Procesos Estocásticos y Series Temporales

Entregable Práctica 4 y 5 (Modelo A)

Francisco Javier Mercader Martínez

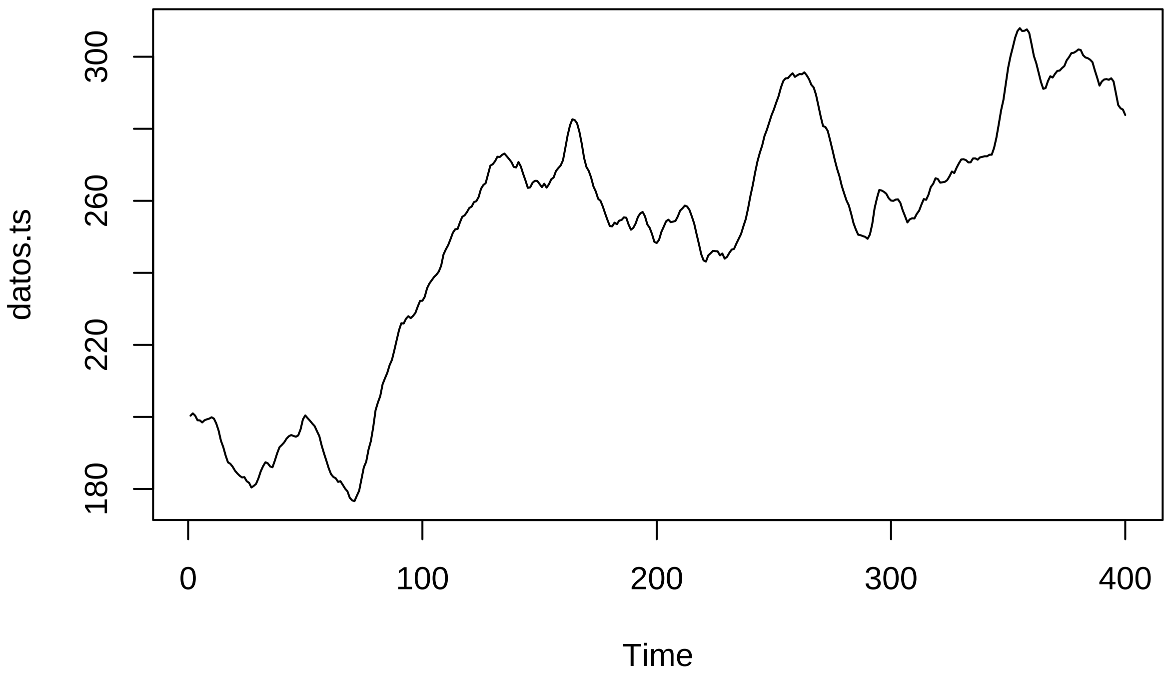
Problema 1

En el fichero **ARIMA_modelo_A.txt**, se ecnuentran los datos horarios correspondientes a la concentración de un gas en una zona salvaje. Responder a los siguientes apartados:

```
library(forecast)
library(tseries)
library(tidyverse)
```

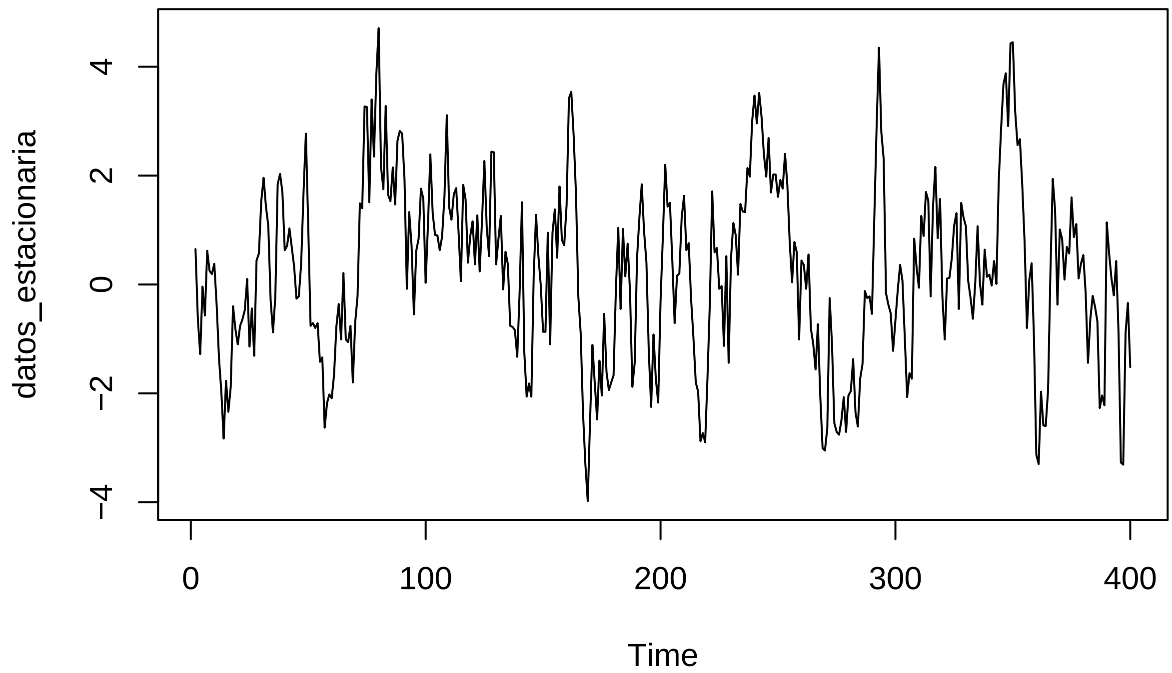
1) Obtener, de forma manual y explicando cada paso, un modelo ARIMA como generador de la serie en estudio (no hay que realizar predicciones, pero sí incluir la validación del modelo).

```
datos <- read.csv2("ARIMA_modelo_A.txt", header = FALSE, dec = ".")
datos.ts <- ts(datos$V1, frequency = 1)
ts.plot(datos.ts)
```

