

DLM con INPC

2023-07-01

Este modelo se parece mucho al último que te mandé, \mathbf{G}_t , \mathbf{W}_t y V se estiman por máxima verosimilitud utilizando datos del 2001 T1 al 2011 T4 con el paquete MARSS (usa un método que se llama BFGS). Se define la \mathbf{G} de tal forma que la estacionalidad se reacomoda:

```
G <- matrix(list(0), 7, 7)
G[1,1] <- 'G.1'
G[2,2] <- 'G.2'
G[3,3] <- 'G.3'
G[4,5] <- 1
G[5,6] <- 1
G[6,7] <- -1
G[7,4] <- 1
G
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,] "G.1" 0    0    0    0    0    0
## [2,] 0     "G.2" 0    0    0    0    0
## [3,] 0     0     "G.3" 0    0    0    0
## [4,] 0     0     0     0    1    0    0
## [5,] 0     0     0     0    0    1    0
## [6,] 0     0     0     0    0    0    1
## [7,] 0     0     0     1    0    0    0
```

Aquí G.1 corresponde al intercepto, G.2 al PIB y G.3 al INPC. Por otro lado, \mathbf{W} tiene la siguiente forma:

```
W <- matrix(list(0), 7, 7)
W[1,1] <- 'W.1'
W[2,2] <- 'W.2'
W[3,3] <- 'W.3'
W[4,4] <- 'W.4'
W
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,] "W.1" 0    0    0    0    0    0
## [2,] 0     "W.2" 0    0    0    0    0
## [3,] 0     0     "W.3" 0    0    0    0
## [4,] 0     0     0     "W.4" 0    0    0
## [5,] 0     0     0     0    0    0    0
## [6,] 0     0     0     0    0    0    0
## [7,] 0     0     0     0    0    0    0
```

W.1 está relacionado con el intercepto, W.2 con el PIB, W.3 con el INPC, y W.4 con el factor estacional en turno. Con esta W, se permite el cambio en el intercepto, el parámetro del PIB, el parámetro del INPC y del factor estacional del trimestre en turno.

Con esto, las variables explicativas se tienen que acomodar así:

```
datos_pib <- read_rds('cache/variables/pib.rds') %>%
  filter(fecha > 2011.75) %>%
  mutate(pib = log(pib))

datos_inpc <- read_rds('cache/variables/last/inpc_last.rds') %>%
  filter(fecha > 2011.75) %>%
  mutate(inpc = log(inpc))

datos_efectivo <- read_rds('cache/variables/last/efectivo_last.rds') %>%
  filter(fecha > 2011.75) %>%
  mutate(efectivo = log(efectivo))

datos_estacionalidad <- datos_efectivo %>%
  mutate(Q1 = 1) %>%
  mutate(Q2=0) %>%
  mutate(Q3=0) %>%
  mutate(Q4=0)

datos_F <- datos_pib %>%
  left_join(datos_estacionalidad) %>%
  left_join(datos_inpc) %>%
  mutate(intercept = 1) %>%
  dplyr::select(9,2, 8, 4:7)
```

```
## Joining, by = "fecha"
## Joining, by = "fecha"
```

datos_F

##	intercept	pib	inpc	Q1	Q2	Q3	Q4
## 1	1	9.656292	1.3164082	1	0	0	0
## 2	1	9.682059	1.4678743	1	0	0	0
## 3	1	9.677390	1.5623463	1	0	0	0
## 4	1	9.719496	1.2725656	1	0	0	0
## 5	1	9.662675	1.4469190	1	0	0	0
## 6	1	9.702708	1.4085450	1	0	0	0
## 7	1	9.691909	1.2208299	1	0	0	0
## 8	1	9.731571	1.3787661	1	0	0	0
## 9	1	9.690423	1.3244190	1	0	0	0
## 10	1	9.725763	1.3217558	1	0	0	0
## 11	1	9.720159	1.4398351	1	0	0	0
## 12	1	9.764730	1.4060970	1	0	0	0
## 13	1	9.723775	1.1442228	1	0	0	0
## 14	1	9.756217	1.0543120	1	0	0	0
## 15	1	9.760290	0.9242589	1	0	0	0
## 16	1	9.790676	0.7561220	1	0	0	0
## 17	1	9.750708	0.9555114	1	0	0	0
## 18	1	9.785893	0.9321641	1	0	0	0
## 19	1	9.777110	1.0885620	1	0	0	0
## 20	1	9.820965	1.2119410	1	0	0	0
## 21	1	9.784568	1.6770966	1	0	0	0

```
## 22      1 9.802828 1.8421357 1 0 0 0
## 23      1 9.792596 1.8484548 1 0 0 0
## 24      1 9.838731 1.9125011 1 0 0 0
## 25      1 9.799163 1.6174061 1 0 0 0
## 26      1 9.834692 1.5368672 1 0 0 0
## 27      1 9.820422 1.6134299 1 0 0 0
## 28      1 9.851422 1.5748465 1 0 0 0
## 29      1 9.811425 1.3862944 1 0 0 0
## 30      1 9.823588 1.3737156 1 0 0 0
## 31      1 9.818550 1.0986123 1 0 0 0
## 32      1 9.844595 1.0402767 1 0 0 0
## 33      1 9.801991 1.1786550 1 0 0 0
## 34      1 9.617721 1.2029723 1 0 0 0
## 35      1 9.731482 1.3887912 1 0 0 0
## 36      1 9.803066 1.1474025 1 0 0 0
## 37      1 9.766174 1.5411591 1 0 0 0
## 38      1 9.796324 1.7715568 1 0 0 0
## 39      1 9.773587 1.7917595 1 0 0 0
## 40      1 9.813187 1.9960599 1 0 0 0
## 41      1 9.784782 2.0082140 1 0 0 0
## 42      1 9.820242 2.0781908 1 0 0 0
## 43      1 9.816343 2.1633230 1 0 0 0
## 44      1 9.848245 2.0566846 1 0 0 0
```

En el caso de V_t , estoy considerando que la varianza es una constante desconocida y tiene una distribución a priori, que es $\phi \sim G[\frac{n_0}{2}, \frac{n_0 S_0}{2}]$ con $\phi = V^{-1}$. En este caso, el estimador puntual de V es S_0 . Lo que hice fue estimar V con los datos de 2001 a 2011 usando MARSS, esa estimación la utilizo como S_0 para lo demás, y en cada periodo se actualiza S_t . Como utilicé 44 observaciones para estimar esto, entonces n_0 es 44.

La distribución posterior de los parámetros en 2011 T4 ($\theta_{2011T4}|D_{2011T4}$), la tomo como la distribución inicial ($\theta_0|D_0$) para las observaciones a partir de 2012 T1, que es en donde mido el desempeño el modelo.

