

DTU - Time Series Analysis - Assignment 1: Forecasting diesel consumption

Emma Demarecaux (s176437)

01/03/2018

Introduction

In Denmark, diesel engines have dominated the market for heavy vehicles. It has also been cost effective to have a diesel car if your annual milage is sufficiently high. This assignment focuses on predicting the annual consumption of diesel in Denmark. The data for this study comes from <www.statistikbanken.dk> and is part of the Environmental-Economic Accounts.

The set of data that has been used is composed by:

- The year that the observation is from (from 1966 to 2016);
- The consumption of diesel in tonnes.

Here is a summary of the consumption of diesel in tonnes from 1966 to 2016:

```
summary(data$Diesel)
```

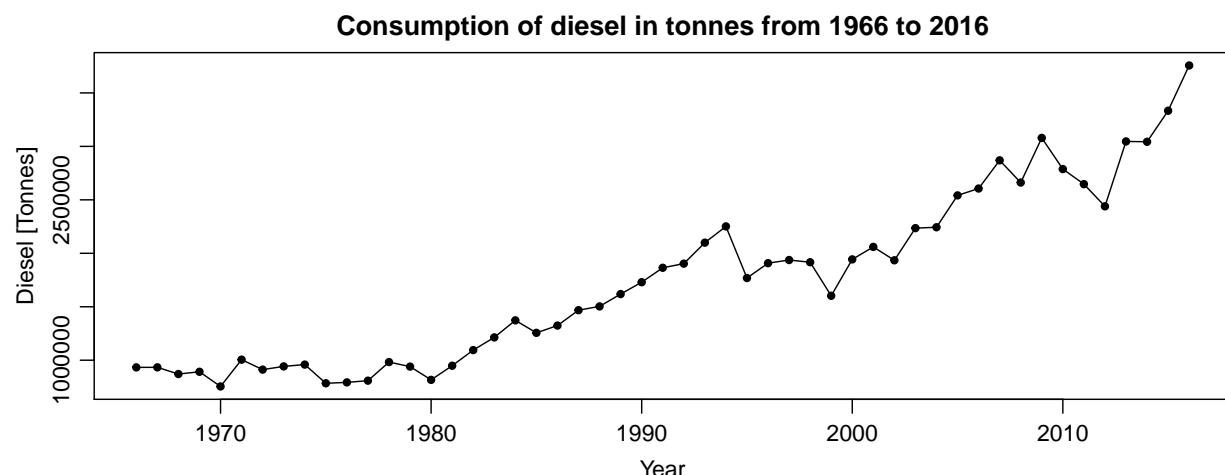
```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 754425  953821 1729586 1751635 2247470 3756795
```

We are going to use GLM model, simple exponential smoothing and local linear trend model to estimate the consumption of diesel in tonnes from 1966 through 2013 and predict the consumption for the years 2014, 2015 and 2016. In order to achieve this, we first create a smaller dataset in which the observations from 2014 through 2016 are removed so that we can compare the predictions from our different models with the real observations.

The study has been elaborated in colaboration with Adélie Marie Kookai Barre (s170075).

Question 1

In the following figure is represented the consumption of diesel in tonnes from 1966 to 2016:



For now, we can observe an increasing trend with major peaks around 1995 and 2009.

Question 2

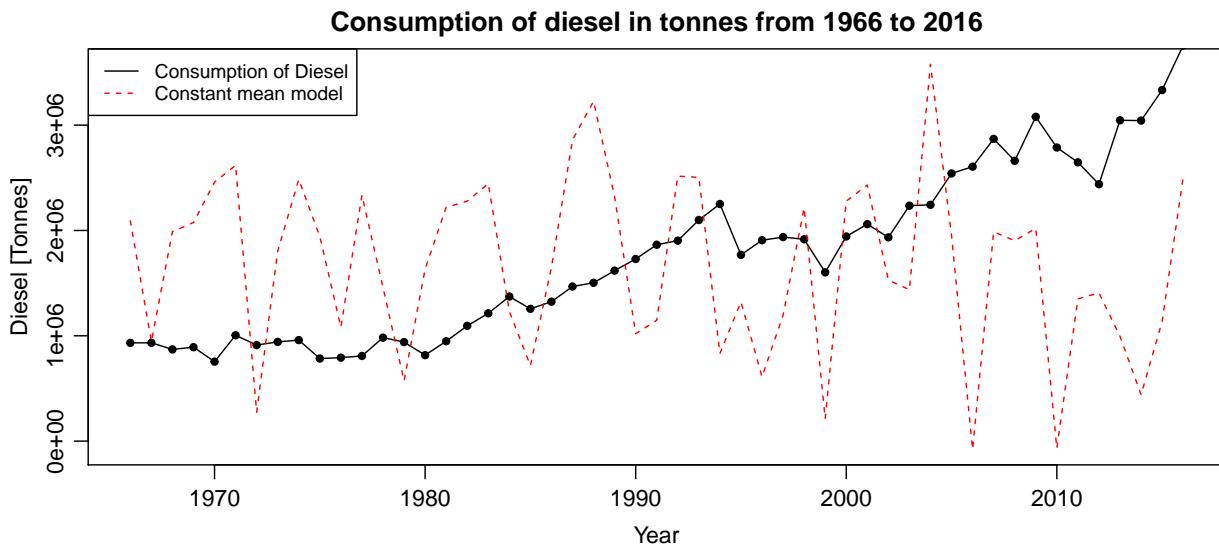
The global mean value and standard deviation of the diesel consumption are represented in the following table:

Data	Mean	Standard deviation
Diesel consumption	1751635	799109.6

Let us then consider the constant mean model (formula 3.10 p33 from the book *Time Series Analysis*): $Y_t = \mu + \epsilon_t$ with

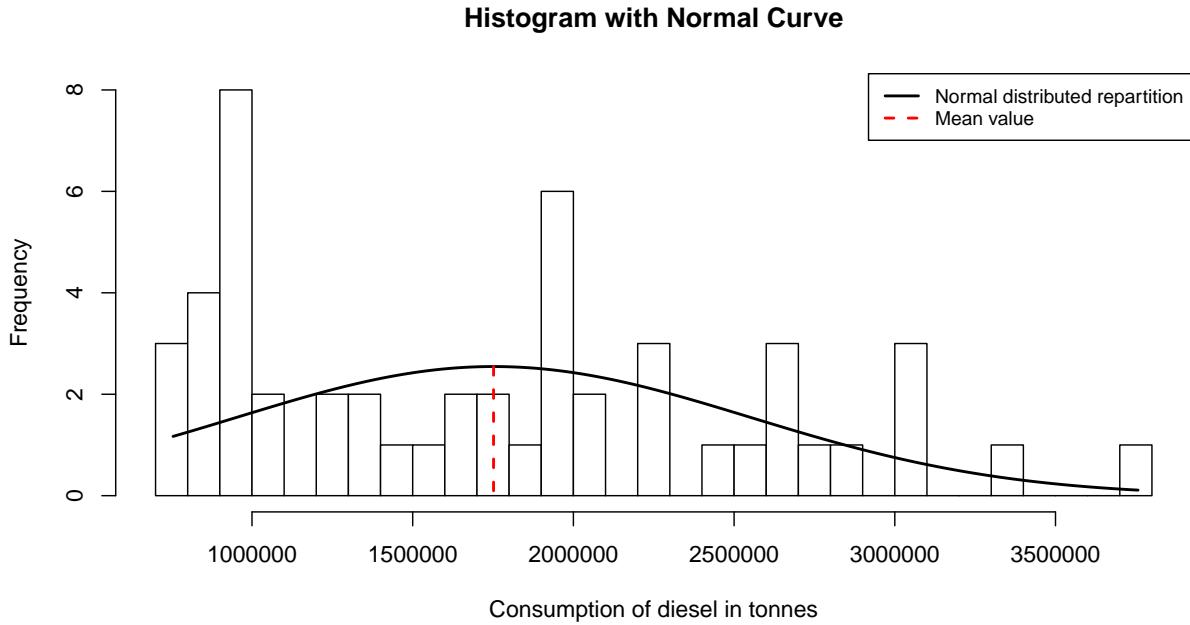
$$\begin{cases} \mu = \text{Mean}(data\$Diesel) \\ \epsilon_t \sim \mathcal{N}(0, \sigma^2) \\ \sigma^2 = \text{Var}(data\$Diesel) \end{cases}$$

and represent this model along with the real data:



This constant mean model doesn't give a good information on the data since we said in the previous question that there is a global trend increasing that we need to estimate. To emphasize this fact, let us take a look at the gaussian distribution that the dataset should follow if the global mean value and standard deviation of the diesel consumption could give a reasonable representation of the data. We must take care of converting the density into a frequency with the following line:

```
yfit <- yfit*diff(h$mids[1:2])*N
```



Hence, this model doesn't fit well the real data.

Question 3

Let us consider now the dataset without the three last observations. We are going to formulate a GLM model in form of a simple linear regression model for all the new dataset. The general linear model (GLM) is a regression model such that (3.9 p33 from the book): $Y_t = x_t^T \theta + \epsilon_t$ where $x_t = (x_{1t}, \dots, x_{pt})^T$ is a known vector and $\theta = (\theta_1, \dots, \theta_p)^T$ are the unknown parameters that we are going to estimate. ϵ_t is a random variable with mean 0 and covariance $\text{Cov}(\epsilon_i \epsilon_j) = \sigma^2 \Sigma_{ij}$.

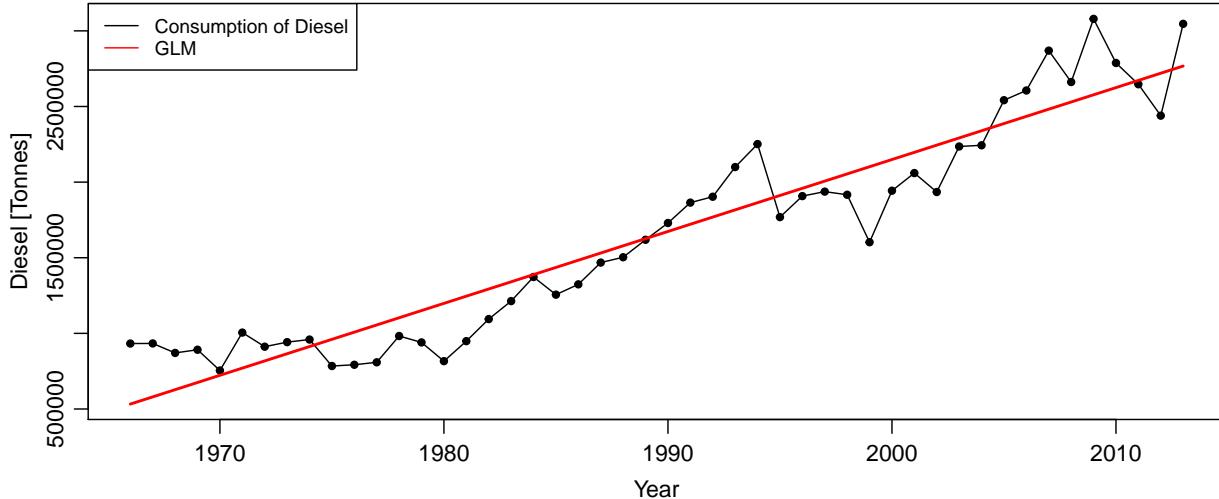
For the simple linear regression model we are only going to estimate two parameters, θ_1 and θ_2 , such that:

$$\begin{pmatrix} Y_1 \\ \vdots \\ Y_{N_{\text{estim}}} \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_{N_{\text{estim}}} \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_{N_{\text{estim}}} \end{pmatrix}$$

In our case, the vector Y represents diesel consumptions and $x_1, \dots, x_{N_{\text{estim}}}$ are the years from 1966 to 2013.

For this part, we use the function `lm()` from R to estimate the least squares estimators θ_1 and θ_2 . We can observe the results in the following figure:

General linear model (simple linear regression)



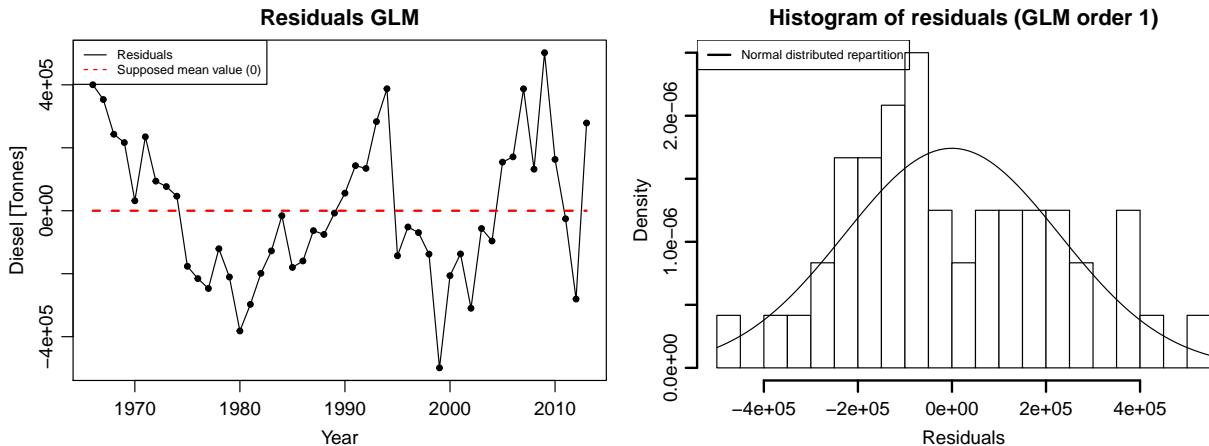
The variance of the model errors is estimated as (slide 6 from lecture 2):

$$\hat{\sigma}^2 = \frac{\epsilon^T \epsilon}{48 - 2}$$

with $N_{\text{estim}} = 48$.

The results for the estimated coefficients are:

Intercept	data_estim\$Year	$\hat{\sigma}$
-92952621.08	47550.95	229035.8

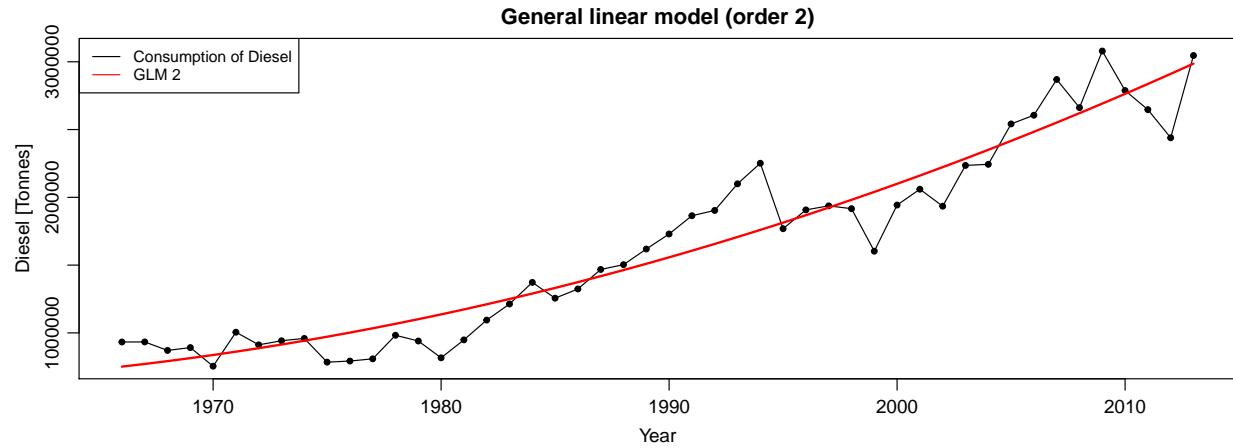


The residuals in the last figure don't seem to have a random behaviour. That means this model isn't accurate enough and therefore isn't useful for making predictions of the future diesel consumption.

