

Laboratorio

Enrique Sayas Bailach

29/9/2023

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':  
##   method           from  
##   as.zoo.data.frame zoo
```

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.3
```

```
##  
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':  
##  
##   date, intersect, setdiff, union
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

Introducción

Los dos datasets utilizados han sido descargados desde Kaggle.

El utilizado para la serie temporal con tendencia es `annual_gold_rate`, que muestra el valor anual del oro desde 1980 hasta 2022 en Dirham de los Emiratos Árabes Unidos.

Por otra parte, el dataset utilizado para la serie temporal con tendencia y estacionalidad es `HospitalityEmployees`, que muestra el número de empleados en la hostelería en California desde enero de 1990 hasta diciembre de 2018.

Parte 1

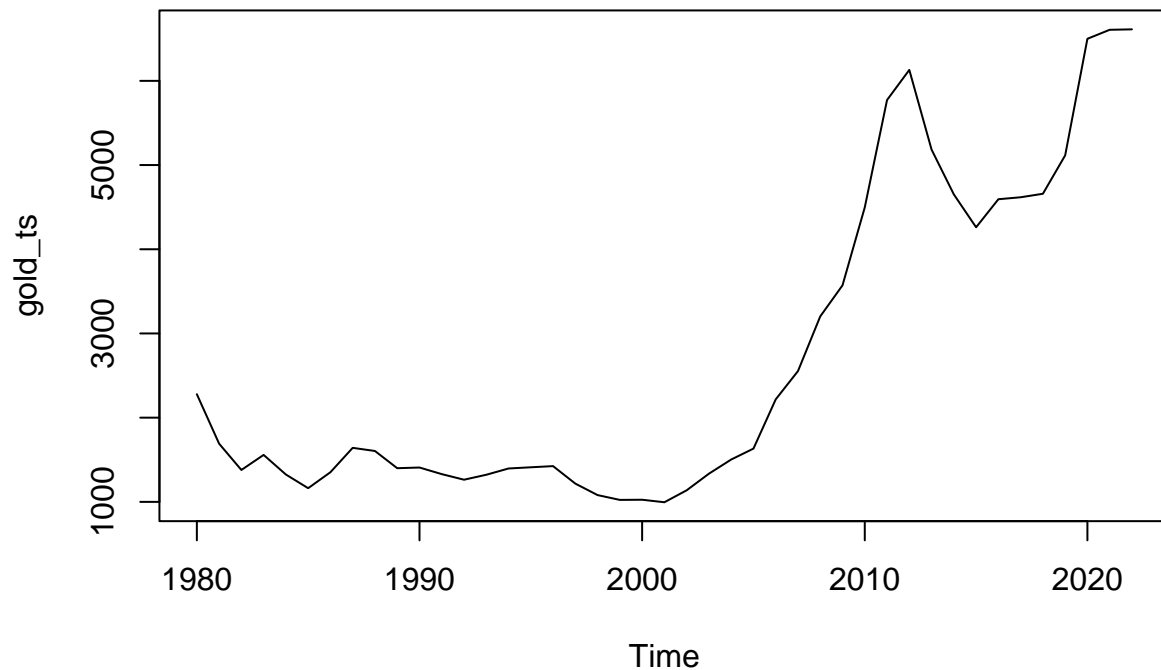
Tendencia

Importamos los datos

```
gold <- read_csv("Datos/annual_gold_rate.csv",  
  col_types = cols(Date = col_date(format = "%Y-%m-%d")))
```

Creación de la serie temporal

```
attach(gold)  
gold_ts <- ts(AED, start=c(1980), frequency=1)  
plot(gold_ts)
```



Se puede observar que hay una tendencia.

Estacionalidad y Tendencia