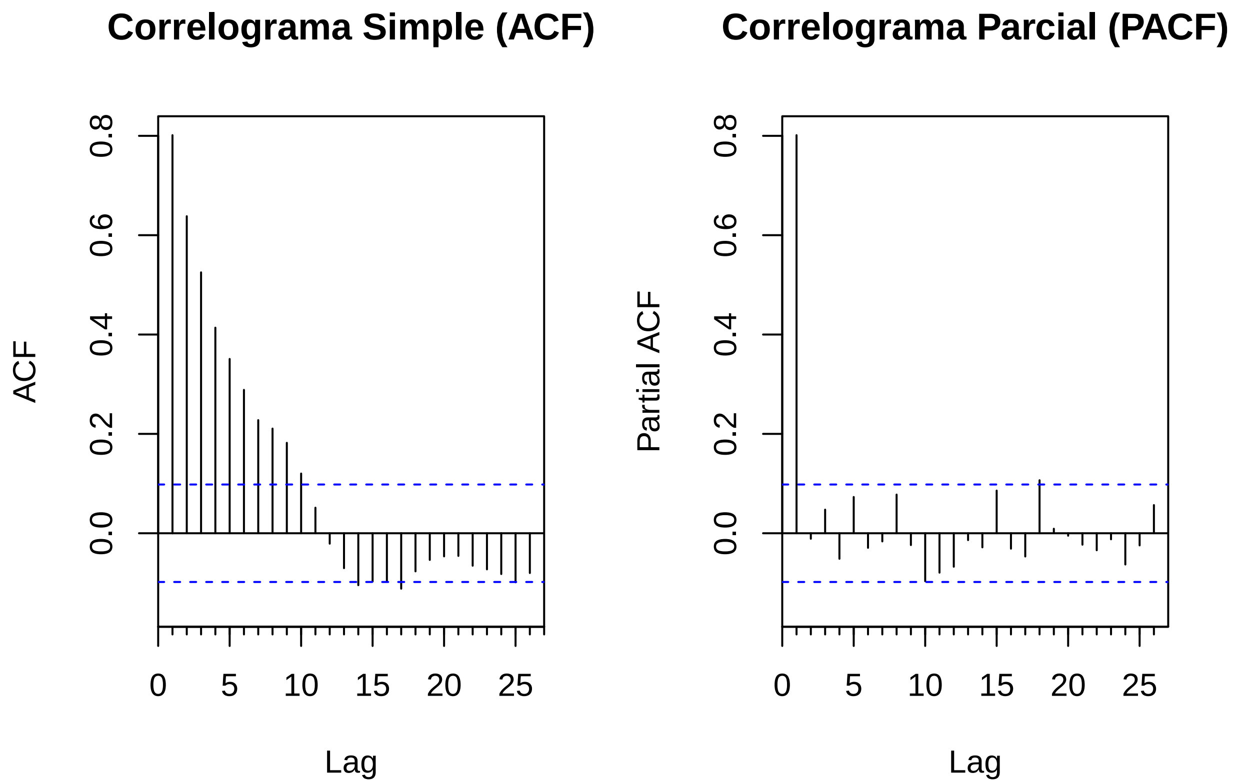


- La serie presenta estacionalidad (hay un patrón periódico ascendente)

```
datos_estacionaria <- diff(datos.ts, differences = 1)
ts.plot(datos_estacionaria)
```




```r
par(mfrow = c(1, 2))
Acf(datos_estacionaria, main = "Correlograma Simple (ACF)")
Pacf(datos_estacionaria, main = "Correlograma Parcial (PACF)")
```



En el correlograma simple observamos un decaimiento muy lento, tenemos correlaciones hasta retardo 10 más o menos, por otro lado, tenemos que en el correlograma parcial observamos una única correlación significativa.

Al observar un decaimiento tan lento en el correlograma simple, no vamos a poder extraer mucha información modelizando la parte de medias móviles pero al observar el correlograma es posible poder proponer un modelo $AR(1)$

Por lo tanto, vamos a proponer este modelo inicial (con estos datos diferenciados) $ARMA(p = 1)$, y en función de la interpretación de los p-valores y de la validación de los residuos plantearemos otros modelos o nos quedaremos con este.

```r
pvalores <- function(modelo) {
  # Extract coefficients
  coefs <- coef(modelo)
  # Extract standard errores
```

```r
  se <- sqrt(diag(vcov(modelo)))
  # Calculate the t-values
  t_values <- coefs / se
  # Calculate the p-values
  p_values <- 2 * (1 - pnorm(abs(t_values)))
  # Create a data frame with coefficients, standard errors, t-values, and p-values
  results_pvalues <- data.frame(
    Coefficients = coefs,
    Std_Errors = se,
    T_values = t_values,
    P_values = p_values
  )
  print(results_pvalues)
}
```

```r
# Primer modelo propuesto
arima.model100 <- Arima(datos_estacionaria, c(1, 0, 0))
arima.model100
```

```
## Series: datos_estacionaria
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.8018  0.1964
## s.e.  0.0297  0.2427
##
## sigma^2 = 0.9467:  log likelihood = -554.74
## AIC=1115.49   AICc=1115.55   BIC=1127.45
```

```r
pvalores(arima.model100)
```

```
##           Coefficients Std_Errors   T_values P_values
## ar1          0.8017651 0.02971603 26.9808935 0.000000
## intercept    0.1964439 0.24266762  0.8095184 0.418217
```

Observamos que el `intercept` del modelo puede ser obviado y el modelo reducido ya que tiene el p-valor mayor a 0.05.

Propongamos un modelo sin tener en cuenta el `intercept`:

```r
arima.model_without_mean <- Arima(datos_estacionaria, c(1, 0, 0), include.mean = FALSE)
arima.model_without_mean
```

```
## Series: datos_estacionaria
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##          ar1
##       0.8048
## s.e.  0.0295
##
## sigma^2 = 0.9458:  log likelihood = -555.06
## AIC=1114.13   AICc=1114.16   BIC=1122.11
```

```r
pvalores(arima.model_without_mean)
```

```
##     Coefficients Std_Errors T_values P_values
## ar1    0.8048375  0.0294873 27.29438        0
```

Propongamos ahora otro modelo modelizando también la parte de medias móviles con un orden $q = 2$ y observemos si podemos mejorar nuestro modelo:

```r
arima.model102 <- Arima(datos_estacionaria, c(1, 0, 2))
arima.model102
```

```
## Series: datos_estacionaria
```

```
## ARIMA(1,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ma1      ma2    mean
##       0.8251  -0.0046  -0.0747  0.1934
## s.e.  0.0438   0.0666   0.0672  0.2525
##
## sigma^2 = 0.9483:  log likelihood = -554.08
## AIC=1118.16   AICc=1118.31   BIC=1138.1
```

```
pvalores(arima.model102)
```

```
##           Coefficients Std_Errors    T_values  P_values
## ar1        0.825061070 0.04381596 18.83015006 0.0000000
## ma1       -0.004570239 0.06662885 -0.06859249 0.9453140
## ma2       -0.074675441 0.06722251 -1.11086964 0.2666245
## intercept  0.193433958 0.25249660  0.76608539 0.4436255
```

Observando los p-valores de los modelos planteados, se puede obviar el uso de `intercept` y modelizar la parte de medias móviles ya que su p-valor es mayor que 0.05 y eso indica que el modelo es reducible.