Leo las estimaciones y los valores iniciales que obtuve en el periodo 2001 T1 a 2011 T4:

```
source('src/funciones/funciones_dlm.R')

##
## Attaching package: 'LaplacesDemon'

## The following object is masked from 'package:purrr':
##
##     partial

C0 <- read_rds('cache/outputs_modelos/inpc/C0.rds')
m0 <- read_rds('cache/outputs_modelos/inpc/m0.rds')
G <- read_rds('cache/outputs_modelos/inpc/G.rds')
W <- read_rds('cache/outputs_modelos/inpc/W.rds')
S0 <- as.numeric(read_rds('cache/outputs_modelos/inpc/V.rds'))
```

Como el paquete MARSS no incluye el caso en el que la varianza es una constante desconocida que se reestima en cada periodo ni se pueden hacer intervenciones fácilmente, hice mi función para poder modelar eso en el filtro de Kalman:

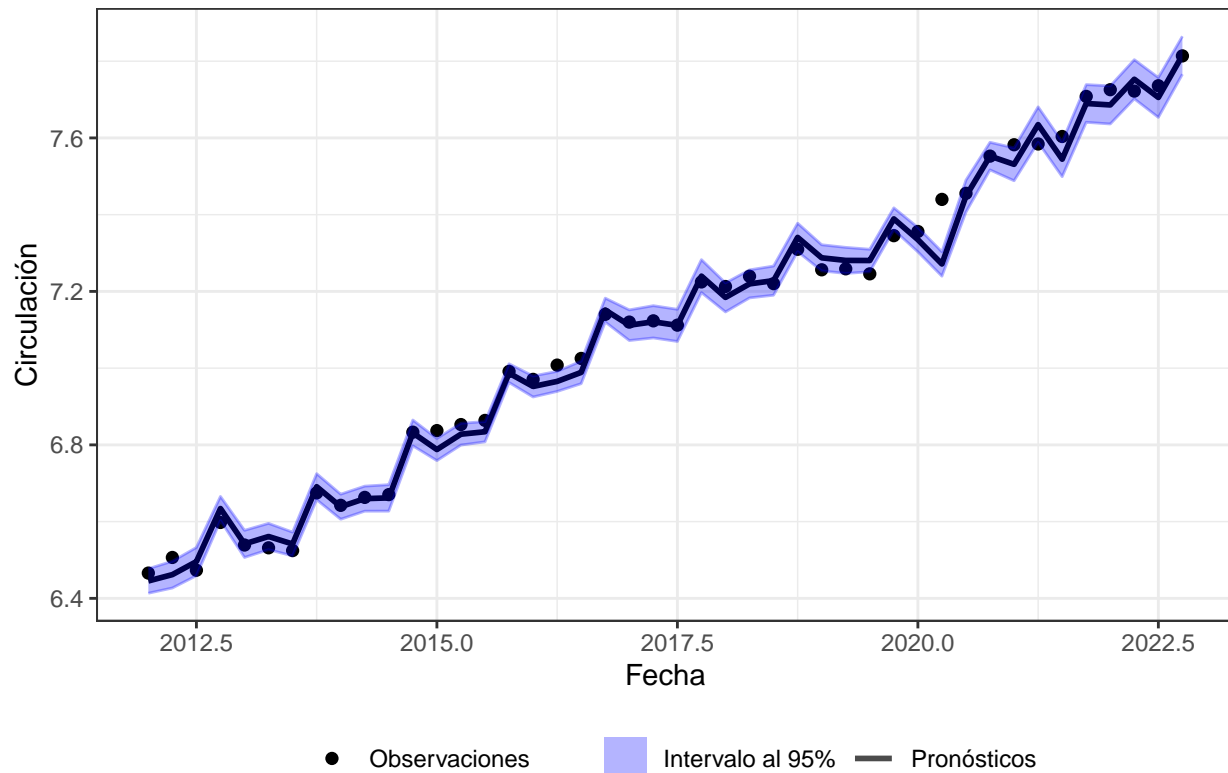
```
dmlm <- actualizacion_dlm_V_desc(y = datos_efectivo$efectivo, variables_F = datos_F,
                                m0 = m0, C0 = C0, G = G, W = W, S0 = S0, n0 = 44,
                                lista_interv = list())
```

Ahora grafico los pronósticos del modelo, los intervalos al 95% y las observaciones reales para ver el ajuste

```
df_ajuste <- data.frame("fecha" = datos_efectivo %>% dplyr::select(fecha),
                        "y_real" = datos_efectivo$efectivo,
                        "y_pronostico" = dmlm$ft %>% unlist(),
                        "CI_inf" = dmlm$CI_inf %>% unlist(),
                        "CI_sup" = dmlm$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))

ggplot(data = df_ajuste, aes(x = fecha)) +
  geom_point(aes(y = y_real, shape = "Observaciones"), size = 2) +
  geom_line(aes(y = y_pronostico, color = 'Pronósticos'), size = 1) +
  geom_line(aes(y = CI_inf), color = "blue", alpha = 0.3) +
  geom_line(aes(y = CI_sup), color = "blue", alpha = 0.3) +
  geom_ribbon(aes(ymax = CI_sup, ymin = CI_inf, fill = 'Intervalo al 95%'), alpha = 0.3) +
  theme_bw() +
  scale_colour_manual(
    name = "", values = c("Intervalo al 95%" = "transparent",
                          "Pronósticos" = "black")) +
  scale_fill_manual(
    name = "", values = c("Intervalo al 95%" = "blue",
                          "Pronósticos" = "transparent")) +
  theme(legend.position = "bottom") +
  labs(shape = "") +
  ylab('Circulación') +
  xlab('Fecha') +
  labs(title = 'Pronósticos e intervalos a un paso')
```

