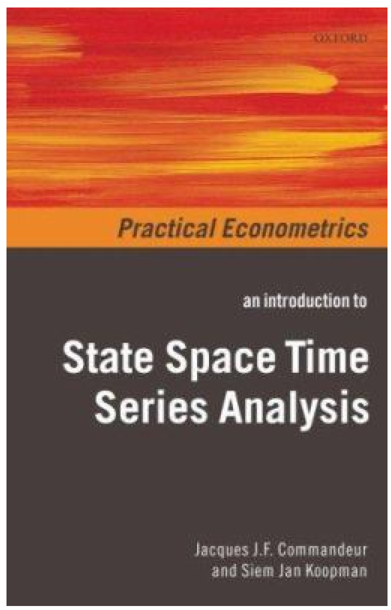
An Introduction to State Space Time Series Analysis Summary

RK

October 27, 2013

Abstract

The purpose of this document is to summarize the book "An Introduction to State Space Time Series Analysis" and provide the supporting R code to work with the book.



Contents

Book Summary

State Space Methodology serves as an umbrella for representing many univariate, multivariate stationary and non stationary time series. For those who have never heard of a "State Space Model" but have used some software for any time series model parameter estimation, the motivation is this : It is likely that the software used has a state space representation of the model in the implementation. For example, in `R`, the following description for the function `arima` says,

> The exact likelihood is computed via a state-space representation of the ARIMA process, and the innovations and their variance found by a Kalman filter. The initialization of the differenced ARMA process uses stationarity and is based on Gardner et al. (1980). For a differenced process the non-stationary components are given a diffuse prior (controlled by kappa). Observations which are still controlled by the diffuse prior (determined by having a Kalman gain of at least 1e4) are excluded from the likelihood calculations. (This gives comparable results to arima0 in the absence of missing values, when the observations excluded are precisely those dropped by the differencing.)

A model as simple as AR(1) for which OLS might suffice is also cast as a state space model in R. Why do you think so?. What are the specific advantages of representing many econometric models in the form of states?. For a econometrics newbie, to make sense of the above statement would mean that he/she understands the importance of Kalman filter in the following aspects

- Predicting
- Filtering
- Estimating
- Forecasting

This book does not explain all the gory details of Kalman Filter or State Space models as the authors make it very clear in the introduction that the content in the book is meant to serve only as an appealing introduction to time series using state space methods. The book does even assume that the reader is conversant with Box Jenkins type of informal time series analysis . All a reader is expected to know is some classic linear regression fundas. Time Series analysis has the primary task to uncover the dynamic evolution of the observations measured over time. It is assumed that the dynamic properties cannot be observed directly from data. The unobserved dynamic process at time t is referred to as the *state* of the time series. The state of a time series may consist of several components. The book is organized in such each chapter deals with one such component.

The first seven chapters of the book are meant to give the reader a preliminary understanding of some state space models. The first six chapters explore one or more combinations of state variables that find a place in the final model in chapter 7. The components explored in the chapters leading to the final model are

- Stochastic Level
- Stochastic Trend
- Seasonality
- Intervention Variables
- Explanatory variables

The model described in chapter 7 combines all the above components and hence in order to understand all the aspects of final model, it is better to go over all the preliminary chapters. I don't think how anyone can merely read up the chapters and numbers given, like a novel. If you see a bunch of numbers for any model, natural inclination would be either to run the code that goes along with it(if the authors have provided) or at least make an effort to get close to verifying them by writing your own code. So, in that sense if some one merely reads the first seven chapters like a novel, I guess the understanding will be somewhat shallow.Having said that, the authors should have pushed the first seven chapters after chapter 10 where they introduce the software that they have used to generate numbers and visuals. Actually , the book should have started with chapter 8 that provides the general framework and then connect various models with in that framework. Well, who can say what's the right way to introduce a subject?. May be seeing a bunch of models and visuals might be motivating for some readers.

In my opinion, it is better to start reading chapter 8 that provides the basic equation of a state space model.There are many notations for state space framework and I am comfortable with the following notation, that is different from the one mentioned in the book.My mind is used to seeing F and G matrices and so I tend to stick with this representation.

$$
\begin{aligned}
Y_t &= F_t\theta_t + v_t, \quad v_t \sim N(0, V_t) \\
\theta_t &= G_t\theta_{t-1} + w_t, \quad w_t \sim N(0, W_t)
\end{aligned}
$$

The first is the observation equation and second describes the state equation. All the models discussed in the first seven chapters can be represented in the above form and hence one can use Kalman filter algo for estimating, filtering , smoothing and forecasting.In chapter 10, the authors mention the software used for the analysis. The authors use `SsfPack` and `Ox` to churn out the numbers and visuals. `SsfPack` is basically a bunch of C routines that can be linked to `Ox`. Never have I used the above software, nor do I have the necessary skills to use such packages. Hence went about doing the analysis in `R`. There are at least five to six CRAN packages that are targeted towards state space models. However the one I found the most easy to understand is `dlm` package. In fact the contributors of the package have written a fantastic book on the same.

This book serves as an excellent primer to state space models. However having some preliminary knowledge of Kalman filter helps in getting to understand the content well. One thing missing in this book is the mention of MCMC sampling. The standard approach to estimation of parameters in the model is MLE. However there is an alternative method: use MCMC to get a posterior distribution of the entire state vector and parameter space. This approach is not dealt in this book. In fact one can try to verify all the numbers mentioned in the book from a Bayesian perspective, i.e use Gibbs sampling to generate the posterior distribution and check whether the model estimates and state vector estimates fall in the credible interval of the marginal posterior distribution of the parameter. That would be an interesting exercise to do and check one's understanding of Bayesian analysis and state space models.

What's in this document ?

I will try to provide the `R` code for various model estimates and the figures given in each of the chapters. I have tried in keeping the figure numbers same as the ones found in the book so that it is easy to refer to any figure in the text and look up for the relevant `R` code that needs to be written to generate the visual. There are close to 90 figures in the book and this document contains code for close to 75 odd figures.

1 Introduction

This chapter gives a brief recap of classical linear regression model using a dataset that is a time series of log of monthly number of drivers killed or seriously injured in UK from Jan 1969 to Dec 1984.

```
> data            <- log(read.table("data/C1/UKdriversKSI.txt",skip=1))
> colnames(data) <- "logKSI"
> t               <- 1:dim(data)[1]
> fit            <- lm(data$logKSI~t)
> (coefs         <- round(as.numeric(coef(fit)),8))
[1]  7.54584273 -0.00144803
> (error.var     <- round(summary(fit)$sigma^2,8))
[1] 0.02299806
> (f.stat        <- round(summary(fit)$fstatistic[1],8))
   value
53.77447
```

Estimates

- Coefficients 7.54584273 and -0.00144803
- Error Variance is 0.02299806
- F Statistic is 53.77446552

```
> par(mfrow=c(1,1))
> plot(data$logKSI,  col = "darkgrey", xlab="",ylab = "log KSI",pch=3,cex=0.5,
      cex.lab=0.8,cex.axis=0.7,xlim=c(0,200))
> abline(coef(lm(data$logKSI~t)) , col = "blue", lwd  = 2, lty=2)
> legend("topright",leg = c("log UK drivers KSI against time(in months)",
                          "  regression line"), cex = 0.6,
                             lty = c(0, 2), col = c("darkgrey","blue"),
                             pch=c(3,NA), bty = "y",horiz = T)
```

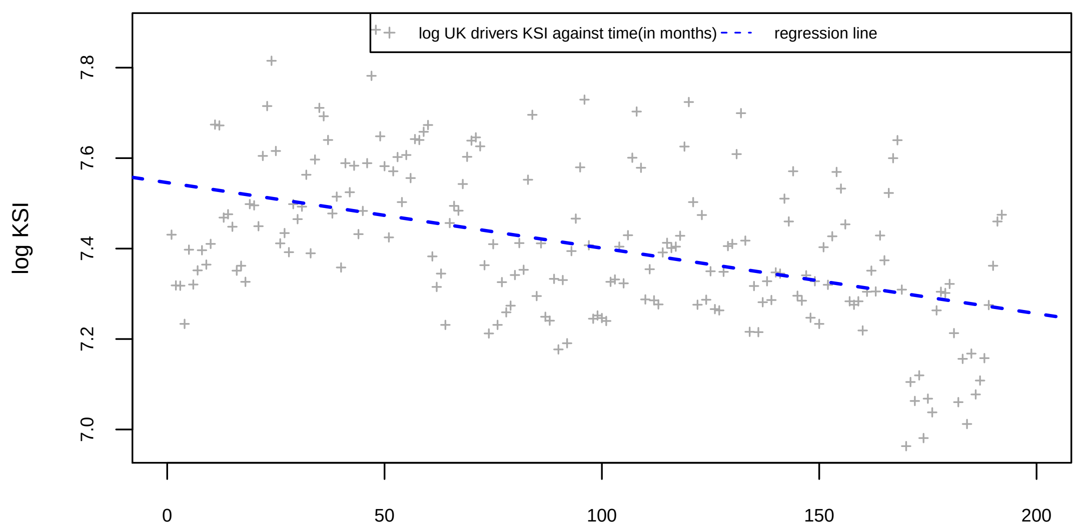


Figure 1.1: Scatter plot of the log of the number of UK drivers KSI against time (in months), including regression line.

```
> par(mfrow=c(1,1))
> plot(ts(data$logKSI),ylab="",xlab="",xlim = c(0,200), col = "darkgrey")
```

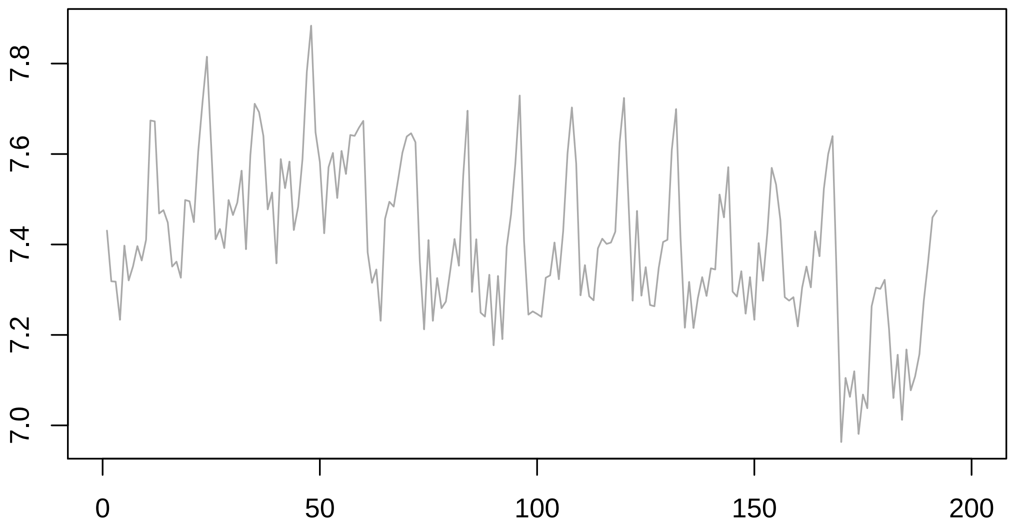


Figure 1.2: Log of the number of UK drivers KSI plotted as a time series.

```
> par(mfrow=c(1,1))
> plot(ts(residuals(fit)),ylab="",xlab="",xlim = c(0,200), col = "darkgrey",lty=2)
> abline(h=0)
```

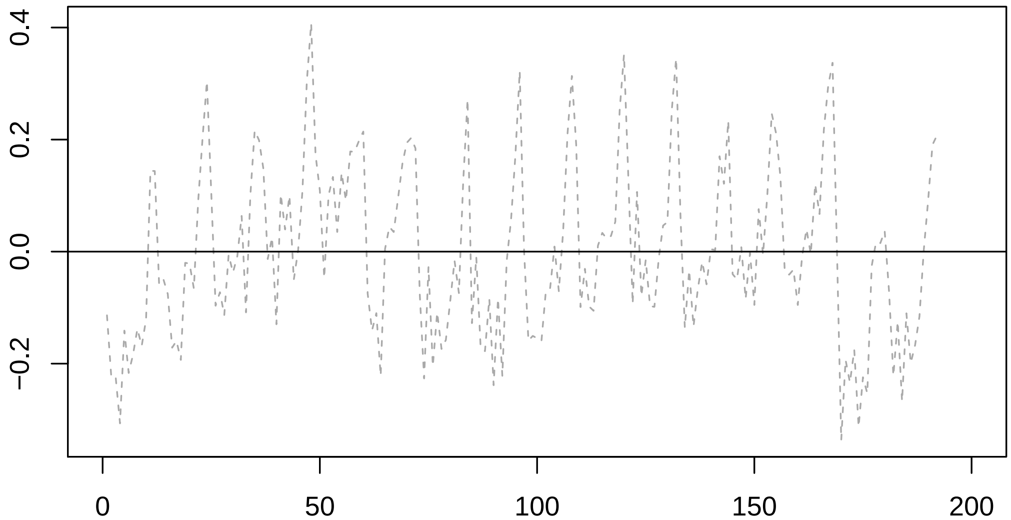


Figure 1.3: Residuals of classical linear regression of the log of the number of UK drivers KSI on time.

Diagnostics

```
> par(mfrow=c(1,1))
> random.series       <- rnorm(dim(data)[1])
> acf(c(random.series),15,main ="")
```

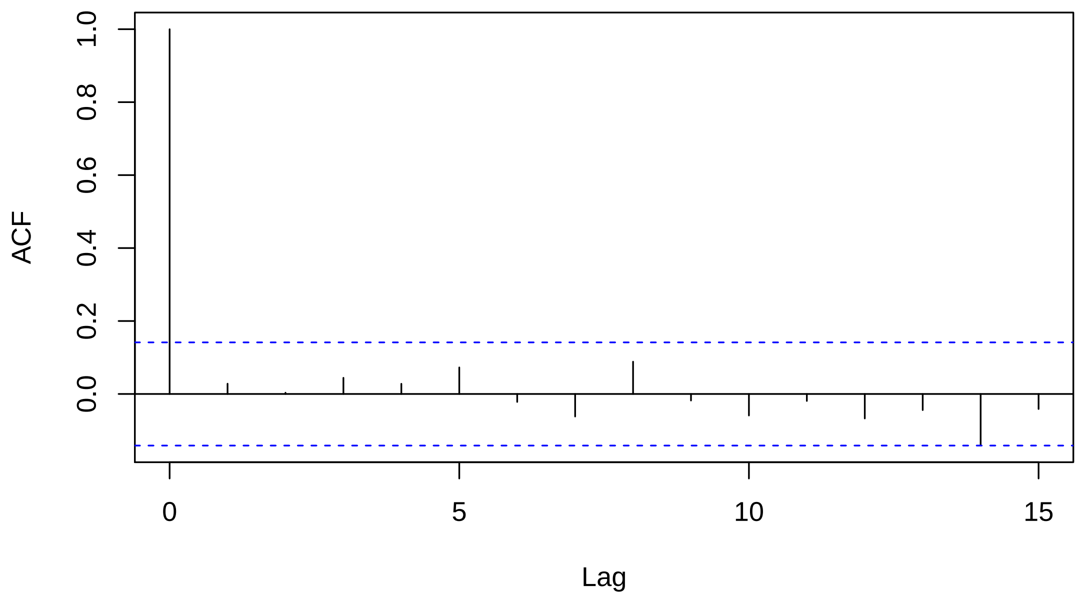


Figure 1.4: Correlogram of random time series.

```
> par(mfrow=c(1,1))
> residuals <- residuals(fit)
> acf(c(residuals),15,main="")
```

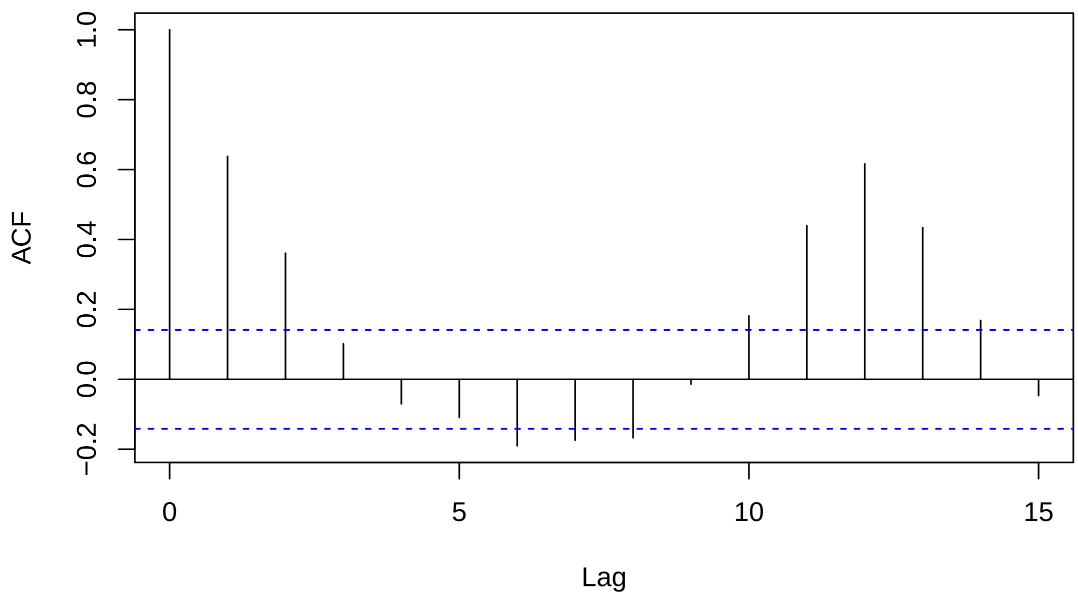


Figure 1.5: Correlogram of classical regression residuals.

The point of illustrating all these diagrams is to drive home this basic point : Inference based on Least squares runs in to problems when the residuals are autocorrelated. In an autocorrelated process, the variance of the least squares model either underestimates or overestimates the true error variance. In the former case, t and F stats are going to more and hence you tend to be over optimistic about the model. In the latter case, the t stat and F stats are going to be less than their true values and hence you tend to be overly pessimistic about the model

2 The Local level Model

This chapter explores *Local level Model* that goes by the following structure:

$$y_t = \mu_t + \epsilon_t \qquad \epsilon_t \sim N(0, \sigma_\epsilon^2)$$
$$\mu_{t+1} = \mu_t + \xi_t \qquad \xi_t \sim N(0, \sigma_\xi^2)$$

The first equation represents the structure for the observed data and the second equation governs the state equation. This model is also called the "random walk plus noise" model. There are variations of the above model that are explored in the chapter. First variation is $\sigma_\xi^2 = 0$, which leads the usual OLS model. Second is $\sigma_\xi^2 > 0$ which leads to random walk plus noise.
The two datasets that are used in the chapter :

```
> data.1            <- log(read.table("data/C1/UKdriversKSI.txt",skip=1))
> colnames(data.1)  <- "logKSI"
> data.1            <- ts(data.1, start = c(1969),frequency=12)
> data.2            <- log(read.table("data/C2/NorwayFinland.txt",skip=1))
> data.2            <- data.2[,2,drop=F]
> colnames(data.2)  <- "logNorFatalities"
> data.2            <- ts(data.2 , start = c(1970,1))
```

2.1 Deterministic Level

```
> fit               <- lm(data.1[,1]~1)
> res               <- residuals(fit)
> (coefs            <- round(as.numeric(coef(fit)),8))
[1] 7.406108
> (error.var        <- round(summary(fit)$sigma^2,8))
[1] 0.02935256
```

Estimates

- Coefficients 7.4061076
- Error Variance is 0.02935256

```
> par(mfrow=c(1,1))
> plot.ts(c(data.1),  col = "darkgrey", xlab="",ylab = "log KSI",pch=3,cex=0.5,
      cex.lab=0.8,cex.axis=0.7,xlim=c(0,200))
> abline(h=coefs[1] , col = "blue", lwd  = 2, lty=2)
> legend("topright",leg = c("log UK drivers KSI",
                          "  deterministic level"),cex = 0.6,
                             lty = c(1, 2),col = c("darkgrey","blue"),
                             pch=c(3,NA), bty = "y", horiz = T)
```

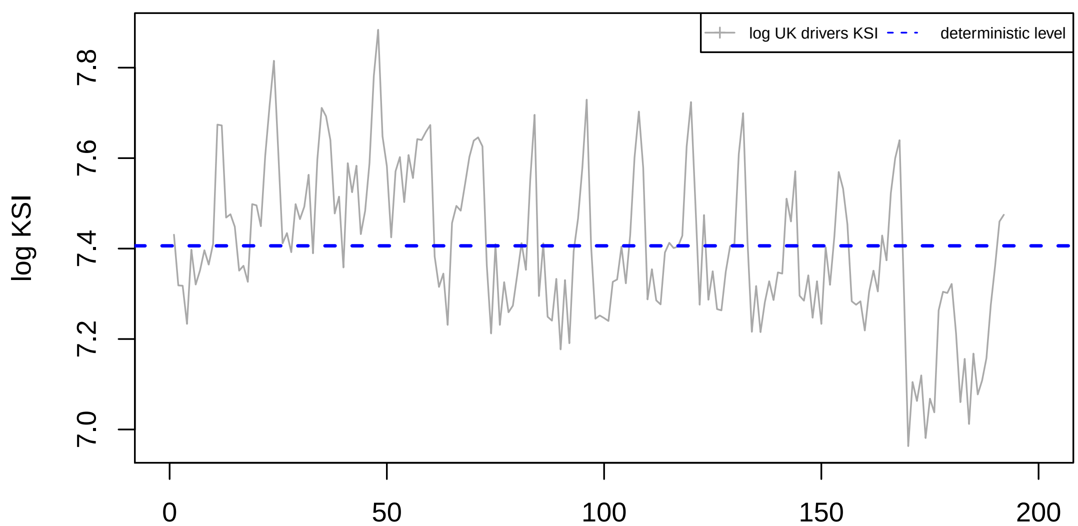


Figure 2.1: Deterministic level.

```
> par(mfrow=c(1,1))
> plot(ts(residuals(fit)),ylab="",xlab="",xlim = c(0,200), col = "darkgrey",lty=2)
> abline(h=0)
```

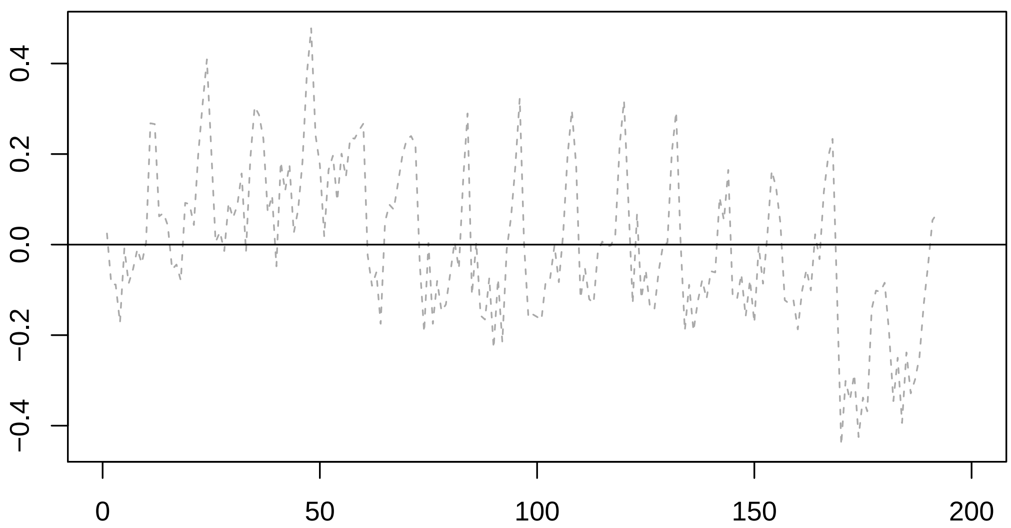


Figure 2.2: Irregular component for deterministic level model.

Diagnostics

- Normality test - PASS

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9902, p-value = 0.2127
```

- Independence - FAIL

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 474.8935, df = 15, p-value < 2.2e-16
> sapply(1:20,function(l){Box.test(res, lag=l, type = "Ljung-Box")$p.value})
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Thus clearly the residuals are autocorrelated and model is a suspect.

2.2 Stochastic Level

To fit a stochastic level, there are several packages in `R` that can be used. I have used `dlm` for estimating the model parameters

```
> fn  <- function(params){
   dlmModPoly(order= 1, dV= exp(params[1]) , dW = exp(params[2]))

 }
> y                <- c(data.1)
> fit              <- dlmMLE(y, rep(0,2),fn)
> mod              <- fn(fit$par)
> (obs.error.var   <- V(mod))
            [,1]
[1,] 0.002221571
> (state.error.var <- W(mod))
           [,1]
[1,] 0.01186592
> (mu.1            <- dropFirst(  dlmFilter(y,mod) $m )[1])
[1] 7.430707
> res              <- residuals(dlmFilter(y,mod),sd=F)
> filtered         <- dlmFilter(y,mod)
> smoothed         <- dlmSmooth(filtered)
> mu               <- dropFirst(smoothed$s)
> mu.1             <- mu[1]
```

Estimates

- Observation eq error variance is 0.00222157

- State eq error variance is 0.01186592
- MLE of the initial value is 7.41495196

```
> par(mfrow=c(1,1))
> plot.ts(y,  col = "darkgrey", xlab="",ylab = "log KSI",pch=3,cex=0.5,
     cex.lab=0.8,cex.axis=0.7,xlim=c(0,200))
> lines(mu , col = "blue", lwd  = 2, lty=2)
> legend("topright",leg = c("log UK drivers KSI","  stochastic level"),
       cex = 0.6,lty = c(1, 2), col = c("darkgrey","blue"),
       pch=c(3,NA), bty = "y", horiz = T)
```

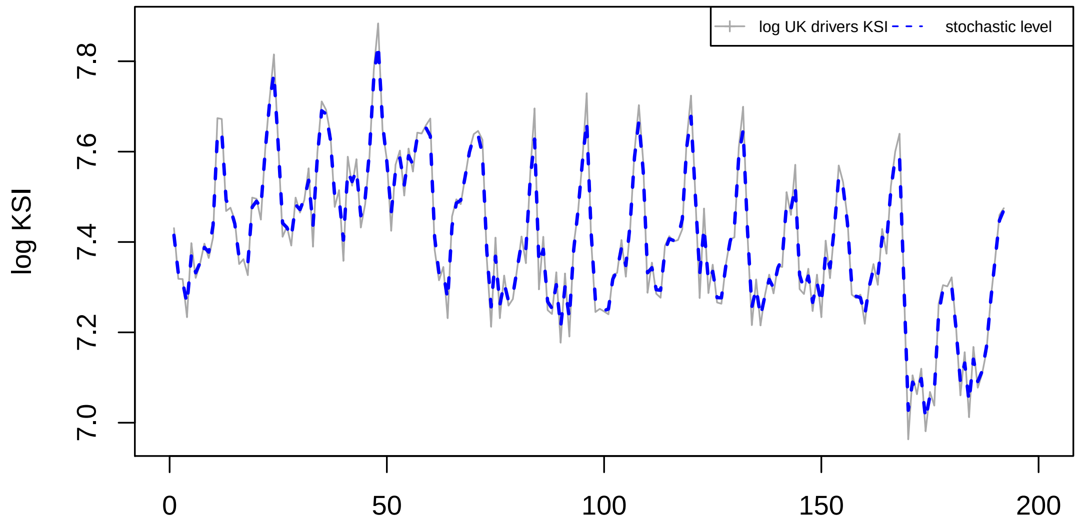


Figure 2.3: Stochastic level.

