intervencion

Se utiliza la base de datos **freeny**.

Se definen los valores iniciales en $t = 20$.

```
#A. Posterior en t=19
m19 <-c(1.5, 1.8,-0.7)
C19 <- matrix(c(0.00002, 0.00001,-0.00002, 0.00001,
                0.00003,-0.00001,-0.00002,-0.00001, 0.00002),
              ncol = 3)
#Valores conocidos de G20 y W20
G20 <-matrix(c(1.001, 0, 0, 0, 1, 0, 0, 0, 1), ncol=3)
W20 <-matrix(c(0.00001, 0, 0, 0, 0.00001,-0.00001, 0,-0.00001, 0.00005), ncol=3)
S19 <- 0.00005# Estimación de V en T=19
n19 <- 19.5# Grados de libertad
```

Actualización de ecuaciones

Función para actualizar más de un periodo cuando la varianza de observación es desconocida y se introduce una lista con las intervenciones deseadas.

```
# Se modifican los datos para ejemplificar la intervención
datos_ej <- freeny %>%
  mutate(intercept = 1) %>%
  slice(20:n()) %>%
  dplyr::select(y, intercept, income.level, price.index) %>%
  mutate(y = ifelse(y>=9.6, y+0.5,y))
#Metemos ruido para el ejemplo de la intervención
set.seed(2)
datos_ej$y[11:20] <- datos_ej$y[11:20] + rnorm(10,0,0.05)
datos_ej
```

```
##              y intercept income.level price.index
## 1967     9.314         1        6.061       4.510
## 1967.25  9.350         1        6.071       4.504
## 1967.5   9.358         1        6.080       4.494
## 1967.75  9.398         1        6.089       4.465
## 1968     9.421         1        6.102       4.449
## 1968.25  9.442         1        6.112       4.440
## 1968.5   9.487         1        6.116       4.420
## 1968.75  9.524         1        6.121       4.411
## 1969     9.540         1        6.122       4.412
## 1969.25  9.581         1        6.131       4.398
## 1969.5  10.056         1        6.147       4.385
## 1969.75 10.154         1        6.153       4.373
## 1970    10.223         1        6.156       4.328
```

```
## 1970.25 10.138         1        6.163       4.320
## 1970.5  10.196         1        6.174       4.309
## 1970.75 10.193         1        6.161       4.309
## 1971    10.253         1        6.182       4.306
## 1971.25 10.237         1        6.188       4.296
## 1971.5  10.375         1        6.194       4.278
## 1971.75 10.287         1        6.200       4.278
```

```r
#Se asume que Gt, Vt y Wt son ctes conocidas para toda t.
# lista_interv define el tiempo y los estados de las intervenciones
actualizacion_V_desc <- function(datos, m0, C0, G, W, S0, n0, lista_interv){

  error_varianza <- function(x) {
  if(any (diag(x)<0)) stop ('Elementos del parametro de escala o varianzas deben ser positivos')
  }

  mt_menos_1 <- m0
  Ct_menos_1 <- C0
  St_menos_1 <- S0
  nt_menos_1 <- n0
  lista_at <- list()
  lista_Rt <- list()
  lista_ft <- list()
  lista_Qt <- list()
  lista_mt <- list()
  lista_Ct <- list()
  lista_CI <- list()
  lista_CI_inf <- list()
  lista_CI_sup <- list()
  lista_St <- list()
  interv <- F
  for(t in 1:length(datos$y)){

    if(t %in% lista_interv$t_int){
      at <- lista_interv$at_int[[match(t,list_interv$t_int)]]
      Rt <- lista_interv$Rt_int[[match(t,list_interv$t_int)]]
      interv <- T
    } else {
      at <- G %*% mt_menos_1
      Rt <- G %*% Ct_menos_1 %*% t(G) + W
    }

    Ft <- as.numeric(datos[t, 2:4])
    ft <- t(Ft) %*% at
    Qt <- t(Ft) %*% Rt %*% Ft + St_menos_1
    CI <- c(qst(0.025, nu = nt_menos_1, mu = ft, sigma = sqrt(Qt)),
            qst(0.975, nu = nt_menos_1, mu = ft, sigma = sqrt(Qt)))
    CI_inf <- CI[1]
    CI_sup <- CI[2]
    Yt <- datos[t,1]
    At <- Rt %*% Ft %*% solve(Qt)
    et <- Yt-ft
    mt <- at + At %*% et
    nt <- nt_menos_1 + 1
```

```
    St <- as.numeric(St_menos_1 + (St_menos_1/nt) * (et^2 %*% solve(Qt) - 1))
    Ct <- (St/St_menos_1)*(Rt - At %*% Qt %*% t(At))

    error_varianza(Rt)
    error_varianza(St)
    error_varianza(Ct)

    lista_at[[t]] <- at
    lista_Rt[[t]] <- Rt
    lista_ft[[t]] <- ft
    lista_Qt[[t]] <- Qt
    lista_CI[[t]] <- CI
    lista_CI_inf[[t]] <- CI_inf
    lista_CI_sup[[t]] <- CI_sup
    lista_mt[[t]] <- mt
    lista_Ct[[t]] <- Ct
    lista_St[[t]] <- St
    mt_menos_1 <- mt
    Ct_menos_1 <- Ct
    nt_menos_1 <- nt
    St_menos_1 <- St

    interv <- F
  }
  return(list("at" = lista_at,  "Rt" = lista_Rt,  "ft" = lista_ft,
              "Qt" = lista_Qt, "CI" = lista_CI, "CI_inf" = lista_CI_inf,
              "CI_sup" = lista_CI_sup, "mt" = lista_mt, "Ct" = lista_Ct,
              "St" = lista_St))
}
```

