LECTURE NOTES 23

TRYGGVI JÓNSSON

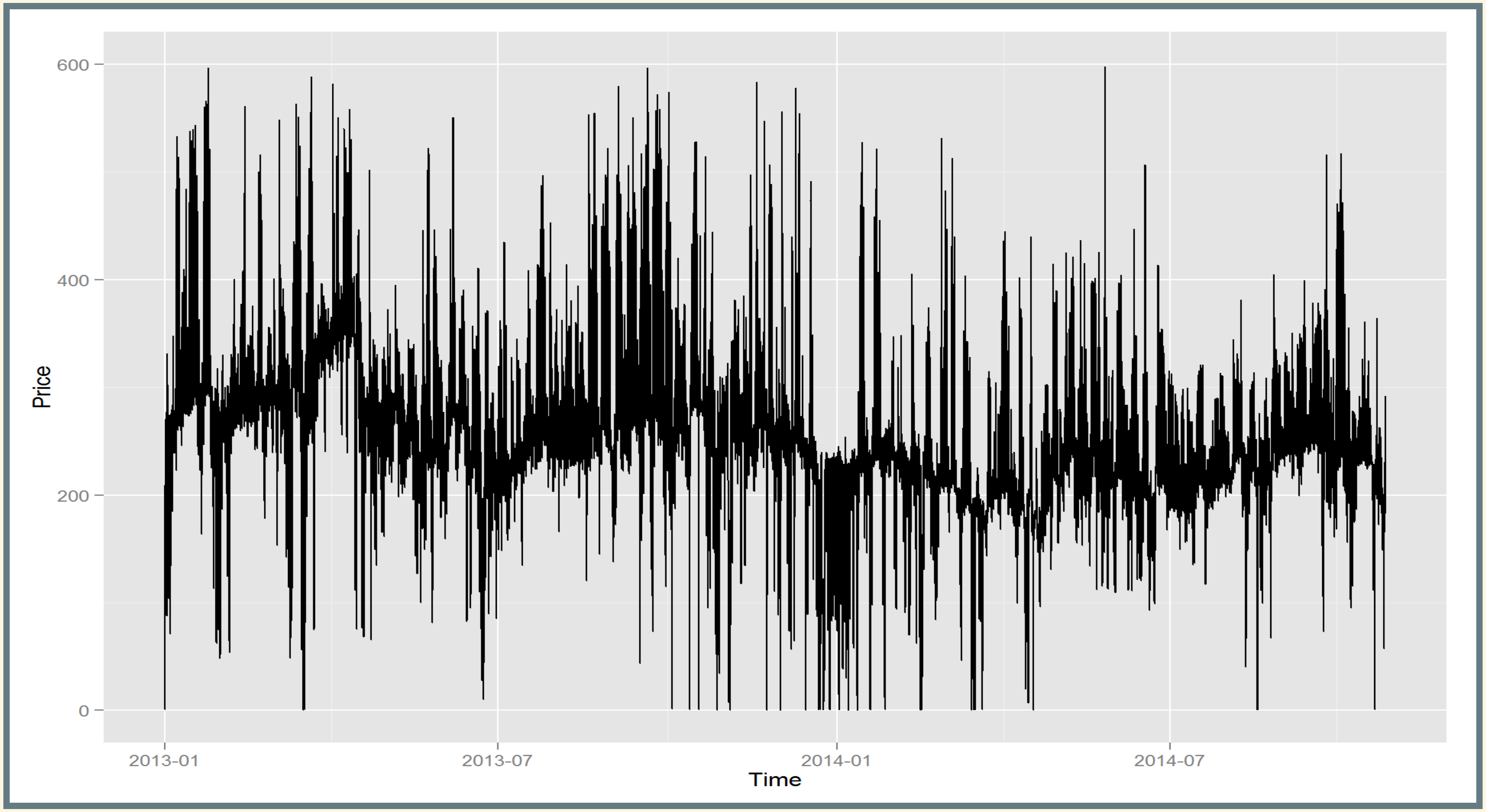
CONTENT OUTLINE

Recursive and adaptive estimation:

- Introduction to Chapter 11
- Recursive LS, Section 11.1

WHY RECURSIVE AND ADAPTIVE ESTIMATION?

- As time passes we get more information
- Models are approximations
- The best approximation change over time
- Makes it possible to produce software which learns as new data becomes available



Price
600
400
200
0
2013-01
2013-07
2014-01
2014-07
Time

TYPES OF MODELS CONSIDERED

REG:

$$Y_t = \mu + \beta_1 U_{1,t} + \beta_2 U_{2,t} + \ldots + \beta_m U_{m,t} + \varepsilon_t$$

FIR:

$$\begin{aligned} Y_t &= \mu + \omega(B) U_t + \varepsilon t \\ &= \mu + \omega_0 U_t + \omega_1 U_{t-1} + \ldots + \omega_s U_{t-s} + \varepsilon_t \end{aligned}$$

