Exercise 2

(1) Simulate and plot 10 timesteps of the following model,

$$y_t = 0.1y_{t-1} + 0.9y_{t-2} + 0.5w_t, \quad y_1 = 0, \quad w_t \sim \mathcal{N}(0,1) \text{ are iid.}$$

(2) Write the model above in state space form. That is,

$$\begin{aligned} Y_{t+1} &= AY_t + HV_t, \\ y_t &= BY_t \end{aligned}$$

for some $Y_t \in \mathbb{R}^3, A \in \mathbb{R}^{3\times 3}, H \in \mathbb{R}^{3\times 1}, B \in \mathbb{R}^{1\times 3}$ and $V_t \in \mathbb{R}$.

Solution. There might be multiple solutions here, but a natural one is,

$$\begin{aligned} Y_{t+1} &= \begin{pmatrix} 0.1 & 0.9 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} Y_t + \begin{pmatrix} 0.5 \\ 0 \\ 0 \end{pmatrix} V_t, \\ y_t &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} Y_t. \end{aligned}$$

(3) Implement the Kalman filter for this model and compute the conditional distribution of y_t given $y_1, \ldots, y_{10}$ for $t = 11, 12, \ldots, 20$. (That is, the density of a multivariate Gaussian distribution with mean,

$$E[y_t | y_1, \ldots, y_{10}],$$

and variance,

$$E[(y_t - E[y_t|y_1, \ldots, y_{10}])^2 | y_1, \ldots, y_{10}].)$$

Solution. You can check out the file "time-series-example.Rmd" under "Learning materials". Here, this (among other things) is done for a different state space model.

(4) Plot,

$$E[y_t | y_1, \ldots, y_{10}],$$

for $t = 11, \ldots, 20$, with approximate 95%-confidence intervals. How are the confidence intervals for t close to 10? What happens when t becomes large?

Hint. Approximate 95% confidence intervals can be computed as,

$$E[y_t|y_1, \ldots, y_{10}] \pm 2E[(y_t - E[y_t|y_1, \ldots, y_{10}])^2 | y_1, \ldots, y_{10}].$$

Solution. You can check out the file "time-series-example.Rmd" under "Learning materials". Here, this (among other things) is done for a different state space model.

(5) Assume now that the observations are given as,

$$z_t = y_t + v_t,$$

where $v_t \sim \mathcal{N}(0,1)$. Plot z_t for $t = 1, \ldots, 10$. Compute the conditional distribution of y_t given $z_1, \ldots, z_{10}$, for $t = 11, \ldots, 20$, and plot it with the observations. Include 95% confidence intervals.

Solution. You can check out the file "time-series-example.Rmd" under "Learning materials". Here, this (among other things) is done for a different state space model.

(6) Determine whether the following difference equation has a stationary and/or causal solution,

$$(1 - 0.6B + 0.05B^2)X_t = W_t.$$

Solution. We have that $1 - 0.6z + 0.05z^2 = 0.05(x - 10)(x - 2)$. Therefore, all roots are outsize the unit circle, and we have a stationary and causal solution.

(7) Follow the instructions on the next page, and try to estimate the parameters of the following $AR(1)$ model with observations noise:

$$X_{t+1} = \alpha X_t + \sigma_1 v_t$$
$$Y_t = X_t + \sigma_2 w_t,$$

where $v_t, w_t \sim \mathcal{N}(0, 1)$ and are iid.

Do this as follows: simulate Y_t for $t = 1, \ldots, 20$, where $\alpha, \sigma_1, \sigma_2 = 0.5$. Implement the likelihood computation, and use the true parameter values as the initial guess (this is the "par" argument in optim).

Solution. You can check out the file "time-series-example.Rmd" under "Learning materials". Here, this (among other things) is done for a different state space model.

