

Parcial 1

Julián Úsuga Ortiz - Ivan Santiago Rojas Martinez

1.

Sea X_t el proceso estocástico dado por:

$$X_t = w_{t-2} + 0.5w_{t-1} + 2w_t + 0.5w_{t+1} + w_{t+2}$$

donde los w_t son independientes con media 0 y varianza $\sigma_w^2 = 4.8$.

2.

Sea X_t el proceso estocástico estacionario dado por:

$$X_t = 3.1 + 0.9X_{t-1} - 0.6X_{t-2} + w_t$$

donde w_t es ruido blanco gaussiano con media 0 y varianza $\sigma_w^2 = 6.2$.

a).

$$X_t = 3.1 + 0.9X_{t-1} - 0.6X_{t-2} + w_t$$

$$\mu_t = 3.1 + 0.9\mu_t - 0.6\mu_t$$

$$\mu_t - 0.9\mu_t + 0.6\mu_t = 3.1$$

$$(1 - 0.9 + 0.6)\mu_t = 3.1$$

$$\mu_t = \frac{3.1}{1 - 0.9 + 0.6} = 4.4285$$

b).

$$\begin{aligned} \text{var}[X_t] &= \text{cov}[X_t, X_t] = \text{cov}[X_t, 3.1 + 0.9X_{t-1} - 0.6X_{t-2} + w_t] \\ &= 0.9\text{cov}[X_t, X_{t-1}] + 0.6\text{cov}[X_t, X_{t-2}] + \text{cov}[X_t, w_t] \\ &= 0.9\gamma(-1) - 0.6\gamma(-2) + \sigma_w^2 \end{aligned}$$

Como $\text{var}[X_t] = \gamma(0) = \sigma_x^2$

$$\begin{aligned} &= 0.9\gamma(-1) - 0.6\gamma(-2) \\ \gamma(0) &= 0.9\gamma(-1) - 0.6\gamma(-2) \end{aligned}$$

Dividiendo por $\gamma(0)$

$$\begin{aligned} 1 &= 0.9 \frac{\gamma(-1)}{\gamma(0)} - 0.6 \frac{\gamma(-2)}{\gamma(0)} + \frac{\sigma_w^2}{\gamma(0)} \\ 1 &= 0.9\rho(1) - 0.6\rho(2) + \frac{\sigma_w^2}{\gamma(0)} \\ 1 - 0.9\rho(1) + 0.6\rho(2) &= \frac{\sigma_w^2}{\rho(0)} \\ \sigma_x^2 = \gamma(0) &= \frac{\sigma_w^2}{1 - 0.9\rho(1) + 0.6\rho(2)} \\ \gamma(1) &= \text{cov}[X_{t-1}, 3.1 + 0.9X_{t-1} - 0.6X_{t-2} + w_t] \\ &= \text{cov}[X_{t-1}, X_{t-1}]0.9 - 0.6\text{cov}[X_{t-1}, X_{t-2}] \\ &= \gamma(0)0.9 - 0.6\gamma(1) \\ 1.6\gamma(1) &= \gamma(0) * 0.9 \\ \gamma(1) &= \gamma(0) \frac{0.9}{1.6} \\ \rho(1) &= \frac{\gamma(1)}{\gamma(0)} = \frac{0.9}{1.6} \\ \gamma(2) &= \text{cov}[X_{t-2}, 3.1 + 0.9X_{t-1} - 0.6X_{t-2} + w_t] \\ &= \text{cov}[X_{t-2}, X_{t-1}]0.9 - 0.6\text{cov}[X_{t-2}, X_{t-2}] \\ &= \gamma(1)0.9 - 0.6\gamma(0) \\ &= \gamma(0) \frac{0.9^2}{1.6} - 0.6\gamma(0) \\ \rho(2) &= \frac{\gamma(2)}{\gamma(0)} = \frac{0.9^2}{1.6} - 0.6 = -0.09375 \end{aligned}$$

Por lo que:

$$\gamma(0) = \frac{6.2}{1 - \frac{0.9^2}{1.6} + 0.6(-0.09375)}$$
$$\gamma(0) = \sigma_x^2 \approx 14.1714$$

3.

Paquetes usados:

```
library(readxl)
library(tidyverse)
library(lubridate)
library(knitr)
library(xtable)
```

Lectura de la Base de datos:

```
data_19 <- read_excel("data/Afluencia_Metro_2019.xlsx",
                      range = "A4:W3566",
                      col_names = FALSE)
data_20 <- read_excel("data/Afluencia_Metro.xlsx",
                      range = "A4:W3767",
                      col_names = FALSE)
data_21 <- read_excel("data/Afluencia_2021.xlsx",
                      range = "A4:W4125",
                      col_names = FALSE)
```

a).

Las dimensiones de los datos de 2019 son 3563 filas y 23 columnas.

Las dimensiones de los datos de 2020 son 3764 filas y 23 columnas.

