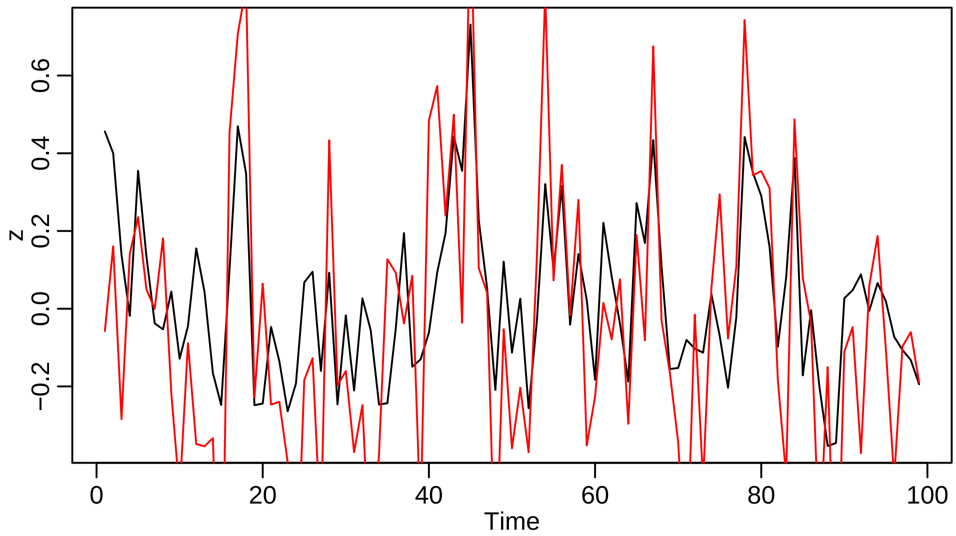
Solution to Series 12

1. a)

```
> set.seed(22)
> E <- rnorm(n=100, 0, 0.2)
> z <- E[2:100] + 0.4 * E[1:99]
> z <- ts(z)
> y <- z+rnorm(99, 0,0.3)
```

b)

```
> plot(z)
> lines(y, col="red")
```



c) The initial state X_t is

$$X_t = \begin{pmatrix} Z_t \\ E_t \end{pmatrix}$$

We thus have to find G_t, F_t, v_t^2, W_t (resp. w_t^2) such that

$$Y_t = F_t \underbrace{\begin{pmatrix} Z_t \\ E_t \end{pmatrix}}_{X_t} + V_t \quad \text{with } V_t \sim \mathcal{N}\left(0, v_t^2\right)$$

$$\underbrace{\begin{pmatrix} Z_t \\ E_t \end{pmatrix}}_{X_t} = G_t \underbrace{\begin{pmatrix} Z_{t-1} \\ E_{t-1} \end{pmatrix}}_{X_{t-1}} + W_t \quad \text{with } W_t \sim \mathcal{N}\left(0, w_t^2\right)$$

This state space formulation of the model can be achieved in the following way:

$$Y_t = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{F_t} \underbrace{\begin{pmatrix} Z_t \\ E_t \end{pmatrix}}_{X_t} + V_t$$

$$\underbrace{\begin{pmatrix} Z_t \\ E_t \end{pmatrix}}_{X_t} = \underbrace{\begin{pmatrix} 0 & 0.4 \\ 0 & 0 \end{pmatrix}}_{G_t} \underbrace{\begin{pmatrix} Z_{t-1} \\ E_{t-1} \end{pmatrix}}_{X_{t-1}} + \underbrace{\begin{pmatrix} 1 \\ 1 \end{pmatrix} E_t}_{W_t}$$

