Assignment 1
Diesel Consumption

Christian Glissov, s146996

March 1, 2018

DTU Compute - Institute for Mathematics and Computer science
Time Series Analysis 02417
Lasse Engbo Christiansen

Collaborated with Thomas Ørkild, s154433 and Frederik Boe Hüttel, s151689

Technical University of Denmark

Contents

Introduction

This assignment looks at the annual consumption of diesel in Denmark from 1966 to 2016. One may find it interesting to look at the diesel consumption over time, perhaps due to environmental or economic reasons. To predict the future consumption of diesel a focus on different methods within time series analysis will be covered. The software used in this assignment is the program *R*.

The data

The annual diesel consumption in Denmark has been registered since 1966 all the way to 2016. The data set consists of two variables:

- **Time**: Indicates the year of the observation.
- **Diesel**: The consumption of diesel in tonnes.

Data set	Min	Max	Median	Standard Deviation	Mean
Year	1966	2016	1991	14.87	1991
Diesel	754425	2862549	1729586	799109.60	1751635

Table 1: Summary Statistics

The diesel consumption as a function of time, throughout 1966 to 2016, can be seen from figure (1).

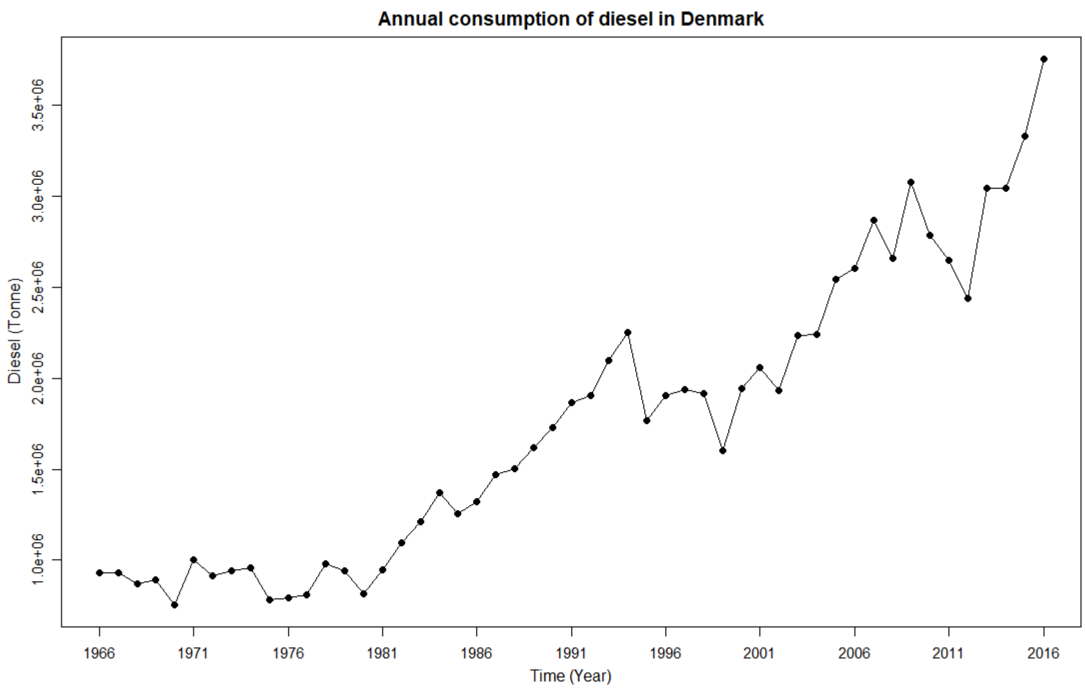

Figure 1: An overall increasing trend can be seen.

It can be seen that the time series data is non-stationary. Due to the global uptrend of diesel consumption over time and that the data oscillates a lot. The mean will therefore fluctuate and increase over time. This means that the global mean makes up a bad representation of the data. The same goes for the standard deviation. For the same reasons, the global standard deviation will be very large and fluctuate. In figure (2) the global mean and the confidence intervals are shown.

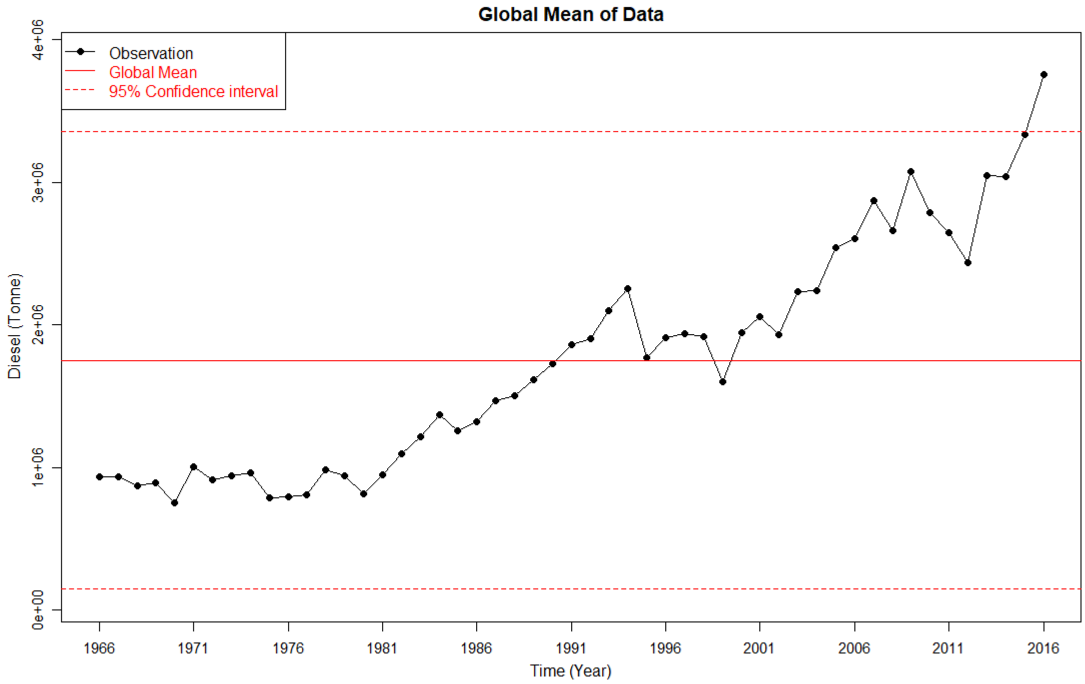

Figure 2: The global mean doesn't represent the data very well.

Clearly a constant mean model would be a bad choice to make predictions and forecasting.

From now on the last 3 observations (2014, 2015, 2016) will be excluded and only used for out of sample comparison of the prediction values of the different models. The three observations will be illustrated by a dashed line connecting them in the future figures.

Statistical analysis

The theory of the models used will be shortly introduced in this section.

General Linear Model

A simple linear regression will be used. This model is given by:

$$Y_t = \theta_0 + \theta_1 t + \epsilon_t$$

t is the annual time, **year**, Y_t is the response variable, which in this case is **diesel**, $\theta_i, i = 0, 1$ are the parameters. Finally it is assumed that ϵ_t is IID and $E[\epsilon_t] = 0$

and $Var[\epsilon_t] = \sigma^2$. The estimates $\hat{\theta}_i$ will be found by using the OLS, so minimising the following method (eq 3.35, in [1]):

$$\hat{\theta} = (\mathbf{t}^T\mathbf{t})^{-1}\mathbf{t}^T\mathbf{Y}$$

Where the design matrix, $\mathbf{t}$, needs to have full rank.

