Assignment 1

2023-02-24

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

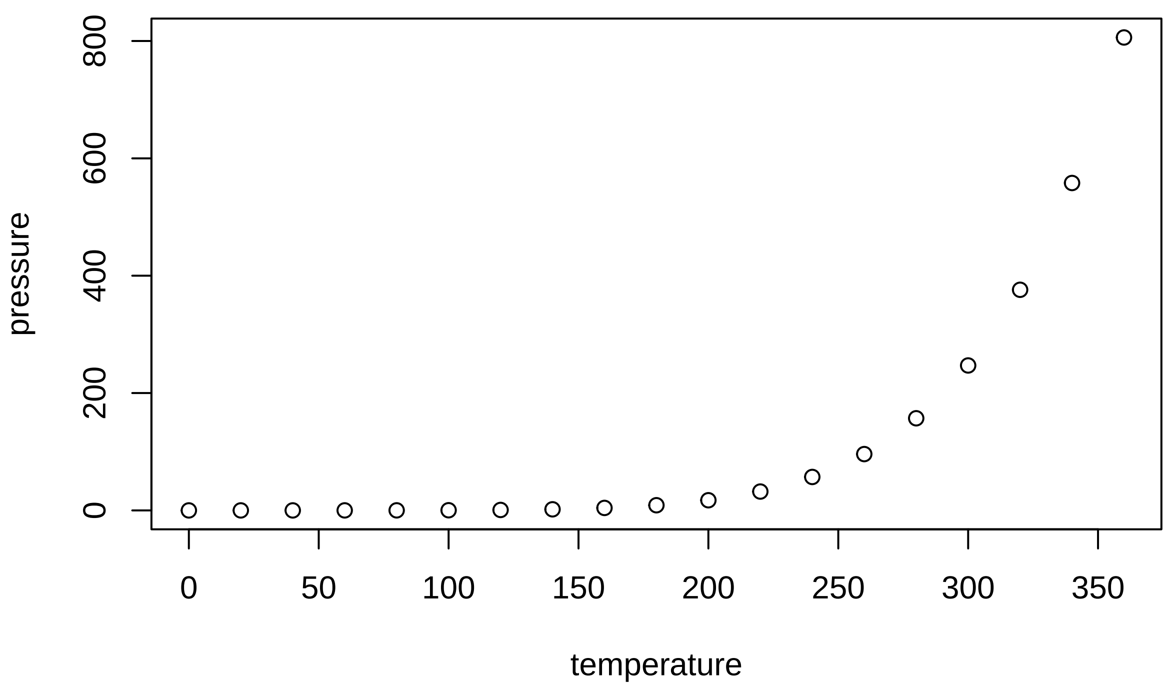
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:

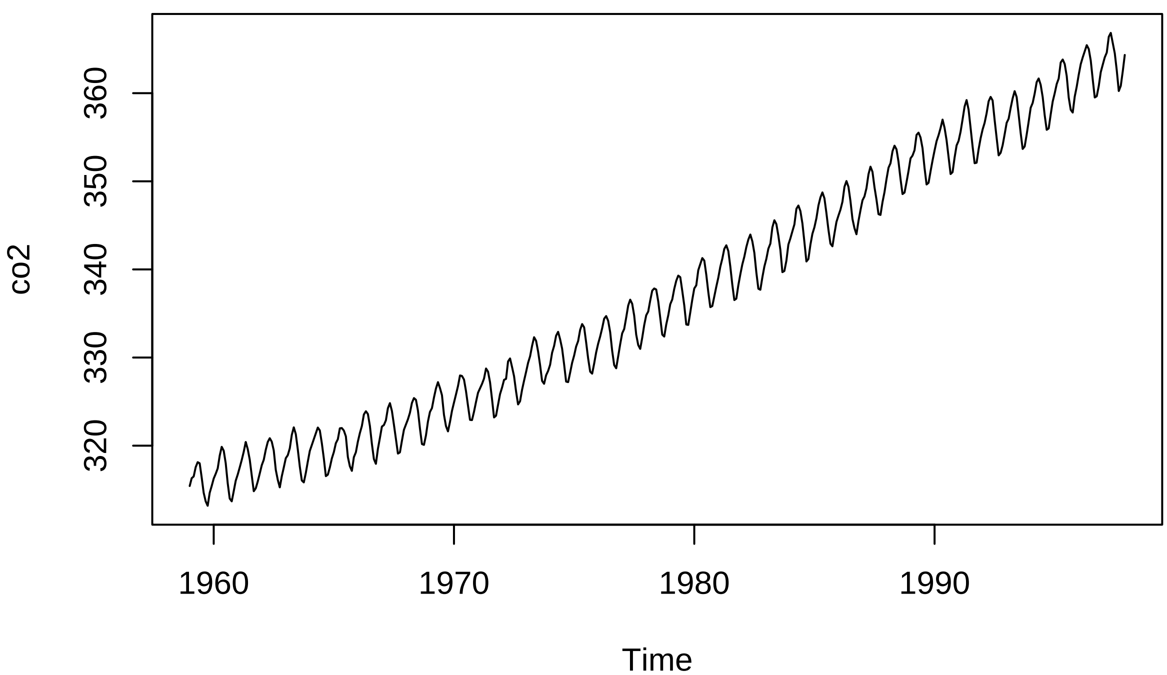


Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
?co2
summary(co2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   313.2   323.5   335.2   337.1   350.3   366.8
```

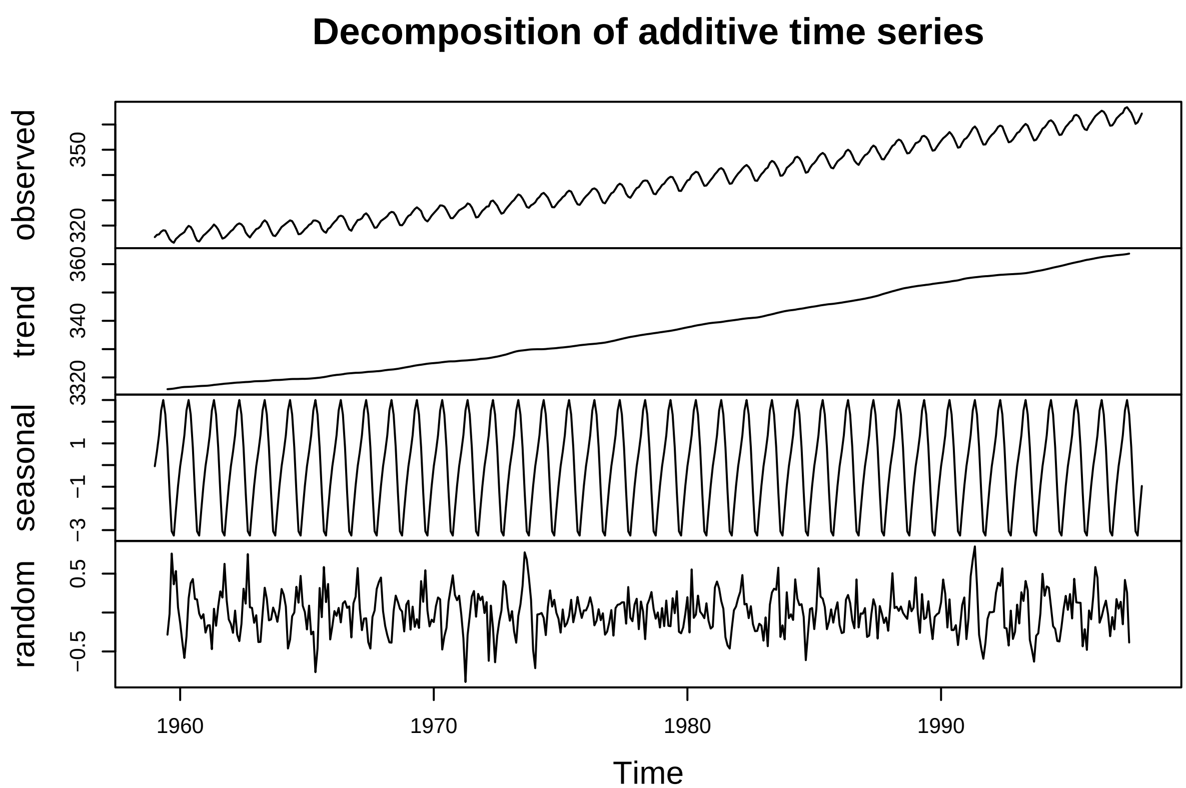
```
plot(co2)
```



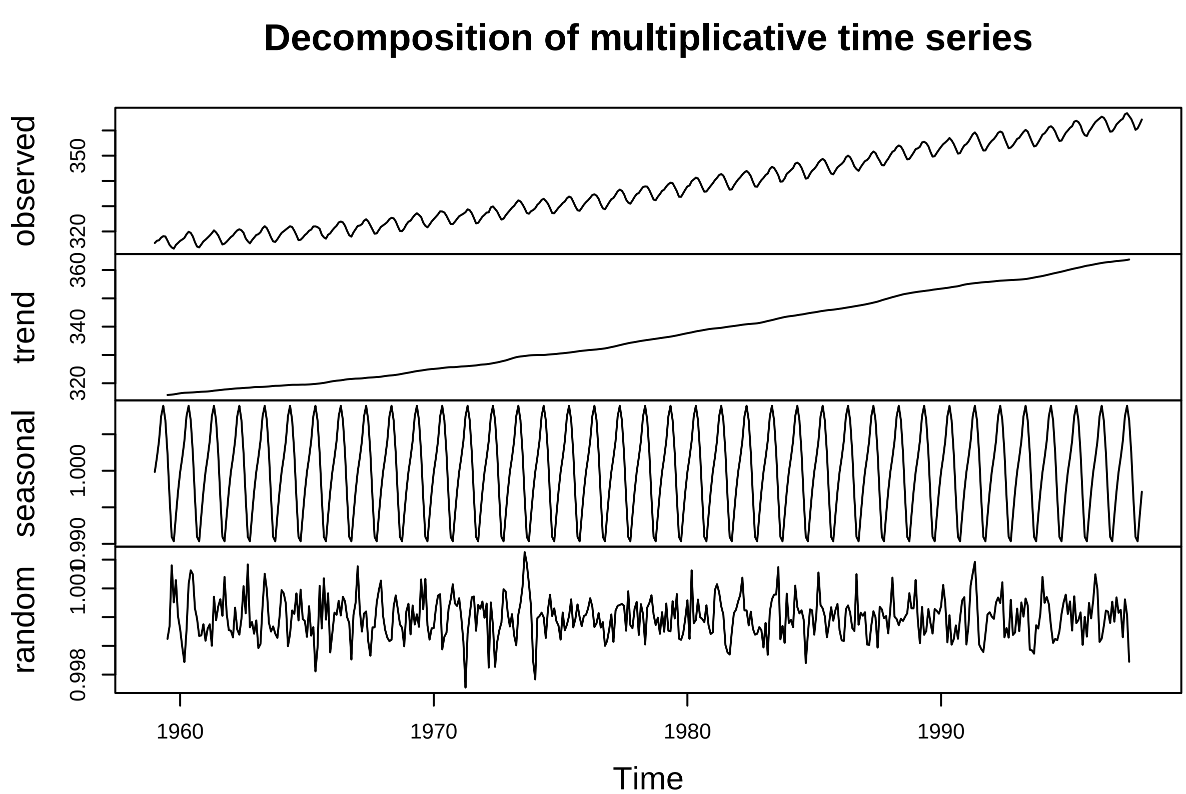

```
m <- decompose(co2)
names(m)
```

```
## [1] "x"        "seasonal" "trend"    "random"   "figure"   "type"
```

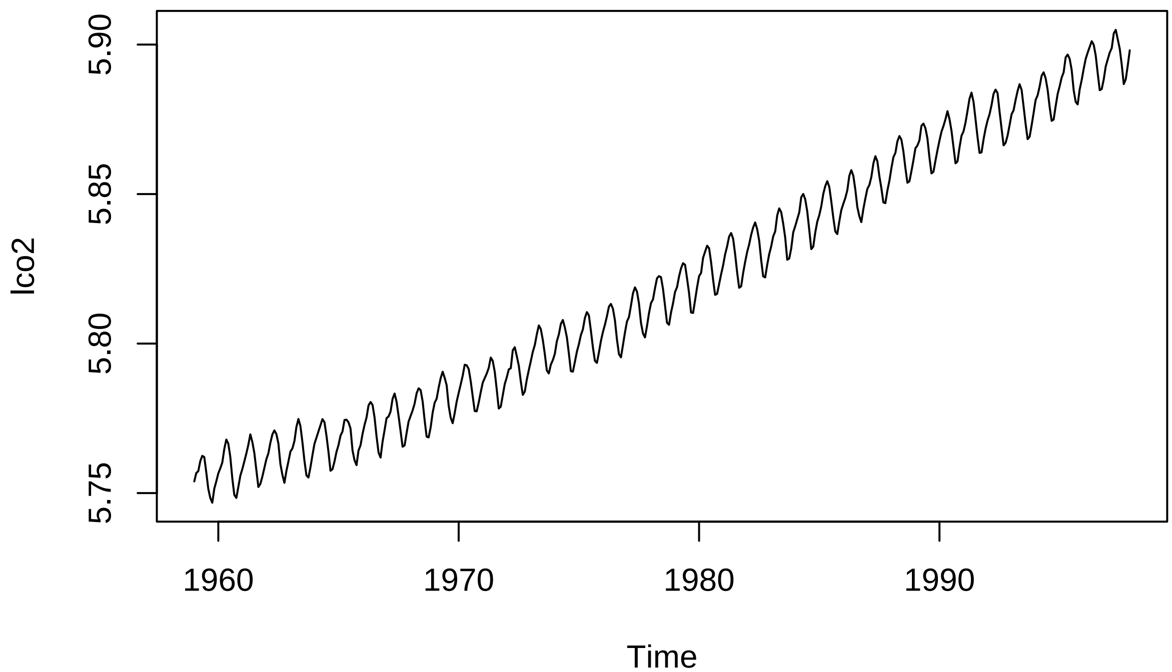
```
plot(m)
plot(decompose(co2, type="additive"))
```



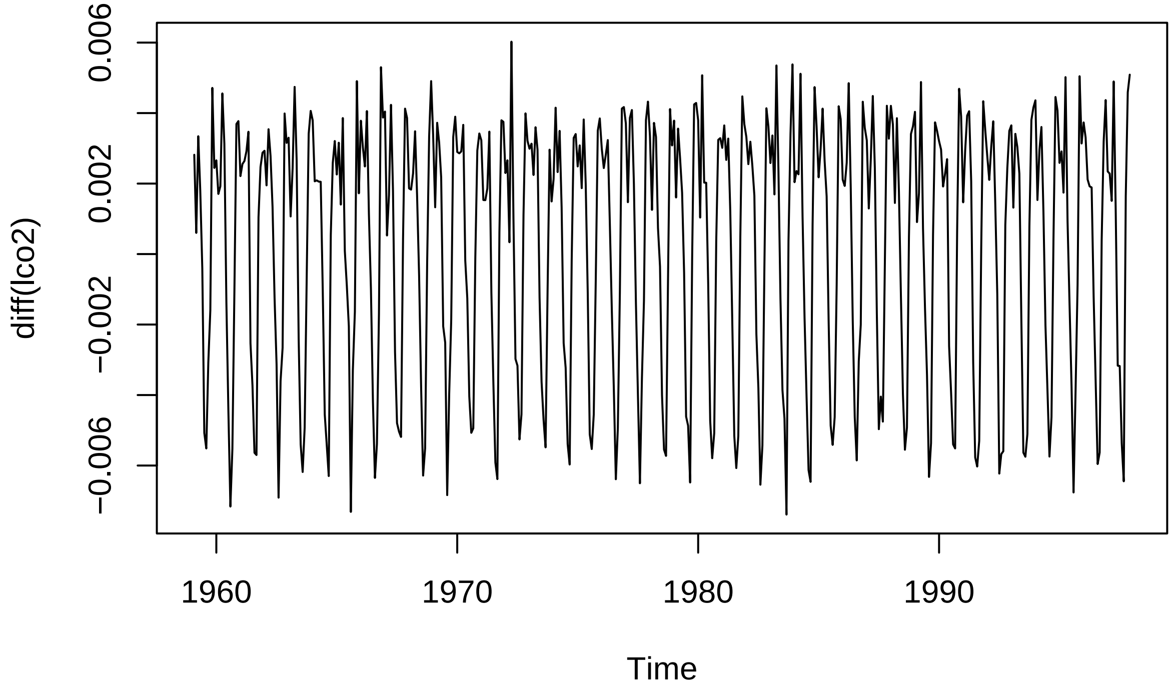