Modelo final escogido:

```
arima.modelo_final <- Arima(datos_estacionaria, c(1, 0, 0), include.mean = FALSE)
```

Pasemos a validar el modelo

El modelo ARIMA ajustado será válido si se cumplen las hipótesis:

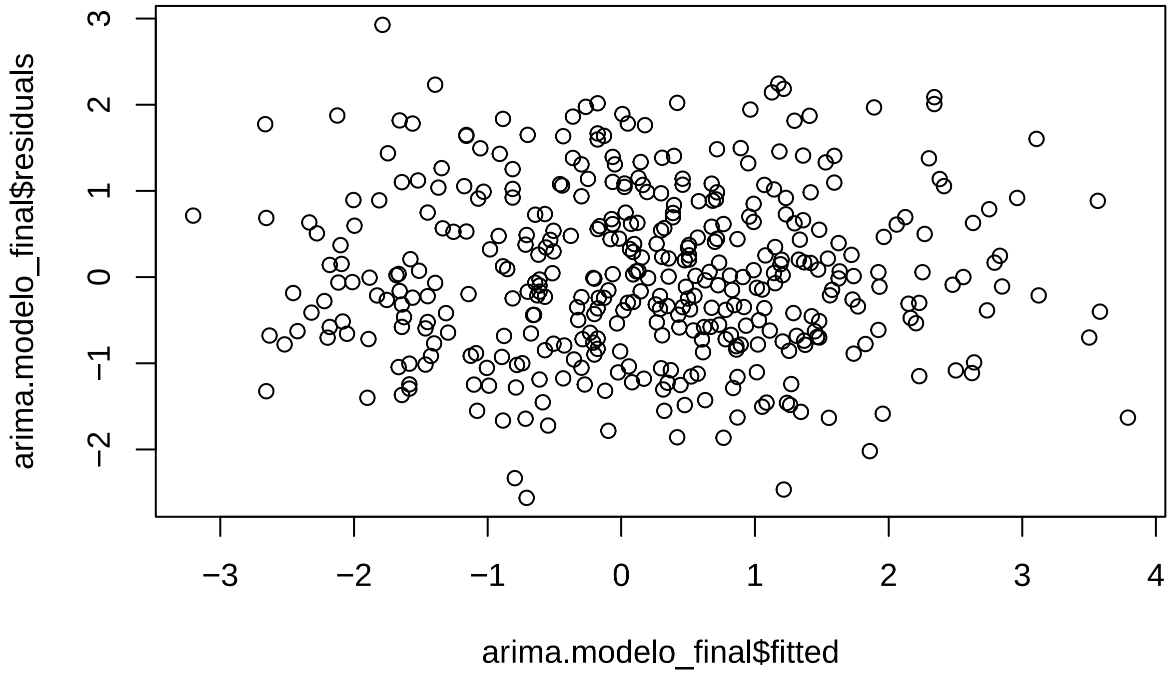
1. Normalidad de los residuos.
2. Homocedasticidad: residuos con varianza constante.
3. Independencia de los residuos.

```
shapiro.test(arima.modelo_final$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  arima.modelo_final$residuals
## W = 0.99262, p-value = 0.04649
```

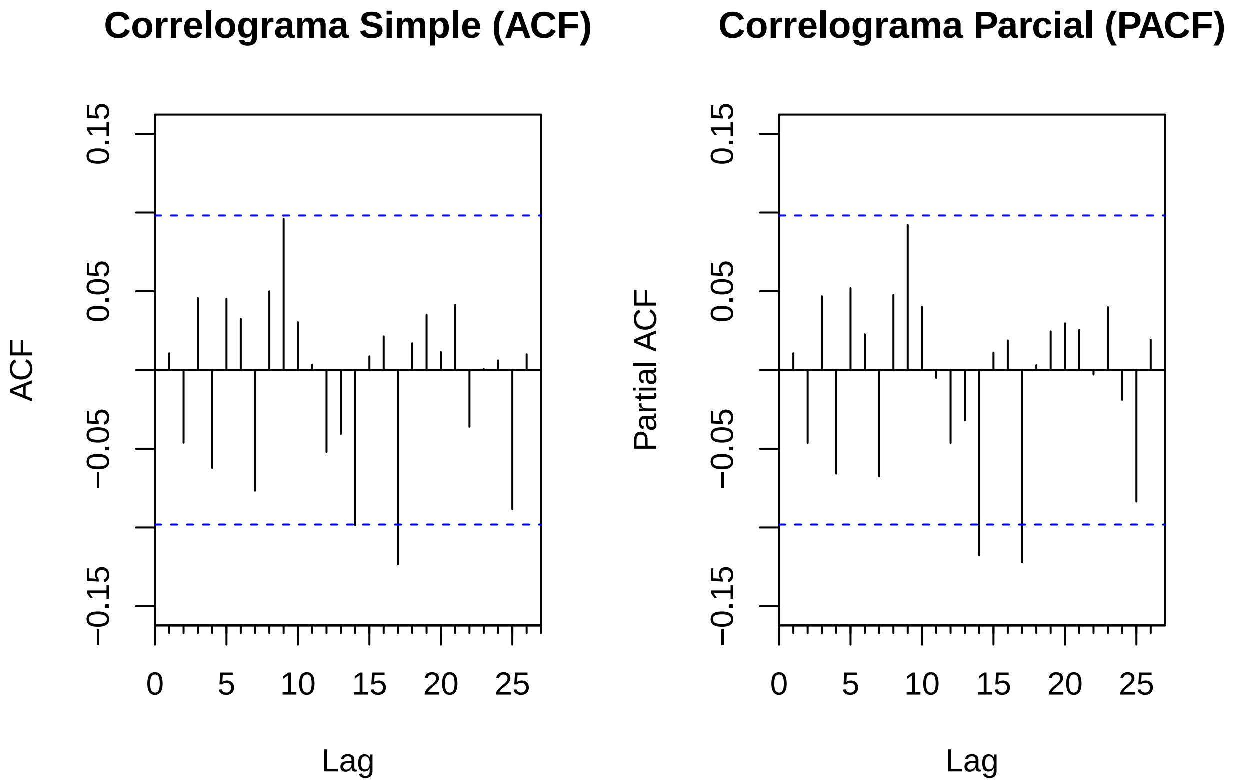
No podemos garantizar que los residuos siguen una normal.

```
plot(arima.modelo_final$fitted, arima.modelo_final$residuals)
```



Podemos observar que se cumple la hipótesis de homocedasticidad, es decir, los residuos presentan una varianza constante ya que muestran una distribución homogénea.

```r
par(mfrow = c(1, 2))
Acf(arima.modelo_final$residuals, main = "Correlograma Simple (ACF)")
Pacf(arima.modelo_final$residuals, main = "Correlograma Parcial (PACF)")
```



Observando la ACF y PACF de los residuos, hay una correlación significativa, podemos suponer una muy pequeña dependencia en los residuos.