Exponential Smoothing

Unlike the constant mean model, where each observation contributes equally to the predictions, the exponential smoothing allows for a weighting of each observation. Here the most recent observations have more weight than the observations in the past. The simple exponential smoothing sequence (eq 3.76, in [1]) is given by:

$$S_N = (1-\lambda)Y_N + \lambda S_{N-1}$$

Where λ ($|\lambda| < 1$) is the forgetting factor and $1-\lambda$ is the smoothing constant α. The influence of the initial value, S_0, won't contribute a lot, in this case S_0 is the first observation given. The l-step prediction is then given by (eq 3.73, [1]):

$$\hat{Y}_{N+l|N} = \hat{Y}_{N|N} \tag{1}$$

Therefore for this model a constant out-of-sample forecast can be expected.

Linear Trend Model

In this assignment the focus will be on the local linear trend model. The local trend model differs from the global trend model, by assigning a given weight to each observation. Where recent observations will have a higher weight than distant ones. A linear trend model is of the form given by (eq 3.83, [1]):

$$Y_{N+j} = \mathbf{f}^T(j)\boldsymbol{\theta} + \epsilon_{N+j}, \quad \mathbf{f}(j) = [1 \quad j]^T \tag{2}$$

Where $Var[\epsilon] = \sigma^2\Sigma$. To find the estimates one has to minimise the following equation (eq. 3.97, [1]):

$$\hat{\boldsymbol{\theta}}_N = \arg\min_{\theta} S(\boldsymbol{\theta}; N) \tag{3}$$

$$S(\boldsymbol{\theta}; N) = \sum_{j=0}^{N-1} \lambda^j [Y_{N-j} - \mathbf{f}^T(-j)\boldsymbol{\theta}]^2, \quad 0 < \lambda < 1 \tag{4}$$

The WLS problem can be solved by Theorem 3.3 (page 38, in [1]) and thus the solution is:

$$\hat{\boldsymbol{\theta}}_N = \mathbf{F}_N^{-1}\mathbf{h}_N \tag{5}$$

Where

$$F_N = \sum_{j=0}^{N-1} \lambda^j \mathbf{f}(-j)\mathbf{f}^T(-j) \tag{6}$$

$$h_N = \sum_{j=0}^{N-1} \lambda^j \mathbf{f}(-j) Y_{N-j} \tag{7}$$

The prediction is then found by:

$$\hat{Y}_{N+l|N} = \mathbf{f}^T(l)\hat{\boldsymbol{\theta}}_N \tag{8}$$

Where l is the step. Recursively equation (4), (5) and (6) will keep getting updated for each time step. The variance estimator is given by [2]:

$$\hat{\sigma}^2 = \frac{\boldsymbol{e}^T \Sigma^{-1} \boldsymbol{e}}{T-p}, \quad T > p \tag{9}$$

Where $\boldsymbol{e}$ is the residuals, $\Sigma = \text{diag}[\frac{1}{\lambda^{N-1}}, ..., \frac{1}{\lambda}, 1]$, T is the total memory $T = \frac{1-\lambda^N}{1-\lambda}$. Finally the variance of the prediction error, e, can be found to make a prediction interval of the forecasts (eq 3.91, [2]):

$$Var[e_N(l)] = \sigma^2[1 + \mathbf{f}^T(l)\mathbf{F}_N^{-1}\mathbf{f}(l)] \tag{10}$$

The prediction interval can then be found for the future value Y_{N+l} (eq. 3.92, [1]):

$$\hat{Y}_{N+l|N} \pm t_{\alpha/2}(N-p)\hat{\sigma}\sqrt{Var[e_N(l)]} \tag{11}$$

Results

The three different methods will now be used on the data set, containing the diesel and time variable.

Simple Linear Regression Model

In figure (3) confidence interval, prediction interval and the simple linear regression fit of annual diesel consumption from 1966 to 2013 can be observed.

$$Y_{t_{year}} = -92952621 + 47551 t_{year} + \epsilon_{t_{year}}$$

Where the residual standard error is $RSE_\epsilon = 229000$. The model does a poor job of describing the data. It captures that there is an overall upward trend, by assuming that for each year there will be a constant increase in diesel consumption. However this also means that the local fluctuations of the observations is a problem for the linear model, the model can only predict a constant increase. This can be seen by how the observations either lie quite low or above the fitted line and the confidence interval. Another thing to notice is that the prediction interval is a lot wider than the confidence interval, this indicates that a given prediction will likely lie far away from the fitted line.

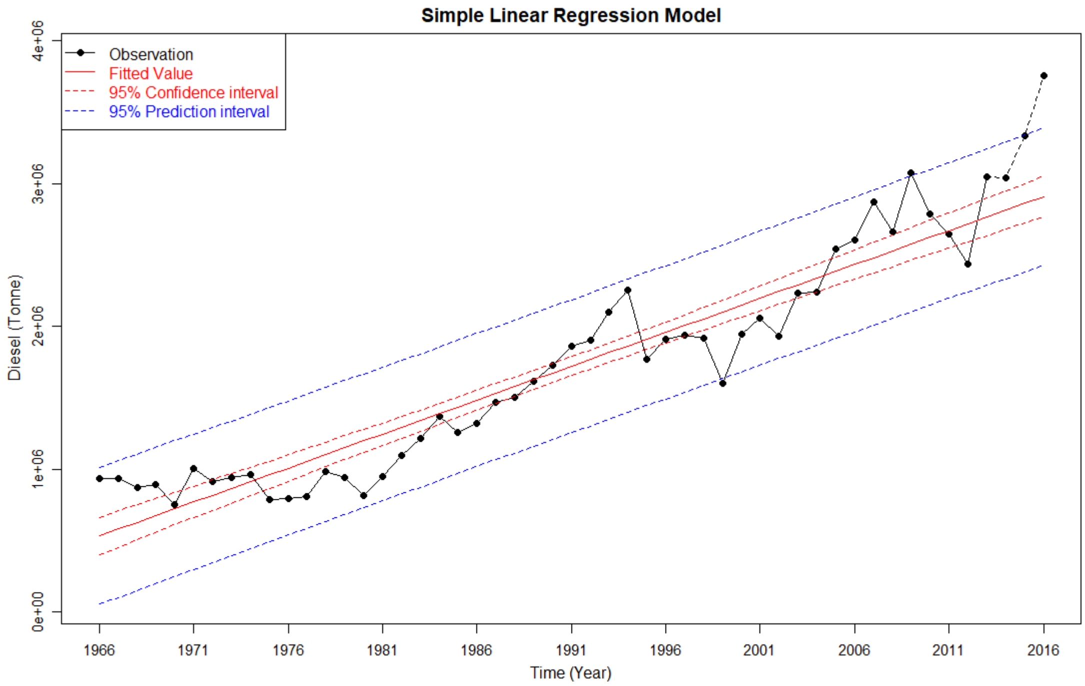

Figure 3: The model doesn't fit very well.