Se aplica la función sin intervención definida.

```
res_dlm_no_int <- actualizacion_V_desc(datos_ej, m19, C19, G20, W20, S19, n19, list())
```

Se define la intervención

```
a30_int <- res_dlm_no_int$at[[11]] %>% as_vector()
a30_int[2] <- 1.9
R30_int <- res_dlm_no_int$Rt[[11]]
R30_int[2,2] <- 0.0002
list_interv <- list("t_int" = list(11), "at_int" = list(a30_int),
     "Rt_int" = list(R30_int))
```

Se aplica la función con la intervención.

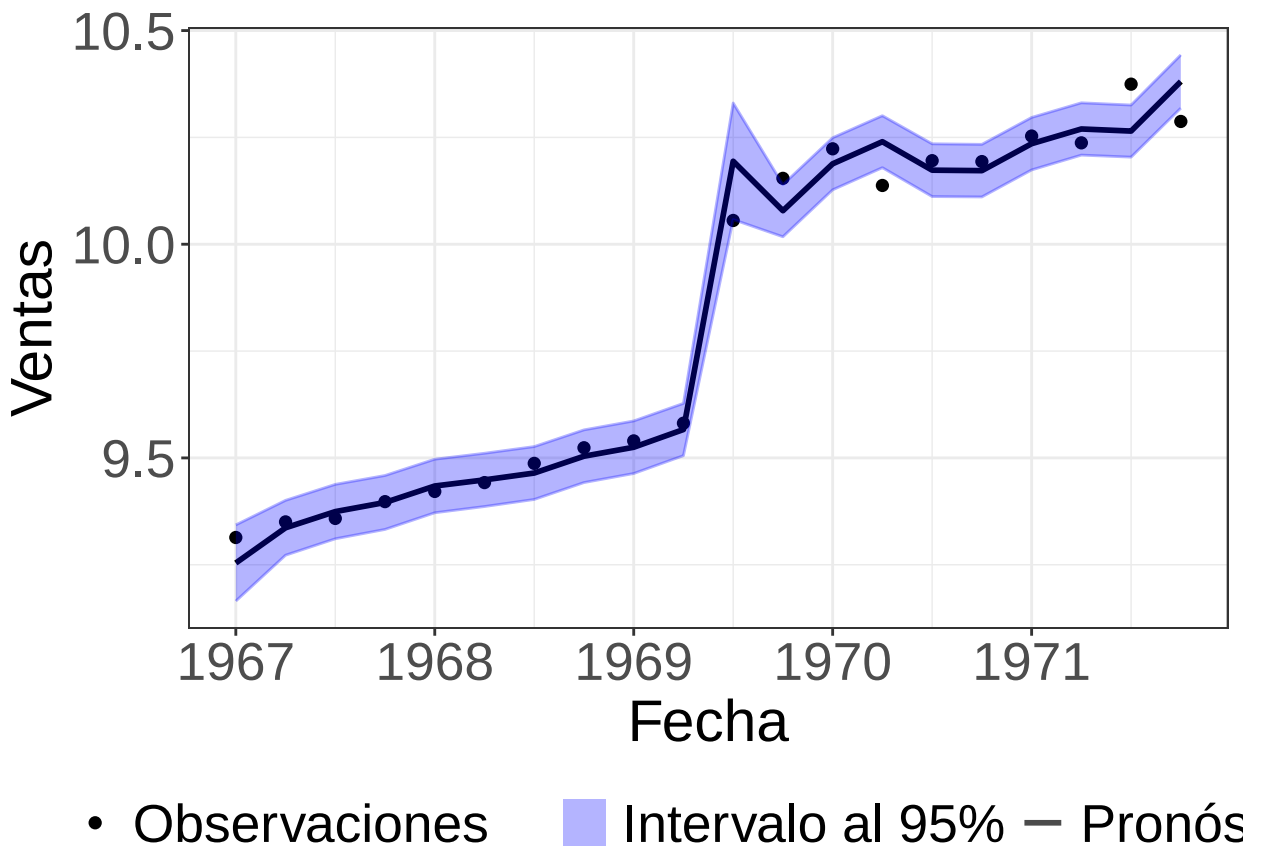
```
res_dlm <- actualizacion_V_desc(datos_ej, m19, C19, G20, W20, S19, n19, list_interv)
list_interv$at_int[[match(11,list_interv$t_int)]]
```

```
##         [,1]
## [1,]  1.5172
## [2,]  1.9000
## [3,] -0.6847
```