Los residuos del modelo no pasan las hipótesis de normalidad ni de independencia por lo que el modelo no puede ser valido.

2) Obtener el modelo ARIMA de forma automática con una función de `R`. Indicar cuál sería la expresión del modelo ARIMA resultante y determinar los intervalos de predicción al 95% para las siguientes 2 horas (no hay que representar las predicciones gráficamente).

```r
auto_model <- auto.arima(datos.ts)
summary(auto_model)
```

```
## Series: datos.ts
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.8048
## s.e.  0.0295
##
## sigma^2 = 0.9459:  log likelihood = -555.06
## AIC=1114.13   AICc=1114.16   BIC=1122.11
##
## Training set error measures:
##                      ME      RMSE       MAE        MPE      MAPE      MASE
## Training set 0.03751645 0.9701494 0.7837385 0.01828656 0.3172549 0.5929419
##                    ACF1
## Training set 0.01075902
```

La obtención automática del modelo nos indica que el modelo correcto para modelizar nuestra serie en estudio sería un $ARIMA(1, 1, 0)$, el mismo que planteamos nosotros manualmente pero teniendo en cuenta la deriva, se planteó un $AR(1)$ con los datos diferenciados, que a la hora de trabajar con la serie en estudio es un $ARIMA(1, 1, 0)$.

Expresión del modelo resultante:

$$(1 - 0.8048377)X_t = \varepsilon_t$$

Predicciones para las 2 siguientes horas.

```
pred <- forecast(auto_model, h = 2, level = 0.95)
pred$mean
```

```
## Time Series:
## Start = 401
## End = 402
## Frequency = 1
## [1] 282.5966 281.6120
```

Problema 2

En el fichero **Consumo_Electrico_modelo_A.csv**, se encuentran los datos horarios del consumo eléctrico (en kW) de los habitantes de una población (datos simulados), entre el 15/05/2009 y el 20/06/2009. También se muestran los registros de Temperatura ambiente (en ºKelvin), humedad relativa y la variable indicadora de festividad "Festivo".

El objetivo es modelizar la serie de consumo eléctrico horario usando como predictores Temperatura, Humedad, Festivo, así como predictores que representen la tendencia y estacionalidad de la serie (si fueran necesarios). Responder a las siguientes cuestiones:

```
datos <- read.csv2("Consumo_Electrico_modelo_A.csv", header = TRUE, sep = ";")
sum(is.na(datos))
```

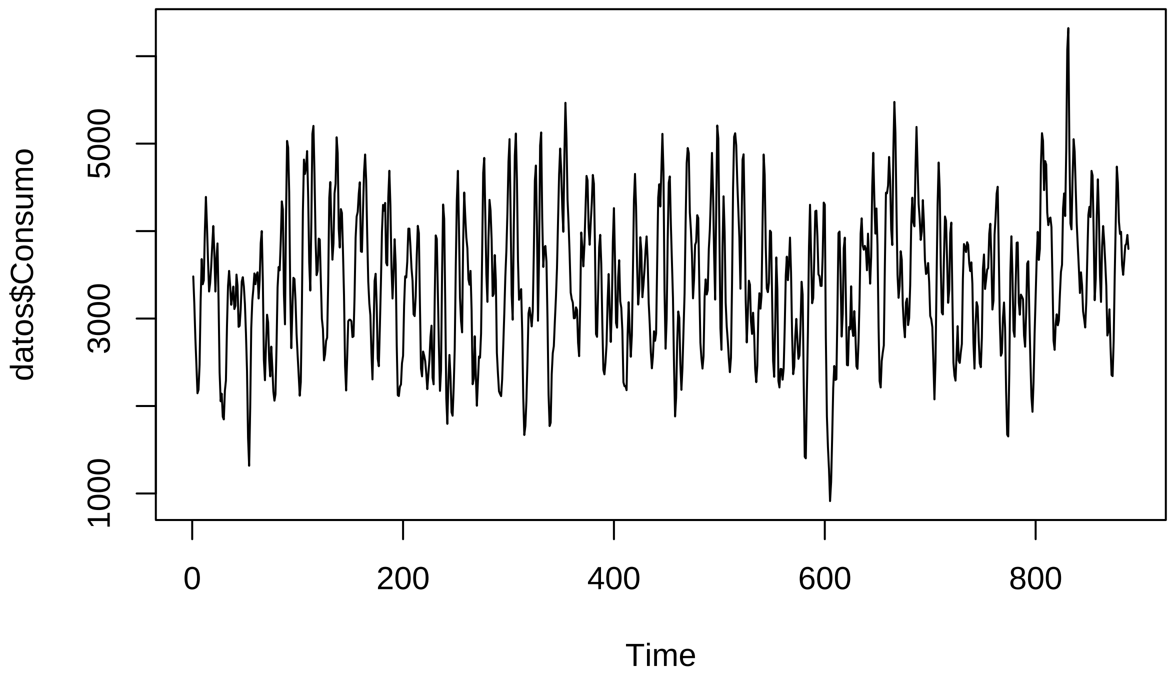
```
## [1] 0
```

```
summary(datos)
```

```
##     Fecha              Consumo      Temperatura       Humedad
##  Length:888         Min.   : 913   Min.   :280.0   Min.   :0.3178
##  Class :character   1st Qu.:2822   1st Qu.:287.5   1st Qu.:0.5696
##  Mode  :character   Median :3392   Median :290.8   Median :0.7564
##                     Mean   :3411   Mean   :291.2   Mean   :0.7156
##                     3rd Qu.:3990   3rd Qu.:295.0   3rd Qu.:0.8408
##                     Max.   :6321   Max.   :303.6   Max.   :0.9906
##     Festivo
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.02703
##  3rd Qu.:0.00000
##  Max.   :1.00000
```

1) Representa la serie de Consumo en un gráfico temporal. ¿Cómo es la tendencia? ¿La serie presenta estacionalidad? ¿Cuál sería el periodo?

```
ts.plot(datos$Consumo)
```



No presenta una tendencia ya que la media se mantiene constante a lo largo de la serie observada pero si se observa una componente estacional, al ser datos horarios, la serie presenta estacionalidad diaria y semanal, es decir, el periodo de la diaria sería $L = 24$ y el periodo de la semanal sería $L = 168$