AR:

$$\varphi(B)Y_t = \mu + \varepsilon_t \Leftrightarrow$$

$$Y_t = \mu - \varphi_1 Y_{t-1} - \varphi_2 Y_{t-2} - \ldots - \varphi_p Y_{t-p} + \varepsilon_t$$

ARX:

$$\varphi(B)Y_t = \mu + \omega(B)U_t + \varepsilon_t \Leftrightarrow$$

$$Y_t = \mu - \varphi_1 Y_{t-1} - \ldots - \varphi_p Y_{t-p}$$

$$+\omega_0 U_t + \ldots + \omega_s U_{t-s} + \varepsilon_t$$

ARMA:

$$\varphi(B)Y_t = \mu + \theta(B)\varepsilon_t \Leftrightarrow$$

$$Y_t = \mu - \varphi_1 Y_{t-1} - \varphi_2 Y_{t-2} - \ldots - \varphi_p Y_{t-p}$$

$$+\theta_1\varepsilon_{t-1} + \ldots + \theta_q\varepsilon_{t-q} + \varepsilon_t$$

ARMAX:

$$\varphi(B)Y_t = \mu + \omega(B)U_t + \theta(B)\varepsilon_t \Leftrightarrow$$

$$Y_t = \mu - \varphi_1 Y_{t-1} - \ldots - \varphi_p Y_{t-p}$$

$$+\omega_0 U_t + \ldots + \omega_s U_{t-s}$$

$$+\theta_1\varepsilon_{t-1} + \ldots + \theta_q\varepsilon_{t-q} + \varepsilon_t$$

GENERIC FORM OF THE MODELS CONSIDERED

$$Y_t = \mathbf{x}_t^T \theta + \varepsilon_t = \theta_1 x_{1,t} + \theta_2 x_{2,t} + \ldots + \theta_n x_{p,t} + \varepsilon_t$$

Example:

$$Y_t = \mu \cdot \underbrace{1}_{x_{1,t}} + \varphi_2 \cdot \underbrace{(-Y_{t-2})}_{x_{2,t}} + \omega_1 \cdot \underbrace{U_{t-1}}_{x_{3,t}} + \varepsilon_t$$

LS-ESTIMATE AT TIME T

Model:

$$Y_t = x_t^T \theta + \varepsilon_t$$

Data (x may contain lagged values of the “real” input):

$$Y_1, Y_2, Y_3, Y_4, \ldots, Y_{t-1}, Y_t$$
$$x_1, x_2, x_3, x_4, \ldots, x_{t-1}, x_t$$

LS-estimate based on t observations:

$$\widehat{\boldsymbol{\theta}}_t = \arg\min_{\hat{\theta}} S_t(\boldsymbol{\theta}) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

$$S_t(\boldsymbol{\theta}) = \sum_{s=1}^{t} (Y_s - \mathbf{x}_s^T\boldsymbol{\theta})^2$$

FROM ONE TIME STEP TO THE NEXT (IN AN EASY WAY)

The trick is to realize that:

$$\mathbf{R}_t = \mathbf{X}^T\mathbf{X} = \mathbf{x}_1\mathbf{x}_1^T + \mathbf{x}_2\mathbf{x}_2^T + \ldots + \mathbf{x}_t\mathbf{x}_t^T = \sum_{s=1}^{t}\mathbf{x}_s\mathbf{x}_s^T$$

$$\mathbf{h}_t = \mathbf{X}^T\mathbf{Y} = \mathbf{x}_1\mathbf{Y}_1 + \mathbf{x}_2\mathbf{Y}_2 + \ldots + \mathbf{x}_t\mathbf{Y}_t = \sum_{s=1}^{t}\mathbf{x}_s\mathbf{Y}_s$$

Where:

$$\mathbf{x}_t\mathbf{x}_t^T = \begin{bmatrix} x_{1,t}x_{1,t} & x_{1,t}x_{2,t} & \cdots & x_{1,t}x_{p,t} \\ x_{2,t}x_{1,t} & x_{2,t}x_{2,t} & \cdots & x_{2,t}x_{p,t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p,t}x_{1,t} & x_{p,t}x_{2,t} & \cdots & x_{p,t}x_{p,t} \end{bmatrix}$$

$$\mathbf{x}_t\mathbf{Y}_t = \begin{bmatrix} x_{1,t}Y_t \\ x_{2,t}Y_t \\ \vdots \\ x_{p,t}Y_t \end{bmatrix}$$

Then:

$$\widehat{\boldsymbol{\theta}}_t = \mathbf{R}_t^{-1}\mathbf{h}_t$$

$$\mathbf{R}_t = \sum_{s=1}^{t} \mathbf{x}_s \mathbf{x}_s^T = \mathbf{x}_t \mathbf{x}_t^T + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^T = \mathbf{x}_t \mathbf{x}_t^T + \mathbf{R}_{t-1}$$

$$\mathbf{h}_t = \sum_{s=1}^{t} \mathbf{x}_s \mathbf{Y}_s = \mathbf{x}_t \mathbf{Y}_t + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{Y}_s = \mathbf{x}_t \mathbf{Y}_t + \mathbf{h}_{t-1}$$