Se grafican los resultados del modelo intervenido.

```
df_graficas <- data.frame("fecha" = datos_ej %>% row.names(), "y_real" = datos_ej$y,
           "y_pronostico" = res_dlm$ft %>% unlist(), "CI_inf" = res_dlm$CI_inf %>% unlist(),
           "CI_sup" = res_dlm$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))
```

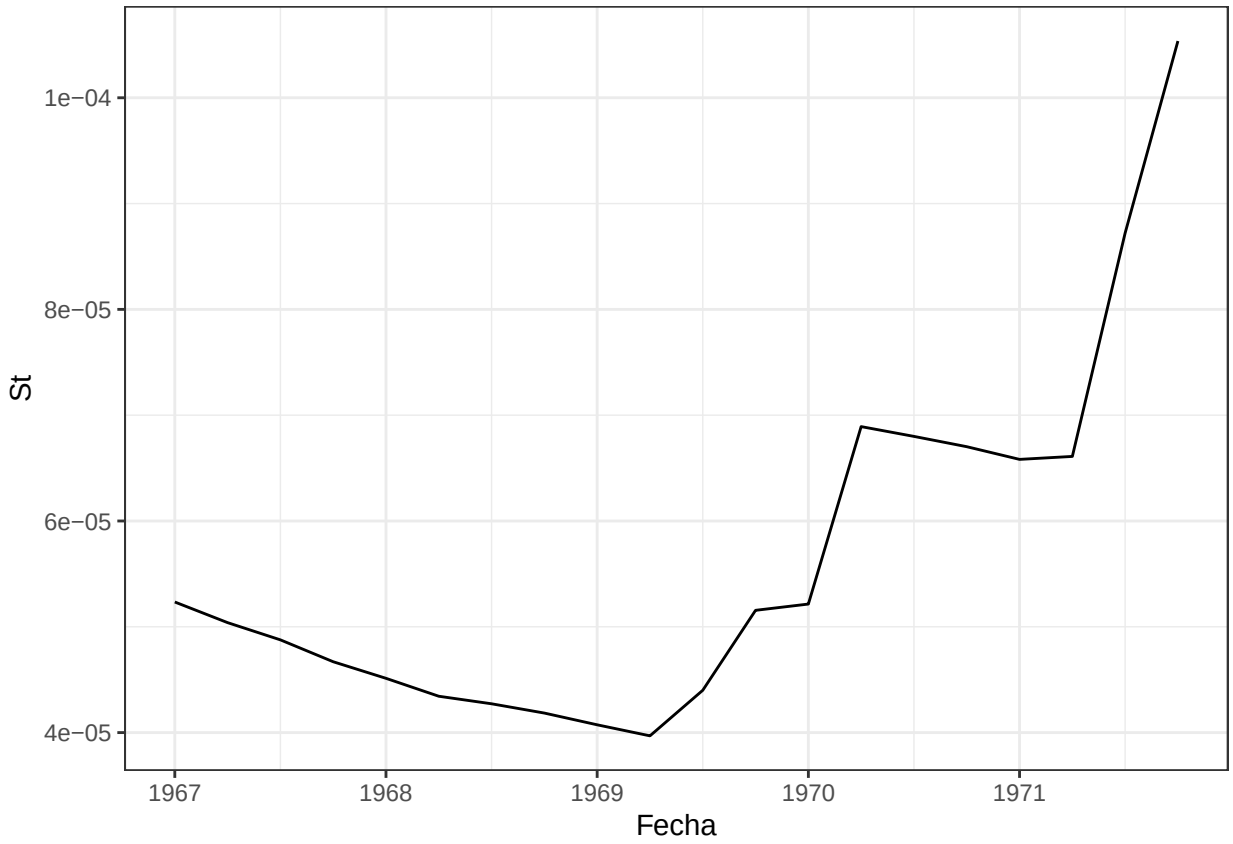
```
ggplot(data = df_graficas, aes(x = fecha)) +
  geom_point(aes(y = y_real, shape = "Observaciones"), size = 2) +
  geom_line(aes(y = y_pronostico, color = 'Pronósticos'), size = 1) +
  geom_line(aes(y = CI_inf), color = "blue", alpha = 0.3) +
  geom_line(aes(y = CI_sup), color = "blue", alpha = 0.3) +
  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('Ventas') +
  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))
```




```
ggsave(filename = "graphs/teoria/intervencion/actualizacion_interv.png", width = 11.7, height = 6)
```