Las dimensiones de los datos de 2021 son 4122 filas y 23 columnas.

b).

```
datos_juntos <- bind_rows(data_19, data_20, data_21)
```

Las dimensiones de los datos_juntos son 11449 filas y 23 columnas.

c).

```
data <- pivot_longer(datos_juntos, cols = ends_with(":00"),
                     names_to = "hora", values_to = "per.num")

data$hora <- hm(data$hora)
data$dia <- day(data$fecha)
data$dia.sem <- weekdays(data$fecha)
data$sem <- week(data$fecha)
data$mes <- month(data$fecha)
data$anio <- year(data$fecha)
```

Las dimensiones del data frame son 228980 filas y 10 columnas.

d).

Table 1: Linea A

fecha	linea	per.dia.tot	hora	per.num	dia	dia.sem	sem	mes	anio
2019-01-01	línea a	183664	4H 0M 0S	14	1	Tuesday	1	1	2019
2019-01-01	línea a	183664	5H 0M 0S	4554	1	Tuesday	1	1	2019
2019-01-01	línea a	183664	6H 0M 0S	5696	1	Tuesday	1	1	2019
2019-01-01	línea a	183664	7H 0M 0S	4893	1	Tuesday	1	1	2019
2019-01-01	línea a	183664	8H 0M 0S	4399	1	Tuesday	1	1	2019

Las dimensiones de los datos de la linea A son 21860 filas y 10 columnas.

Table 2: Linea B

fecha	linea	per.dia.tot	hora	per.num	dia	dia.sem	sem	mes	anio
2019-01-01	línea b	25852	4H 0M 0S	NA	1	Tuesday	1	1	2019
2019-01-01	línea b	25852	5H 0M 0S	678	1	Tuesday	1	1	2019

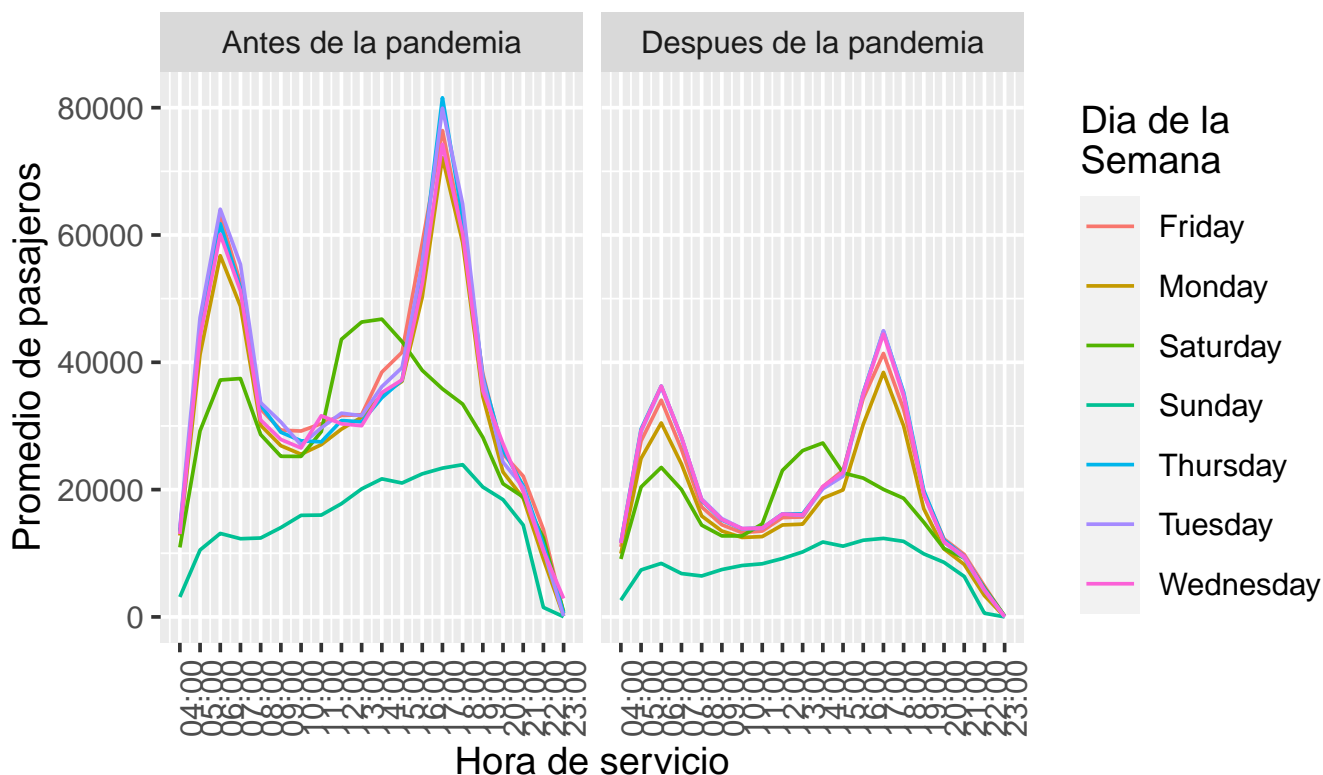
fecha	línea	per.dia.tot	hora	per.num	dia	dia.sem	sem	mes	año
2019-01-01	línea b	25852	6H 0M 0S	833	1	Tuesday	1	1	2019
2019-01-01	línea b	25852	7H 0M 0S	682	1	Tuesday	1	1	2019
2019-01-01	línea b	25852	8H 0M 0S	676	1	Tuesday	1	1	2019

Las dimensiones de los datos de la línea B son 21860 filas y 10 columnas.

e).