```
HospitalityEmployees <- read_csv("Datos/HospitalityEmployees.csv")
```

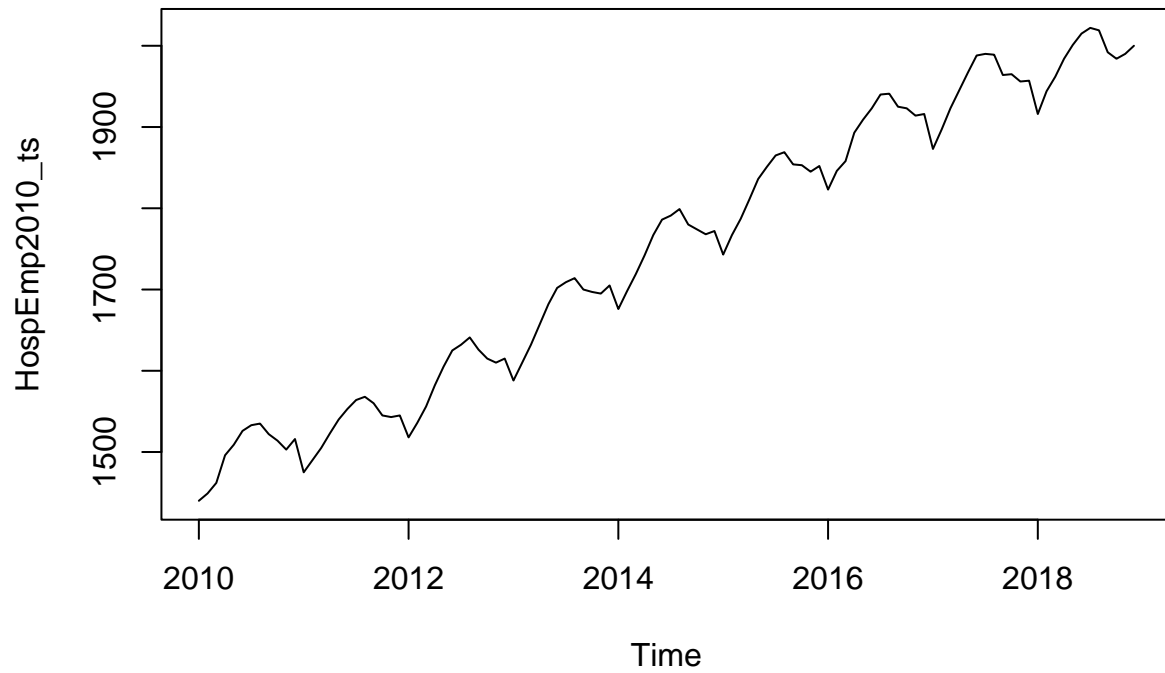
```
## Rows: 348 Columns: 2
## -- Column specification -----
## Delimiter: ","
## chr (1): Date
## dbl (1): Employees
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
HospitalityEmployees$Date <- as.Date(HospitalityEmployees$Date, format = "%m/%d/%Y")
HospitalityEmployees$Employees <- floor(HospitalityEmployees$Employees)
```

```
attach(HospitalityEmployees)
```

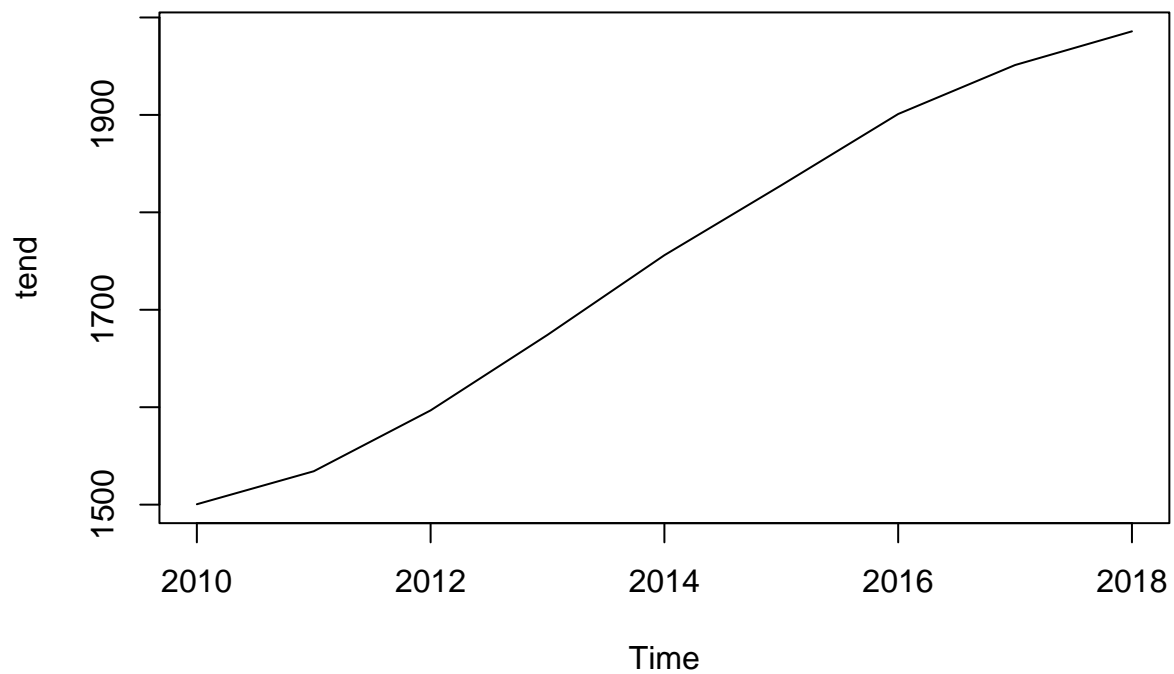
```
## The following object is masked from gold:
##
## Date
```

