

# 2do Trabajo Práctico de Econometría

Max Murmis y Juan Camilo Gutman

En el presente trabajo nos propusimos analizar la inflación...

## Carga y limpieza de la base de datos

Utilizamos las siguientes librerías:

```
library(tidyverse) #Para manejar bases de datos
library(tsibble) #Para manejar bases para series de tiempo
library(fable) #Para correr modelos de series de tiempo
library(ggplot2) #Para graficar
library(knitr) #Para tablas
library(fpp3)
```

Utilizaremos una base del CEPED, disponible en su portal de estadísticas CEPED.data, en el siguiente enlace [ceped-data.shinyapps.io/ceped-data/](http://ceped-data.shinyapps.io/ceped-data/)

```
ipc <- readRDS("base_ipc.RDS")
```

```
# Pequeña exploración
colnames(ipc)
```

```
[1] "AN04"          "valor"         "sub"           "cod.variable" "var"
```

```
unique(ipc$cod.variable)
```

```
[1] "IPC_Anual_2017"      "IPC_Mensual_2006"  "Ondas_EPH_2017"
[4] "Trimestres_EPH_2017"
```

```
# Para trabajar con valores mensuales, desde 2003:
ipc <- ipc %>%
  filter(cod.variable == "IPC_Mensual_2006") %>%
  filter(ANO4 >= 2003)

# Formato fechas:
ipc <- ipc %>%
  mutate(periodo = yearmonth(make_date(ANO4, sub)))

# Nos quedamos con las columnas de interés
ipc <- ipc %>%
  select(c("periodo", "valor"))
```

En el Anexo (agregar link) se describe la construcción de dicha base empalmada. En Table 1 puede verse el head del dataframe.

Table 1: Base CEPED.data

periodo	valor
2003 ene.	80.97979
2003 feb.	81.43816
2003 mar.	81.91355
2003 abr.	81.95881
2003 may.	81.64447
2003 jun.	81.57452
2003 jul.	81.93679
2003 ago.	81.95674
2003 sep.	81.98922
2003 oct.	82.47252
2003 nov.	82.67591
2003 dic.	82.85145
2004 ene.	83.19972
2004 feb.	83.28342
2004 mar.	83.77755

Para modelar la inflación mensual ( $\pi_t$ ) realizaremos el siguiente cómputo:

$$\pi_t = \ln(ipc_t) - \ln(ipc_{t-1})$$

Table 2: Inflación 2003-2023

periodo	inflacion
2003 ene.	NA
2003 feb.	0.0056443
2003 mar.	0.0058205
2003 abr.	0.0005523
2003 may.	-0.0038427
2003 jun.	-0.0008571
2003 jul.	0.0044311
2003 ago.	0.0002435
2003 sep.	0.0003962
2003 oct.	0.0058774
2003 nov.	0.0024632
2003 dic.	0.0021209
2004 ene.	0.0041948
2004 feb.	0.0010054
2004 mar.	0.0059156

Vamos a separar los últimos 6 meses para realizar pronóstico.

```
infl <- infl %>% as_tsibble(index = periodo)
```

```
# Simple gráfico de la evolución en el tiempo de la inflación
autoplot(infl, inflacion) +
  labs(title = "Inflación mensual ",
        subtitle = "2003-2023",
        y = "Inflación")
```

Warning: Removed 1 row containing missing values or values outside the scale range (`geom\_line()`).

