

# DLM

2023-04-21

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.9
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union
```

```
library('MARSS')
Sys.setlocale(locale = "es_ES.UTF-8")
```

```
## [1] "es_ES.UTF-8/es_ES.UTF-8/es_ES.UTF-8/C/es_ES.UTF-8/en_US.UTF-8"
```

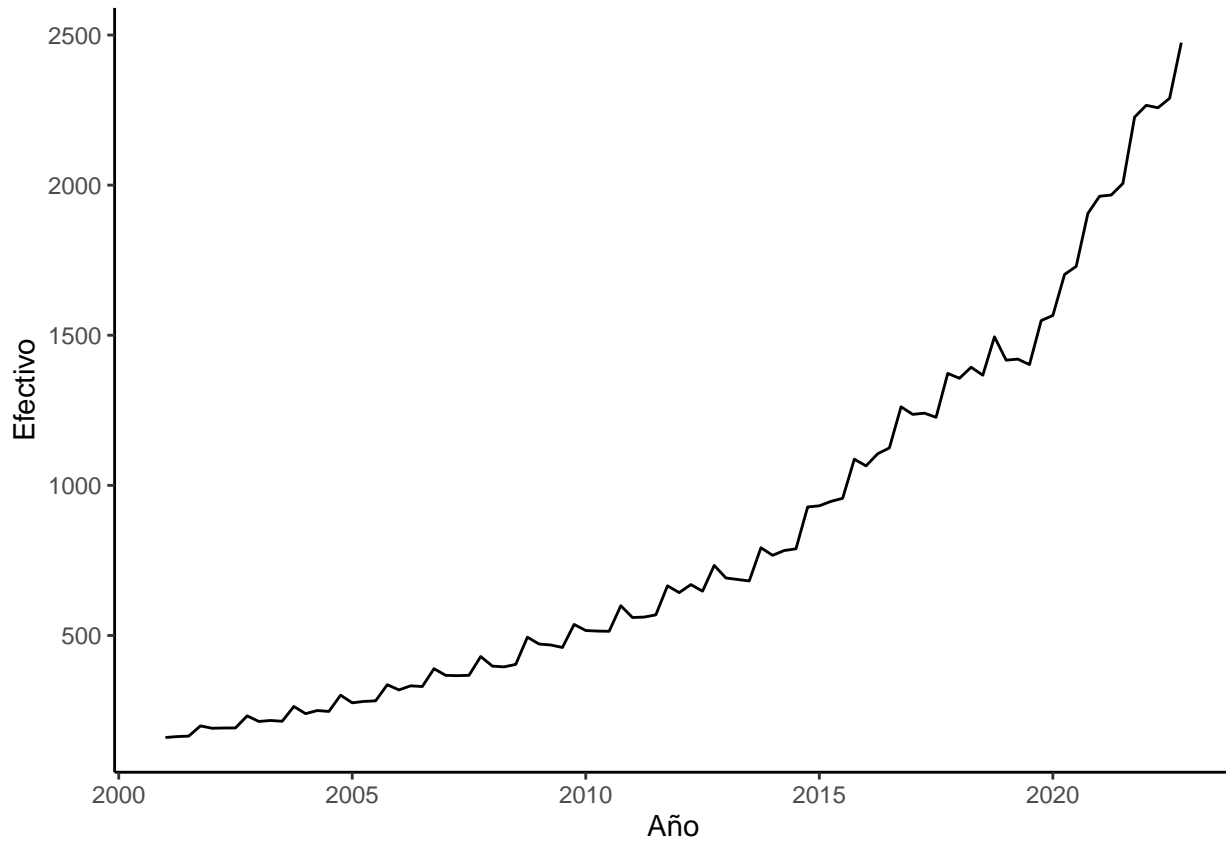
## Variables

En la literatura se menciona que se utilizan variables de escala, como el PIB, y variables de costo de oportunidad, como tasas de interés. La relación entre el PIB y la demanda de efectivo es bastante clara, ambas variables tienen una tendencia creciente con incrementos en los cuartos trimestres:

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

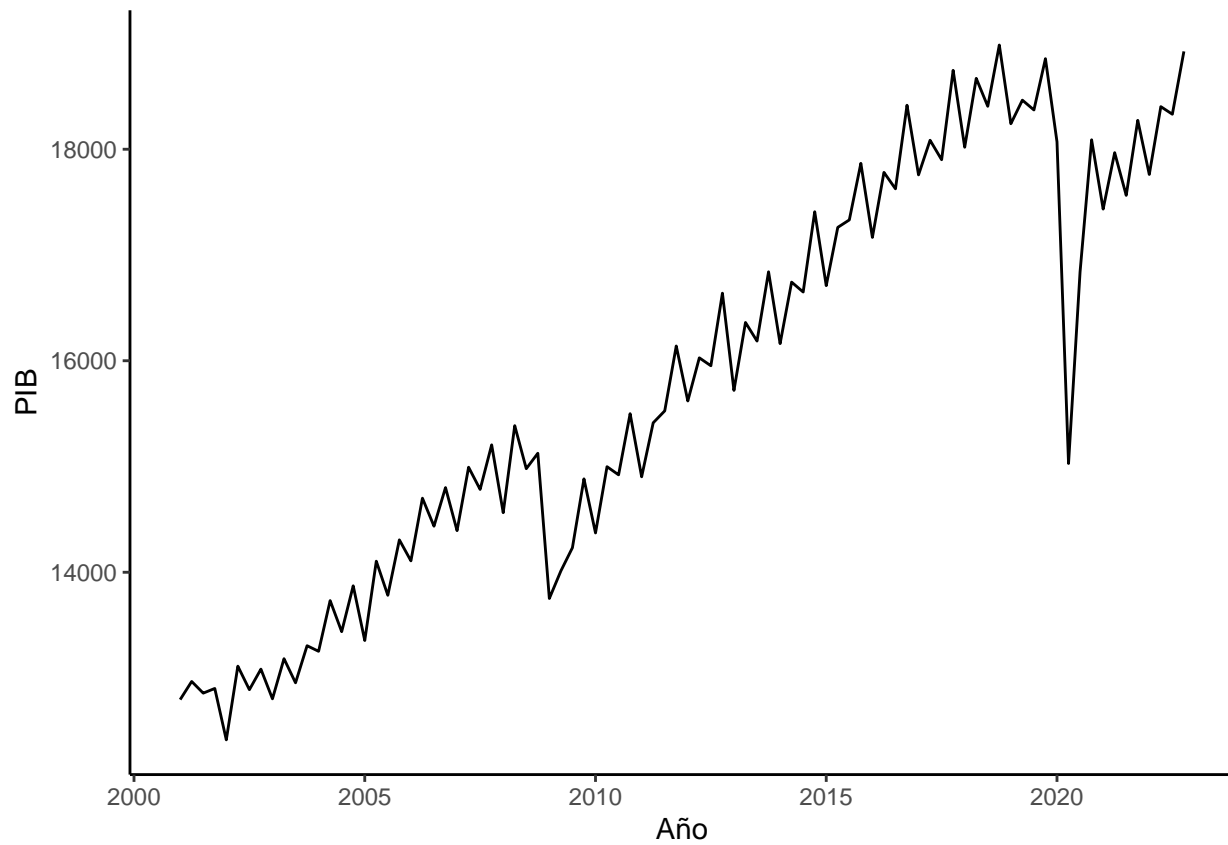
```
ggplot(read_rds('cache/variables/last/efectivo_last.rds'), aes(fecha, efectivo)) +
  geom_line() +
  theme_classic() +
  xlab("Año") +
  ylab("Efectivo")
```