2) Divide los datos en entrenamiento y test, dejando para test sólo los días 19 y 20 de junio.

```
inicio_test <- which(datos$Fecha == "19/06/2009")
indices_train <- 1:(inicio_test - 1)
indices_test <- inicio_test:nrow(datos)
datos_train <- datos[indices_train, ]
datos_test <- datos[indices_test, ]

consumo.ts <- ts(datos$Consumo, start = c(1, 1), frequency = 24)
consumo.ts_train <- window(consumo.ts, end = c(35, 24))
consumo.ts_test <- window(consumo.ts, start = c(36, 1))
```

3) Con los datos de entrenamiento, ajusta un modelo regresión lineal múltiple (RLM) para la serie de Consumo **usando como predictores Temperatura, Humedad, Festivo y variables dummy para la estacionalidad diaria (no usar tendencia)**. Indicar el coeficiente de regresión obtenido para la variable "Festivo" e interpretarlo. ¿Se está teniendo en cuenta la estacionalidad semanal?

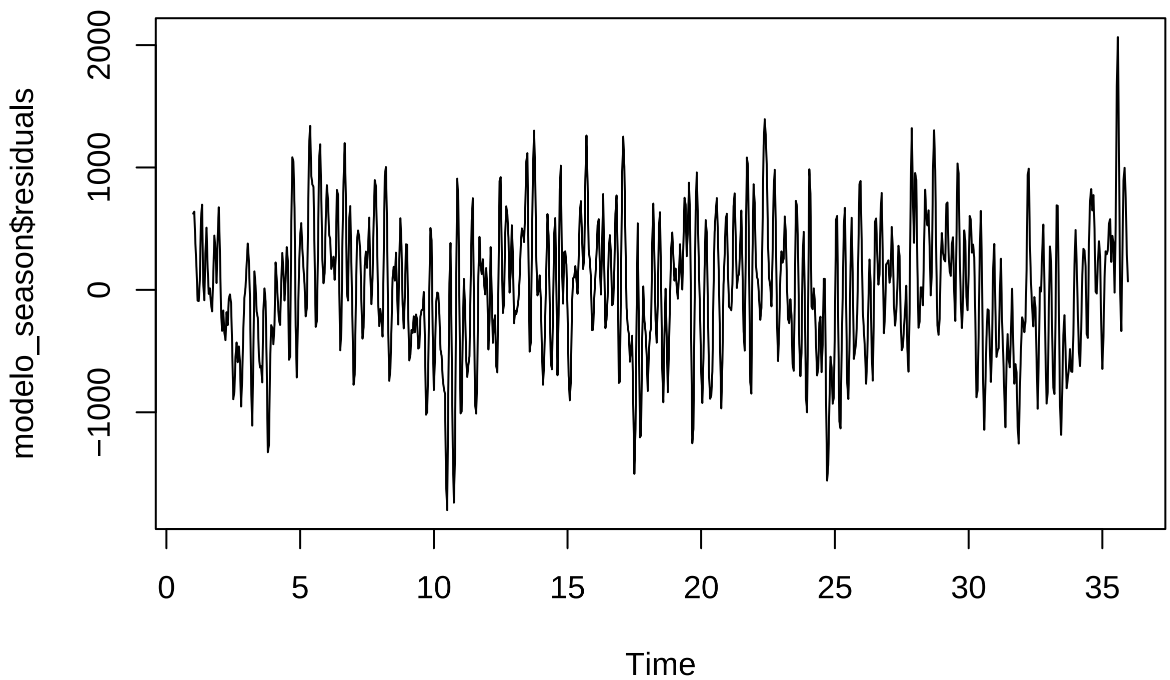
```
# Eliminamos la variable Fecha y Consumo del dataframe para poder construir el modelo mediante
↪ las varaibles restantes.
p1 <- datos |>
  select(-Fecha, -Consumo)

p1.train <- p1[indices_train, ]
p1.test <- p1[indices_test, ]

p1.train_matrix <- as.matrix(p1.train)

modelo_season <- tslm(consumo.ts_train ~ season + p1.train_matrix)
modelo_season
```

```
##
## Call:
## tslm(formula = consumo.ts_train ~ season + p1.train_matrix)
##
## Coefficients:
##                (Intercept)                     season2                     season3
##                   -9525.64                     -306.16                     -446.54
##                    season4                     season5                     season6
##                    -519.06                     -578.19                     -527.83
```

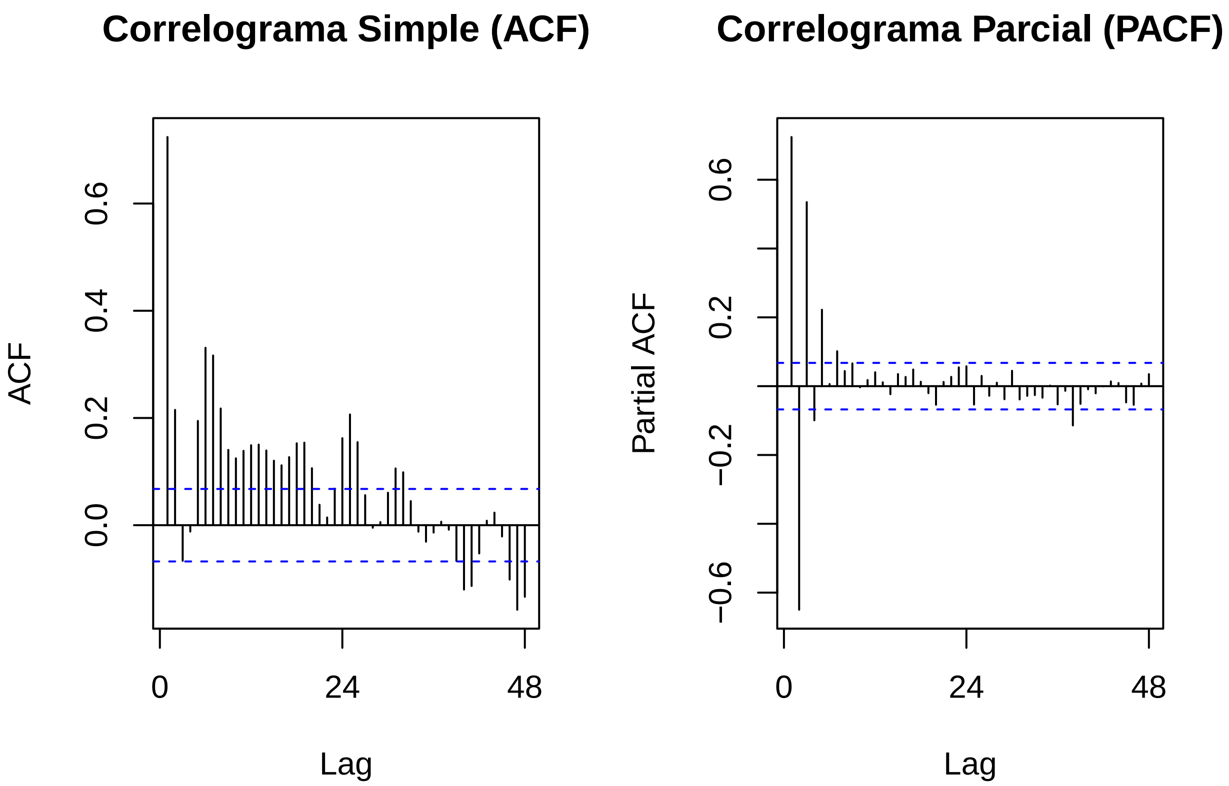
```
##                     season7                   season8                   season9
##                     -405.87                   -171.00                    234.11
##                    season10                  season11                  season12
##                      548.13                    662.03                    724.30
##                    season13                  season14                  season15
##                      829.11                    765.39                    522.93
##                    season16                  season17                  season18
##                       89.58                    267.20                    486.50
##                    season19                  season20                  season21
##                      440.82                    316.46                    158.87
##                    season22                  season23                  season24
##                       86.13                    219.17                    236.90
## p1.train_matrixTemperatura      p1.train_matrixHumedad      p1.train_matrixFestivo
##                       44.61                   -277.78                   -728.87
```