We can try to estimate one more parameter, to get a global linear model with a higher order, using the LS estimator given by (3.35 p36 of the book) : $\hat{\theta} = (x^T x)^{-1} x^T Y$ with

$$x^T = \begin{pmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{N_{\text{estim}}} & x_{N_{\text{estim}}}^2 \end{pmatrix}$$

The results are given in the following figure:

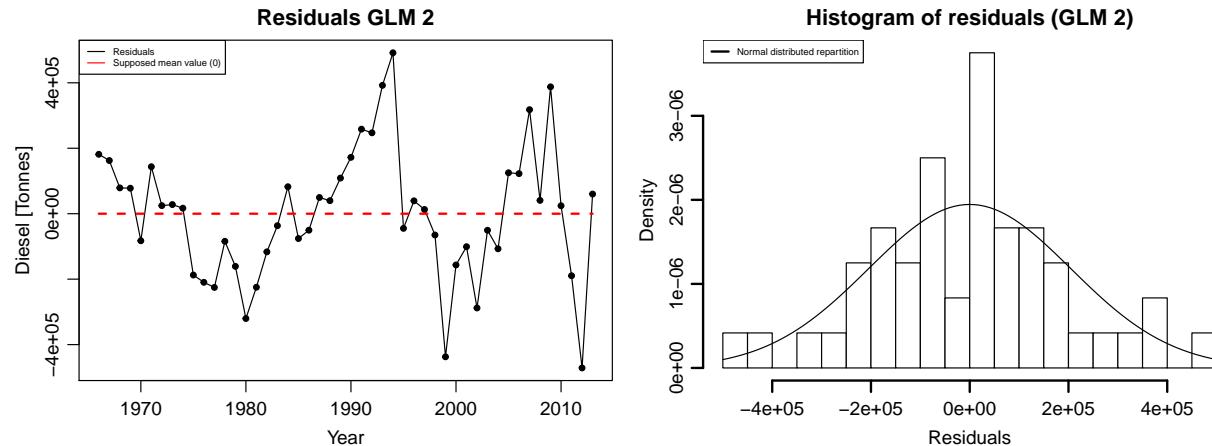


The variance of the model errors is estimated as:

$$\hat{\sigma}^2 = \frac{\epsilon^T \epsilon}{48 - 3}$$

The estimated coefficients are:

Intercept	data_estim\$Year	data_estim\$Year ²	$\hat{\sigma}$
2986202.8745	76084.2598	607.0916	205071.7



Again, the residuals in the last figure don't seem to have a random behaviour. Hence, the GLM isn't useful for making predictions of the future diesel consumption.

Question 4

Now we will apply methods which considers data more locally. We are going to use simple exponential smoothing to predict the diesel consumption. A simple exponential smoothing is a sequence $\hat{\mu}_N$ defined as (3.76 p51 from the book)

$$\hat{\mu}_N = (1 - \lambda)Y_N + \lambda\hat{\mu}_{N-1}$$

In this part, we consider $\lambda = 0.8$. We use this sequence as a prediction model $\hat{Y}_{N+l|N} = \hat{\mu}_N$, and we can update our predictions with the new observations:

$$\hat{Y}_{N+1+l|N+1} = (1 - \lambda)Y_{N+1} + \lambda\hat{Y}_{N+l|N}$$

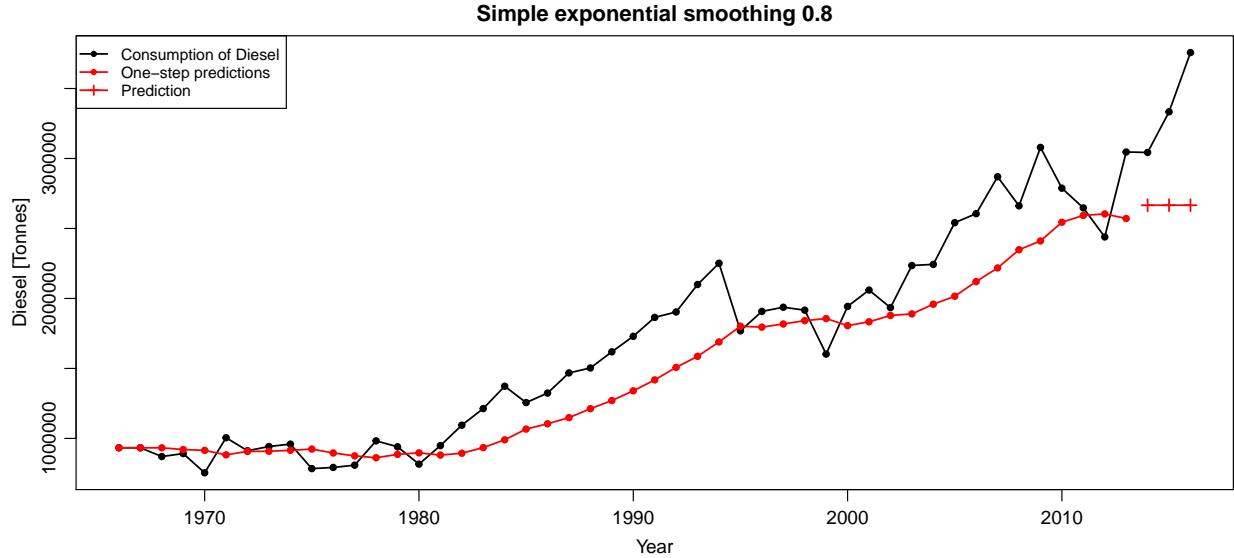
For N large, we have:

$$\hat{\mu}_{N+1} = (1 - \lambda)Y_{N+1} + \lambda\hat{\mu}_N$$

With this model, we can compute $\hat{\mu}$ and then the corresponding one-step predictions for all N_{estim} with an iterative process. We can see the results of exponential smoothing in the following table:

Year	2014	2015	2016
Prediction [Tonnes]	2666069.0	2666069.0	2666069.0

The data, the corresponding one-step predictions for all N_{estim} and the predictions for the three years that were left out are presented in the following figure:



We use $\lambda = 0.8$ which means that we remember a lot of things from the past. It may explain why the red curve is that smooth. The one-step predictions aren't far from the real values, however a constant prediction over the years seems to be irrelevant and in this case because it can't predict the new trend of the new observations.