```
plot(decompose(co2, type="multiplicative"))
```



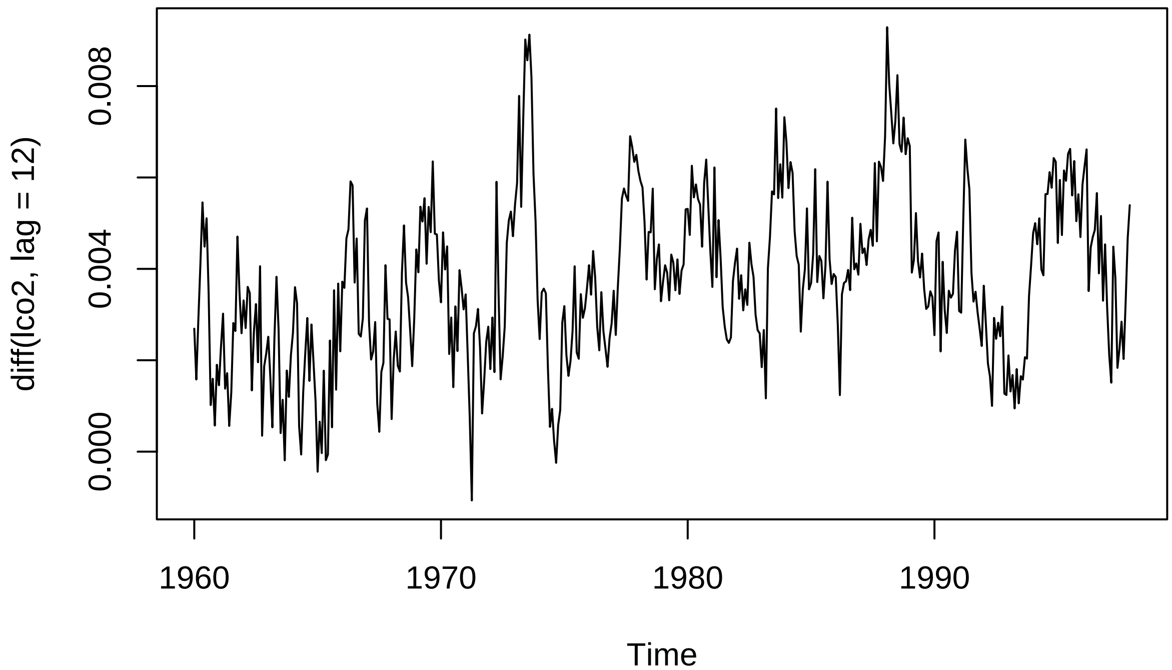
```
lco2=log(co2)
plot(lco2)
```



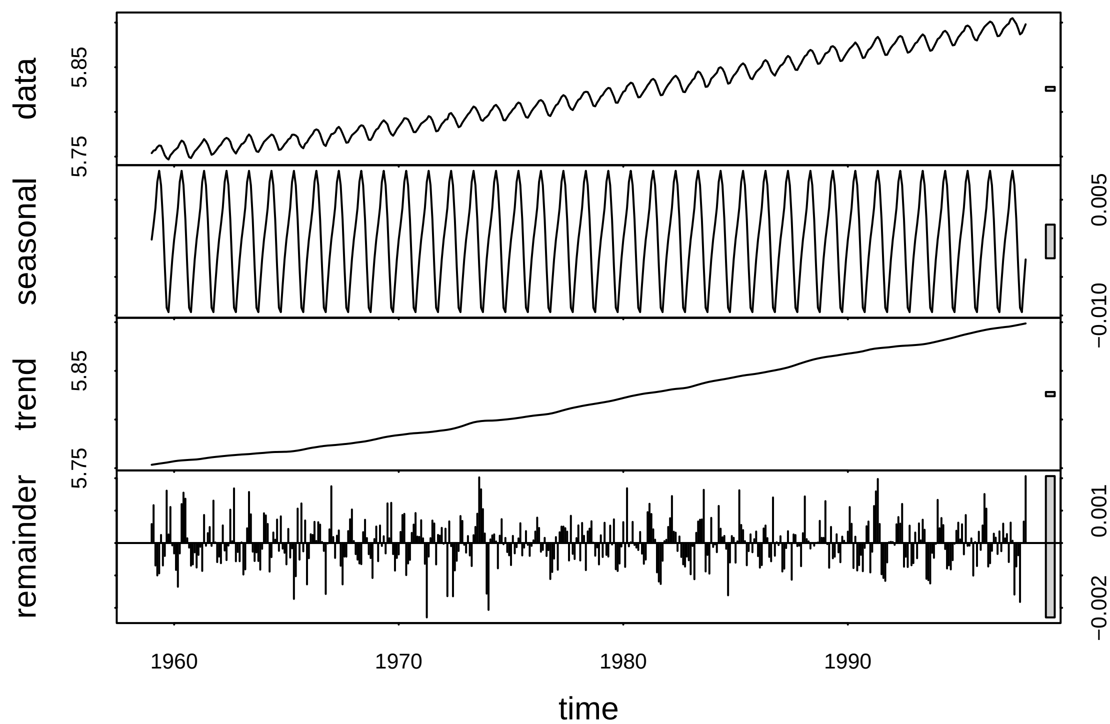

```
plot(diff(lco2))
```




```
plot(diff(lco2, lag=12))
```




```
plot(stl(lco2, s.window="periodic"))
```



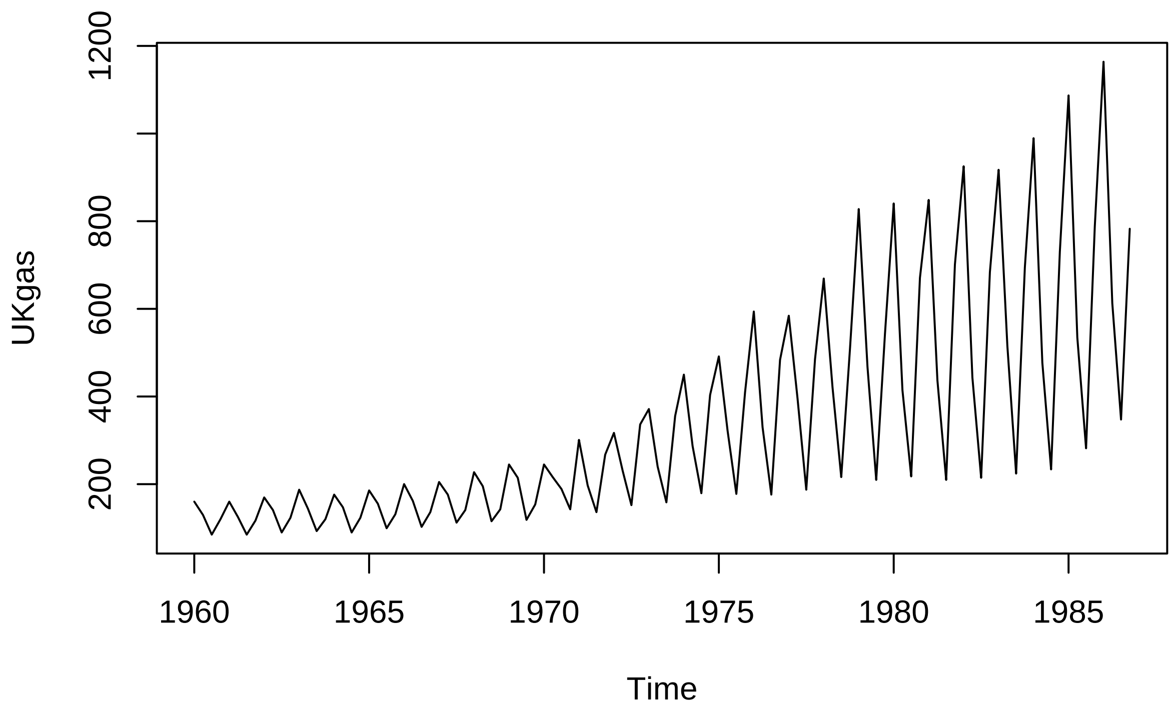

```
# Decomposing the time series shows a positive trend in atmospheric concentrations of CO2 between 1959
```