El 23 de Marzo de 2020 el Gobierno de Colombia expidió el Decreto 457 para el período de aislamiento preventivo obligatorio a causa del virus del COVID-19.

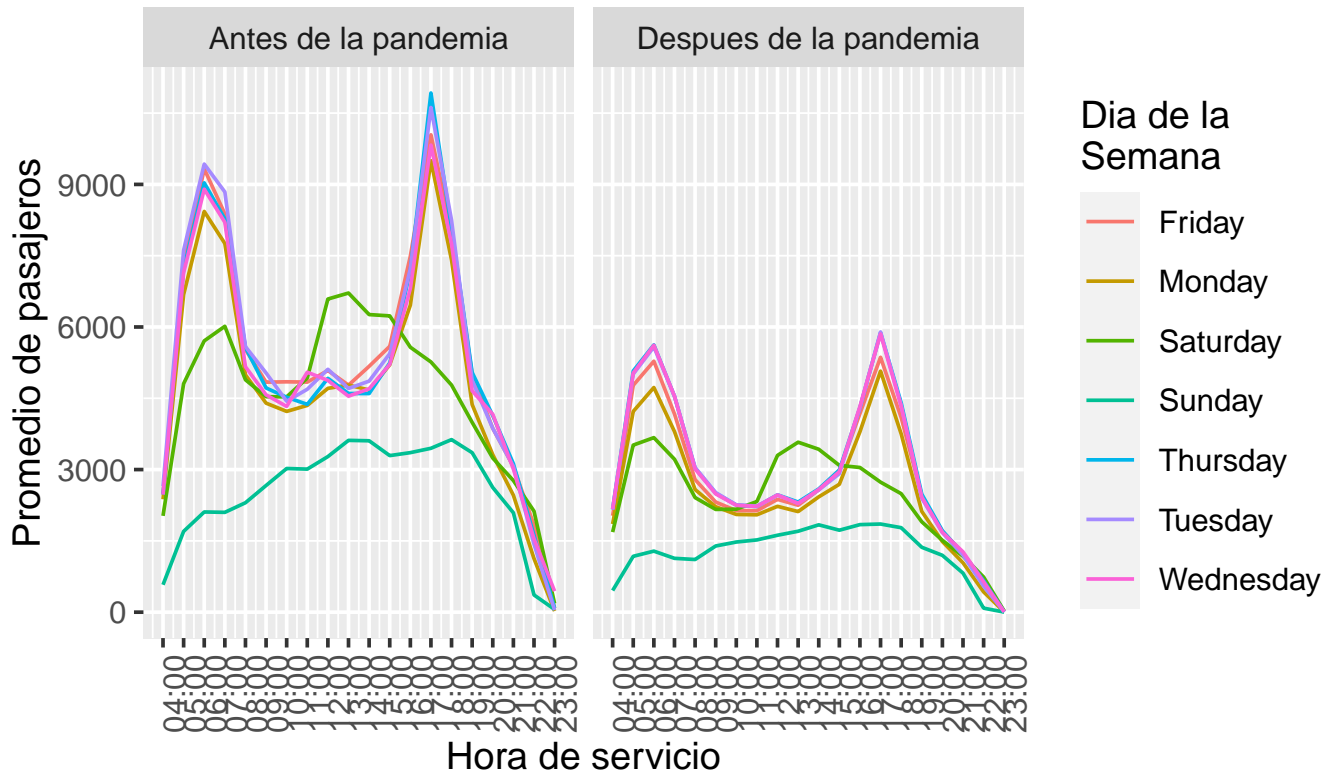
Promedio de pasajeros por hora para la línea A



Se puede observar que después de la pandemia se disminuyó aproximadamente a la mitad la afluencia de pasajeros en la **línea A** y durante las semanas el comportamiento es muy similar para antes de la pandemia y después de la pandemia.

Los días con menor afluencia de pasajeros son los días **sábado** y **domingo**

Promedio de pasajeros por hora para la línea B



Se puede observar que después de la pandemia se disminuyó aproximadamente a la mitad la afluencia de pasajeros en la **línea B** y durante las semanas el comportamiento es muy similar para antes de la pandemia y después de la pandemia. Los días con menor afluencia de pasajeros son los días **sábado** y **domingo**

f).

```
lin.a <- dat_lin_A %>%
  group_by(fecha) %>%
  summarise(dia = day(fecha), dia.sem = weekdays(fecha),
            sem = week(fecha), mes = month(fecha), anio = year(fecha),
            per.dia.tot = sum(per.num, na.rm = TRUE))

lin.b <- dat_lin_B %>%
  group_by(fecha) %>%
  summarise(dia = day(fecha), dia.sem = weekdays(fecha),
```

```
sem = week(fecha), mes = month(fecha), anio = year(fecha),
per.dia.tot = sum(per.num, na.rm = TRUE))
```