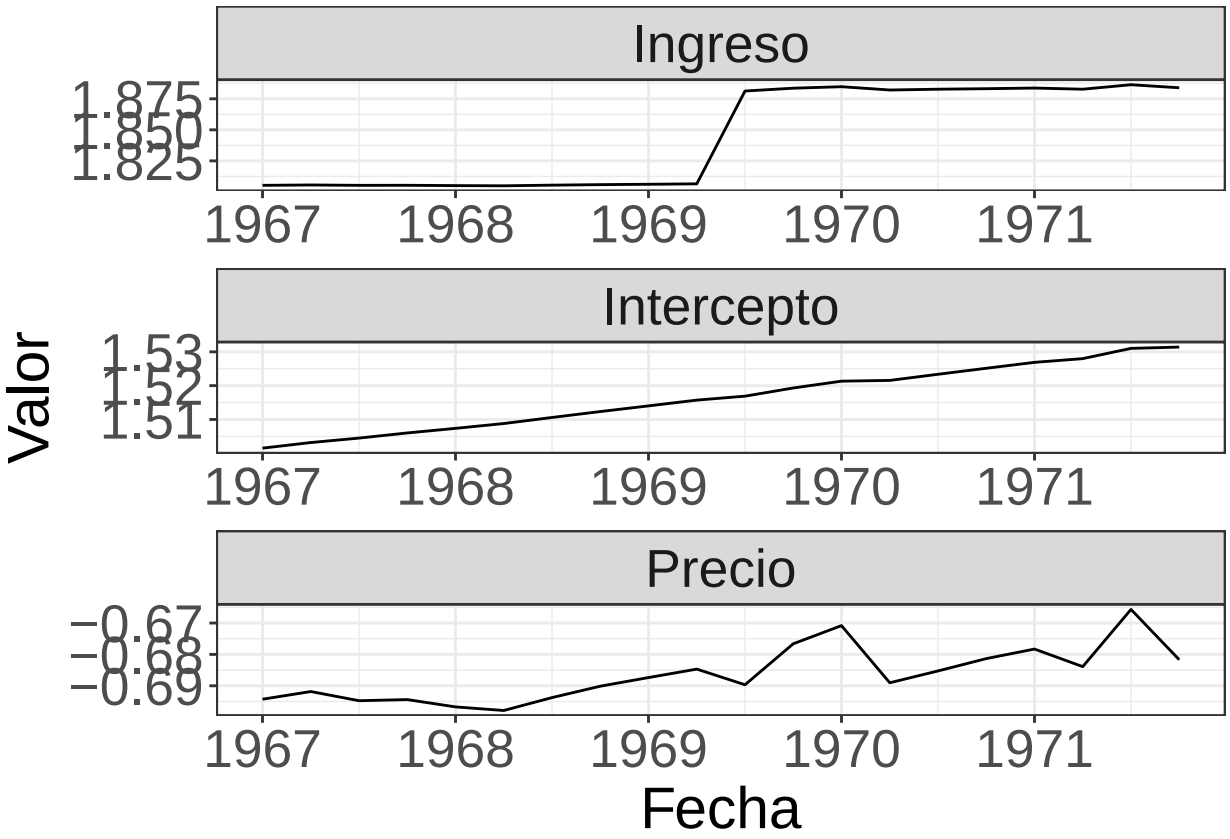
```
df_St <- data.frame(res_dlm$St %>% unlist(), fecha = df_graficas$fecha) %>%
  rename(St = 1)
```

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




```
df_params <- data.frame(reduce(res_dlm$mt, cbind) %>% t(), fecha = df_graficas$fecha)  %>%
  rename(Intercepto = X1, Ingreso = X2, Precio = X3) %>%
  pivot_longer(names_to = "parametro", values_to = "valor",
               cols = c(Intercepto, Ingreso, Precio))
```

```
ggplot(df_params, aes(x=fecha, y = valor)) +
  geom_line() +
  facet_wrap(~parametro, nrow = 3, 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))
```




```
#ggsave(filename = "graphs/teoria/actualizacion/parametros.png", width = 11.7, height = 6)
```