By looking at figure (4) it seems like the assumption of homoscedasticity is not satisfied in the scale-location plot, a slight upwards curve is noticed. In the residuals vs fitted plot there seem to be a non-linear relationship and the assumption of the residuals being independent and identically distributed is not satisfied.

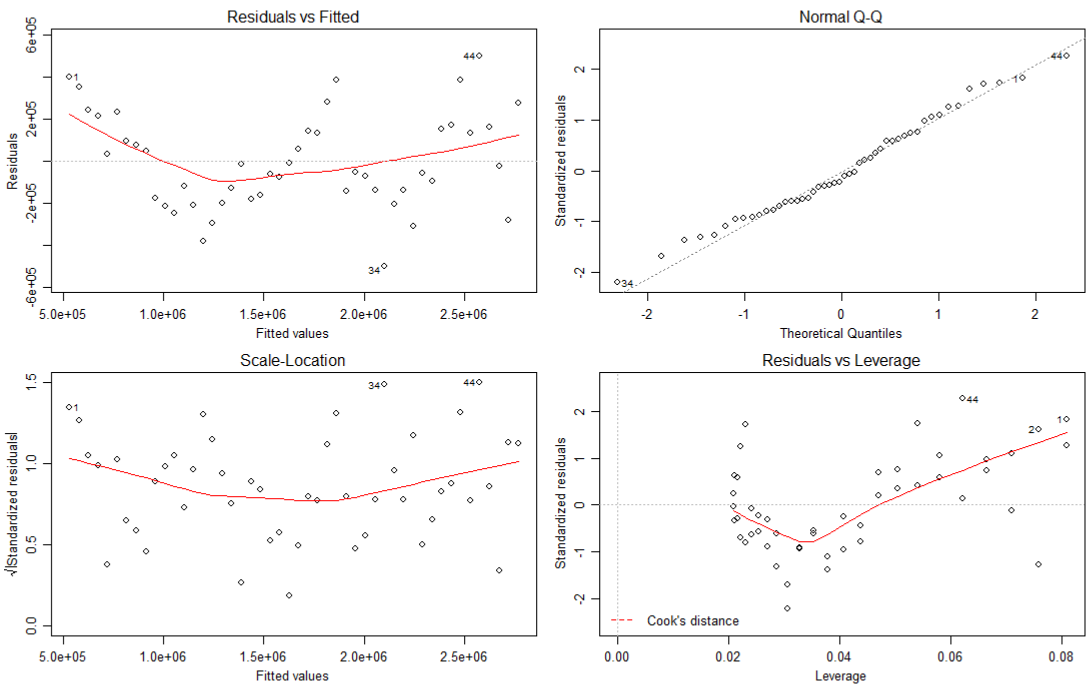

Figure 4: Residual plots of the simple linear regression model.

When comparing the fitted line to the last three observations, which were excluded, it can be seen that the model predicts an increase in those years, which is correct, however this is merely due to the observations following the upward trend and not to the model making an optimal prediction. Based upon the above reasons, the simple linear regression is rejected for making predictions in this case.

Year	Observation	Prediction	Uncertainty 95%	Residual
2014	3043197	2814999	$\pm$ 480439	228198
2015	3333391	2862549	$\pm$ 481638	470842
2016	3756795	2910100	$\pm$ 482882	846695

Table 2: Table of predictions for the years that were left out of the simple linear regression model.

Simple Exponential Smoothing

The simple exponential smoothing was used with a forgetting factor $\lambda = 0.8$ and with a 1-step prediction, therefore each prediction is based on the last prediction and current observation. The 1-step fit can be seen in figure (5). The method smooths out the time series by having a low smoothing factor $\alpha = 1 - \lambda$. The low smoothing factor gives a slow response to changes in the mean value of the process. This explains the rather slow and smooth increase of the predictions e.g. from 1991 when there is a positive trend.

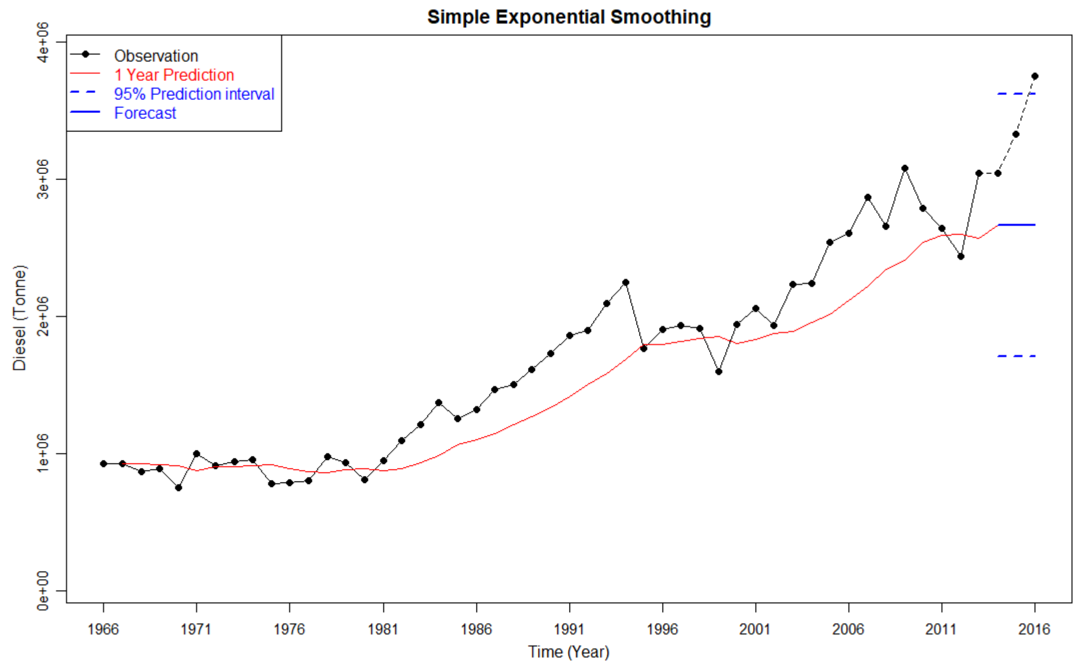

Figure 5: The simple exponential smoothing is a constant trend model, therefore not a very good prediction.

The future forecast will be the last estimate of the exponential smoothing, this is why a constant forecast can be seen for the years 2014, 2015 and 2016. It can be seen that as long as the model has recent observations to use, it is a better fit compared to the simple linear regression model. However due to the method being dependent on the most recent observations the future forecast of a constant annual consumption is not optimal. The prediction intervals is found by using the same method as for finding the prediction intervals for the linear trend model. The intervals are wide and therefore the uncertainty is high.

Year	Observation	Prediction	PI 95%	Residual
2014	3043197	2666069	± 953002	377128
2015	3333391	2666069	± 953002	667322
2016	3756795	2666069	± 953002	1090726

Table 3: Table of predictions for the three years left out.

From table (3) one can notice that the residual error increases, due to the constant forecast and increase of annual diesel consumption of the excluded observations.