```
> par(mfrow=c(1,1))
> plot.ts(res,ylab="",xlab="", col = "darkgrey", main = "",cex.main = 0.8)
> abline(h=0)
```

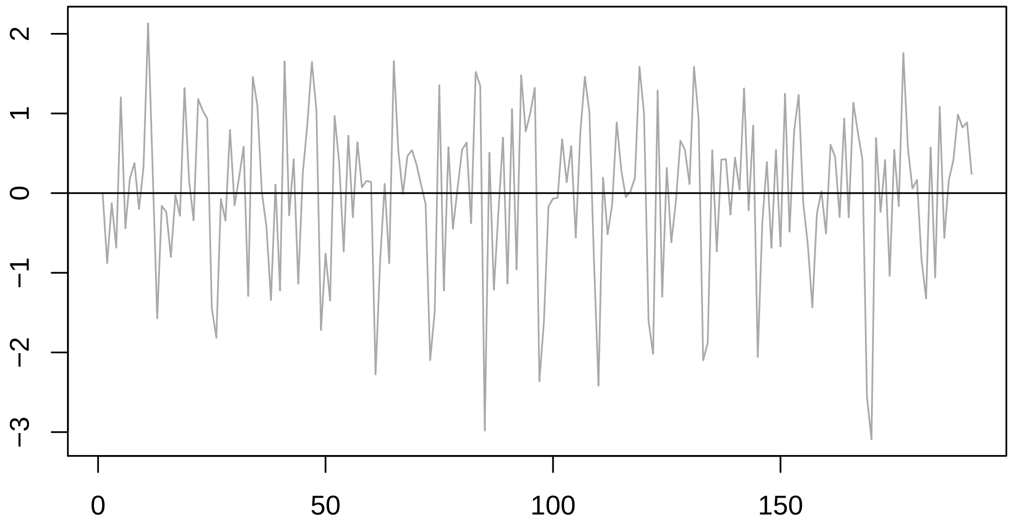


Figure 2.4: Irregular component for local level model.

Diagnostics

- Normality test - FAIL

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9713, p-value = 0.0005683
```

- Independence - FAIL

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 105.8983, df = 15, p-value = 9.992e-16
> sapply(1:20,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.9047 0.8605 0.1864 0.0171 0.0338 0.0152 0.0245 0.0013 0.0011 0.0015
[11] 0.0003 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

Thus the first few lags are ok but later lags violate independent assumptions.

2.3 The local level model and Norwegian fatalities

```
> fn  <- function(params){
  dlmModPoly(order= 1, dV= exp(params[1]) , dW = exp(params[2]))

 }
> fit              <- dlmMLE(data.2, rep(0,2),fn)
> mod              <- fn(fit$par)
> (obs.error.var   <- V(mod))
            [,1]
[1,] 0.003268212
> (state.error.var <- W(mod))
            [,1]
[1,] 0.004703001
> filtered         <- dlmFilter(data.2,mod)
> smoothed         <- dlmSmooth(filtered)
> mu               <- dropFirst(smoothed$s)
> mu.1             <- mu[1]
> res              <- residuals(filtered,sd=F)
>
```

Estimates

- Observation eq error variance is 0.00326821
- State eq error variance is 0.004703
- MLE of the initial value is 6.3048035

```
> par(mfrow=c(1,1))
> temp            <- window(cbind(data.2,mu))
> plot(temp , plot.type="single" , col =c("darkgrey","blue"),lty=c(1,2),
    xlab="",ylab = "log KSI")
> legend("topright",leg = c("log UK drivers KSI","  stochastic level"),
      cex = 0.7, lty = c(1, 2),
      col = c("darkgrey","blue"),pch=c(3,NA),bty = "y", horiz = T)
```

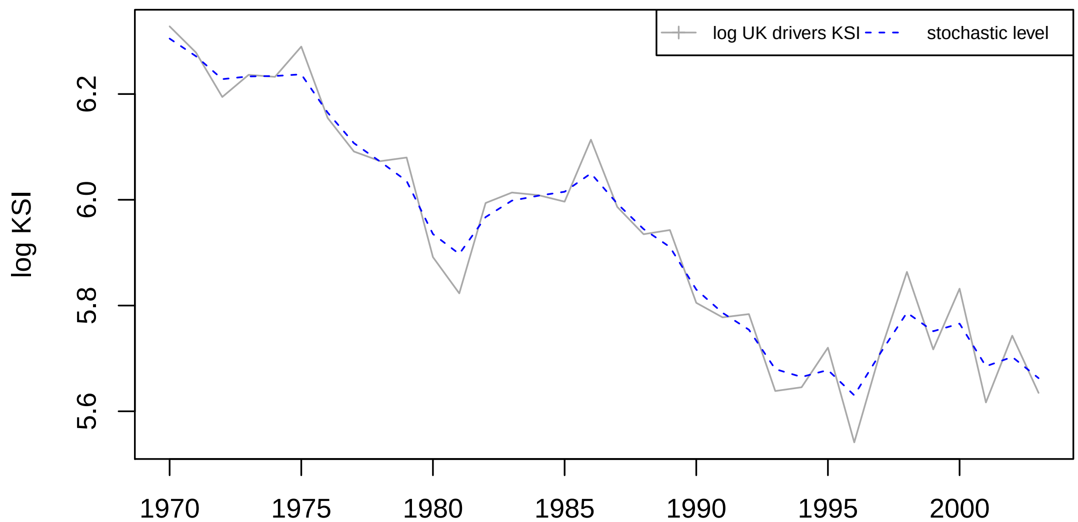


Figure 2.5: Stochastic level for Norwegian fatalities.

```
> par(mfrow=c(1,1))
> plot(res,ylab="",xlab="", col = "darkgrey",lty=2,type="l")
> abline(h=0)
```

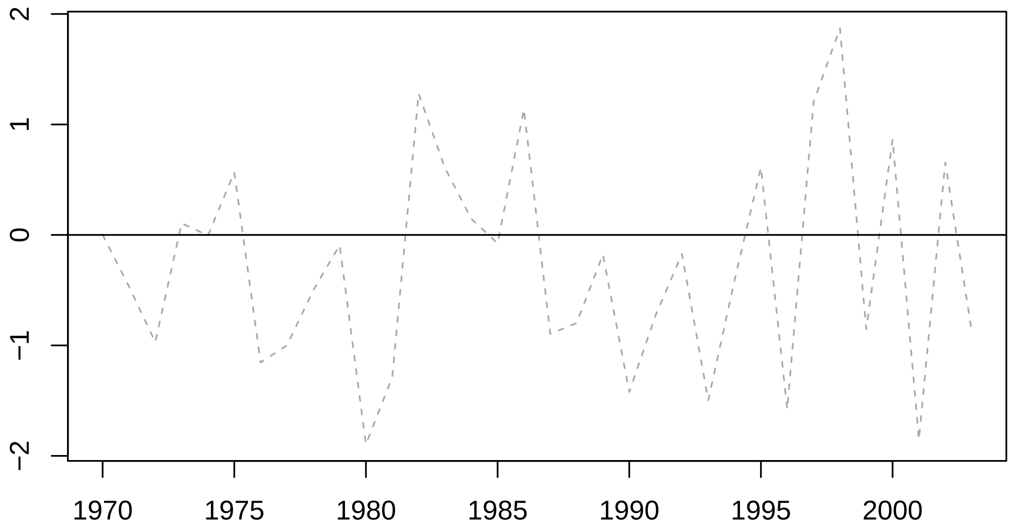


Figure 2.6: Irregular component for Norwegian fatalities.

Diagnostics

- Normality test - PASS

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9765, p-value = 0.6602
```

- Independence - PASS

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 9.1902, df = 15, p-value = 0.8674
> sapply(1:20,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.4326 0.7300 0.7701 0.8164 0.8110 0.5921 0.5667 0.6600 0.7416 0.7534
[11] 0.8020 0.8006 0.7753 0.8255 0.8674 0.7212 0.6076 0.6303 0.6756 0.7335
```

3 The local linear trend model

$$y_t = \mu_t + \epsilon_t \qquad \epsilon_t \sim N(0, \sigma_\epsilon^2)$$
$$\mu_{t+1} = \mu_t + \nu_t + \xi_t \qquad \xi_t \sim N(0, \sigma_\xi^2)$$
$$\nu_{t+1} = \nu_t + \zeta_t \qquad \zeta_t \sim N(0, \sigma_\zeta^2)$$

There are two datasets that are used in the chapter.

```
> data.1             <- log(read.table("data/C3/UKdriversKSI.txt",skip=1))
> colnames(data.1)  <- "logKSI"
> data.1             <- ts(data.1, start = c(1969),frequency=12)
> data.2             <- log(read.table("data/C2/NorwayFinland.txt",skip=1))
> data.2             <- data.2[,3,drop=F]
> colnames(data.2)  <- "logNorFatalities"
> data.2             <- ts(data.2 , start = c(1970,1))
```

3.1 Deterministic Level

```
> t              <- 1:dim(data.1)[1]
> y              <-  c(data.1)
> fit            <- lm(y~t)
> res            <- residuals(fit)
> (coefs         <- round(as.numeric(coef(fit)),4))
[1]  7.5458 -0.0014
> (error.var     <- round(summary(fit)$sigma^2,6))
[1] 0.022998
```

Estimates

- Coefficients 7.5458, -0.0014
- Error Variance is 0.022998

Diagnostics

- Normality test FAIL

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9853, p-value = 0.04319
```

- Independence - FAIL

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 300.148, df = 15, p-value < 2.2e-16
```

```
> sapply(1:15,function(l){Box.test(res, lag=l, type = "Ljung-Box")$p.value})
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Thus clearly the residuals are autocorrelated and model is a suspect.

3.2 Stochastic Level and Stochastic slope

Using `StructTS` function

```
> fit      <-  StructTS(y,"trend")
> (coefs   <- fit$coef)
      level       slope     epsilon
0.011989769 0.000000000 0.002150853
```

Using `dlm` package
To fit a stochastic level, there are several packages in `R` that can be used. I have used `dlm` for estimating the model parameters

```
> level0             <- y[1]
> slope0             <- mean(diff(y))
> fn                 <- function(params){
                         dlmModPoly(dV = exp(params[1]), dW = exp(params[2:3]),
                         m0 = c(level0, slope0),C0 = 2*diag(2))
 }
> fit                <- dlmMLE(y, rep(0,3),fn)
> mod                <- fn(fit$par)
> (obs.error.var     <- V(mod))
            [,1]
[1,] 0.002118253
> W(mod)
           [,1]         [,2]
[1,] 0.01212771 0.000000e+00
[2,] 0.00000000 1.518358e-11
> (state.error.var.1 <- W(mod)[1,1])
[1] 0.01212771
> (state.error.var.2 <- W(mod)[2,2])
[1] 1.518358e-11
> filtered           <- dlmFilter(y,mod)
> smoothed           <- dlmSmooth(filtered)
> sm                 <- dropFirst(smoothed$s)
> mu.1               <- sm[1,1]
> nu.1                <-sm[1,2]
> mu                 <- c(sm[,1])
> nu                 <- c(sm[,2])
> res                <- c(residuals(filtered,sd=F))
```

Estimates

- Observation eq error variance is 0.00211825
- Level eq error variance is 0.01212771
- Slope eq error variance is 0

- MLE of the initial value of level state is 7.41574825
- MLE of the initial value of slope state is 0.00028841

```
> par(mfrow=c(1,1))
> temp            <- window(cbind(data.1,mu),start = 1969, end = 1984)
> plot(temp , plot.type="single" , col =c("darkgrey","blue"),lty=c(1,2),
     xlab="",ylab = "log KSI")
> legend("topright",leg = c("log UK drivers KSI","  stochastic level and Slope"),
                           cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
                           pch=c(3,NA),bty = "y", horiz = T,
                          )
```

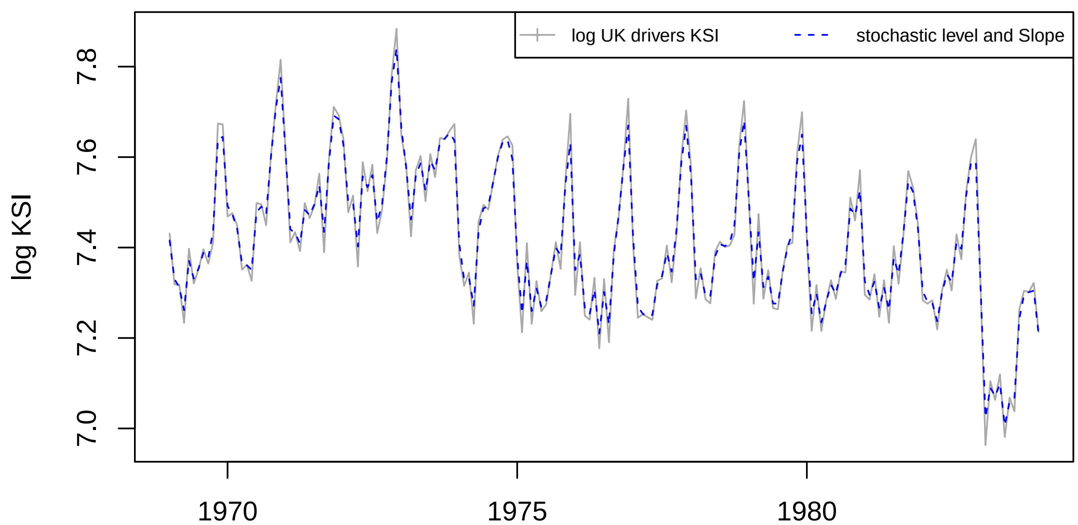


Figure 3.1: Trend of stochastic linear trend model.

```
> par(mfrow=c(1,1))
> plot(ts(nu*10^4,start =1969,end=1984,frequency =12),xlab="",
      col = "darkgrey",lty=2,ylab=expression(10^(-4)))
>
```

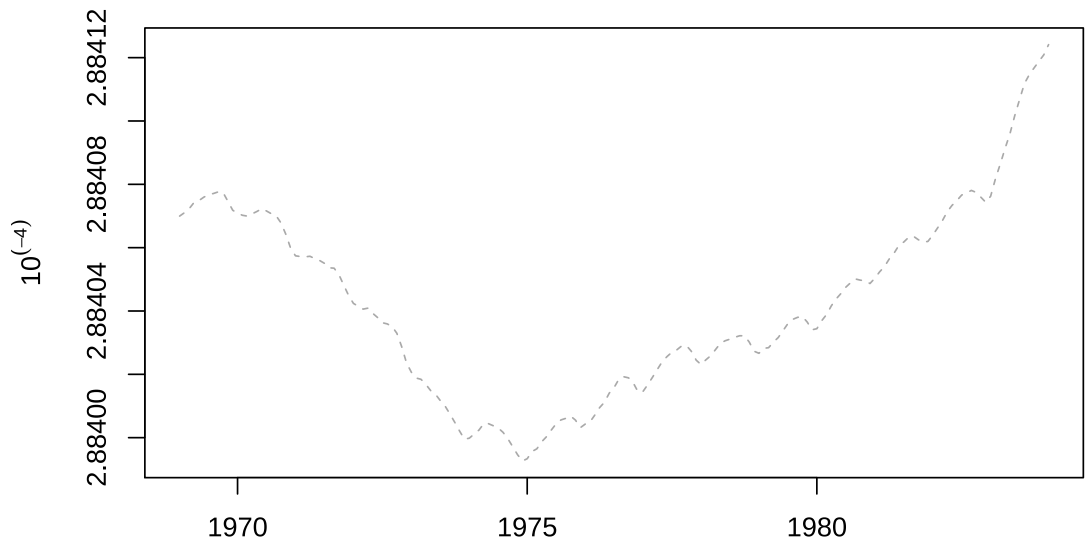


Figure 3.2: Slope of stochastic linear trend model.

```
> par(mfrow=c(1,1))
> plot(ts(res,c(1969),frequency=12),ylab="",xlab="", col = "darkgrey",
     lty=2, main = 'Irregular component', cex.main = 0.7)
> abline(h=0)
```

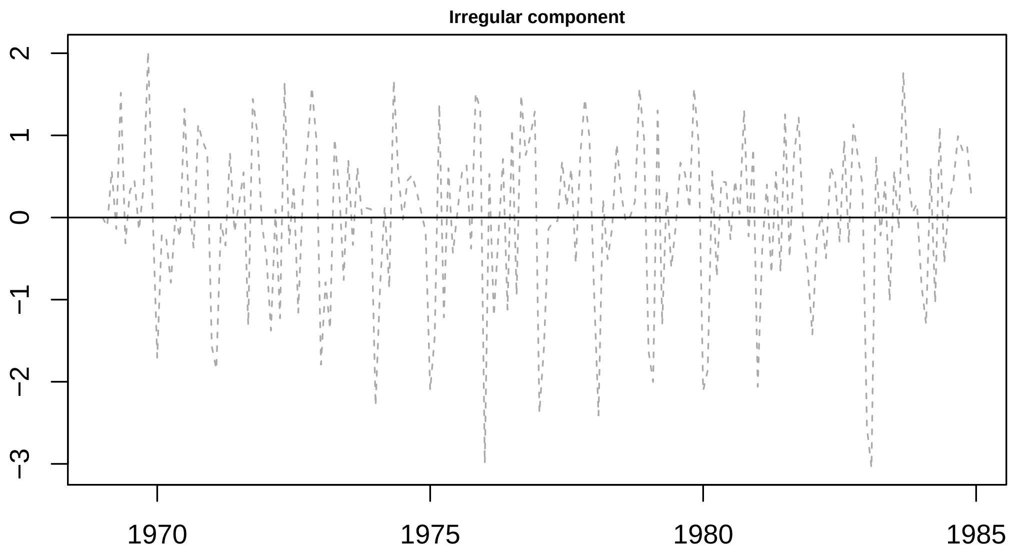


Figure 3.3: Irregular component of stochastic linear trend model.

Diagnostics

- Normality test - FAIL

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9666, p-value = 0.0001563
```

- Independence - FAIL

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 101.8532, df = 15, p-value = 5.773e-15
> sapply(1:15,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.9507 0.8869 0.2293 0.0230 0.0436 0.0212 0.0346 0.0023 0.0020 0.0025
[11] 0.0007 0.0000 0.0000 0.0000 0.0000
```

Thus the first few lags are ok but later lags violate independent assumptions.

3.3 Stochastic level model and Deterministic Slope

```
> fn                  <- function(params){
                          dlmModPoly(dV = exp(params[1]), dW = c(exp(params[2]),0))
 }
> fit                 <- dlmMLE(y, rep(0,2),fn)
> mod                 <- fn(fit$par)
> (obs.error.var      <- V(mod))
            [,1]
[1,] 0.002118081
> (state.error.var.1 <- W(mod)[1,1])
[1] 0.01212834
> filtered            <- dlmFilter(y,mod)
> (mu.1             <- dropFirst(  dlmFilter(y,mod) $m )[1,1])
[1] 7.430707
> (nu.1             <-  tail(dropFirst(  dlmFilter(y,mod) $m )[,2],1))
[1] 0.0002889568
> mu                  <- ((filtered$m)[,1])[-1]
> nu                  <- ((filtered$m)[,2])[-1]
> res                 <- c(residuals(filtered,sd=F))
> filtered            <- dlmFilter(y,mod)
> smoothed            <- dlmSmooth(filtered)
> sm                  <- dropFirst(smoothed$s)
> mu.1                <- sm[1,1]
> nu.1                 <-sm[1,2]
> mu                  <- c(sm[,1])
> nu                  <- c(sm[,2])
> res                 <- c(residuals(filtered,sd=F))
>
```

Estimates

- Observation eq error variance is 0.00211808
- Level equation error variance is 0.01212834
- MLE of the initial value of level state is 7.41573577
- MLE of the initial value of slope state is 0.00028945

```
> par(mfrow=c(1,1))
> temp              <- window(cbind(data.1,mu),start = 1969, end = 1984)
> plot(temp , plot.type="single" , col =c("darkgrey","blue"),lty=c(1,2),
     xlab="",ylab = "log KSI", )
> legend("topright",leg = c("log UK drivers KSI","  stochastic level and deterministic slope"),
                          cex = 0.6, lty = c(1, 2),col = c("darkgrey","blue"),
                          pch=c(3,NA),bty = "y", horiz = T,
                         )
```

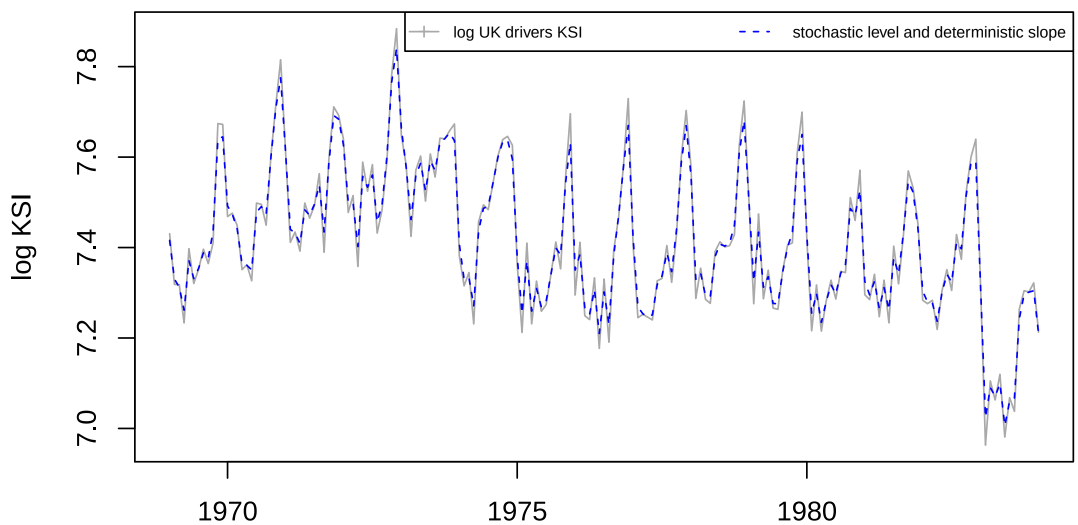


Figure 3.4: Trend of stochastic linear trend model.

3.4 The local linear trend model and Finnish fatalities

Using `StructTS` function

```
> y <- c(data.2)
> (coefs<-StructTS(y,"trend")$coef)
      level       slope     epsilon
0.000000000 0.001505921 0.003096982
```

Estimates

- Observation eq error variance is 0.00309698
- Level equation error variance is 0
- Slope equation error variance is 0.00150592

Refit the model as a deterministic level and stochastic slope model

```
> fn       <- function(params){
             dlmModPoly(dV = exp(params[1]), dW = c(0,exp(params[2])))
 }
> fit                <- dlmMLE(data.2, rep(0,2),fn)
> mod                <- fn(fit$par)
> (obs.error.var     <- V(mod))
            [,1]
[1,] 0.003200851
> (state.error.var.1 <- W(mod)[2,2])
[1] 0.001533121
> filtered           <- dlmFilter(data.2,mod)
> smoothed           <- dlmSmooth(filtered)
> sm                 <- dropFirst(smoothed$s)
> mu.1               <- sm[1,1]
> nu.1                <-sm[1,2]
> mu                 <- c(sm[,1])
> nu                 <- c(sm[,2])
> res                <- c(residuals(filtered,sd=F))
```

Estimates

- Observation eq error variance is 0.00320085
- Slope error variance is 0.00153312
- MLE of the initial value of level state is 7.01334332
- MLE of the initial value of slope state is 0.00684645

```
> par(mfrow=c(1,1))
> plot(data.2,  col = "darkgrey", xlab="",ylab = "log KSI",type="l",pch=3,cex=0.5,
      cex.lab=0.8,cex.axis=0.7)
> lines(mu.1+nu.1 , col = "blue", lwd  = 1, lty=2)
> legend("topright",leg = c("log fatalities Finland",
                          "deterministic level, stochastic slope"),cex = 0.6,
                           lty = c(1, 2),col = c("darkgrey","blue"),
                           pch=c(3,NA), bty = "y", horiz = T,
                            )


> par(mfrow=c(1,1))
> plot(1970:2003,nu,ylab="",xlab="", col = "darkgrey",
      lty=2,type="l",main = "" ,cex.main = 0.7)
> abline(h=0)
```

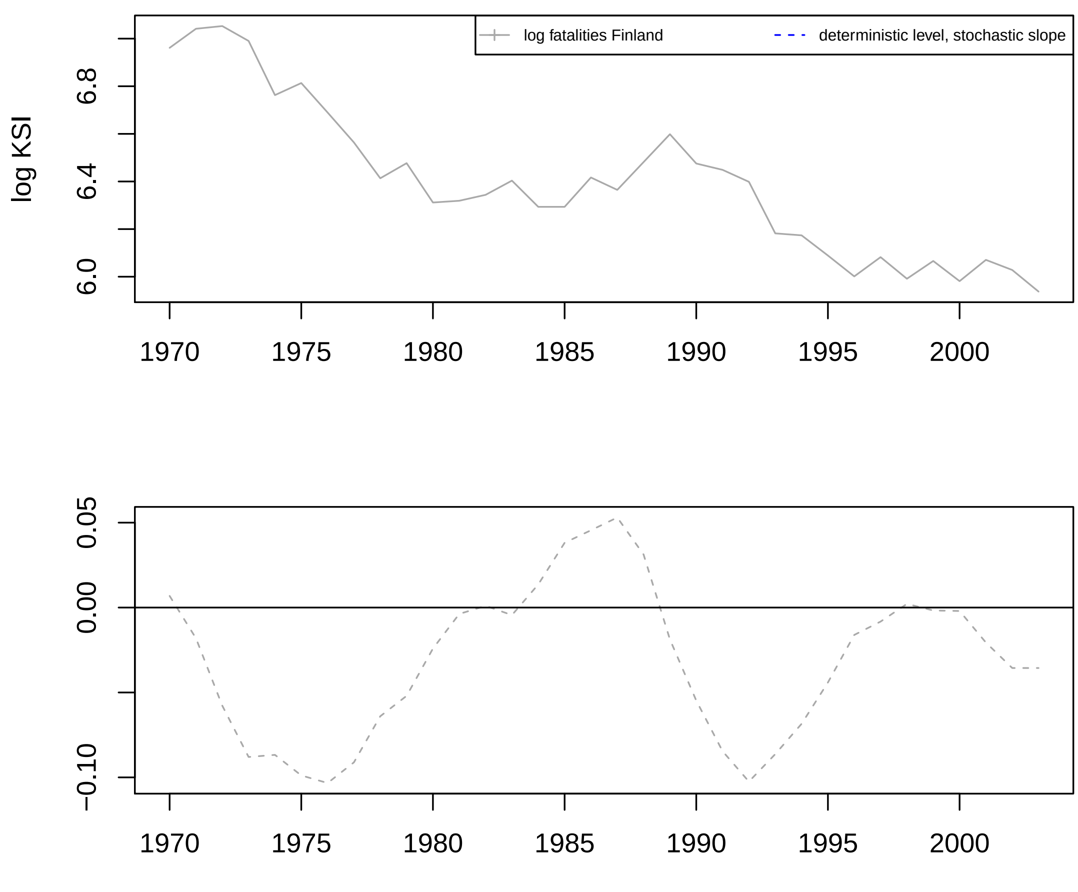


Figure 3.5: Trend of deterministic level and stochastic slope model for Finnish fatalities(top), and stochastic slope component (bottom).