Ahora, se muestra paso por paso la actualización de ecuaciones con intervención. Se parte de la distribución posterior en $t = 29$ y se obtiene la distribución a priori en $t = 30$. Se supone que la varianza de observación es deconocida. En este ejemplo, se realizó una intervención en $t = 30$.

```
#Valores conocidos de G30 y W30 no son necesarios por la intervención

#Priori de parámetros en t=30 intervenidos
a30_int
```

```
##         [,1]
## [1,]  1.5172
## [2,]  1.9000
## [3,] -0.6847
```

```
R30_int
```

```
##            [,1]       [,2]       [,3]
## [1,]  1.126e-04  6.172e-06 -0.0000318
## [2,]  6.172e-06  2.000e-04 -0.0001445
## [3,] -3.180e-05 -1.445e-04  0.0002463
```

Con los valores de las variables explciativas se calculan el pronóstico y su varianza utilizando los valores de la intervención.

```
F30 <- c(1, 6.14705, 4.38513) #Variables explicativas en t=30. El 1 es para
                              #agregar el intercepto
S29 <- res_dlm$St[[10]] # Estimación de V en T=29
n29 <- 29.5 # Grados de libertad
#C. Pronóstico a un periodo.
f30 <- as.numeric(t(F30) %*% a30_int)
Q30 <- as.numeric(t(F30) %*% R30_int %*% F30 + S29)
f30
```

```
## [1] 10.19
```

```
Q30
```

```
## [1] 0.004455
```

Intervalo del pronóstico. En este caso las distribuciones son t de Student.

```
c(qst(0.025, nu = n29, mu = f30, sigma = sqrt(Q30)),
  qst(0.975, nu = n29, mu = f30, sigma = sqrt(Q30)))
```

```
## [1] 10.06 10.33
```

Una vez que se observa el valor real, se calcula la distribución posterior en t=30 y se reestima el valor de la varianza de observación desconocida.

```
#Valor observado de Y30:
Y30 <- 10.05563427
#D. Posterior en t=20
A30 <- R30_int %*% F30 / Q30
e30 <- Y30-f30
m30 <- a30_int + A30 %*% e30
n30 <- n29 + 1
S30 <- S29 + (S29/n30)*(e30^2/Q30-1)
C30 <- (S30/S29)*(R30_int-A30 %*% t(A30) * Q30)
m30
```

```
##         [,1]
## [1,]  1.5169
## [2,]  1.8813
## [3,] -0.6897
```

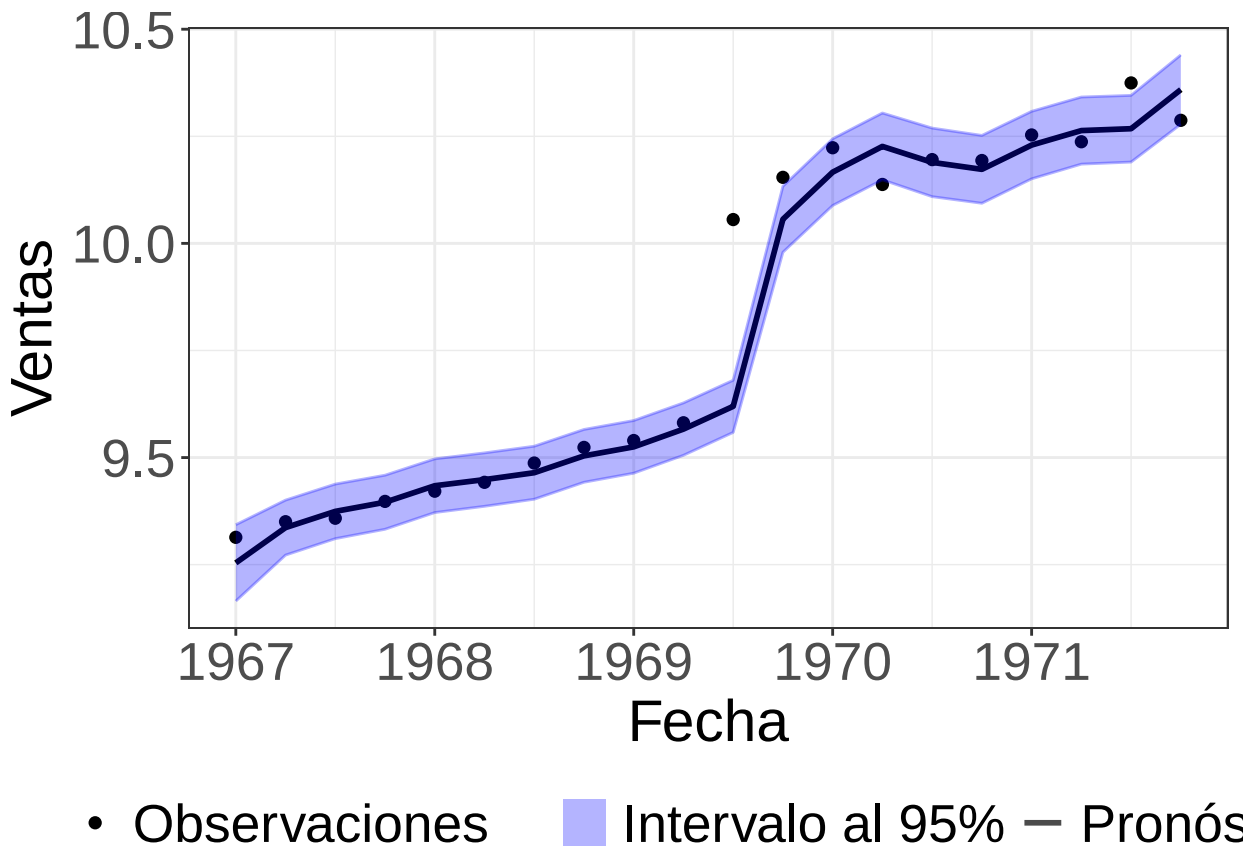
```
C30
```

```
##             [,1]       [,2]       [,3]
## [1,]  1.248e-04  5.178e-06 -3.569e-05
## [2,]  5.178e-06  1.315e-04 -1.842e-04
## [3,] -3.569e-05 -1.842e-04  2.667e-04
```