Local Linear Trend Model

In figure (7) the local linear trend fit can be seen with a burn-in period of 5 observations and $\lambda = 0.8$. The out of sample observations is predicted by (eq. 8):

$$\hat{Y}_{N+l|N} = \begin{bmatrix} 1 & l \end{bmatrix} \begin{bmatrix} 2885230 \\ 54795.33 \end{bmatrix}, \quad l \in 1, 2, 3 \tag{12}$$

The standardized residuals against fitted values can be seen in figure (6)

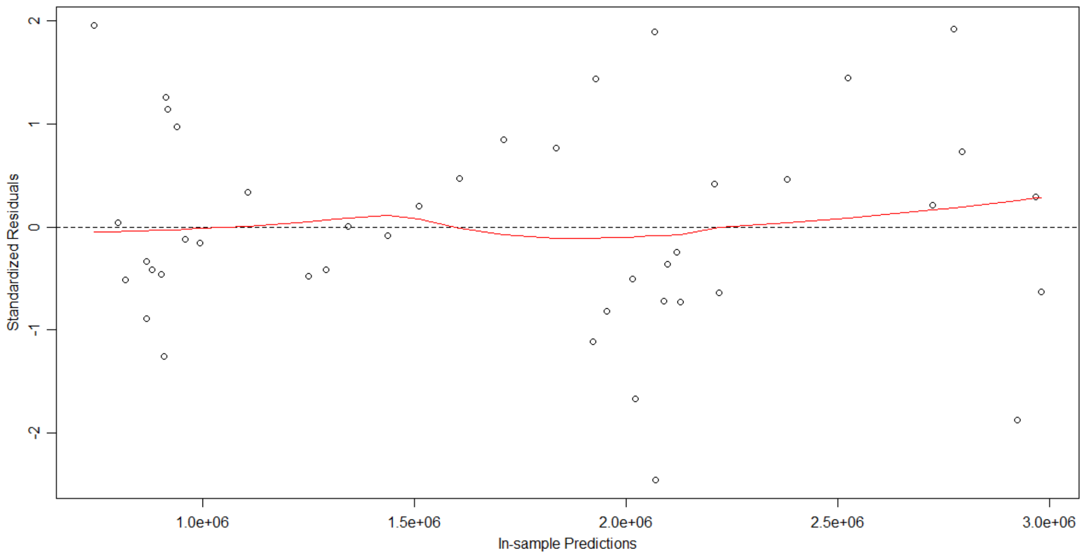

Figure 6: Residuals plot

The residuals seems to be fairly IID, however there is a slight increase in the variance at the end, but quite few observations so difficult to tell. However an improvement compared to the linear regression. The linear trend model captures the behaviour of the time series a lot better compared to the last two methods.

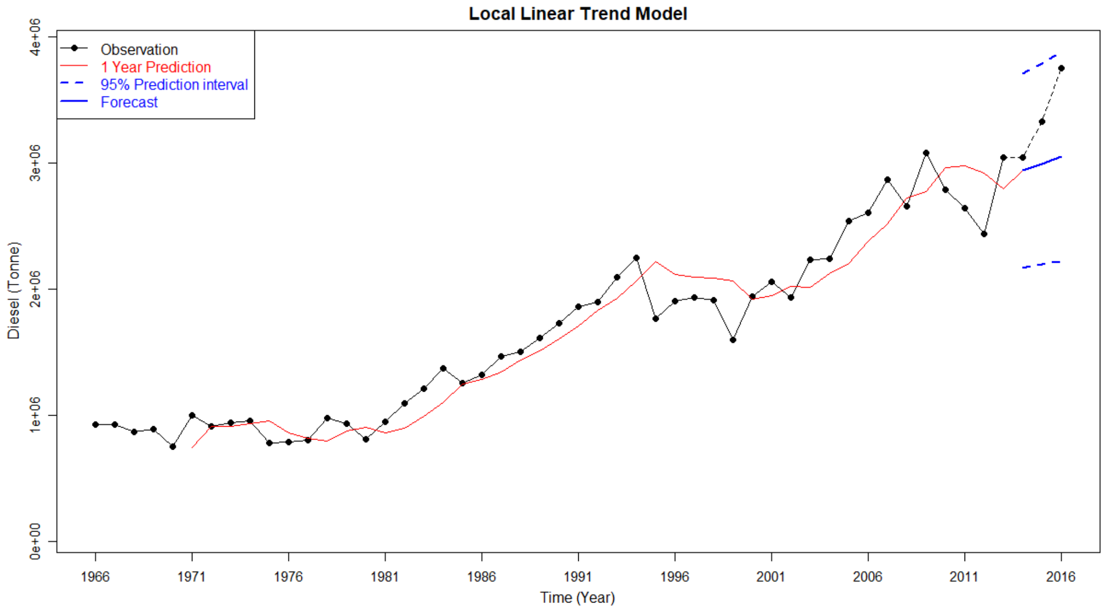

Figure 7: The model captures the trend better than the previous models.

By using the WLS method it is based on the most recent observations and the emphasis of each observation can be seen in figure (8). The weight in 2013 is 1 compared to e.g 1971, which is basically 0. This means for a future prediction, 2013 will have the greatest impact. It can be seen in forecast predictions, even though the time series decrease in 2009 to 2012, the increase in 2013 is enough to make the series predict a positive increase in diesel consumption. This is because of the weight of 2013.

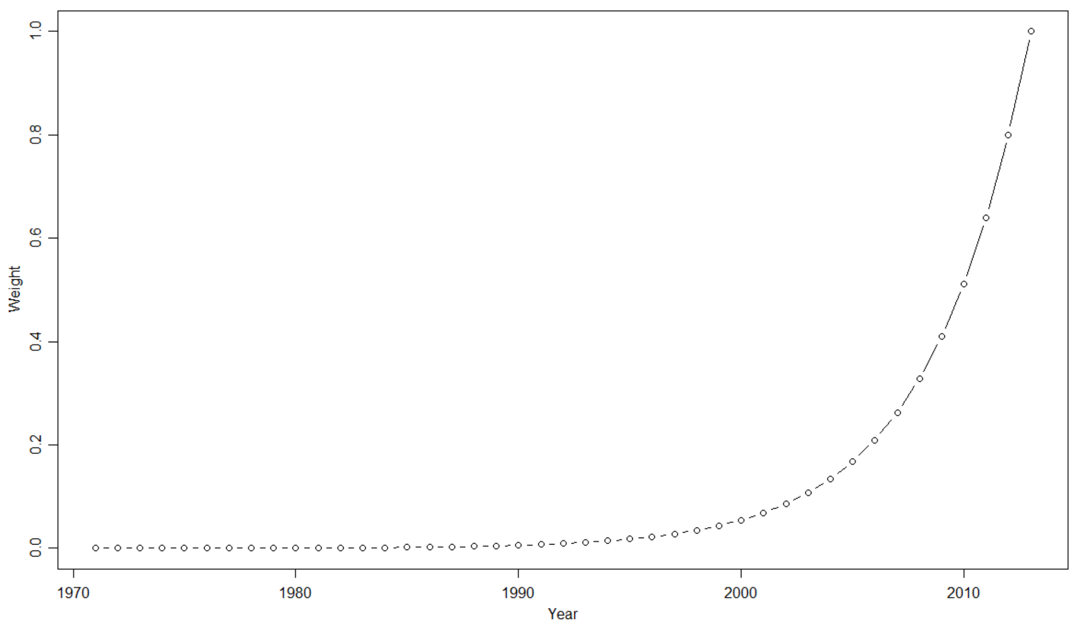

Figure 8: Weights for each year of the model with $\lambda = 0.8$.