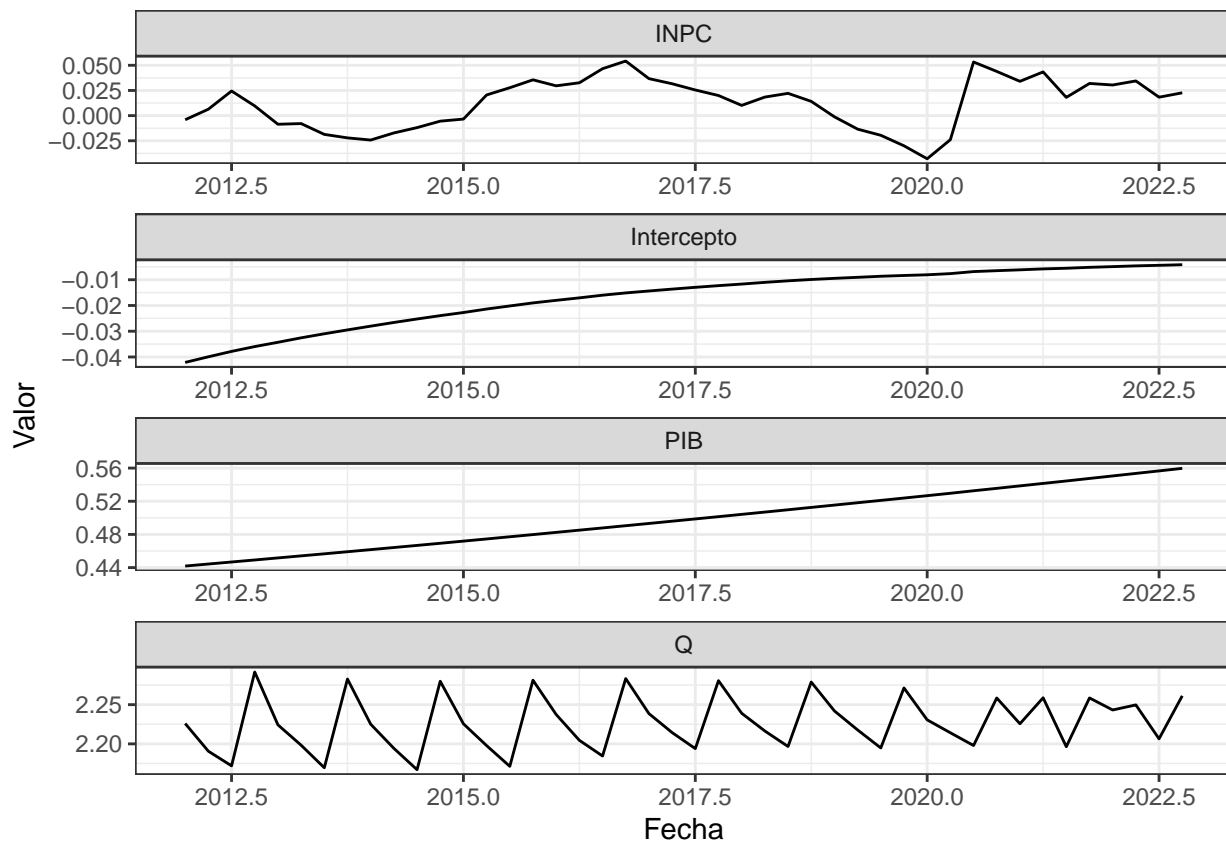
Pronósticos e intervalos a un paso



Ajusta muy bien, el único punto en donde no ajusta bien es al inicio de la pandemia. Esto lo puedo solucionar fácilmente con una intervención (viene al final del pdf). La evolución de los parámetros se ve así:

```
df_params <- data.frame(reduce(dlm$at, cbind) %>% t(), fecha = df_ajuste$fecha) %>%
  rename(Intercepto = X1, PIB = X2, INPC = X3, Q = X4) %>%
  pivot_longer(names_to = "parametro", values_to = "valor",
    cols = c(Intercepto, PIB, INPC, Q))

ggplot(df_params, aes(x=fecha, y = valor)) +
  geom_line() +
  facet_wrap(~parametro, nrow = 4, scales = "free") +
  theme_bw() +
  ylab("Valor") +
  xlab("Fecha")
```

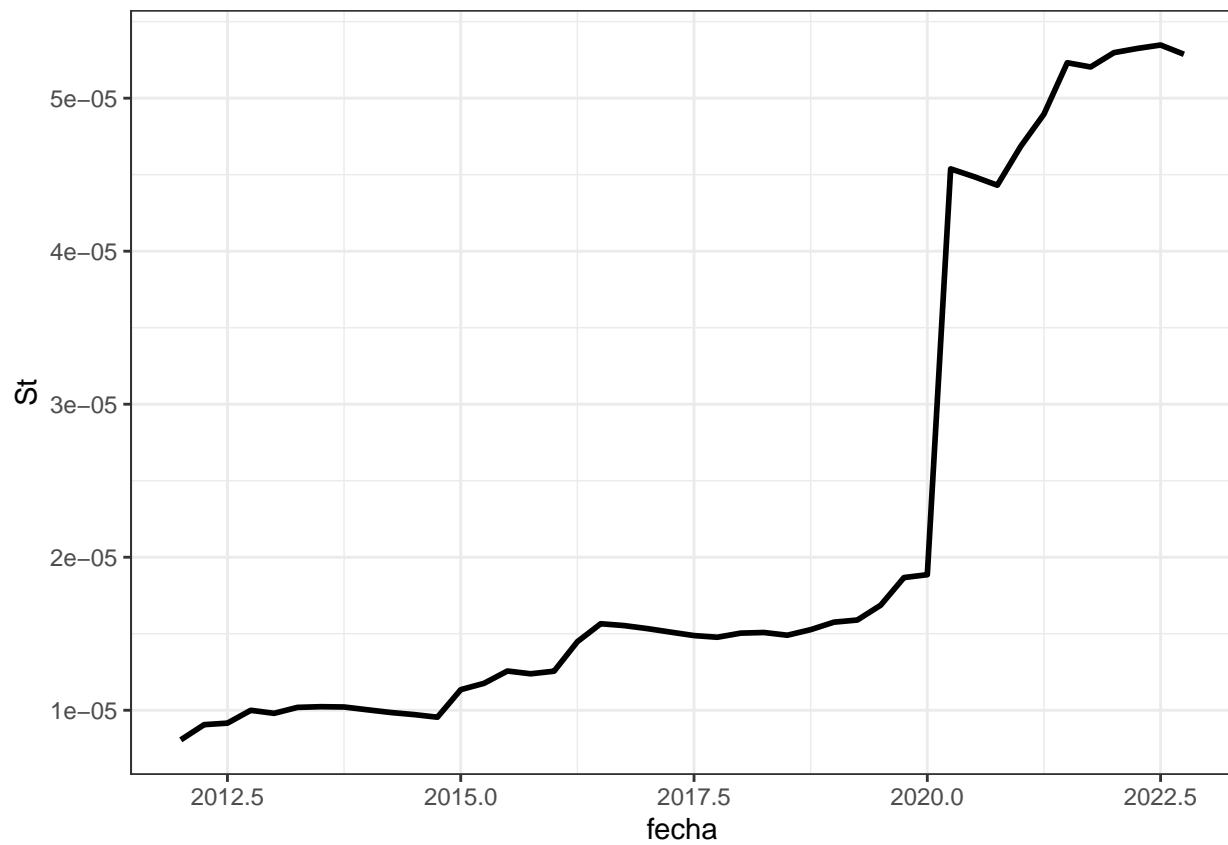


El PIB es positivo, eso tiene mucho sentido. El INPC la mayoría del tiempo es positivo, pero a veces es negativo, no sé bien cómo interpretar esto, cuando estuve leyendo literatura sobre modelos de demanda de dinero en general los estudios indican que la inflación debe ser negativa (porque al haber inflación alta las personas se cubren con activos reales y no con dinero), aunque hay varios autores que indican que puede ser positiva debido a que cuando se espera una inflación alta, las personas tienen más efectivo para hacer frente a la subida de los precios. La estacionalidad tiene picos en Q4, aunque al final del periodo ya no es tan notable esto (tiene mucho sentido por lo que se ve en los datos).

Igual podemos ver cómo evoluciona la estimación de la varianza en cada periodo. Se ve que hay un incremento muy grande al inicio de la pandemia.

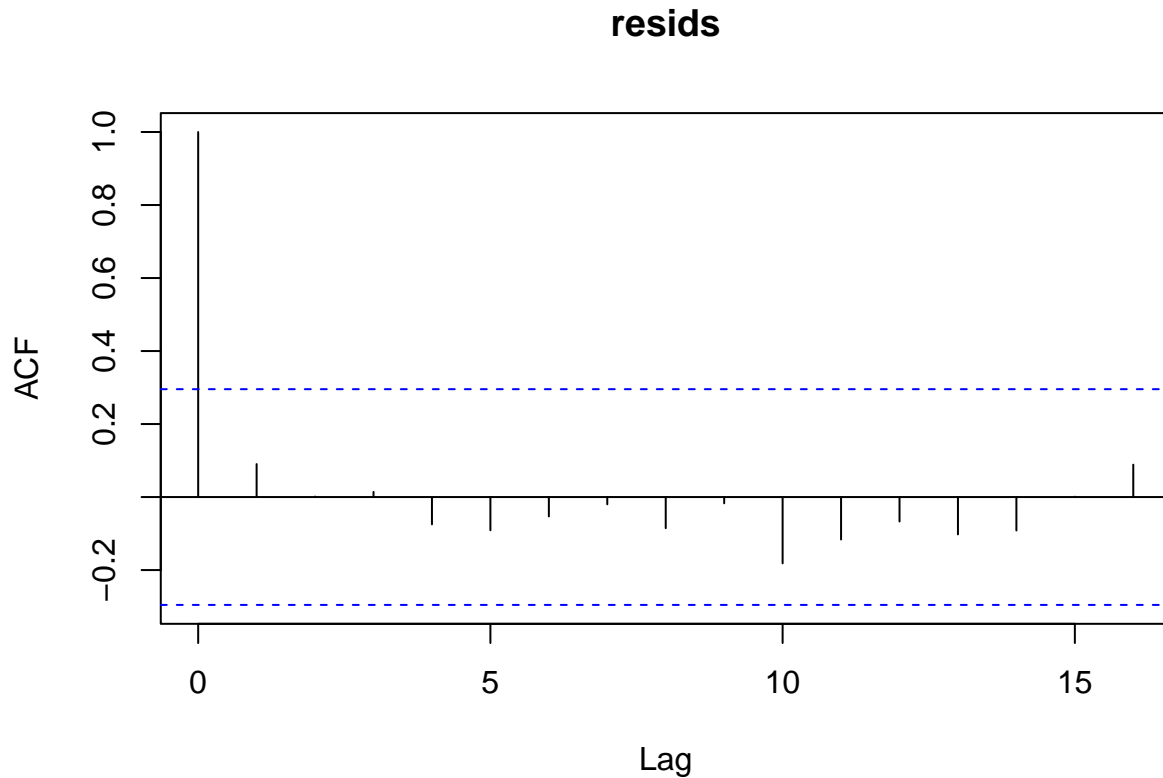
```
df_St <- data.frame(dlm$St %>% unlist(), fecha = df_ajuste$fecha) %>%
  dplyr::rename(St = 1)

ggplot(df_St, aes(x=fecha, y = St)) +
  geom_line(size = 1) +
  theme_bw() +
  ylab("St")
```



