

TareaII Parte 1

Javier Aos

Parte 1: Series temporales

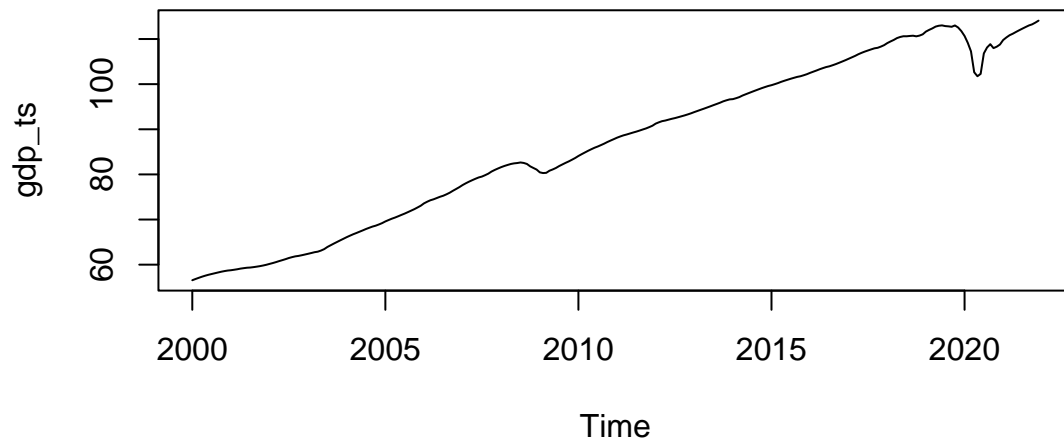
Presentación de la serie a analizar

En esta ocasión vamos a analizar una serie temporal del producto interior bruto (PIB) mundial obtenida de “U.S. Energy Information Administration Data”. Contamos con datos mensuales desde el año 2000 hasta el 2021 incluidos.

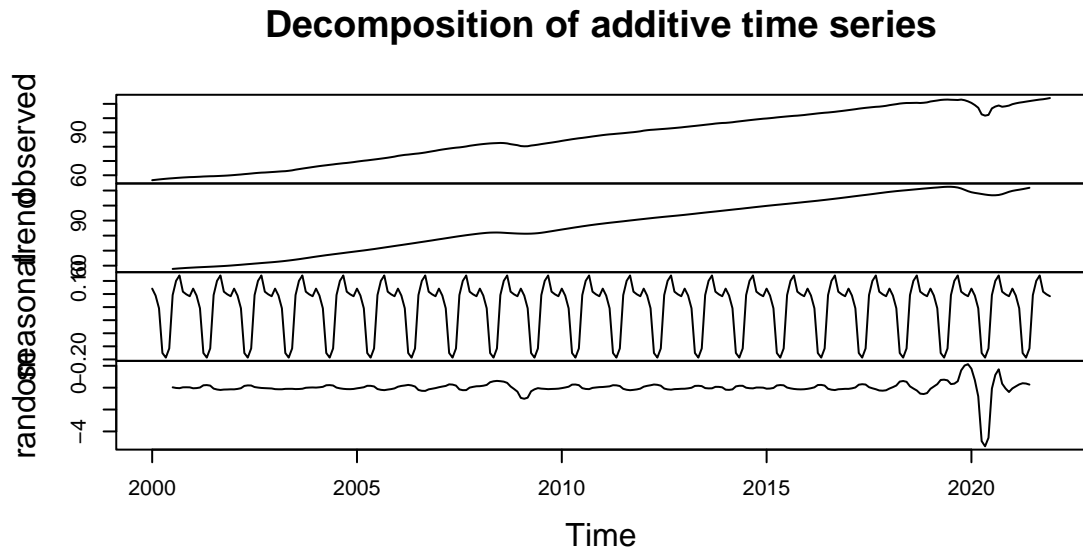
```
df <- read.csv(  
  'G:/My Drive/1.3 Master/Modules/08_Data_mining/TareaII/EIA-STEOR_GDPQ_WORLD_M.csv',  
  sep = ",")  
df <- df[order(as.numeric(rownames(df))), decreasing = TRUE],,drop=FALSE]  
gdp_ts <- ts(df$Value, start=c(2000,1), frequency = 12)
```