Question 5

We are going to use a local linear trend model to predict the diesel consumption. Again we first use $\lambda = 0.8$. The idea is that we forget old observations in an exponential manner (slide 13 from lecture 4):

$$\hat{\theta}_N = \operatorname{argmin}_{\theta} \sum_{j=0}^{N-1} \lambda^j [Y_{N-j} - f^T(-j)\theta]^2$$

The criterion that we want to minimize is a weighted LS criterion, that means $\epsilon_t \sim \mathcal{N}(0, \sigma^2 \Sigma)$ with

$$\Sigma = \begin{pmatrix} \frac{1}{\lambda^{N-1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\lambda^{N-2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Hence, the WLS solution is such that $\hat{\theta}_N = (x_N^T \Sigma^{-1} x_N)^{-1} (x_N^T \Sigma^{-1} Y)$ with

$$\begin{cases} \hat{\theta}_N = F_N^{-1} h_N \\ F_N = \sum_{j=0}^{N-1} \lambda^j f(-j) f^T(-j) \\ h_N = \sum_{j=0}^{N-1} \lambda^j f(-j) Y_{N-j} \end{cases}$$

We can calculate the one-step predictions with an iterative process by updating the estimates when Y_{N-1} is available:

$$\begin{cases} F_{N+1} = F_N \lambda^N f(-N) f^T(-N) \\ h_{N+1} = \lambda L^{-1} h_N + f(0) Y_{N+1} \\ \hat{\theta}_{N+1} = F_{N+1}^{-1} h_{N+1} \end{cases}$$

with $L = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$

As initial values we can use $h_0 = 0$ and $F_0 = 0$. In order to build this model, we first initialize the matrices F and h using a burning period $n = 5$, which means that the first n predictions are ignored to stabilize the estimator. Once we get F_n , h_n and $\hat{\theta}_n$, we compute an estimation of the current observation (which will allow us to estimate σ^2 locally), a one-step prediction and then we update the matrices with the new observation until we reach the year 2013. We can then make a prediction for the years 2014, 2015 and 2016.

To use the local linear trend model, we need to define the total memory (slide 17 from lecture 4): $T = \frac{1-\lambda^N}{1-\lambda}$.

We are going to compare two different estimators for σ^2 :

- the local estimator is given as:

$$\hat{\sigma}_l^2 = \frac{1}{T-p} (Y - x_N \hat{\theta}_N)^T \Sigma^{-1} (Y - x_N \hat{\theta}_N) \quad T > p$$

With this estimator, we only look at the last observations and make a fit. In this situation we only have one model ($\hat{\theta}_N$) and look at the predictions errors.

- there is also a global estimator given as:

$$\hat{\sigma}_g^2 = \frac{1}{N-n} \sum_{j=n+1}^N \frac{(Y_j - \hat{Y}_{j|j-1})^2}{1 + f^T(1) F_{j-1}^{-1} f(1)}$$

With this estimator, every one-step prediction has the same weight.

In this case, the local estimator will be computed differently. In fact, as we use a different $\hat{\theta}$ for each estimation, we don't have one model for all the estimations, but a different model for each one of them. Therefore, the estimation error for each step is already weighted with the right power of λ . Hence, we compute the local estimator with the following line:

```
sigmal2 <- 1/(T-2)*sum((data_estim$Diesel-Diesel_est)^2)

#data_estim$Diesel -> real data
#Diesel_est -> one-step estimations
```

We found:

```
sigmal #Local Estimator
```

```
## [1] 492076
```

```
sigmag #Global Estimator
```

```
## [1] 170937.1
```

that we use to compute the 95% confidence interval given as:

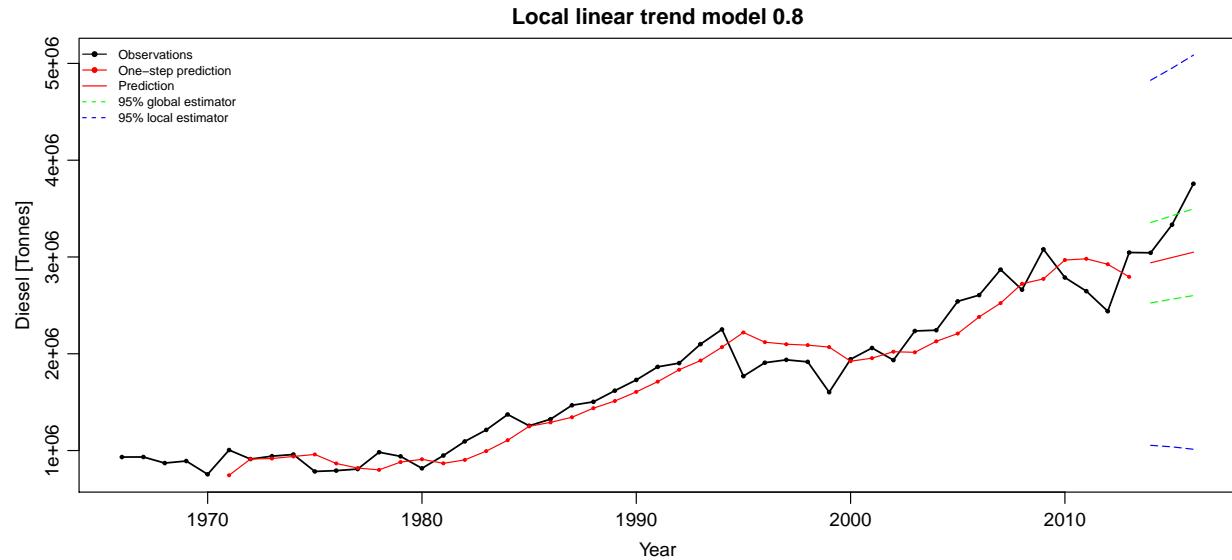
- $\hat{Y}_{N+l|N} \pm t_{(0.0025)}(T - 2)\sigma_l \sqrt{1 + f^T(l)F_N^{-1}f(l)}$ for the local estimator of σ
- $\hat{Y}_{N+l|N} \pm t_{(0.0025)}(N - n)\sigma_g \sqrt{1 + f^T(l)F_N^{-1}f(l)}$ for the global estimator of σ

We can see the predictions for the three years that were left out along with the predictions intervals in the following table:

```
Pred
```

##	Year	pred	lower_loc	upper_loc	lower_glob	upper_glob
## 1	2014	2940026	1053910	4826141	2524840	3355211
## 2	2015	2994821	1038331	4951311	2564144	3425497
## 3	2016	3049616	1013088	5086145	2601321	3497911

The data along with the corresponding one-step predictions, the predictions for the three years that were left out and the predictions intervals are presented in the following figure:



This model with $\lambda = 0.8$ seems better than the simple exponential smoothing model with the same parameter as it takes into account the trend of this time series. However, we may have expected a better interval with the local estimator than with the global one. Maybe we observe the opposite situation because we used a different $\hat{\theta}$ for each estimation and we don't have only one model ($\hat{\theta}_N$).

Question 6

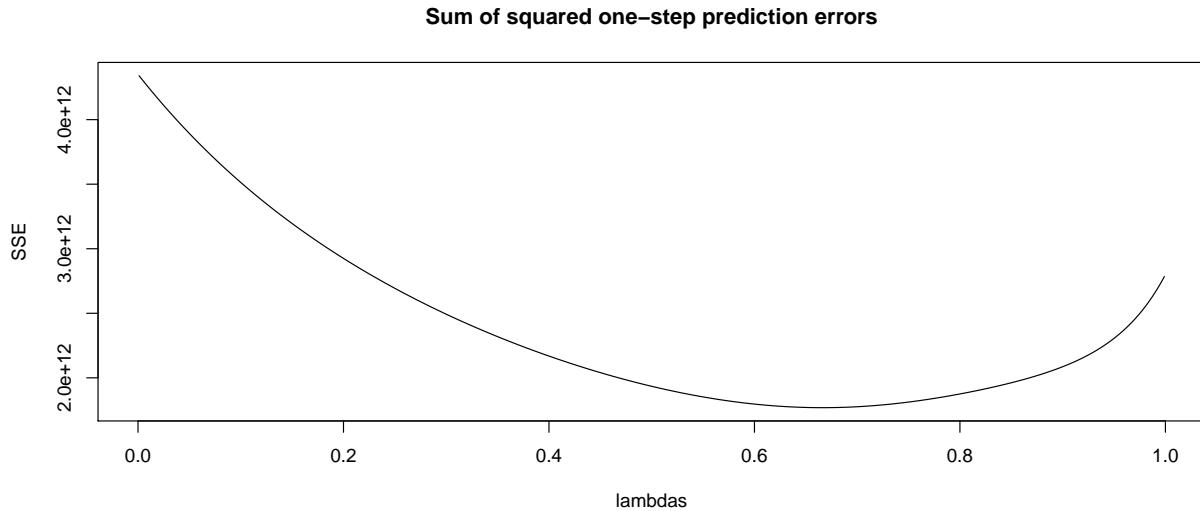
We have seen that the local linear trend model was better than the simple exponential smoothing model, however is the λ parameter optimal ?