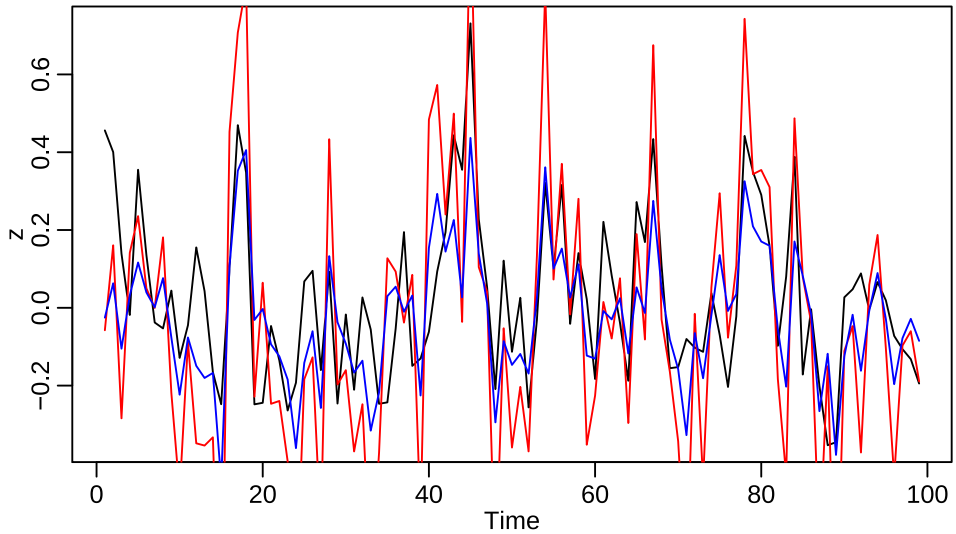
We thus have

$$
\begin{aligned}
F_t &= \begin{pmatrix} 1 & 0 \end{pmatrix} \\
G_t &= \begin{pmatrix} 0 & 0.4 \\ 0 & 0 \end{pmatrix} \\
W_t &= \begin{pmatrix} E_t \\ E_t \end{pmatrix}, \; W_t \sim \mathcal{N}\left(0, w_t^2 = \begin{pmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{pmatrix}\right) \\
V_t &\sim \mathcal{N}\left(0, v_t^2 = 0.3\right)
\end{aligned}
$$

d) The matrices we need for the SS-function are:

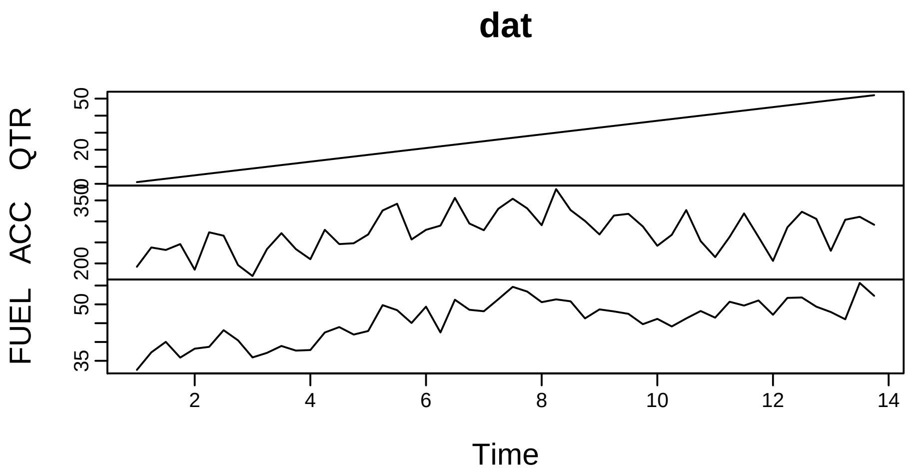
- `Fmat` $= \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- `Gmat` $= \begin{pmatrix} 0 & 0.4 \\ 0 & 0 \end{pmatrix}$
- `Vmat` $= (0.3)$
- `Wmat` $= \text{Cov}\left(\begin{pmatrix} E_t \\ E_t \end{pmatrix}\right) = \begin{pmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{pmatrix}$
- `m0` $= \mathbf{E}\left[(Z_t, \quad E_t)\right] = \mathbf{E}\left[(E_t + 0.4E_{t-1}, \quad E_t)\right] = (0, \quad 0)$
- `C0` $= \text{Cov}\left((Z_t, \quad E_t)\right) = \text{Cov}\left((E_t + 0.4E_{t-1}, \quad E_t)\right) = \begin{pmatrix} 0.2 + 0.4^2 \cdot 0.2 & 0.2 \\ 0.2 & 0.2 \end{pmatrix}$

```
> ## load the package for kalman filtering
> library(sspir)
> ## state space formulation
> ssf <- SS(y=as.matrix(y),
            Fmat=function(tt,x,phi) {return(matrix(c(1,0), nrow=2))},
            Gmat=function(tt,x,phi) {return(matrix(c(0,0,0.4,0), nrow=2))},
            Vmat=function(tt,x,phi) {return(matrix(0.3))},
            Wmat=function(tt,x,phi) {return(matrix(c(0.2,0.2,0.2, 0.2), nrow=2))},
            m0=matrix(c(0,0), nrow=1),
            C0=matrix(c(0.2+0.4^2*0.2, 0.2,0.2,0.2), nrow=2))
> fit <- kfilter(ssf)
> plot(z)
> lines(y, col="red")
> lines(fit$m[,1], col="blue")
```