Initialization:

$$\mathbf{R}_0 = \mathbf{0} \text{ (matrix of zeros)}$$

$$\mathbf{h}_0 = \mathbf{0} \text{ (vector of zeros)}$$

First estimate of $\widehat{\boldsymbol{\theta}}_t$ when $\mathbf{R}_t$ becomes invertible

OTHER FORMULATIONS - I

Eliminating $\mathbf{h}_t$:

$$\mathbf{R}_t = \mathbf{x}_t \mathbf{x}_t^T + \mathbf{R}_{t-1}$$

$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} + \mathbf{R}_t^{-1} \mathbf{x}_t \left(Y_t - \mathbf{x}_t^T \widehat{\boldsymbol{\theta}}_{t-1} \right)$$

OTHER FORMULATIONS - II

Eliminating $\mathbf{h}_t$ and avoiding matrix-inversion:

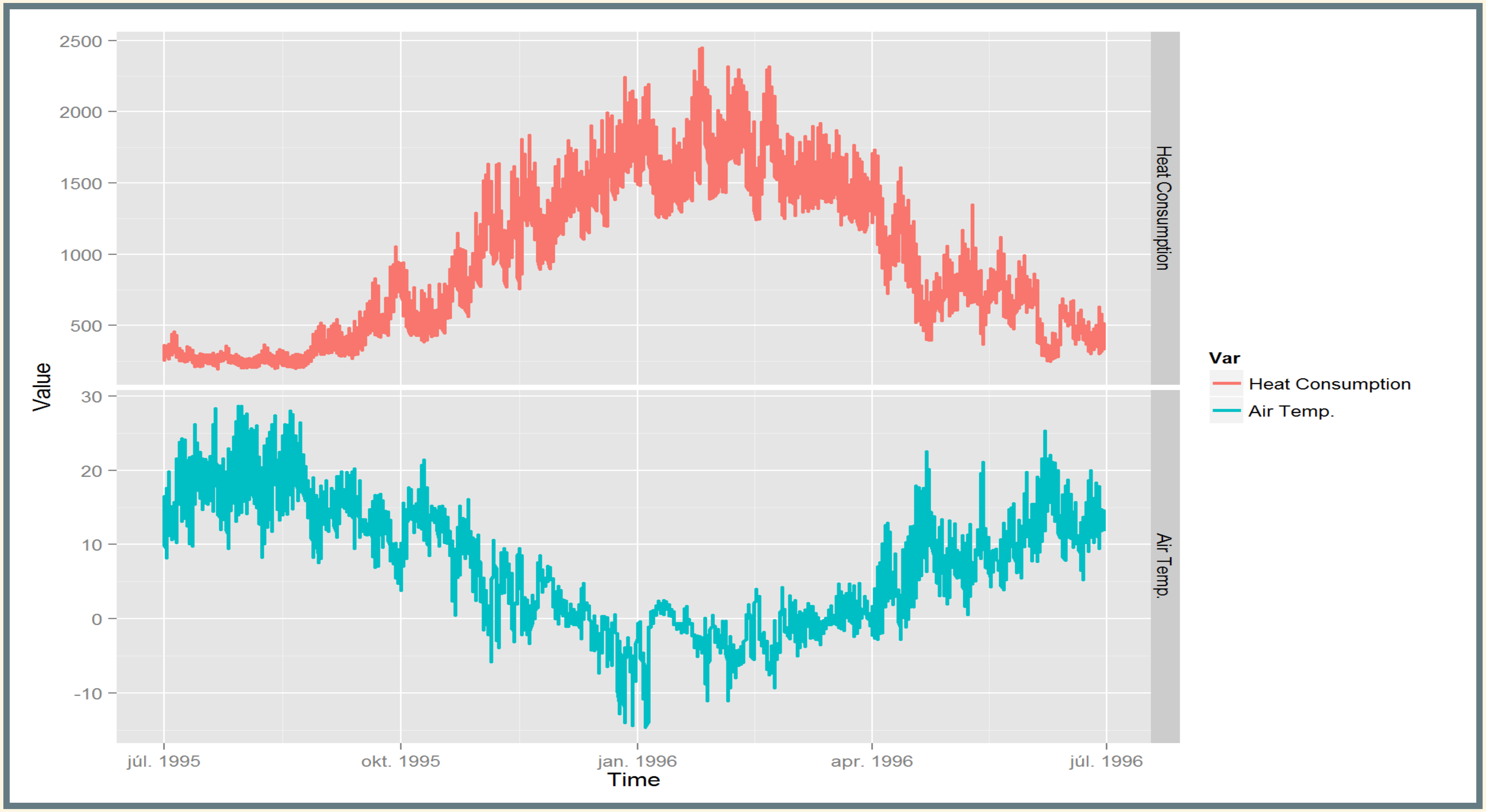
$$\mathbf{K}_t = \frac{\mathbf{P}_{t-1}\mathbf{x}_t}{1 + \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t}$$

$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} + \mathbf{K}_t\left(Yt - \mathbf{x}_t^T\widehat{\boldsymbol{\theta}}_{t-1}\right)$$

$$\mathbf{P}_t = \mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1}\mathbf{x}_t\mathbf{x}_t^T\mathbf{P}_{t-1}}{1 + \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t}$$

