

Actualizacion de ecuaciones univariada

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

```
#La base de datos freeny tiene la siguiente estructura:  
head(freeny)
```

```
##           y lag.quarterly.revenue price.index income.level market.potential  
## 1962.25 8.792                8.796        4.710         5.821         12.97  
## 1962.5  8.791                8.792        4.702         5.826         12.97  
## 1962.75 8.815                8.791        4.689         5.831         12.98  
## 1963    8.813                8.815        4.686         5.840         12.98  
## 1963.25 8.908                8.813        4.640         5.850         12.98  
## 1963.5  8.937                8.908        4.626         5.865         12.99
```

Actualización de ecuaciones

Se parte de la distribución posterior en $t = 19$ y se obtiene la distribución a priori 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)  
  
#B. Priori de parámetros en t=20  
a20 <- G20 %*% m19  
R20 <- G20 %*% C19 %*% t(G20) + W20  
a20
```

```
##           [,1]  
## [1,]  1.501  
## [2,]  1.800  
## [3,] -0.700
```

```
R20
```

```
##           [,1]      [,2]      [,3]  
## [1,] 3.004e-05 1.001e-05 -2.002e-05  
## [2,] 1.001e-05 4.000e-05 -2.000e-05  
## [3,] -2.002e-05 -2.000e-05 7.000e-05
```

Con los valores de las variables explicativas se calculan el pronóstico y su varianza.

```
freeny[20,]
```

```
##          y lag.quarterly.revenue price.index income.level market.potential
## 1967 9.314                9.284         4.51         6.061         13.07
```

```
F20 <- c(1, 6.06093, 4.51018) #Variables explicativas en t=20. El 1 es para
                                #agregar el intercepto
```

```
V20 <- 0.00005
```

```
#C. Pronóstico a un periodo.
```

```
f20 <- t(F20) %*% a20
```

```
Q20 <- t(F20) %*% R20 %*% F20 + V20
```

```
f20
```

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

```
## [1,] 9.254
```

```
Q20
```

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

```
## [1,] 0.001821
```

Intervalo del pronóstico.

```
c(qnorm(0.025, mean = f20, sd = sqrt(Q20)), qnorm(0.975, mean = f20,
```

```
## [1] 9.170 9.338
```

Una vez que se observa el valor real, se calcula la distribución posterior en $t=20$.

```
#Valor observado de Y20:
```

```
Y20 <- 9.31378
```

```
#D. Posterior en t=20
```

```
A20 <- R20 %*% F20 %*% solve(Q20)
```

```
e20 <- Y20-f20
```

```
m20 <- a20 + A20 %*% e20
```

```
C20 <- R20 - A20 %*% Q20 %*% t(A20)
```

```
m20
```

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

```
## [1,] 1.5015
```

```
## [2,] 1.8053
```

```
## [3,] -0.6943
```

```
C20
```

```
##          [,1]      [,2]      [,3]
```

```
## [1,] 3.004e-05 9.973e-06 -2.006e-05
```

```
## [2,] 9.973e-06 2.554e-05 -3.555e-05
```

```
## [3,] -2.006e-05 -3.555e-05 5.328e-05
```

Función para actualizar más de un periodo

```
# Se acomodan los datos en la estructura necesaria
datos_ej <- freeny %>%
  mutate(intercept = 1) %>%
  slice(20:n()) %>%
  dplyr::select(y, intercept, income.level, price.index)

#Se asume que Gt, Vt y Wt son ctes conocidas para toda t.
actualizacion <- function(datos, m0, C0, G, W, V){
  mt_menos_1 <- m0
  Ct_menos_1 <- C0
  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()
  for(t in 1:length(datos$y)){