La autocorrelación de los residuos se ve muy bien.

```
resids <- df_ajuste %>%  
  mutate(resids = y_real - y_pronostico) %>%  
  dplyr::select(resids)  
  
acf(resids)
```



Por último, el error cuadrático medio es muy bajo:

```
ecm_1 <- mean(resids$resids^2)
ecm_1
```

```
## [1] 0.001402401
```

Puedo graficar los ajustes a más de un paso. Igual para esto definí una función, solo es necesario pasarle las variables explicativas, el número de periodos futuros que se desea pronosticar y el dlm obtenido anteriormente

K = 2

```
k <- 2
prons2 <- pronosticos_k_pasos(datos_F, k = k, modelo = dlm)

df_prons2 <- data.frame("fecha" = datos_efectivo$fecha[k:nrow(datos_F)],
                        "y_real" = datos_efectivo$efectivo[k:nrow(datos_F)],
                        "y_pronostico" = prons2$ft_k %>% unlist(),
                        "CI_inf" = prons2$CI_inf %>% unlist(),
                        "CI_sup" = prons2$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))

ggplot(data = df_prons2, aes(x = fecha)) +
  geom_point(aes(y = y_real, shape = "Observaciones"), size = 2) +
  geom_line(aes(y = y_pronostico, color = 'Pronósticos'), size = 1) +
  geom_line(aes(y = CI_inf), color = "blue", alpha = 0.3) +
  geom_line(aes(y = CI_sup), color = "blue", alpha = 0.3) +
  geom_ribbon(aes(ymax = CI_sup, ymin = CI_inf, fill = 'Intervalo al 95%'), alpha = 0.3) +
  theme_bw() +
```



```

scale_colour_manual(
  name = "", values = c("Intervalo al 95%" = "transparent",
                        "Pronósticos" = "black")) +

scale_fill_manual(
  name = "", values = c("Intervalo al 95%" = "blue",
                        "Pronósticos" = "transparent")) +

theme(legend.position = "bottom") +
labs(shape = "", title = 'Pronósticos a 2 pasos') +
ylab('Circulación') +
xlab('Fecha')

```



```

resids2 <- df_prons2 %>%
  mutate(resids = y_real - y_pronostico) %>%
  dplyr::select(resids)

```

```
ecm_2 <- mean(resids2$resids^2)
```

```
ecm_2
```

```
## [1] 0.002231555
```

Lo mismo para k=3

```

k <- 3
prons3 <- pronosticos_k_pasos(datos_F, k = k, modelo = dlm)

```

```
df_prons3 <- data.frame("fecha" = datos_efectivo$fecha[k:nrow(datos_F)],
                        "y_real" = datos_efectivo$efectivo[k:nrow(datos_F)],
                        "y_pronostico" = prons3$ft_k %>% unlist(),
                        "CI_inf" = prons3$CI_inf %>% unlist(),
                        "CI_sup" = prons3$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))

ggplot(data = df_prons3, aes(x = fecha)) +
  geom_point(aes(y = y_real, shape = "Observaciones"), size = 2) +
  geom_line(aes(y = y_pronostico, color = 'Pronósticos'), size = 1) +
  geom_line(aes(y = CI_inf, color = "blue", alpha = 0.3) +
  geom_line(aes(y = CI_sup, color = "blue", alpha = 0.3) +
  geom_ribbon(aes(ymin = CI_inf, ymax = CI_sup, fill = 'Intervalo al 95%'), alpha = 0.3) +
  theme_bw() +
  scale_colour_manual(
    name = "", values = c("Intervalo al 95%" = "transparent",
                          "Pronósticos" = "black")) +
  scale_fill_manual(
    name = "", values = c("Intervalo al 95%" = "blue",
                          "Pronósticos" = "transparent")) +
  theme(legend.position = "bottom") +
  labs(shape = "", title = 'Pronósticos a 3 pasos') +
  ylab('Circulación') +
  xlab('Fecha')
```



```

resids3 <- df_prons3 %>%
  mutate(resids = y_real - y_pronostico) %>%
  dplyr::select(resids)

ecm_3 <- mean(resids3$resids^2)

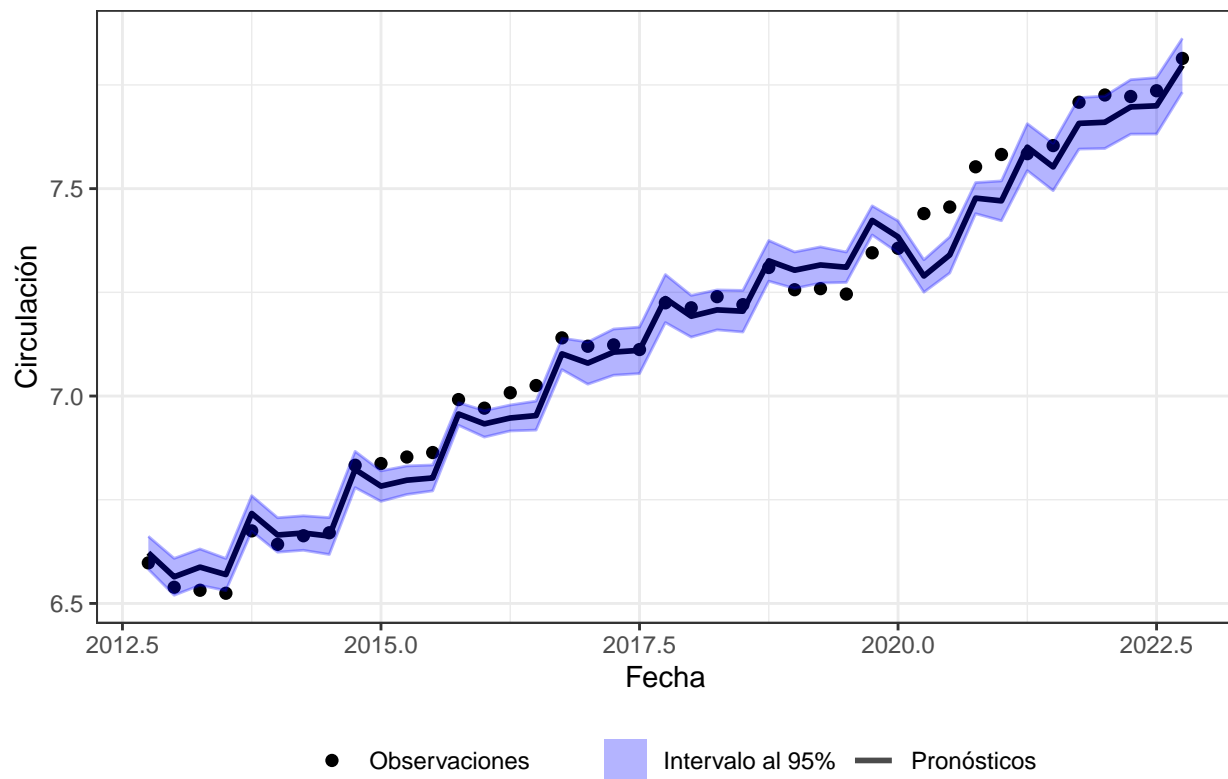
ecm_3