```
HospitalityEmployees_ts <- ts(Employees,start=c(1990,1),end=c(2018,12),frequency=12)
HospEmp2010_ts <- window(HospitalityEmployees_ts, start = c(2010,1))
plot(HospEmp2010_ts)
```



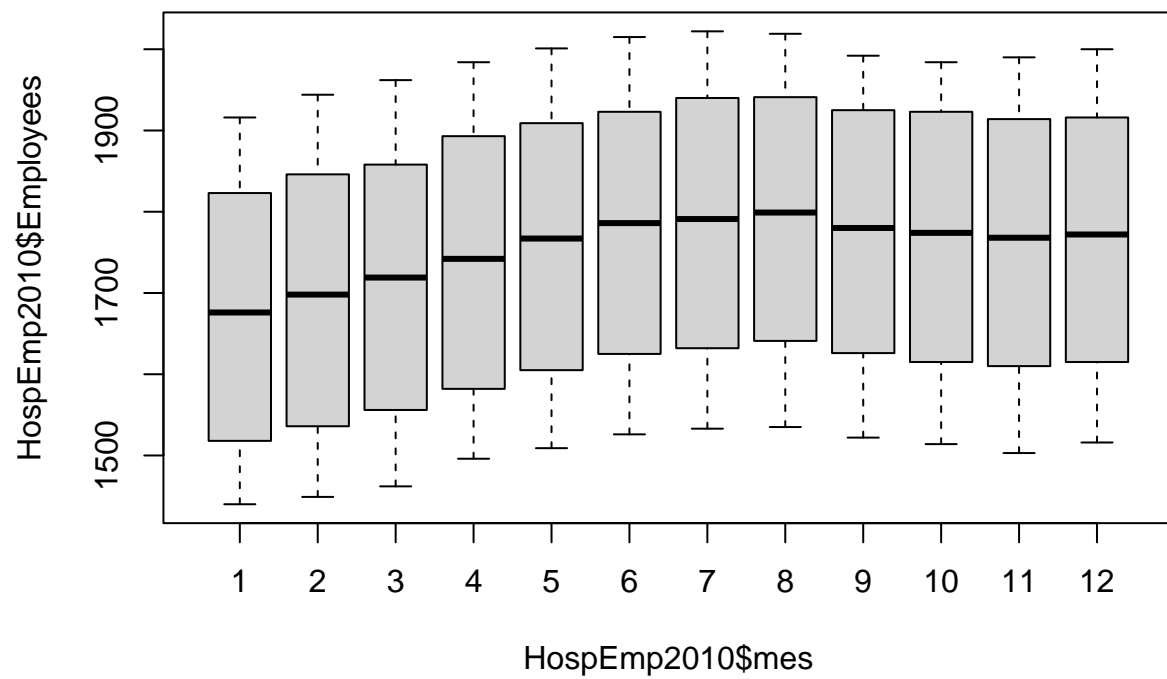
```
tend <- aggregate(HospEmp2010_ts, FUN = mean)
```