    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 + V
    CI <- c(qnorm(0.025, mean = ft, sd = sqrt(Qt)), qnorm(0.975, mean = ft,
                                                         sd = 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
    Ct <- Rt - At %**% Qt %**% t(At)
    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
    mt_menos_1 <- mt
    Ct_menos_1 <- Ct
  }
  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))
}
```

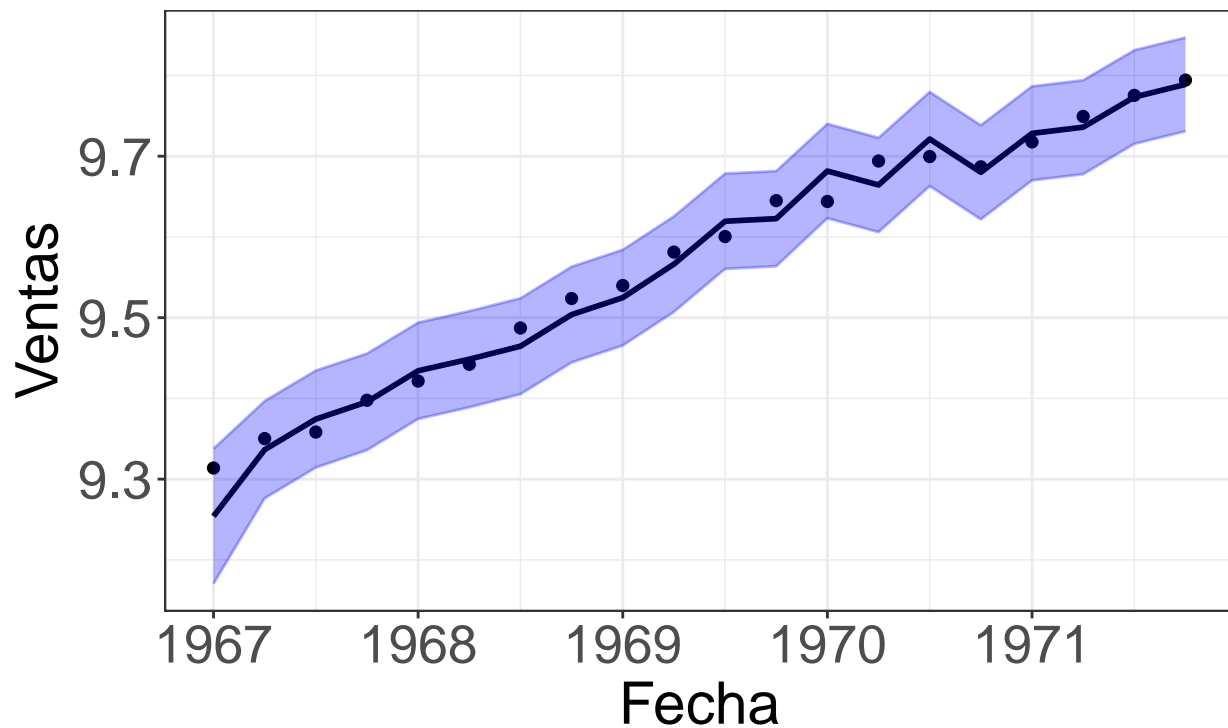
Se aplica la función.

```
res_dlm <- actualizacion(datos_ej, m19, C19, G20, W20, V20)
```

Se grafican los pronósticos a un paso y sus intervalos.