```
> par(mfrow=c(1,1))
> plot(1970:2003,res,ylab="",xlab="", col = "darkgrey",
     lty=2,type="l",main = "Residuals" ,cex.main = 0.7)
> abline(h=0)
```

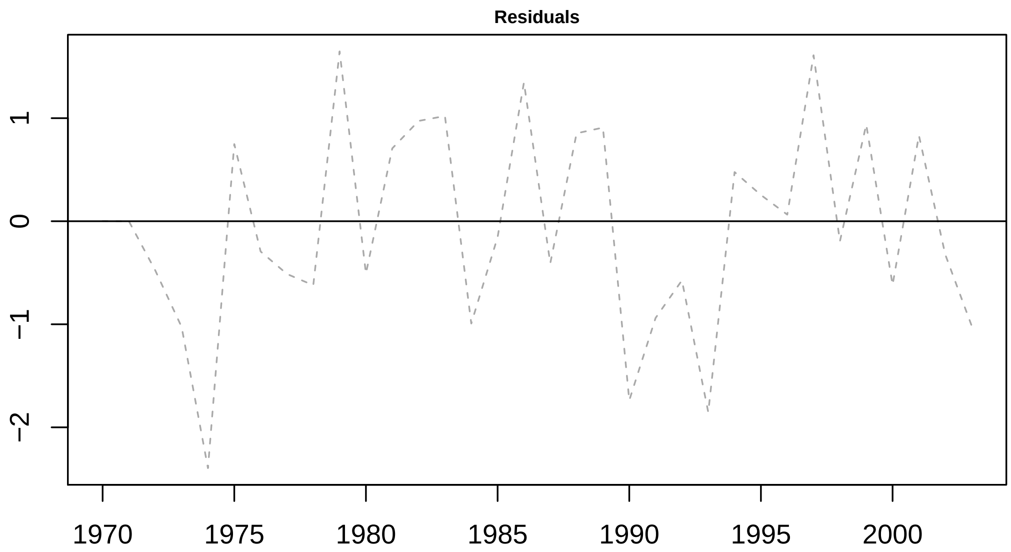


Figure 3.6: Irregular component for Finnish fatalities.

Diagnostics

- Normality test - PASS

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9714, p-value = 0.5005
```

- Independence - PASS

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 10.045, df = 15, p-value = 0.8169
> sapply(1:15,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.8585 0.9189 0.7769 0.8286 0.4640 0.5761 0.6690 0.6441 0.6218 0.7054
[11] 0.7731 0.7835 0.7672 0.7617 0.8169
```

4 The local level model with seasonal

This chapter deals with observations driven by two state variables, one being the level variable and second being the seasonality variable. The model takes the following form

$$
\begin{aligned}
y_t &= \mu_t + \gamma_t + \epsilon_t & \epsilon_t &\sim N(0, \sigma^2_\epsilon) \\
\mu_{t+1} &= \mu_t + \xi_t & \xi_t &\sim N(0, \sigma^2_\xi) \\
\gamma_{1,t+1} &= -\gamma_{1,t} - \gamma_{2,t} - \gamma_{3,t} + \omega_t, & \omega_t &\sim\sim N(0, \sigma^2_\omega) \\
\gamma_{2,t+1} &= \gamma_{1,t} \\
\gamma_{3,t+1} &= \gamma_{2,t}
\end{aligned}
$$

There are two datasets that are used in the chapter.

```
> data.1            <- log(read.table("data/C4/UKdriversKSI.txt",skip=1))
> colnames(data.1)  <- "logKSI"
> data.1            <- ts(data.1, start = c(1969),frequency=12)
> data.2            <- read.table("data/C4/UKinflation.txt",skip=1)
> colnames(data.2)  <- "IR"
> data.2            <- ts(data.2 , start = 1950,frequency  = 4)
```

```
> par(mfrow=c(1,1))
> plot(data.1,  col = "darkgrey", xlab="",ylab = "log KSI",pch=3,cex=0.5,
      cex.lab=0.8,cex.axis=0.7)
> abline(v=1969:2003, lty= "dotted",col="sienna")
>
```

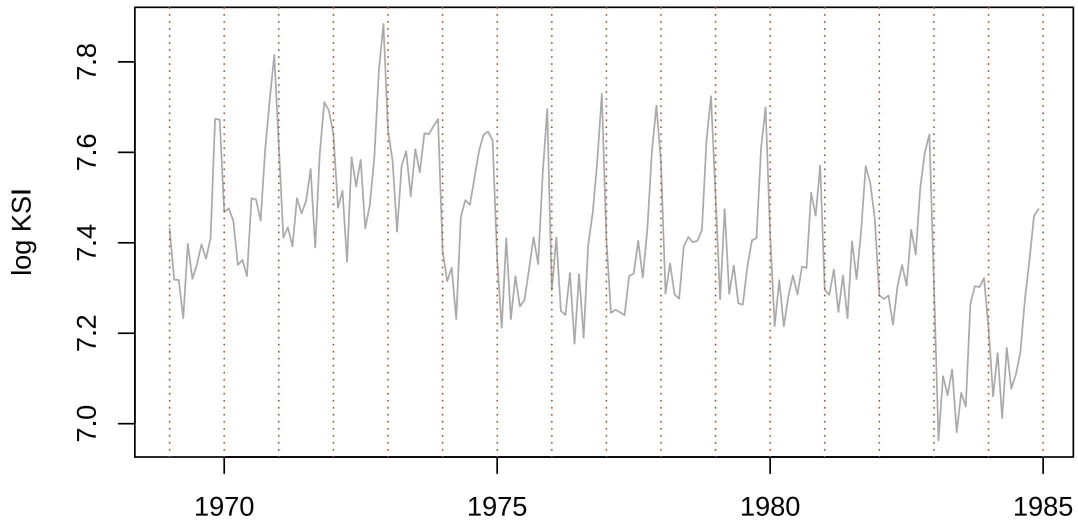


Figure 4.1: Log of number of UK drivers KSI with time lines for years.

4.1 Deterministic level and seasonal

```
> temp <- cbind(data.1, rep(c(1:12),times= 16))
> fit <- lm(temp[,1] ~ as.factor(temp[,2]))
> (coefs          <- round(as.numeric(coef(fit)),8))
 [1]  7.42826254 -0.12741266 -0.08907726 -0.16675216 -0.07626917 -0.11420957
 [7] -0.06578303 -0.05553695 -0.01852728  0.06152429  0.16598222  0.22020236
> (obs.error.var       <- round(summary(fit)$sigma^2,8))
[1] 0.01758853
```

Estimates

- Observation eq error variance is 0.01758853
- μ_1 is 7.42826254

```
> par(mfrow=c(1,1))
> fit.val <- ts(fitted(fit),start = c(1969),frequency = 12)
> temp1 <- cbind(data.1, fit.val)
> plot(temp1 , plot.type="single" , col =c("darkgrey","blue"),lty=c(1,2),
      xlab="",ylab = "log KSI")
> legend("topright",leg = c("log UK drivers KSI"," deterministic level and seasonal"),
                              cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
                              pch=c(3,NA),bty = "y", horiz = T,
                             )
```

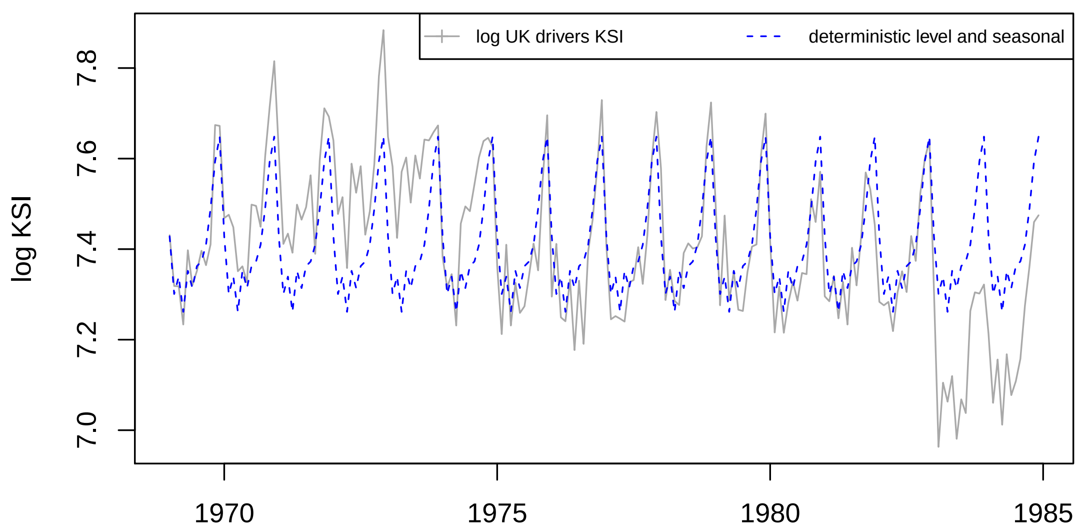


Figure 4.2: Combined deterministic level and seasonal.

```
> par(mfrow=c(1,1))
> plot(temp1[,1] , plot.type="single" , col =c("darkgrey","blue"),lty=c(1,2),
     xlab="",ylab = "log KSI")
> abline(h=coefs[1], col="sienna")
> legend("topright",leg = c("log UK drivers KSI","  deterministic level "),
                            cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
                            pch=c(3,NA),bty = "y", horiz = T,
                           )
```

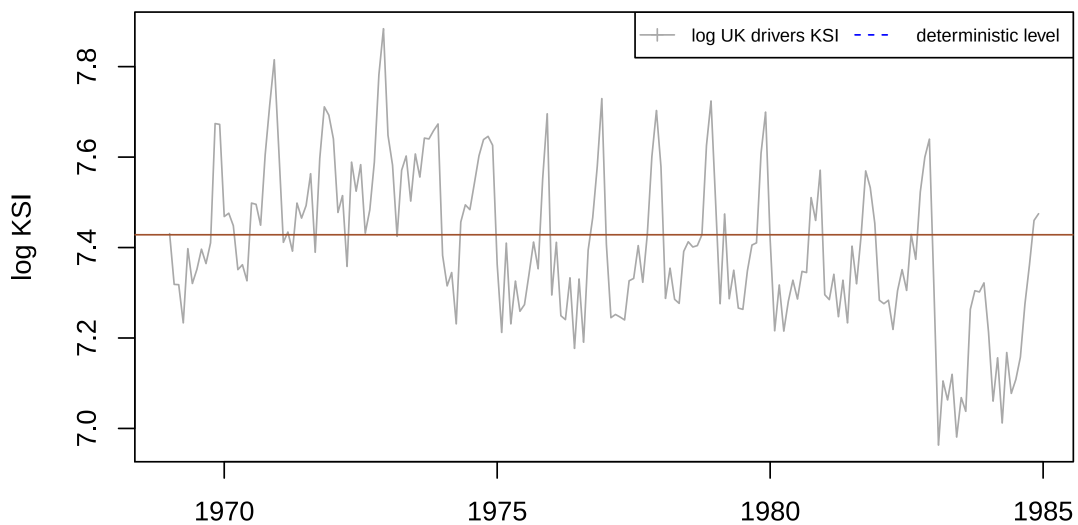


Figure 4.3: Deterministic level.

```
> par(mfrow=c(1,1))
> df <- model.matrix(fit)
> seas <- ts(df[,2:12]%*%coefs[2:12],start = 1960, frequency = 12)
> plot(seas, xlab="",ylab = "",col = "darkgrey")
> abline(h=0, col = "sienna", lty = 3)
```

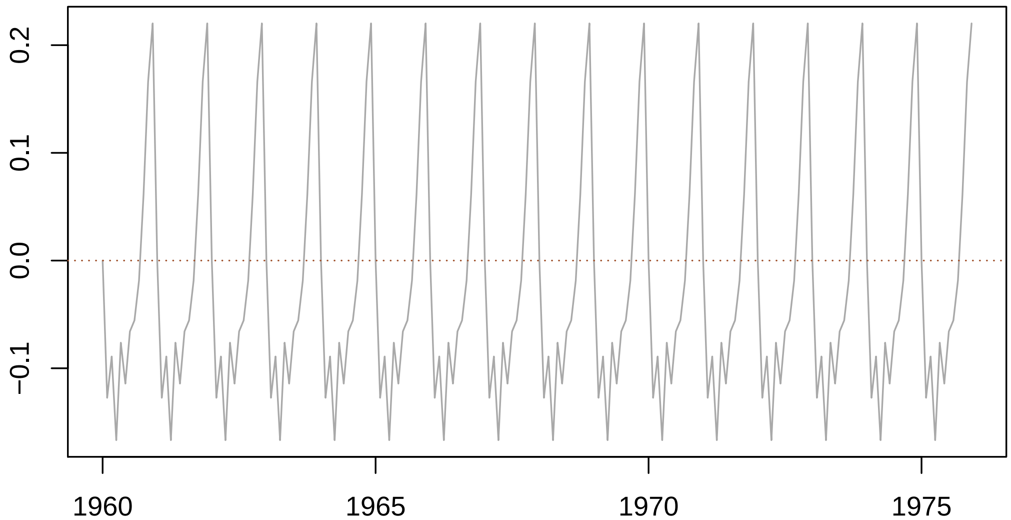


Figure 4.4: Deterministic seasonal.

```
> par(mfrow=c(1,1))
> res        <- ts(resid(fit),start = 1960, frequency = 12)
> plot(res, xlab="",ylab = "",col = "darkgrey")
> abline(h=0, col = "sienna", lty = 3)
```

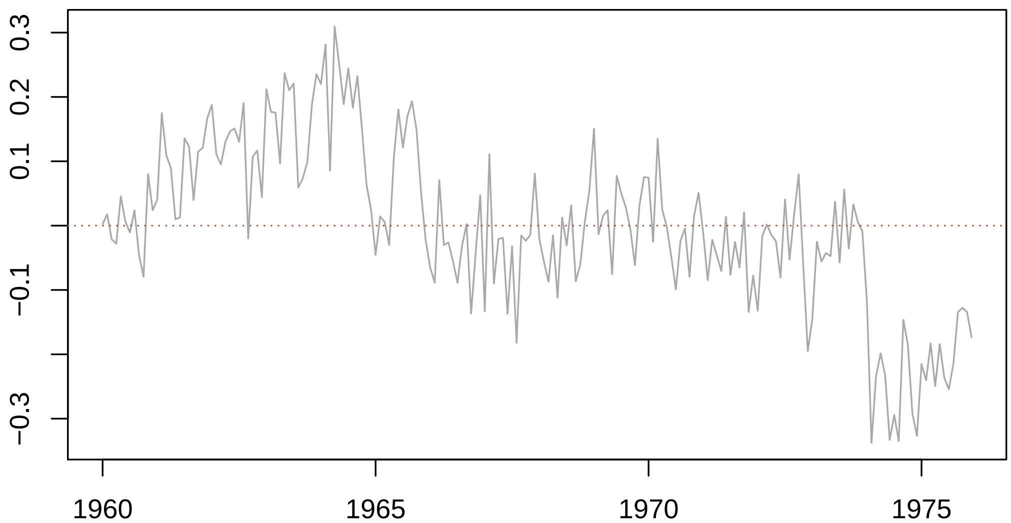


Figure 4.5: Irregular component for deterministic level and seasonal model.

Diagnostics

- Normality test - FAIL

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9849, p-value = 0.03719
```

- Independence - FAIL

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 1045.514, df = 15, p-value < 2.2e-16
> sapply(1:15,function(l){Box.test(res, lag=l, type = "Ljung-Box")$p.value})
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Thus clearly the residuals are autocorrelated and model is a suspect.

4.2 Stochastic level and seasonal

```
> fn                  <- function(params){
   mod               <- dlmModPoly(order = 1 ) +
                      dlmModSeas(frequency =12)
    V(mod)           <- exp(params[1])
   diag(W(mod))[1:2] <- exp(params[2:3])
   return(mod)
 }
> fit                 <- dlmMLE(data.1, rep(0,3),fn)
> mod                 <- fn(fit$par)
> (obs.error.var      <- V(mod))
            [,1]
[1,] 0.003513922
> (seas.var           <- (diag(W(mod))[2]))
[1] 1.891993e-10
> (level.var          <- (diag(W(mod))[1]))
[1] 0.0009456294
> filtered           <- dlmFilter(data.1,mod)
> smoothed           <- dlmSmooth(filtered)
> sm                 <- dropFirst(smoothed$s)
> mu                 <- c(sm[,1])
> nu                 <- c(sm[,2])
> res                <- c(residuals(filtered,sd=F))
```

Estimates

- Observation eq error variance is 0.00351392
- Level eq error variance is 0
- Seasonal eq error variance is 0.00094563

```
> par(mfrow=c(1,1))
> temp            <- ts(cbind(data.1,mu)[-c(1:12),],start = 1970,
                       frequency = 12)
> plot.ts(temp , plot.type="single" , col =c("darkgrey","blue"),
        lty=c(1,2), xlab="",ylab = "log KSI",ylim=c(7,8))
>         legend("topright",
        leg = c("log UK drivers KSI","  stochastic level"),
         cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
         pch=c(3,NA),bty = "y", horiz = T)
```

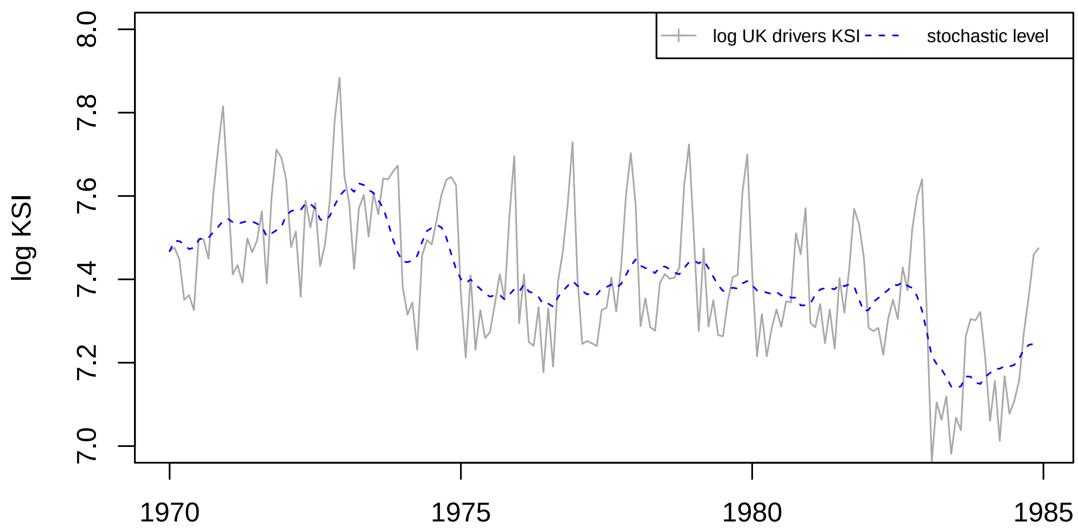


Figure 4.6: Stochastic level.

```
> par(mfrow=c(1,1))
> plot(ts(nu,start =1969,frequency =12),ylab="",xlab="",
     col = "darkgrey",lty=2)
> abline(h=0,col  ="sienna",lty = "dotted")
```

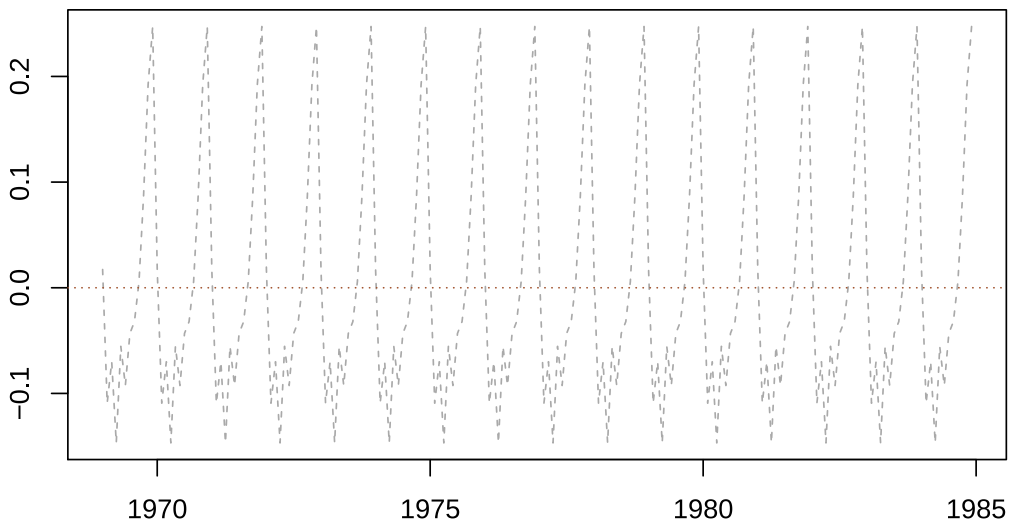


Figure 4.7: Stochastic seasonal.

```
> par(mfrow=c(1,1))
> plot(ts(nu[c(1:12)],start =1969,frequency =12),ylab="",
     xlab="", col = "darkgrey",lty=2,xlim= c(1969,1970))
> abline(h=0,col  ="sienna",lty = "dotted")
```

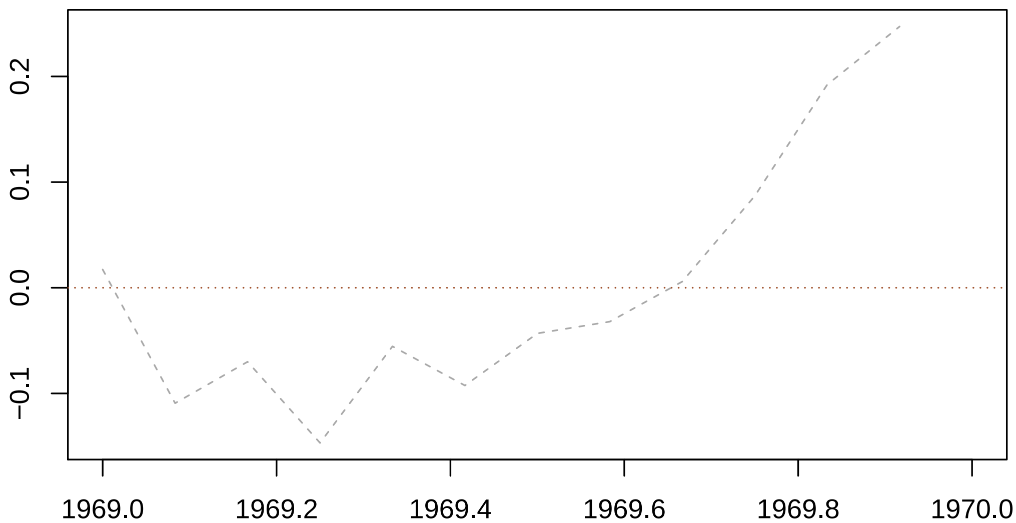


Figure 4.8: Stochastic seasonal for the year 1969.

```
> par(mfrow=c(1,1))
> plot(ts(res,start =1969,frequency =12),ylab="",xlab="",
      col = "darkgrey",lty=2)
> abline(h=0,col  ="sienna",lty = "dotted")
```

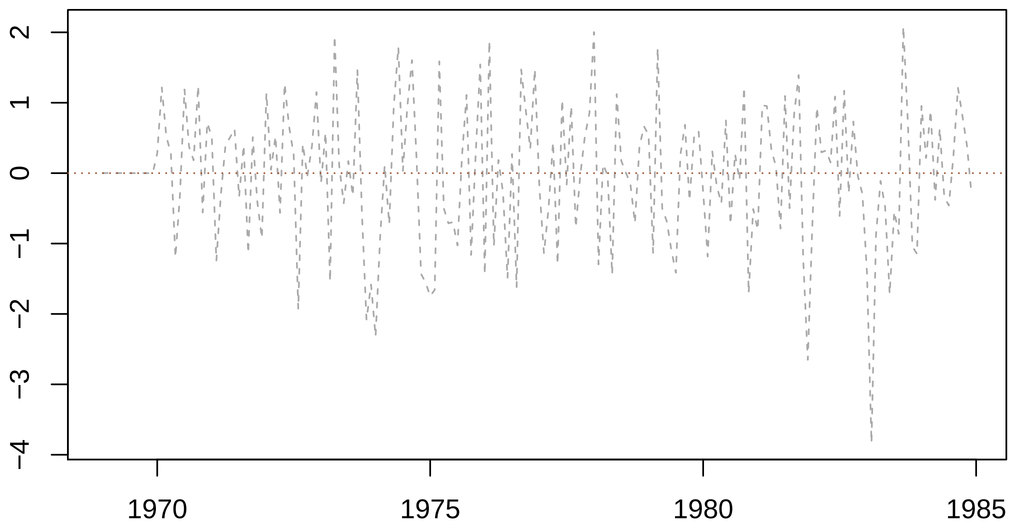


Figure 4.9: Irregular component for stochastic level and seasonal model.