```
plot(tend)
```



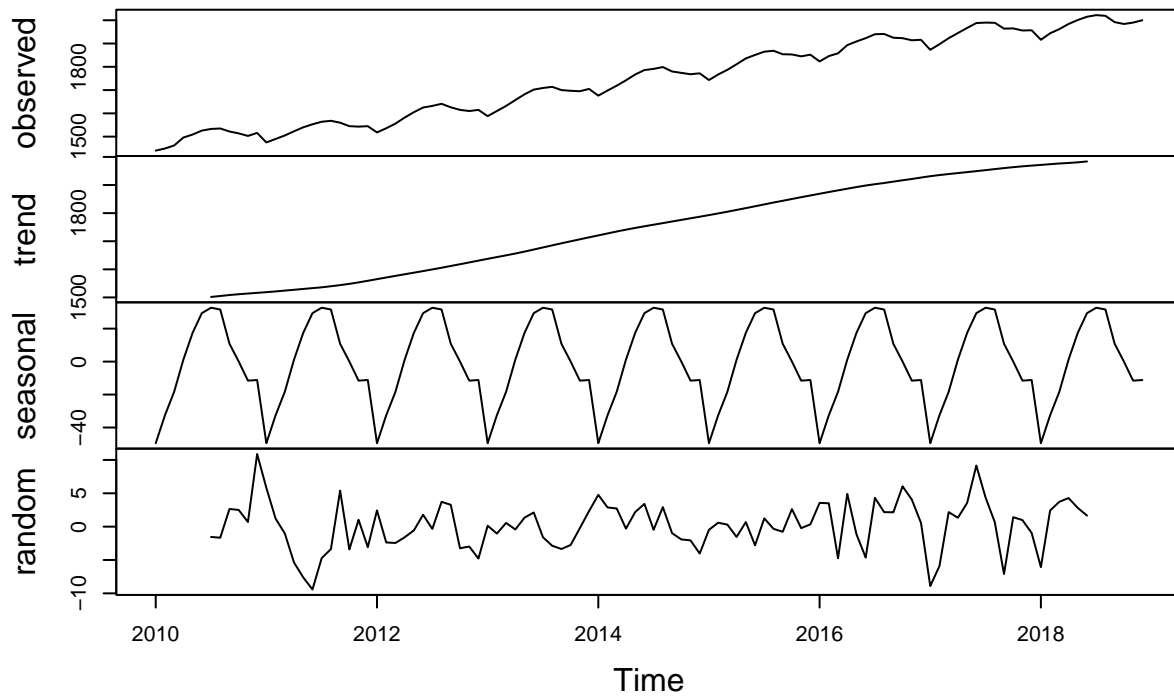
```
HospEmp2010 <- HospitalityEmployees %>% mutate(mes = month(Date),  
                                                anyo = year(Date)) %>%  
  filter(anyo >= 2010)
```