```
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))
```



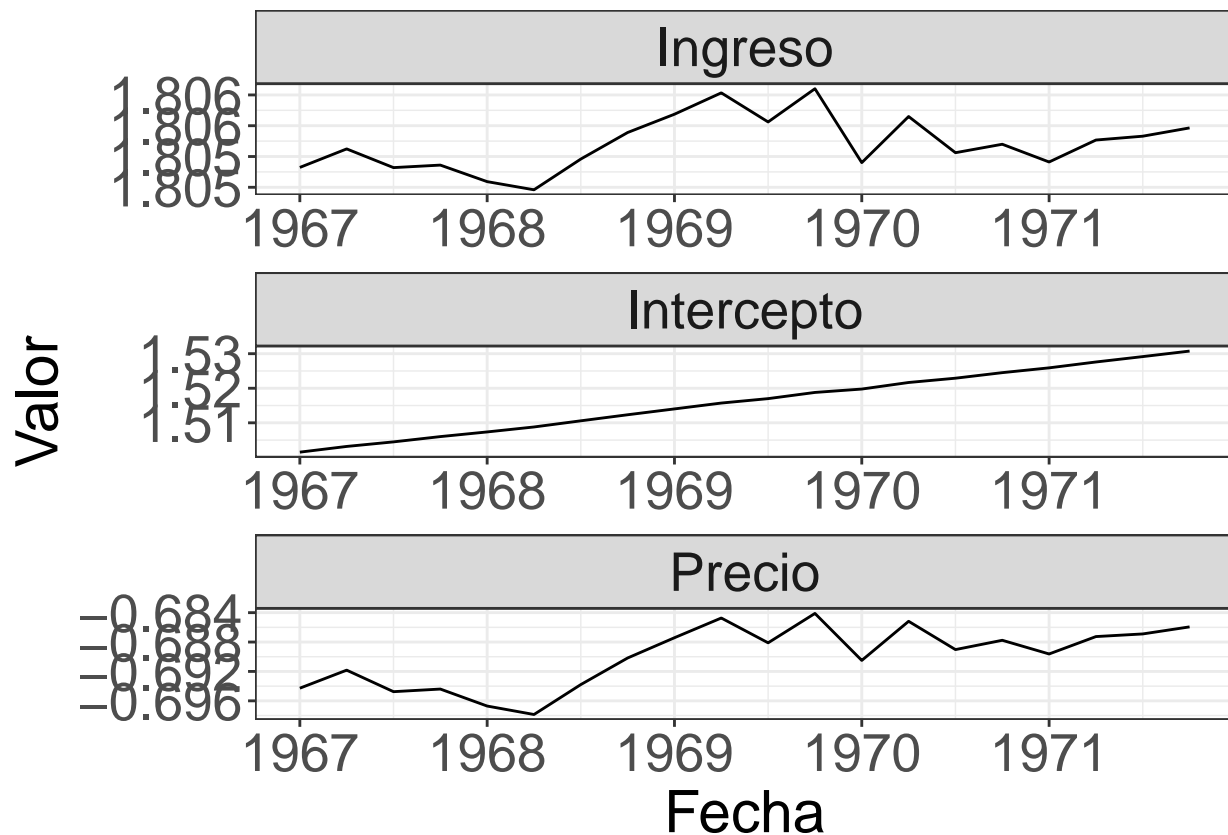
- Observaciones Intervalo al 95% — Pronóstico

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

Se grafican la evolución de los parámetros.

```
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)
```

Pronósticos a más de un paso.

```
freeny[21,]
```

```
##          y lag. quarterly.revenue price.index income.level market.potential
## 1967.25 9.35                9.314      4.504      6.071      13.07
```

```
#Valores iniciales.
#m20 y C20 se definieron en el ejemplo pasado
a20_0 <- m20
R20_0 <- C20

#k = 1 (pronóstico a un paso)
#Valores conocidos de F21, G21, V21 y W21
F21 <- c(1, 6.07103, 4.50352)
G21 <- matrix(c(1.001, 0, 0, 0, 1, 0, 0, 0, 1), ncol=3)
V21 <- 0.00005
W21 <- matrix(c(0.00001, 0, 0, 0, 0.00001, -0.00001, 0, -0.00001, 0.00005), ncol=3)

#Distribución de estados en t=21
a20_1 <- G21 %*% a20_0
R20_1 <- G21 %*% R20_0 %*% t(G21) + W21
```

```
#Distribución de pronóstico de Y21
f20_1 <- t(F21) %*% a20_1
Q20_1 <- t(F21) %*% R20_1 %*% F21 + V21
f20_1
```