```

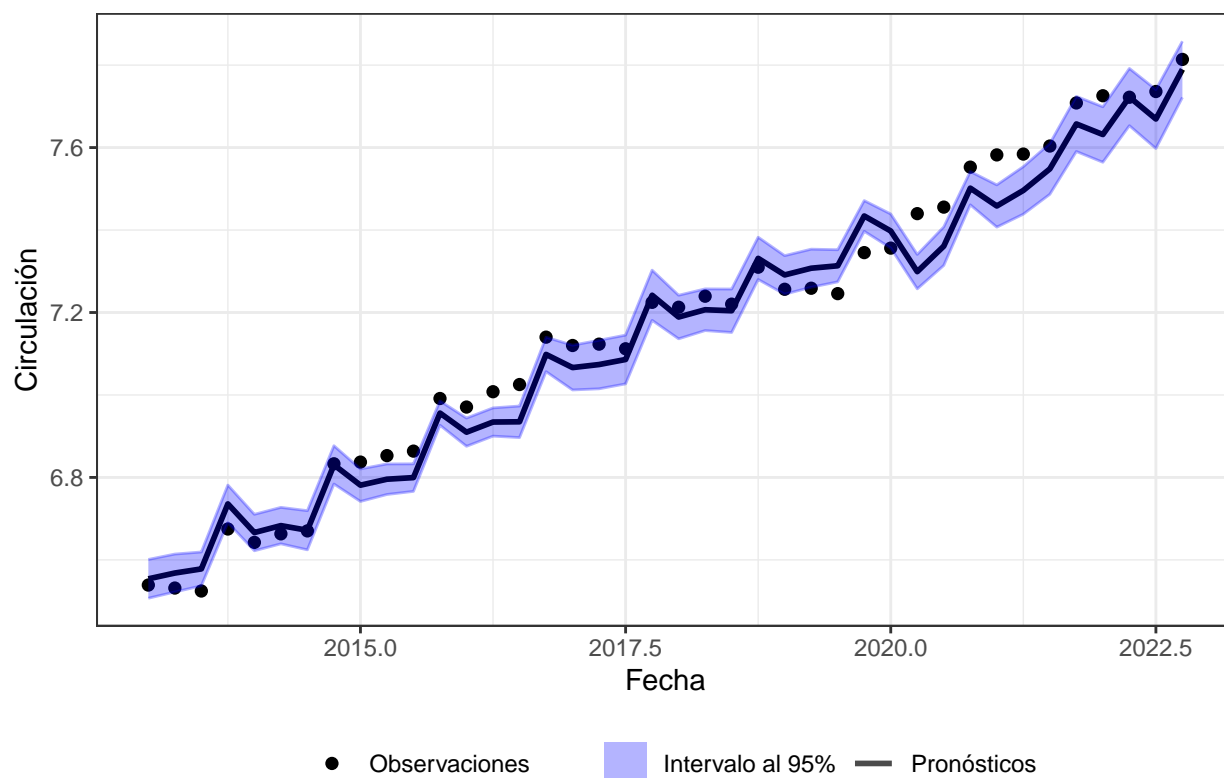
```
## [1] 0.002602316
```

Y lo mismo para cualquier número de periodos futuros.