State space models, Kalman filter, and computing the likelihood of $Y_1, \ldots, Y_T$

A Gaussian and linear state space model is given as follows,

$$X_{t+1} = AX_t + HV_t,$$
$$Y_t = BX_t + DW_t,$$

where $X_t, X_1 \in \mathbb{R}^N$ and $Y_t \in \mathbb{R}^M$ has are zero mean multivariate Gaussian, and, $H \in \mathbb{R}^{N\times K}, D \in \mathbb{R}^{M\times L}$, for integers N, M, K, L. Furthermore, $W_t \sim \mathcal{N}(0, I)$ is L-dimensional multivariate Gaussian, and $V_t \sim \mathcal{N}(0, I)$ is K-dimensional multivariate Gaussian, with W_t, V_t independent for each t.

We use the short notation, $X_{k|m} \in \mathbb{R}^N$ as the best linear predictor of X_k using $Y_1, \ldots, Y_m$. In the Gaussian case, this is just $E[X_k|Y_1, \ldots, Y_m]$.

Recall that the best linear predictor of X in terms of Y can be found using the best-linear-predictor-condition: for some $M \in \mathbb{R}^{N\times M}$, we must have,

$$X = MY \iff E[(X - MY)Y^T] = 0,$$

where the expectiation acts compenentwise on the the $R^{N\times M}$ entries of $(X - MY)Y^T$. We need to solve this equation for some matrix M. And it turns out that M is given by,

$$M = E[XY^T](E[YY^T])^{-1},$$

(where $(E[YY^T])^{-1}$ is the psuedo inverse of $E[YY^T]$).

We define the k'th innovation,

$$I_k := Y_k - Y_{k|k-1},$$

and we also introduce use the simplified notation,

$$S_{k|m} = E[(X_k - X_{k|m})(X_k - X_{k|m})^T|Y_1, \ldots, Y_m].$$

Then we can compute iteratively I_t and $C_t := E[I_t I_t^T]$ using the Kalman filter.

Algorithm 1 Kalman filter

(Initial condition, $X \sim \mathcal{N}(X_1, S_1)$)
$X_{1|0} \leftarrow X_1$
$S_{1|0} \leftarrow S_1$
for $t = 1, \ldots, T$ **do**
  Compute innovation, I_t
  $I_t = Y_t - BX_{t|t-1}$
  Compute projection matrix M
  $M = S_{t|t-1}B^T C_t^{-1}, \quad C_t = BS_{t|t-1}B^T + DD^T$
  Condition on innovation/observation
  $X_{t|t} \leftarrow X_{t|t-1} + MI_t$
  Update $X_{t|t}, S_{t|t-1}$
  $X_{t+1|t} \leftarrow AX_{t|t}$
  $S_{t+1|t} \leftarrow A(I - MB)S_{t|t-1}A^T + HH^T$
end for.

We can use the computed innovations and their variance to compute the likelihood in this case. Since the innovations are Gaussian and uncorrelated, they are independent, and we have,

$$L(\theta) = C \prod_{t=1}^{T} \det(C_t)^{-1/2} e^{-\frac{1}{2} I_t^T C_t^{-1} I_t},$$

for some constant C independent of θ.

In practice, we need to optimize the logarithm of the likelihood, since this is computationally easier.

In R, you can minimize any function, "obj", that you have defined, by using optim as follows,

```
opt <- optim(par=initial_param, fn=obj, method="BFGS",
           control=list(maxit=300, reltol=1e-8)).
```

use ?optim in the console in Rstudio to get more info on the arguments of optim, and what is returned.

Hint. A skeleton of the likelihood implementation in R is,

```
n <- 3

# objective function to be minimized
obj <- function(param) {
  # param is a vector of length 3 with the parameters in the model
  alpha <- param[1]
  sigma1 <- param[2]
  sigma2 <- param[3]

  # define the matrix A
  A <- matrix(0,nrow=n,ncol=n)
  A[1,1] = ...
  A[2,1] = ...

  # observation matrix
  B <- matrix(0,nrow=1,ncol=n)
  B[1,1] = ...

  # system noise
  HH <- ...

  # observation noise
  DD <- ...

  # compute likelihood

  # initial condition
  x <- matrix(0,nrow=n,ncol=1)
  S <- diag(n) # just use something here :)

  # log likelihood
  ll <- 0.
  for (t in 1:tt) {
    # compute innovation

    # innovation variance

    # compute M matrix
```

```
    # update x, S

    # add to ll
    ll <- ll + ...

  }

  return(ll)
}
```