The prediction intervals are still fairly wide, but this is expected due to the high variance. The local linear trend model is a significant improvement compared to the two other methods.

Year	Observation	Prediction	Uncertainty 95%	Residual
2014	3043197	2940026	± 766465	103171
2015	3333391	2994821	± 795063	338570
2016	3756795	3049616	± 827589	707179

Table 4: Table of predictions for the three years left out.

The residual error is relatively low (table 4) and for the 1-step prediction the linear trend model does a fine job. It captures the trend well, however it underestimate the increase for 2015 and 2016, but this is expected.

Optimising the Forgetting Factor

In the previous sections a forgetting factor was randomly set to 0.8. To optimise the chosen factor, the sum of squares will be used (eq 3.79 in [1]):

$$\hat{\lambda} = \arg\min_{\lambda} S(\lambda; N) \tag{13}$$

$$S(\lambda; N) = \sum_{t=0}^{N} (Y_t - \hat{Y}_{t|t-1}(\lambda)) \tag{14}$$

Running this for the same local linear trend model in R and in figure (9) the optimal value of forgetting factor is $\lambda = 0.67$ when using a burn in period of 5.

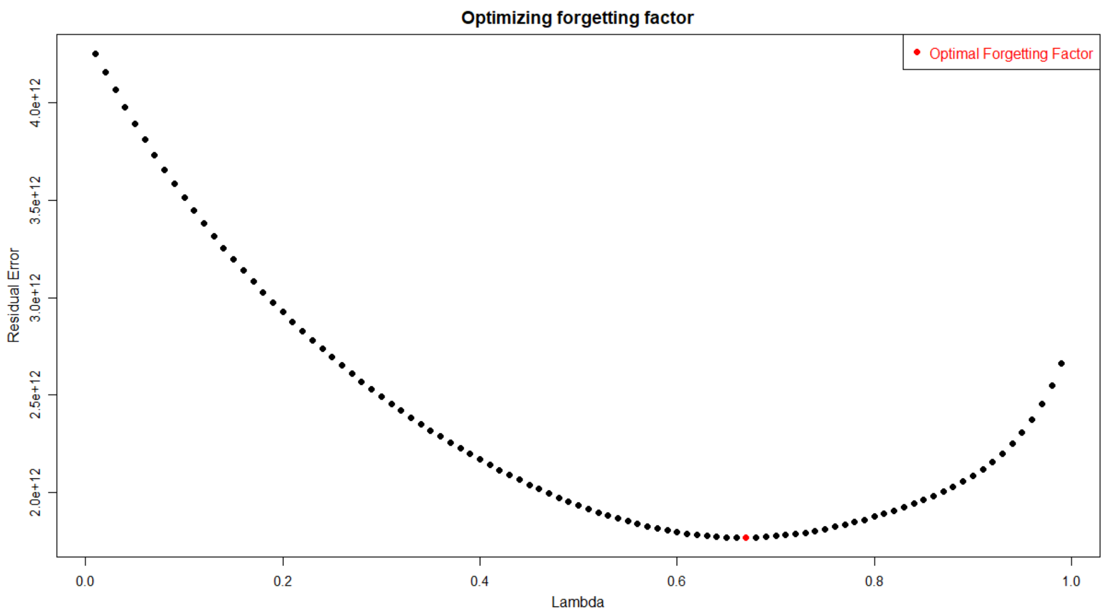

Figure 9: Optimising the forgetting factor λ. 0.67 seems to be optimal.

The optimised linear trend can be seen in figure (10).

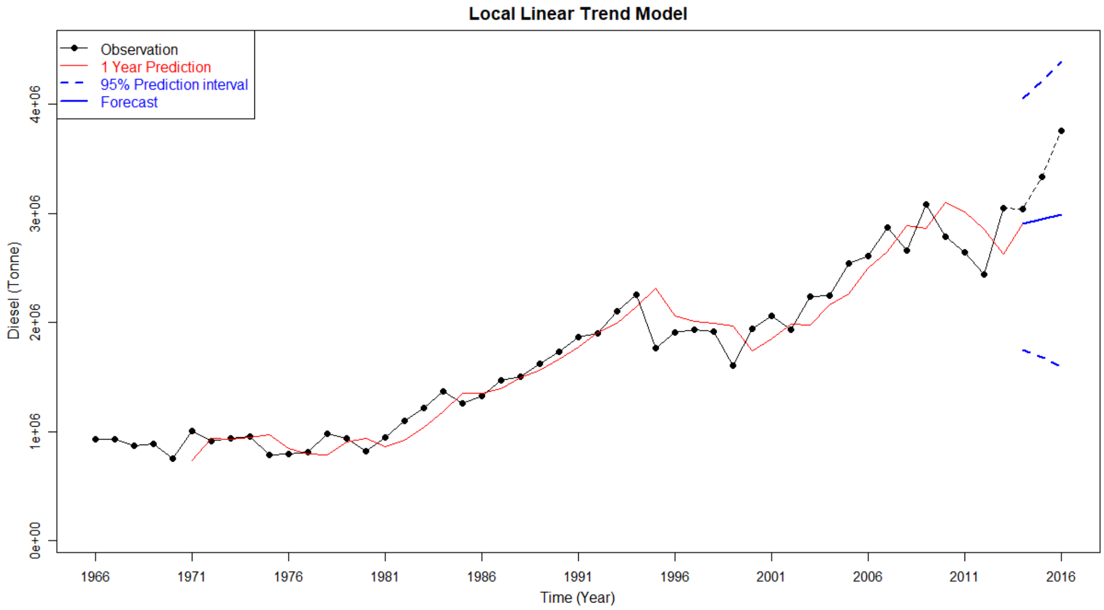

Figure 10: New local linear trend model with optimal forgetting factor.

One may notice that the prediction intervals are a lot wider compared to the model with a lower smoothing factor. As explained earlier, the smoothing factor is the judge of how many past observations will be used. Looking at figure (11) and comparing it to figure (8) it can be seen that the weights all up until 2013 have been decreased. The influence of recent observations are therefore even more significant than before. This will give lower total memory value T and due to the lower total memory, the variance estimator $\hat{\sigma}^2$ will increase, which further means a greater uncertainty. This makes intuitively sense as well, the fewer observations our prediction is based on, the more uncertain the prediction will be.