```
?UKgas
summary(UKgas)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##    84.8   153.3   220.9   337.6   469.9  1163.9
```

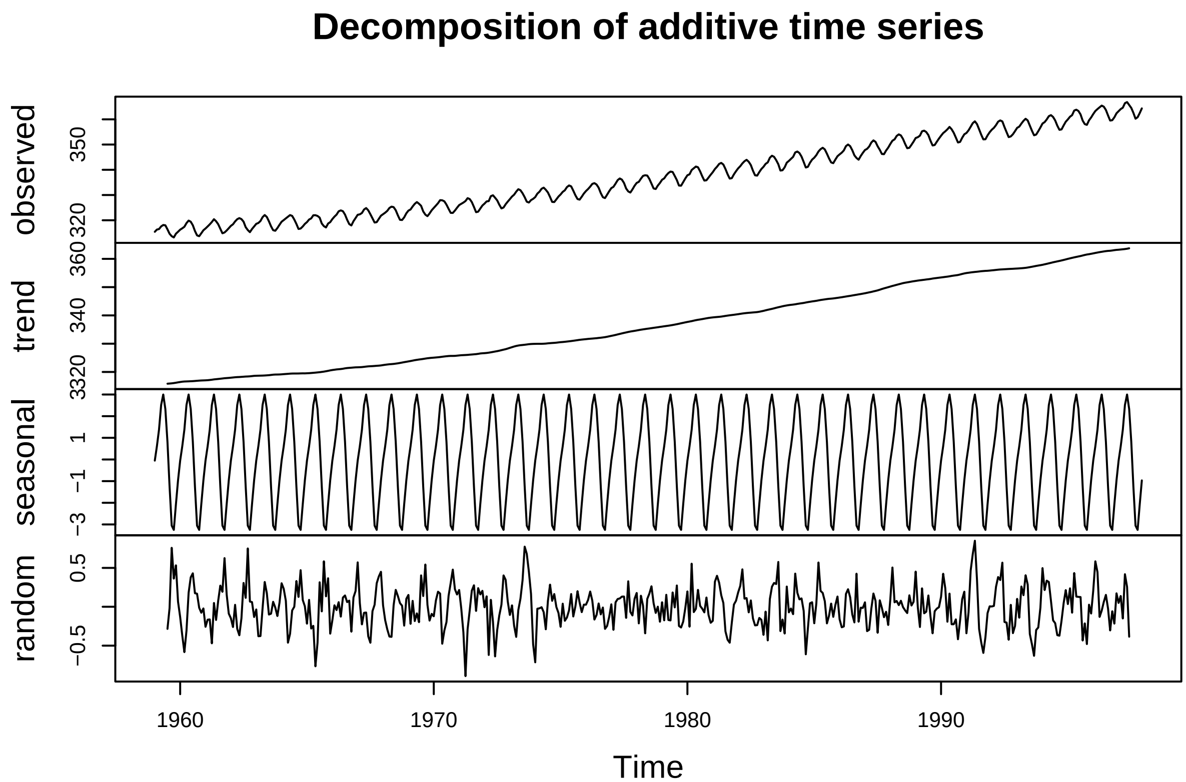
```r
plot(UKgas)
```



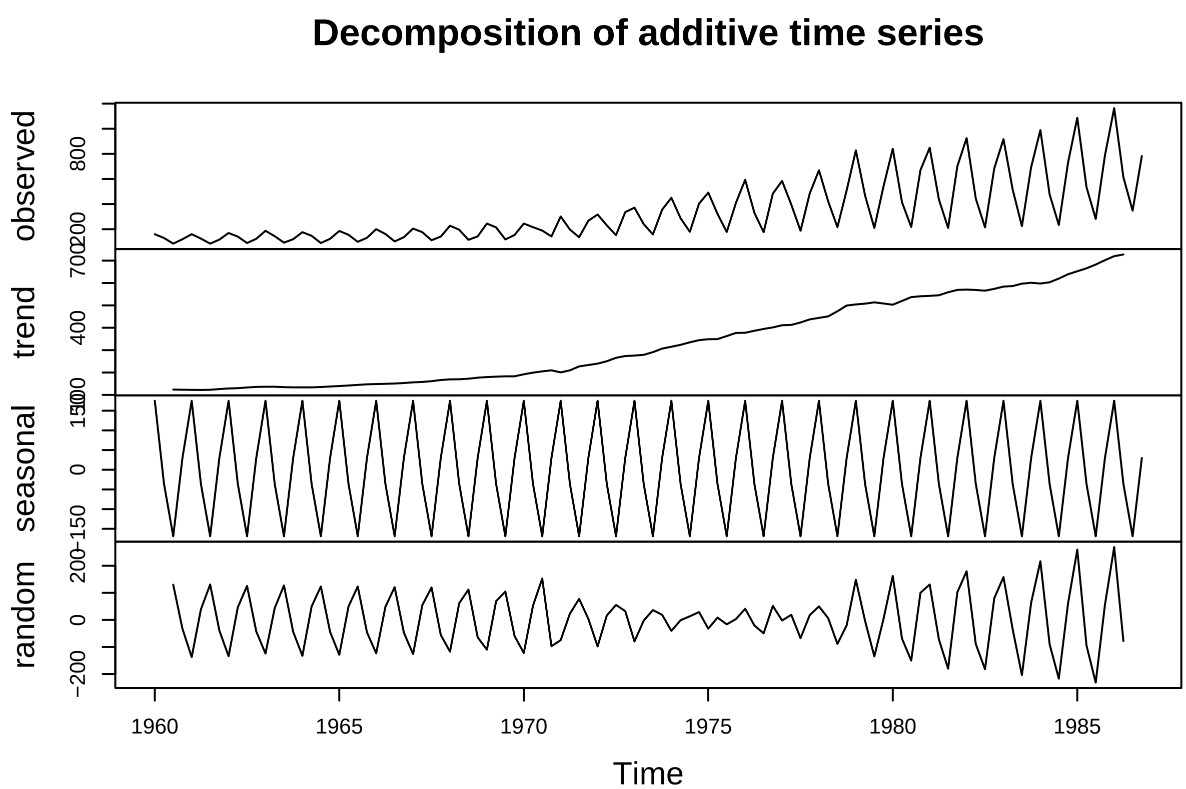