We are going to find the optimal value of the forgetting factor which corresponds to the parameter that gives the best one-step predictions. We are still using the same burning period $n = 5$ and considering all the observations until the year 2013. We choose the optimal λ so that the sum of squared one-step predictions errors is minimized:

$$SSE = \sum_{i=n+1}^{N_{2013}} (Y_i - \hat{Y}_{i|i-1})^2$$

Hence we compute the SSE for 999 values of the forgetting factor:

```
lambdas <- seq(0.001, 0.999, 0.001)
```



The result is shown in the following line:

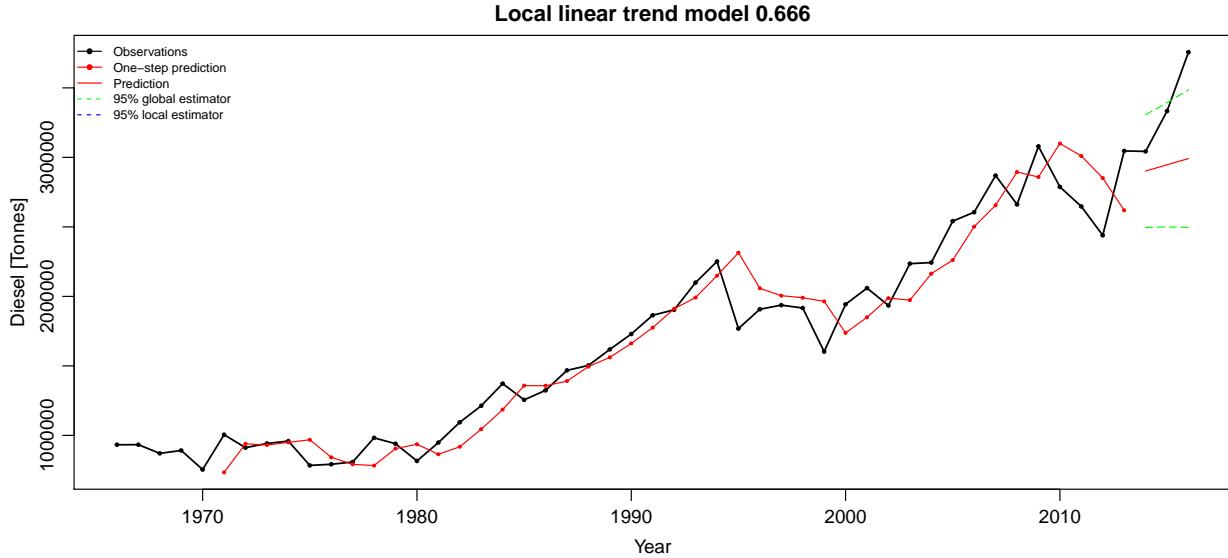
```
lambda_min
```

```
## [1] 0.666
```

We can then compute again the one-step predictions, the predictions for the three last years that were left out and the predictions intervals with $\lambda = 0.666$:

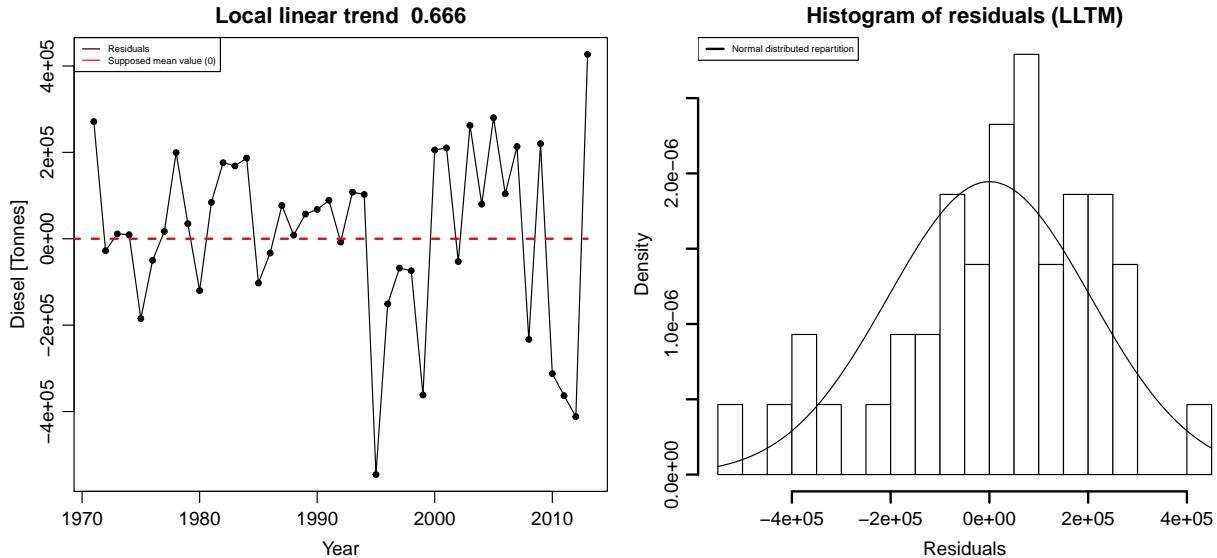
```
Pred
```

```
##   Year     pred lower_loc upper_loc lower_glob upper_glob
## 1 2014 2901764 -7274043 13077571   2496009   3307520
## 2 2015 2946662 -8260561 14153884   2499779   3393544
## 3 2016 2991559 -9416858 15399975   2496779   3486338
```



We didn't show the predictions intervals with the local estimator of the standard deviation because they are very large. This might raise some questions such as, why the "local" interval for $\lambda = 0.666$ is much larger than the one for $\lambda = 0.8$? Maybe it's because of the way we computed it as we have already explained before.

We can observe the residuals of the last model:



We are now starting to see a random behaviour of the residuals that show we are on the right path to make a very good prediction.

The one-step predictions with the optimal forgetting factor doesn't seem really different than the one with $\lambda = 0.8$. It may be because the values of the diesel consumption are quite high. However, the predictions intervals with the global estimator of the standard deviation seems more accurate with the optimal λ .

Question 7

Clearly, the global mean and the simple linear regression are not relevant for this study. The simple exponential smoothing is not relevant either, maybe the double exponential smoothing or the Holt-Winters procedure

could have been better for these predictions. The local trend model gives the best predictions among those that we used for this study.

To improve the predictions, other types of model could be used such as the ARMA and ARIMA models or the SARIMA model (which takes into account the seasonal variations).