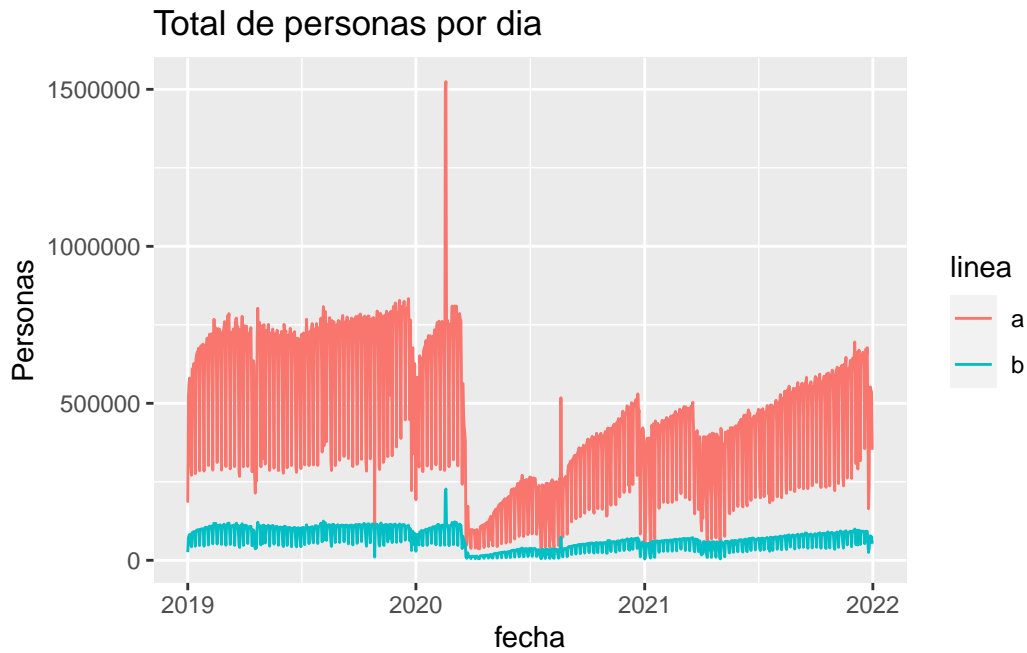
Table 3: Linea A

fecha	dia	dia.sem	sem	mes	anio	per.dia.tot
2019-01-01	1	Tuesday	1	1	2019	183664
2019-01-01	1	Tuesday	1	1	2019	183664
2019-01-01	1	Tuesday	1	1	2019	183664
2019-01-01	1	Tuesday	1	1	2019	183664
2019-01-01	1	Tuesday	1	1	2019	183664
2019-01-01	1	Tuesday	1	1	2019	183664

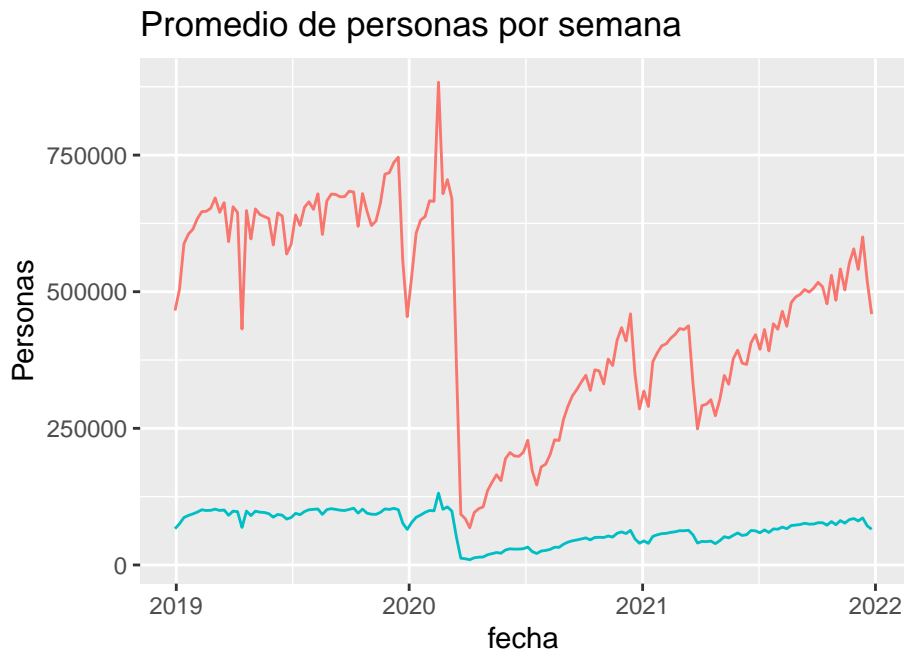
Table 4: Linea B

fecha	dia	dia.sem	sem	mes	anio	per.dia.tot
2019-01-01	1	Tuesday	1	1	2019	25852
2019-01-01	1	Tuesday	1	1	2019	25852
2019-01-01	1	Tuesday	1	1	2019	25852
2019-01-01	1	Tuesday	1	1	2019	25852
2019-01-01	1	Tuesday	1	1	2019	25852
2019-01-01	1	Tuesday	1	1	2019	25852

g).