```r
n <- decompose(co2)
names(n)
```

```
## [1] "x"        "seasonal" "trend"    "random"   "figure"   "type"
```

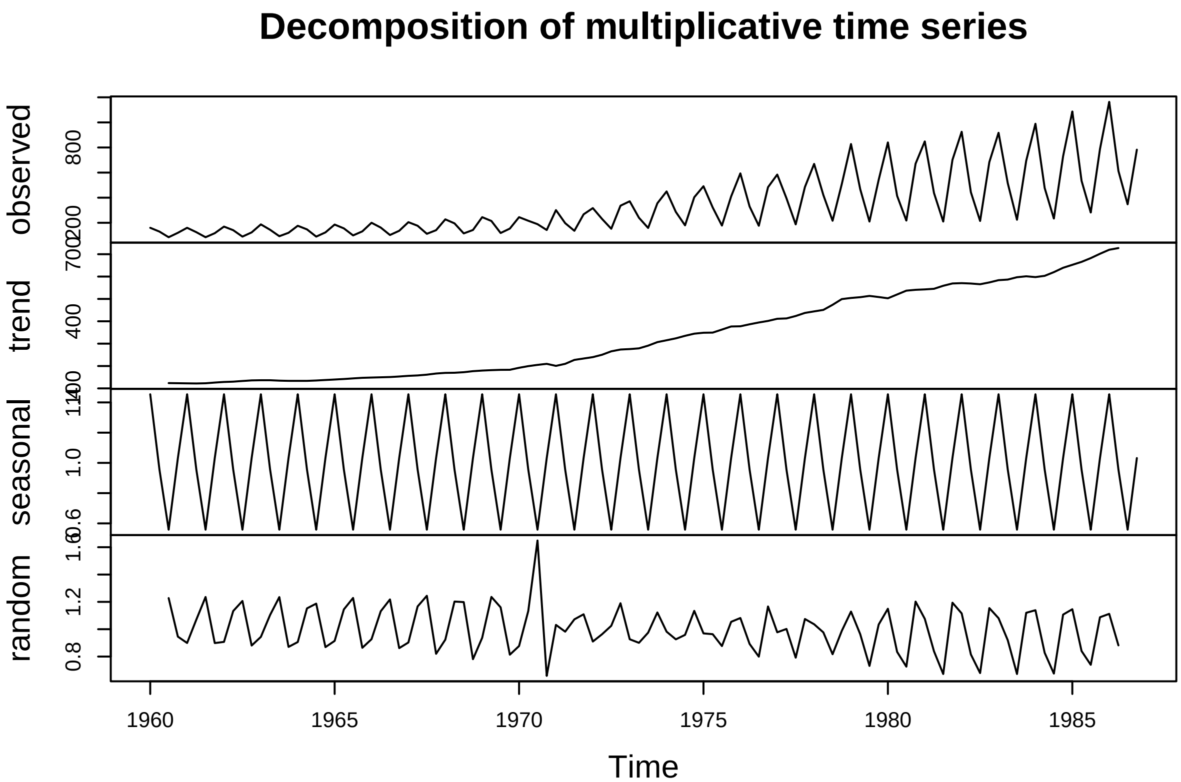
```r
plot(n)
```



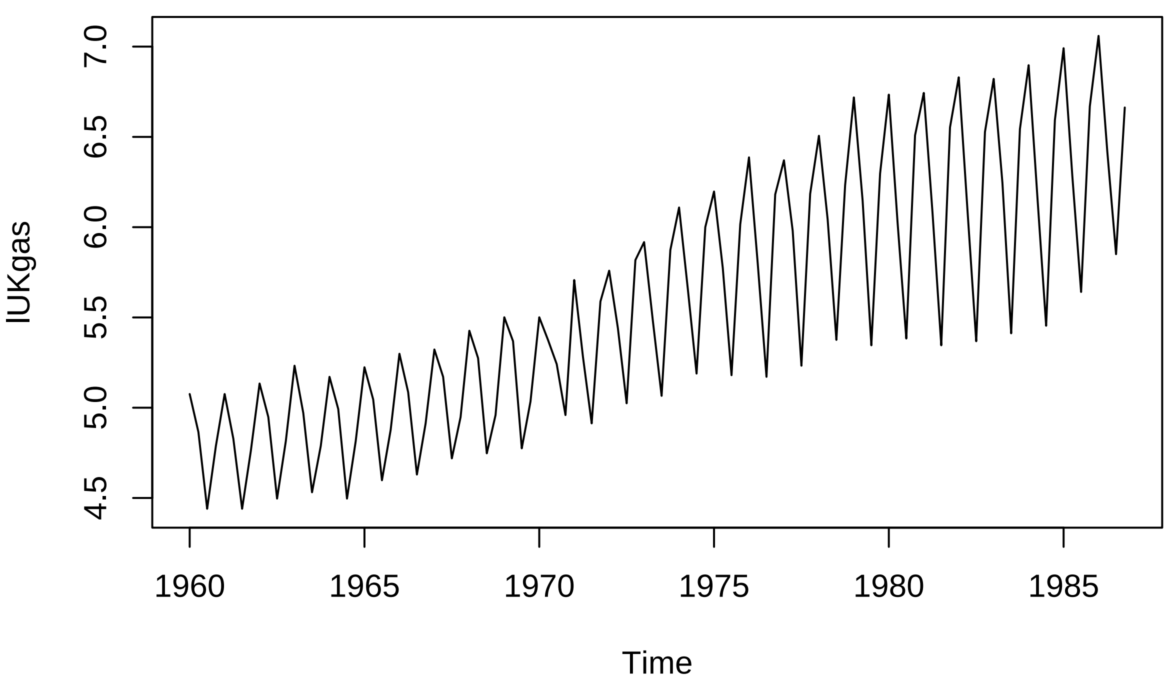

```
plot(decompose(UKgas, type="additive"))
```



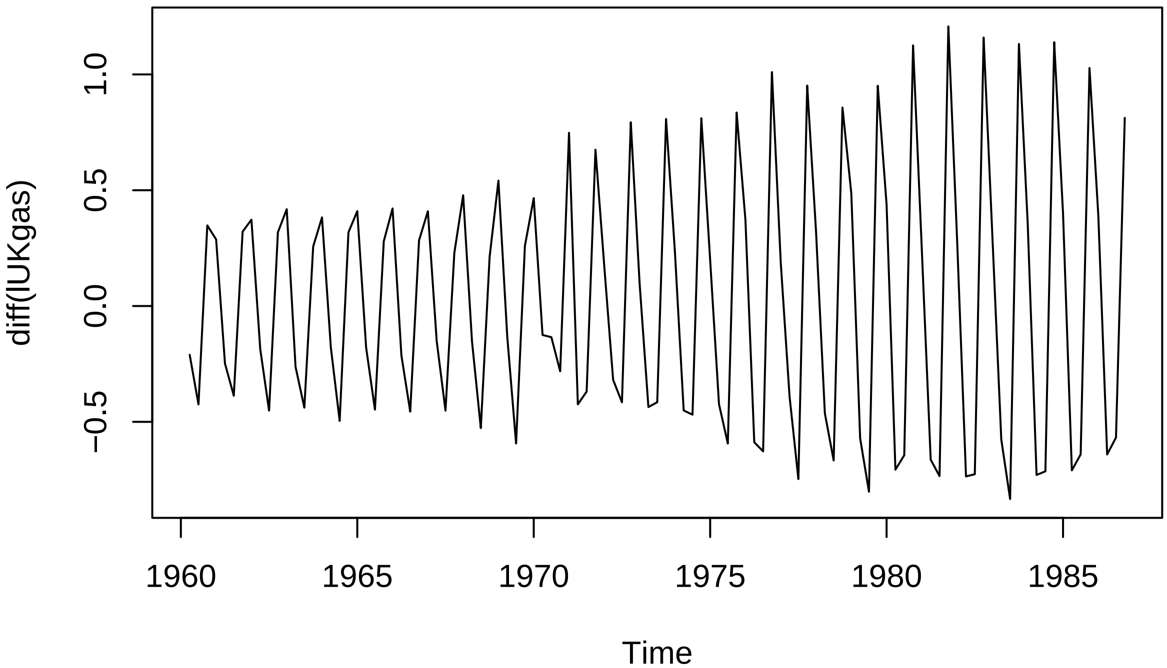
```r
plot(decompose(UKgas, type="multiplicative"))
```



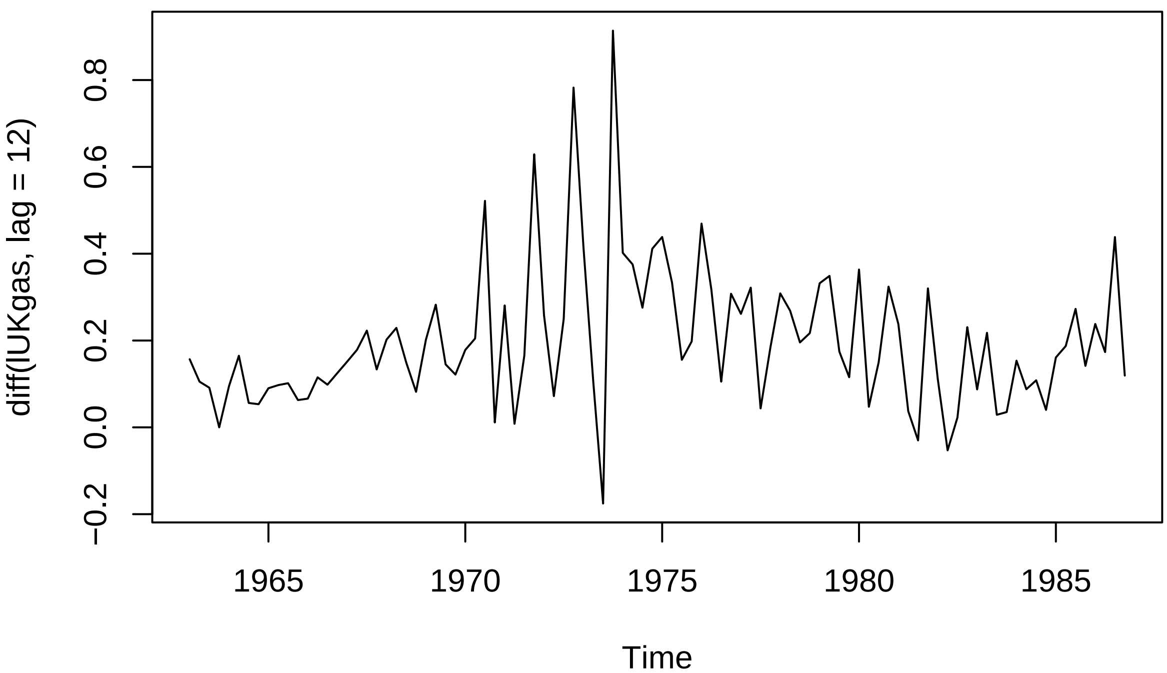

