Homework01

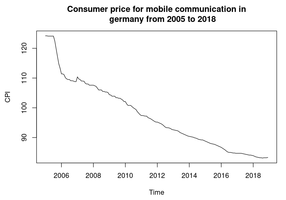
2022-05-06

R Markdown

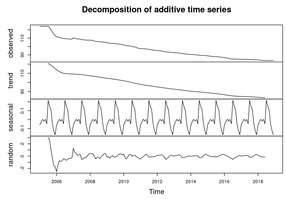
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

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:

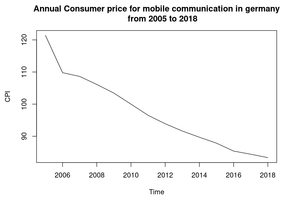
```
library(xlsx)
#(a) read the data and plot the time series
mobile.data <- read.xlsx('statistic-mobile-communications.xlsx',
                         sheetName = 'Data', header=T)
attach(mobile.data)
ts.mobile.data <- ts(mobile.data$CPI, start=c(2005,1), end=c(2018,12), frequency = 12)
plot(ts.mobile.data, main="Consumer price for mobile communication in
     germany from 2005 to 2018", ylab = "CPI")
```



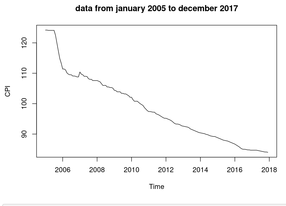

```
#(b) decompose the time series into three parts: estimating trends, seasonal
#effects, and random series.
ts.mobile.data.decompose <- decompose(ts.mobile.data)
plot(ts.mobile.data.decompose)
```




```
#(c)Use the aggregate function to remove any seasonal effects within each
#year and produce an annual series of mean CPI for the period 2005-2018
ts.mobile.annual <- aggregate(ts.mobile.data) / 12
plot(ts.mobile.annual, main ="Annual Consumer price for mobile communication in germany
     from 2005 to 2018", ylab="CPI")
```

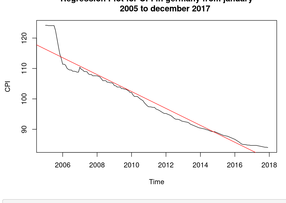



```
#(d) Use the window function to plot the data from January 2005 to December 2017
help("window")
ts.mobile.window <- window(ts.mobile.data, start=c(2005,1), end=c(2017,12))
plot(ts.mobile.window, main="data from january 2005 to december 2017", ylab="CPI")
```

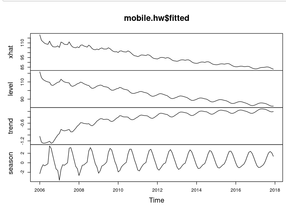



```
#(e) Use the command lm to estimate the parameters `α and `β in the simple
#linear regression model.
#(f) Use commands summary and abline to add lines to existing plots in
#step (e).
linearRegressionModel <- function(data) {
  t <- time(data)
  reg <- lm(data~t)
  print("========coefficients for Linear Regression Model:")
  print(reg)
  print("==========Summary of Linear Regression Model:")
  print(summary(reg))

  plot(data, ylab="CPI", main ="Regression Plot for CPI in germany from january
       2005 to december 2017")
  abline(reg, col="red")
}
#apply regression model
linearRegressionModel(ts.mobile.window)
```

```
## [1] "========coefficients for Linear Regression Model:"
##
## Call:
## lm(formula = data ~ t)
##
## Coefficients:
## (Intercept)            t
##    5694.604      -2.782
##
## [1] "==========Summary of Linear Regression Model:"
##
## Call:
## lm(formula = data ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9260 -1.4874 -0.6613  0.6117  9.1555
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5694.60420  105.10543   54.18   <2e-16 ***
## t             -2.78218    0.05225  -53.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.449 on 154 degrees of freedom
## Multiple R-squared:  0.9485, Adjusted R-squared:  0.9481
## F-statistic:  2835 on 1 and 154 DF,  p-value: < 2.2e-16
```




```
#######################
#PREDICTION BY HAND (g)
#######################

#####################JUNE 2018##################
#e_hat(2018) = a + bt = 5694.604 -2.782 * (2018+5/12) = 34,369

#####################JUNE 2019##################
#y_hat(2019) = a + bt = 5694.604 -2.782 * (2019+5/12) = 31,587

#####################JUNE 2020##################
#y_hat(2020) = a + bt = 5694.604 -2.782 * (2020+5/12) = 28,805

#(h)

#Holt-Winters
mobile.hw <- HoltWinters(ts.mobile.window)
plot(mobile.hw$fitted)
```




```
summary(mobile.hw)
```

```
##               Length Class  Mode
## fitted           576   mts  numeric
## x                156   ts   numeric
## alpha              1   -none- numeric
## beta               1   -none- numeric
## gamma              1   -none- numeric
## coefficients  14   -none- numeric
## seasonal         1   -none- character
## SSE              1   -none- numeric
## call             2   -none- call
```

```
#(j) make predictions using holt-winters
predict(mobile.hw, n.ahead = 36)
```

```
##             Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2018 82.73340 81.70623 81.06730 80.85304 80.84591 80.93902 81.32052 82.00581
## 2019 80.58376 79.55659 78.91766 78.70339 78.69626 78.78958 79.17088 79.85617
## 2020 78.43411 77.40695 76.76801 76.55375 76.54662 76.63994 77.02124 77.70653
##             Sep      Oct      Nov      Dec
## 2018 82.67596 82.04044 82.66783 81.85636
## 2019 80.52631 80.79079 80.51819 79.70071
## 2020 78.37667 78.64115 78.36854 77.55107
```