```
## Don't know how to automatically pick scale for object of type yearqtr. Defaulting to continuous.
```



```
ggplot(read_rds('cache/variables/pib.rds'), aes(fecha, pib)) +  
  geom_line() +  
  theme_classic() +  
  xlab("Año") +  
  ylab("PIB")
```

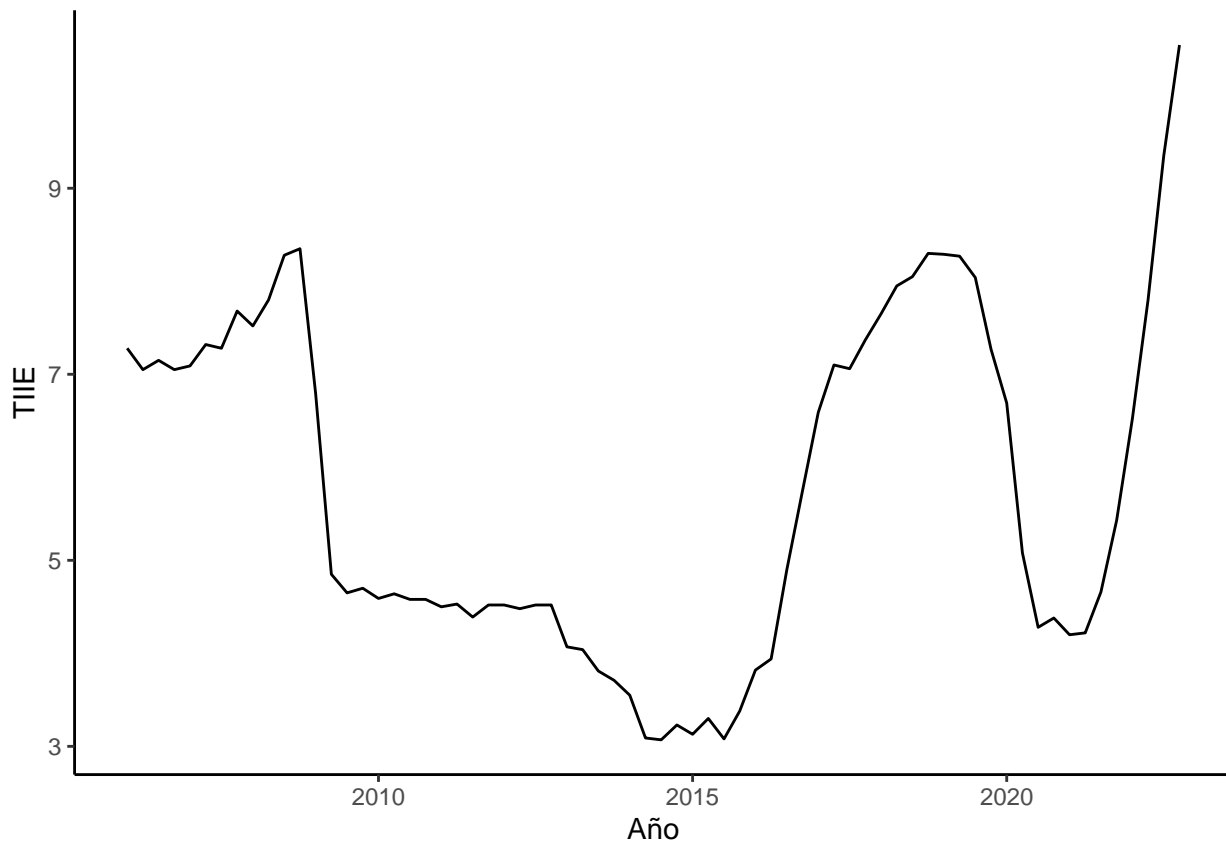
```
## Don't know how to automatically pick scale for object of type yearqtr. Defaulting to continuous.
```



Sin embargo, la TIE tiene un comportamiento muy diferente:

```
ggplot(read_rds('cache/variables/last/tie_last.rds'), aes(fecha, tie)) +
  geom_line() +
  theme_classic() +
  xlab("Año") +
  ylab("TIE")
```

## Don't know how to automatically pick scale for object of type yearqtr. Defaulting to continuous.



Tiene caídas y subidas muy bruscas que en nada parecen afectar a la demanda de efectivo. Además, cuando considero la TIIE el modelo empeora mucho, por esto decidí no incluir la TIIE en el modelo a pesar de que en la literatura se suele incluir una variable de costo de oportunidad.

Por lo tanto, en el modelo incluí un intercepto, el PIB y una variable dummy que vale 1 en los cuartos trimestres y 0 en los otros trimestres.

En los modelos, todas las variables están en logaritmos.

## Problema de pronosticar con MARSS

MARSS utiliza máxima verosimilitud para estimar los valores desconocidos como  $\mathbf{G}$  o  $\mathbf{W}$ . El problema es que utiliza toda la información disponible para hacer la estimación. Entonces, cuando corre el filtro de Kalman, en cada pronóstico a un paso no está considerando únicamente la información anterior, debido a que en las estimaciones de  $\mathbf{G}$  o  $\mathbf{W}$  se tomó en cuenta toda la información de toda la serie. Por esto, lo ideal sería estimar  $\mathbf{G}$  y  $\mathbf{W}$  únicamente con información pasada para ver el verdadero poder predictivo del DLM.

## Enfoque 1: estimar $\mathbf{G}$ , $\mathbf{W}$ y $\mathbf{V}$ como constantes.

En el enfoque 1 se utilizan los datos del 2001 al 2011 para estimar los valores constantes de  $\mathbf{G}$ ,  $\mathbf{W}$  y  $\mathbf{V}$  que se utilizarán para el periodo 2012-2022. Estos valores se obtuvieron con ayuda del paquete MARSS, que utiliza máxima verosimilitud para estimarlos. Los valores obtenidos para  $\mathbf{G}$  son:

```
read_rds('cache/outputs_modelos/G.rds')
```

```
##           [,1]      [,2] [,3]
## [1,] 0.971105 0.000000  0
## [2,] 0.000000 1.002675  0
## [3,] 0.000000 0.000000  1
```

Mientras que los de  $\mathbf{W}$  son:

```
read_rds('cache/outputs_modelos/W.rds')
```

```
##           [,1]      [,2] [,3]
## [1,] 2.233301e-07 0.000000e+00  0
## [2,] 0.000000e+00 1.903777e-11  0
## [3,] 0.000000e+00 0.000000e+00  0
```

Por último, el valor de  $V$  es

```
read_rds('cache/outputs_modelos/V.rds')
```

```
##           Y1
## Y1 0.001240637
```

El problema con este enfoque es que hubo cambios en la estructura de la demanda de efectivo en el periodo 2012-2022 comparado con el periodo 2001-2011. Al utilizar los valores constantes obtenidos, el DLM funciona bien en los primeros trimestres de 2012-2022, pero después se ve claramente que el modelo no es adecuado.

```
datos_pib <- read_rds('cache/variables/pib.rds') %>%
  filter(fecha >= 2012.0) %>%
  mutate(pib = log(pib))

datos_efectivo <- read_rds('cache/variables/last/efectivo_last.rds') %>%
  filter(fecha >= 2012.0) %>%
  mutate(efectivo = log(efectivo))

datos_estacionalidad <- datos_efectivo %>%
  mutate(Q4 = ifelse(fecha - floor(fecha) == 0.75, 1, 0))

#Leo las funciones que hice para el dlm
source('src/funciones/funciones_dlm.R')
```

```
##
## Attaching package: 'LaplacesDemon'

## The following objects are masked from 'package:lubridate':
##
##     dst, interval

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

```

#Valores estimados de 2001-2012
Vt <- read_rds('cache/outputs_modelos/V.rds')

Gt <- read_rds('cache/outputs_modelos/G.rds')

Wt <- read_rds('cache/outputs_modelos/W.rds')

C0 <- read_rds('cache/outputs_modelos/C0.rds')

m0 <- read_rds('cache/outputs_modelos/m0.rds')

datos_F <- datos_pib %>%
  left_join(datos_estacionalidad) %>%
  mutate(intercept = 1) %>%
  dplyr::select(5,2,4)

## Joining, by = "fecha"

dlm_constantes <- actualizacion_dlm(y = datos_efectivo$efectivo, variables_F = datos_F,
                                     m0 = m0, C0 = C0, G = Gt, W = Wt, V = Vt, lista_interv = list())

```

Ahora graficaré los pronósticos a un paso del modelo lineal dinámico. Aquí se nota que el modelo no logra capturar adecuadamente los cambios de la serie después del 2016, y empeora en el 2020 con los cambios provocados por la pandemia.

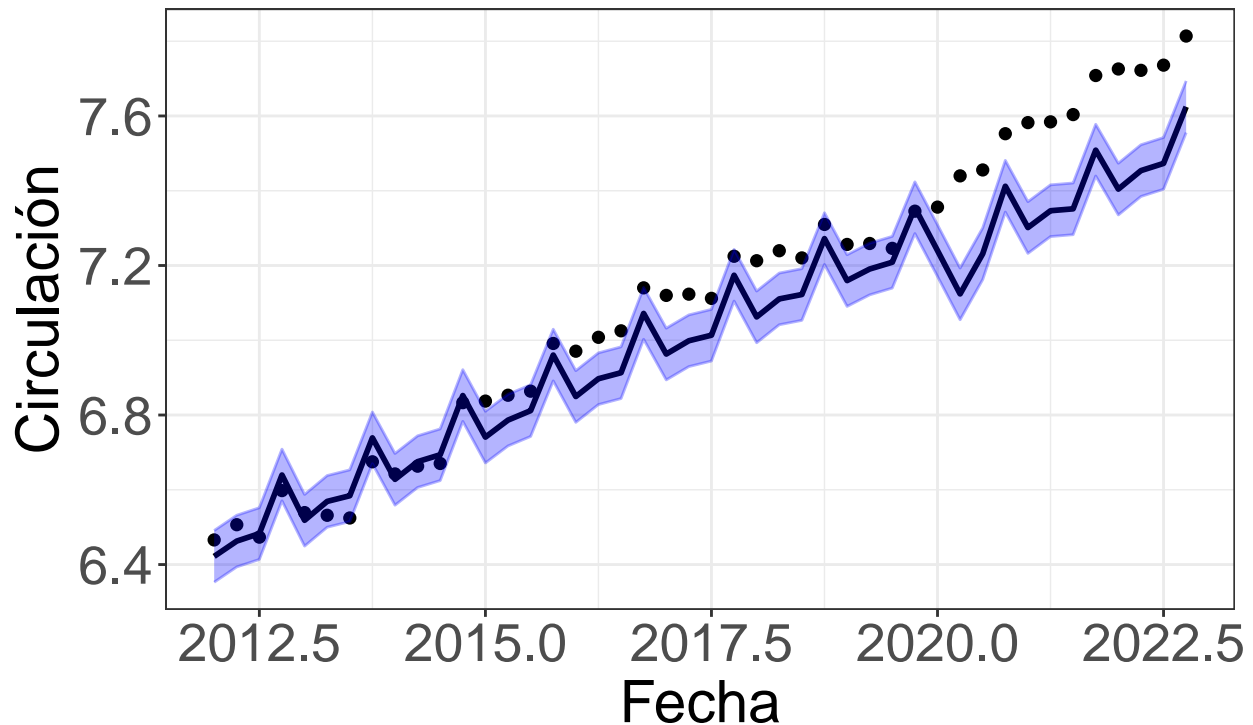
```

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

ggplot(data = df_graficas, 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') +
  theme(axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 20),