```r
lUKgas=log(UKgas)
plot(lUKgas)
```



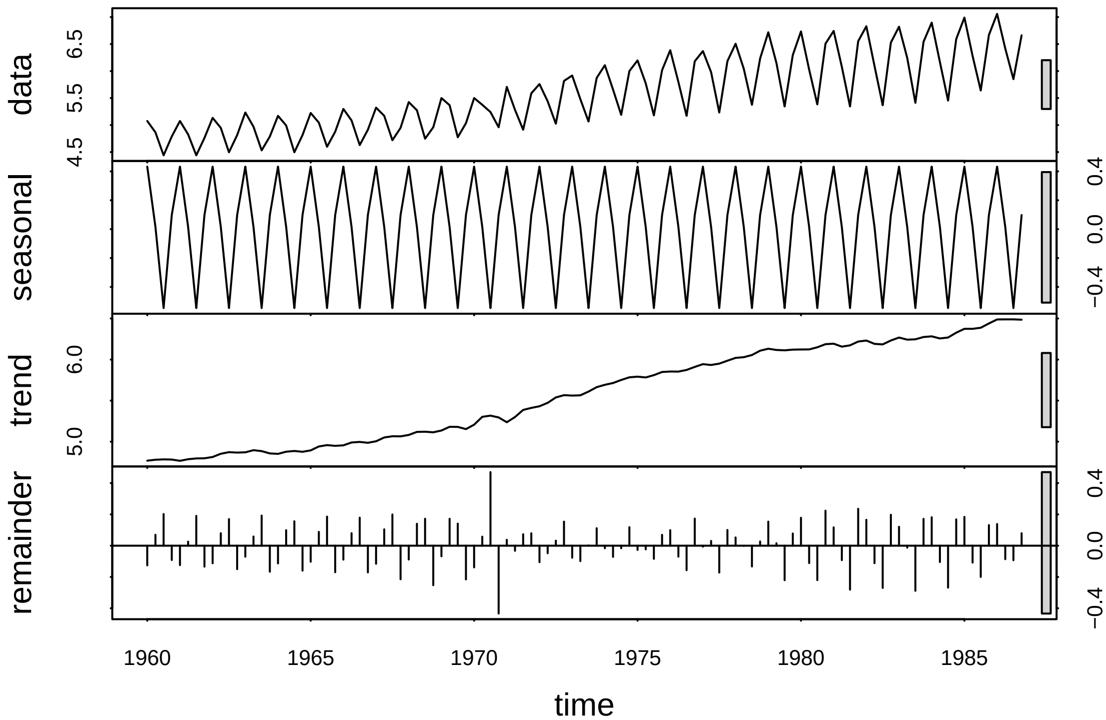
```r
plot(diff(lUKgas))
```



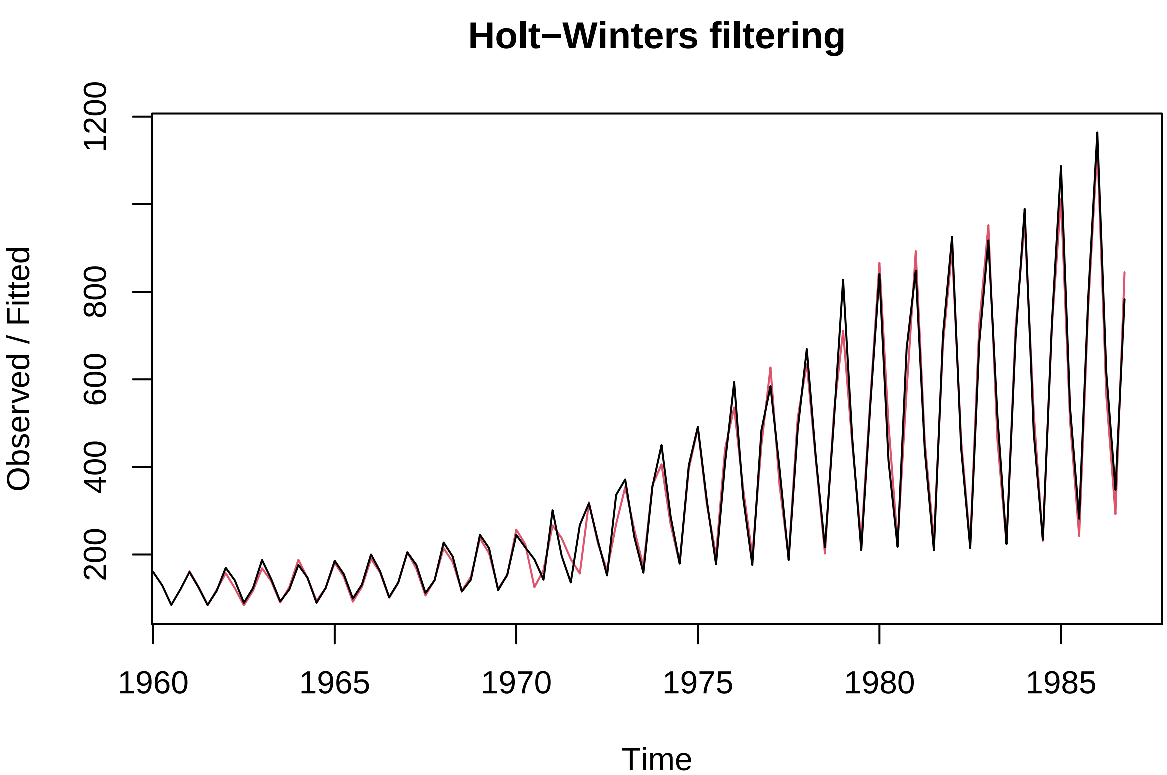