Diagnostics

- Normality test - PASS

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9868, p-value = 0.06919
```

- Independence - PASS

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 15.2477, df = 15, p-value = 0.4337
> sapply(1:20,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.5716 0.7699 0.7020 0.3464 0.4371 0.4275 0.4476 0.1850 0.2523 0.2745
[11] 0.3344 0.3960 0.3119 0.3782 0.4337 0.5000 0.5305 0.2660 0.2830 0.3023
```

4.3 Stochastic level and deterministic seasonal

```
> fn          <- function(params){
    mod  <- dlmModPoly(order = 1, dV =exp(params[1]), dW = exp(params[2])) +
            dlmModSeas(frequency = 12,dV =exp(params[1]) , dW= rep(0,11))
      return(mod)
 }
> fit                <- dlmMLE(data.1, rep(0,2),fn)
> mod                <- fn(fit$par)
> (obs.error.var     <- V(mod))
            [,1]
[1,] 0.003513989
> (level.var         <- (diag(W(mod))[1]))
[1] 0.0009456422
> filtered          <- dlmFilter(data.1,mod)
> smoothed          <- dlmSmooth(filtered)
> sm                <- dropFirst(smoothed$s)
> mu                <- c(sm[,1])
> nu                <- c(sm[,2])
> res               <- c(residuals(filtered,sd=F))
```

This model is slightly better than stochastic level and stochastic seasonal.
Estimates

- Observation eq error variance is 0.00351399
- Level eq error variance is 0.00094564

4.4 The local level and seasonal model and UK inflation

```
> fn                    <- function(params){
   mod                 <- dlmModPoly(order = 1 ) +
                        dlmModSeas(frequency =4)
    V(mod)             <- exp(params[1])
   diag(W(mod))[1:2] <- exp(params[2:3])
   return(mod)
 }
> fit                   <- dlmMLE(data.2, rep(0,3),fn)
> mod                   <- fn(fit$par)
> (obs.error.var        <- V(mod))
             [,1]
[1,] 3.371299e-05
> (seas.var             <- (diag(W(mod))[2]))
[1] 4.345416e-07
> (level.var            <- (diag(W(mod))[1]))
[1] 2.124086e-05
> filtered             <- dlmFilter(data.2,mod)
> smoothed             <- dlmSmooth(filtered)
> sm                   <- dropFirst(smoothed$s)
> mu                   <- c(sm[,1])
> nu                   <- c(sm[,2])
> res                  <- c(residuals(filtered,sd=F))
```

ESTIMATES

- Observation eq error variance is 3.371e-05
- Level eq error variance is 2.124e-05
- Seasonal eq error variance is 4.3e-07

```
> par(mfrow=c(1,1))
> temp       <- window(cbind(data.2,mu),start = 1951, end = 2001)
> plot(temp , plot.type="single",
     col =c("darkgrey","blue"),lty=c(1,2), xlab="",ylab = "", )
> legend("topright",
       leg = c("quarterly price changes in UK","  stochastic level"),
       cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
       pch=c(3,NA),bty = "y", horiz = F)

> par(mfrow=c(1,1))
> temp      <- ts(nu,start = 1950,frequency  =4)
> plot(temp , col =c("darkgrey"),lty=1,   xlab="",
     ylab = "stochastic seasonaol",main="" )

> par(mfrow=c(1,1))
> temp      <- ts(res,start = 1950,frequency  =4)
> plot(temp , col =c("darkgrey"),lty=1,   xlab="",ylab = "irregular",main="" )
```

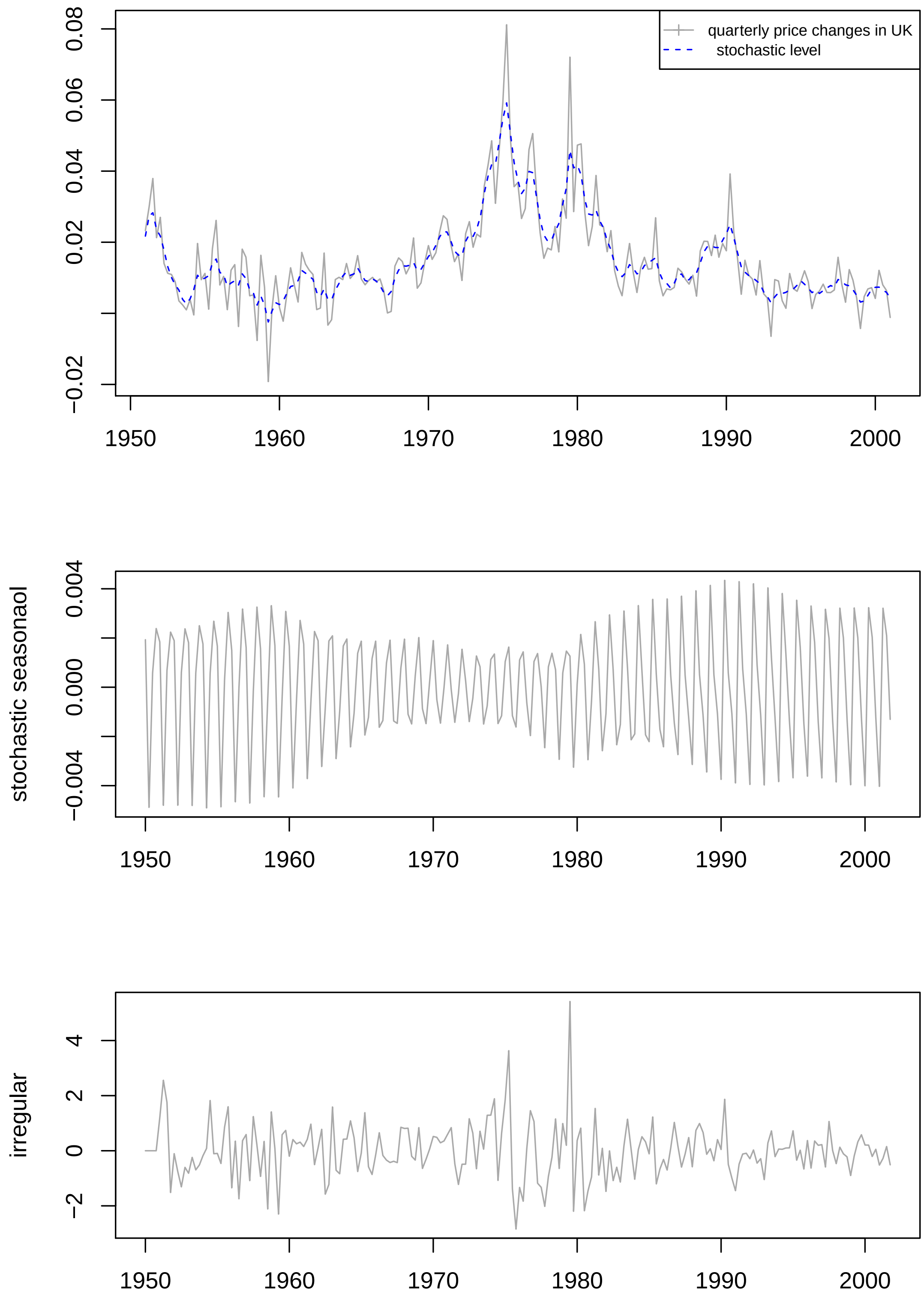


Figure 4.10: Stochastic level, seasonal and irregular in UK inflation series.

Diagnostics

- Normality test - FAIL

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.951, p-value = 1.533e-06
```

- Independence - PASS

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 16.2819, df = 15, p-value = 0.3636
> sapply(1:15,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.4728 0.5820 0.4941 0.5250 0.5794 0.6723 0.7602 0.8129 0.6194 0.6583
[11] 0.7231 0.7400 0.7359 0.6744 0.3636
```

5 The local level model with explanatory variable

$$
\begin{aligned}
y_t &= \mu_t + \beta_t x_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_\epsilon^2) \\
\mu_{t+1} &= \mu_t + \nu_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2) \\
\beta_{t+1} &= \beta_t + \tau_t, \quad \tau_t \sim N(0, \sigma_\tau^2)
\end{aligned}
$$

Dataset used in the chapter

```
> data.1            <- log(read.table("data/C5/UKdriversKSI.txt",skip=1))
> colnames(data.1)  <- "logKSI"
> data.1            <- ts(data.1, start = c(1969),frequency=12)
> x                 <- (read.table("data/C5/logUKpetrolprice.txt",skip=1))
> x                 <- ts(x, start = c(1969),frequency=12)
> t                 <- 1:dim(data.1)[1]
> n                 <- dim(data.1)[1]
```

5.1 Deterministic level and explanatory variable

TIME AS EXPLANATORY VARIABLE

```
> fit           <- lm(data.1~t)
> (coefs        <- round(as.numeric(coef(fit)),8))
[1]  7.54584273 -0.00144803
> (error.var    <- round(summary(fit)$sigma^2,8))
[1] 0.02299806
```

PETROL PRICES AS EXPLANATORY VARIABLE

```
> fit           <- lm(data.1~x)
> (coefs        <- round(as.numeric(coef(fit)),8))
[1]  5.8787308 -0.6716644
> (error.var    <- round(summary(fit)$sigma^2,8))
[1] 0.02301367
> fit.val       <- ts(fitted(fit),start = 1969, frequency = 12)
```

```
> par(mfrow=c(1,1))
> temp <- cbind(data.1, fit.val)
> plot(temp,  plot.type="single",col = c("darkgrey","blue"),
      lty=c(1,2), xlab="",ylab = "",pch=3,cex=0.5,
      cex.lab=0.8,cex.axis=0.7)
> legend("topright",leg = c("log UK drivers KSI",
                     "  deterministic level + beta*log(PETROL PRICE"),
                     cex = 0.5, lty = c(0, 2), col = c("darkgrey","blue"),
                     pch=c(3,NA), bty = "y",horiz = F)
```

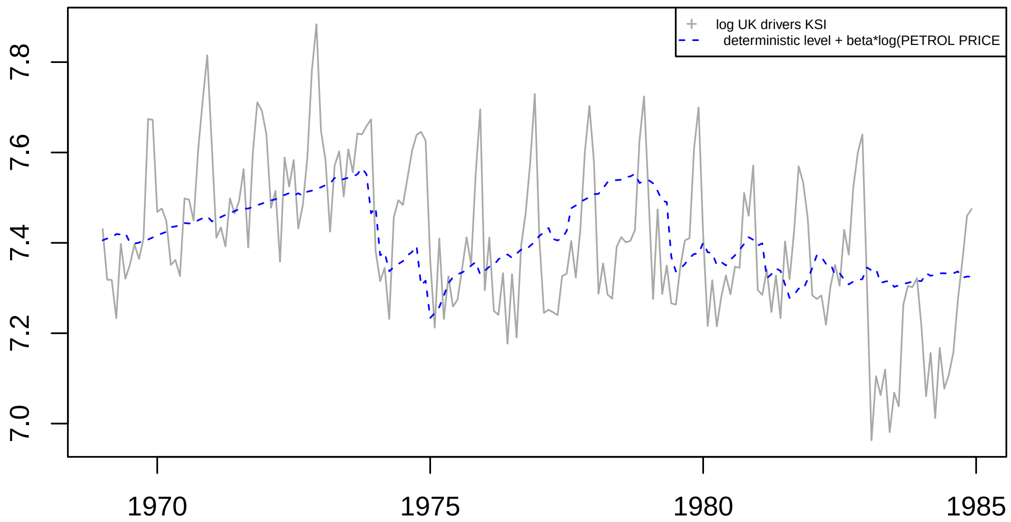


Figure 5.1: Deterministic level and explanatory variable log petrol price.

```
> par(mfrow=c(1,1))
> plot(c(x),c(data.1),  col = "darkgrey", xlab="",ylab = "",pch=3,cex=0.5,
      cex.lab=0.8,cex.axis=0.7)
> abline(coef(fit) , col = "blue", lwd  = 1, lty=1)
> legend("topright",leg = c("log UK drivers KSI against log PETROL PRICE",
                          "  deterministic level + beta*log(PETROL PRICE)"), cex = 0.6,
                             lty = c(0, 1), col = c("darkgrey","blue"),
                             pch=c(3,NA), bty = "y",horiz = F)
```

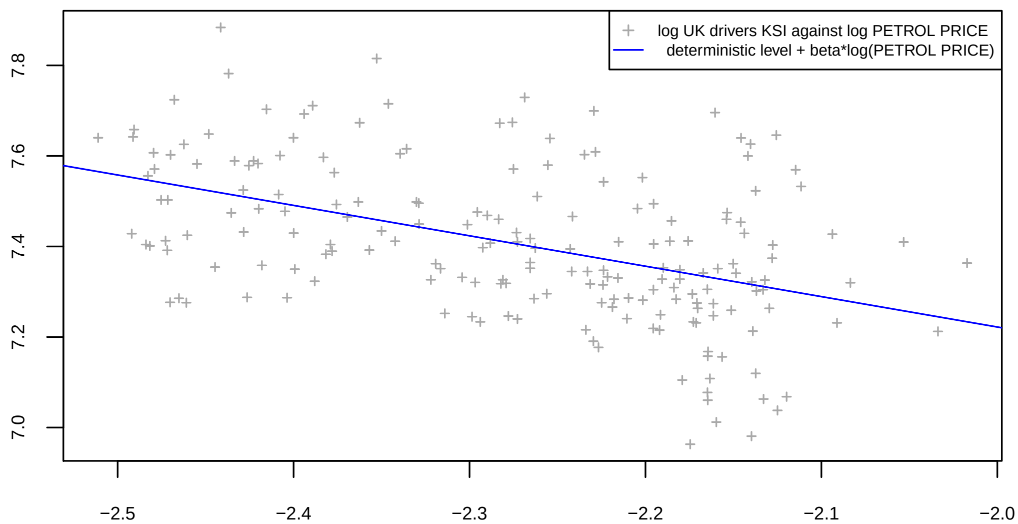


Figure 5.2: Conventional classical regression representation of deterministic level and explanatory variable log petrol price.

```
> par(mfrow=c(1,1))
> plot(ts(residuals(fit)),ylab="",xlab="", col = "darkgrey",lty=2)
> abline(h=0)
```

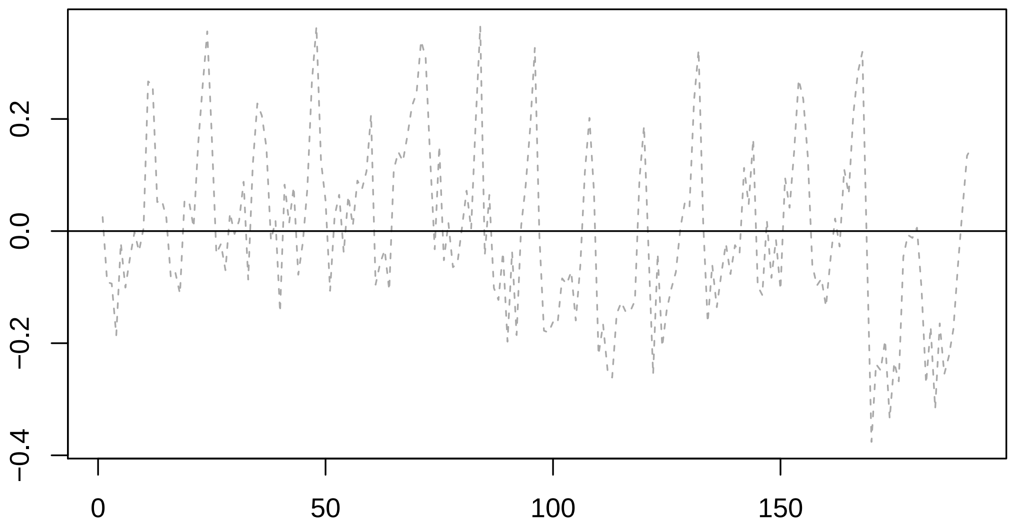


Figure 5.3: Irregular component for deterministic level model with explanatory variable log petrol price.

5.2 Stochastic level and explanatory variable

```
> fn                  <- function(params){
   mod           <- dlmModPoly(order = 1 ) +
                    dlmModReg(x, addInt=FALSE)
   V(mod)        <- exp(params[1])
   diag(W(mod))[1] <- exp(params[2])
   mod
 }
> fit                 <- dlmMLE(data.1, rep(0,2),fn)
> mod                 <- fn(fit$par)
> (obs.error.var      <- V(mod))
            [,1]
[1,] 0.002347963
> (level.var          <- (diag(W(mod))[1]))
[1] 0.01166743
> filtered           <- dlmFilter(data.1,mod)
> smoothed           <- dlmSmooth(filtered)
> sm                 <- dropFirst(smoothed$s)
> mu.1               <- sm[1,1]
> beta.1             <-sm[1,2]
> mu                 <- c(sm[,1])
> nu                 <- c(sm[,2])
> res                <- c(residuals(filtered,sd=F))
```

Estimates

- μ_1 = 6.82036082
- β_1 = -0.26104943
- Observation eq error variance is 0.00234796
- Level eq error variance is 0.01166743

```
> par(mfrow=c(1,1))
> temp            <- ts(cbind(data.1,mu),start = 1970, frequency = 12)
> plot.ts(temp , plot.type="single" , col =c("darkgrey","blue"),
        lty=c(1,2), xlab="",ylab = "log KSI",ylim=c(7,8))
>  legend("topright",
  leg = c("log UK drivers KSI","  stochastic level + beta*log(PETROL PRICE)"),
  cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
  pch=c(NA,NA),bty = "y", horiz = F)
```

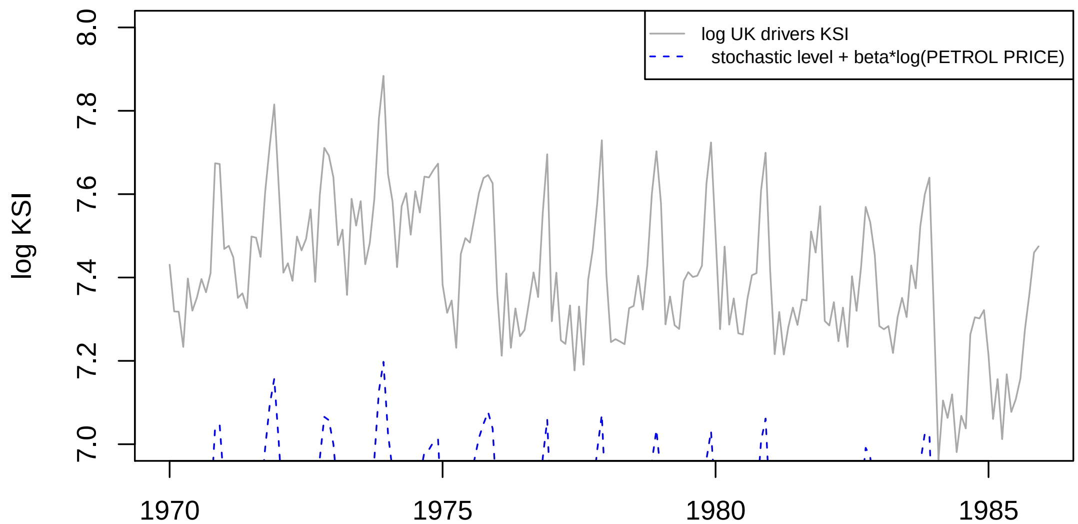


Figure 5.4: Stochastic level and deterministic explanatory variable log petrol price

```
> par(mfrow=c(1,1))
> plot.ts(res,ylab="",xlab="", col = "darkgrey",lty=2)
> abline(h=0,col  ="sienna")
```

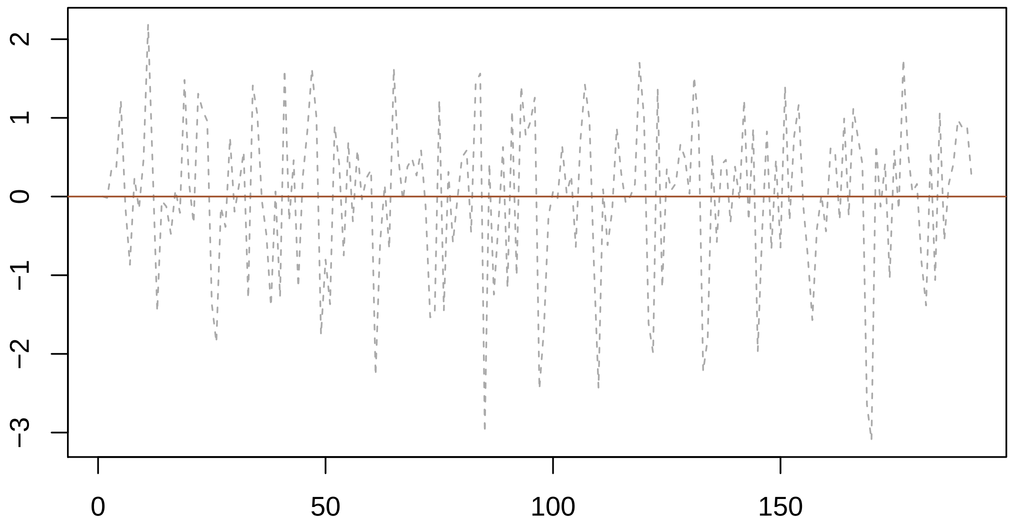


Figure 5.5: Irregular for stochastic level model with deterministic explanatory variable log petrol price.

Diagnostics

- Normality test - FAIL

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.968, p-value = 0.0002242
```

- Independence - FAIL

```
> Box.test(res, lag = 10, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 27.0775, df = 10, p-value = 0.002532
> sapply(1:10,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.8015 0.7141 0.1320 0.0118 0.0234 0.0188 0.0326 0.0028 0.0024 0.0025
```

First few lags are ok but the rest show dependence.

6 The local level model with intervention variable

This chapter models “level shift” as an intervention variable.

$$
\begin{aligned}
y_t &= \mu_t + \lambda_t w_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_\epsilon^2) \\
\mu_{t+1} &= \mu_t + \nu_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2) \\
\lambda_{t+1} &= \lambda_t + \rho_t, \quad \rho_t \sim N(0, \sigma_\rho^2)
\end{aligned}
$$

Dataset used in the chapter

```
> data.1            <- log(read.table("data/C6/UKdriversKSI.txt",skip=1))
> colnames(data.1)  <- "logKSI"
> data.1            <- ts(data.1, start = c(1969),frequency=12)
> x                 <- rep(1,dim(data.1)[1])
> x[1:169]          <- 0
> x                 <- ts(x, start = c(1969),frequency=12)
```

6.1 Deterministic level and intervention variable

```
> fit           <- lm(data.1~x)
> (coefs        <- round(as.numeric(coef(fit)),4))
[1]  7.4374 -0.2611
> (error.var    <- round(summary(fit)$sigma^2,6))
[1] 0.022243
> fit.val       <- ts(fitted(fit),start = 1969, frequency = 12)
```

Estimates

- coefs 7.4374 ,-0.2611
- Observation eq error variance is 0.022243

```
> par(mfrow=c(1,1))
> temp <- cbind(data.1, fit.val)
> plot(temp,  plot.type="single",col = c("darkgrey","blue"),
      lty=c(1,2), xlab="",ylab = "",pch=3,cex=0.5,
      cex.lab=0.8,cex.axis=0.7)
> legend("topright",leg = c("log UK drivers KSI",
                     "  deterministic level + lambda*(SEATBELT LAW"),
                     cex = 0.5, lty = c(0, 2), col = c("darkgrey","blue"),
                     pch=c(3,NA), bty = "y",horiz = F)
```

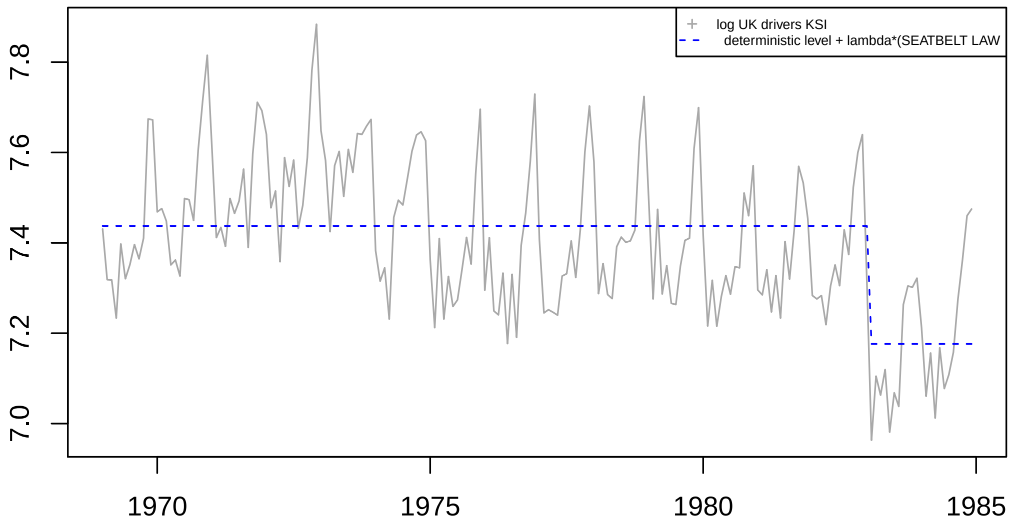


Figure 6.1: Deterministic level and intervention variable.