```

```
axis.title = element_text(size = 22),
legend.text = element_text(size=20))
```



- Observaciones
- Intervalo al 95% — Pronóstico

El hecho de que  $\mathbf{G}$ ,  $\mathbf{W}$  y  $V$  sean constantes limita mucho el modelo, por lo que decidí aplicar otro enfoque que al final funcionó mucho mejor.

## Enfoque 2: estimar $\mathbf{G}_t$ y $\mathbf{W}_t$ diferentes en cada periodo.

La idea es tomar en cuenta los últimos 12 periodos para estimar  $\mathbf{G}_t$  y  $\mathbf{W}_t$  mediante máxima verosimilitud. Por ejemplo, si nos encontramos en el periodo  $j$ , entonces se utilizan los periodos ya observados de  $j - 12$  a  $j - 1$  para estimar  $\mathbf{G}_j$  y  $\mathbf{W}_j$ . Para estimar por máxima verosimilitud, MARSS necesita que yo indique los valores de la esperanza y varianza de la distribución de parámetros  $\theta_{j-13}$  para iniciar el filtro de Kalman para el periodo de  $j - 12$  a  $j - 1$  y estimar  $\mathbf{G}_j$  y  $\mathbf{W}_j$ . La distribución de parámetros de  $\theta_{j-13}$  que yo meto a MARSS para que pueda iniciar es  $(\theta_{j-13} | D_{j-13}) \sim N[\mathbf{m}_{j-13}, \mathbf{C}_{j-13}]$  que yo obtengo con mi modelo.

Utilizo los primeros 12 periodos, de 2001 a 2003, para estimar los valores iniciales con MARSS de 2004. Después corro el filtro de Kalman usual de 2004 a 2022, y solo utilizo MARSS para obtener  $\mathbf{G}_t$  y  $\mathbf{W}_t$  en cada periodo.

El código por ahora está muy desordenado, tengo que limpiarlo y simplificarlo, realmente lo que me interesa es mostrarte el modelo a ver qué te parece.

## Modelo 1.

En este modelo se estiman  $\mathbf{G}_t$ ,  $\mathbf{W}_t$  y  $V_t$  por máxima verosimilitud en cada periodo, con el mecanismo que se describió anteriormente.

```

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

# Datos efectivo -----

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

# Estacionalidad -----

datos_estacionalidad <- datos_efectivo %>%
  mutate(Q4 = ifelse(fecha - floor(fecha) == 0.75,1,0))

# Inputs para MARSS

## number of periods of data
TT <- length(datos_efectivo$efectivo)

## get predictor variable
pib <- matrix(datos_pib$pib, nrow=1)
q4 <- matrix(datos_estacionalidad$Q4, nrow = 1)

## number of regr params (slope + intercept)
m <- 3

## se definen los valores de G a estimar.
G <- matrix(list(0), m, m)
G[1,1] <- 'G.1'
G[2,2] <- 'G.2'
G[3,3] <- 1

#Se definen los valores de W a estimar
W <- matrix(list(0), m, m) ## 2x2; all 0 for now #Es Wt
W[1,1] <- 'W.1'
W[2,2] <- 'W.2'
W[3,3] <- 0

## for observation eqn
F <- array(NA, c(1, m, TT)) ## NxMxT; empty for now #Es Ft transpuesta
F[1, 1, ] <- rep(1, TT) ## Nx1; 1's for intercept
F[1, 2, ] <- pib ## Nx1; predictor variable
F[1, 3, ] <- q4 ## Nx1; predictor variable

y_hist <- matrix(datos_efectivo$efectivo, nrow = 1)

#Leo las funciones que hice para el DLM
source('src/funciones/funciones_sin_tie.R')

```



```

Vt <- matrix("v")

C0 <- read_rds('cache/outputs_modelos/12_periodos/sin tiie/C0.rds')

m0 <- read_rds('cache/outputs_modelos/12_periodos/sin tiie/m0.rds')

datos_F <- datos_pib %>%
  filter(fecha >= 2004) %>%
  left_join(datos_estacionalidad %>%
    filter(fecha >= 2004)) %>%
  mutate(intercept = 1) %>%
  dplyr::select(5,2,4)

## Joining, by = "fecha"

datos_efectivo_y <- datos_efectivo %>%
  filter(fecha >= 2004)

dlm_1 <- actualizacion_dlm_estima_G_W(y = datos_efectivo_y$efectivo, variables_F = datos_F,
  y_hist = y_hist, variables_F_hist = F, inicio = 12,
  m0 = m0, C0 = C0, G_forma = G, W_forma = W, V_forma = Vt,
  lista_interv = list())

