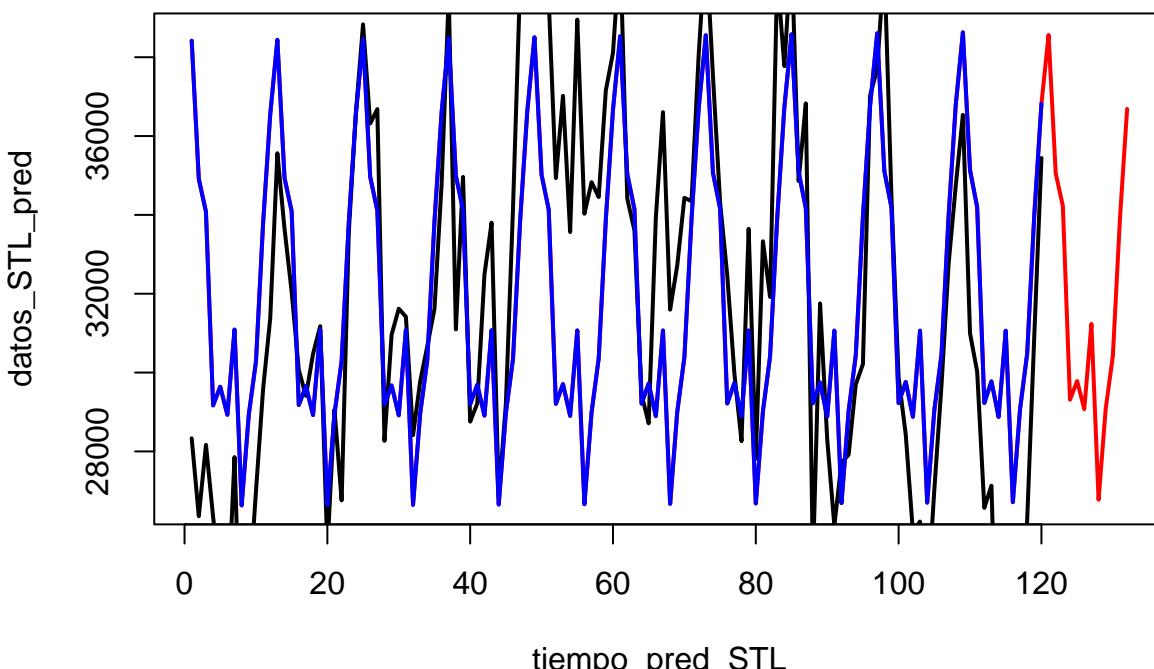
Consumo predicho para junio de 2014 (STL): 29785.27 gigavatios hora

13. **Representa en un mismo gráfico** la serie completa observada (en negro, incluye tanto tramo de entrenamiento como test), la serie ajustada (en azul) y la serie predicha para 2014 (en rojo), que resultan de la **descomposición STL**.

```

plot(tiempo_pred_STL, datos_STL_pred, type = "l", col = "red", lwd = 2)
lines(tiempo, training_data, type = "l", col = "black", lwd = 2)
lines(tiempo, datos_STL_pred[1:length(training_data)], type = "l", col = "blue", lwd = 2)

```



14. Usando sólo los datos reales disponibles para 2014, calcula las medidas de error MAE y RMSE que se obtienen con Análisis Clásico y con STL. Compáralas y comenta los resultados.

```

# Residuos análisis clásico
residuos_clasico <- datos_ts - datos_pred[1:length(datos_ts)]
MAE_clasico <- mean(abs(residuos_clasico))
RMSE_clasico <- sqrt(mean(residuos_clasico^2))
print("Análisis Clásico")

## [1] "Análisis Clásico"
cat("MAE:", MAE_clasico)

## MAE: 3336.338
cat("RMSE:", RMSE_clasico)

```

```

## RMSE: 4239.612

# Residuos STL
residuos_STL <- datos_ts - datos_STL_pred[1:length(datos_ts)]
MAE_STL <- mean(abs(residuos_STL))
RMSE_STL <- sqrt(mean(residuos_STL^2))
print("STL")

## [1] "STL"
cat("MAE:", MAE_STL)

## MAE: 3278.488
cat("RMSE:", RMSE_STL)

## RMSE: 4163.037

```

La diferencia entre los métodos puede observarse en las medidas de error MAE y RMSE. Debido a que numéricamente hablando obtenemos resultados muy similares, podemos concluir que ambos métodos son adecuados para este conjunto de datos, aunque la descomposición STL puede ofrecer una mayor flexibilidad en la modelización de la estacionalidad.