Appendix

```

## This empties the work space.
rm(list=ls())
library(ggplot2)
## change directory
## setwd("/path/to/work_dir")
setwd("~/Travail/DTU/Time_series/Ass1")

## reading data, regarding the first lines in the file as names:
data <- read.table(file="A1_diesel.txt", sep="\t", header=TRUE)
head(data)
names(data) <- c("Year", "Diesel")

#Dimension of the dataset
N <- length(data$Year)

#I create a new dataset without the last three years
data_estim <- data.frame(data$Year[1:48], data$Diesel[1:48])
names(data_estim) <- c('Year', 'Diesel')
data_comp <- data.frame(data$Year[49:51], data$Diesel[49:51])
names(data_comp) <- c('Year', 'Diesel')

#Length of the new dataset
N_estim <- length(data_estim$Year)

#####Question 1#####
par(mfrow=c(1,1))
par(mgp=c(2, 0.7, 0), mar=c(3,3,2,1))
plot(data$Year, data$Diesel, xlab='Year', ylab="Diesel [Tonnes]", type='o', pch=20,
      main='Consumption of diesel in tonnes from 1966 to 2016')

#####Question 2#####

#Computation of the mean and the standard deviation
mean <- mean(data$Diesel)
sd <- sd(data$Diesel)

#Constant mean model
eps <- rnorm(N, mean = 0, sd = sd)
CMM <- mean + eps

par(mfrow=c(1,1))
par(mgp=c(2, 0.7, 0), mar=c(3,3,2,1))
plot(data$Year, data$Diesel, xlab='Year', ylab="Diesel [Tonnes]", type='o', pch=20,
      main='Consumption of diesel in tonnes from 1966 to 2016', ylim=c(min(CMM), max(CMM)))

```

```

lines(data$Year, CMM, type='l', col="red", lty=2)

legend("topleft", legend=c("Consumption of Diesel", "Constant mean model"),
       col=c("black", "red"), lty=1:2, cex=0.8)

#histogram
breaks <- 25

h <- hist(data$Diesel,col="white",breaks = breaks,
           main = "Histogram with Normal Curve",
           xlab='Consumption of diesel in tonnes')

xfit <- seq(min(data$Diesel),max(data$Diesel),length=10000)
yfit <- dnorm(xfit,mean=mean,sd=sd)
yfit <- yfit*diff(h$mids[1:2])*N
lines(xfit, yfit, col="black", lwd=2)
segments(mean,
         dnorm(mean,mean=mean,sd=sd)*diff(h$mids[1:2])*N,
         mean,0,lwd=2,col="red",lty=2)
legend('topright',legend=c("Normal distributed repartition",
                           "Mean value"),col=c("black",'red'),
       lty =c(1,2),lwd=c(2,2),cex = 0.8)

#####Question 3#####
#####General linear model#####

#GLM (simple linear regression) on the new dataset using the function lm()
GLM <- lm(data_estim$Diesel ~ data_estim$Year)

#Theta_hat
coefficients(GLM)
## No need to remove least significant term because Pr(>|t|) << 1
summary(GLM)

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data_estim$Year, data_estim$Diesel,
      main="General linear model (simple linear regression)",type='o',pch=20,
      ylim=c(GLM$fitted.values[1],max(data_estim$Diesel)),xlab='Year',ylab="Diesel [Tonnes]")
lines(data_estim$Year, GLM$fitted.values, type='l', col="red",lty=1, lwd=2)

legend("topleft", legend=c("Consumption of Diesel", "GLM"),
       col=c("black", "red"), lty=c(1,1), cex=0.8)

#I estimate sigma - 2 parameters estimated
sigma_hat <- sqrt(sum(GLM$residuals^2)/(N_estim-2))

#residuals:
par(mfrow=c(1,2))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data_estim$Year, data_estim$Diesel-GLM$fitted.values,
      main="Residuals GLM",type='o',pch=20,
      xlab='Year', ylab='Diesel [Tonnes]')

```

```

lines(data_estim$Year, rep(0,N_estim), type='l', col="red", lty=2, lwd=2)

legend("topleft", legend=c("Residuals", "Supposed mean value (0)"),
       col=c("black", "red"), lty=c(1,1), cex=0.5)
## Checking distribution of residuals:
h <- hist(GLM$residuals, probability=T, col="white", breaks = breaks,
           main = "Histogram of residuals (GLM order 1)", xlab='Residuals',
           lwd=2,lty=1)
curve(dnorm(x,mean = 0, sd = sigma_hat),add=TRUE,col='black', lwd=1)
legend('topleft',legend=c("Normal distributed repartition"),col=c("black"),
       lty =1,lwd=2,cex = 0.5)

#mean 0

# Predict 3 step ahead, i.e. for time 49:51 i.e. from year 2014:2016
X <- cbind(1,data_estim$Year)
predict.intervals <- numeric(3)

#Prediction interval 95% - 2 parameters estimated
for(i in 1:3){
  predict.intervals[i] <- qt(0.975, N_estim-2)*sigma_hat*
    sqrt( (1+t(c(1,2013+i)) %*% solve(t(X)%*%X) %*% c(1,2013+i)) )
}

Prediction <- data.frame(pred = GLM$coefficients[1]*1+
                           GLM$coefficients[2]*c(2014,2015,2016))

Prediction$lower <- Prediction$pred - predict.intervals
Prediction$upper <- Prediction$pred + predict.intervals
head(Prediction)

par(mfrow=c(1,2))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data_estim$Year, data_estim$Diesel, xlab='Year',ylab='Diesel',
      main="Prediction - GLM (order 1)", xlim=c(1966, 2016),
      ylim=c(min(data_estim$Diesel),max(data_comp$Diesel)))
lines(data_estim$Year, GLM$fitted.values, type="l", lwd=2)
lines(2014:2016, Prediction$pred, col='red',lwd=2)
lines(2014:2016, Prediction$lower, col='green',lwd=2)
lines(2014:2016, Prediction$upper, col='green',lwd=2)
lines(data_comp$Year,data_comp$Diesel,type='b',col='blue',lwd=2)

#Let's try to formulate a GLM model with three parameters manually

X <- cbind(1,data_estim$Year-2013,I(data_estim$Year-2013)^2)
theta <- solve(t(X)%*%X, t(X)%*% data_estim$Diesel)
theta
#I estimate the new data
Diesel_estim <- X%*%theta

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data_estim$Year, data_estim$Diesel, xlab = 'Year', ylab = 'Diesel [Tonnes]',
```

```

    main="General linear model (order 2)",type='o',pch=20)
lines(data_estim$Year, Diesel_estim, type='l', col="red",lty=1, lwd=2)
legend("topleft", legend=c("Consumption of Diesel", "GLM 2"),
       col=c("black", "red"), lty=c(1,1), cex=0.8)

#I estimate the new sigma_hat - 3 parameters estimated
res <- data_estim$Diesel-Diesel_estim
sigma_hat <- sqrt( sum((res)^2)/(N_estim-3) )

#residuals:
par(mfrow=c(1,2))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data_estim$Year, res,
      main="Residuals GLM 2",type='o',pch=20,
      xlab='Year', ylab='Diesel [Tonnes]')
lines(data_estim$Year, rep(0,N_estim),type='l', col="red",lty=2, lwd=2)

legend("topleft", legend=c("Residuals", "Supposed mean value (0)"),
       col=c("black", "red"), lty=c(1,1), cex=0.5)
## Checking distribution of residuals:
h <- hist(res, probability=T, col="white",breaks = breaks,
           main = "Histogram of residuals (GLM 2)",xlab='Residuals',
           lwd=2,lty=1)
curve(dnorm(x,mean = 0, sd = sigma_hat),add=TRUE,col='black', lwd=1)
legend('topleft',legend=c("Normal distributed repartition"),col=c("black"),
       lty =1,lwd=2,cex = 0.5)
#mean zero

# Predict 3 step ahead, i.e. for time 49:51 i.e. for year 2014:2016
predict.intervals <- numeric(3)

#Prediction interval 95%
for(i in 1:3){
  predict.intervals[i] <- qt(0.975, N_estim-3) * sigma_hat*
    sqrt( (1+t(c(1,i,i^2)) %*% solve(t(X)%*%X) %*% c(1,i,i^2)) )
}

X_pred = cbind(1,c(1,2,3),I(c(1,2,3)^2))

Prediction <- data.frame(pred = X_pred%*%theta)
# Prediction intervals
Prediction$lower <- Prediction$pred - predict.intervals
Prediction$upper <- Prediction$pred + predict.intervals
head(Prediction)

plot(data_estim$Year, data_estim$Diesel, xlab='Year',ylab='Diesel',
      main="Prediction - GLM (order 2)", xlim=c(1966, 2016),
      ylim=c(min(data_estim$Diesel),
             max(max(data_comp$Diesel),max(Prediction$upper))))
lines(data_estim$Year, Diesel_estim, type="l", lwd=2)
lines(2014:2016, Prediction$pred, col='red',lwd=2)
lines(2014:2016, Prediction$lower, col='green',lwd=2)
lines(2014:2016, Prediction$upper, col='green',lwd=2)