4) Analiza si los residuos que resultan del apartado (3) son independientes. Justifica si te parece adecuado el modelo planteado en el apartado (3) y qué plantearías en este escenario.

```
ts.plot(modelo_season$residuals)
```



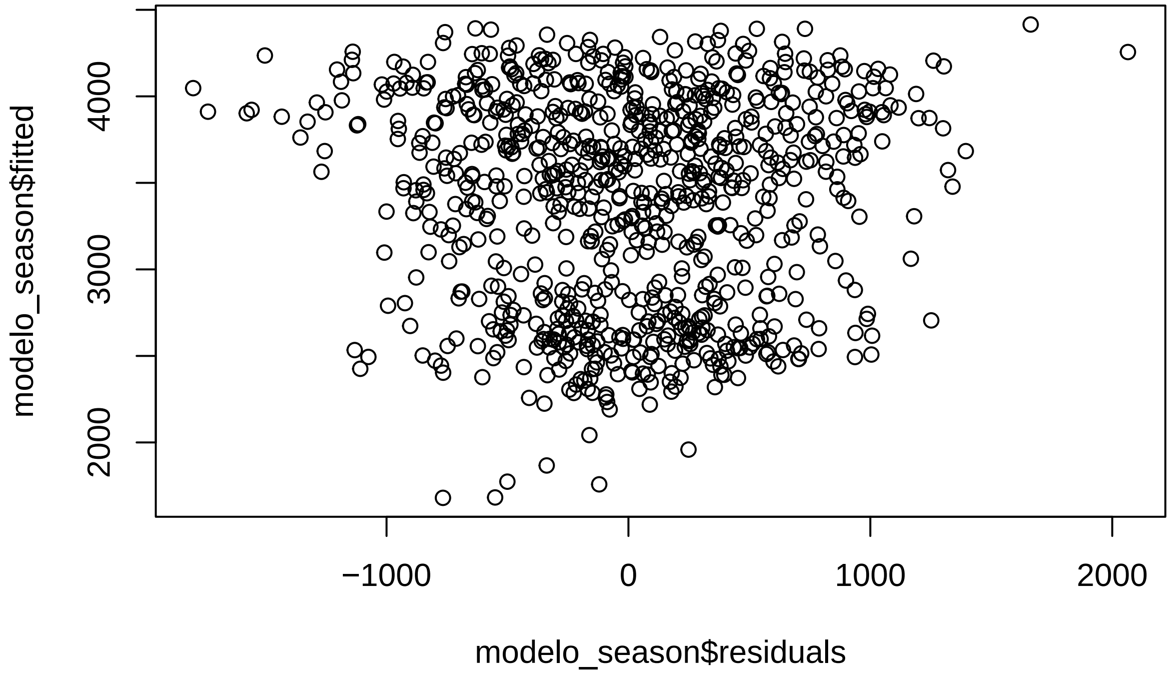

```
par(mfrow = c(1, 2))
Acf(modelo_season$residuals, main = "Correlograma Simple (ACF)")
Pacf(modelo_season$residuals, main = "Correlograma Parcial (PACF)")
```




```
shapiro.test(modelo_season$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelo_season$residuals
## W = 0.99898, p-value = 0.9291
```

```
plot(modelo_season$residuals, modelo_season$fitted)
```



Se puede observar que los residuos superan la hipótesis de normalidad y la de homocedasticidad por poco pero se puede observar que no tienen un comportamiento como el que tendría un ruido blanco gaussiano y , además, si interpretamos los correlogramas normal y parcial de estos residuos, podemos observar cierta correlación entre ellos, es decir, que no son Independientes. Esto se puede estar dando porque no se está captando toda la información de la serie objetivo y esta información se ha ido a los residuos.

Para corregir esto podemos llegar a plantear realizar una Regresión Dinámica para así modelizar los residuos con un modelo ARIMA y poder explicar la información restante no explicada por los regresores previamente empleados.