```r
plot(diff(lUKgas, lag=12))
```



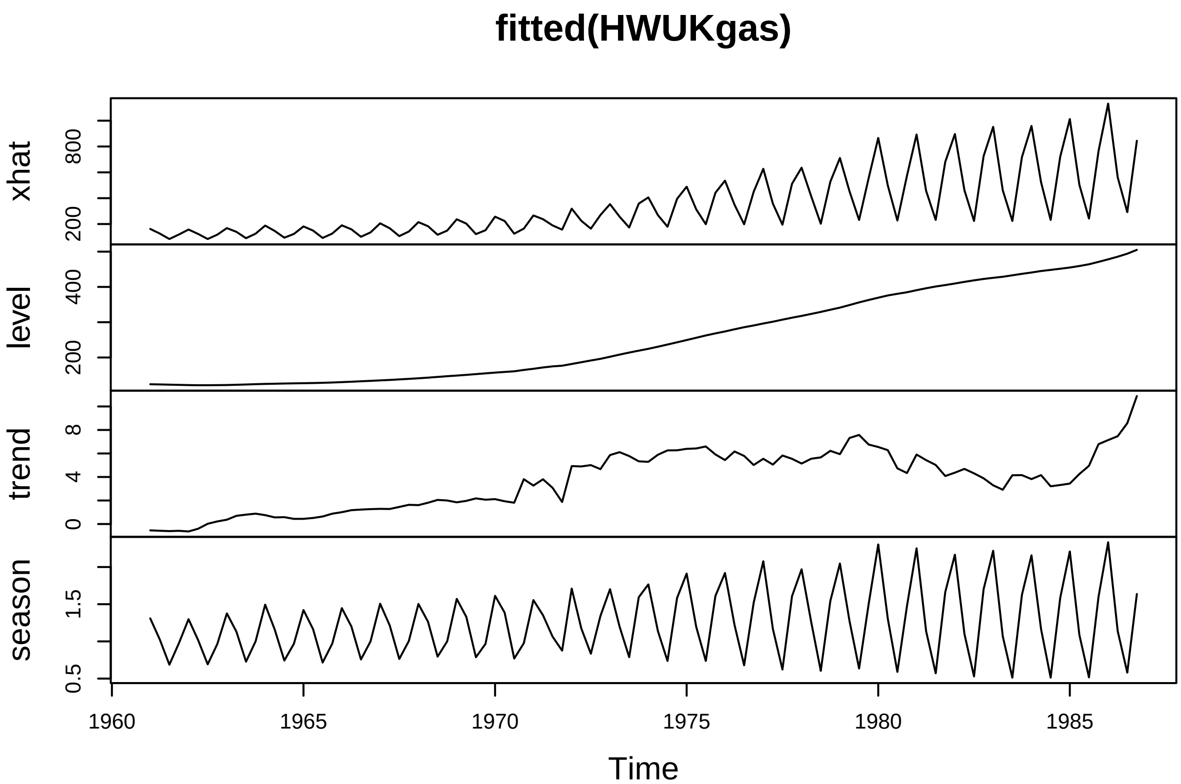

```r
plot(stl(lUKgas, s.window="periodic"))
```




```
# A multiplicative time series decomposition is better suited because the seasonal component increases
# The 12 month lag removes the trend and the seasonality.
# What is unsatisfactory is that the variance increases over time and cannot be stabilized fully. This
HWUKgas <- HoltWinters(UKgas, seasonal = "mult")
plot(HWUKgas)
```

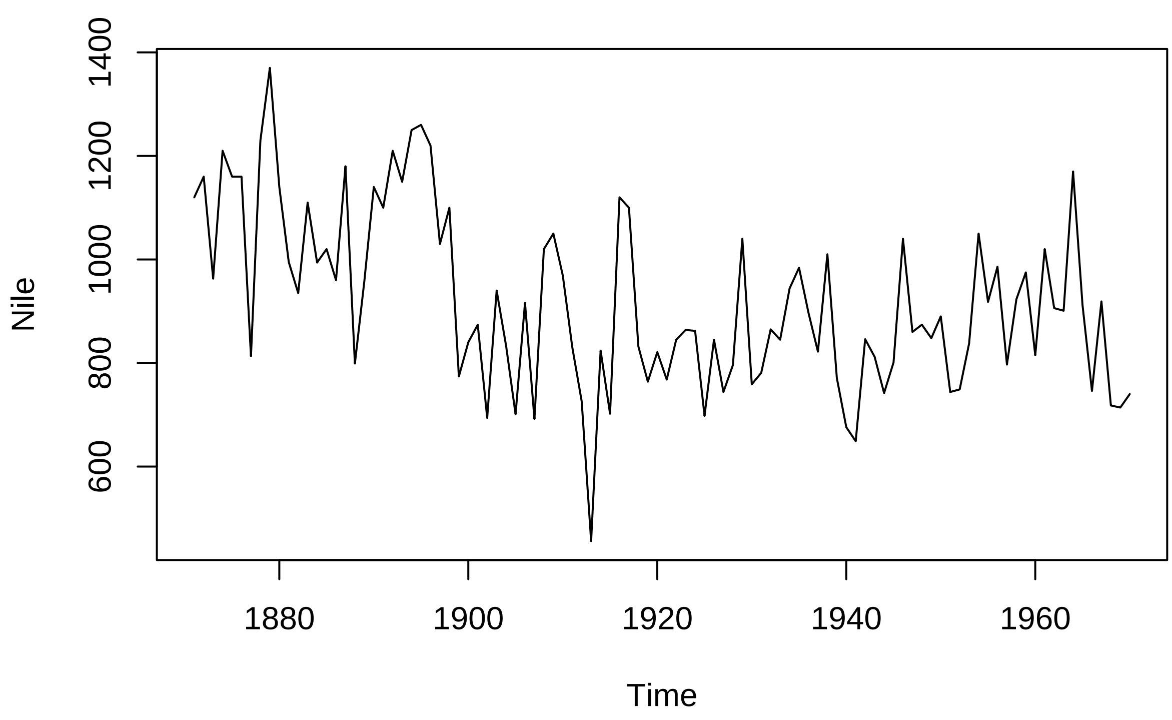


```
plot(fitted(HWUKgas))
```




```
# Using Holt & Winters exponential smoothing with trend and seasonality we identify a seasonal componen
```

```
?Nile
plot(Nile)
```



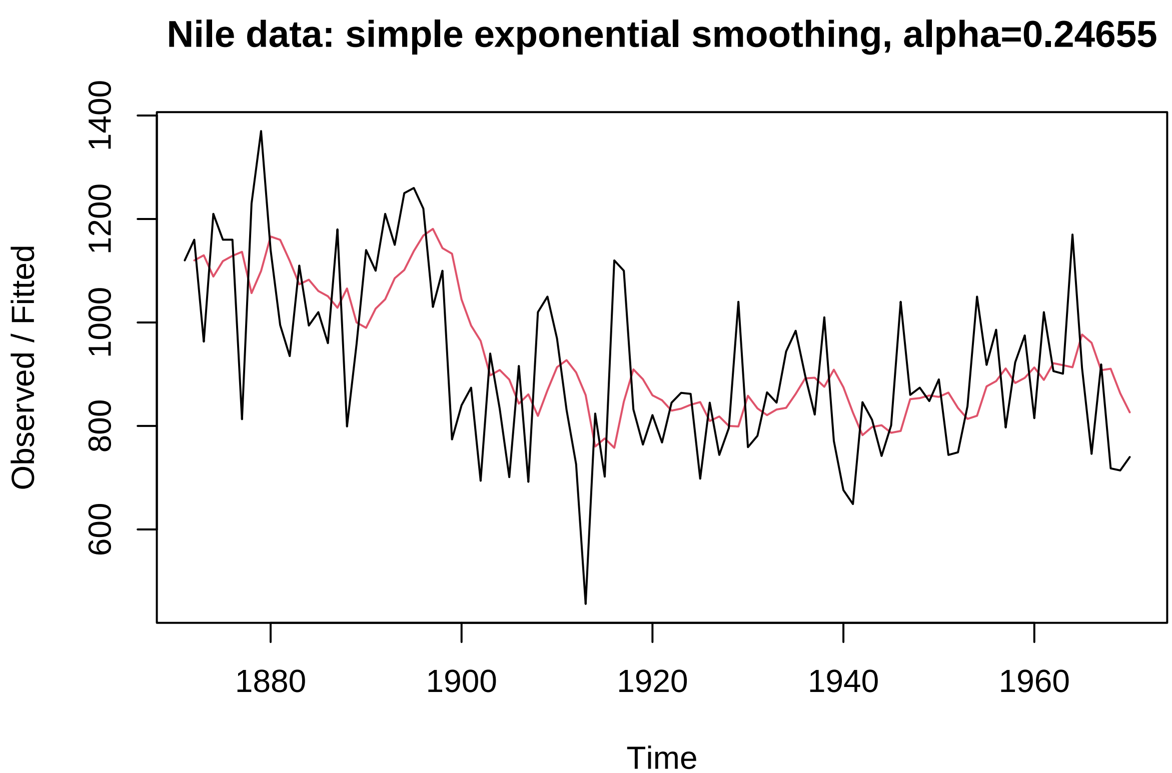
```r
HWNile <- HoltWinters(Nile, beta=F, gamma=F)
HWNile
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = Nile, beta = F, gamma = F)
##
## Smoothing parameters:
##  alpha: 0.2465579
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 805.0389
```

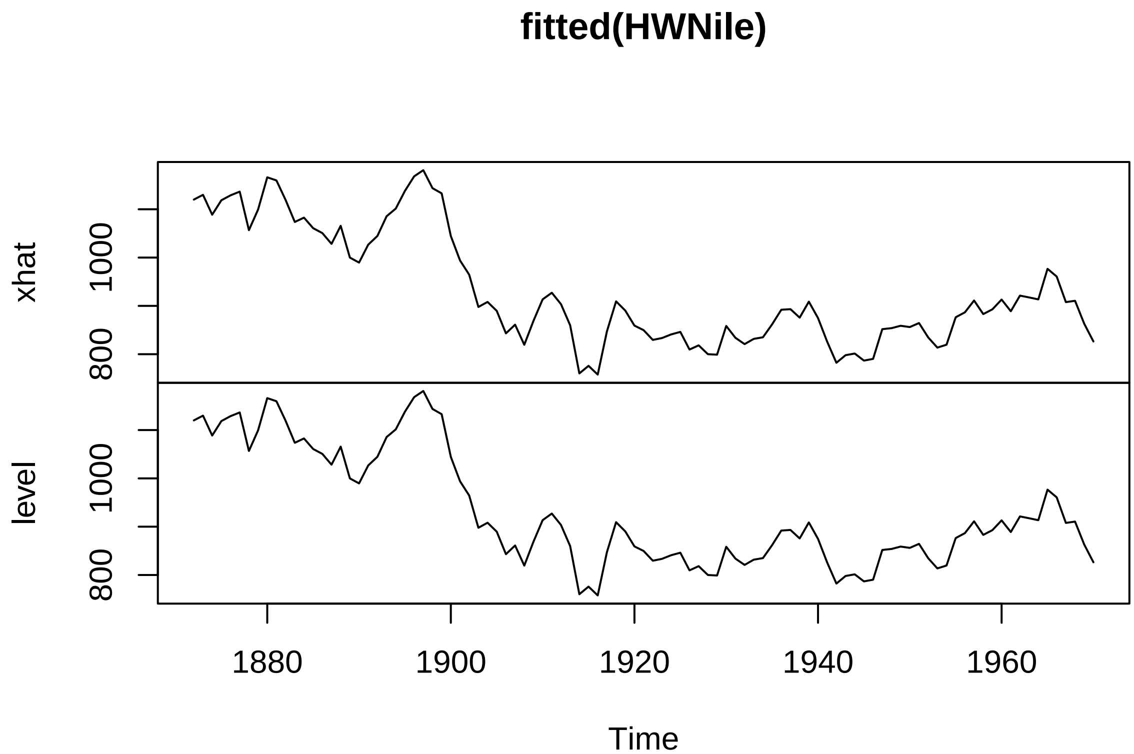
```r
HWNile$alpha
```