```
boxplot(HospEmp2010$Employees ~ HospEmp2010$mes)
```



```
plot(decompose(HospEmp2010_ts, type="additive"))
```

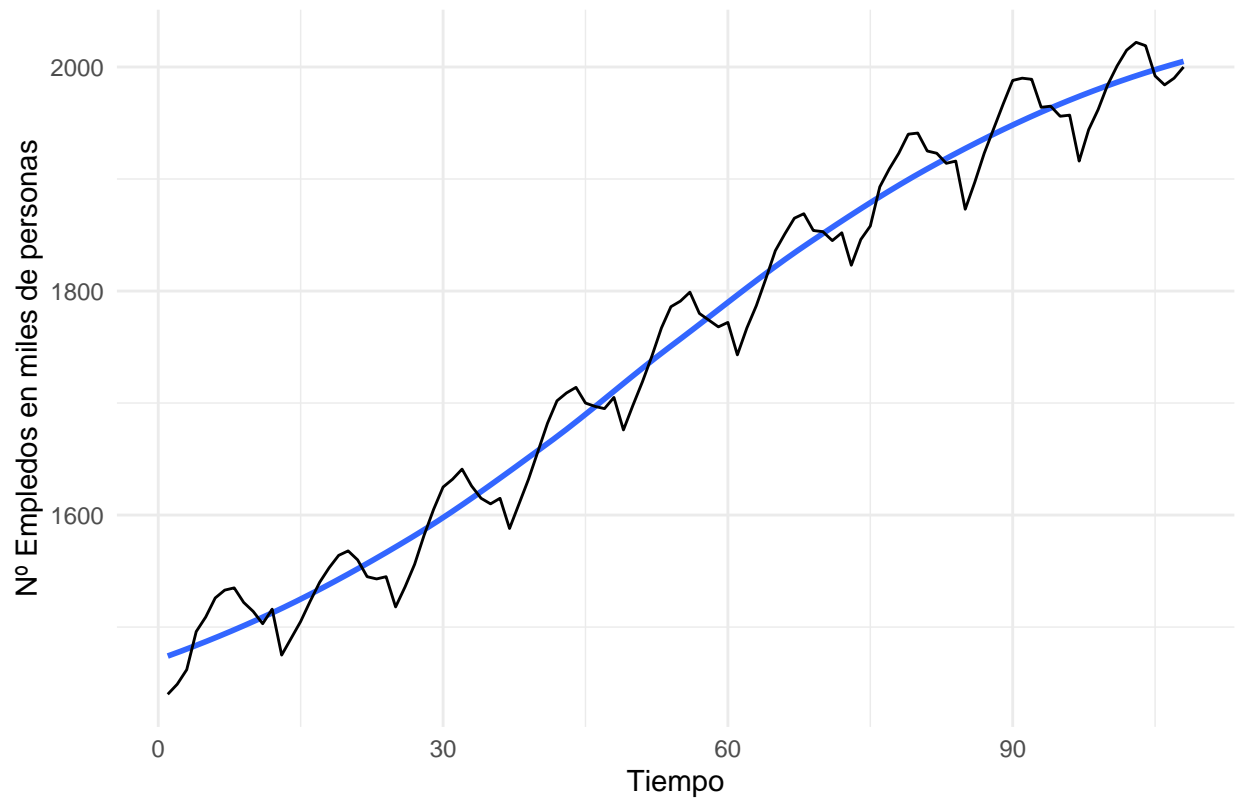
Decomposition of additive time series



```
ggplot(data = as.data.frame(HospEmp2010_ts),  
  aes(x = c(1:length(HospEmp2010_ts)),  
    y = HospEmp2010_ts)) +  
  geom_smooth(se = FALSE) +  
  geom_line() +  
  xlab("Tiempo") +  
  ylab("Nº Empleados en miles de personas") +  
  labs(title = "Evolución del número de empleados en la Hostelería") +  
  theme_minimal()
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting  
## to continuous.  
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

Evolución del número de empleados en la Hostelería



Parte 2

Serie temporal con tendencia

En base a la evolución temporal del valor del oro, se puede observar que existe una tendencia pero no una estacionalidad.

Por tanto, el mejor modelo de suavizado exponencial será el modelo de Host.

Las ecuaciones de observación y actualización del modelo son las siguientes:

$$\begin{aligned}x_t &= L_{t-1} + T_{t-1} + \epsilon, \epsilon \sim N(0, \sigma^2) \\L_t &= \alpha \cdot x_t + (1 - \alpha) \cdot (L_{t-1} + T_{t-1}) \\T_t &= \beta \cdot (L_t - L_{t-1}) + (1 - \beta) \cdot T_{t-1}\end{aligned}$$

Ecuación de predicción:

$$\hat{x}_{n+k} = L_n + k \cdot T_n$$

Creación del modelo


```
gold_holt <- HoltWinters(gold_ts, gamma = FALSE)
```

Visualimos los coeficientes y los parámetros

```
gold_holt$coefficients
```

```
##          a          b
## 6611.7100  141.3085
```

```
gold_holt$alpha
```

```
## alpha
##      1
```

```
gold_holt$beta
```

```
##      beta
## 0.6672362
```

Al ser $\alpha = 1$, sólo se tendrá en cuenta el valor anterior para calcular el nivel del valor siguiente.