Representación gráfica y descomposición estacional

```
plot.ts(gdp_ts)
```



```
gdp_desc <- decompose(gdp_ts, type = 'additive')
plot(gdp_desc)
```



Podemos ver que la serie presenta una tendencia ascendente lineal con ciertas recesiones en algunos periodos. A priori no hay comportamiento estacional lo cual va a dificultar la predicción.

Contraste de normalidad de los residuos

```
ks.test(gdp_desc$random, 'pnorm')
```

```
##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  gdp_desc$random
## D = 0.33464, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
shapiro.test(gdp_desc$random)
```

```
##
## Shapiro-Wilk normality test
##
## data:  gdp_desc$random
## W = 0.49804, p-value < 2.2e-16
```

Vemos que los residuos no pasan los tests de normalidad.

Tratamiento de la serie

Eliminar la heterocedasticidad. Estabilización de la varianza.

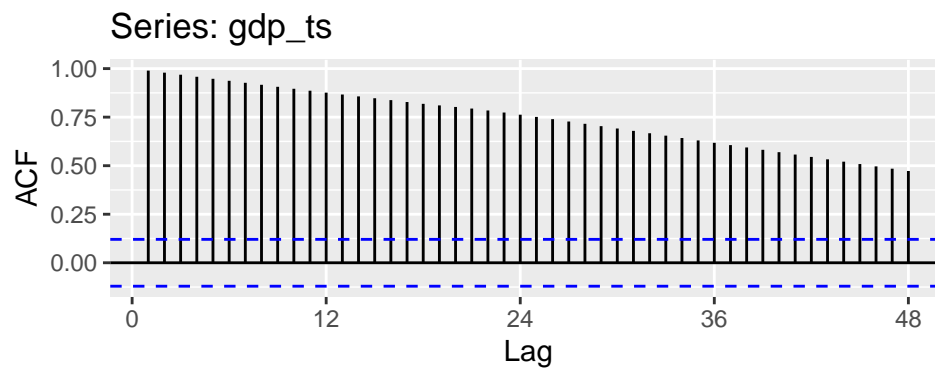
```
gdplog <- log(gdp_ts)
```

```
# Eliminar tendencia  
gdp.diff_1<-diff(gdplog)
```

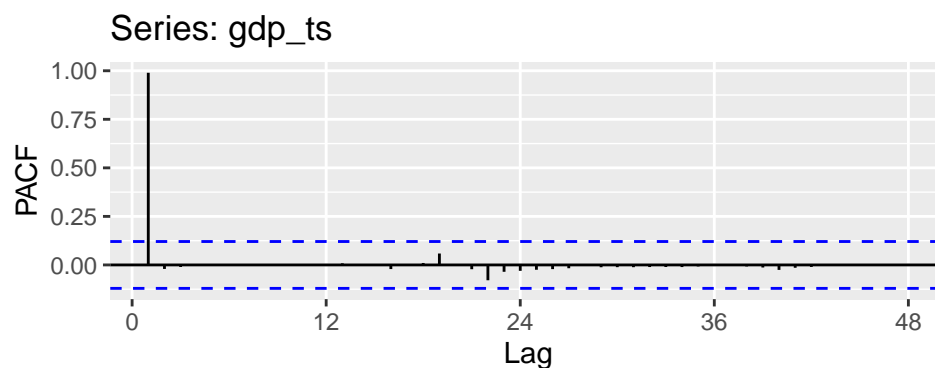
Eliminar tendencia

Comprobación y corrección de la estacionalidad (Correlogramas)

```
library(forecast)  
#Calculamos las autocorrelaciones simples hasta el retardo 48  
ggAcf(gdp_ts, lag=48) # Decrecimineto lento--> No estacionaria-->diferenciar
```



```
#Calculamos las autocorrelaciones parciales hasta el retardo 48  
ggPacf(gdp_ts, lag=48)
```