```

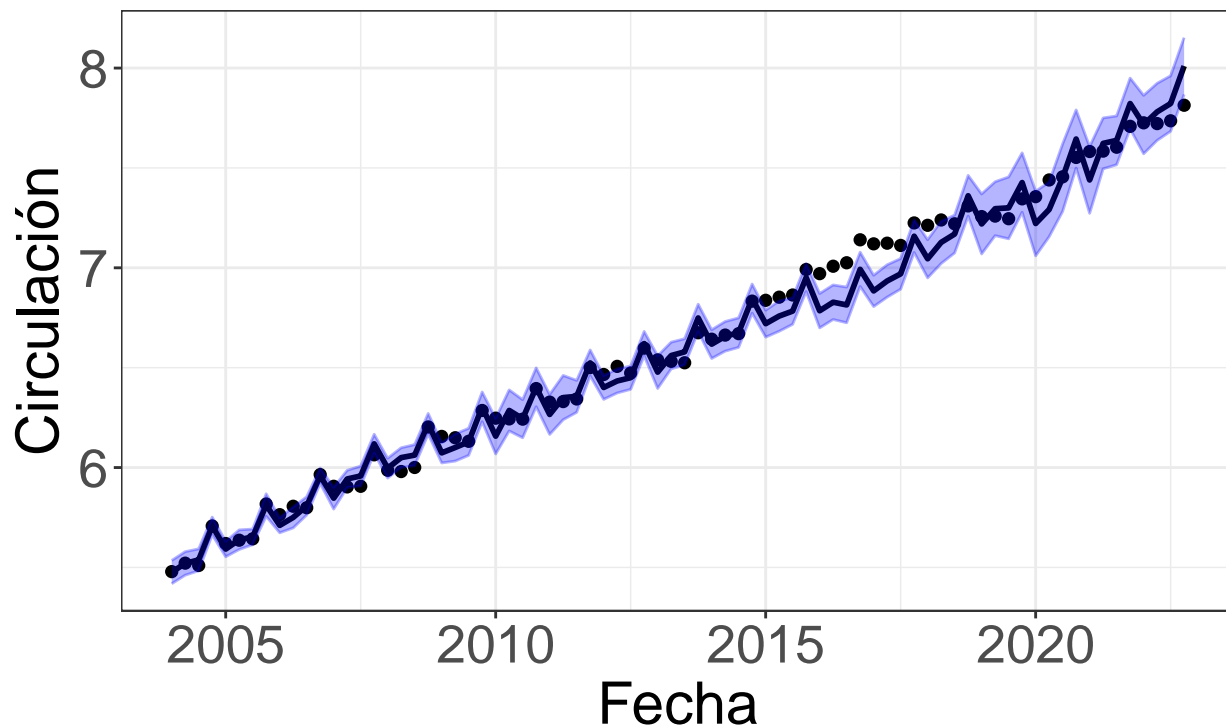
Ahora grafiquemos el resultado de este modelo.

```

df_graficas <- data.frame("fecha" = datos_efectivo_y %>% dplyr::select(fecha),
  "y_real" = datos_efectivo_y$efectivo,
  "y_pronostico" = dlm_1$ft %>% unlist(), "CI_inf" = dlm_1$CI_inf %>% unlist(),
  "CI_sup" = dlm_1$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))

ggplot(data = df_graficas, 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 = "") +
  ylab('Circulación') +
  xlab('Fecha') +
  theme(axis.text.x = element_text(size = 20),
    axis.text.y = element_text(size = 20),
    axis.title = element_text(size = 22),
    legend.text = element_text(size=20))

```



- Observaciones ■ Intervalo al 95% — Pronóstico

La mejora en comparación con el modelo del enfoque 1 es evidente. Sin embargo, en los años 2016 y 2017 los pronósticos se quedaron por debajo de lo que en realidad pasó. Intenté mejorar esto con otro modelo.

## Modelo 2.

Es muy parecido al modelo anterior, la única diferencia es que en este modelo no se estima  $V_t$  por máxima verosimilitud. Aquí utilizo la versión del DLM en la que  $V$  es una constante desconocida, y su valor estimado se va actualizando dentro del filtro de Kalman, no por máxima verosimilitud.

```
dml_2 <- actualizacion_dlm_V_desc_estima_G_W(y = datos_efectivo_y$efectivo,
                                             variables_F = datos_F,
                                             y_hist = y_hist, variables_F_hist = F, inicio = 12,
                                             m0 = m0, C0 = C0, G_forma = G, W_forma = W,
                                             lista_interv = list(), S0 = as.numeric(0.00006940448), n0 = 1)

df_graficas2 <- data.frame("fecha" = datos_efectivo_y %>% dplyr::select(fecha),
                           "y_real" = datos_efectivo_y$efectivo,
                           "y_pronostico" = dml_2$ft %>% unlist(), "CI_inf" = dml_2$CI_inf %>% unlist(),
                           "CI_sup" = dml_2$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))

ggplot(data = df_graficas2, 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) +
```

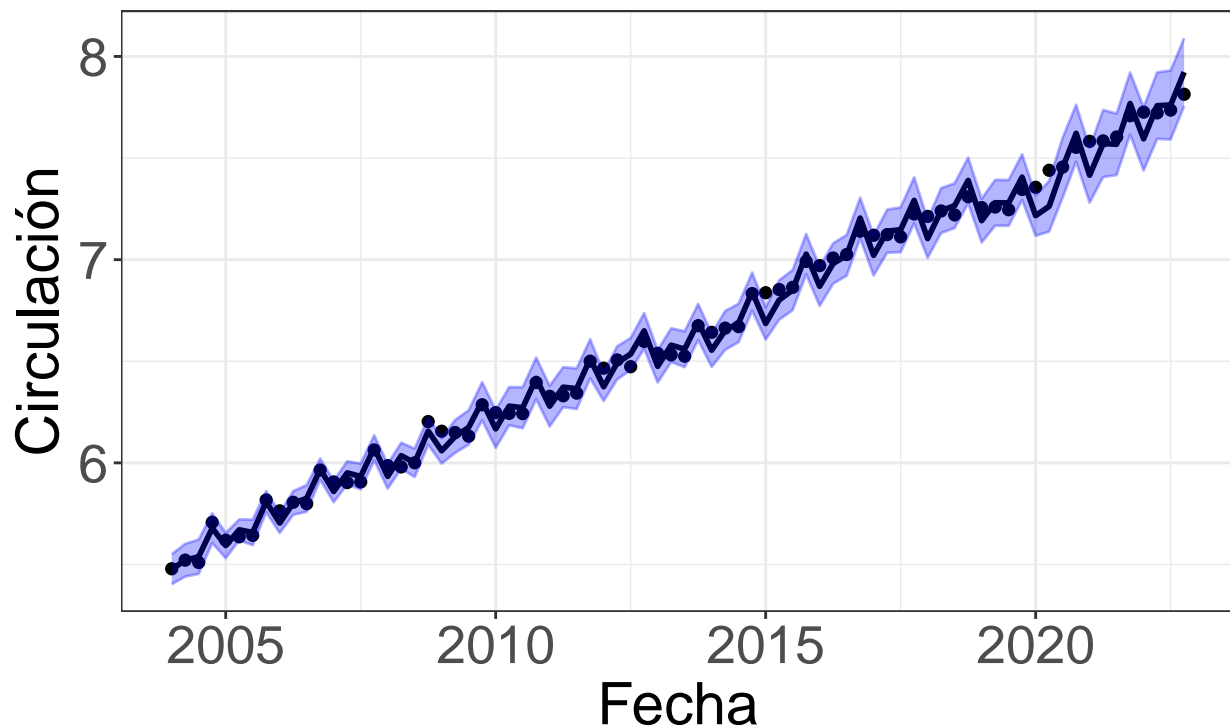
```

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') +
theme(axis.text.x = element_text(size = 20),
      axis.text.y = element_text(size = 20),
      axis.title = element_text(size = 22),
      legend.text = element_text(size=20))