```
##           [,1]
## [1,] 9.336
```

```
Q20_1
```

```
##           [,1]
## [1,] 0.0009445
```

```
freeny[22,]
```

```
##           y lag.quarterly.revenue price.index income.level market.potential
## 1967.5 9.358           9.35      4.494           6.08           13.08
```

```
#k = 2 (pronóstico a dos pasos)
#Valores conocidos de F22, G22, V22 y W22
F22 <- c(1, 6.08018, 4.4936)
G22 <- matrix(c(1.001, 0, 0, 0, 1, 0, 0, 0, 1), ncol=3)
V22 <- 0.00005
W22 <- matrix(c(0.00001, 0, 0, 0, 0.00001, -0.00001, 0, -0.00001, 0.00005), ncol=3)
```

```
#Distribución de estados en t=22
a20_2 <- G22 %*% a20_1
R20_2 <- G22 %*% R20_1 %*% t(G22) + W22
```

```
#Distribución de pronóstico de Y22
f20_2 <- t(F22) %*% a20_2
Q20_2 <- t(F22) %*% R20_2 %*% F22 + V22
f20_2
```

```
##           [,1]
## [1,] 9.361
```

```
Q20_2
```

```
##           [,1]
## [1,] 0.001784
```

```
#Covarianzas
C20_1_1 <- R20_1
C20_2_1 <- G22 %*% C20_1_1 #Covarianza entre theta22 y theta 21
cov_Y22_Y21 <- t(F22) %*% C20_2_1 %*% F21
cov_Y22_Y21
```

```
##           [,1]
## [1,] 0.000893
```

```
a20_1
```

```
##           [,1]
## [1,]  1.5030
## [2,]  1.8053
## [3,] -0.6943
```

```
R20_1
```

```
##           [,1]      [,2]      [,3]
## [1,]  4.010e-05  9.983e-06 -2.008e-05
## [2,]  9.983e-06  3.554e-05 -4.555e-05
## [3,] -2.008e-05 -4.555e-05  1.033e-04
```

```
a20_2
```

```
##           [,1]
## [1,]  1.5045
## [2,]  1.8053
## [3,] -0.6943
```

```
R20_2
```

```
##           [,1]      [,2]      [,3]
## [1,]  5.018e-05  9.993e-06 -2.010e-05
## [2,]  9.993e-06  4.554e-05 -5.555e-05
## [3,] -2.010e-05 -5.555e-05  1.533e-04
```

```
datos_ej_prons <- datos_ej[-1,]
```

```
# Se define una función para calcular los pronósticos a más de un paso
# Los datos deben empezar en el primer periodo futuro (no observado)
```

```
pronosticos <- function(datos, at_0, Rt_0, G, W, V){
  at_k <- at_0
  Rt_k <- Rt_0
  lista_at_k <- list()
  lista_Rt_k <- list()
  lista_ft_k <- list()
  lista_Qt_k <- list()
  lista_CI_inf <- list()
  lista_CI_sup <- list()
  for(k in 1:length(datos$y)){

    at_k <- G %*% at_k
    Rt_k <- G %*% Rt_k %*% t(G) + W
    Ft <- as.numeric(datos[k, 2:4])
    ft_k <- t(Ft) %*% at_k
    Qt_k <- t(Ft) %*% Rt_k %*% Ft + V
    CI <- c(qnorm(0.025, mean = ft_k, sd = sqrt(Qt_k)),
            qnorm(0.975, mean = ft_k, sd = sqrt(Qt_k)))
```



```

CI_inf <- CI[1]
CI_sup <- CI[2]
lista_at_k[[k]] <- at_k
lista_Rt_k[[k]] <- Rt_k
lista_ft_k[[k]] <- ft_k
lista_Qt_k[[k]] <- Qt_k
lista_CI_inf[[k]] <- CI_inf
lista_CI_sup[[k]] <- CI_sup

}
return(list("at_k" = lista_at_k, "Rt_k" = lista_Rt_k, "ft_k" = lista_ft_k,
           "Qt_k" = lista_Qt_k, "CI_inf" = lista_CI_inf, "CI_sup" = lista_CI_sup))
}

```

Se aplica la función.