Vemos que la serie no es estacionaria, por lo tanto, la diferenciamos.

```
gdp.diff_1_12<-diff(gdp.diff_1, lag = 12)
```

```
## Contraste de normalidad de los residuos  
ks.test(gdp.diff_1_12,'pnorm')
```

```
##  
## Asymptotic one-sample Kolmogorov-Smirnov test  
##  
## data: gdp.diff_1_12  
## D = 0.48387, p-value < 2.2e-16  
## alternative hypothesis: two-sided
```

```
shapiro.test(gdp.diff_1_12)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: gdp.diff_1_12  
## W = 0.57604, p-value < 2.2e-16
```

Sigue sin pasar los tests de normalidad.

Ventanas de ajuste y evaluación

```
gdp_tr<-window(x = gdp_ts, end = c(2020,12))  
gdp_tst<-window(x = gdp_ts, start = c(2021,1))
```

Modelos de suavizado exponencial

Suavizado exponencial simple con predicción a un año

```
gdp_s1=ses(gdp_tr, h=12)
```

Suavizado Exponencial doble de Holt

```
gdp_sh <- holt(gdp_tr, h=12)
```

Ajuste de modelo

```
gdp_hw_add <- hw(gdp_tr, h=12,level = c(80, 95))  
gdp_hw_mul <- hw(gdp_tr, h=12, seasonal="multiplicative",level = c(80, 95))
```

Predicciones utilizando ETS

```
ETS_pred<-forecast(gdp_tr,h=12)
```

Modelos ARIMA

Ajuste con la función auto.arima

```
fitgdp <- auto.arima(gdp_tr,seasonal=TRUE)
pred <- forecast(fitgdp, h=12)
```

Ajuste manual de ARIMA

```
fitgdp_manual <- gdp_tr %>% Arima(order=c(1,1,1), seasonal=c(0,0,1))
pred_manual <- forecast(fitgdp_manual, h=12)
```

Se prueba la precisión de las distintas predicciones de los modelos

```
accuracy(gdp_s1,gdp_tst)
```

```
##                ME        RMSE        MAE        MPE        MAPE        MASE
## Training set  0.2073179  0.5382369  0.3256878  0.2581361  0.3749126  0.1049427
## Test set     3.2651359  3.5098134  3.2651359  2.9010985  2.9010985  1.0520881
##                ACF1 Theil's U
## Training set  0.5361040      NA
## Test set     0.7234365  9.068741
```

```
accuracy(gdp_sh,gdp_tst)
```

```
##                ME        RMSE        MAE        MPE        MAPE        MASE
## Training set  0.0008501899  0.4708800  0.1373190  0.002020022  0.1412512  0.04424675
## Test set     0.4902944407  0.5342683  0.4902944  0.439481903  0.4394819  0.15798208
##                ACF1 Theil's U
## Training set  0.009804084      NA
## Test set     0.809612483  1.336649
```

```
accuracy(gdp_hw_add,gdp_tst)
```

```
##                ME        RMSE        MAE        MPE        MAPE        MASE
## Training set -0.03801679  0.501254  0.2144489 -0.04924557  0.2410272  0.06909946
## Test set     2.88699581  3.080212  2.8869958  2.56583099  2.5658310  0.93024428
##                ACF1 Theil's U
## Training set  0.5181978      NA
## Test set     0.6941635  7.951705
```

```
accuracy(gdp_hw_mul,gdp_tst)
```

```
##                ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.004536171 0.7126354 0.3800441 -0.005421379 0.4570778 0.1224573
## Test set      2.141824205 2.3047906 2.1418242  1.909111116 1.9091111 0.6901360
##                ACF1 Theil's U
## Training set 0.4678362      NA
## Test set     0.4459574  5.956788
```

```
accuracy(ETS_pred,gdp_tst)
```

```
##                ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.04429215 0.4373885 0.1478183 0.05605733 0.1567605 0.04762985
## Test set     1.88499621 2.0696707 1.8849962 1.67378570 1.6737857 0.60738119
##                ACF1 Theil's U
## Training set -0.04945731      NA
## Test set     0.72762515  5.340513
```

```
accuracy(pred,gdp_tst)
```

```
##                ME      RMSE      MAE      MPE      MAPE
## Training set 0.0004856905 0.3644419 0.1490017 0.001399768 0.1591885
## Test set     -4.7615218616 5.4652867 4.7615219 -4.240614347 4.2406143
##                MASE      ACF1 Theil's U
## Training set 0.04801114 0.2073478      NA
## Test set     1.53425178 0.5869337 14.26192
```