2. a)

```
> batmobile <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/batmobile.dat",
                          header=T, quote="\"")
> dat       <- ts(batmobile, start=1, freq=4)
> plot(dat)
```



b) The "normal" time series regression model for the number of accidents (without time-varying coefficients) is the following:

For the season we have to construct a dummy variable:

Season	δ_2	δ_3	δ_4
Q1	0	0	0
Q2	1	0	0
Q3	0	1	0
Q4	0	0	1

$$\texttt{ACC}_t = \mu + \alpha \cdot \texttt{FUEL}_t + \beta_{season2} \cdot \delta_{2,t} + \beta_{season3} \cdot \delta_{3,t} + \beta_{season4} \cdot \delta_{4,t} + E_t$$

c)

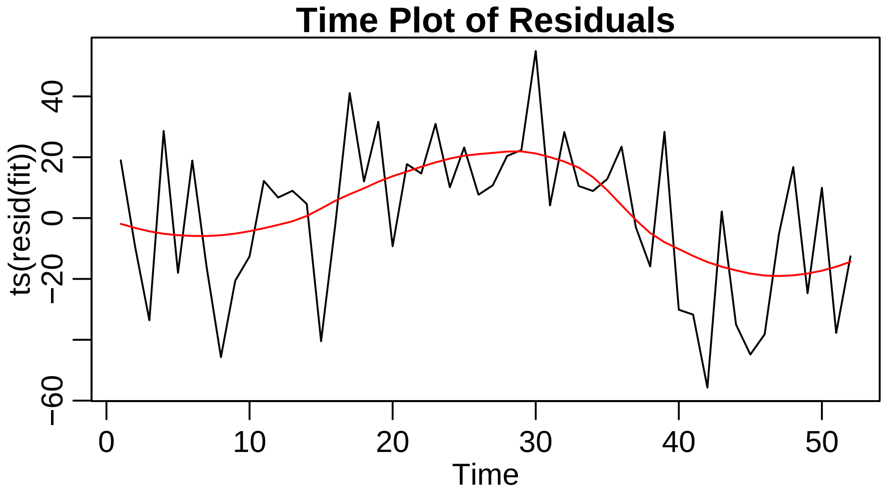
```
> regdat    <- cbind(batmobile, season=factor(cycle(dat),
                                 lab=c("Q1","Q2","Q3","Q4")))
> ## Regression
> fit <- lm(ACC ~ FUEL + season, data=regdat)
> summary(fit)
```

```
Call:
lm(formula = ACC ~ FUEL + season, data = regdat)

Residuals:
    Min      1Q  Median      3Q     Max
-55.724 -16.554   7.217  18.025  54.907

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1636    30.9102  -0.005   0.9958
FUEL          5.3140     0.6827   7.783 5.37e-10 ***
seasonQ2     49.5403    10.3170   4.802 1.64e-05 ***
seasonQ3     53.0029    10.5988   5.001 8.39e-06 ***
seasonQ4     26.9848    10.3373   2.610   0.0121 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.18 on 47 degrees of freedom
Multiple R-squared:  0.715,       Adjusted R-squared:  0.6908
F-statistic: 29.48 on 4 and 47 DF,  p-value: 2.745e-12
```

d) Residual checking:

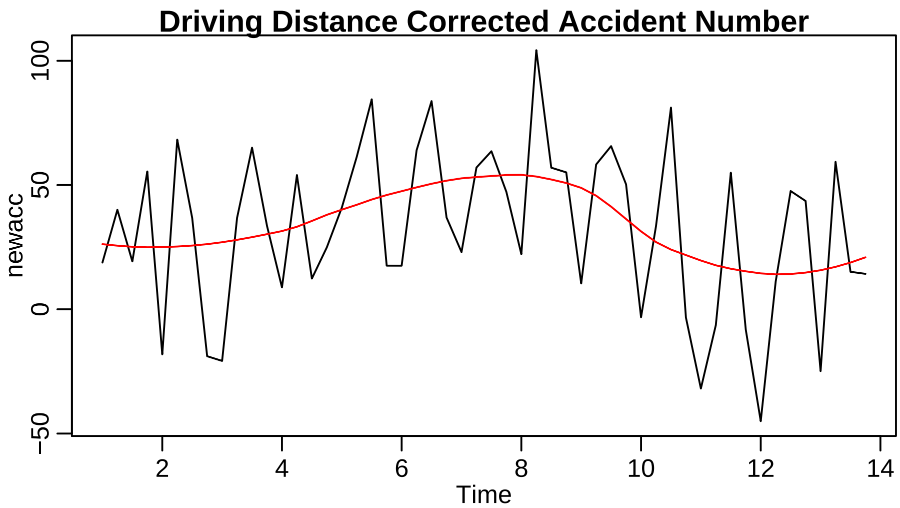
```
> times     <- 1:52
> fit.loess <- loess(resid(fit) ~ times, span=0.65, degree=2)
> par(mfrow=c(1,1)); plot(ts(resid(fit)))
```

```
> lines(times, fitted(fit.loess), col="red")
> title("Time Plot of Residuals")
```

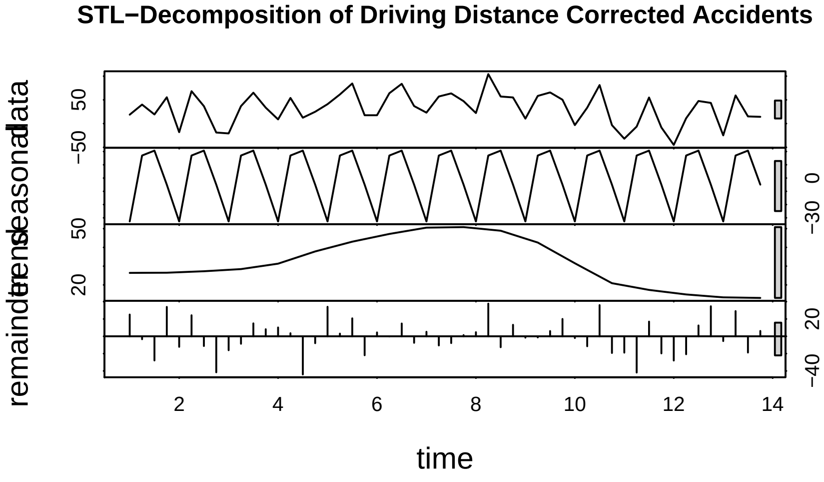


The residuals look correlated. There is no trust in the standard errors and p-values.