```



- Observaciones  Intervalo al 95% — Pronóstico

Este modelo funciona mucho mejor. Podemos ver cómo evolucionaron los parámetros a lo largo del tiempo:

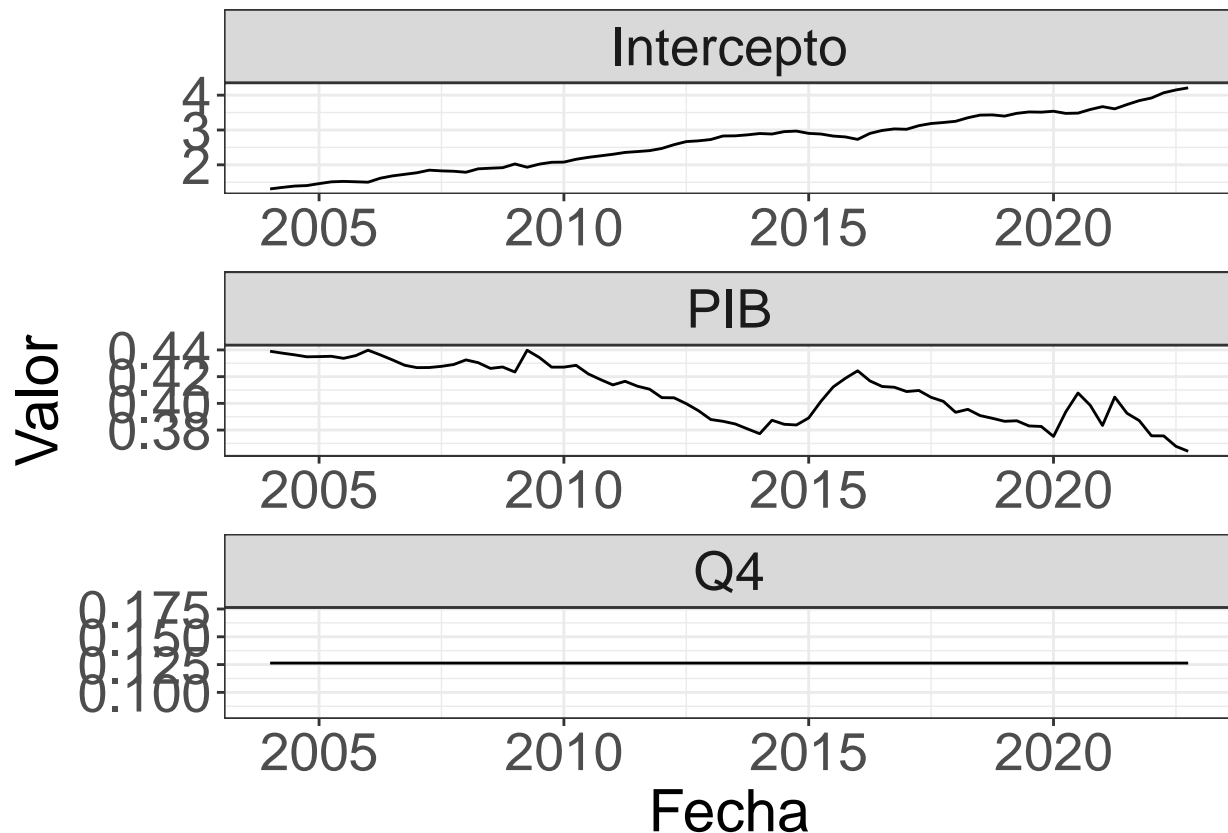
```

df_params2 <- data.frame(reduce(dlm_2$at, cbind) %>% t(), fecha = df_graficas2$fecha) %>%
  rename(Intercepto = X1, PIB = X2, Q4 = X3) %>%
  pivot_longer(names_to = "parametro", values_to = "valor",
               cols = c(Intercepto, PIB, Q4))

ggplot(df_params2, aes(x=fecha, y = valor)) +
  geom_line() +

```

```
facet_wrap(~parametro, nrow = 4, scales = "free") +
theme_bw() +
ylab("Valor") +
xlab("Fecha") +
theme(axis.text.x = element_text(size = 20),
      axis.text.y = element_text(size = 20),
      axis.title = element_text(size = 22),
      strip.text = element_text(size=20))
```



### Modelo 3

El modelo 2 funciona bien, pero veo que a partir del 2016 los pronósticos en los cuartos trimestres superan a la realidad, por lo que el coeficiente de Q4 debe estar muy alto. Un modelador que se encuentra en el 2018 pudo haber notado este problema en 2016 y 2017, entonces en 2018 decidió intervenir el modelo para corregir esto. La intervención consiste en disminuir el coeficiente de la variable Q4 en 2018 T4 ( $t=60$ ).