Se grafican los resultados del modelo no intervenido

```r
df_graficas_no_int <- data.frame("fecha" = datos_ej %>% row.names(),
                                 "y_real" = datos_ej$y,
                                 "y_pronostico" = res_dlm_no_int$ft %>% unlist(),
                                 "CI_inf" = res_dlm_no_int$CI_inf %>% unlist(),
                                  "CI_sup" = res_dlm_no_int$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))
```

```r
ggplot(data = df_graficas_no_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('Ventas') +
  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))
```




```
ggsave(filename = "graphs/teoria/intervencion/actualizacion_no_interv.png", width = 11.7, height = 6)
```

Distribuciones filtradas

Se define una función para calcular distribuciones filtradas más atrás de un periodo pasado cuando hubo intervenciones.

```
#Se pasan manualmente las G historicas en G, las de intervencion se calculan
suavizamiento_V_desc <- function(datos, G, nt, lista_interv){

  error_varianza <- function(x) {
  if(any (diag(x)<0)) stop ('Elementos del parametro de escala o varianzas deben ser positivos')
  }

  lista_ft_k_filt <- list()
  lista_at_k_filt <- list()
  lista_Rt_k_filt <- list()
  lista_resp_med_esc <- list()
  lista_CI_inf <- list()
  lista_CI_sup <- list()

  #Se calculan las distribuciones filtradas para k = T
  at_k_filt_mas_1 <- res_dlm$mt[[length(datos$y)]]
  Rt_k_filt_mas_1 <- res_dlm$Ct[[length(datos$y)]]
  St <- res_dlm$St[[length(datos$y)]]
```

```
Ft <- as.numeric(datos[length(datos$y), 2:4])
lista_ft_k_filt[[length(datos$y)]] <- t(Ft) %*% at_k_filt_mas_1
lista_at_k_filt[[length(datos$y)]] <- at_k_filt_mas_1
lista_Rt_k_filt[[length(datos$y)]] <- Rt_k_filt_mas_1
lista_resp_med_esc[[length(datos$y)]] <- t(Ft) %*% Rt_k_filt_mas_1 %*% Ft
lista_CI_inf[[length(datos$y)]] <- qst(0.025, nu = nt, mu = lista_ft_k_filt[[length(datos$y)]],
                                       sigma = sqrt(lista_resp_med_esc[[length(datos$y)]]))
lista_CI_sup[[length(datos$y)]] <- qst(0.975, nu = nt, mu = lista_ft_k_filt[[length(datos$y)]],
                                       sigma = sqrt(lista_resp_med_esc[[length(datos$y)]]))
interv <- F

error_varianza(Rt_k_filt_mas_1)
error_varianza(St)

for(i in length(datos$y):2){
  #Se calculan las distribuciones filtradas para i-1. De lo mas reciente a lo mas viejo

  if(i %in% lista_interv$t_int){
    at_int <- lista_interv$at_int[[match(i,list_interv$t_int)]]
    Rt_int <- lista_interv$Rt_int[[match(i,list_interv$t_int)]]
    Rt<- res_dlm_no_int$Rt[[i]]
    Ut <- t(chol(Rt_int))
    Zt <- t(chol(Rt))
    Kt <- Ut %*% solve(Zt)
    Gt_int <- Kt %*% G
    interv <- T
  }

  Ct_k <- res_dlm$Ct[[i-1]]
  Rt_k_mas_1 <- res_dlm$Rt[[i]]
  mt_k <- res_dlm$mt[[i-1]]
  at_k_mas_1 <- res_dlm$at[[i]]
  St_k_mas_1 <- res_dlm$St[[i]]
  St_k <- res_dlm$St[[i-1]]

  Bt_k <- Ct_k %*% t(`if`(interv,Gt_int, G)) %*% solve(Rt_k_mas_1)
  at_k_filt <- mt_k + Bt_k %*% (at_k_filt_mas_1 - at_k_mas_1)
  Rt_k_filt <- Ct_k + Bt_k %*% (Rt_k_filt_mas_1 - Rt_k_mas_1) %*% t(Bt_k)
  params_esc <- (St/St_k)*Rt_k_filt

  Ft_k <- as.numeric(datos[i-1, 2:4])
  ft_k_filt <- t(Ft_k) %*% at_k_filt
  resp_med_esc <- (St/St_k)*(t(Ft_k) %*% Rt_k_filt %*% Ft_k)
  CI <- c(qst(0.025, nu = nt, mu = ft_k_filt, sigma = sqrt(resp_med_esc)),
          qst(0.975, nu = nt, mu = ft_k_filt, sigma = sqrt(resp_med_esc)))

  error_varianza(Rt_k_mas_1)
  error_varianza(Ct_k)
  error_varianza(St_k_mas_1)
  error_varianza(St_k)
  error_varianza(Rt_k_filt)

  lista_ft_k_filt[[i-1]] <- ft_k_filt
```

```
    lista_at_k_filt[[i-1]] <- at_k_filt
    lista_Rt_k_filt[[i-1]] <- Rt_k_filt
    lista_resp_med_esc[[i-1]] <- resp_med_esc
    lista_CI_inf[[i-1]] <- CI[1]
    lista_CI_sup[[i-1]] <- CI[2]

    at_k_filt_mas_1 <- at_k_filt
    Rt_k_filt_mas_1 <- Rt_k_filt
    interv <- F
  }

  return(list("ft_k_filt" = lista_ft_k_filt, "at_k_filt" = lista_at_k_filt,
              "Rt_k_filt" = lista_Rt_k_filt, "resp_media_esc" = lista_resp_med_esc,
              "CI_inf" = lista_CI_inf, "CI_sup" = lista_CI_sup))
}
```

