MATH 8090: State-Space Models I

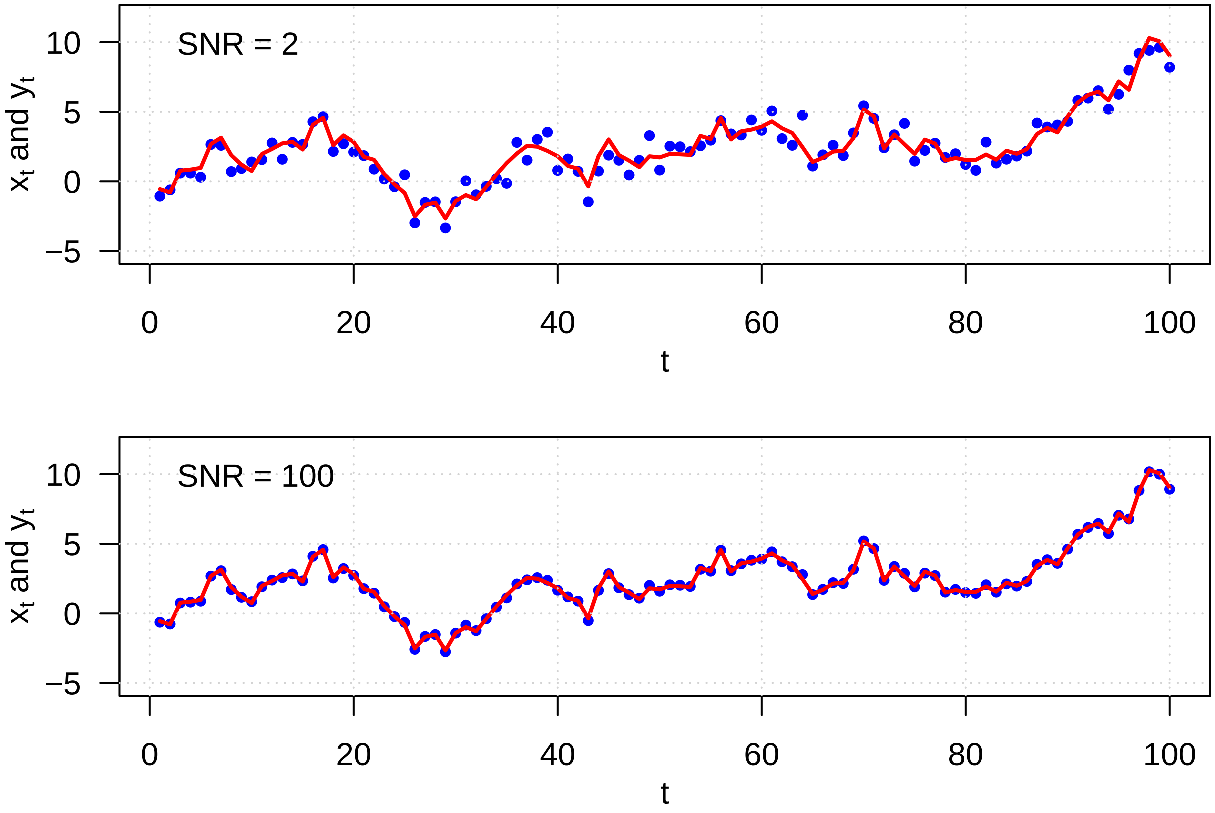
Whitney Huang, Clemson University

11/11-13/2025

Contents

Simulate data from local level model

```r
set.seed(123)
m.1 <- 0; sig.1 <- 1
sig2.V <- 1
X.1 <- rnorm(1, mean = m.1, sd = sqrt(sig.1))
X <- cumsum(c(X.1, rnorm(99, sd = sqrt(sig2.V))))
W <- rnorm(100)
SNR <- 2
Y.2 <- X + W * sqrt(sig2.V / SNR)
SNR <- 100
Y.100 <- X + W * sqrt(sig2.V / SNR)

par(las = 1, mgp = c(2, 1, 0), mar = c(3.5, 3.5, 1, 0.6), mfrow = c(2, 1))
plot(Y.2, col = "blue", pch = 16, cex = 0.75, xlab = "t",
     ylab = expression(paste(x[t], " and ", y[t])), main = "", ylim = c(-5.25, 12))