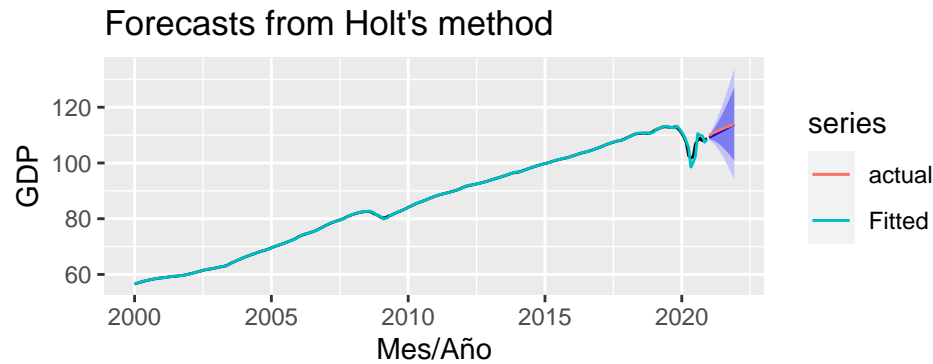
```
accuracy(pred_manual,gdp_tst)
```

```
##                ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.09405538 0.4269676 0.184619 0.1170861 0.2043887 0.05948772
## Test set     1.96530901 2.3793267 1.965309 1.7414905 1.7414905 0.63325948
##                ACF1 Theil's U
## Training set -0.04658652      NA
## Test set     0.77654148  6.13155
```

Vemos que la predicción que más se ajusta a los datos test es la del método Holt's (gdp_sh). Por lo tanto lo seleccionamos y ajustamos a los datos totales.

Representación de la predicción del mejor modelo (Doble Holt)

```
#Representamos los valores observados y los suavizados con la predicción
autoplot(gdp_sh) +
  autolayer(fitted(gdp_sh), series="Fitted") +autolayer(gdp_tst, series="actual") +
  ylab("GDP") + xlab("Mes/Año")
```



Ajuste del modelo final a los datos completos y predicción

```
### Suavizado Exponencial doble de Holt
gdp_sh_total <- holt(gdp_ts, h=12)
# Inspección del objeto creado y Distribución de residuos
print(gdp_sh_total)
```