```
> par(mfrow=c(1,1))
> plot(c(x),c(data.1),  col = "darkgrey", xlab="",ylab = "",pch=3,cex=0.5,
     cex.lab=0.8,cex.axis=0.7)
> abline(coef(fit) , col = "blue", lwd  = 1, lty=1)
> legend("topright",leg = c("log UK drivers KSI against SEATBELT LAW",
                          "  deterministic level + lambda*(SEATBELT LAW)"), cex = 0.6,
                             lty = c(0, 1), col = c("darkgrey","blue"),
                             pch=c(3,NA), bty = "y",horiz = F)
```

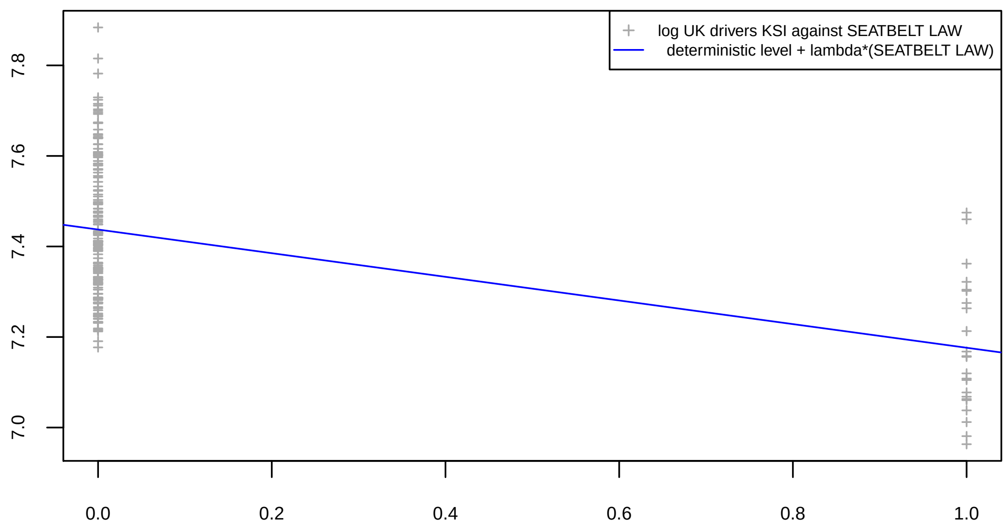


Figure 6.2: Conventional classical regression representation of deterministic level and intervention variable.

```
> par(mfrow=c(1,1))
> plot(ts(residuals(fit)),ylab="",xlab="", col = "darkgrey",lty=2)
> abline(h=0)
```

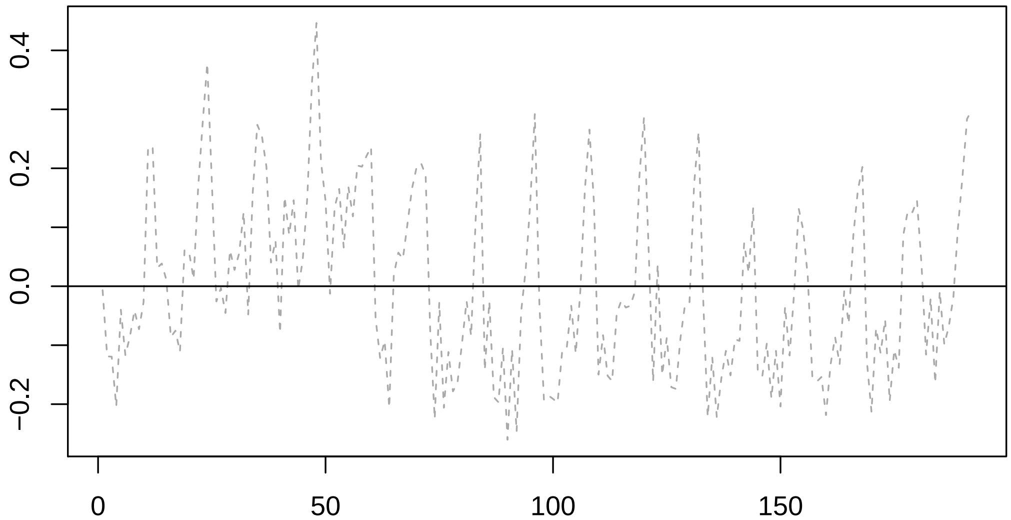


Figure 6.3: Irregular component for deterministic level model with intervention variable.

6.2 Stochastic level and intervention variable

```
> fn                    <- function(params){
       dlmModReg(x, dV = exp(params[1]), dW = exp(params[2 : 3]))
 }
> fit                   <- dlmMLE(data.1, rep(0,3),fn)
> mod                   <- fn(fit$par)
> (obs.error.var        <- V(mod))
           [,1]
[1,] 0.00269272
> (level.var            <- (diag(W(mod))[1]))
[1] 0.01041147
> filtered             <- dlmFilter(data.1,mod)
> smoothed             <- dlmSmooth(filtered)
> sm                   <- dropFirst(smoothed$s)
> mu.1                 <- sm[1,1]
> lam.1               <-sm[1,2]
> mu                   <- c(sm[,1])
> nu                   <- c(sm[,2])
> res                  <- c(residuals(filtered,sd=F))
```

ESTIMATES

- μ_1 = 7.41066292
- λ_1 = -0.37849681
- Observation eq error variance is 0.00269272
- Level eq error variance is 0.01041147

```
> par(mfrow=c(1,1))
> temp            <- ts(cbind(data.1,mu),start = 1970, frequency = 12)
> plot.ts(temp , plot.type="single" , col =c("darkgrey","blue"),lty=c(1,2),
      xlab="",ylab = "log KSI",ylim=c(7,8))
>  legend("topright",
  leg = c("log UK drivers KSI","  stochastic level + lambda*(SEATBELT LAW)"),
  cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
  pch=c(NA,NA),bty = "y", horiz = F)
```

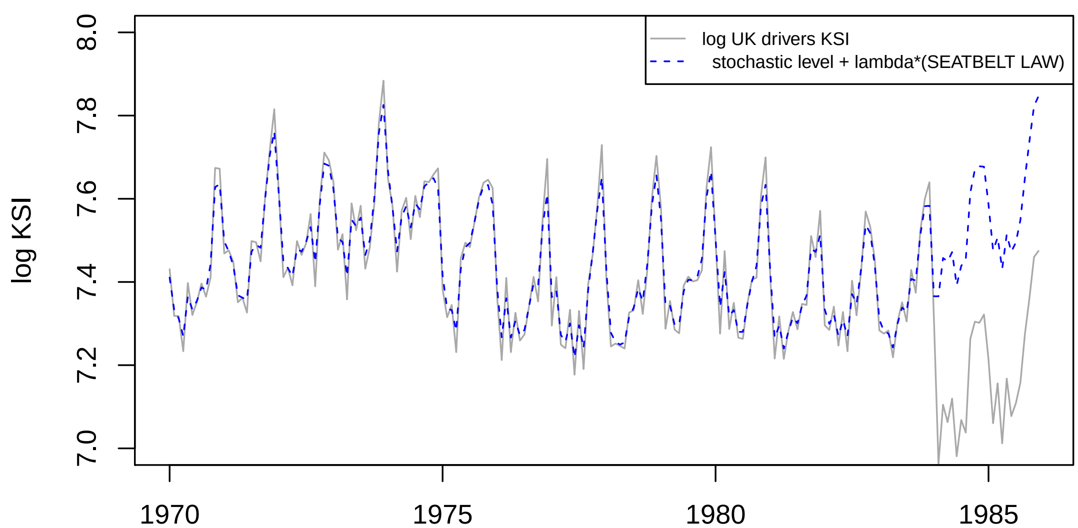


Figure 6.4: Stochastic level and intervention variable.

```
> par(mfrow=c(1,1))
> plot.ts(res,ylab="",xlab="", col = "darkgrey",lty=2)
> abline(h=0,col  ="sienna")
```

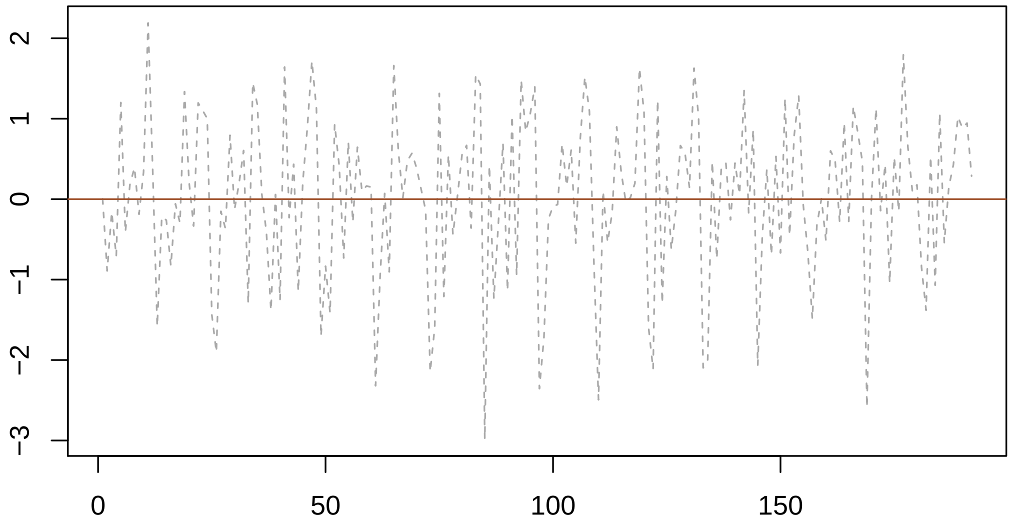


Figure 6.5: Irregular component for stochastic level model with intervention variable.

Diagnostics

- Normality test - FAIL

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.975, p-value = 0.00166
```

- Independence - FAIL

```
> Box.test(res, lag = 10, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 28.7907, df = 10, p-value = 0.001347
> sapply(1:15,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.8864 0.8216 0.1909 0.0122 0.0249 0.0132 0.0230 0.0013 0.0010 0.0013
[11] 0.0002 0.0000 0.0000 0.0000 0.0000
```

First three lags are ok but the rest show dependence.

7 The UK seat belt and inflation models

This chapter combines all the components described in the previous chapters and fits a model to the KSI dataset and UK inflation data.

$$
\begin{aligned}
y_t &= \mu_t + \gamma_{1,t} + \beta_t x_t + \lambda_t w_t + \epsilon_t, & \epsilon_t &\sim N(0, \sigma_\epsilon^2) \\
\mu_{t+1} &= \mu_t + \xi_t, & \xi_t &\sim N(0, \sigma_\xi^2) \\
\gamma_{1,t+1} &= -\gamma_{1,t} - \gamma_{2,t} - \gamma_{3,t} + \omega_t, & \omega_t &\sim N(0, \sigma_\omega^2) \\
\gamma_{2,t+1} &= \gamma_{1,t} \\
\gamma_{3,t+1} &= \gamma_{2,t} \\
\beta_{t+1} &= \beta_t + \tau_t, & \tau_t &\sim N(0, \sigma_\tau^2) \\
\lambda_{t+1} &= \lambda_t + \rho_t, & \rho_t &\sim N(0, \sigma_\rho^2)
\end{aligned}
$$

dataset used in the chapter

```
> data.1             <- log(read.table("data/C7/UKdriversKSI.txt",skip=1))
> colnames(data.1)   <- "logKSI"
> data.1             <- ts(data.1, start = c(1969),frequency=12)
> x                  <- rep(1,dim(data.1)[1])
> x[1:169]           <- 0
> x                  <- ts(x, start = c(1969),frequency=12)
> y                  <- (read.table("data/C7/logUKpetrolprice.txt",skip=1))
> y                  <- ts(y, start = c(1969),frequency=12)
> data.2             <- read.table("data/C7/UKinflation.txt",skip=1)
> colnames(data.2)   <- "IR"
> data.2             <- ts(data.2 , start = 1950,frequency  = 4)
>
```

7.1 Deterministic level and seasonal variable

```
> temp              <- cbind(data.1, rep(c(1:12),times= 16))
> fit               <- lm(temp[,1] ~  x + y+ as.factor(temp[,2] ))
> (coefs            <- round(as.numeric(coef(fit)),4))
 [1]  6.4083 -0.1971 -0.4521 -0.1133 -0.0733 -0.1486 -0.0610 -0.0943 -0.0437
[10] -0.0362 -0.0016  0.0739  0.1807  0.2369
> (error.var        <- round(summary(fit)$sigma^2,6))
[1] 0.007402
> fit.val           <- model.matrix(fit)[,c(1,2,3)] %*%coefs[1:3]
> res.1             <- residuals(fit)
```

Estimates

- μ_1 : 6.4083

- λ_1 : -0.1971
- β_1 : -0.4521
- Observation eq error variance is 0.007402

```
> par(mfrow=c(1,1))
> temp <- cbind(data.1, fit.val)
> plot(temp,  plot.type="single",col = c("darkgrey","blue"),
     lty=c(1,2), xlab="",ylab = "",pch=3,cex=0.5,
     cex.lab=0.8,cex.axis=0.7)
> legend("topright",leg = c("log UK drivers KSI",
      "deterministic level + beta*log(PETROL PRICE) + lambda*(SEATBELT LAW)"),
                  cex = 0.5, lty = c(0, 2), col = c("darkgrey","blue"),
                  pch=c(3,NA), bty = "y",horiz = F)
```

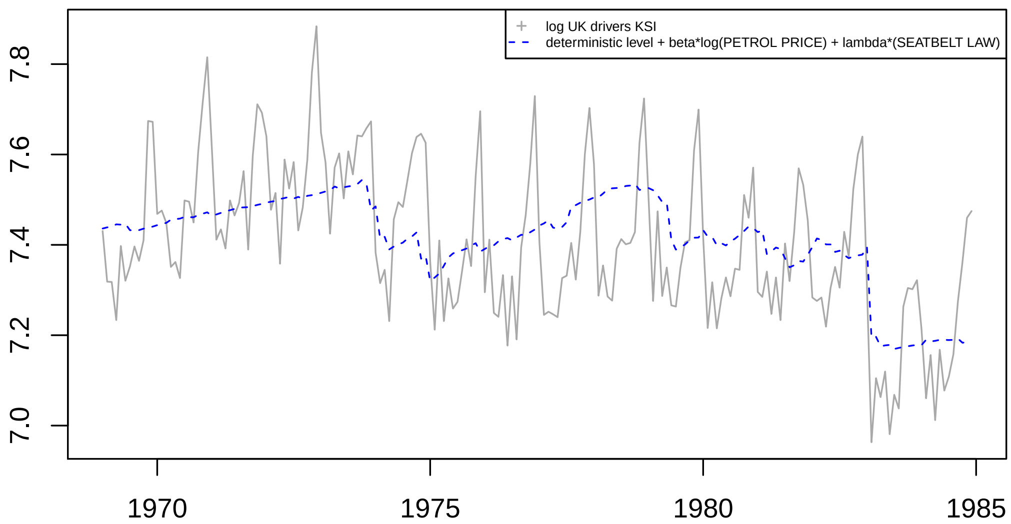


Figure 7.1: Deterministic level plus variables log petrol price and seat belt law.

7.2 Stochastic level and seasonal

```
> X        <- data.frame(intervention  = x, logpprice = y)
> fn       <- function(params){
   level <- dlmModPoly(order = 1,dV=exp(params[1]))
   seas  <- dlmModSeas(frequency = 12,dV=exp(params[1]),dW=rep(0,11))
   reg   <- dlmModReg(X,addInt=FALSE,dV=exp(params[1]),dW=c(0,0))
   mod   <- level+seas+reg
   diag(W(mod))[1:2] <- c(exp(params[2]),exp(params[3]))
   mod
 }
> fit                  <- dlmMLE(data.1, rep(0,3),fn)
> mod                  <- fn(fit$par)
> (obs.error.var       <- V(mod))
            [,1]
[1,] 0.004033224
> (level.var           <- (diag(W(mod))[1]))
[1] 0.0002682244
> (seas.var           <- (diag(W(mod))[2]))
[1] 1.196847e-10
> filtered            <- dlmFilter(data.1,mod)
> smoothed            <- dlmSmooth(filtered)
> sm                  <- dropFirst(smoothed$s)
> lam.1                <- sm[1,13]
> beta.1               <- sm[1,14]
> mu                  <- c(sm[,1])
> nu                  <- c(sm[,2])
> res                 <- c(residuals(filtered,sd=F))
> str(smoothed,1)
List of 3
 $ s  : mts [1:193, 1:14] 6.78 6.78 6.78 6.78 6.78 ...
  ..- attr(*, "tsp")= num [1:3] 1969 1985 12
  ..- attr(*, "class")= chr [1:2] "mts" "ts"
 $ U.S:List of 193
  .. [list output truncated]
 $ D.S: num [1:193, 1:14] 0.247 0.247 0.247 0.246 0.246 ...
```

Estimates

- β_1 = -0.27672329
- λ_1 = -0.23759134
- Observation eq error variance is 0.00403322
- Level eq error variance is 0.00026822
- Seasonal eq error variance is 0

```
> par(mfrow=c(1,1))
> temp <- sm[,1]+sm[,13]*x+sm[,14]*y
> temp<- ts(cbind(c(data.1),temp),start = 1969,
                       frequency = 12)
> plot.ts(temp , plot.type="single" , col =c("darkgrey","blue"),lty=c(1,2),
     xlab="",ylab = "log KSI",ylim=c(7,8))
> legend("topright",
 leg = c("log UK drivers KSI",
  "stochastic level + beta*log(PETROL PRICE) + lambda*(SEATBELT LAW)"),
  cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
  pch=c(3,NA),bty = "y", horiz = F)
```

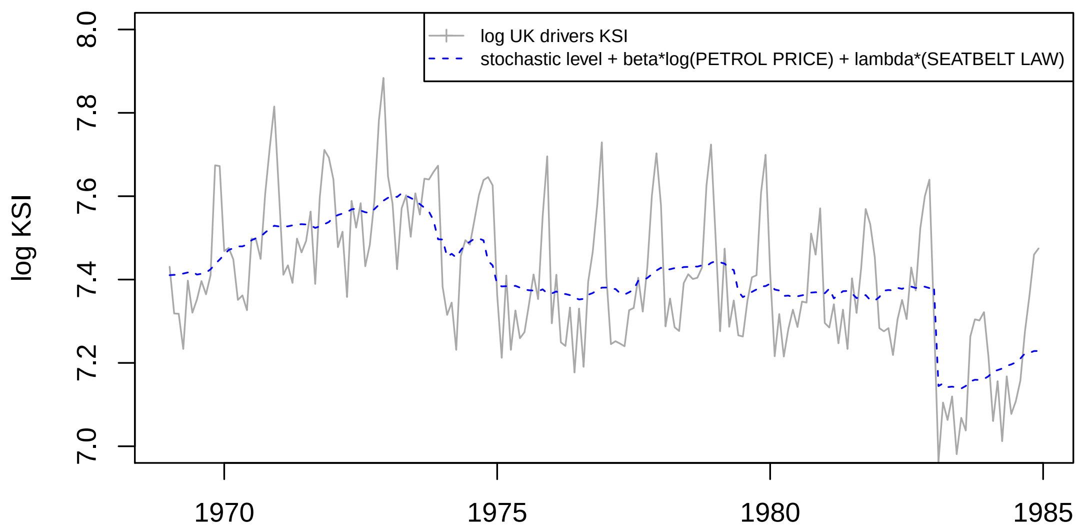


Figure 7.2: Stochastic level plus variables log petrol price and seat belt law.

```
> par(mfrow=c(1,1))
> plot(ts(nu[-c(1:12)],start =1970,frequency =12),ylab="",xlab="", col = "darkgrey",lty=2)
> abline(h=0,col  ="sienna",lty = "dotted")
```

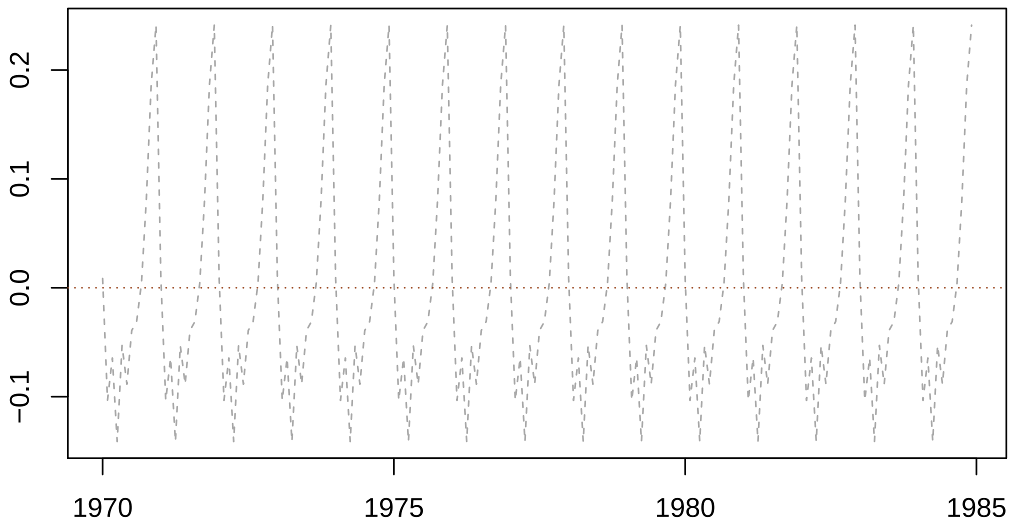


Figure 7.3: Stochastic seasonal.

```
> par(mfrow=c(1,1))
> plot(ts(res[-c(1:12)],start =1970,frequency =12),ylab="",xlab="", col = "darkgrey",lty=2)
> abline(h=0,col  ="sienna",lty = "dotted")
```

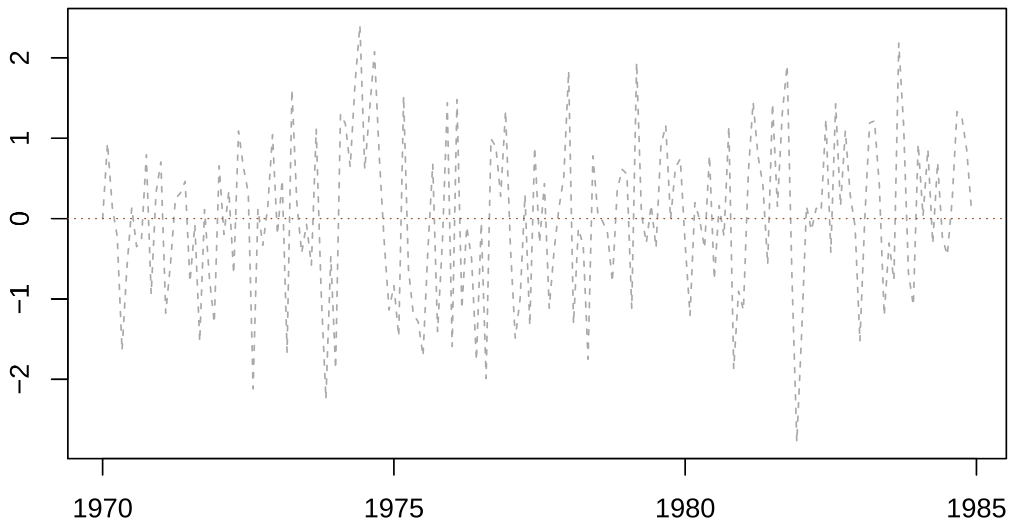


Figure 7.4: Irregular component for stochastic level and seasonal model.