lines(X, col = "red", lwd = 2)
grid()
legend("topleft", legend = "SNR = 2", bty = "n")
plot(Y.100, col = "blue", pch = 16, cex = 0.75, xlab = "t",
     ylab = expression(paste(x[t], " and ", y[t])), main = "", ylim = c(-5.25, 12))
lines(X, col = "red", lwd = 2)
grid()
legend("topleft", legend = "SNR = 100", bty = "n")
```



Prediction, Filtering and Smoothing for the Local Level Model

This example is taken from Shumway and Stoffer (2017), specifically from example 6.5.

Generate data

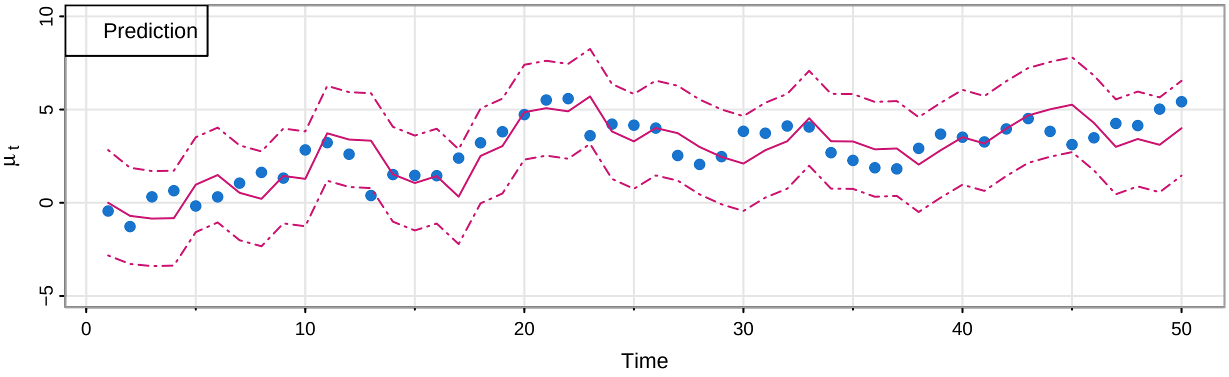
```
set.seed(1)
n = 50
w = rnorm(n + 1, 0, 1)
v = rnorm(n, 0, 1)
mu = cumsum(w)
y = mu[-1] + v
```