```

```

lines(data_comp$Year,data_comp$Diesel,type='b',col='blue',lwd=2)

#####
#Exponential Smoothing
lambda <- 0.8

mu <- rep(0,N_estim)
mu[1] <- data$Diesel[1]
for (i in 2:(N_estim))
{
  mu[i]<-(1-lambda)*data$Diesel[i]+lambda*mu[i-1]
}

#Prediction :  $Y(N+l) = \mu(N)$ 
Y_SexpS <- c(data$Diesel[1],mu[1:N_estim],mu[48],mu[48])

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data$Year, data$Diesel, xlab = 'Year', ylab = 'Diesel [Tonnes]',
      main="Simple exponential smoothing 0.8",type='o',pch=20,lwd=1.5)
lines(data$Year[1:48], Y_SexpS[1:48], type='o',pch=20, col="red",lty=1,lwd=1.5)
lines(data$Year[49:51], Y_SexpS[49:51], type='o', col="red",lty=1,lwd=1.5,pch=3)
legend("topleft", legend=c("Consumption of Diesel", "One-step predictions","Prediction"),
       col=c("black", "red","red"),lty=c(1,1,1),lwd=c(1.5,1.5,1.5),
       pch=c(20,20,3), cex=0.8)

#####

#Local linear trend model
lambda<-0.8

#Burning period
burn<-5

#Defining matrix
f<- function(j) rbind(1,j)
F<-rbind(c(0,0),c(0,0))
h<-rbind(0,0)
L<-rbind(c(1,0),c(1,1))

#Initiating the prediction during the burning period
Diesel_est<-rbind(data$Diesel[1],data$Diesel[2],data$Diesel[3],data$Diesel[4])

for (j in 0:(burn-1))
{
  F<- F + lambda^(j)*f(-j)%*%t(f(-j))
  h<- h + lambda^(j)*f(-j)%*%t(data$Diesel[burn-j])
}

theta<-solve(F,h)
#F5, h5, theta5
weight_sigmag <- 1/( 1 + t(f(1))%*%solve(F)%*%f(1))

```

```

Diesel_est<-rbind(Diesel_est,t(f(0))%*%theta)
Diesel_pred<-t(f(1))%*%theta

#looping on the new observations
for (j in burn:(N_estim-2))
{
  F <- F + lambda^(j)*f(-j)%*%t(f(-j))
  h <- lambda*solve(L)%*%h + f(0)*data$Diesel[j+1]
  theta<- solve(F,h)

  Diesel_est<-rbind(Diesel_est,t(f(0))%*%theta )
  Diesel_pred<-rbind(Diesel_pred,t(f(1))%*%theta)
  weight_sigmag <- rbind(weight_sigmag,1/(1 + t(f(1))%*%solve(F)%*%f(1)))
}

#F(N_estim -1)
#Diesel_perd -> YN_estim/N_estim-1
j <- N_estim-1
F <- F + lambda^(j)*f(-j)%*%t(f(-j))
h <- lambda*solve(L)%*%h + f(0)*data$Diesel[j+1]
theta<- solve(F,h)

Diesel_est<-rbind(Diesel_est,t(f(0))%*%theta )

#Standard deviation of the residual LLM
T <- (1-lambda^(N_estim))/(1-lambda)
#sigmal2 <- 1/(T-2)*sum(lambda^(seq(N_estim-1,0,-1))
#                                *(data_estim$Diesel-Diesel_est)^2)

#In this case, we used a different theta for each estimation
#Hence, the weight is already given to the error
#there is no need to multiply the error with ambda^(seq(N_estim-1,0,-1))
sigmal2 <- 1/(T-2)*sum((data_estim$Diesel-Diesel_est)^2)

sigmag2 <- 1/(N_estim-burn)*
  sum((data$Diesel[6:N_estim]-Diesel_pred)^2*weight_sigmag)

X_pred <- cbind(1,c(1,2,3))
Y_pred <- X_pred%*%theta

# Predict 3 step ahead, i.e. for time 49:51 i.e. for year 2014:2016
predict.intervals <- data.frame(loc=numeric(3),glob=numeric(3))

#Prediction interval 95%
for(i in 1:3){
  predict.intervals$loc[i] <- qt(0.975, T-2) * sqrt(sigmal2)*
    sqrt(1 + t(f(i))%*%solve(F)%*%f(i))
  predict.intervals$glob[i] <- qt(0.975, N_estim-burn) * sqrt(sigmag2)*
    sqrt(1 + t(f(i))%*%solve(F)%*%f(i))
}

Pred <- data.frame(pred = Y_pred)
# Prediction intervals

```

```

Pred$lower_loc <- Pred$pred - predict.intervals$loc
Pred$upper_loc <- Pred$pred + predict.intervals$loc
Pred$lower_glob <- Pred$pred - predict.intervals$glob
Pred$upper_glob <- Pred$pred + predict.intervals$glob

Pred <- data.frame(Year=c(2014,2015,2016),Pred)

par(mfrow=c(1,1))
par(mgp=c(2, 0.7, 0), mar=c(3,3,2,1))
plot(data$Year, data$Diesel, xlab='Year', ylab='Diesel [Tonnes]',
      main="Local linear trend model 0.8", type='o', pch=20, lwd=1.5,
      cex = 0.5, ylim = c(min(Diesel_pred),max(Pred$upper_loc)))
lines(data$Year[(burn+1):48], Diesel_pred, pch = 20, cex = 0.5,
      col = 'red', type = 'o')
lines(data$Year[49:51], Y_pred, pch = 3, cex = 0.5, col = 'red', type = 'l')
lines(data$Year[49:51], Pred$lower_glob, pch = 3, cex = 0.5,
      col = 'green', type = 'l', lty=5)
lines(data$Year[49:51], Pred$upper_glob, pch = 3, cex = 0.5,
      col = 'green', type = 'l', lty=5)
lines(data$Year[49:51], Pred$lower_loc, pch = 3, cex = 0.5,
      col = 'blue', type = 'l', lty=5)
lines(data$Year[49:51], Pred$upper_loc, pch = 3, cex = 0.5,
      col = 'blue', type = 'l', lty=5)
legend('topleft', legend = c("Observations", "One-step prediction", "Prediction",
                            "95% global estimator", '95% local estimator'),
       col = c("black","red","red","green",'blue'), lty = c(1,1,1,2,2),
       pch = c(16,16,NA,NA,NA), cex=0.7, bty = 'n')

#####
#####Question 6#####
lambdas <- seq(0.001,0.999,0.001)

#sum of squared one-step prediction errors
SSE <- c()
sigmals <- c()
burn <- 5

#looping on the lambda
for (lambda in lambdas)
{
  #Defining matrix
  f<- function(j) rbind(1,j)
  F<-rbind(c(0,0),c(0,0))
  h<-rbind(0,0)
  L<-rbind(c(1,0),c(1,1))

  for (j in 0:(burn-1))
  {
    F<- F + lambda^(j)*f(-j)%*%t(f(-j))
    h<- h + lambda^(j)*f(-j)%*%t(data$Diesel[burn-j])
  }

  theta<-solve(F,h)
  #F5, h5, theta5
}