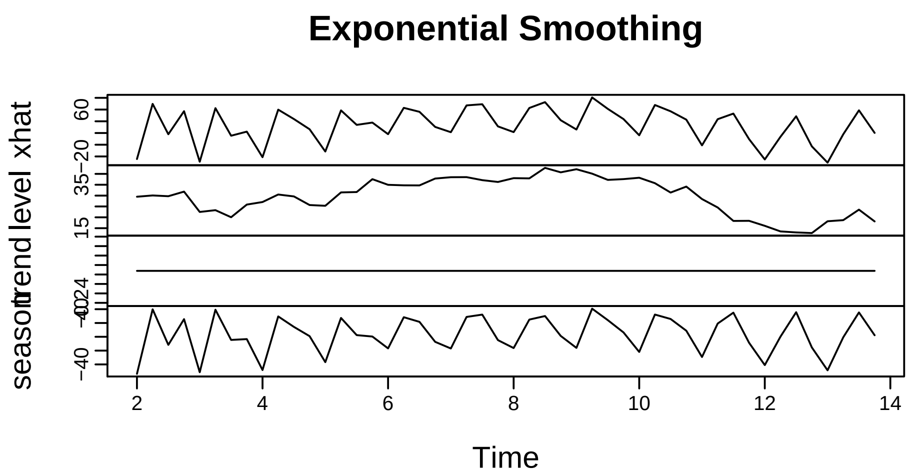
e)
```
> newacc <- ts(batmobile$ACC-coef(fit)[2]*batmobile$FUEL, freq=4)
> plot(newacc, main="Driving Distance Corrected Accident Number")
> fit.loess <- loess(newacc ~ time(newacc), span=0.65, degree=2)
> lines(as.numeric(time(newacc)), fitted(fit.loess), col="red")
```



f)
```
> stl.fit <- stl(newacc, s.window="periodic", t.window=23)
> plot(stl.fit, main="STL-Decomposition of Driving Distance Corrected Accidents")
> acf(stl.fit$time.series[,3])
```



g)
```
> ## Exponential Smoothing für die um die Fahrleistung korrigierten Unfallzahlen
> plot(HoltWinters(newacc)$fitted, main = "Exponential Smoothing")
```



h) The regression model is

$$\texttt{ACC}_t = \mu_t + \alpha\texttt{FUEL}_t + \beta_{season2}\cdot\delta_{2,t} + \beta_{season3}\cdot\delta_{3,t} + \beta_{season4}\cdot\delta_{4,t} + E_t$$

OLS regression models can always be formulated in state space form in the follwoing way.

$$\underbrace{\texttt{ACC}_t}_{Y_t} = \underbrace{\begin{pmatrix}1, & \delta_{2,t} & \delta_{3,t} & \delta_{4,t}\end{pmatrix}}_{F_t} \underbrace{\begin{pmatrix}\mu_t \\ \beta_{season2} \\ \beta_{season3} \\ \beta_{season4}\end{pmatrix}}_{X_t} + \underbrace{E_t}_{V_t}$$

$$\underbrace{\begin{pmatrix}\mu_t \\ \beta_{season2} \\ \beta_{season3} \\ \beta_{season4}\end{pmatrix}}_{X_t} = \underbrace{\begin{pmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1\end{pmatrix}}_{G_t} \underbrace{\begin{pmatrix}\mu_{t-1} \\ \beta_{season2} \\ \beta_{season3} \\ \beta_{season4}\end{pmatrix}}_{X_{t-1}} + \underbrace{\begin{pmatrix}\triangle\mu_t \\ 0 \\ 0 \\ 0\end{pmatrix}}_{W_t}$$