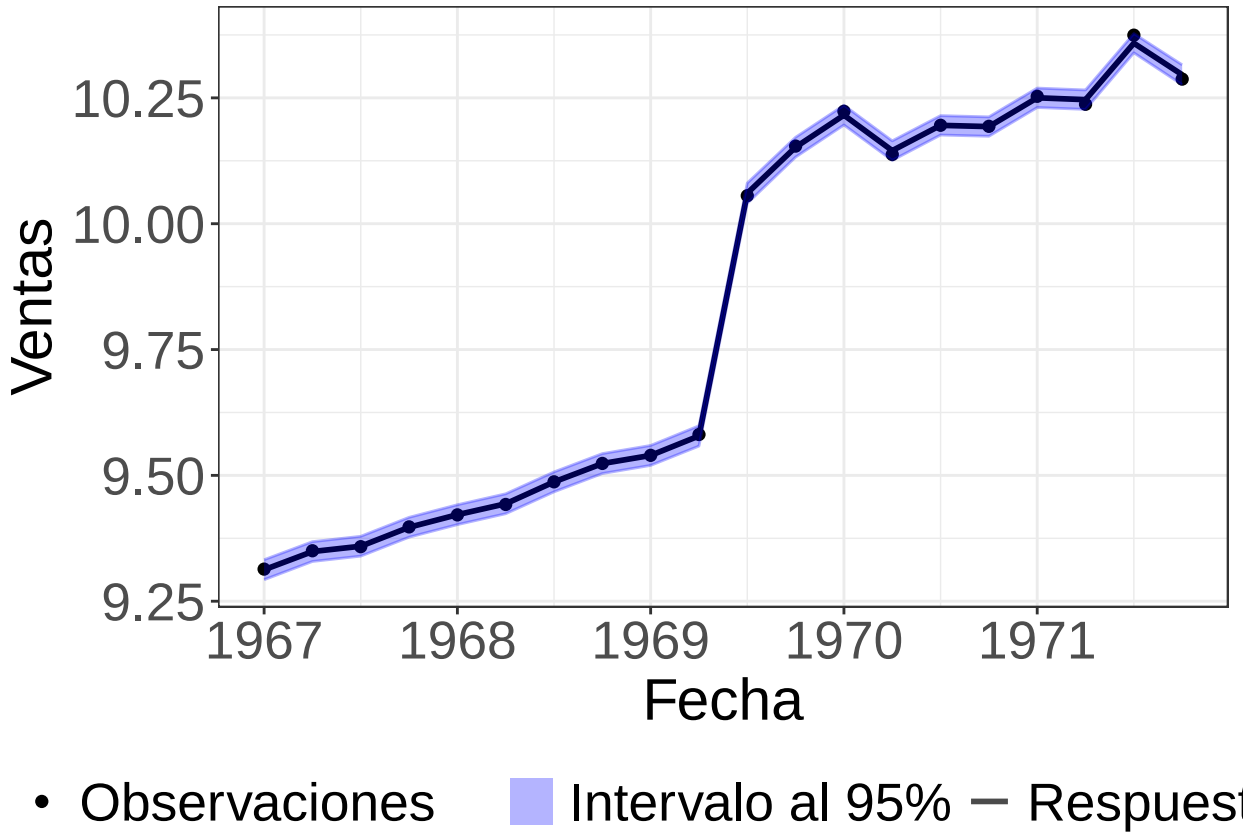
Se aplica la función con las intervenciones definidas anteriormente.

```
res_dlm_suav <- suavizamiento_V_desc(datos_ej, G20, 39.5, list_interv)
```