EXAMPLE - HEAT CONSUMPTION REVISITED

$$HC_t = \mu + \theta_1 T_t + \varepsilon_t$$



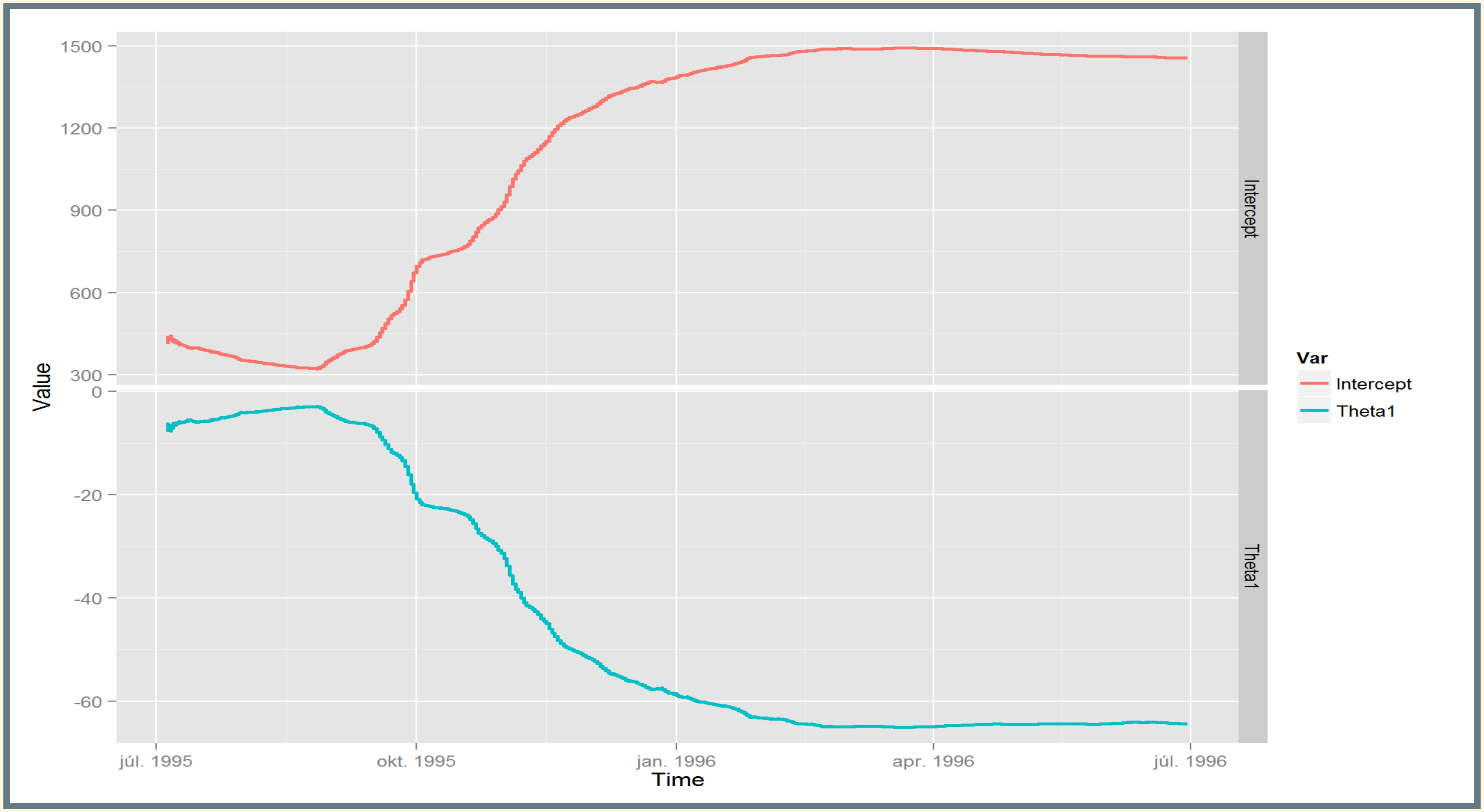
R - INITIALIZATION

```
dat <- read.csv("VEKS.csv")

#Use lm to setup matrices
mdl <- lm(HC.f~Ta.f,data=dat,x=T,y=T)

XX <- as.matrix(mdl$x)
Y <- as.matrix(mdl$y)

# Initialize Rt. ht and theta
Rt <- matrix(0,2,2)

theta <- matrix(0,2,1)

# Initialize empty matix and vector for storing
theta.store <- matrix(NA,nrow(XX),2)
yhat <- rep(NA,nrow(XX))

lmbd <- 1
```

R - RECURSIVE ESTIMATION

```
for(tt in 1:(nrow(XX)-1)){
  xt <- XX[tt,]
  #Update Rt
  Rt <- xt%*%t(xt) + lmbd*Rt
  # Use "try" until Rt becomes invertible
  theta.try <- try(theta + solve(Rt)%*%xt%*%(Y[tt] - t(xt)%*%theta),silent=T)
  if(class(theta.try) != "try-error"){
    #Update theta
    theta <- theta.try
    #Store
    theta.store[tt,] <- theta
    #Predict
    yhat[tt+1] <- t(XX[(tt+1),]) %*% theta
  }
}
```