```

```

Diesel_pred<-t(f(1))%*%theta
#Y6/5

#looping on the new observations
for (j in burn:(N_estim-2))
{
  F <- F + lambda^(j)*f(-j)%*%t(f(-j))
  h <- lambda*solve(L)%*%h + f(0)*data$Diesel[j+1]
  theta<- solve(F,h)
  Diesel_pred<-rbind(Diesel_pred,t(f(1))%*%theta)
}
#F(N_estim -1)
#Diesel_perd -> YN_estim/N_estim-1

error <- sum((data$Diesel[6:N_estim]-Diesel_pred)^2)
SSE <- rbind(SSE,error)
}

plot(lambdas,SSE,main='Sum of squared one-step prediction errors',type = 'l')

lambda_min <- lambdas[which.min(SSE)]
#0.666

#Local linear trend model
lambda<-lambda_min

#Burning period
burn<-5

#Defining matrix
f<- function(j) rbind(1,j)
F<-rbind(c(0,0),c(0,0))
h<-rbind(0,0)
L<-rbind(c(1,0),c(1,1))

#Initiating the prediction during the burning period
Diesel_est<-rbind(data$Diesel[1],data$Diesel[2],data$Diesel[3],data$Diesel[4])

for (j in 0:(burn-1))
{
  F<- F + lambda^(j)*f(-j)%*%t(f(-j))
  h<- h + lambda^(j)*f(-j)%*%t(data$Diesel[burn-j])
}

theta<-solve(F,h)
#F5, h5, theta5
weight_sigmag <- 1/( 1 + t(f(1))%*%solve(F)%*%f(1))

Diesel_est<-rbind(Diesel_est,t(f(0))%*%theta)
Diesel_pred<-t(f(1))%*%theta

#looping on the new observations
for (j in burn:(N_estim-2))

```

```

{
  F <- F + lambda^(j)*f(-j)%*%t(f(-j))
  h <- lambda*solve(L)%*%h + f(0)*data$Diesel[j+1]
  theta<- solve(F,h)

  Diesel_est<-rbind(Diesel_est,t(f(0))%*%theta )
  Diesel_pred<-rbind(Diesel_pred,t(f(1))%*%theta)
  weight_sigmag <- rbind(weight_sigmag,1/(1 + t(f(1))%*%solve(F)%*%f(1)))
}
#F(N_estim -1)
#Diesel_perd -> YN_estim/N_estim-1
j <- N_estim-1
F <- F + lambda^(j)*f(-j)%*%t(f(-j))
h <- lambda*solve(L)%*%h + f(0)*data$Diesel[j+1]
theta<- solve(F,h)

Diesel_est<-rbind(Diesel_est,t(f(0))%*%theta )

#Standard deviation of the residual LLM
T <- (1-lambda^(N_estim))/(1-lambda)
#sigmal2 <- 1/(T-2)*sum(lambda^(seq(N_estim-1,0,-1))
#                                *(data.estim$Diesel-Diesel_est)^2)

#In this case, we used a different theta for each estimation
#Hence, the weight is already given to the error
#there is no need to multiply the error with ambda^(seq(N_estim-1,0,-1))
sigmal2 <- 1/(T-2)*sum((data.estim$Diesel-Diesel_est)^2)

sigmag2 <- 1/(N_estim-burn)*
  sum((data$Diesel[6:N_estim]-Diesel_pred)^2*weight_sigmag)

X_pred = cbind(1,c(1,2,3))
Y_pred <- X_pred%*%theta

# Predict 3 step ahead, i.e. for time 49:51 i.e. for year 2014:2016
predict.intervals <- data.frame(loc=numeric(3),glob=numeric(3))

#Prediction interval 95%
for(i in 1:3){
  predict.intervals$loc[i] <- qt(0.975, T-2) * sqrt(sigmal2)*
    sqrt(1 + t(f(i))%*%solve(F)%*%f(i))
  predict.intervals$glob[i] <- qt(0.975, N_estim-burn) * sqrt(sigmag2)*
    sqrt(1 + t(f(i))%*%solve(F)%*%f(i))
}

Pred <- data.frame(pred = Y_pred)
# Prediction intervals
Pred$lower_loc <- Pred$pred - predict.intervals$loc
Pred$upper_loc <- Pred$pred + predict.intervals$loc
Pred$lower_glob <- Pred$pred - predict.intervals$glob
Pred$upper_glob <- Pred$pred + predict.intervals$glob

Pred <- data.frame(Year=c(2014,2015,2016),Pred)

```

```

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data$Year, data$Diesel, xlab='Year', ylab='Diesel [Tonnes]',
      main="Local linear trend model 0.666",type='o',pch=20,lwd=1.5,
      cex = 0.5,ylim = c(min(Diesel_pred),max(data$Diesel)))
lines(data$Year[(burn+1):48], Diesel_pred, pch = 20, cex = 0.5,
      col = 'red', type = 'o')
lines(data$Year[49:51], Y_pred, pch = 20, cex = 0.5, col = 'red', type = 'l')
lines(data$Year[49:51], Pred$lower_glob, pch = 3, cex = 0.5,
      col = 'green', type = 'l',lty=5)
lines(data$Year[49:51], Pred$upper_glob, pch = 3, cex = 0.5,
      col = 'green', type = 'l',lty=5)
lines(data$Year[49:51], Pred$lower_loc, pch = 3, cex = 0.5,
      col = 'blue', type = 'l',lty=5)
lines(data$Year[49:51], Pred$upper_loc, pch = 3, cex = 0.5,
      col = 'blue', type = 'l',lty=5)
legend('topleft',legend = c("Observations", "One-step prediction", "Prediction",
                           "95% global estimator", '95% local estimator'),
       col = c("black","red","red",'green','blue'),lty = c(1,1,1,2,2),
       pch = c(16,16,NA,NA,NA),cex=0.7, bty = 'n')

#I estimate the new sigma_hat - 3 parameters estimated
res <- data$Diesel[6:N_estim]-Diesel_pred

#residuals:
par(mfrow=c(1,2))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data$Year[6:N_estim], res,
      main="Residuals local linear trend model 0.666",type='o',pch=20,
      xlab='Year', ylab='Diesel [Tonnes]')
lines(data_estim$Year, rep(0,N_estim),type='l', col="red",lty=2, lwd=2)

legend("topleft", legend=c("Residuals", "Supposed mean value (0)"),
       col=c("black", "red"), lty=c(1,1), cex=0.5)
## Checking distribution of residuals:
h <- hist(res, probability=T, col="white",breaks = breaks,
           main = "Histogram of residuals (LLTM)",xlab='Residuals',
           lwd=2,lty=1)
curve(dnorm(x,mean = 0, sd = sigma_hat),add=TRUE,col='black', lwd=1)
legend('topleft',legend=c("Normal distributed repartition"),col=c("black"),
      lty =1,lwd=2,cex = 0.5)
#mean zero

```