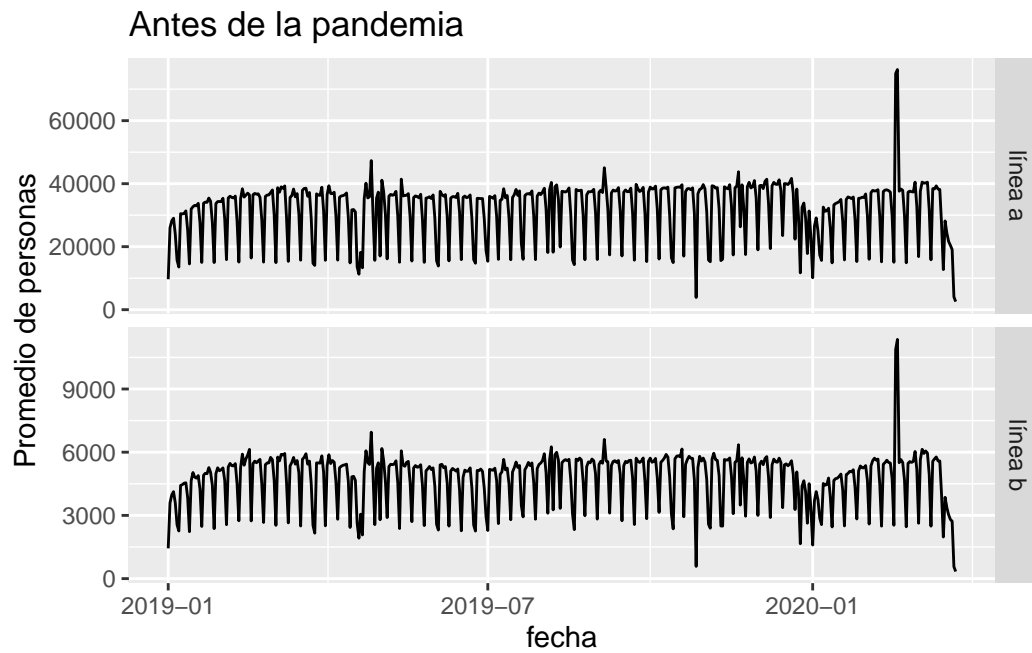


Se ve una caída(**cisne negro**) en marzo de 2020 y una lenta recuperación, no se ha llegado a niveles de 2019.



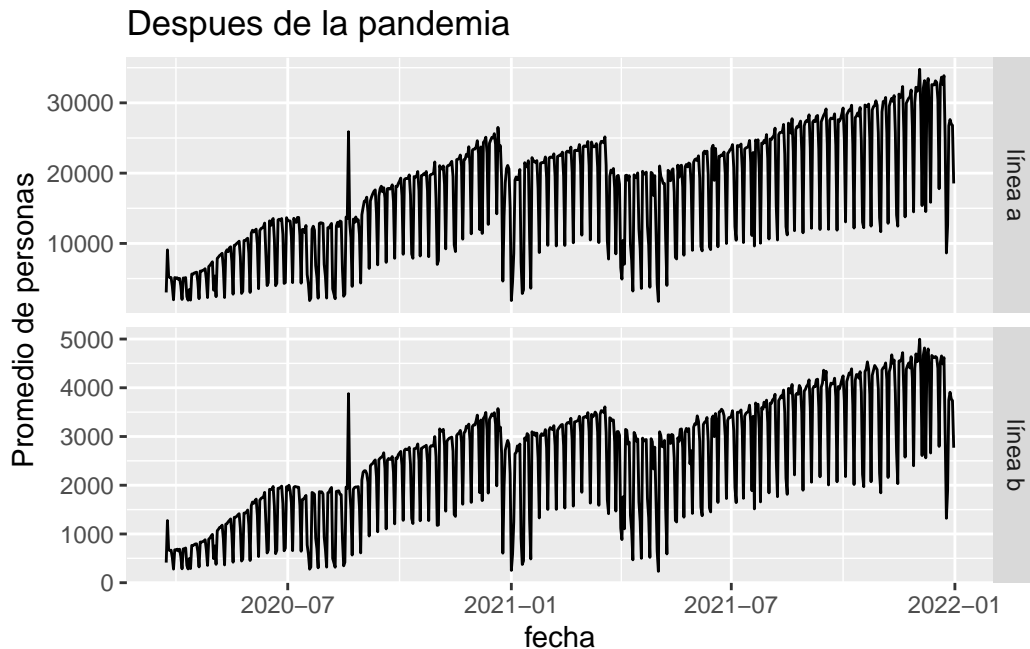
Se observa que la línea que cuenta con mayor afluencia de pasajeros es la **línea A**.

h).



Para antes de la pandemia la línea a y b tuvieron un comportamiento muy similar.