PARAMETERS



1500
1200
900
600
300
0
-20
-40
-60
Value
Intercept
Theta1
júl. 1995
okt. 1995
jan. 1996
apr. 1996
júl. 1996
Time
Var
Intercept
Theta1

FORGETTING OLD OBSERVATIONS

So far we have a way of updating the estimates as the data set grows

If we want a method which forgets old observations we apply weights which start at 1 and goes to 0 when observations gets old

$$\widehat{\boldsymbol{\theta}}_t = \arg\min_{\hat{\theta}} S_t(\boldsymbol{\theta}) = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{Y}$$

$$S_t(\boldsymbol{\theta}) = \sum_{s=1}^{t} \beta(t, s)(Y_s - \mathbf{x}_s^T\boldsymbol{\theta})^2$$

where $\mathbf{W} = diag(\beta(t,1), \beta(t,2), \ldots, \beta(t,t-1), 1)$

$\beta(t, s)$ expresses how we assign weights to old observations

EXPONENTIAL DECAY OF WEIGHTS

- Let's first consider $\beta(t, s) = \lambda^{t-s} (0 < \lambda \leq 1)$
  - $\lambda = 1$: What we did with the previous algorithms
  - $\lambda < 1$: We forget in an exponential manner
- In the general case it turns out that the estimates can be updated recursively if the sequence of weights can be written as
  - $\beta(t, s) = \lambda(t)\beta(t-1, s), \quad 1 \leq s \leq t-1$
  - $\beta(t, t) = 1$

THE ADAPTIVE RECURSIVE LS ALGORITHM

$$\mathbf{R}_t = \mathbf{x}_t \mathbf{x}_t^T + \lambda(t) \mathbf{R}_{t-1}$$

$$\mathbf{h}_t = \mathbf{x}_t \mathbf{Y}_t + \lambda(t) \mathbf{h}_{t-1} \widehat{\boldsymbol{\theta}}_t = \mathbf{R}_t^{-1} \mathbf{h}_t$$

OTHER FORMULATIONS - I

Eliminating $\mathbf{h}_t$:

$$\mathbf{R}_t = \mathbf{x}_t \mathbf{x}_t^T + \lambda(t) \mathbf{R}_{t-1}$$

$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} + \mathbf{R}_t^{-1} \mathbf{x}_t \left(Y_t - \mathbf{x}_t^T \widehat{\boldsymbol{\theta}}_{t-1} \right)$$

OTHER FORMULATIONS - II

Eliminating $\mathbf{h}_t$ and avoiding matrix-inversion:

$$\mathbf{K}_t = \frac{\mathbf{P}_{t-1}\mathbf{x}_t}{\lambda(t) + \mathbf{x}_t^T \mathbf{P}_{t-1}\mathbf{x}_t}$$

$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} + \mathbf{K}_t \left(Yt - \mathbf{x}_t^T \widehat{\boldsymbol{\theta}}_{t-1} \right)$$

$$\mathbf{P}_t = \mathbf{P}_{t-1} - \frac{1}{\lambda(t)} \left(\frac{\mathbf{P}_{t-1}\mathbf{x}_t\mathbf{x}_t^T\mathbf{P}_{t-1}}{\lambda(t) + \mathbf{x}_t^T \mathbf{P}_{t-1}\mathbf{x}_t} \right)$$

CONSTANT FORGETTING

If $\lambda(t) = \lambda$ we call λ the forgetting factor and define the memory as

$$T_0 = \sum_{i=0}^{\infty} \lambda^i = 1 + \lambda + \lambda^2 + \lambda^3 + \lambda^4 + \ldots = \frac{1}{1-\lambda}$$