5) Con el modelo resultante del apartado (3), realiza las predicciones de la zona test (19 y 20 de junio). ¿Cuál es la predicción puntual de consumo eléctrico para el 19 junio a las 6h? ¿Podemos decir que el consumo en esa hora será superior a 2000 kW con un 90%? ¿Se trata de predicciones ex-ante o ex-post?

```
pred_season <- forecast(modelo_season, newdata = p1.test, level = 0.9)
pred_season
```

```
##          Point Forecast    Lo 90    Hi 90
## 36.00000       3462.700 2517.507 4407.893
## 36.04167       3093.670 2148.993 4038.348
## 36.08333       2909.398 1964.946 3853.849
## 36.12500       2813.400 1868.821 3757.979
## 36.16667       2692.516 1748.624 3636.407
## 36.20833       2692.360 1748.875 3635.844
## 36.25000       2773.677 1830.438 3716.916
## 36.29167       2978.049 2034.968 3921.130
## 36.33333       3365.289 2422.439 4308.139
## 36.37500       3728.779 2786.024 4671.534
## 36.41667       3892.187 2948.833 4835.540
## 36.45833       3993.279 3049.429 4937.129
## 36.50000       4147.651 3203.586 5091.717
## 36.54167       4162.397 3218.699 5106.095
## 36.58333       4003.435 3060.158 4946.712
## 36.62500       3642.505 2699.617 4585.394
## 36.66667       3854.463 2911.632 4797.294
## 36.70833       4026.261 3082.229 4970.292
## 36.75000       3919.232 2974.577 4863.887
## 36.79167       3771.904 2828.199 4715.609
## 36.83333       3575.576 2632.843 4518.310
## 36.87500       3408.464 2466.015 4350.914
## 36.91667       3426.275 2483.695 4368.855
## 36.95833       3371.332 2428.887 4313.776
## 37.00000       3093.103 2150.788 4035.418
## 37.04167       2756.851 1814.558 3699.144
## 37.08333       2591.439 1649.178 3533.700
## 37.12500       2468.216 1525.869 3410.564
## 37.16667       2419.742 1477.491 3361.992
## 37.20833       2452.221 1509.985 3394.457
## 37.25000       2554.401 1612.168 3496.634
## 37.29167       2773.171 1830.939 3715.403
## 37.33333       3203.462 2261.260 4145.664
## 37.37500       3556.248 2613.932 4498.563
## 37.41667       3737.448 2794.853 4680.043
## 37.45833       3889.150 2946.527 4831.774
## 37.50000       4085.805 3143.317 5028.293
## 37.54167       4108.851 3166.457 5051.245
## 37.58333       3937.786 2995.405 4880.167
## 37.62500       3556.330 2613.936 4498.724
## 37.66667       3765.117 2822.657 4707.577
## 37.70833       3986.975 3044.440 4929.509
## 37.75000       3928.957 2986.372 4871.542
## 37.79167       3767.605 2824.812 4710.398
## 37.83333       3530.645 2587.523 4473.768
## 37.87500       3347.792 2404.362 4291.222
## 37.91667       3364.139 2420.787 4307.492
## 37.95833       3299.381 2355.963 4242.798
```

Predicción puntual cosumo del 19 de junio a las 6h: 2773.677 kW por lo que podemos asegurar que el consumo será superior a 2000 kW.

Estamos realizando predicciones ex-post porque no estamos prediciendo los valores de los regresores sino que utilizamos los valores reales de estos, una sola fuente de incertidumbre ya que solamente tenemos que buscar predecir el consumo para los 2 siguientes días, sin embargo, este tipo de predicción no es del todo realista ya que estamos suponiendo que se pueden predecir los regresores de manera perfecta.

6) Calcula el RMSE, que resulta en la zona de entrenamiento y el RMSE en la zona test.

```
RMSE_train <- sqrt(mean((consumo.ts_train - modelo_season$fitted.values)^2))
RMSE_test <- sqrt(mean((consumo.ts_test - pred_season$mean)^2))
cat("RMSE training:", RMSE_train, "\nRMSE test:", RMSE_test)
```

```
## RMSE training: 555.0068
## RMSE test: 464.5907
```

Ahora vamos a construir un modelo de regresión dinámica para la serie Consumo

7) Con los datos de entrenamiento, ajusta un modelo de **regresión dinámica** (de froma automática) para la serie de Consumo **usando como predictores la Temperatura, Festivo, Humedad y series de Fourier (con K=6) para la estacionalidad diaria.** ¿Qué modelo ARIMA se obtiene para los residuos del modelo RLM? ¿Contiene parte estacional? En caso afirmativo, explica a qué puede deberse.

```
p1.train_fourier <- data.frame(p1.train, fourier(consumo.ts_train, K = 6))
p1.train_fourier_matrix <- as.matrix(p1.train_fourier)

dyreg_fourier <- auto.arima(consumo.ts_train, xreg = p1.train_fourier_matrix)
summary(dyreg_fourier)
```

```
## Series: consumo.ts_train
## Regression with ARIMA(3,0,2)(0,0,1)[24] errors
##
## Coefficients:
##          ar1      ar2     ar3      ma1      ma2    sma1   intercept  Temperatura
##       1.7896  -1.3792  0.5515  -0.2666  -0.4586  0.1915  -11909.067      52.9305
## s.e.  0.0426   0.0607  0.0332   0.0466   0.0426  0.0334    3832.825      12.9995
##        Humedad     Festivo      S1.24      C1.24     S2.24     C2.24    S3.24
##       -96.0504   -777.0494  -372.9967  -332.0132  -94.5813  285.8446  59.2925
## s.e.  371.1596    103.1549   101.4423    40.5558   38.8405   39.6180  49.2371
##          C3.24    S4.24    C4.24     S5.24     C5.24    S6.24     C6.24
##       -45.9254  53.3607  60.8635  -44.6867  100.9265  -7.4196  -29.4346
## s.e.   50.0038  50.7580  50.6947   29.0536   28.9697  16.4552   16.4694
##
## sigma^2 = 63288:  log likelihood = -5826.02
## AIC=11698.05   AICc=11699.4   BIC=11806.91
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE         ACF1
## Training set -1.354253 248.2539 194.6751 -0.6319186 6.012474 0.3272411 -0.001024034
```

Se obtiene un modelo ARIMA(3,0,2)(0,0,1) para modelizar los residuos, sería un modelo SARIMA por lo que los residuos contienen información sobre la estacionalidad, esto se debe a que deberíamos de incrementar el valor de K de los términos de Fourier para poder captar y estimar toda la información de la estacionalidad.

8) Con el modelo resultante del apartado (7), analiza la independencia de los residuos y realiza las predicciones de la zona test (19 y 20 de junio). ¿Cuál es la predicción puntual de consumo eléctrico para el 19 de junio a las 6h?

```
p1.test_fourier <- data.frame(p1.test, fourier(consumo.ts_test, K = 6))
p1.test_fourier_matrix <- as.matrix(p1.test_fourier)

pred_reg_dinamica <- forecast(dyreg_fourier, xreg = p1.test_fourier_matrix)
pred_reg_dinamica
```