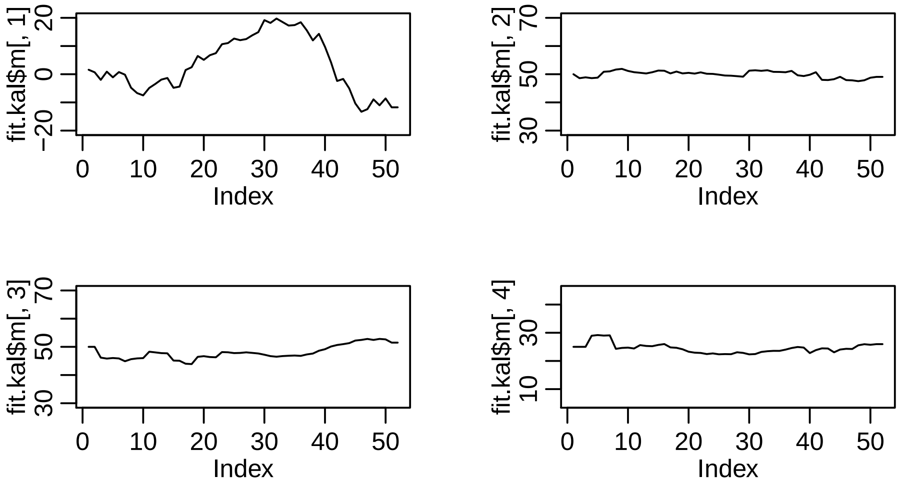
i)
```
> regdat2     <- cbind(newacc, season=factor(cycle(dat), lab=c("Q1","Q2","Q3","Q4")))
> regdat2 <- data.frame(regdat2)
> regdat2$season <- as.factor(regdat2$season)
> fit2 <- lm(newacc~season, data=regdat2)
> summary(fit2)
```

```
Call:
lm(formula = newacc ~ season, data = regdat2)

Residuals:
    Min      1Q  Median      3Q     Max 
-55.724 -16.554   7.217  18.025  54.907 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.1636     7.1843  -0.023   0.9819    
season2      49.5403    10.1601   4.876 1.23e-05 ***
season3      53.0029    10.1601   5.217 3.84e-06 ***
season4      26.9848    10.1601   2.656   0.0107 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 25.9 on 48 degrees of freedom
Multiple R-squared:  0.4204,	Adjusted R-squared:  0.3842 
F-statistic: 11.61 on 3 and 48 DF,  p-value: 7.706e-06
```

Plugging in "roughly" the estimates of the fit from b) we get:

- Gmat $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- Wmat $= w_t^2 = \mathrm{Cov}(W_t) = \begin{pmatrix} 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
- Vmat $= v_t^2 = \mathrm{Var}(V_t) = 26.18^2 \approx 650$
- m0 $= \begin{pmatrix} \mu_0, & \beta_{season2}, & \beta_{season3}, & \beta_{season4} \end{pmatrix} \approx \begin{pmatrix} 0, & 50, & 50, & 25 \end{pmatrix}$
- C0 $= \mathrm{Var}(X_t) = \begin{pmatrix} \mathrm{Var}(\mu) & 0 & 0 & 0 \\ 0 & \mathrm{Var}(\beta_{season2}) & 0 & 0 \\ 0 & 0 & \mathrm{Var}(\beta_{season3}) & 0 \\ 0 & 0 & 0 & \mathrm{Var}(\beta_{season4}) \end{pmatrix}$
  $\approx \begin{pmatrix} 50 & 100 & 100 & 100 \end{pmatrix}$
- Fmat $= \begin{pmatrix} 1, & \delta_{2,t}, & \delta_3 3, t, & \delta_{4,t} \end{pmatrix}$

```
> ## State Space Model for fuel corrected data
> library(sspir)
> fit     <- lm(newacc ~ regdat$season)
> y.mat   <- as.matrix(newacc)
> x.mat   <- model.matrix(fit)
> ssf     <- SS(y=y.mat, x=x.mat,
               Fmat=function(tt,x,phi) return(t(x[tt,,drop=FALSE])),
               Gmat=function(tt,x,phi) return(diag(4)),
               Wmat=function(tt,x,phi) return(diag(c(10,0,0,0))),
               Vmat=function(tt,x,phi) return(matrix(650)),
               m0=matrix(c(0,50,50,25),1,4),
               C0=diag(c(50,100,100,100)))
> fit.kal <- kfilter(ssf)
> par(mfrow=c(2,2))
> plot(fit.kal$m[,1], type="l", ylim=c(-20,20))
> plot(fit.kal$m[,2], type="l", ylim=c(30,70))
> plot(fit.kal$m[,3], type="l", ylim=c(30,70))
> plot(fit.kal$m[,4], type="l", ylim=c(5,45))
```