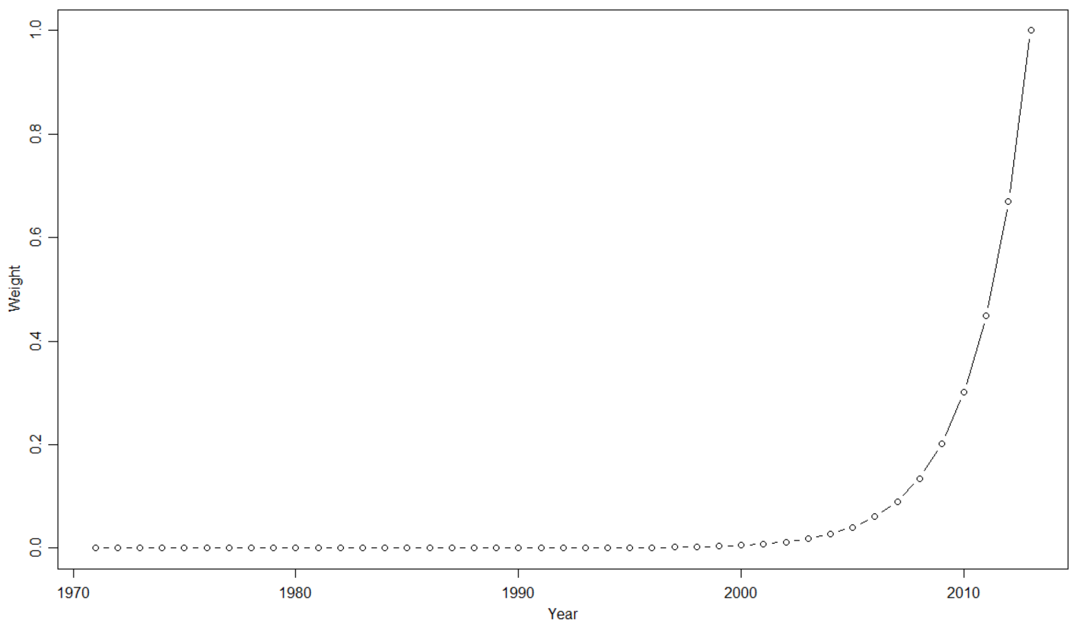

Figure 11: Weights for each year of the optimized model, $\lambda = 0.67$.

Looking at the predictions in table (5) it is seen that the predictions haven't improved much, they have actually worsened. Once again this makes sense. The recent downtrend will have a larger relative influence, when the smoothing is higher and the uptrend before 2009 will almost have no influence, the short uptrend in 2013, can't make up for the decrease in 2009-2012, so a slightly lower forecast prediction is made in this case.

Year	Observation	Prediction	Uncertainty 95%	Residual
2014	3043197	2902168	$\pm$ 1151418	141029
2015	3333391	2947200	$\pm$ 1265404	386191
2016	3756795	2992231	$\pm$ 1398130	764564

Table 5: Table of predictions for the three years left out for optimized λ model.

Discussion

Comparing all the models it is clear that the local linear trend model does a better job. However there are room for improvements in all the models. The linear regression model might be able at describing a global trend but a linear fit is not optimal for prediction. A factor that might improve the prediction of the global trend, could be a quadratic fit, but it is tough to tell whether the curvature of the global trend is quadratic or linear, more data is needed.

For the exponential smoothing, this model is good for looking at the local level of the time series. An idea could be to also include seasonality into the exponential smoothing, by using Holt Winters method. There seems to be certain periods where diesel consumption increases and decreases and another improvement would be to optimise the forgetting factor. The exponential smoothing model fails when having to predict further than one unit forward, due to it acting like a constant mean model, when no new observations is given.

The local linear trend is like a combination of the properties of the linear regression and the exponential smoothing method. By looking at the adaptive component and the linearity property makes the prediction a lot better compared to the previous models. Once again there are room for improvements. A linear model might be too simple, it would be interesting to see how the model predicts if a more complex trend model was used. Such as a quadratic trend model.

Also a completely different model might be needed to get better predictions. By looking at the data there might be hidden factors. Diesel are a byproduct of crude oil. Looking into factors such as how the oil price/supply increase or decrease over time ([3]), might help predicting the annual diesel consumption. Taking a look at 2008, the financial crisis made oil prices drop a lot, however already in 2009 they had almost recovered and steadily increased all the way to 2014, which might explain the decrease in diesel consumption from 2009 to 2013. A combination of a possible end of the financial crisis and low oil prices might explain the sharp increase of diesel consumption in 2014-2016. The point of this is, a model containing more than simply the time as an exogenous variable, but other variables might help optimise the predictability of diesel consumption. Models which takes into account the linear dependencies between these variables would be interesting to apply.

The forecasts are mainly based on the previous years observation and because of the large variance in the data, the uncertainty is large. One should not put trust in the exponential smoothing and simple linear regression, because of respectively the constant mean and the constant linear increase, which is not optimal for this data set. For the local linear trend model it correctly predicted an increase and the predictions are more trustworthy than the other two models, but there is still a large uncertainty and one may ask if the model is too simple to achieve optimal predictions of future observations for this data set. Therefore cautiousness is recommended with trusting the forecast for the out of sample observations, especially for more than 1-step predictions.

Conclusion

The three models have been implemented. The local linear trend is the model to prefer. It had the best predictions and captured the trend quite well. It has the linearity of the simple linear regression model, furthermore it has the properties of adaptation and locality of the exponential smoothing method, which makes it a better model. Hidden factors, large uncertainty and over simplicity of the models, concludes that one should be cautious of trusting the results of the local linear trend model. Future predictions should not be trusted for the exponential smoothing method and linear regression model for this kind of data set.

References

[1] Henrik Madsen, *Time Series Analysis*, Chapman & Hall/CRC, 2008, Chapter 3.

[2] Lasse Engbo Christensen, Slides from lecture 4.

[3] Crude oil price chart: http://www.macrotrends.net/1369/crude-oil-price-history-chart