Diagnostics

- Normality test - PASS

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9929, p-value = 0.4849
```

- Independence - PASS

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 19.6531, df = 15, p-value = 0.1856
> sapply(1:15,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.2724 0.3446 0.3774 0.2392 0.2797 0.3923 0.5069 0.1814 0.2026 0.1422
[11] 0.1933 0.2122 0.1181 0.1418 0.1856
```

7.3 Stochastic level and deterministic seasonal

```
> X       <- data.frame(intervention  = x, logpprice = y)
> fn      <- function(params){
   level <- dlmModPoly(order = 1,dV=exp(params[1]))
   seas  <- dlmModSeas(frequency = 12,dV=exp(params[1]),dW=rep(0,11))
   reg   <- dlmModReg(X,addInt=FALSE,dV=exp(params[1]),dW=c(0,0))
   mod   <- level+seas+reg
   diag(W(mod))[1:2] <- c(exp(params[2]),0)
   mod
 }
> fit                 <- dlmMLE(data.1, rep(0,2),fn)
> mod                 <- fn(fit$par)
> (obs.error.var      <- V(mod))
            [,1]
[1,] 0.004033987
> (level.var          <- (diag(W(mod))[1]))
[1] 0.0002680762
> (seas.var          <- (diag(W(mod))[2]))
[1] 0
> filtered           <- dlmFilter(data.1,mod)
> smoothed           <- dlmSmooth(filtered)
> sm                 <- dropFirst(smoothed$s)
> lam.1               <- sm[1,13]
> beta.1              <- sm[1,14]
> mu                 <- c(sm[,1])
> nu                 <- c(sm[,2])
> res                <- c(residuals(filtered,sd=F))
```

Estimates

- β_1 = -0.2767341
- λ_1 = -0.2375879
- Observation error variance is 0.004034
- Level error variance is 0.000268

```
> par(mfrow=c(1,1))
> acf(res.1,main="")
```

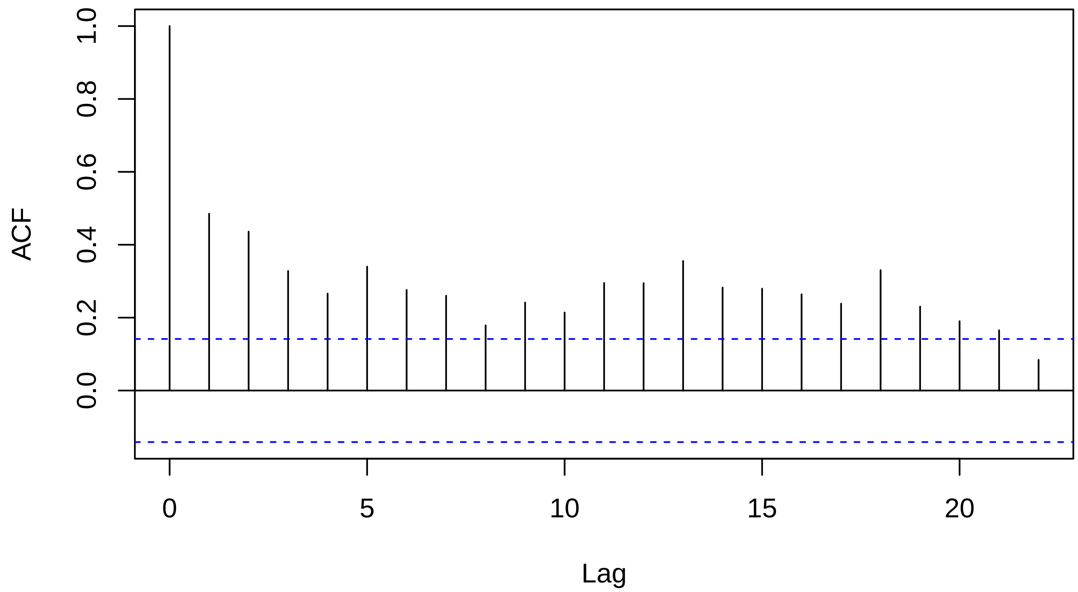


Figure 7.5: Correlogram of irregular component of completely deterministic level and seasonal model.

```
> par(mfrow=c(1,1))
> acf(res,main  ="")
```

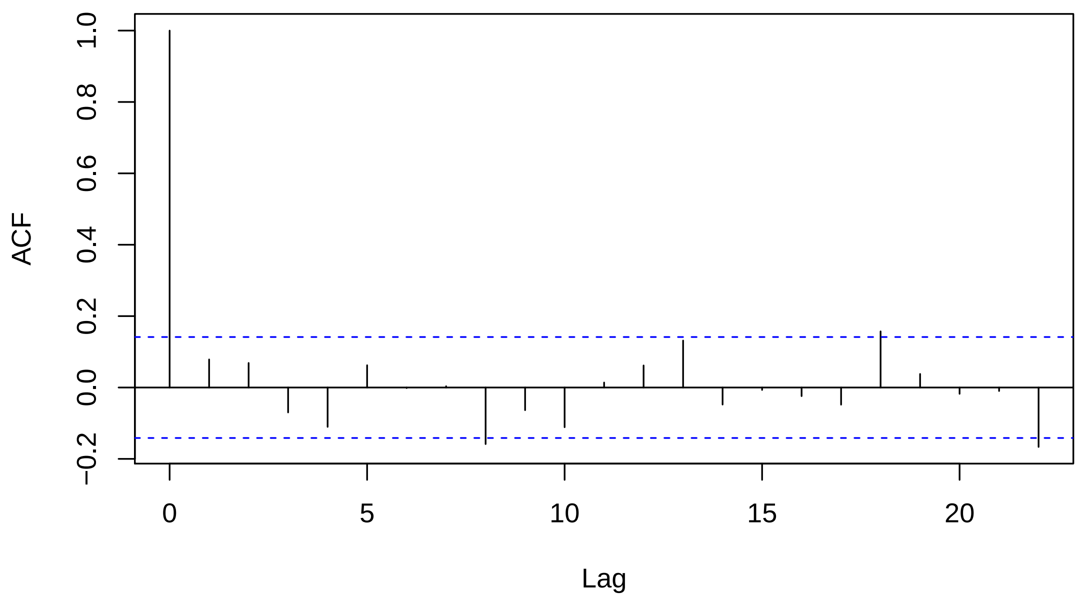


Figure 7.6: Correlogram of irregular component of stochastic level and deterministic seasonal model.

7.4 UK Inflation model

Analysis using pulse intervention variables

```
> n                    <- length(c(data.2))[1]
> x                    <- rep(0,n)
> x[which(time(data.2) == 1975.25)] = 1
> y                    <- rep(0,n)
> y[which(time(data.2) == 1979.50)] = 1
> X                    <- data.frame(intervention.1  = x, intervention.2 = y)
> fn                   <- function(params){
   mod                <- dlmModPoly(order = 1) +
                         dlmModSeas(frequency =4) +
                         dlmModReg(X,addInt=FALSE )
    V(mod)            <- exp(params[1])
   diag(W(mod))[1:2]  <- exp(params[2:3])
   return(mod)
 }
> fit                  <- dlmMLE(data.2, rep(0,3),fn)
> fit$convergence
[1] 0
> mod                  <- fn(fit$par)
> (obs.error.var       <- V(mod))
             [,1]
[1,] 2.202695e-05
> (level.var           <- (diag(W(mod))[1]))
[1] 1.863965e-05
> (seas.var           <- (diag(W(mod))[4]))
[1] 0
> filtered             <- dlmFilter(data.2,mod)
> smoothed            <- dlmSmooth(filtered)
> sm                  <- dropFirst(smoothed$s)
> mu                  <- sm[,1] + sm[,5] *X[,1] + sm[,6] *X[,2]
> nu                   <- sm[,2]
> res                  <- residuals(filtered,sd=F)
```

Estimates

- Observation eq error variance is 2.2026949e-05
- Level eq error variance is 1.8639652e-05
- Seasonal eq error variance is 0

```
> par(mfrow=c(1,1))
> temp        <- window(cbind(data.2,mu),start = 1950, frequency = 4)
> plot(temp , plot.type="single",
      col =c("darkgrey","blue"),lty=c(1,2), xlab="",ylab = "", )
> legend("topright",
        leg = c("quarterly price changes in UK",
        "  stochastic level + pulse intervention variables "),
        cex = 0.7, lty = c(1, 2),col = c("darkgrey","blue"),
        pch=c(3,NA),bty = "y", horiz = F)


> par(mfrow=c(1,1))
> temp      <- ts(nu,start = 1950,frequency  =4)
> plot(temp , col =c("darkgrey"),lty=1,   xlab="",
      ylab = "stochastic seasonaol",main="" )


> par(mfrow=c(1,1))
> temp      <- ts(res,start = 1950,frequency  =4)
> plot(temp , col =c("darkgrey"),lty=1,   xlab="",ylab = "irregular",main="" )
> abline(h=0)
```

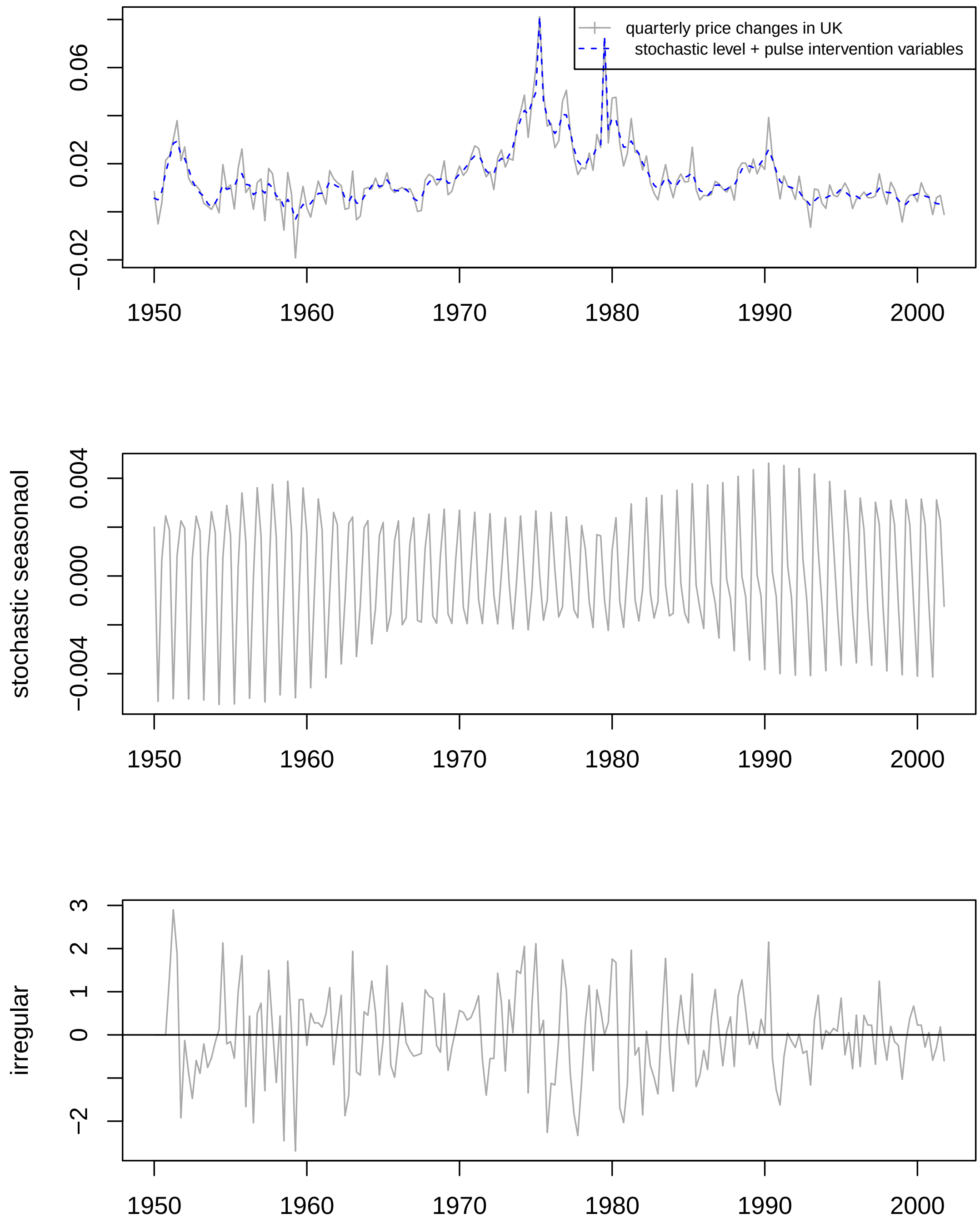


Figure 7.7: Local level (including pulse interventions), local seasonal and irregular for UK inflation time series data.

Diagnostics

- Normality test - PASS

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9947, p-value = 0.6706
```

- Independence - PASS

```
> Box.test(res, lag = 10, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 14.364, df = 10, p-value = 0.157
> sapply(1:10,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.2382 0.0548 0.0825 0.0928 0.1325 0.1836 0.2626 0.3537 0.2183 0.1570
```

8 General treatment of univariate state space models

There are many notation for state space framework and I am comfortable with the following notation, that is different from the one mentioned in the book.

$$Y_t = F_t\theta_t + v_t, \quad v_t \sim N(0, V_t)$$
$$\theta_t = G_t\theta_{t-1} + w_t, \quad w_t \sim N(0, W_t)$$

datasets used in the chapter

```
> data.1            <- log(read.table("data/C8/UKdriversKSI.txt",skip=1))
> colnames(data.1)  <- "logKSI"
> data.1            <- ts(data.1, start = c(1969),frequency=12)
> intervention      <- rep(1,dim(data.1)[1])
> intervention[1:169]<- 0
> intervention     <- ts(intervention, start = c(1969),frequency=12)
> prices           <- (read.table("data/C8/logUKpetrolprice.txt",skip=1))
> prices           <- ts(prices, start = c(1969),frequency=12)
> data.2            <- log(read.table("data/C8/NorwayFinland.txt",skip=1))
> data.2            <- data.2[,2,drop=F]
> colnames(data.2)  <- "logNorFatalities"
> data.2            <- ts(data.2 , start = c(1970,1))
> data.3            <- log(read.table("data/C8/NorwayFinland.txt",skip=1))
> data.3            <- data.3[,3,drop=F]
> colnames(data.3)  <- "logFinFatalities"
> data.3            <- ts(data.3 , start = c(1970,1))
>
```

8.1 Confidence intervals

```
> fn                  <- function(params){
 mod <- dlmModPoly(order = 1, dV =exp(params[1]), dW = exp(params[2])) +
        dlmModSeas(frequency = 12,dV =exp(params[1]) , dW= rep(0,11))
       return(mod)
 }
> fit            <- dlmMLE(data.1, rep(0,2),fn)
> mod            <- fn(fit$par)
> filtered       <- dlmFilter(data.1,mod)
> smoothed       <- dlmSmooth(filtered)
> cov            <- dlmSvd2var(smoothed$U.S, smoothed$D.S)
> lev.var        <- sapply(cov, function(x){x[1,1]})
> mu             <- ((smoothed$s)[,1])[-1]
> nu             <- ((smoothed$s)[,2])[-1]
```

```
> res           <- residuals(smoothed,sd=F)
> lev.ts        <- ts(lev.var[-1],start = 1969, frequency  =12)
> wid           <- qnorm(0.05, lower = FALSE) *sqrt(lev.ts)
> temp          <- cbind(mu, mu + wid %o% c(-1, 1))
> temp          <- ts(temp,start = 1969,frequency =12)

> par(mfrow=c(1,1))
> plot(lev.ts,xlab="",ylab = "level estimation variance")
```

Figure 8.1: Level estimation error variance for stochastic level and deterministic seasonal model applied to the log of UK drivers KSI.

```
> par(mfrow=c(1,1))
> plot(temp, plot.type = "s", type = "l",lty = c(1, 5, 5),
     ylab = "Level", xlab = "", ylim = range(data.1),col=c("blue","red","red"),lwd=2)
> lines(data.1, type = "o", col = "darkgrey")
> legend("topright",
       leg = c("log UK drivers KSI","  stochastic level +/- 1.64SE"),
       cex = 0.7, lty = c(1, 5),col = c("darkgrey","red"),
       ,bty = "y", horiz = F)
```

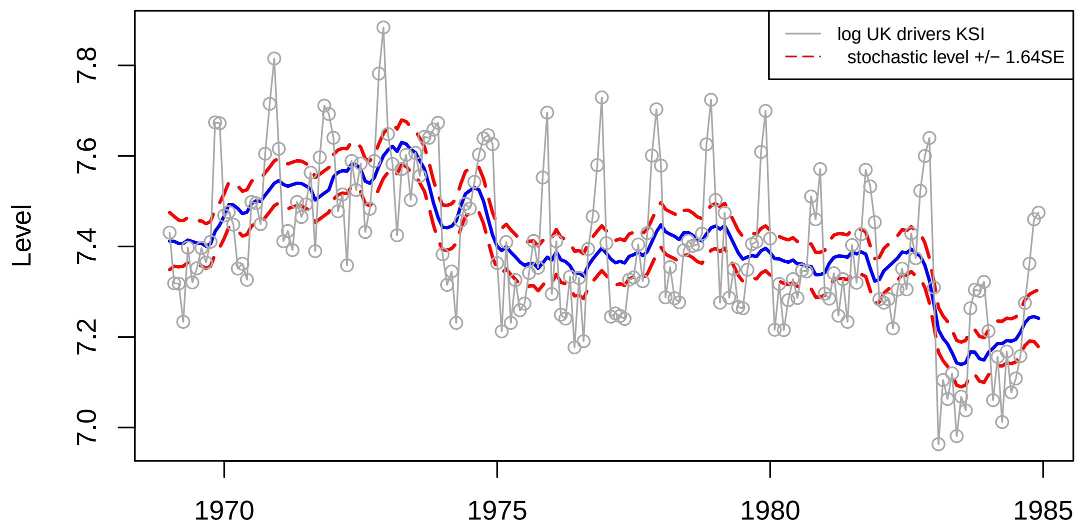


Figure 8.2: Stochastic level and its 90% confidence interval for stochastic level and deterministic seasonal model applied to the log of UK drivers KSI.

```
> nu.var       <- sapply(cov, function(x){x[2,2]})
> nu.var.ts   <- ts(nu.var[-1],start = 1969, frequency  =12)
> wid           <- qnorm(0.05, lower = FALSE) *sqrt(nu.var.ts)
> temp          <- cbind(nu, nu + wid %o% c(-1, 1))
> temp          <- ts(temp,start = 1969,frequency =12)
> par(mfrow=c(1,1))
> plot(temp, plot.type = "s", type = "l",lty = c(1, 5, 5),
      ylab = "Level", xlab = "",col=c("blue","red","red"),lwd=1)
> legend("topright",
        leg =   "deterministic level +/- 1.64SE",
        cex = 0.7, lty = c(5),col = c("red"),
        bty = "y", horiz = F)
```

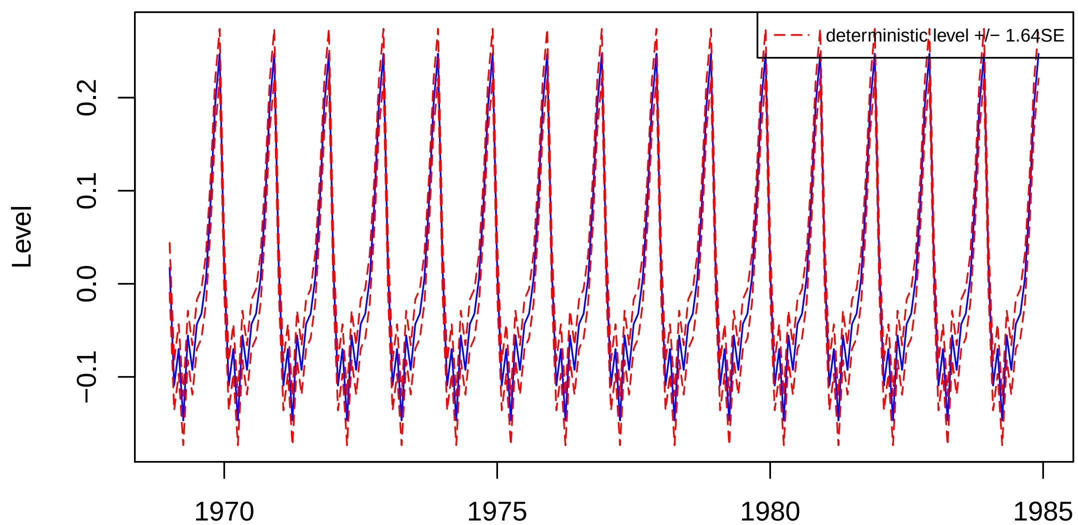


Figure 8.3: Deterministic seasonal and its 90% confidence interval for stochastic level and deterministic seasonal model applied to the log of UK drivers KSI.

```
> wid1           <- qnorm(0.05, lower = FALSE) *sqrt(lev.ts)
> temp1         <- cbind(mu, mu + wid %o% c(-1, 1))
> temp1         <- ts(temp1,start = 1969,frequency =12)
> wid2           <- qnorm(0.05, lower = FALSE) *sqrt(nu.var.ts)
> temp2         <- cbind(nu, nu + wid %o% c(-1, 1))
> temp2         <- ts(temp2,start = 1969,frequency =12)
> temp3         <- temp1+temp2
> par(mfrow=c(1,1))
> plot(temp3, plot.type = "s", type = "l",lty = c(1, 5, 5),
      ylab = "Level", xlab = "",col=c("blue","red","red"),lwd=1)
> legend("topright",
        leg =   "signal +/- 1.64SE",
        cex = 0.7, lty = c(5),col = c("red"),
        bty = "y", horiz = F)
```

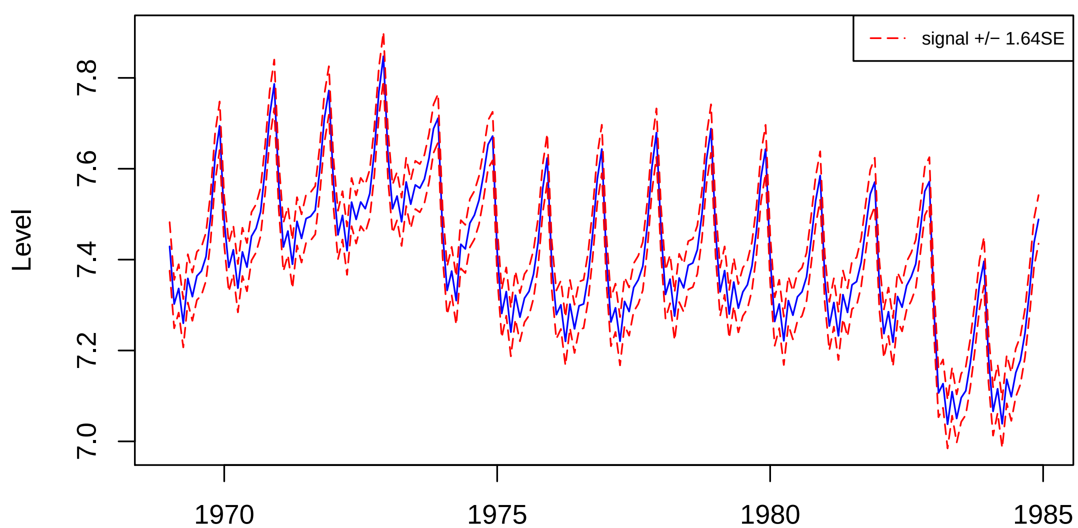


Figure 8.4: Stochastic level plus deterministic seasonal and its 90interval for stochastic level and deterministic seasonal model applied to the log of UK drivers KSI.

8.2 Filtering and prediction

```
> fn                    <- function(params){
 mod <- dlmModPoly(order = 1, dV =exp(params[1]), dW = exp(params[2]))
       return(mod)
 }
> fit               <- dlmMLE(data.2, rep(0,2),fn)
> mod               <- fn(fit$par)
> filtered          <- dlmFilter(data.2,mod)
> smoothed          <- dlmSmooth(filtered)
> mu.s              <- dropFirst(smoothed$s)
> mu.f              <- dropFirst(filtered$a)
> mu.m              <- dropFirst(filtered$m)

> temp          <- cbind( mu.s,mu.f)
> par(mfrow=c(1,1))
> plot(temp, plot.type = "s", type = "l",lty = c(1, 2),
      ylab = "Level", xlab = "",col=c("blue","sienna"),lwd=1)
> legend("topright",leg = c("smoothed level","filtered level"),
        cex = 0.7,lty = c(1, 2), col = c("blue","sienna"),
        bty = "y", horiz = T)
```

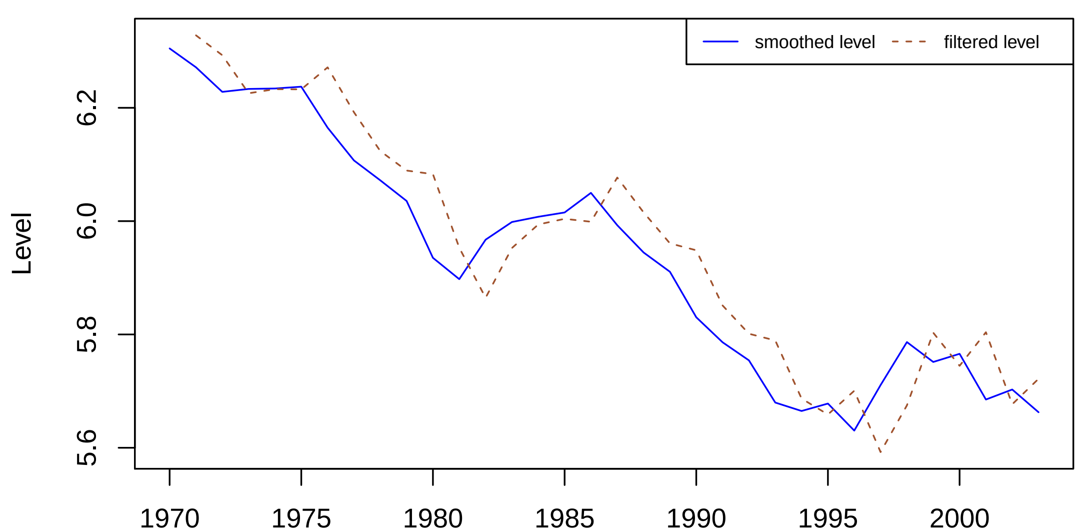


Figure 8.5: Smoothed and filtered state of the local level model applied to Norwegian road traffic fatalities.

```
> temp         <- window(cbind( data.2,mu.f),start = 1978,end=1983)
> plot(temp, plot.type = "s", type = "l",lty = c(1, 2),
      ylab = "Level", xlab = "",col=c("darkgrey","blue"),lwd=1)
> legend("topright",leg = c("log fatalities Norway","filtered level"),
        cex = 0.6,lty = c(1, 2), col = c("darkgrey","blue"),
        bty = "y", horiz = F)
>
```

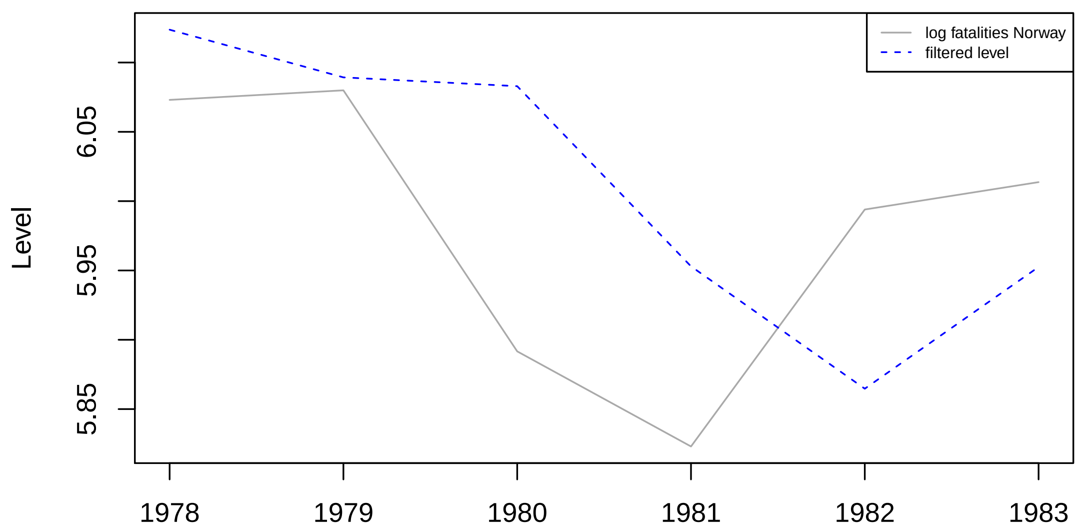


Figure 8.6: Illustration of computation of the filtered state for the local level model applied to Norwegian road traffic fatalities.