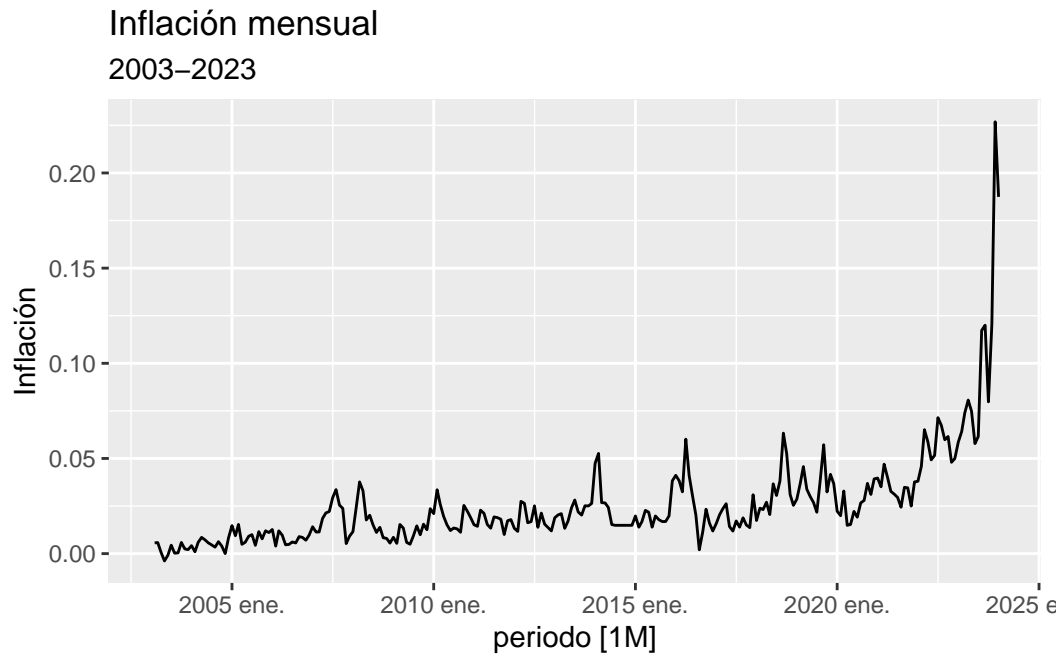
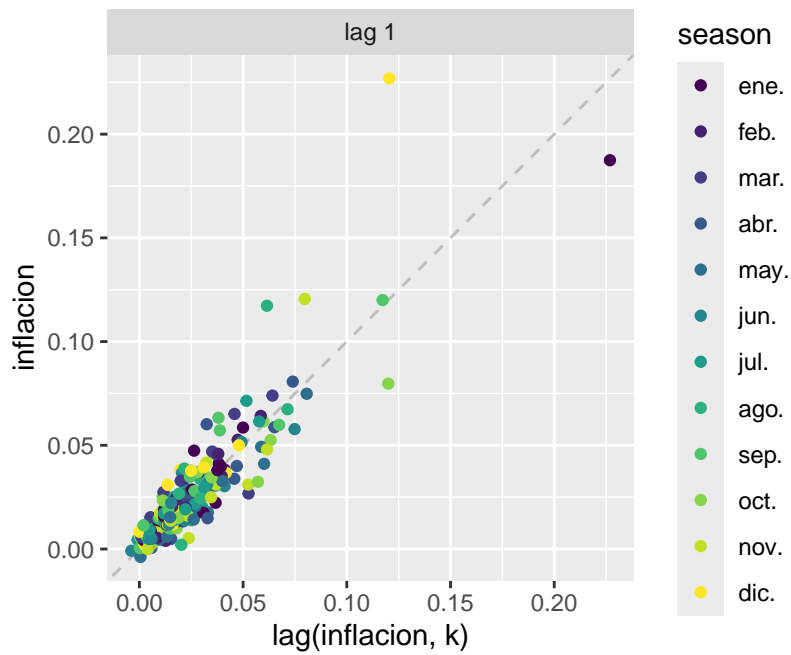


Gráfico de dispersión entre la inflación corriente y la del periodo inmediatamente anterior (esto es, si no me equivoco, un gráfico que ilustra la autocorrelación con un solo rezago:

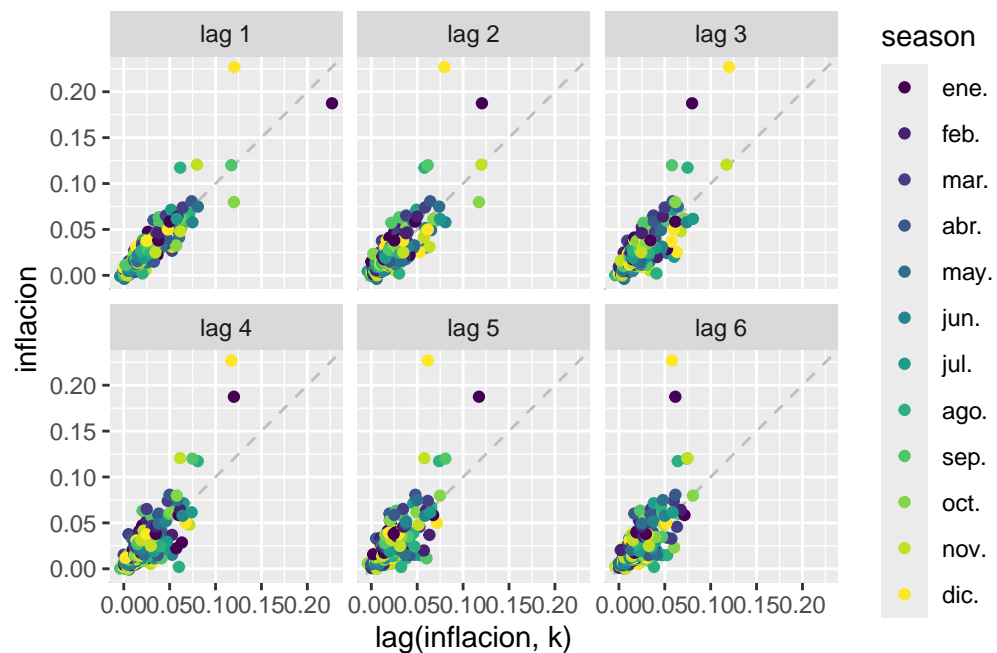
```
infl %>% gg_lag(  
  inflacion, geom = "point", lags = 1) +  
  labs(x = "lag(inflacion, k)")
```