Se ve que es estacionaria alrededor de una media, además, en la última semana hay una baja afluencia de personas en ambas líneas muy probablemente a causa del covid.



en esta serie de tiempo se ve una recuperación en ambas líneas (a y b), no es estacionaria ya que se ve una clara tendencia de crecimiento positivo no lineal.

i).

regresion con lm para dos dataset: antes y despues de la pandemia

puede ser con las covariables: indicadora dia de la semana (lunes, martes etc), mes, hora?

son 4 modelos: antes pandemia linea a antes pandemia linea a despues pandemia linea b
despues pandemia linea b

contrastar, dar una interpretación del summary del model

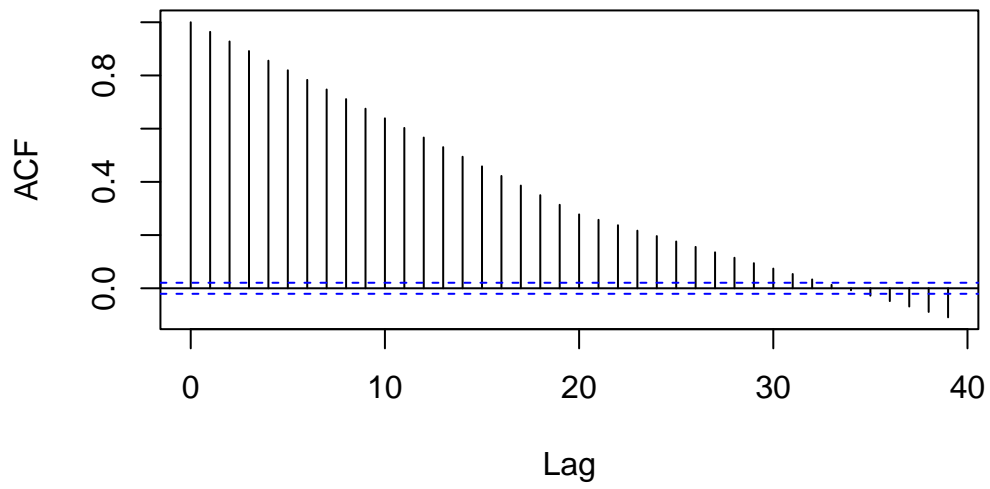
graficar el modelo vs real

Modelo linea A antes de pandemia:

```
df.lin.A <- dat_lin_A |> filter(pandemia == "no")
df.lin.A$dia.sem <- as.factor(df.lin.A$dia.sem)

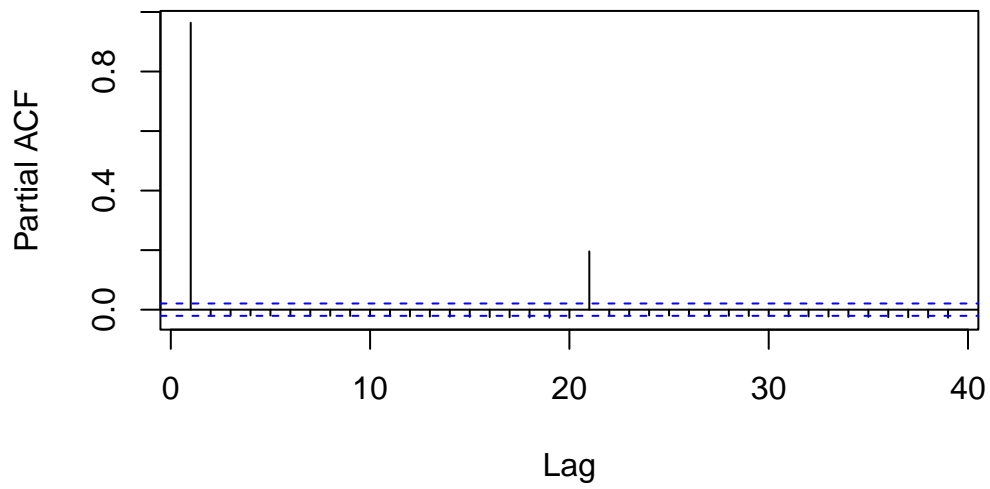
acf(df.lin.A$per.dia.tot)
```