A partir de los coeficientes obtenidos calculamos las ecuaciones de actualización:

$$L_t = x_t$$

$$T_t = 0.6672362 \cdot (L_t - L_{t-1}) + (1 - 0.6672362) \cdot T_{t-1}$$

Y la ecuación de predicción:

$$\hat{x}_{n+k} = 6611.71 + k \cdot 141.3085$$

Cálculo de la bondad del ajuste

```
fitval_gold <- fitted(gold_holt)
tail(fitval_gold, 10)
```

```
## Time Series:
## Start = 2013
## End = 2022
## Frequency = 1
##      xhat    level    trend
## 2013 6736.377 6130.18  606.19713
## 2014 4753.495 5183.46 -429.96532
## 2015 4153.513 4651.52 -498.00651
## 2016 3834.546 4260.90 -426.35435
## 2017 4674.664 4594.17   80.49448
## 2018 4659.736 4617.43   42.30557
## 2019 4701.048 4659.14   41.90818
## 2020 5433.078 5114.98  318.09846
## 2021 7529.487 6499.70 1029.78691
## 2022 7020.103 6606.30  413.80323
```

```
rmse <- sqrt(mean((gold_ts - fitval_gold[,1])^2))  
rmse
```

```
## [1] 431.5184
```

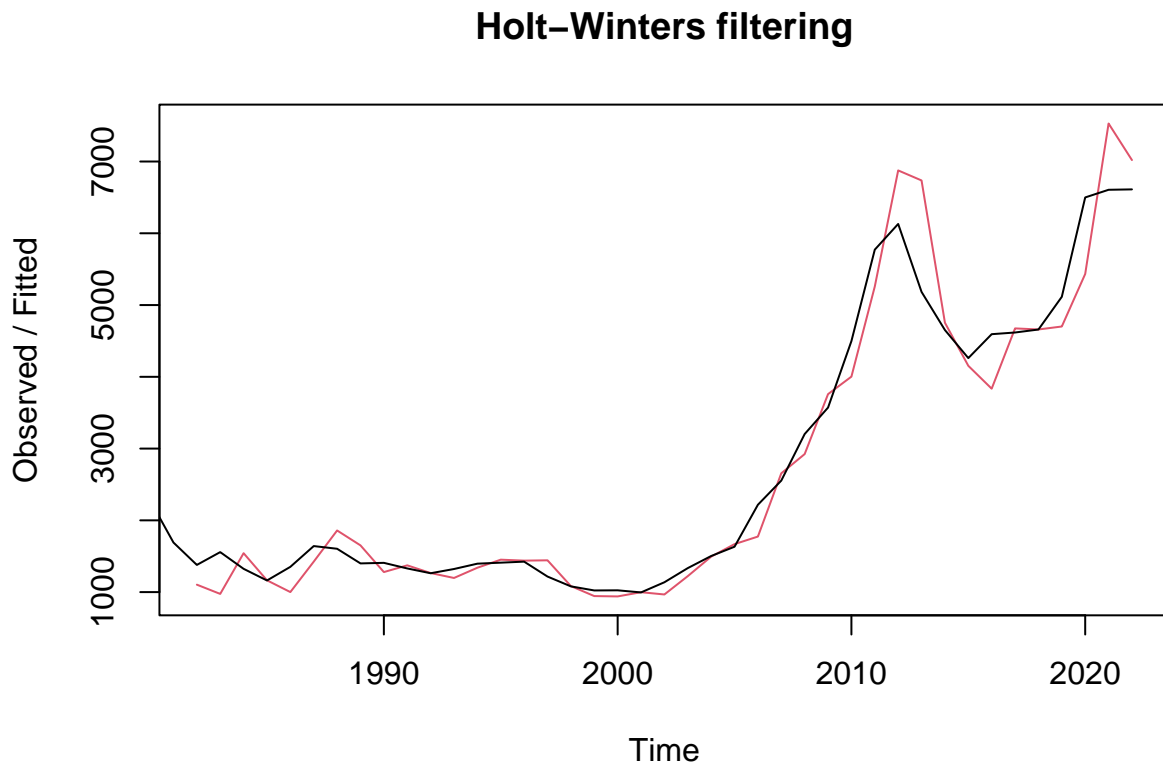
```
mape <- 100*mean(abs(gold_ts-fitval_gold[,1])/gold_ts)  
mape
```

```
## [1] 9.892781
```

A partir del mape se puede concluir que se tiene un error medio del 9.89%.

Representación de la serie real frente a la serie ajustada

```
plot(gold_holt)
```



En el gráfico se puede observar el error medio del 9.89% respecto a la serie original.