Otro fallo sucedió al inicio de la pandemia, en 2020. Esto era de esperarse, debido a que el PIB tuvo una fuerte caída, pero la demanda de efectivo comenzó a subir por la incertidumbre generada por la pandemia, la gente quería liquidez en caso de perder sus trabajos o de enfermarse. Este es un ejemplo perfecto de conocimiento externo que no está incluido en el modelo. Realizaré una intervención en 2020 T2 ( $t=66$ ), subiré el valor de la ordenada al origen.

```
# Se definen las dos intervenciones
list_interv <- list("t_int" = list(60, 66), "at_int" = list(matrix(c(3.4331459,
                                                                    0.3888267,
                                                                    0.09), nrow=3),
```

```

matrix(c(3.7,
        0.3857346,
        0.09), nrow=3)),

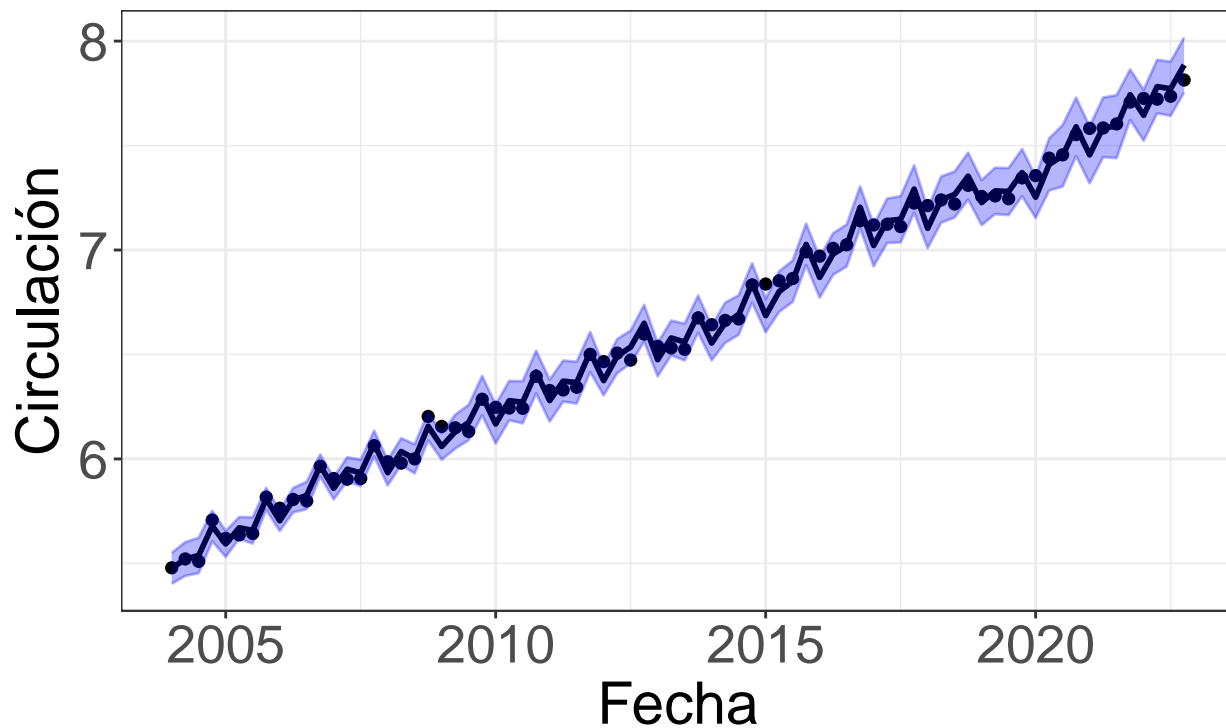
    "Rt_int" = list(dlm_2$Rt[[60]], dlm_2$Rt[[66]]))

dlm_3 <- actualizacion_dlm_V_desc_estima_G_W(y = datos_efectivo_y$efectivo,
      variables_F = datos_F,
      y_hist = y_hist, variables_F_hist = F, inicio = 12,
      m0 = m0, C0 = C0, G_forma = G, W_forma = W,
      lista_interv = list_interv, S0 = as.numeric(0.00006940448)

df_graficas3 <- data.frame("fecha" = datos_efectivo_y %>% dplyr::select(fecha),
      "y_real" = datos_efectivo_y$efectivo,
      "y_pronostico" = dlm_3$ft %>% unlist(), "CI_inf" = dlm_3$CI_inf %>% unlist()
      "CI_sup" = dlm_3$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))

ggplot(data = df_graficas3, 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') +
  theme(axis.text.x = element_text(size = 20),
    axis.text.y = element_text(size = 20),
    axis.title = element_text(size = 22),
    legend.text = element_text(size=20))