Series df.lin.A\$per.dia.tot

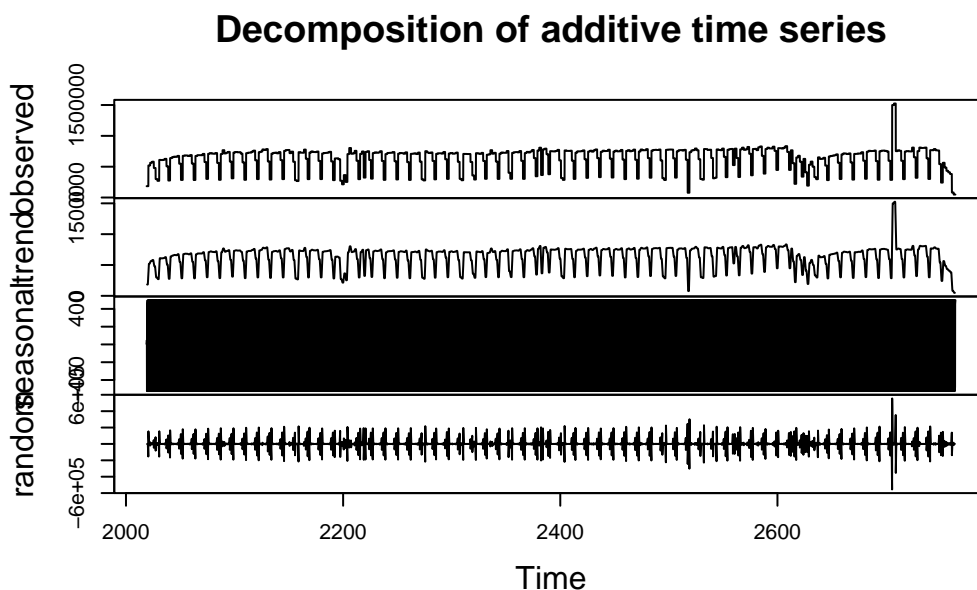


```
pacf(df.lin.A$per.dia.tot)
```

Series df.lin.A\$per.dia.tot



```
serie <- ts(df.lin.A$per.dia.tot, start = 2019, frequency = 12)
decom <- decompose(serie)
plot(decom)
```



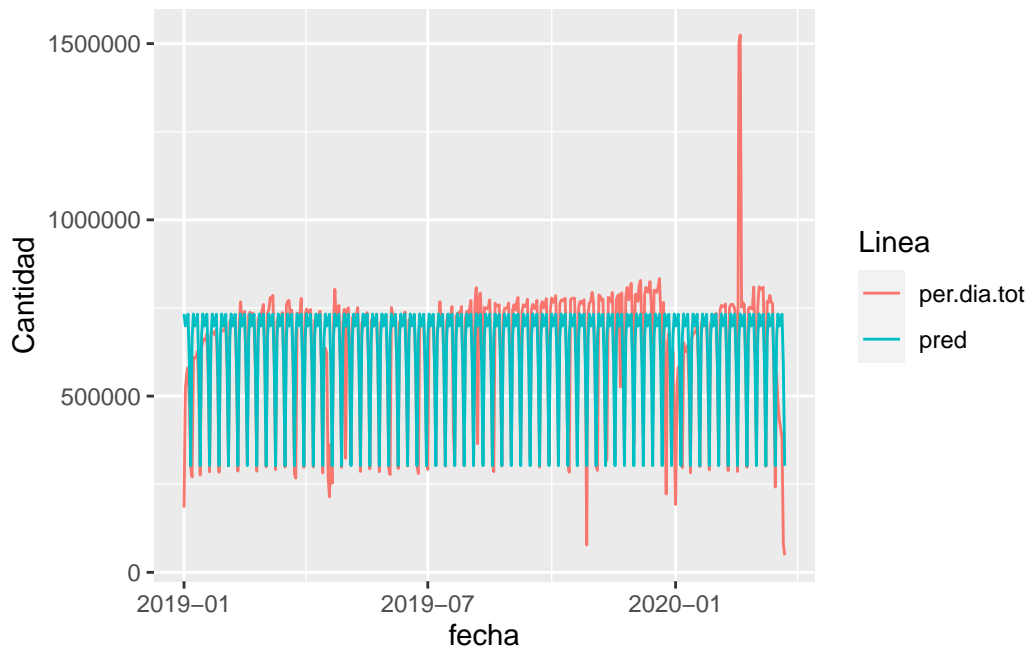
```
model.lin.A <- lm(per.dia.tot ~ dia.sem, data = df.lin.A)
kable(xtable(summary(model.lin.A)))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	732335.1719	3353.006	218.4115132	0.0000000
dia.semMonday	-66069.9814	4760.647	-13.8783627	0.0000000
dia.semSaturday	-144624.7500	4741.867	-30.4995378	0.0000000
dia.semSunday	-429790.9219	4741.867	-90.6374911	0.0000000
dia.semThursday	-18168.1094	4741.867	-3.8314254	0.0001283
dia.semTuesday	617.1406	4741.867	0.1301472	0.8964529
dia.semWednesday	-33569.0938	4741.867	-7.0792990	0.0000000

```
df.lin.A$pred <- predict(model.lin.A, newdata = df.lin.A)
df.lin.A$per.dia.tot <- as.numeric(df.lin.A$per.dia.tot)
df.lin.A$pred <- as.numeric(df.lin.A$pred)
```