```
> temp         <- data.2 - mu.f
> par(mfrow=c(1,1))
> plot(temp, type = "l", ylab = "", xlab = "",lty=2, col = "darkgrey")
> abline(h=0, col = "sienna")

> cov        <- (dlmSvd2var(filtered$U.R, filtered$D.R))
> var        <- (sapply(cov,function(x) mod$FF%*%x%*%t(mod$FF)))+V(mod)
> lev.ts        <- ts(var[-1],start = 1969, frequency  =1)
> par(mfrow=c(1,1))
> plot(lev.ts,xlab="",ylab = "")
```

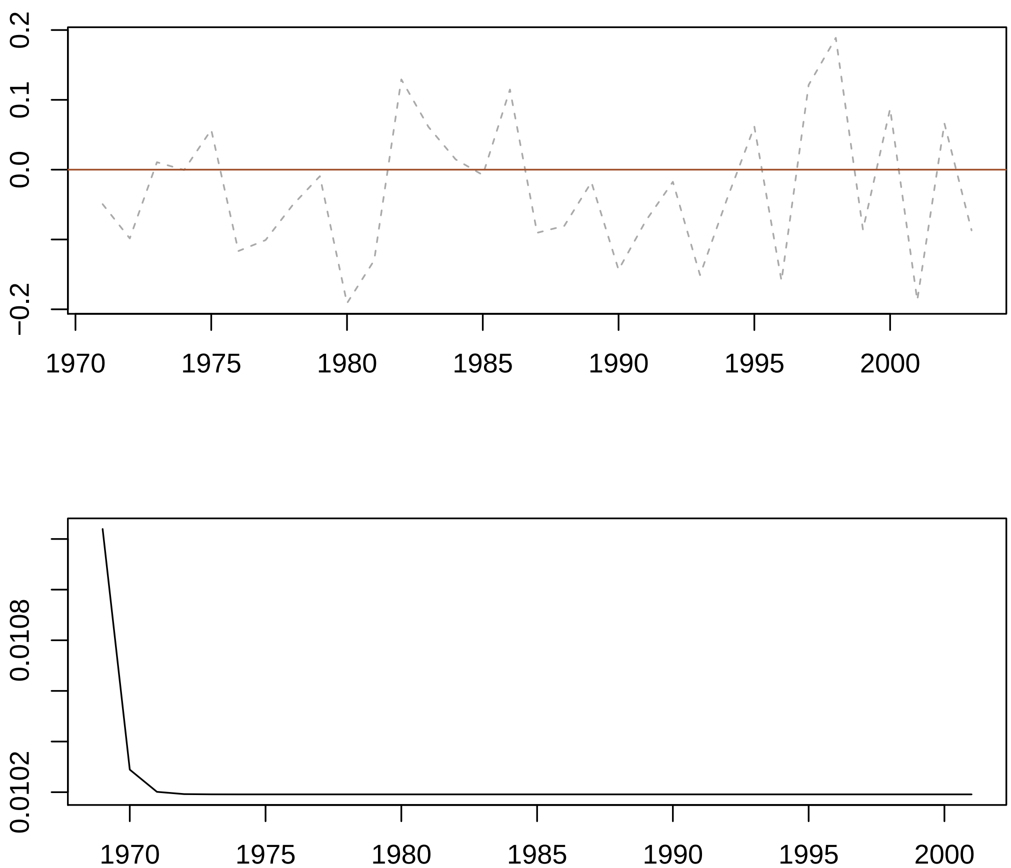


Figure 8.7: One-step ahead prediction errors (top) and their variances (bottom).

8.3 Diagnostic tests

```
> X       <- data.frame(intervention  = intervention, logpprice = prices)
> fn      <- function(params){
   level <- dlmModPoly(order = 1,dV=exp(params[1]))
   seas  <- dlmModSeas(frequency = 12,dV=exp(params[1]),dW=rep(0,11))
   reg   <- dlmModReg(X,addInt=FALSE,dV=exp(params[1]),dW=c(0,0))
   mod   <- level+seas+reg
   diag(W(mod))[1:2] <- c(exp(params[2]),0)
   mod
 }
> fit                 <- dlmMLE(data.1, rep(0,2),fn)
> mod                 <- fn(fit$par)
> filtered           <- dlmFilter(data.1,mod)
> smoothed           <- dlmSmooth(filtered)
> sm                 <- dropFirst(smoothed$s)
> lam.1               <- sm[1,13]
> beta.1              <- sm[1,14]
> mu                 <- c(sm[,1])
> nu                 <- c(sm[,2])
> res                <- c(residuals(filtered,sd=F))
> res                 <- ts(res[-c(1:14)], start = 1970.167)
>
```

```
> par(mfrow=c(1,1))
> plot(res,col = "darkgrey",ylab="",xlab="")
> abline(h=0,col="sienna",lty=2)
```

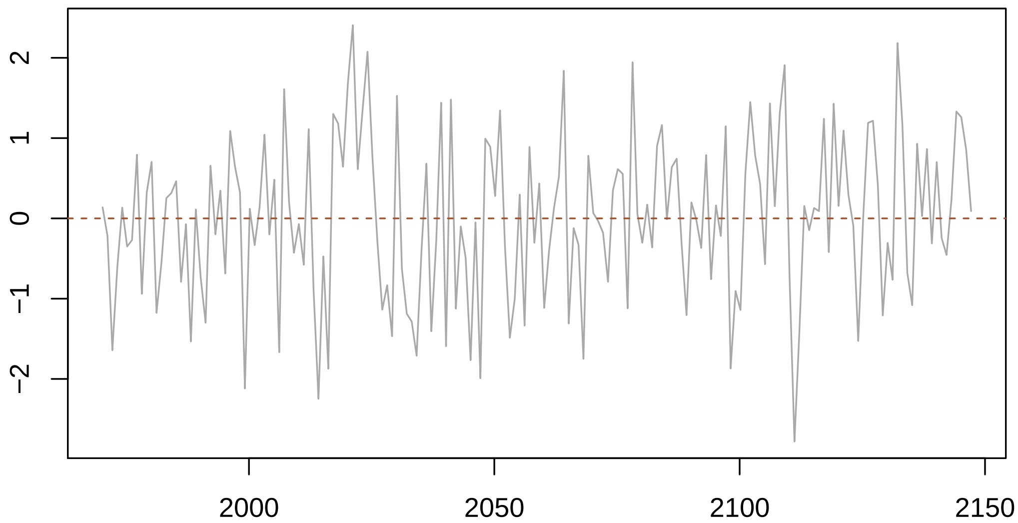


Figure 8.8: Standardised one-step prediction errors of model in Section 7.3.

```
> par(mfrow=c(1,1))
> acf(res,lag=10,main ="")
```

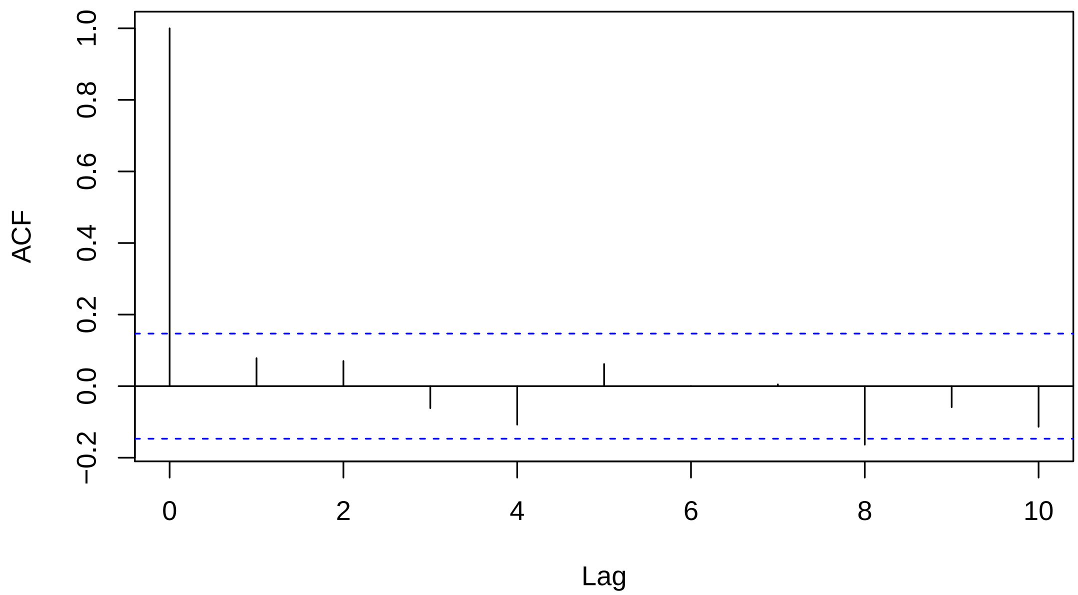


Figure 8.9: Correlogram of standardised one-step prediction errors in Figure 8.8,first 10 lags.

```
> par(mfrow=c(1,1))
> hist(res,breaks=seq(-3.5,3,length.out=30),
      prob=T,col="grey",main = "")
```

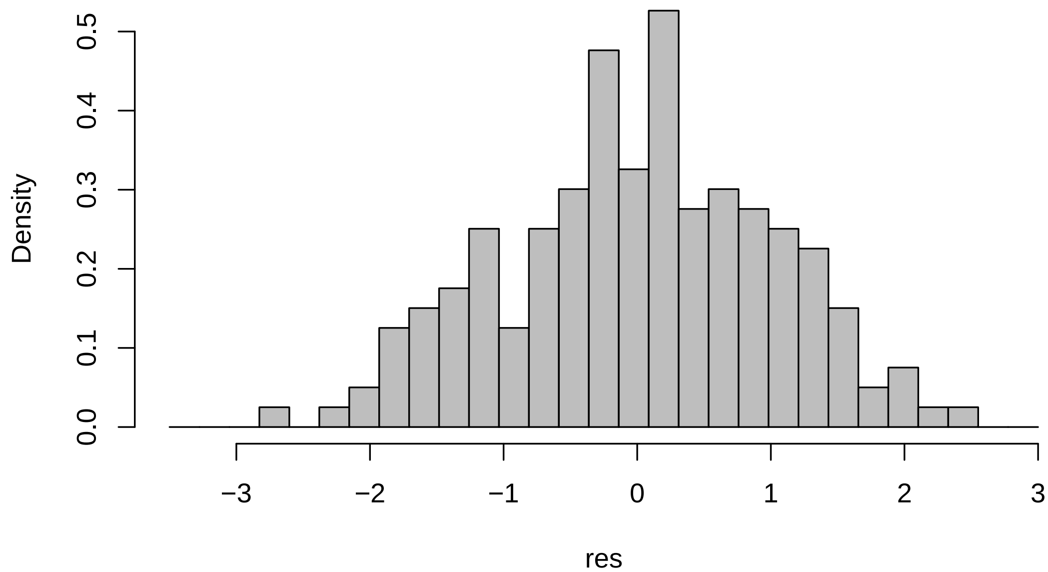


Figure 8.10: Histogram of standardised one-step prediction errors in Figure 8.8.

Diagnostic Tests

- Normality test - PASS

```
> shapiro.test(res)
        Shapiro-Wilk normality test

data:  res
W = 0.9943, p-value = 0.7231
```

- Independence - PASS

```
> Box.test(res, lag = 15, type = "Ljung")
        Box-Ljung test

data:  res
X-squared = 18.6759, df = 15, p-value = 0.2288
> sapply(1:15,function(l){round(Box.test(res, lag=l, type = "Ljung-Box")$p.value,4)})
 [1] 0.2921 0.3662 0.4398 0.3044 0.3526 0.4756 0.5927 0.2255 0.2585 0.1862
[11] 0.2477 0.2618 0.1507 0.1783 0.2288
```

8.4 Forecasting

```
> fn  <- function(params){
   dlmModPoly(order= 1, dV= exp(params[1]) , dW = exp(params[2]))
 }
> fit            <- dlmMLE(data.2, rep(0,2),fn)
> mod            <- fn(fit$par)
> filtered       <- dlmFilter(data.2,mod)
> var            <- unlist(dlmSvd2var(filtered$U.R, filtered$D.R))
> wid             <- qnorm(0.05, lower = FALSE) *sqrt(c(var))
> temp           <- cbind(filtered$f, filtered$f + wid,filtered$f-wid)
> temp           <- ts(temp,start = 1970,frequency =1)
> forecast       <- dlmForecast(filtered,nAhead=5)
> var.2          <- unlist(forecast$Q)
> wid.2          <- qnorm(0.05, lower = FALSE) *sqrt(c(var.2))
> temp.2         <- cbind(forecast$f, forecast$f +wid.2 , forecast$f- wid.2)
> temp.3         <-ts(rbind(temp,temp.2),start = 1970, frequency= 1,end=2008)


> par(mfrow=c(1,1))
> plot(dropFirst(temp.3),plot.type="s",col=c("blue","red","red"),
     xlim = c(1970,2010),lty=c(1,2,2),lwd=c(2,1,1),ylab="")
> lines(data.2,col="darkgrey")
> abline(v=2004,col = "sienna",lwd=3)
> legend("topright",
       leg = c("log fatalities in Norway",
               " filtered level and forecasts",
                "bands"),
       cex = 0.7, lty = c(1, 1,2),
       lwd  =c(1,2,1),col = c("darkgrey","blue","red"),
       bty = "y", horiz = F)
```

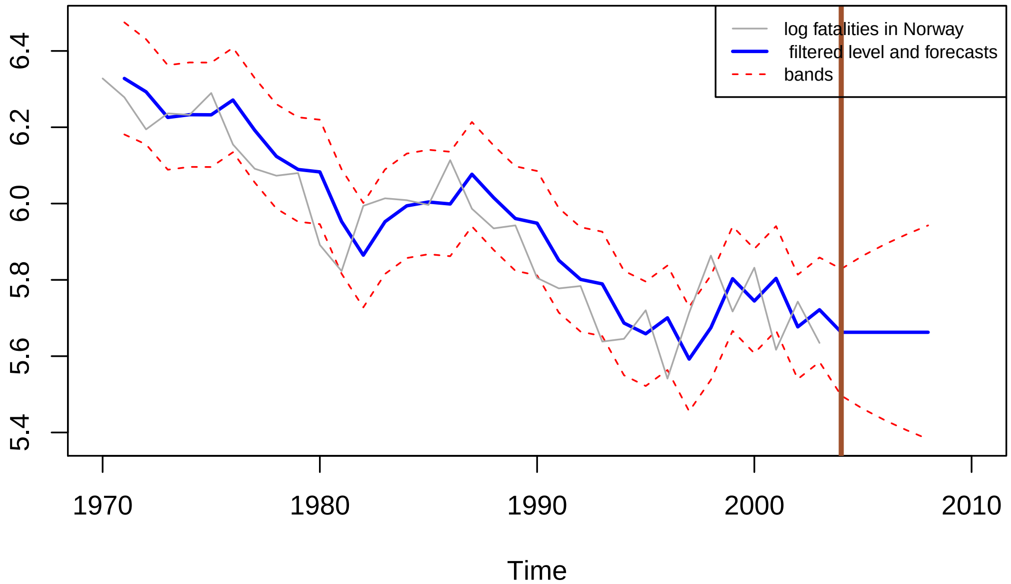


Figure 8.13: Filtered level, and five year forecasts for Norwegian fatalities, including their 90% confidence interval.

```
> fn  <- function(params){
   dlmModPoly(order= 2, dV= exp(params[1]) , dW = exp(params[2:3]))
 }
> fit            <- dlmMLE(data.3, rep(0,3),fn)
> mod            <- fn(fit$par)
> filtered       <- dlmFilter(data.3,mod)
> cov        <- (dlmSvd2var(filtered$U.R, filtered$D.R))
> var        <- (sapply(cov,function(x) mod$FF%*%x%*%t(mod$FF)))+V(mod)
> wid        <- qnorm(0.05, lower = FALSE) *sqrt(c(var))
> filt.trend <- filtered$f[,1]
> temp       <- cbind(filt.trend, filt.trend + wid,filt.trend-wid)
> temp           <- ts(temp[-c(1:2),],start = 1972,frequency =1)
> forecast       <- dlmForecast(filtered,nAhead=5)
> var.2          <- unlist(forecast$Q)
> wid.2          <- qnorm(0.05, lower = FALSE) *sqrt(c(var.2))
> temp.2         <- cbind(forecast$f, forecast$f +wid.2 , forecast$f- wid.2)
> temp.3         <-ts(rbind(temp,temp.2),start = 1972, frequency= 1,end=2008)


> par(mfrow=c(1,1))
> plot(temp.3,plot.type="s",col=c("blue","red","red"),
     xlim = c(1970,2010),lty=c(1,2,2),lwd=c(2,1,1),ylab="")
> lines(data.3,col="darkgrey")
> abline(v=2004,col = "sienna",lwd=3)
> legend("topright",
       leg = c("log fatalities in Norway",
               " filtered level and forecasts",
                "bands"),
       cex = 0.7, lty = c(1, 1,2),
       lwd  =c(1,2,1),col = c("darkgrey","blue","red"),
       bty = "y", horiz = F)
```

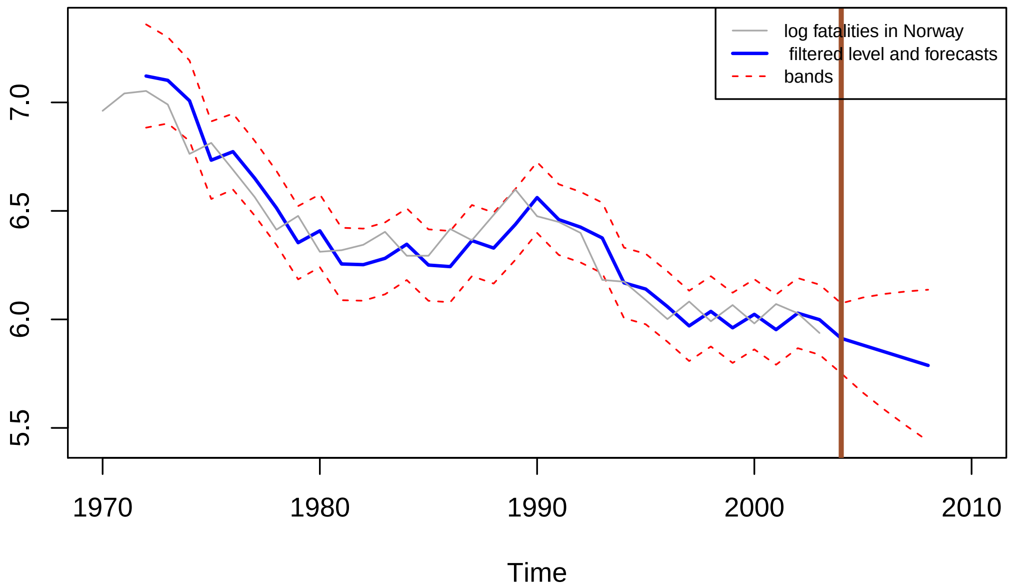


Figure 8.14: Filtered trend, and five-year forecasts for Finnish fatalities, including their 90% confidence limits..

```
> start    <- time(data.1)[1]
> data.1.m <- ts(c(c(data.1)[1:169],rep(NA,23)),frequency =12,start = start)
> x        <- prices
> fn                  <- function(params){
   mod               <- dlmModPoly(order = 1 ) +
                        dlmModSeas(frequency =12)+
                        dlmModReg(x, addInt=FALSE)

  V(mod)              <- exp(params[1])
  diag(W(mod))[1]     <- exp(params[2])
   mod
 }
> fit           <- dlmMLE(data.1.m, rep(0,2),fn)
> mod           <- fn(fit$par)
> filtered      <- dlmFilter(data.1.m,mod)
> forecasted    <- ts(c(filtered$f)[171:193],start = time(data.1)[171],frequency  =12)
> forecasted.mod.1 <- forecasted
> sig.plus.forecast <- filtered$f


> par(mfrow=c(1,1))
> plot(forecasted.mod.1,col="grey",main = "",xlab = "",ylab="")
```

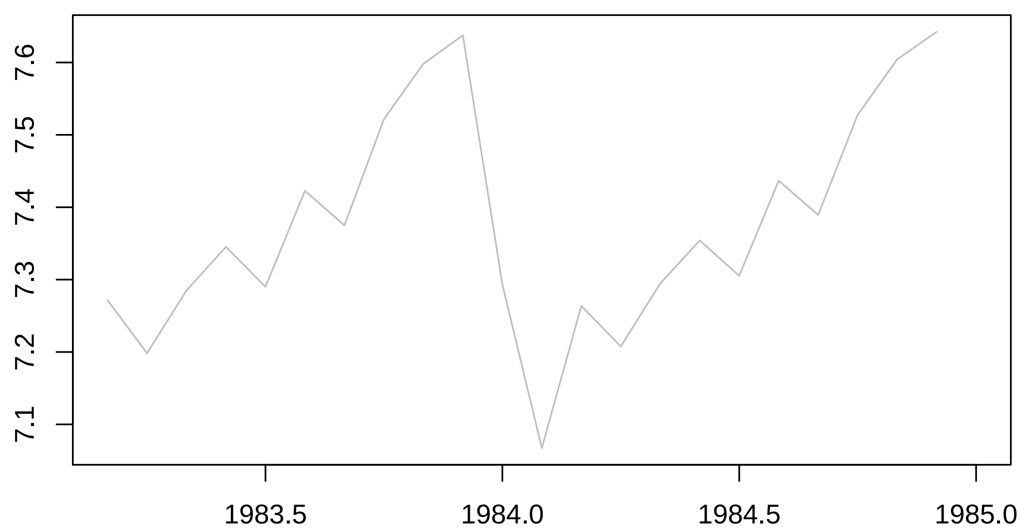


Figure 8.15: Forecasts for t = 170, . . . , 192.

```
> X       <- data.frame(intervention  = intervention, logpprice = prices)
> fn      <- function(params){
   level <- dlmModPoly(order = 1,dV=exp(params[1]))
   seas  <- dlmModSeas(frequency = 12,dV=exp(params[1]),dW=rep(0,11))
   reg   <- dlmModReg(X,addInt=FALSE,dV=exp(params[1]),dW=c(0,0))
   mod   <- level+seas+reg
   diag(W(mod))[1:2] <- c(exp(params[2]),0)
   mod
 }
> fit                <- dlmMLE(data.1, rep(0,2),fn)
> mod                <- fn(fit$par)
> filtered          <- dlmFilter(data.1,mod)
> smoothed          <- dlmSmooth(filtered)
> sm                <- dropFirst(smoothed$s)
> mu                <- c(sm[,1])
> nu                <- c(sm[,2])

> par(mfrow=c(1,1))
> temp <- sm[,1]+sm[,2]+ sm[,13]*intervention+sm[,14]*prices
> temp<- ts(cbind(c(data.1),temp),start = 1969,
                     frequency = 12)
> temp <- window(temp, start = 1982.5)
> temp.2 <- window(sig.plus.forecast, start = 1982.5)
> plot.ts(temp , plot.type="single" , col =c("darkgrey","blue"),lty=c(1,2),
     xlab="",ylab = "log KSI",ylim=c(6.8,7.8))
> legend("topright",
 leg = c("log UK drivers KSI",
  "signal plus forecasts","signal complete model"),
  cex = 0.7, lty = c(1, 2,1),col = c("darkgrey","blue","sienna"),
  pch=c(3,NA),bty = "n", horiz = T)
> lines(sig.plus.forecast, col = "sienna")
```

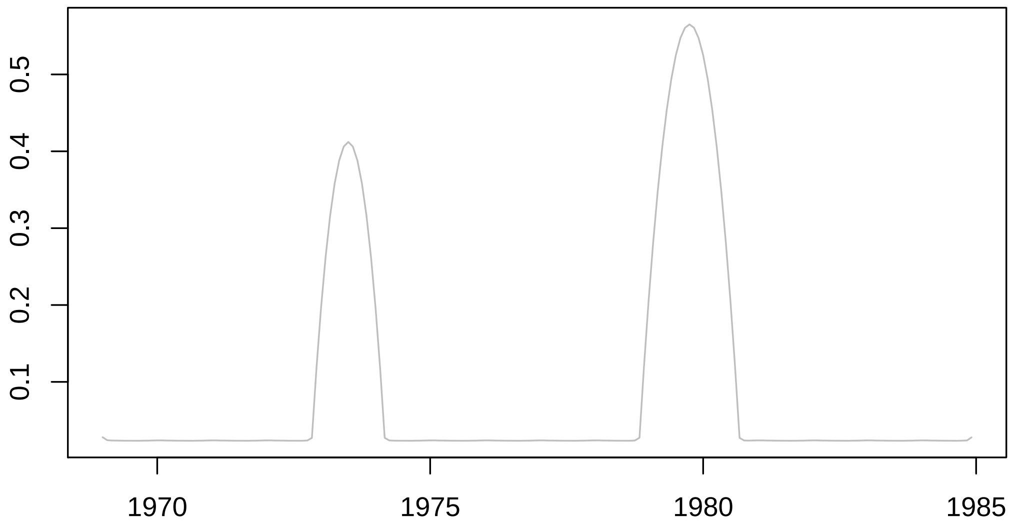


Figure 8.16: The years (1982.5Ű1984) in the time series of the log of numbers of drivers KSI: observed series, forecasts obtained from the analysis up to February 1983, and modelled development for the complete series including an intervention variable for February 1983.

8.5 Missing Observations

```
> data.missing  <- c(data.1)
> data.missing[48:62] <- NA
> data.missing[120:140] <- NA
> data.missing <- ts(data.missing, start= 1969, frequency  = 12)
> fn           <- function(params){
    mod  <- dlmModPoly(order = 1, dV =exp(params[1]), dW = exp(params[2])) +
            dlmModSeas(frequency = 12,dV =exp(params[1]) , dW= rep(0,11))
       return(mod)
 }
> fit           <- dlmMLE(data.missing, rep(0,2),fn)
> mod           <- fn(fit$par)
> filtered      <- dlmFilter(data.missing,mod)
> smoothed      <- dlmSmooth(filtered)
> sm            <- dropFirst(smoothed$s)
> cov           <- (dlmSvd2var(smoothed$U.S, smoothed$D.S))
> var           <- sapply(cov,function(x) {x[1,1]})
> mu            <- ((smoothed$s)[,1])[-1]
> nu            <- ((smoothed$s)[,2])[-1]
> var.ts        <- ts(var[-1],start = 1969, frequency = 12)
> wid           <- qnorm(0.05, lower = FALSE) *sqrt(var.ts)
> temp          <- cbind(mu, mu + wid %o% c(-1, 1))
> temp          <- ts(temp,start = 1969,frequency =12)