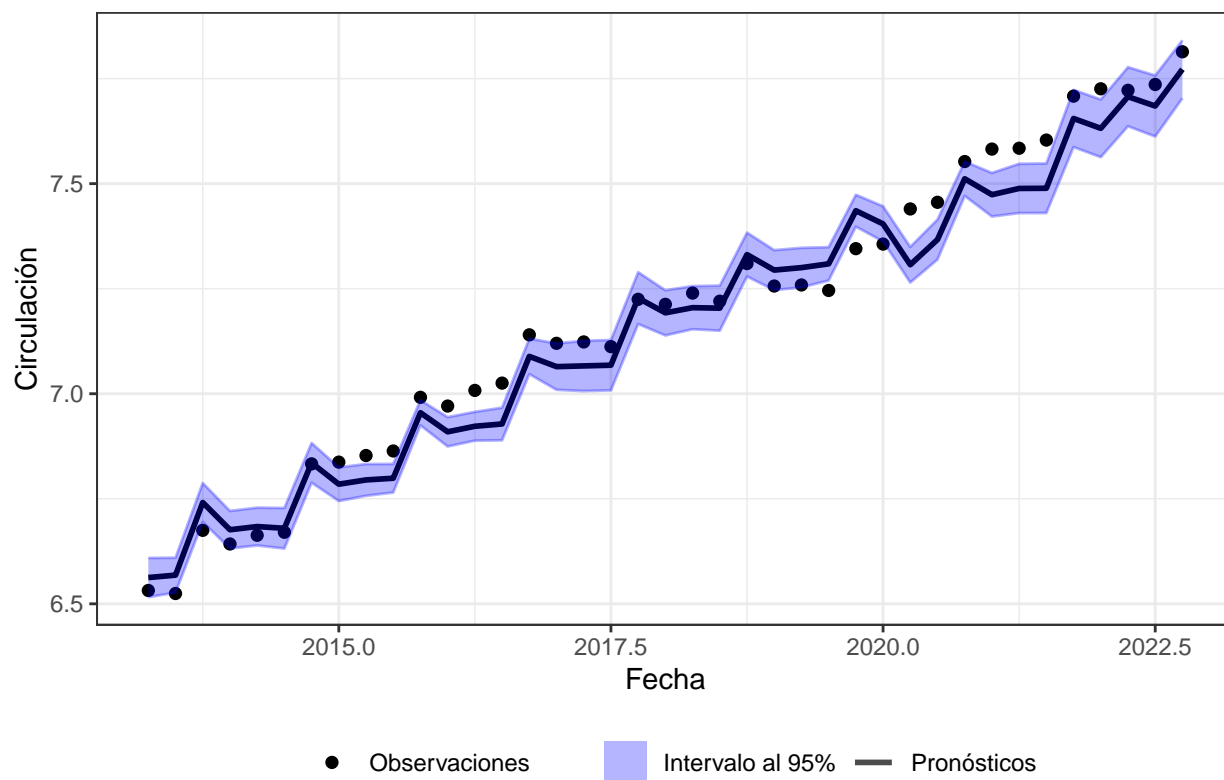
Pronósticos a 4 pasos



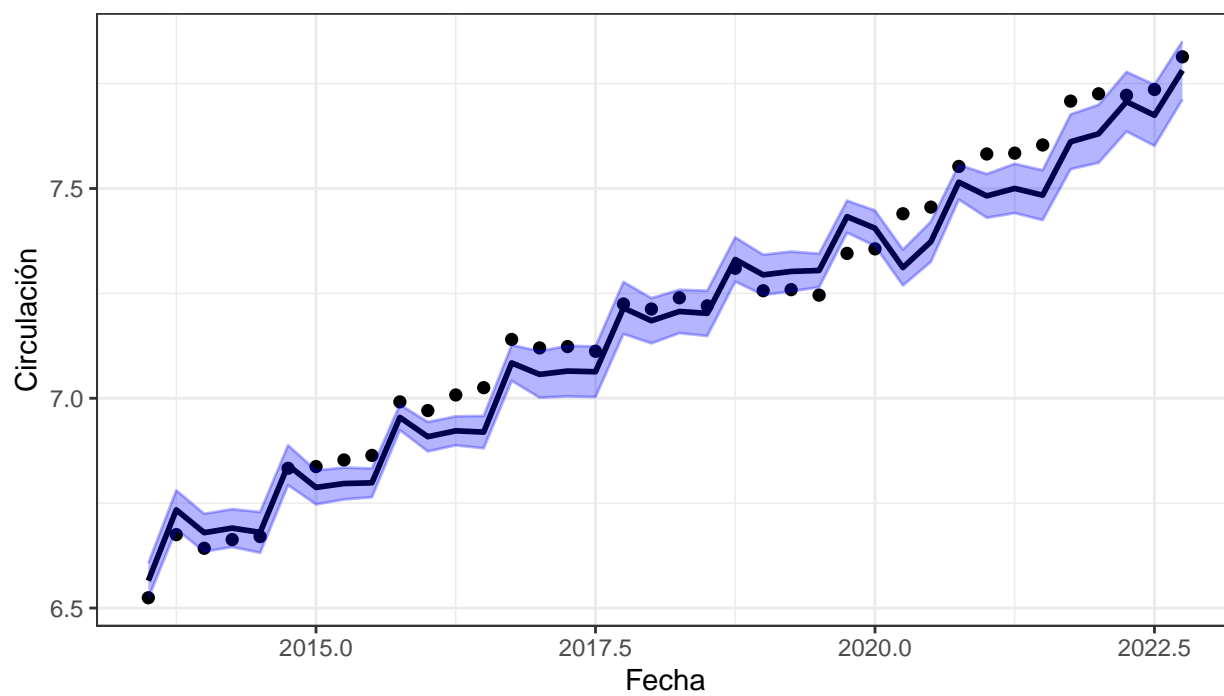
Pronósticos a 5 pasos



Pronósticos a 6 pasos

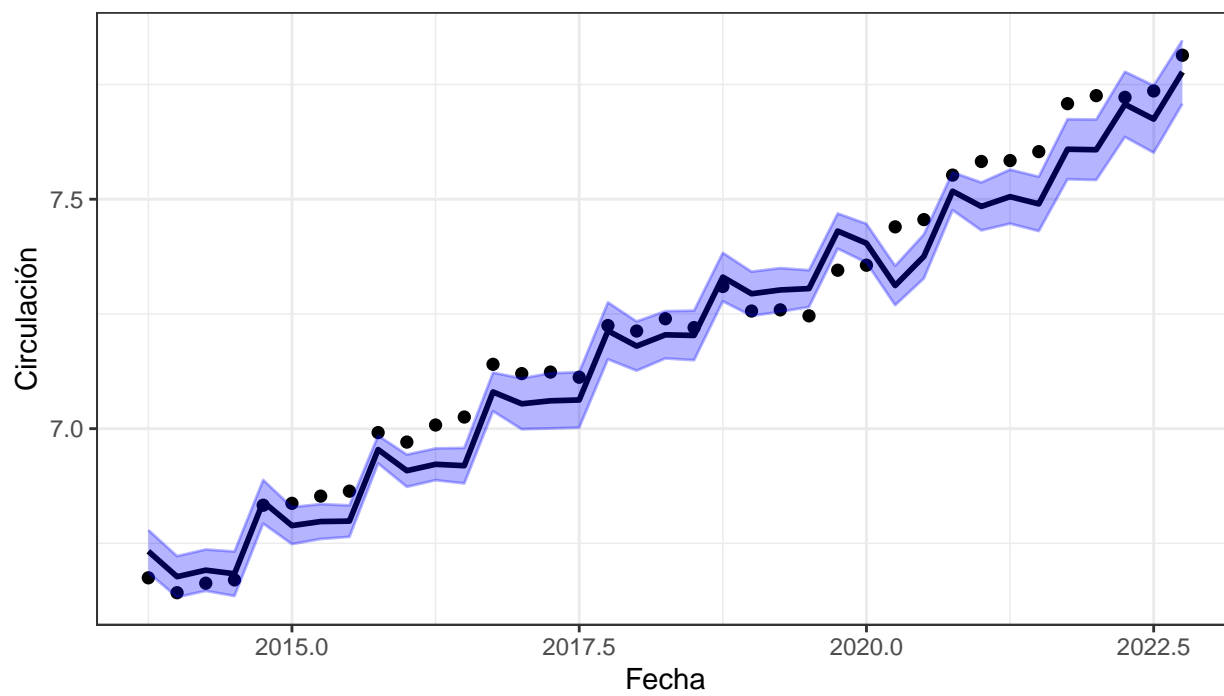


Pronósticos a 7 pasos



● Observaciones Intervalo al 95% — Pronósticos

Pronósticos a 8 pasos



● Observaciones Intervalo al 95% — Pronósticos

Intervención

Solo realizaré una intervención al inicio de la pandemia en 2020 T2, es el único punto en el que el modelo falla en los pronósticos a un paso. Este punto equivale a $t = 34$. \mathbf{a}_{34} y \mathbf{R}_{34} sin intervenir tienen los siguientes valores:

```
a_34 <-dlm$at[[34]]
R_34 <-dlm$Rt[[34]]
a_34
```

```
##           [,1]
## [1,] -0.007629454
## [2,]  0.529625712
## [3,] -0.023940013
## [4,]  2.214134534
## [5,]  2.186593984
## [6,]  2.256141019
## [7,]  2.236028803
```

```
R_34
```

```
##           [,1]           [,2]           [,3]           [,4]           [,5]
## [1,]  3.168271e-06 -9.103572e-09 -1.221518e-06 -9.602044e-07 -1.045443e-06
## [2,] -9.103572e-09  1.008294e-07 -3.415044e-07 -4.072090e-07 -4.320521e-07
## [3,] -1.221518e-06 -3.415044e-07  1.474815e-04 -3.210693e-05 -2.911377e-05
## [4,] -9.602044e-07 -4.072090e-07 -3.210693e-05  1.090831e-04  5.746143e-05
## [5,] -1.045443e-06 -4.320521e-07 -2.911377e-05  5.746143e-05  8.947897e-05
## [6,] -1.111013e-06 -4.545001e-07 -3.420829e-05  4.967325e-05  5.437397e-05
## [7,] -1.098229e-06 -4.403606e-07 -5.419758e-05  5.496149e-05  4.955892e-05
##           [,6]           [,7]
## [1,] -1.111013e-06 -1.098229e-06
## [2,] -4.545001e-07 -4.403606e-07
## [3,] -3.420829e-05 -5.419758e-05
## [4,]  4.967325e-05  5.496149e-05
## [5,]  5.437397e-05  4.955892e-05
## [6,]  8.497077e-05  5.602390e-05
## [7,]  5.602390e-05  9.254365e-05
```