Appendix

```
rm(list=ls())
#############################ASSIGNMENT 1####################
#############################################################
options("scipen" = -1)
#par(mgp=c(3, 1, 0), mar=c(5.1, 4.1, 4.1, 2.1), las=0,
    pty="m")

setwd("SET␣DIRECTORY␣HERE")


A1.full <- read.table("A1_diesel.txt", sep="\t", header=T)
names(A1.full) <- c("year", "diesel")
str(A1.full)
sum(is.na(A1.full))

#comp<-A1.full[c((51-2):51),]
A1<-A1.full[c(1:(51-3)),]
N<-nrow(A1.full)

par(mfrow=c(1,1))
par(mgp=c(2,0.8,0), mar=c(3,3,2,1),las=0, pty="m")
plot(diesel ~ year, A1.full, ylab="Diesel␣(Tonne)",
              xlab="Time␣(Year)",
              main="Annual␣consumption␣of␣diesel␣in␣Denmark",
              cex.axis=0.95,
     pch= 19,xaxt='n')
lines(diesel ~ year, A1.full)
axis(1, seq(1966, 2016, by=5),
    seq(1966, 2016, by=5), cex.axis = 0.95)

###################QUESTION 2
plot(diesel ~ year, A1.full, ylab="Diesel␣(Tonne)",
     xlab="Time␣(Year)",
     main="Global␣Mean␣of␣Data", cex.axis=0.95,
     pch= 19,xaxt='n', ylim=c(70000,3900000))
axis(1, seq(1966, 2016, by=5),
    seq(1966, 2016, by=5), cex.axis = 0.95)
lines(diesel ~ year, A1.full)
```

```
#Global mean and standard deviation of data
abline(mean(A1.full$diesel), 0, col='red')
legend("topleft",
       legend=c("Observation", "Global␣Mean",
        "95\%␣Confidence␣interval"),
       col=c("black","red", "red"),
        text.col=c("black","red", "red"),
       lty=c(1, 1, 2),pch = c(19, NA, NA))
#text(1967, 1700000, label=c("1751635 tonnes"),
        cex=0.8, col="red")

#conf int
abline(mean(A1.full$diesel)+qt(0.975,df=51-1)*
        sd(A1.full$diesel),0, col='red', lty=2)
abline(mean(A1.full$diesel)-qt(0.975,df=51-1)*
        sd(A1.full$diesel),0, col='red', lty=2)

###############QUESTION 3
#time<-A1$year-1966
lm1<-lm(diesel ~ year, A1)
par(mfrow=c(2,2))
plot(lm1)
summary(lm1)
par(mfrow=c(1,1))

plot(diesel ~ year, A1.full, ylab="Diesel␣(Tonne)",
     xlab="Time␣(Year)",
        main="Simple␣Linear␣Regression␣Model", cex.axis=0.95,
     pch= 19,xaxt='n', ylim=c(70000,3900000),
        xlim=c(1966,2016))
lines(2013:2016, A1.full[48:51,2], lty=2)
lines(diesel ~ year, A1)
axis(1, seq(1966, 2016, by=5),
    seq(1966, 2016, by=5), cex.axis = 0.95)

newx = seq(1966,2016,by = 1)
conf.int <- predict(lm1, newdata=data.frame(year=newx),
                    interval="conf",
                    level = 0.95)
pred.int <- predict(lm1, newdata=data.frame(year=newx),
                    interval="pred",
                    level = 0.95)
lines(1966:2016, pred.int[,1], col='red')
lines(newx, conf.int[,2], col="red", lty=2)
lines(newx, conf.int[,3], col="red", lty=2)
lines(newx, pred.int[,2], col="blue", lty=2)
lines(newx, pred.int[,3], col="blue", lty=2)
legend("topleft",
       legend=c("Observation", "Fitted␣Value",
```

```
        "95\% Confidence interval", "95\% Prediction interval"),
        col=c("black","red", "red", "blue"),
        text.col=c("black","red", "red","blue"),
        lty=c(1, 1, 2,2),pch = c(19, NA, NA, NA))

#QUESTION 4
lambda<-0.8
N<-nrow(A1)

#make smoothing
S<-0
e<-0
S[1]<-A1$diesel[1]
for(i in 2:(N-1)){
  S[i]<-(1-lambda)*A1$diesel[i]+lambda*S[i-1]
}

#residuals
for(i in 2:(N)){
  e[i-1]<-A1$diesel[i]-S[i-1]
}

#1-step prediction for non observable value is:
for(i in (N):(N+2)){
  S[i]<-(1-lambda)*A1$diesel[N]+lambda*S[N-1]
}

T=0
for(i in 0:46){
T<-T+lambda^i
}

F<-0
for(i in 0:46){
  F=F+lambda^i
}

SIGMA<-diag(1/(lambda^(46:0)))

sigma.var <- (t(e)%*%solve(SIGMA)%*%e)/(T-1)

var.res <- sigma.var*(1+solve(F))

upper<-0
lower<-0
for(i in 1:3){
  upper[i] = S[48]+qt(0.975, df=46)*sqrt(var.res)
  lower[i] = S[48]-qt(0.975, df=46)*sqrt(var.res)
}
```

```
#alternative confidence interval
#psi.sq<-(1-0.8)^2
#conf.lower <- 0
#conf.upper <- 0
#for(i in 1:3){
#  conf.lower[i] <- S[48]-1.959964*
                    sqrt(var(e,na.rm=T)*(1+(i-1)*psi.sq))
#  conf.upper[i] <- S[48]+1.959964*
                    sqrt(var(e,na.rm=T)*(1+(i-1)*psi.sq))
#}

plot(diesel ~ year, A1.full, ylab="Diesel␣(Tonne)",
     xlab="Time␣(Year)",
     main="Simple␣Exponential␣Smoothing", cex.axis=0.95,
     pch= 19,xaxt='n', ylim=c(70000,3900000), xlim=c(1966,2016))
lines(2013:2016, A1.full[48:51,2], col='black', lty = 2)
lines(diesel ~ year, A1)
axis(1, seq(1966, 2016, by=5), seq(1966, 2016, by=5),
    cex.axis = 0.95)
lines(1967:2016, c(S[c(1:47)], rep(S[48],3)), col='red')
lines(2014:2016, rep(S[48],3), col='blue', lty=1, lwd=2)
lines(2014:2016, upper, col="blue", lty=2, lwd=2)
lines(2014:2016, lower, col="blue", lty=2, lwd=2)
legend("topleft",
       c("Observation", "1␣Year␣Prediction",
       "95\%␣Prediction␣interval", "Forecast"),
       col=c("black","red", "blue", "blue"),
       text.col=c("black","red", "blue","blue"),
       lty=c(1, 1, 2,1),pch = c(19, NA, NA, NA), lwd=c(1,1,2,2))

#Question 1.5

L<- matrix(c(1,0,1,1), 2,2, byrow=T )
f <- function(j){
  return(rbind(1,j))
}
lambda<-0.8

F5<-matrix(0, 2, 2)
h5 <- matrix(0, 2, 1)
init<-5
for(j in 0:(init-1)){
  F5 <- F5+lambda^(j)*f(-j)%*%t(f(-j))
  h5 <- h5 + lambda^(j)*f(-j) * A1$diesel[init-j]
}

pred.one <- rep(NA,48)
for(i in 1:init){
  pred.one[i] <- NA
```

```
}

F=F5
h=h5
p<-length(h)
N<-length(A1$diesel)
residual<-matrix(NA, ncol=1, nrow=N)
theta<-matrix(NA, ncol=p, nrow=N)
theta.hat <- solve(F, h)

Linv <- solve(L)

#make 1-step predictions and estimate each theta.
for(i in (init+1):N){
  pred.one[i] <- t(f(1)) %*% theta.hat
  F <- F + lambda^(i-1)*f(-(i-1)) %*% t(f(-(i-1)))
  h <- lambda*Linv %*% h + f(0) * A1$diesel[i]
  theta.hat <- solve(F,h)
  theta[i,] <- theta.hat