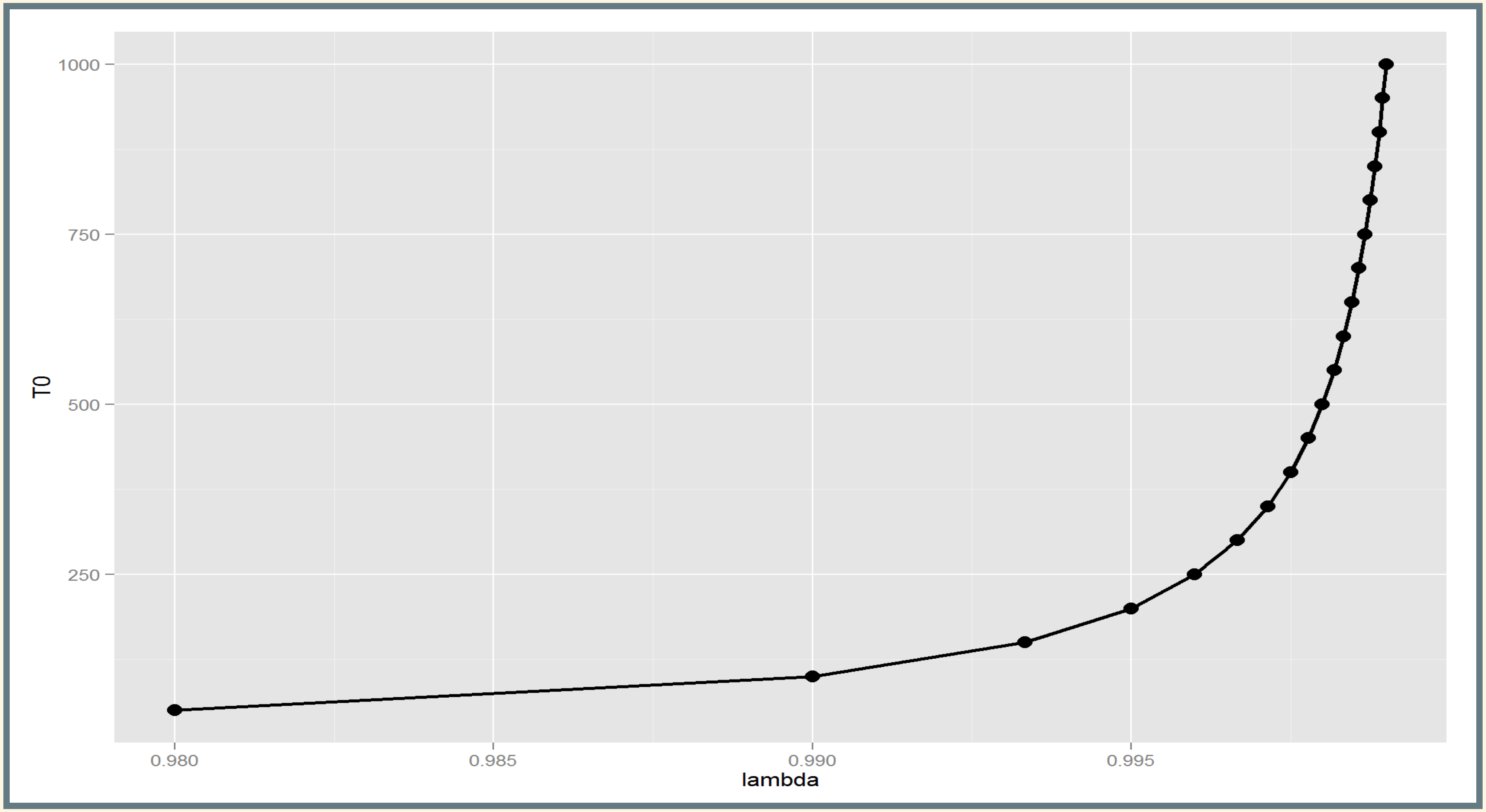
Given a data set an optimal value of λ can be found by "trial and error"

It is often a good idea to select the values of λ to be investigated so that the corresponding values of T_0 are approximately equidistant

The criteria to evaluate may depend on the application, but the sum of squared one-step prediction errors is often appropriate

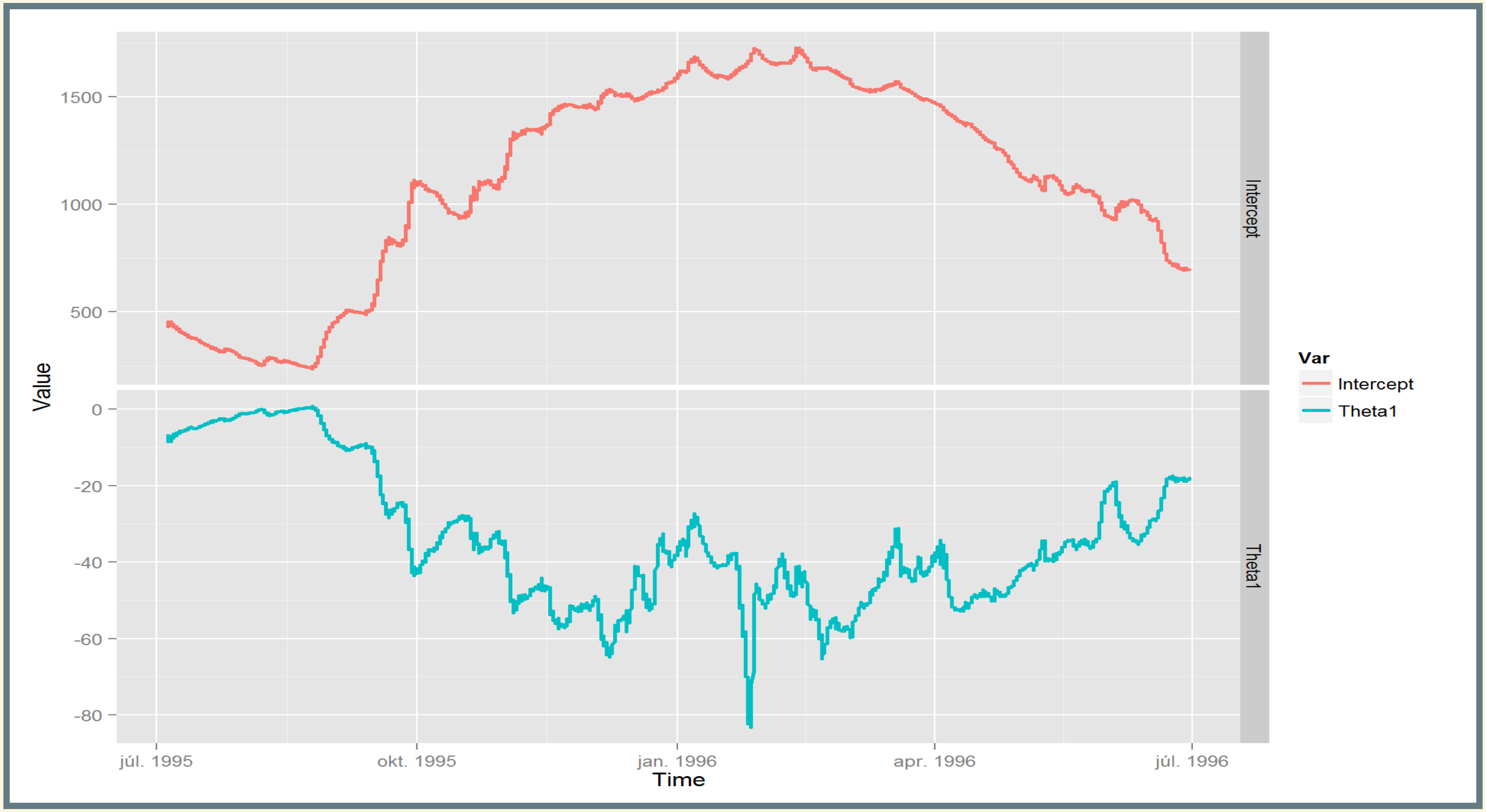
The initialization period should be excluded from the evaluation