Predicción para $h=2$

```
pred_gold <- predict(gold_holt,2)
```

Representación de la predicción junto a la serie

```
ts.plot(gold_ts,pred_gold,lty=1:2)
```



Serie temporal con tendencia + estacionalidad

```
emp_ad <- HoltWinters(HospEmp2010_ts, seasonal = "additive")  
fit_emp_ad <- fitted(emp_ad)
```

```
#rmse  
rmse_ad <- sqrt(mean((HospEmp2010_ts - fit_emp_ad[,1])^2))  
rmse_ad
```

```
## [1] 6.192158
```

```
#mape  
mape_ad <- 100*mean(abs(HospEmp2010_ts-fit_emp_ad[,1])/HospEmp2010_ts)  
mape_ad
```

```
## [1] 0.2809599
```

```
emp_mult <- HoltWinters(HospEmp2010_ts, seasonal = "multiplicative")  
fit_emp_mult <- fitted(emp_mult)
```

```
#rmse
```

```
rmse_mult <- sqrt(mean((HospEmp2010_ts - fit_emp_mult[,1])^2))  
rmse_mult
```

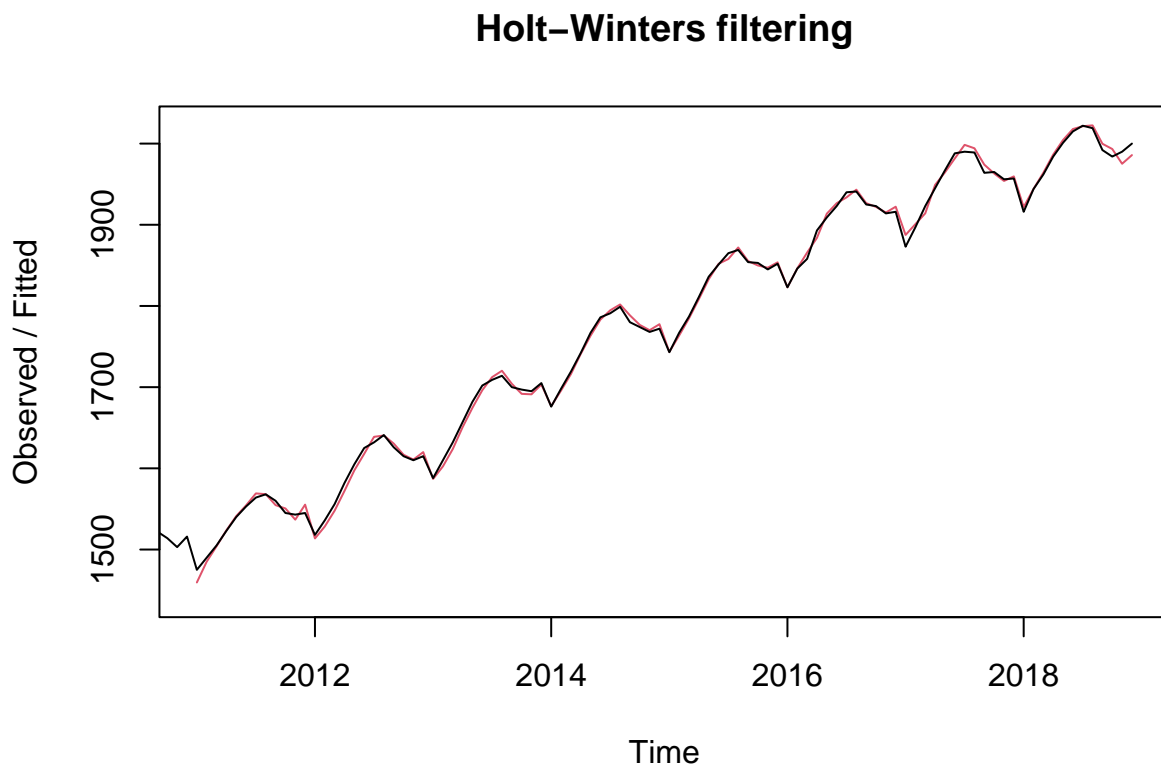
```
## [1] 5.668459
```

```
#mape
```

```
mape_mult <- 100*mean(abs(HospEmp2010_ts-fit_emp_mult[,1])/HospEmp2010_ts)  
mape_mult
```

```
## [1] 0.2562788
```

```
plot(emp_mult)
```



Holt-Winters	RMSE	MAPE
Additive	6.192158	0.2809599
Multiplicative	5.668459	0.2562788

En base a la evolución temporal del número de empleados en la hostelería, se puede observar que existe una tendencia y una estacionalidad aditiva.