}

#lav forecast, 1 -> 2 -> 3 step
fcast<-rep(NA,3)
for(i in 1:3){
  fcast[i] <- theta[48,1] + i*theta[48,2]
}

des.n<-length(A1$diesel)-init
#Finding residuals, variance, sigma
#vi skal ikke bruge de foerste 5 (husk vi kigger bagud i tid)
x<-cbind(rep(1,des.n),-(des.n-1):0)
residuals <- A1$diesel[(init+1):48] - x %*% theta.hat

T<-(1-lambda^des.n)/(1-lambda)
lambda.list<-c()
for(i in (init+1):48){
  lambda.list <- c(lambda.list,1/lambda^(48-i))
}
SIGMA<- diag(lambda.list, des.n)
sigma.est <- t(residuals) %*% solve(SIGMA) %*% residuals/(T-2)
Var <- sigma.est *(1+t(f(1))%*% solve(F) %*% f(1))

pred_high <- rep(NA,3)
pred_low <- rep(NA,3)
cal<-0
for (i in 1:3){
  cal[i] <- qt(0.975, des.n-2)*
             sqrt(sigma.est*(1+t(f(i))%*%solve(F)%*%f(i)))
  pred_high[i] <- fcast[i] + cal[i]
  pred_low[i]<- fcast[i] - cal[i]
```

```
}

e1<-residuals
par(mfrow=c(1,1))
s.e<-(e1-mean(e1))/sd(e1)
plot(pred.one[6:48],s.e, ylab="Standardized␣Residuals", xlab="In-sample␣Pre
abline(mean(s.e), 0, lty=2)
lines(lowess(s.e~pred.one[6:48]),col="red")

plot(diesel ~ year, A1.full, ylab="Diesel␣(Tonne)",
     xlab="Time␣(Year)",
     main="Local␣Linear␣Trend␣Model", cex.axis=0.95,
     pch= 19,xaxt='n', ylim=c(70000,3900000),
     xlim=c(1966,2016))
lines(2013:2016, A1.full[48:51,2], col='black', lty = 2)
lines(diesel ~ year, A1)
axis(1, seq(1966, 2016, by=5), seq(1966, 2016, by=5),
    cex.axis = 0.95)
lines(1966:2016, c(pred.one,fcast), col='red')
lines(2014:2016, fcast, col='blue', lty=1, lwd=2)
lines(2014:2016, pred_high, col="blue", lty=2, lwd=2)
lines(2014:2016, pred_low, col="blue", lty=2, lwd=2)
legend("topleft",
       c("Observation", "1␣Year␣Prediction",
       "95%␣Prediction␣interval", "Forecast"),
       col=c("black","red", "blue", "blue"),
       text.col=c("black","red", "blue","blue"),
       lty=c(1, 1, 2,1),pch = c(19, NA, NA, NA),
       lwd=c(1,1,2,2))

weight <- c(solve(SIGMA)[which(solve(SIGMA)!=0, arr.ind=T)])
plot(1971:2013, weight, ylab="Weight", xlab="Year", type='b')

#### Question 1.6
newL<-seq(0.01, 0.99, by=0.01)
SSE<-rep(NA,length(newL))
pred.two <- rep(NA,48)
init<-5
for(i in 1:init){
  pred.two[i] <- NA
}
for(k in 1:length(newL)){
F5<-matrix(0, 2, 2)
h5 <- matrix(0, 2, 1)
for(j in 0:(init-1)){
  F5 <- F5+newL[k]^(j)*f(-j)%*%t(f(-j))
  h5 <- h5 + newL[k]^(j)*f(-j) * A1$diesel[init-j]
}
F=F5
h=h5
```

```
p<-length(h)
N<-length(A1$diesel)
residual<-matrix(NA, ncol=1, nrow=N)
theta<-matrix(NA, ncol=p, nrow=N)
theta.hat <- solve(F, h)
Linv <- solve(L)

for(i in (init+1):N){
  pred.two[i] <- t(f(1)) %*% theta.hat
  F <- F + newL[k]^(i-1)*f(-(i-1)) %*% t(f(-(i-1)))
  h <- newL[k]*Linv %*% h + f(0) * A1$diesel[i]
  theta.hat <- solve(F,h)
  theta[i,] <- theta.hat
}

residuals <- A1$diesel[(init+1):48] -  pred.two[(init+1):48]
SSE[k] <- sum(residuals^2)
}

plot(newL, SSE, ylab="Residual␣Error",
     xlab="Lambda", main="Optimizing␣forgetting␣factor",
     cex.axis=0.95,
     pch= 19)
points(0.67, min(SSE), col="red", pch=19)
legend("topright",
       c("Optimal␣Forgetting␣Factor"),
       col=c("red"), text.col=c("red"),pch = c(19))


#Q1.7
#linear fits not optimal

#Now with optimal forgetting factor

L<- matrix(c(1,0,1,1), 2,2, byrow=T )
lambda<-0.67

F5<-matrix(0, 2, 2)
h5 <- matrix(0, 2, 1)
init<-5
for(j in 0:(init-1)){
  F5 <- F5+lambda^(j)*f(-j)%*%t(f(-j))
  h5 <- h5 + lambda^(j)*f(-j) * A1$diesel[init-j]
}

pred.one <- rep(NA,48)
for(i in 1:init){
  pred.one[i] <- NA
}
```

```
F=F5
h=h5
p<-length(h)
N<-length(A1$diesel)
residual<-matrix(NA, ncol=1, nrow=N)
theta<-matrix(NA, ncol=p, nrow=N)
theta.hat <- solve(F, h)

Linv <- solve(L)

#make 1-step predictions and estimate each theta.
for(i in (init+1):N){
  pred.one[i] <- t(f(1)) %*% theta.hat
  F <- F + lambda^(i-1)*f(-(i-1)) %*% t(f(-(i-1)))
  h <- lambda*Linv %*% h + f(0) * A1$diesel[i]
  theta.hat <- solve(F,h)
  theta[i,] <- theta.hat
}

#lav forecast, 1 -> 2 -> 3 step
fcast<-rep(NA,3)
for(i in 1:3){
  fcast[i] <- theta[48,1] + i*theta[48,2]
}

des.n<-length(A1$diesel)-init
#Finding residuals, variance, sigma
x<-cbind(rep(1,des.n),-(des.n-1):0)
residuals <- A1$diesel[(init+1):48] - x %*% theta.hat

T<-(1-lambda^des.n)/(1-lambda)
lambda.list<-c()
for(i in (init+1):48){
  lambda.list <- c(lambda.list,1/(lambda^(48-i)))
}
SIGMA<- diag(lambda.list, des.n)
sigma.est <- t(residuals) %*% solve(SIGMA) %*% residuals/(T-2)

pred_high <- rep(NA,3)
pred_low <- rep(NA,3)
cal<-0
for (i in 1:3){
  cal[i] <- qt(0.975, des.n-2)*sqrt(sigma.est)*
          sqrt(1+t(f(i))%*%solve(F)%*%f(i))
  pred_high[i] <- fcast[i] + cal[i]
  pred_low[i]<- fcast[i] - cal[i]
}


plot(diesel ~ year, A1.full, ylab="Diesel (Tonne)",
```

```
      xlab="Time␣(Year)", main="Local␣Linear␣Trend␣Model",
      cex.axis=0.95,
      pch= 19,xaxt='n', ylim=c(70000,4500000),
     xlim=c(1966,2016))
lines(2013:2016, A1.full[48:51,2], col='black', lty = 2)
lines(diesel ~ year, A1)
axis(1, seq(1966, 2016, by=5), seq(1966, 2016, by=5),
     cex.axis = 0.95)
lines(1966:2016, c(pred.one,fcast), col='red')
lines(2014:2016, fcast, col='blue', lty=1, lwd=2)
lines(2014:2016, pred_high, col="blue", lty=2, lwd=2)
lines(2014:2016, pred_low, col="blue", lty=2, lwd=2)
legend("topleft",
       c("Observation", "1␣Year␣Prediction",
       "95%␣Prediction␣interval", "Forecast"),
       col=c("black","red", "blue", "blue"),
       text.col=c("black","red", "blue","blue"),
       lty=c(1, 1, 2,1),pch = c(19, NA, NA, NA),
       lwd=c(1,1,2,2))

weight <- c(solve(SIGMA)[which(solve(SIGMA)!=0, arr.ind=T)])
plot(1971:2013, weight, ylab="Weight", xlab="Year", type='b')
```