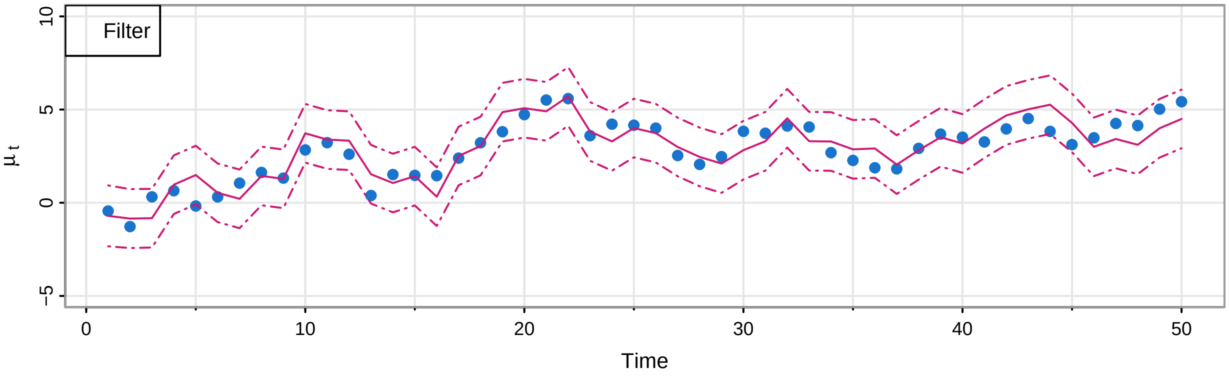
Filtering and smoothing

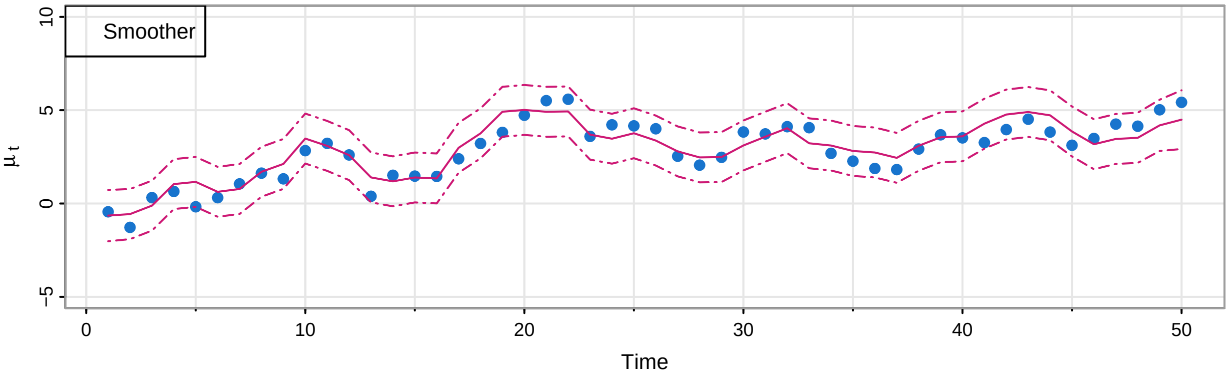
```
library(astsa)
mu0 = 0; sigma0 = 1; phi = 1; sQ = 1; sR = 1
ks = Ksmooth(y, A = 1, mu0, sigma0, phi, sQ, sR)
```

Plot the results

```r
par(mfrow = c(3, 1), mar = c(3.5, 3.5, 0.5, 0.5), mgp = c(2, 1, 0))

tsplot(mu[-1], type = 'p', col = 4, pch = 19,
       ylab = expression(mu[~t]), main = "", ylim = c(-5, 10))
lines(ks$Xp, col = 6)
lines(ks$Xp + 2 * sqrt(ks$Pp), lty = 6, col = 6)
lines(ks$Xp - 2 * sqrt(ks$Pp), lty = 6, col = 6)
legend("topleft", legend = "Prediction")

tsplot(mu[-1], type = 'p', col = 4, pch = 19,
       ylab = expression(mu[~t]), main = "", ylim = c(-5, 10))
lines(ks$Xf, col = 6)
lines(ks$Xf + 2 * sqrt(ks$Pf), lty = 6, col = 6)
lines(ks$Xf - 2 * sqrt(ks$Pf), lty = 6, col = 6)
legend("topleft", legend = "Filter")

tsplot(mu[-1], type = 'p', col = 4, pch = 19,
       ylab = expression(mu[~t]), main = "", ylim = c(-5, 10))
lines(ks$Xs, col = 6)
lines(ks$Xs + 2 * sqrt(ks$Ps), lty = 6, col = 6)
lines(ks$Xs - 2 * sqrt(ks$Ps), lty = 6, col = 6)
legend("topleft", legend = "Smoother")
```







Initial value info

```
mu[1]; ks$X0n; sqrt(ks$P0n)
```

```
## [1] -0.6264538
```

```
##             [,1]
## [1,] -0.3241541
```

```
##           [,1]
## [1,] 0.7861514
```

References

Shumway, Robert H, and David S Stoffer. 2017. *Time Series Analysis and Its Applications*. 4th ed. Springer.