Por tanto, el mejor modelo de suavizado exponencial será el modelo de Host-Winters con estacionalidad aditiva.

Las ecuaciones de observación y actualización del modelo son las siguientes:

$$\hat{x}_t = L_{t-1} + T_{t-1} + S_{t-c}$$

$$L_t = \alpha \cdot (x_t - S_{t-c}) + (1 - \alpha) \cdot (L_{t-1} + T_{t-1})$$

$$T_t = \beta \cdot (L_t - L_{t-1}) + (1 - \beta) \cdot T_{t-1}$$

$$S_t = \gamma \cdot (x_t - L_t) + (1 - \gamma) \cdot S_{t-c}$$

Y la ecuación de predicción:

$$\hat{x}_{n+k} = L_n + k \cdot T_n + S_{n+k-c}$$

Creación del modelo

```
emp_fit <- HoltWinters(HospEmp2010_ts, seasonal = "additive")
```

Visualizamos los coeficientes y los parámetros

```
emp_fit$coefficients
```

```
##          a          b          s1          s2          s3          s4
## 2009.266255  4.174587 -50.815982 -25.284286 -9.900944  8.480798
##          s5          s6          s7          s8          s9          s10
##  21.666781 31.274928  32.127005  26.301510  1.009574 -3.544818
##          s11          s12
##  -8.699546 -9.266255
```

```
emp_fit$alpha
```

```
##      alpha
## 0.6583277
```

```
emp_fit$beta
```

```
##      beta
## 0.05505944
```

```
emp_fit$gamma
```

```
##  gamma
##      1
```

Al ser $\beta = 0.055$ la tendencia es constante y como $\gamma = 1$, la estacionalidad la calculará a través de la diferencia entre el valor de x_t y L_t .

A partir de los coeficientes obtenidos calculamos las ecuaciones de actualización:

$$L_t = 0.6583277 \cdot (x_t - S_{t-12}) + (1 - 0.6583277) \cdot (L_{t-1} + T_{t-1})$$

$$T_t = 0.05505944 \cdot (L_t - L_{t-1}) + (1 - 0.05505944) \cdot T_{t-1}$$

$$S_t = x_t - L_t$$

Y la ecuación de predicción

$$\hat{x}_{n+k} = 2009.266255 + k \cdot 4.174587 + S_{n+k-12}$$

Cálculo de la bondad del ajuste

```
fitval_emp <- fitted(emp_fit)
tail(fitval_emp, 10)
```

```
##           xhat    level    trend    season
## Mar 2018 1964.329 1969.284 4.149945 -9.1051677
## Apr 2018 1984.679 1971.901 4.065523  8.7129286
## May 2018 2001.345 1975.519 4.040897 21.7845324
## Jun 2018 2014.448 1979.333 4.028405 31.0862977
## Jul 2018 2018.811 1983.725 4.048416 31.0373608
## Aug 2018 2021.033 1989.873 4.164014 26.9962024
## Sep 2018 2000.808 1992.698 4.090316  4.0189341
## Oct 2018 1994.962 1990.990 3.771060  0.2006361
## Nov 2018 1978.181 1987.545 3.373715 -12.7378949
## Dec 2018 1989.725 1998.700 3.802133 -12.7770725
```

```
rmse <- sqrt(mean((HospEmp2010_ts - fitval_emp[,1])^2))
rmse
```

```
## [1] 6.192158
```

```
mape <- 100*mean(abs(HospEmp2010_ts - fitval_emp[,1])/HospEmp2010_ts)
mape
```

```
## [1] 0.2809599
```

El ajuste es muy bueno pues sólo se tiene un error medio del 0.28%.

Representación de la serie real frente a la serie ajustada

```
plot(emp_fit)
```

Holt-Winters filtering