j) The regression model for the original data is

$$\mathtt{ACC}_t = \mu_t + \alpha \cdot \mathtt{FUEL}_t + \beta_{season2} \cdot \delta_{2,t} + \beta_{season3} \cdot \delta_{3,t} + \beta_{season4} \cdot \delta + 4, t + E_t$$

OLS regression models can always be formulated in state space form in the follwoing way.

$$\underbrace{\mathtt{ACC}_t}_{Y_t} = \underbrace{\begin{pmatrix} 1, & \mathtt{FUEL}_t, & \delta_{2,t} & \delta_{3,t} & \delta_{4,t} \end{pmatrix}}_{F_t} \underbrace{\begin{pmatrix} \mu_t \\ \alpha \\ \beta_{season2} \\ \beta_{season3} \\ \beta_{season4} \end{pmatrix}}_{X_t} + \underbrace{E_t}_{V_t}$$

$$\underbrace{\begin{pmatrix} \mu_t \\ \alpha \\ \beta_{season2} \\ \beta_{season3} \\ \beta_{season4} \end{pmatrix}}_{X_t} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{G_t} \underbrace{\begin{pmatrix} \mu_{t-1} \\ \alpha \\ \beta_{season2} \\ \beta_{season3} \\ \beta_{season4} \end{pmatrix}}_{X_{t-1}} + \underbrace{\begin{pmatrix} \triangle\mu_t \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{W_t}$$

```
> ## State Space Model für die Originaldaten
> library(sspir)
> fit     <- lm(ACC ~ FUEL + season, data=regdat)
> summary(fit)
Call:
lm(formula = ACC ~ FUEL + season, data = regdat)

Residuals:
    Min      1Q  Median      3Q     Max
-55.724 -16.554   7.217  18.025  54.907

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1636    30.9102  -0.005   0.9958
FUEL          5.3140     0.6827   7.783 5.37e-10 ***
seasonQ2     49.5403    10.3170   4.802 1.64e-05 ***
seasonQ3     53.0029    10.5988   5.001 8.39e-06 ***
seasonQ4     26.9848    10.3373   2.610   0.0121 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.18 on 47 degrees of freedom
```

```
Multiple R-squared:  0.715,        Adjusted R-squared:  0.6908
F-statistic: 29.48 on 4 and 47 DF,  p-value: 2.745e-12
```

```
> y.mat   <- as.matrix(regdat$ACC)
> x.mat   <- model.matrix(fit)
> ssf     <- SS(y=y.mat, x=x.mat,
               Fmat=function(tt,x,phi) return(t(x[tt,,drop=FALSE])),
               Gmat=function(tt,x,phi) return(diag(5)),
               Wmat=function(tt,x,phi) return(diag(c(10,0,0,0,0))),
               Vmat=function(tt,x,phi) return(matrix(600)),
               m0=matrix(c(0,5,50,50,25),1,5),
               C0=diag(c(900,1,100,100,100)))
> fit.kal <- kfilter(ssf)
> par(mfrow=c(2,2))
> plot(fit.kal$m[,1], type="l", ylim=c(-20,20))
> plot(fit.kal$m[,2], type="l", ylim=c(4.75, 5.64))
> plot(fit.kal$m[,3], type="l", ylim=c(30,70))
> plot(fit.kal$m[,4], type="l", ylim=c(30,70))
> plot(fit.kal$m[,5], type="l", ylim=c(5,45))
```