λ vs. T_0

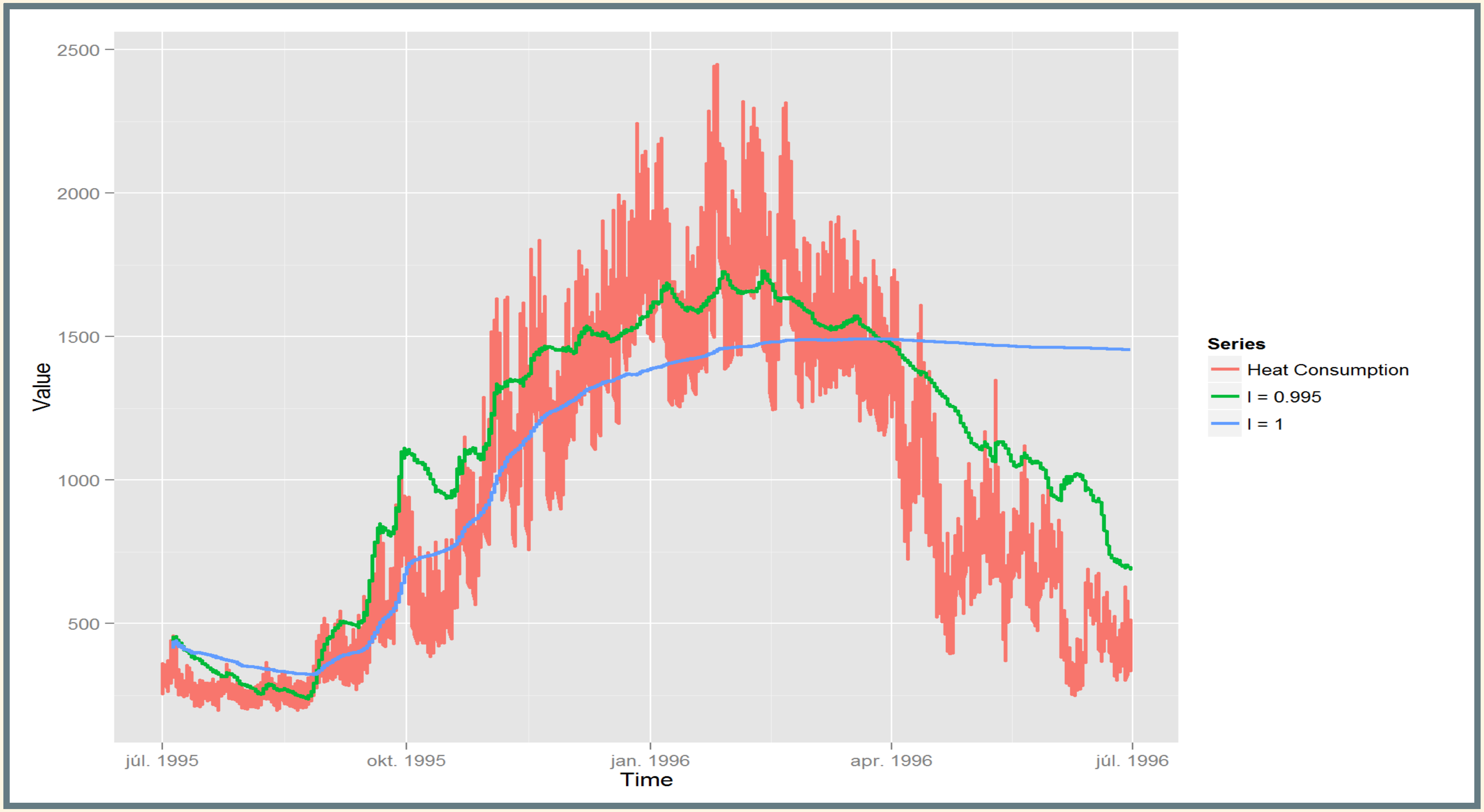


HEAT CONSUMPTION CONTINUED

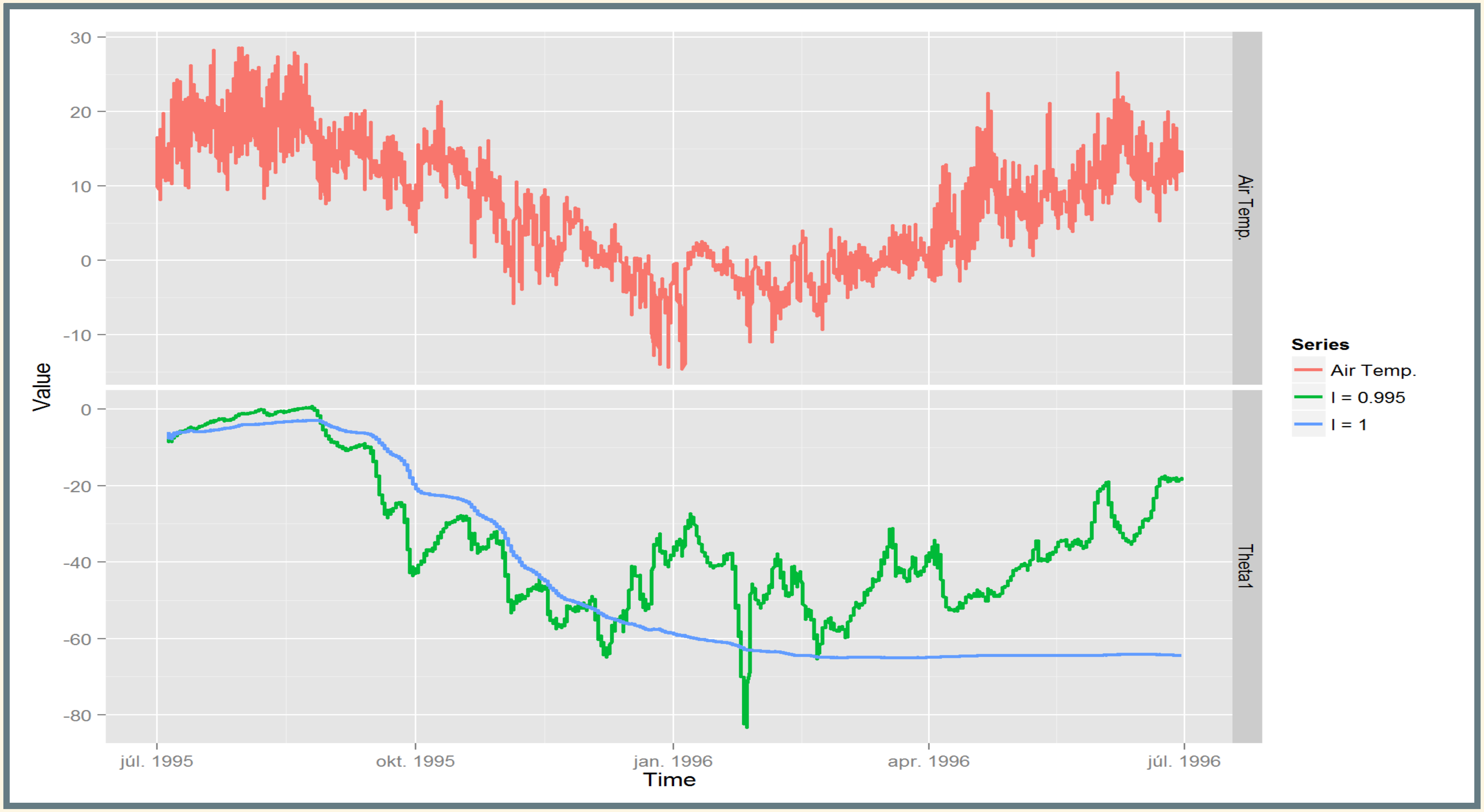
$$HC_t = \mu + \theta_1 T_t + \varepsilon_t, \lambda = 0.995$$



THE INTERCEPT



THE SLOPE



30
20
10
0
-10
Air Temp.
Value
0
-20
-40
-60
-80
Theta1
júl. 1995
okt. 1995
jan. 1996
apr. 1996
júl. 1996
Time
Series
Air Temp.
l = 0.995
l = 1