```

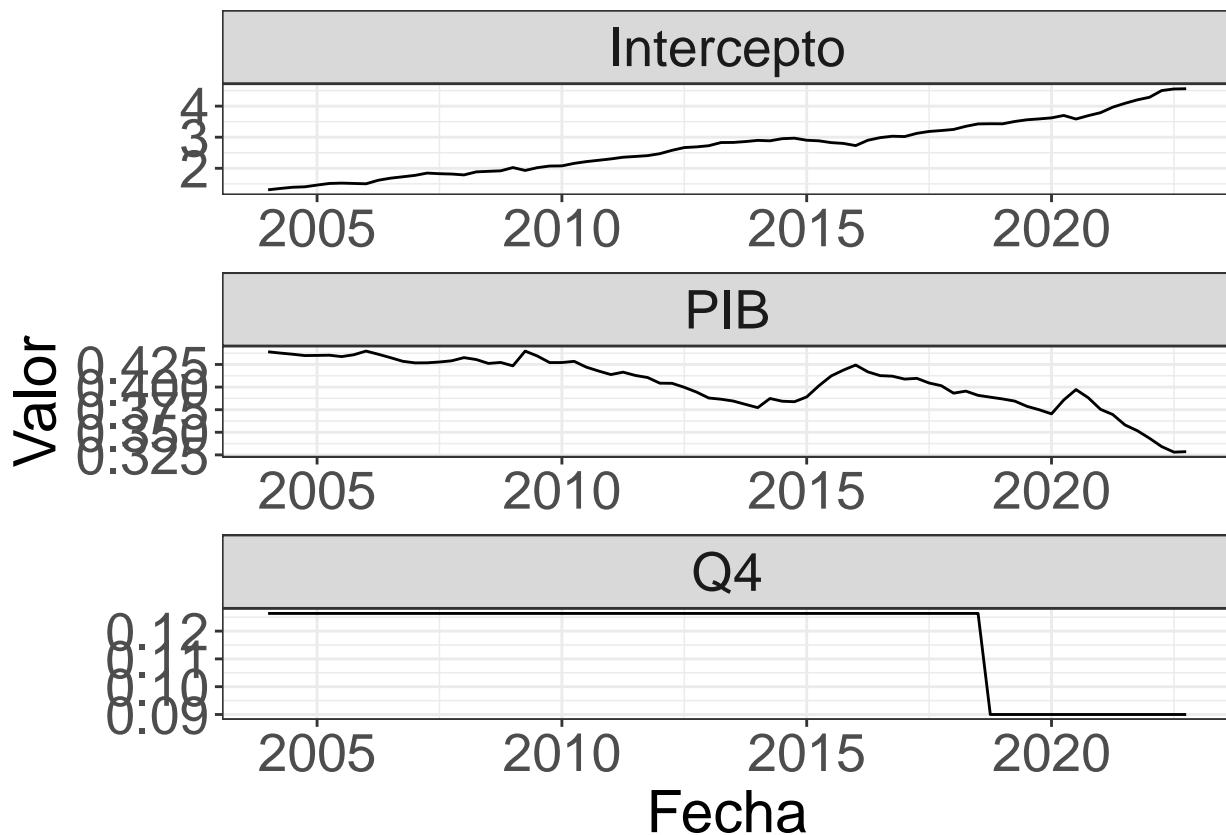


- Observaciones  Intervalo al 95% — Pronóstico

Con esto ya se corrigieron estos problemas. El efecto de las intervenciones se pueden notar en las gráficas de los coeficientes:

```
df_params3 <- data.frame(reduce(dlm_3$at, cbind) %>% t(), fecha = df_graficas3$fecha) %>%
  rename(Intercepto = X1, PIB = X2, Q4 = X3) %>%
  pivot_longer(names_to = "parametro", values_to = "valor",
    cols = c(Intercepto, PIB, Q4))

ggplot(df_params3, aes(x=fecha, y = valor)) +
  geom_line() +
  facet_wrap(~parametro, nrow = 4, scales = "free") +
  theme_bw() +
  ylab("Valor") +
  xlab("Fecha") +
  theme(axis.text.x = element_text(size = 20),
    axis.text.y = element_text(size = 20),
    axis.title = element_text(size = 22),
    strip.text = element_text(size=20))
```

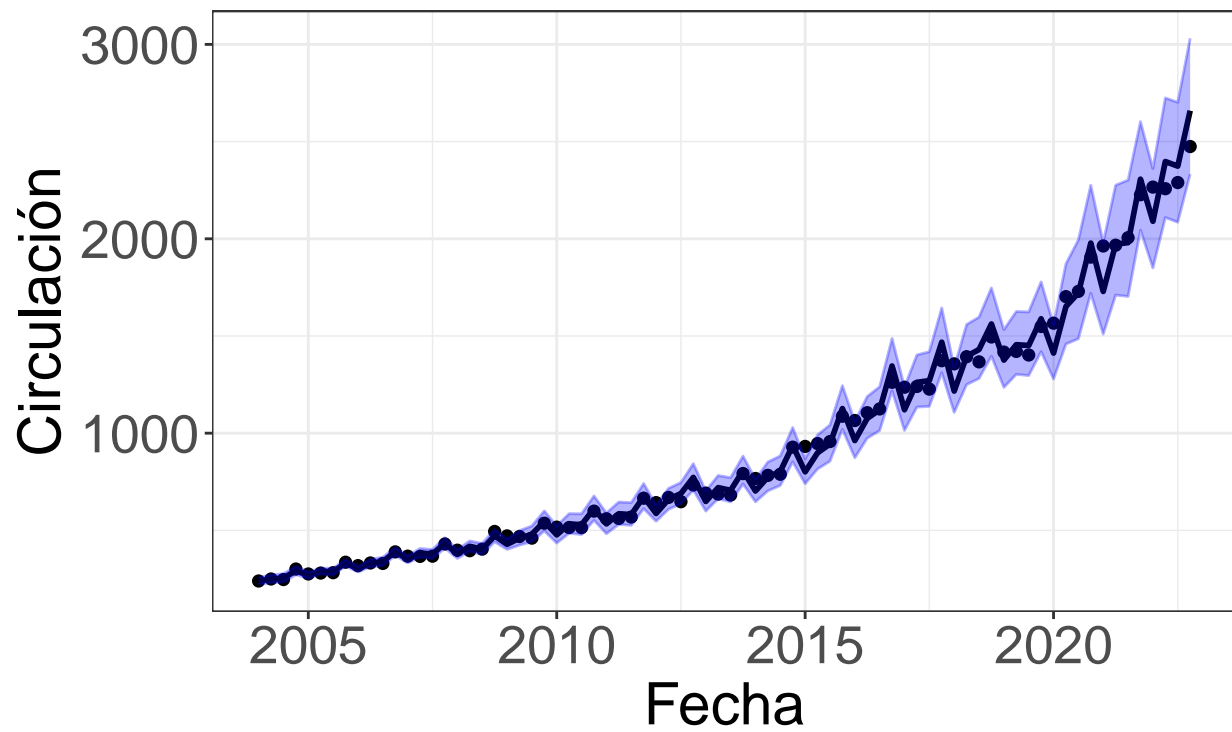


## Gráfica final.

Como mencioné al principio, el PIB y la demanda de efectivo en los modelos están en logaritmos. Voy a convertir los resultados del modelo a la escala original para ver cómo resulta el modelo:

```
ggplot(data = df_graficas3 %>%
  mutate(across(y_real:CI_sup, ~exp(.))), 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') +
  theme(axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 20),
```

```
axis.title = element_text(size = 22),  
legend.text = element_text(size=20))
```



- Observaciones
- Intervalo al 95% — Pronós