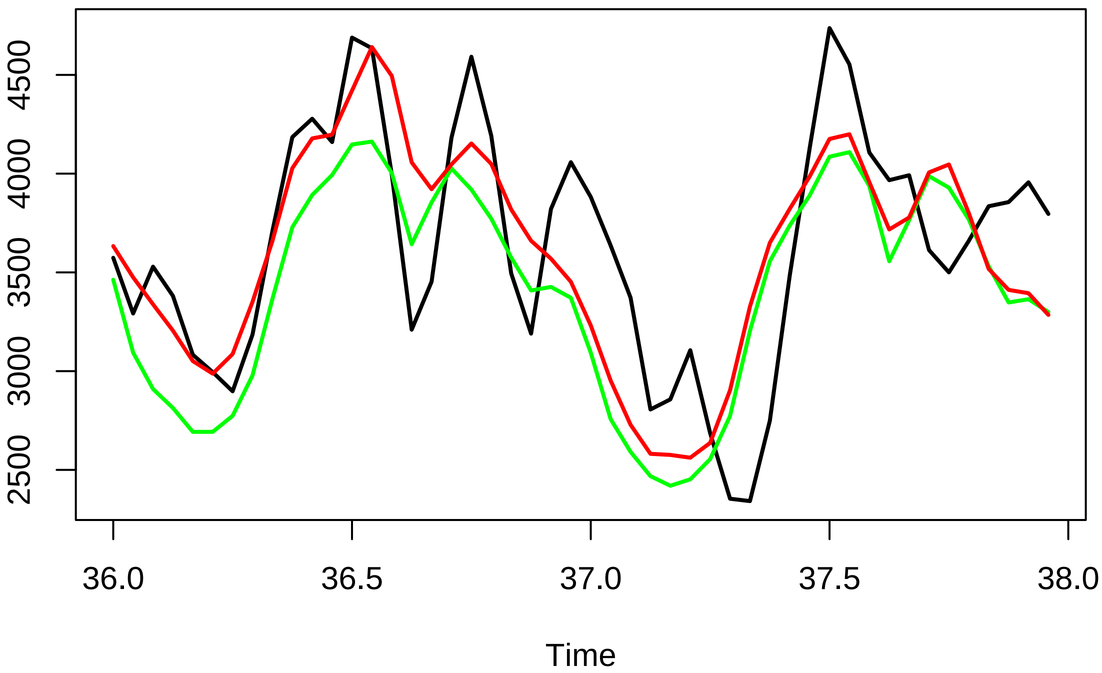
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 36.00000       3633.211 3310.811 3955.611 3140.143 4126.279
## 36.04167       3475.009 2887.615 4062.403 2576.667 4373.351
## 36.08333       3338.447 2685.027 3991.866 2339.128 4337.765
## 36.12500       3205.151 2551.605 3858.696 2205.639 4204.662
## 36.16667       3051.032 2389.734 3712.329 2039.665 4062.399
## 36.20833       2987.281 2324.742 3649.820 1974.015 4000.547
## 36.25000       3086.477 2419.843 3753.110 2066.949 4106.005
## 36.29167       3350.671 2671.044 4030.298 2311.271 4390.071
## 36.33333       3658.731 2969.842 4347.621 2605.166 4712.297
```

```
## 36.37500       4027.576 3336.099 4719.053 2970.053 5085.099
## 36.41667       4178.334 3486.419 4870.249 3120.142 5236.527
## 36.45833       4197.302 3504.980 4889.624 3138.487 5256.117
## 36.50000       4420.272 3726.710 5113.835 3359.560 5480.985
## 36.54167       4641.211 3945.466 5336.957 3577.160 5705.262
## 36.58333       4494.757 3796.885 5192.629 3427.453 5562.060
## 36.62500       4057.087 3357.807 4756.367 2987.630 5126.544
## 36.66667       3921.253 3221.115 4621.392 2850.484 4992.023
## 36.70833       4046.880 3346.061 4747.700 2975.070 5118.691
## 36.75000       4152.776 3451.229 4854.322 3079.853 5225.698
## 36.79167       4049.266 3346.913 4751.619 2975.109 5123.423
## 36.83333       3818.755 3115.615 4521.894 2743.396 4894.113
## 36.87500       3660.202 2956.394 4364.010 2583.821 4736.583
## 36.91667       3569.321 2864.972 4273.670 2492.112 4646.529
## 36.95833       3452.462 2747.653 4157.270 2374.551 4530.373
## 37.00000       3231.473 2521.424 3941.522 2145.547 4317.399
## 37.04167       2952.125 2232.374 3671.876 1851.360 4052.889
## 37.08333       2728.604 2004.667 3452.541 1621.438 3835.770
## 37.12500       2581.456 1857.115 3305.796 1473.673 3689.239
## 37.16667       2575.699 1851.358 3300.040 1467.915 3683.483
## 37.20833       2561.699 1837.269 3286.129 1453.780 3669.618
## 37.25000       2636.791 1911.645 3361.936 1527.776 3745.805
## 37.29167       2903.580 2177.185 3629.976 1792.654 4014.507
## 37.33333       3326.902 2599.511 4054.292 2214.453 4439.350
## 37.37500       3649.807 2921.912 4377.702 2536.587 4763.026
## 37.41667       3821.713 3093.556 4549.870 2708.093 4935.332
## 37.45833       3985.399 3257.006 4713.792 2871.418 5099.380
## 37.50000       4175.705 3447.003 4904.406 3061.252 5290.158
## 37.54167       4199.372 3470.303 4928.441 3084.357 5314.386
## 37.58333       3957.176 3227.764 4686.589 2841.636 5072.717
## 37.62500       3717.510 2987.827 4447.193 2601.556 4833.464
## 37.66667       3777.863 3047.972 4507.754 2661.591 4894.134
## 37.70833       4006.286 3276.217 4736.355 2889.742 5122.830
## 37.75000       4046.452 3316.213 4776.692 2929.647 5163.258
## 37.79167       3800.217 3069.810 4530.624 2683.156 4917.278
## 37.83333       3517.320 2786.757 4247.882 2400.020 4634.619
## 37.87500       3411.428 2680.729 4142.128 2293.920 4528.937
## 37.91667       3394.565 2663.748 4125.383 2276.876 4512.254
## 37.95833       3284.830 2553.908 4015.751 2166.982 4402.677
```

Predicción para el 19 de junio a las 6h: 3086.477 kW.

9) Representa conjuntamente, para la zona test, los datos reales (en negro), las predicciones obtenidas con el modelo RLM (en verde) y las predicciones obtenidas con el modelo de regresión dinámica (en rojo).

```r
ts.plot(consumo.ts_test, pred_season$mean, pred_reg_dinamica$mean,
  col = c("black", "green", "red"),
  lwd = 2
)
```



10) Si se pretende analizar la mejora del modelo de regresión dinámica frente al RLM, ¿te parece adecuado el planteamiento que se ha hecho comparando el modelo del apartado (3) con el del apartado (7)? Justifica tu respuesta argumentando qué harías.

Podemos ver que las predicciones con el modelo de regresión dinámica son mucho mejores porque consiguen modelizar los residuos y se corrigen así mismo por lo que consigue hacer mejores predicciones.