Se grafican los resultados.

```
df_graficas_suav <- data.frame("fecha" = datos_ej %>% row.names(),
                               "y_real" = datos_ej$y,
                               "resp_media" = res_dlm_suav$ft_k_filt %>% unlist(),
                               "CI_inf" = res_dlm_suav$CI_inf %>% unlist(),
                               "CI_sup" = res_dlm_suav$CI_sup %>% unlist()) %>%
  mutate(fecha = as.numeric(fecha))
```

```
ggplot(data = df_graficas_suav, aes(x = fecha)) +
  geom_point(aes(y = y_real, shape = "Observaciones"), size = 2) +
  geom_line(aes(y = resp_media, color = 'Respuesta media'), 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",
                          "Respuesta media" = "black")) +
  scale_fill_manual(
    name = "",  values = c("Intervalo al 95%" = "blue",
                           "Respuesta media" = "transparent")) +
  theme(legend.position = "bottom") +
  labs(shape = "") +
  ylab('Ventas') +
  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))
```




```
ggsave(filename = "graphs/teoria/intervencion/suavizamiento_interv.png", width = 11.7, height = 6)
```

Se muestra paso por paso cómo calcular las distribuciones filtradas cuando hubo una intervención. Se supone que pasaron los años y nos encontramos en $t = 39$. El objetivo es obtener las distribuciones filtradas de los estados en $t = 29$ cuando hubo una intervención en $t = 30$ dada la información en $t = 39$.

Primero hay que obtener G_{30}^* y W_{30}^* para que la ecuación de evolución coincida con la intervención.

```
G30 <-matrix(c(1.001, 0, 0, 0, 1, 0, 0, 0, 1), ncol=3)
W30 <-matrix(c(0.00001, 0, 0, 0, 0.00001,-0.00001, 0,-0.00001, 0.00005), ncol=3)
R30 <- res_dlm_no_int$Rt[[11]]
# La descomposicion de Cholesky en R devuelve una triangular superior. Se transpone
# para hacerla triangular inferior para que coincida con la teoría presentada
U30 <- t(chol(R30_int))
Z30 <- t(chol(R30))
K30 <- U30 %*% solve(Z30)
G30_int <- K30 %*% G30
W30_int <- K30 %*% W30 %*% t(K30)
```

Se aplica el procedimiento usual para obtener las distribuciones filtradas en $t = 29$, pero se usan los valores de **G30_int** y **W30_int**.

```
C29 <- res_dlm$Ct[[10]]
m29 <- res_dlm$mt[[10]]
a39_menos_9 <- res_dlm_suav$at_k_filt[[11]]
R39_menos_9 <- res_dlm_suav$Rt_k_filt[[11]]
```

```
B29 <- C29 %*% t(G30_int) %*% solve(R30_int)
a39_menos_10 <- m29 + B29 %*% (a39_menos_9 - a30_int)
R39_menos_10 <- C29 + B29 %*% (R39_menos_9 - R30_int) %*% t(B29)

a39_menos_10
```

```
##         [,1]
## [1,]  1.5160
## [2,]  1.7960
## [3,] -0.6706
```

```
S39 <- res_dlm$St[[20]] # Estimación de V en T=39

S39/S29 * R39_menos_10
```

```
##            [,1]       [,2]       [,3]
## [1,]  5.839e-04  4.242e-05 -0.0001916
## [2,]  4.242e-05  3.633e-04 -0.0005233
## [3,] -1.916e-04 -5.233e-04  0.0007883
```

Ahora se obtiene la respuesta media en $t = 29$ dada la información en $t = 39$.

```
F29 <- c(1, 6.131, 4.398) #Variables explicativas en t=29.
t(F29) %*% a39_menos_10
```

```
##       [,1]
## [1,] 9.578
```

```
S39/S29 * t(F29) %*% R39_menos_10 %*% F29
```

```
##           [,1]
## [1,] 9.959e-05
```