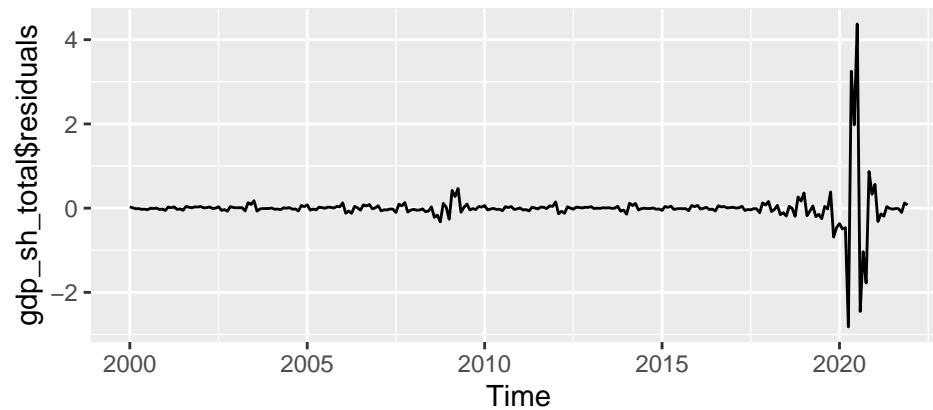
```
##          Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## Jan 2022      114.4964  113.8996  115.0932  113.5836  115.4092
## Feb 2022      114.9209  113.6805  116.1614  113.0238  116.8180
## Mar 2022      115.3454  113.3370  117.3539  112.2738  118.4171
## Apr 2022      115.7699  112.8845  118.6554  111.3570  120.1828
## May 2022      116.1945  112.3345  120.0544  110.2912  122.0977
## Jun 2022      116.6190  111.6958  121.5422  109.0896  124.1484
## Jul 2022      117.0435  110.9750  123.1120  107.7625  126.3245
## Aug 2022      117.4680  110.1776  124.7585  106.3182  128.6178
## Sep 2022      117.8925  109.3081  126.4770  104.7637  131.0213
## Oct 2022      118.3170  108.3702  128.2638  103.1047  133.5293
## Nov 2022      118.7416  107.3674  130.1157  101.3463  136.1369
## Dec 2022      119.1661  106.3023  132.0299   99.4926  138.8395
```

```
gdp_sh_total$model
```

```
## Holt's method
##
## Call:
## holt(y = gdp_ts, h = 12)
##
## Smoothing parameters:
##   alpha = 0.9975
##   beta  = 0.8246
##
## Initial states:
##   l = 56.2792
##   b = 0.2389
##
## sigma: 0.4657
##
```

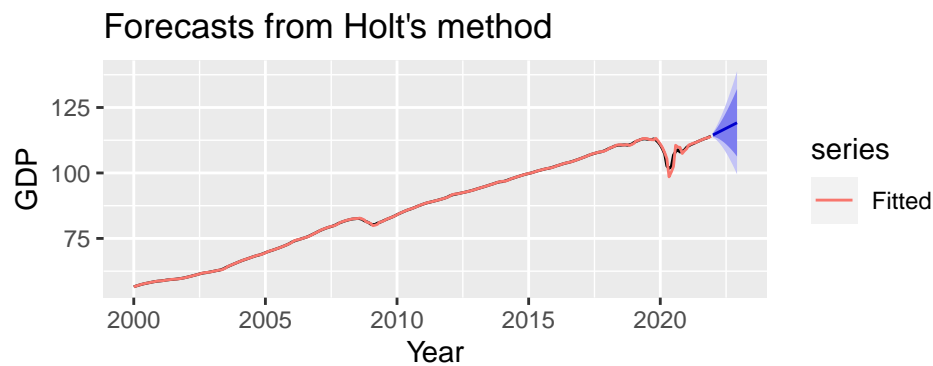
```
##      AIC      AICc      BIC
## 1074.518 1074.751 1092.398
```

```
autoplot(gdp_sh_total$residuals)
```