Para la intervención subiré el coeficiente del intercepto en \mathbf{a}_{34} de -0.0076 a 0.1. En \mathbf{R}_{34} duplicaré el valor de la varianza del intercepto. En la función de dlm que definí puedo agregar una lista con los valores que quiero intervenir y el tiempo en el que los quiero intervenir. Nada más le tengo que decir que quiero intervenir $t = 34$ y pasarle los valores de \mathbf{a}_{34} y \mathbf{R}_{34} intervenidos.

```
a_34_int <- a_34
a_34_int[1,] <- 0.1
a_34_int
```

```
##           [,1]
## [1,]  0.10000000
## [2,]  0.52962571
## [3,] -0.02394001
## [4,]  2.21413453
```

```
## [5,] 2.18659398
## [6,] 2.25614102
## [7,] 2.23602880
```

```
R_34_int <- R_34
R_34_int[1,1] <- R_34[1,1]*2
R_34_int
```

```
##           [,1]           [,2]           [,3]           [,4]           [,5]
## [1,] 6.336542e-06 -9.103572e-09 -1.221518e-06 -9.602044e-07 -1.045443e-06
## [2,] -9.103572e-09 1.008294e-07 -3.415044e-07 -4.072090e-07 -4.320521e-07
## [3,] -1.221518e-06 -3.415044e-07 1.474815e-04 -3.210693e-05 -2.911377e-05
## [4,] -9.602044e-07 -4.072090e-07 -3.210693e-05 1.090831e-04 5.746143e-05
## [5,] -1.045443e-06 -4.320521e-07 -2.911377e-05 5.746143e-05 8.947897e-05
## [6,] -1.111013e-06 -4.545001e-07 -3.420829e-05 4.967325e-05 5.437397e-05
## [7,] -1.098229e-06 -4.403606e-07 -5.419758e-05 5.496149e-05 4.955892e-05
##           [,6]           [,7]
## [1,] -1.111013e-06 -1.098229e-06
## [2,] -4.545001e-07 -4.403606e-07
## [3,] -3.420829e-05 -5.419758e-05
## [4,] 4.967325e-05 5.496149e-05
## [5,] 5.437397e-05 4.955892e-05
## [6,] 8.497077e-05 5.602390e-05
## [7,] 5.602390e-05 9.254365e-05
```

```
list_interv <- list("t_int" = list(34),
                  "at_int" = list(a_34_int),
                  "Rt_int" = list(R_34_int))
```

A la función le paso la lista de intervención definida.

```
dmlm_interv <- actualizacion_dlm_V_desc(y = datos_efectivo$efectivo, variables_F = datos_F,
                                       m0 = m0, C0 = C0, G = G, W = W, S0 = S0, n0 = 44,
                                       lista_interv = list_interv)
```

Ahora grafico los pronósticos del modelo, los intervalos al 95% y las observaciones reales para ver el ajuste

```
df_ajuste_int <- data.frame("fecha" = datos_efectivo %>% dplyr::select(fecha),
                           "y_real" = datos_efectivo$efectivo,
                           "y_pronostico" = dmlm_interv$ft %>% unlist(),
                           "CI_inf" = dmlm_interv$CI_inf %>% unlist(),
                           "CI_sup" = dmlm_interv$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))

ggplot(data = df_ajuste_int, aes(x = fecha)) +
  geom_point(aes(y = y_real, shape = "Observaciones"), size = 2) +
  geom_line(aes(y = y_pronostico, color = 'Pronósticos'), size = 1) +
  geom_line(aes(y = CI_inf, color = "blue", alpha = 0.3) +
  geom_line(aes(y = CI_sup, color = "blue", alpha = 0.3) +
  geom_ribbon(aes(ymax = CI_sup, ymin = CI_inf, fill = 'Intervalo al 95%'), alpha = 0.3) +
  theme_bw() +
```

```

scale_colour_manual(
  name = "", values = c("Intervalo al 95%" = "transparent",
                        "Pronósticos" = "black")) +

scale_fill_manual(
  name = "", values = c("Intervalo al 95%" = "blue",
                        "Pronósticos" = "transparent")) +

theme(legend.position = "bottom") +
labs(shape = "") +
ylab('Circulación') +
xlab('Fecha') +
labs(title = 'Pronósticos e intervalos a un paso, modelo intervenido')

```



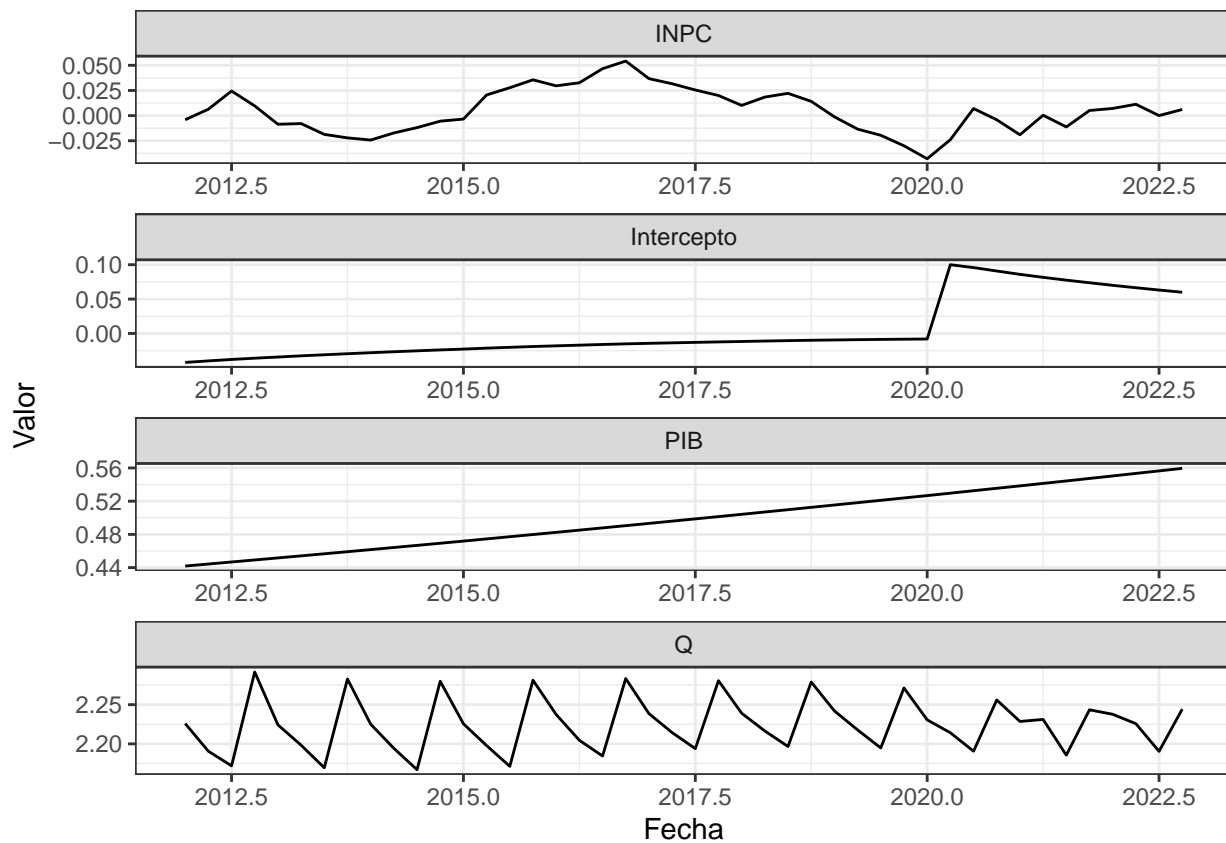
Mucho mejor! Ahora se puede ver el efecto de la intervención en los parámetros:

```

df_params_int <- data.frame(reduce(dlm_interv$at, cbind) %>% t(),
                             fecha = df_ajuste_int$fecha) %>%
  rename(Intercepto = X1, PIB = X2, INPC = X3, Q = X4) %>%
  pivot_longer(names_to = "parametro", values_to = "valor",
               cols = c(Intercepto, PIB, INPC, Q))

ggplot(df_params_int, aes(x=fecha, y = valor)) +
  geom_line() +
  facet_wrap(~parametro, nrow = 4, scales = "free") +
  theme_bw() +
  ylab("Valor") +
  xlab("Fecha")

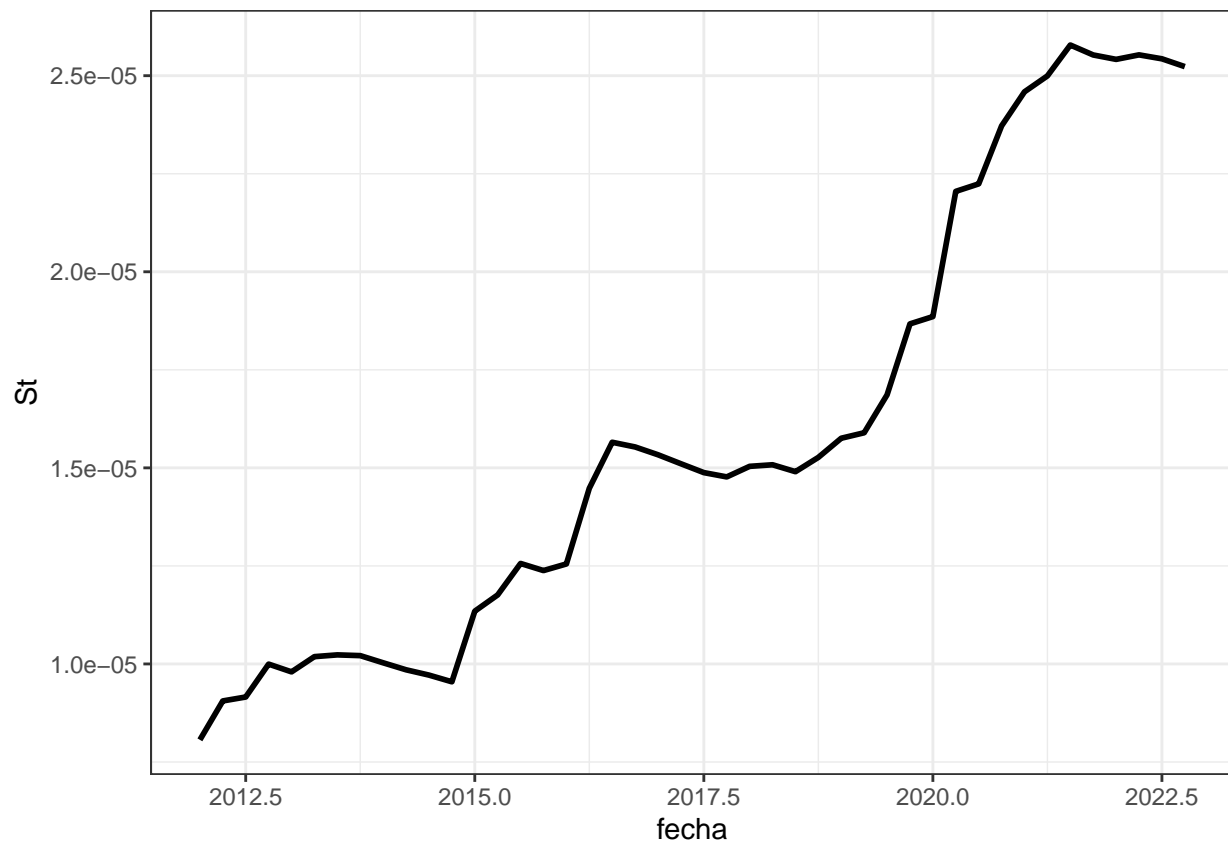
```