> par(mfrow=c(1,1))
> plot(var.ts,col="grey",main = "",xlab = "",ylab="")
```

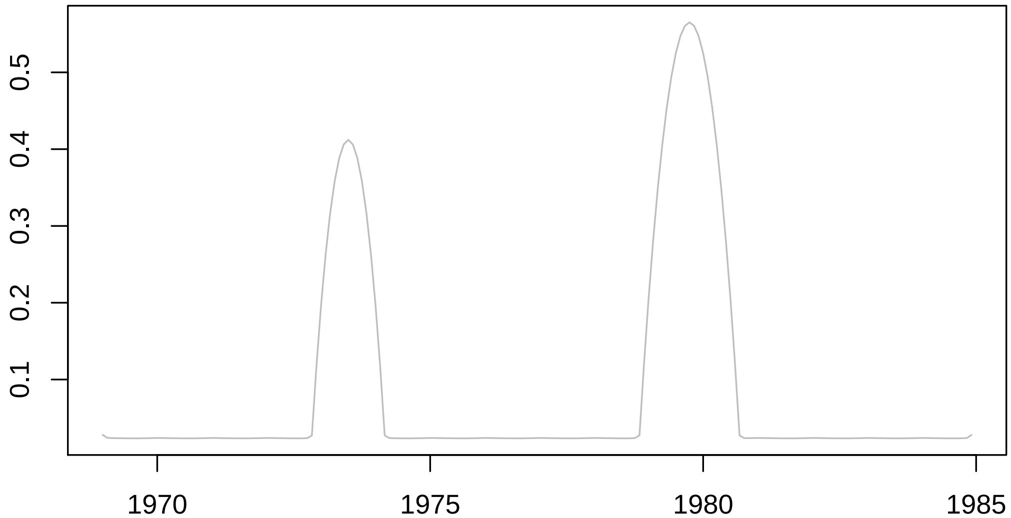


Figure 8.17: Stochastic level estimation error variance for log drivers KSI with observations at t = 48 to 62 and t = 120 to 140 treated as missing.

I could not get the same numbers as the stochastic level estimation variance given in the illustration. May be there are implementation differences of missing treatment in the package as compared to others

```
> par(mfrow=c(1,1))
> plot(temp,plot.type="s",col=c("blue","red","red"),
     ,lty=c(1,2,2),lwd=c(2,1,1),ylab="")
> lines(data.missing,col="darkgrey")
> legend("topright",
       leg = c("log fatalities in Norway",
               " filtered level and forecasts",
                "bands"),
       cex = 0.7, lty = c(1, 1,2),
       lwd  =c(1,2,1),col = c("darkgrey","blue","red"),
       bty = "y", horiz = F)
```

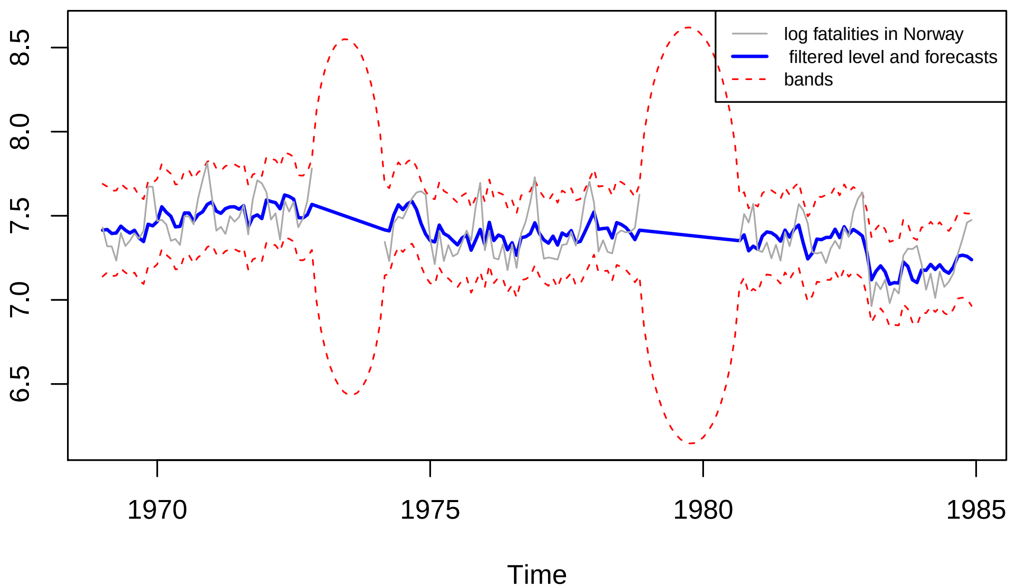


Figure 8.18: Filtered level, and five year forecasts for Norwegian fatalities, including their 90% confidence interval.

I could not get the same numbers as the confidence intervals for stochastic level. May be there are implementation differences of missing treatment in the package as compared to others.

9 Multivariate time series analysis

My fundas on to multivariate state space models are not sound and hence have skipped this section.

10 State space and Box-Jenkins methods for time series analysis

This section is more a formality in this text. I don't think any reader would be even going through this kind of book with out having being exposed to Box-Jenkins methods. In any case, most of the illustrations in the book can be shown with a few lines of code by any econometrics newbie. Just for completion sake, let me quickly write a few lines of code to create these graphs

```
> set.seed(1234)
> x <- rnorm(200)
> par(mfrow=c(1,1))
> plot.ts(x,xlab ="", ylab = "")
```

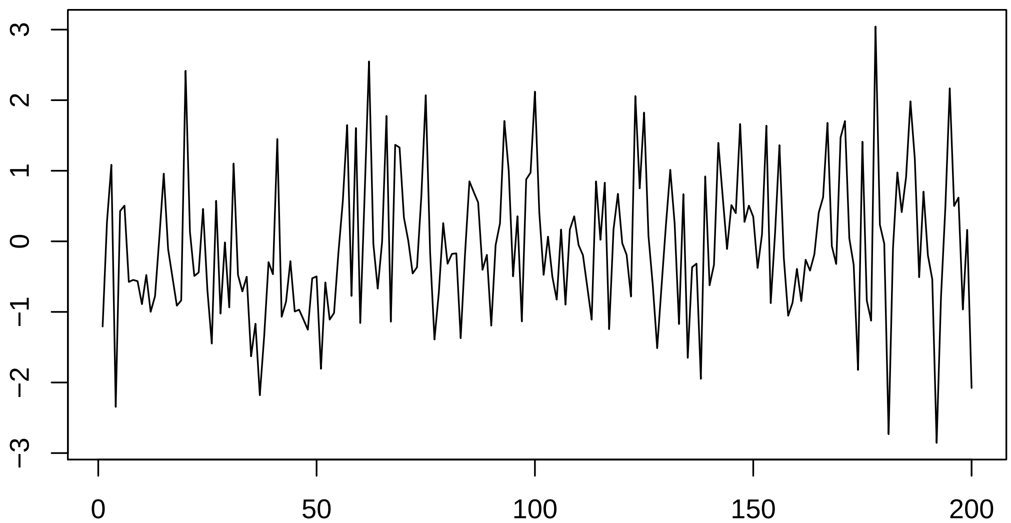


Figure 10.1: Realisation of a random process.

```
> par(mfrow=c(1,1))
> acf(x,lag= 12, main = "")
```

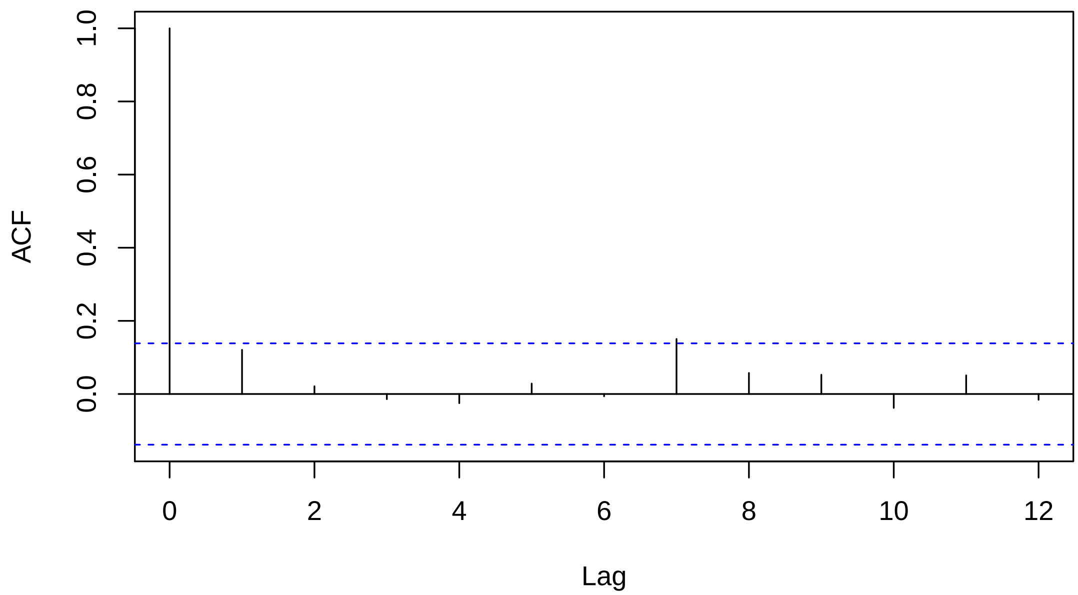


Figure 10.2: Correlogram for lags 1 to 12.

```
> par(mfrow=c(1,1))
> plot.ts(cumsum(x),xlab ="", ylab = "",main="")
```

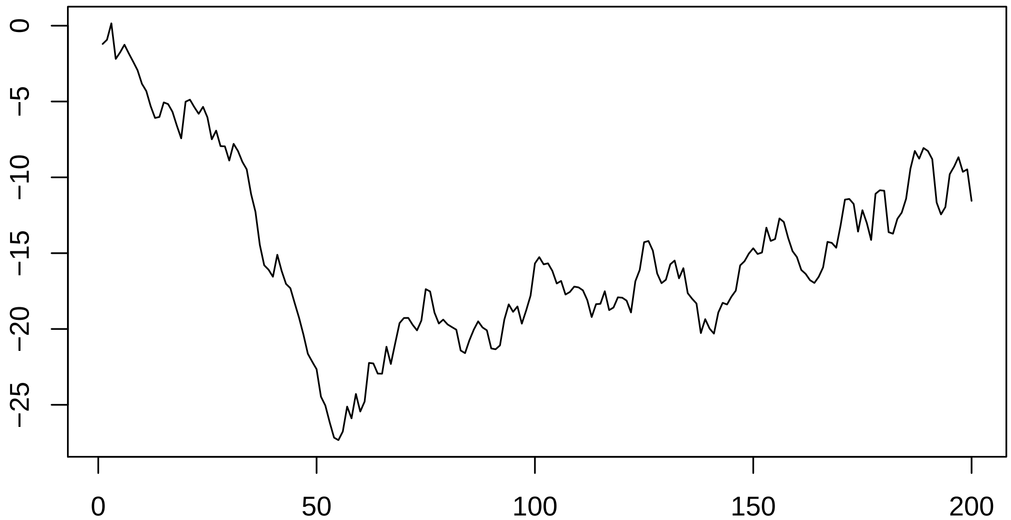


Figure 10.3: Example of a random walk.

```
> par(mfrow=c(1,1))
> acf(cumsum(x),lag= 12, main = "")
```

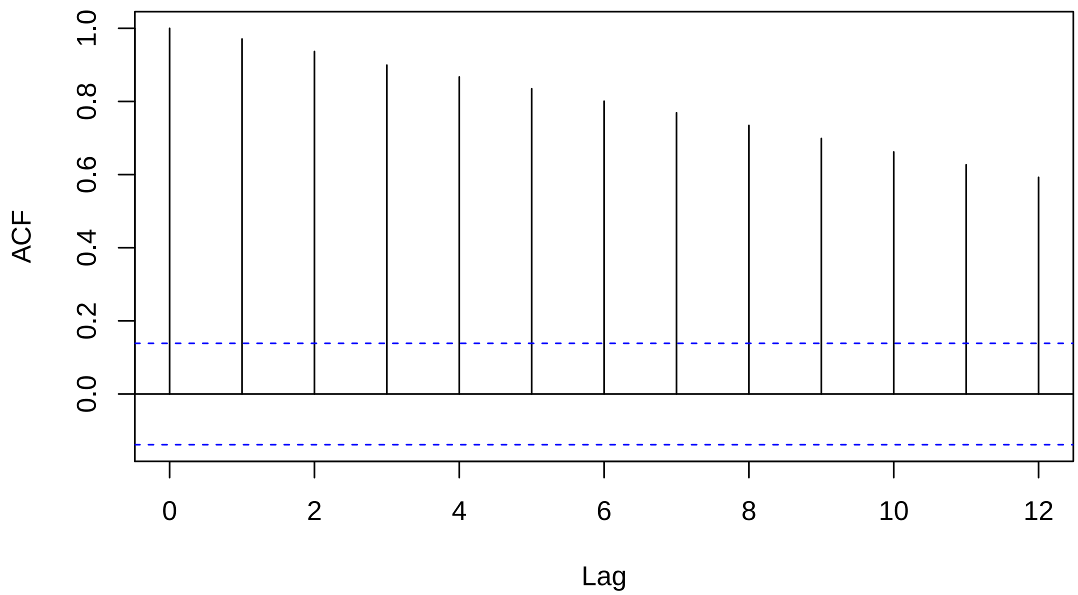


Figure 10.4: Correlogram for lags 1 to 12 for the random walk.

```
> set.seed(1234)
> x  <- arima.sim(model = list(ma=c(0.5)) , n = 200)
> par(mfrow=c(1,1))
> plot.ts(x,xlab ="", ylab = "",main="")
```

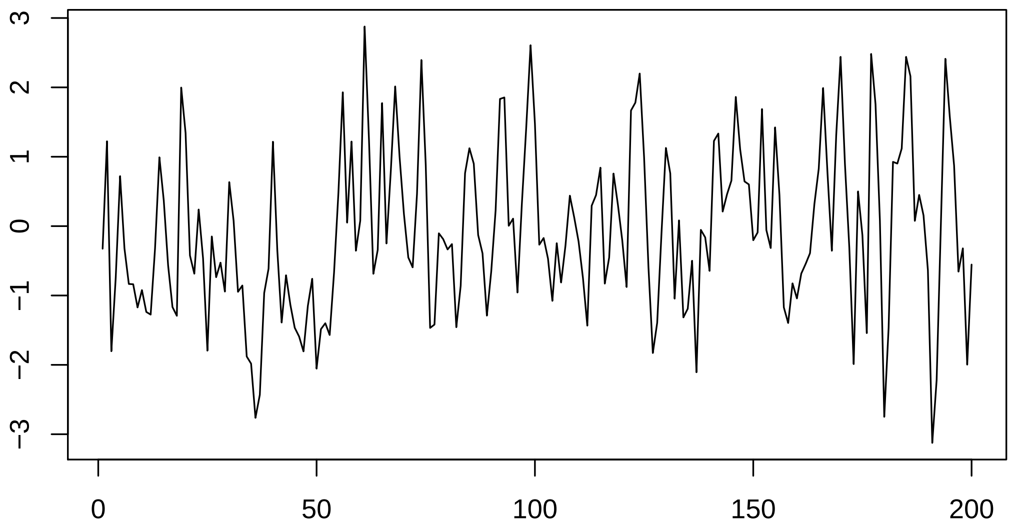


Figure 10.5: Realisation of a MA(1) process with $\theta_1 = 0.5$

```
> par(mfrow=c(1,1))
> acf(x,lag= 12, main = "")
```

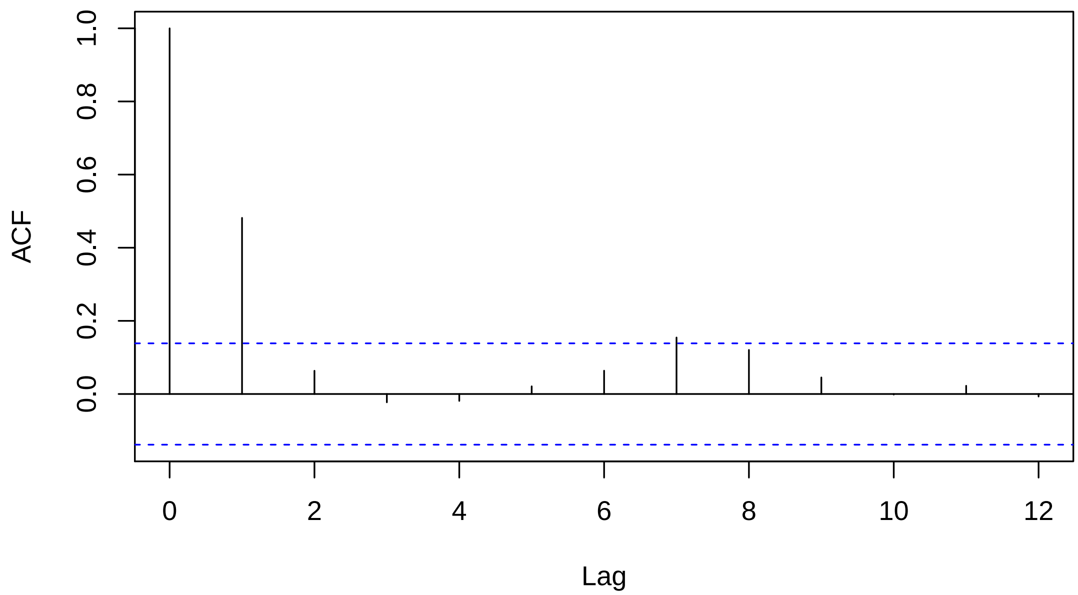


Figure 10.6: Realisation of a MA(1) process with $\theta_1 = 0.5$

```
> set.seed(1234)
> x  <- arima.sim(model = list(ar=0.5) , n = 200)
> par(mfrow=c(1,1))
> plot.ts(x,xlab ="", ylab = "",main="")
```

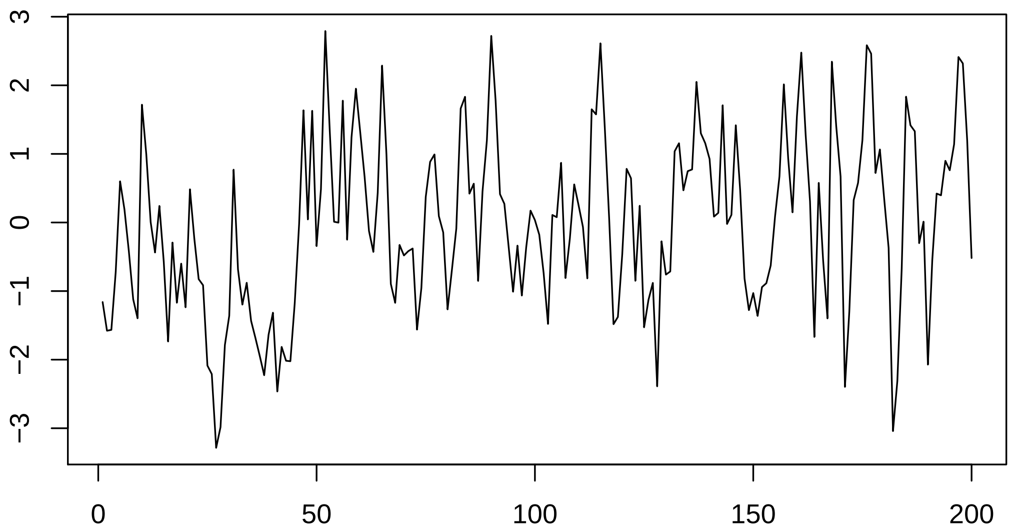


Figure 10.7: Realisation of a AR(1) process with $\phi_1 = 0.5$

```
> par(mfrow=c(1,1))
> acf(x,lag= 12, main = "")
```

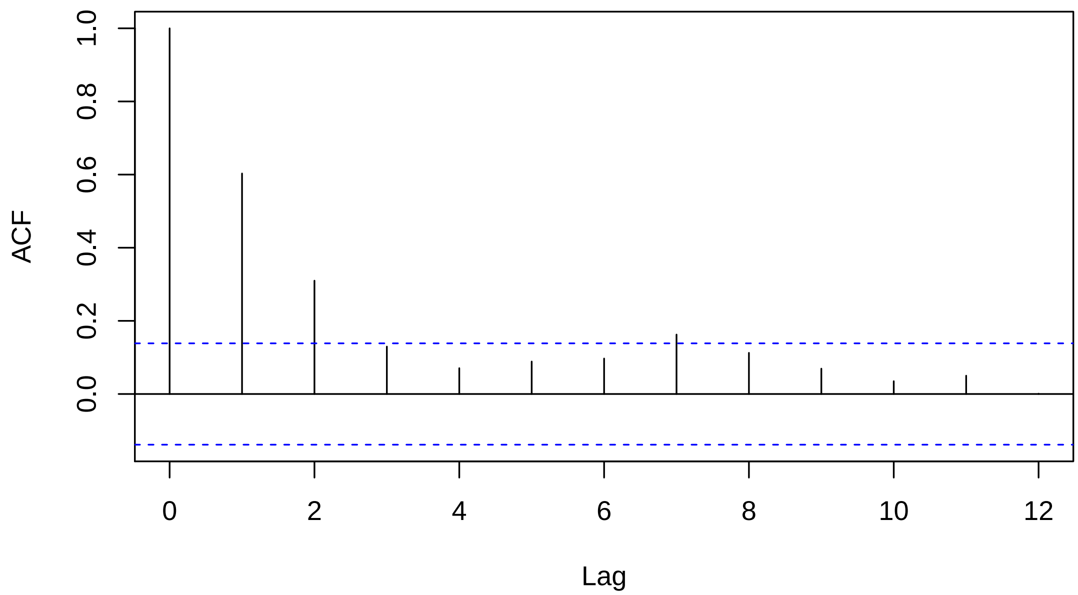


Figure 10.8: Correlogram for lags 1 to 12 for AR(1) data

```
> set.seed(1234)
> x  <- arima.sim(model = list(ar=0.5, ma =c(0.5)) , n = 200)
> par(mfrow=c(1,1))
> plot.ts(x,xlab ="", ylab = "",main="")
```

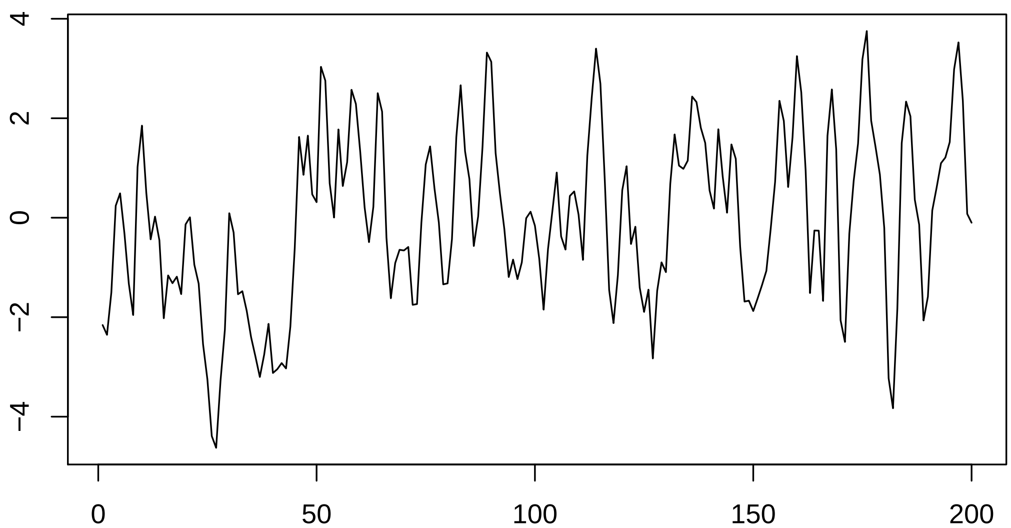


Figure 10.9: Realisation of a ARMA(1,1) process with $\phi_1 = \theta_1 = 0.5$

```
> par(mfrow=c(1,1))
> acf(x,lag= 12, main = "")
```

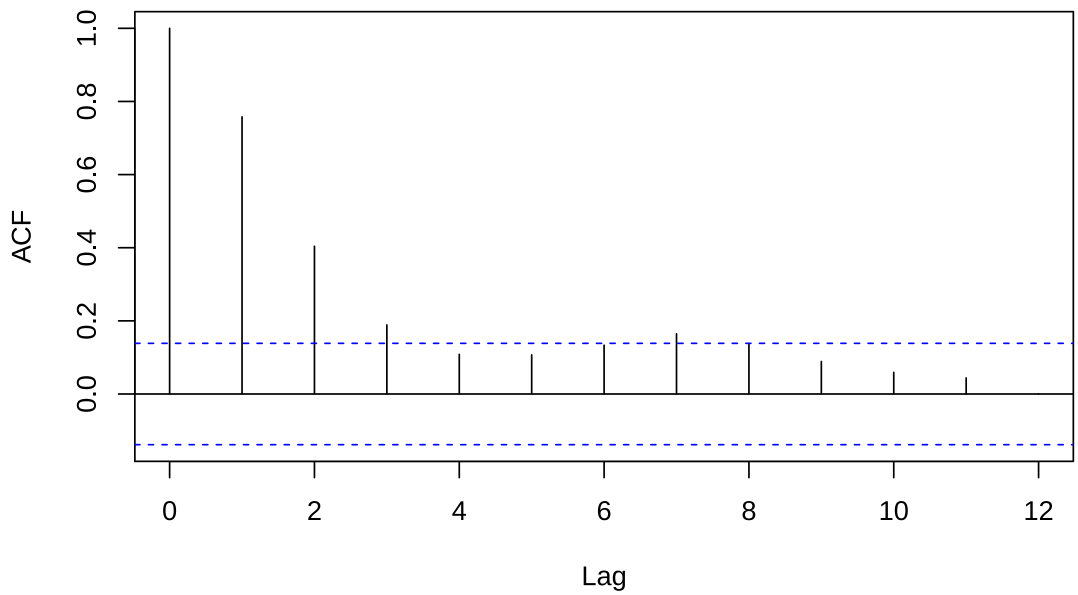


Figure 10.10: Correlogram for lags 1 to 12 for ARMA(1,1) data