Representación gráfica de la predicción

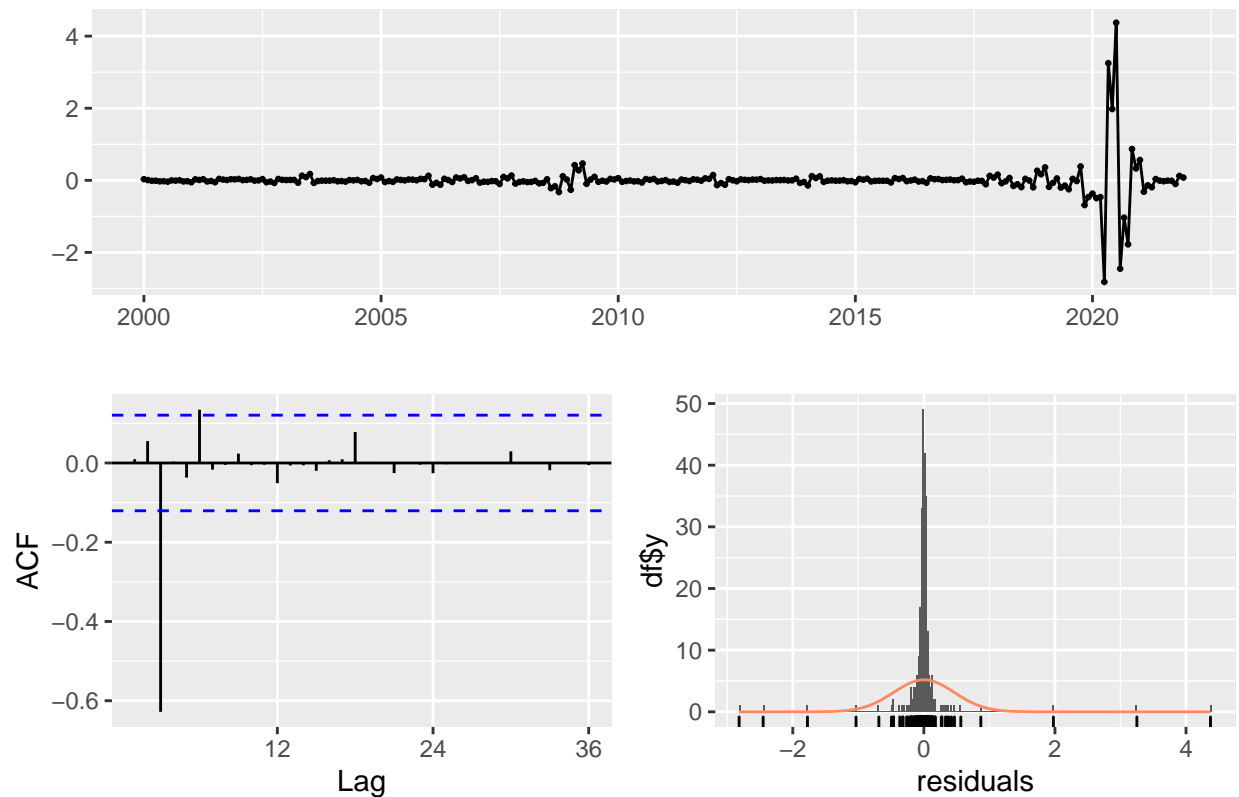
```
autoplot(gdp_sh_total)+ autolayer(fitted(gdp_sh_total), series="Fitted") +
  ylab("GDP") + xlab("Year")
```



Test de Ljung-Box

```
forecast::checkresiduals(gdp_sh_total)
```


Residuals from Holt's method



```
##  
## Ljung-Box test  
##  
## data: Residuals from Holt's method  
## Q* = 115.49, df = 20, p-value = 1.887e-15  
##  
## Model df: 4. Total lags used: 24
```

A pesar de ser el mejor modelo no pasa el test de Ljung-Box de residuos incorrelados.

Conclusiones

Parece que ninguno de los modelos es lo bastante adecuado como para poder sacar conclusiones y predicciones acertadas. Además, como hemos visto, el mejor modelo no pasa el test de Ljung-Box de residuos incorrelados, por lo que las predicciones que hagamos no serán precisas. Aún así, con el objetivo de sacar una conclusión final de esta práctica, siguiendo la tendencia y la predicción del suavizado exponencial doble Holt, podemos concluir que el PIB mundial seguirá su tendencia creciente en 2022.