Igual podemos ver cómo evoluciona la estimación de la varianza en cada periodo. Gracias a la intervención, la estimación de la varianza no subió tanto en la pandemia comparado con el modelo sin intervenir.

```
df_St_int <- data.frame(dlm_interv$St %>% unlist(), fecha = df_ajuste_int$fecha) %>%
  dplyr::rename(St = 1)

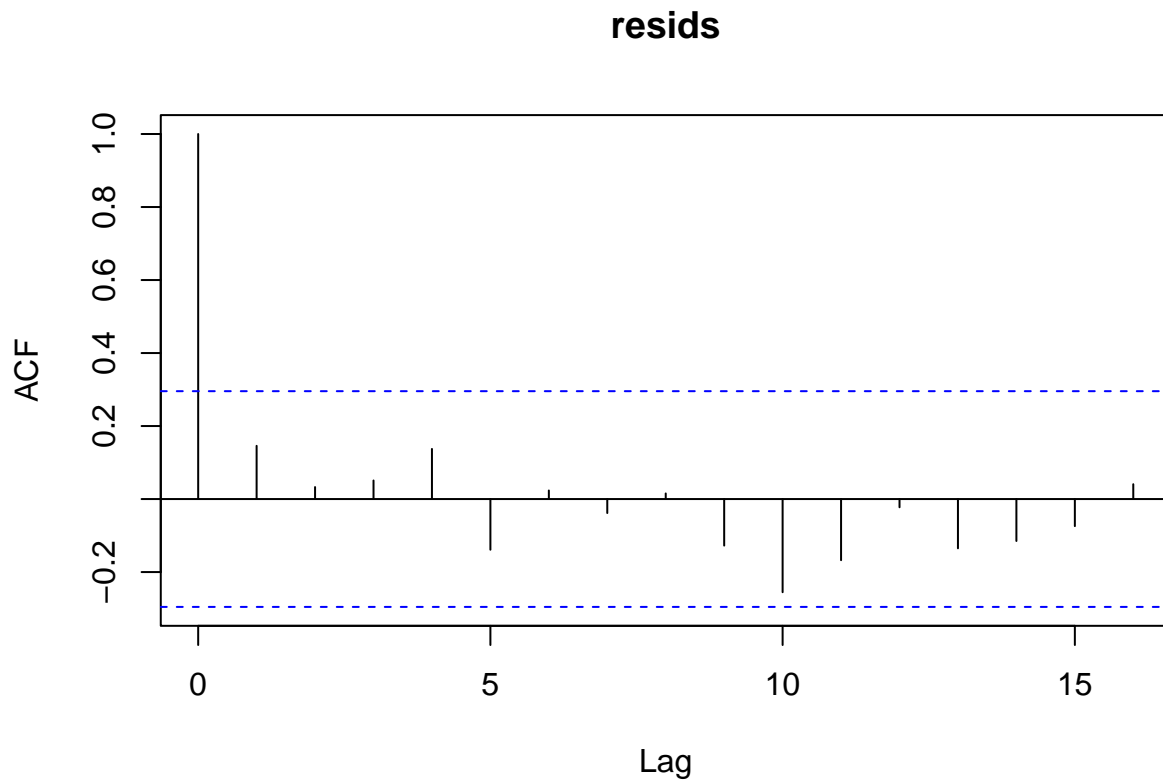
ggplot(df_St_int, aes(x=fecha, y = St)) +
  geom_line(size = 1) +
  theme_bw() +
  ylab("St")
```



La autocorrelación de los residuos se ve muy bien.

```
resids_int <- df_ajuste_int %>%
  mutate(resids = y_real - y_pronostico) %>%
  dplyr::select(resids)

acf(resids_int)
```



Por último, el error cuadrático medio es muy bajo:

```
ecm_int <- mean(resids_int$resids^2)
ecm_int
```

```
## [1] 0.000742012
```

Gracias a la intervención, el error cuadrático medio es mucho más bajo.

Mis función para pronosticar a más de un paso todavía no está listo para las intervenciones, nada más me falta agregar eso para poder tener todos los pronósticos, por ahora con intervenciones solo puedo pronosticar a un paso, no creo tardar tanto en modificar las funciones para poder incluir las intervenciones en los pronósticos a más de un paso.