Warning: Removed 1 rows containing missing values (gg\_lag).



```
infl %>% gg_lag(
  inflacion, geom = "point", lags = 1:6) +
  labs(x = "lag(inflacion, k)")
```

Warning: Removed 1 rows containing missing values (gg\_lag).



ACF (autocorrelation function):

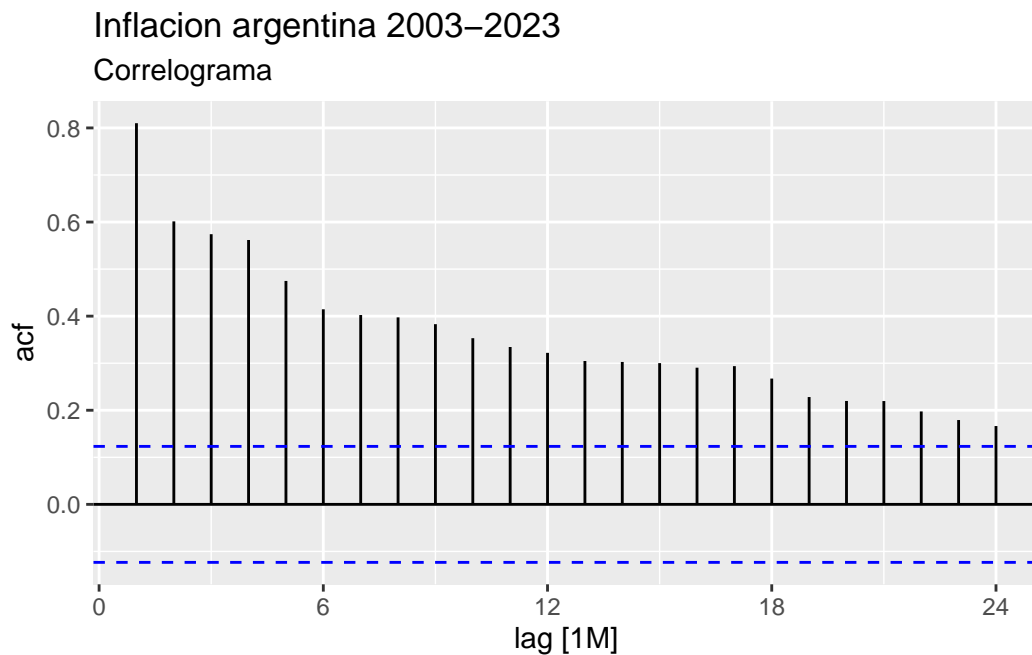
```
infl %>% ACF(inflacion, lag_max = 9) %>%
  knitr::kable()
```

lag	acf
1M	0.8100238
2M	0.6014314
3M	0.5741591
4M	0.5618177
5M	0.4748505
6M	0.4145770
7M	0.4023700
8M	0.3974182
9M	0.3829822

Se observa fácil como decae la autocorrelación con cada mes que nos alejamos...

¿Decae exponencialmente?

```
infl |>
  ACF(inflacion) |>
  autoplot() + labs(title= "Inflacion argentina 2003-2023",
                    subtitle= "Correlograma")
```



Acá el gráfico de la inflación sin la función autoplot

```
p_static <- ggplot(infl, aes(x = periodo, y = inflacion)) +
  geom_line(color = "blue") +
  labs(title = "Inflation in Argentina",
       x = "Date",
       y = "Inflation (%)") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title.x = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    axis.text = element_text(size = 10)
  )

print(p_static)
```

Warning: Removed 1 row containing missing values or values outside the scale range

```
(`geom_line()`).
```