```

pronos_dlm <- pronosticos(datos_ej_prons, m20, C20, G20, W20, V20)

```

Se grafican los resultados.

```

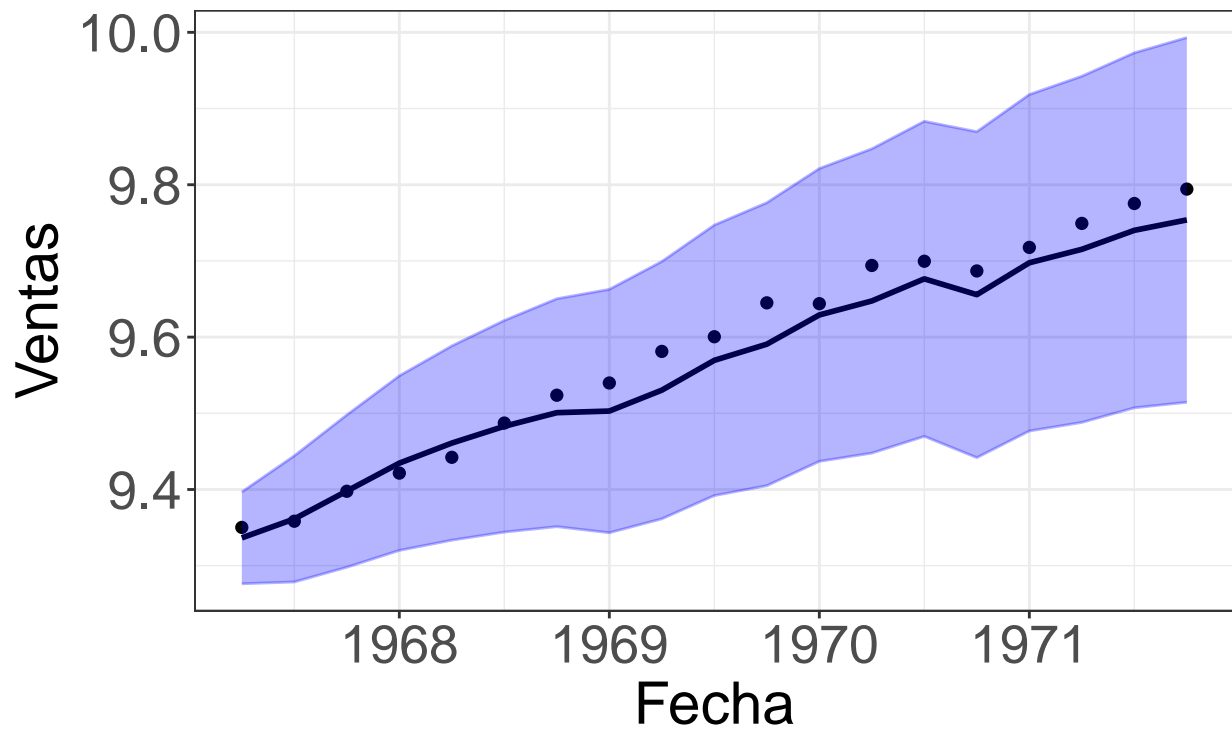
df_pronos <- data.frame("fecha" = datos_ej_prons %>% row.names(), "y_real" = datos_ej_prons$y,
                        "y_pronostico" = pronos_dlm$ft_k %>% unlist(),
                        "CI_inf" = pronos_dlm$CI_inf %>% unlist(),
                        "CI_sup" = pronos_dlm$CI_sup %>% unlist()) %>%
mutate(fecha = as.numeric(fecha))

```

```

ggplot(data = df_pronos, 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))

```



- Observaciones
- Intervalo al 95% — Pronós

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