```
## [1] 0.2465579
```

```r
plot(HWNile, main="Nile data: simple exponential smoothing, alpha=0.24655")
```

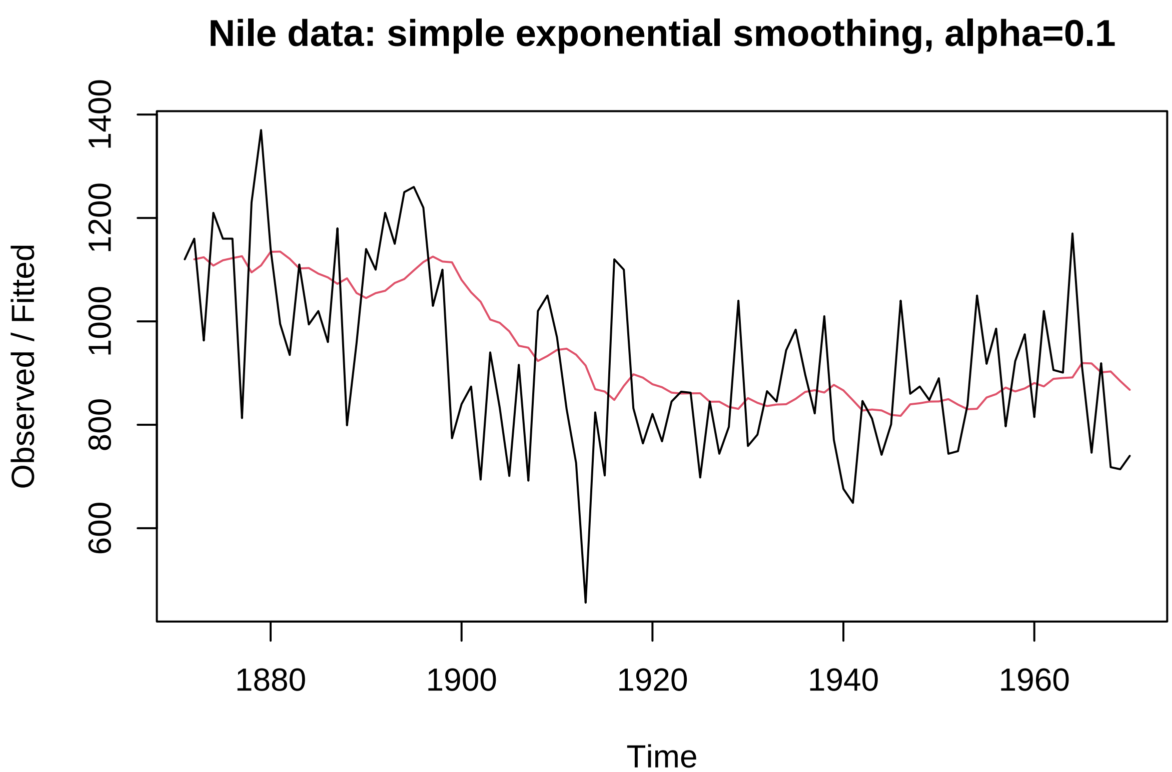



```r
plot(fitted(HWNile))
```

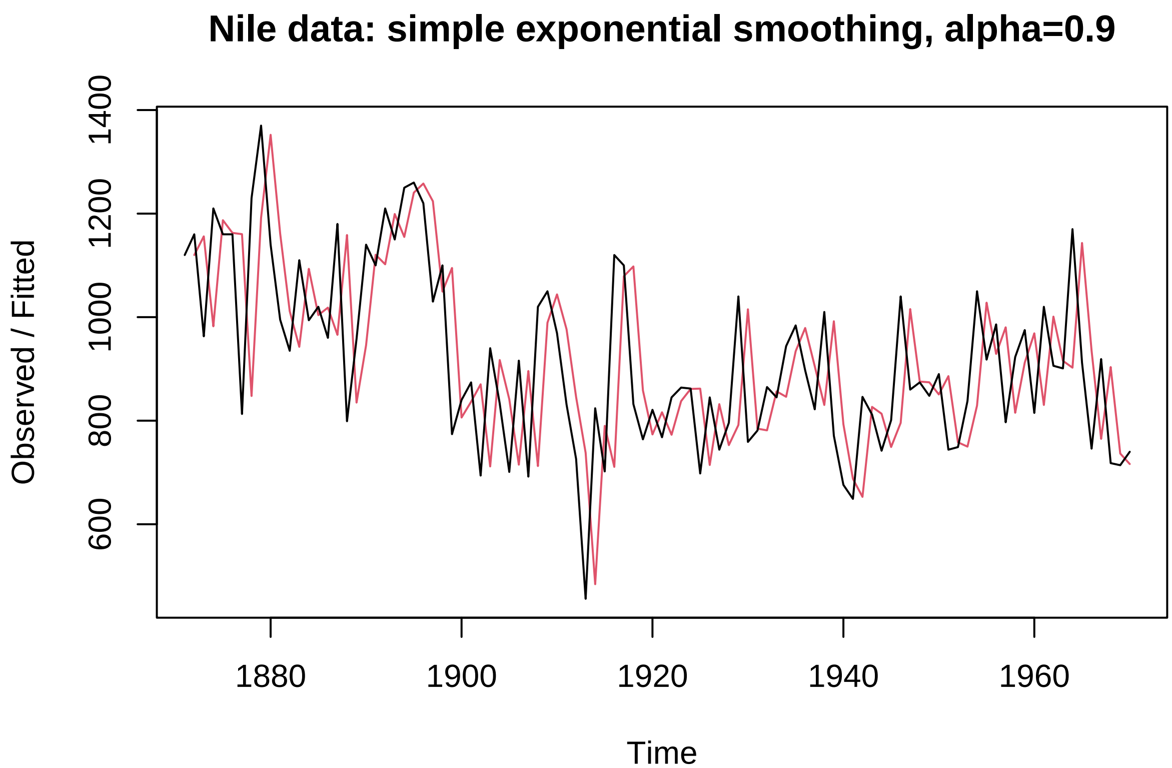



```
# The chosen value of alpha is approximately 0.24655.

HWNile1 <- HoltWinters(Nile, alpha=.1, beta=F, gamma=F)
plot(HWNile1, main="Nile data: simple exponential smoothing, alpha=0.1")
```



```r
HWNile2 <- HoltWinters(Nile, alpha=.9, beta=F, gamma=F)
plot(HWNile2, main="Nile data: simple exponential smoothing, alpha=0.9")
```




```r
# Setting a parameter closer to 0 allocates less weight to the previous year's observation and more to
mape0 = function(Nile, HWNile){ mean(abs(Nile - HWNile)/Nile) }
mape1 = function(Nile, HWNile1){ mean(abs(Nile - HWNile1)/Nile) }
mape2 = function(Nile, HWNile2){ mean(abs(Nile - HWNile2)/Nile) }
```