```
ajust <- df.lin.A %>% gather(key="Linea", value="Cantidad", per.dia.tot, pred)
```

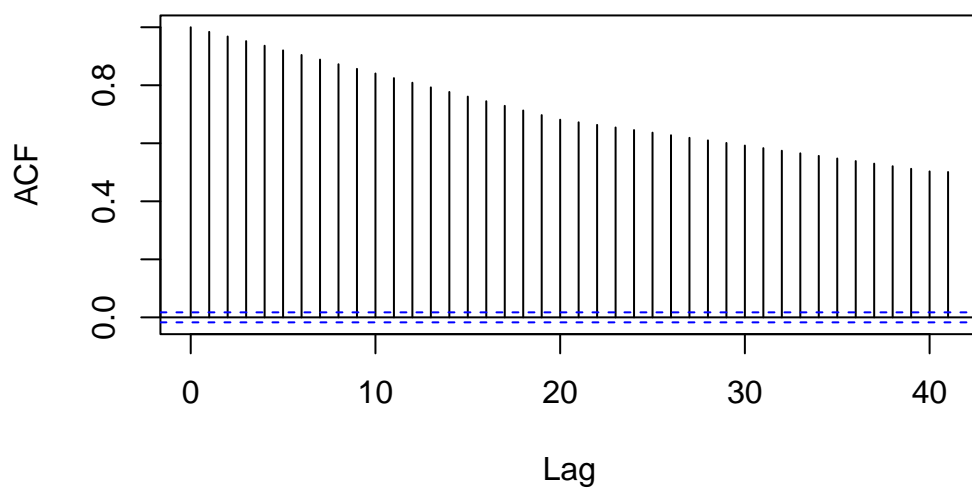
```
ajust %>% ggplot(aes(x=fecha, y=Cantidad,  
col=Linea))+  
geom_line()
```



Modelo linea A despues de pandemia:

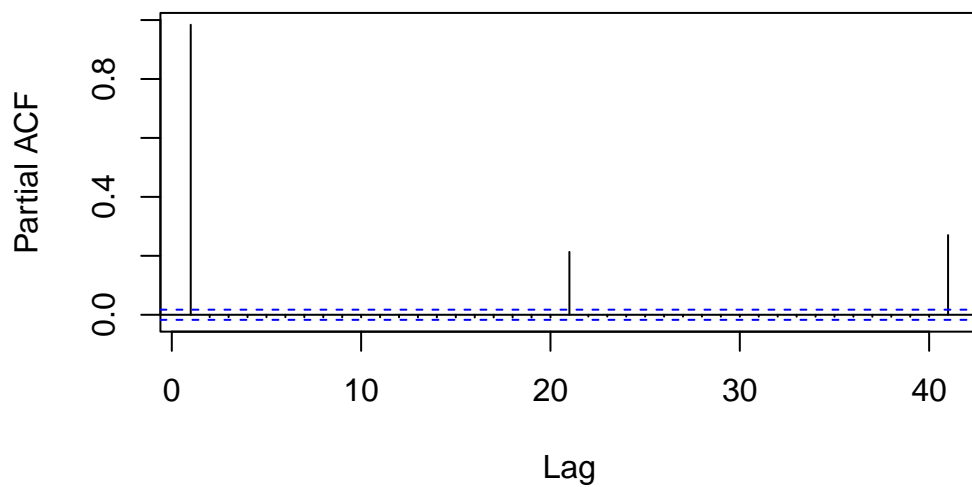
```
df.lin.A.p <- dat_lin_A |> filter(pandemia == "si")  
df.lin.A.p$dia.sem <- as.factor(df.lin.A.p$dia.sem)  
  
acf(df.lin.A.p$per.dia.tot)
```

Series df.lin.A.p\$per.dia.tot



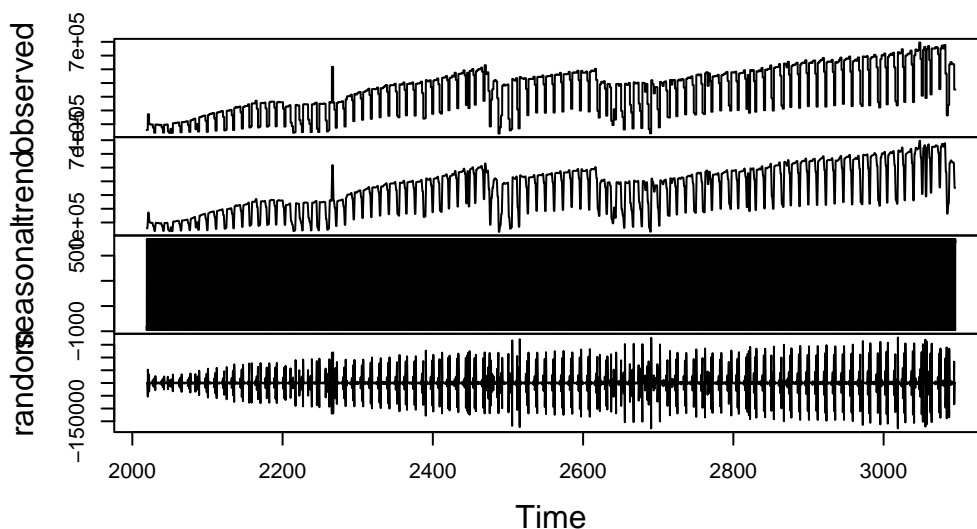
```
pacf(df.lin.A.p$per.dia.tot)
```

Series df.lin.A.p\$per.dia.tot



```
serie <- ts(df.lin.A.p$per.dia.tot, start = 2019, frequency = 12)
decom <- decompose(serie)
plot(decom)
```

Decomposition of additive time series



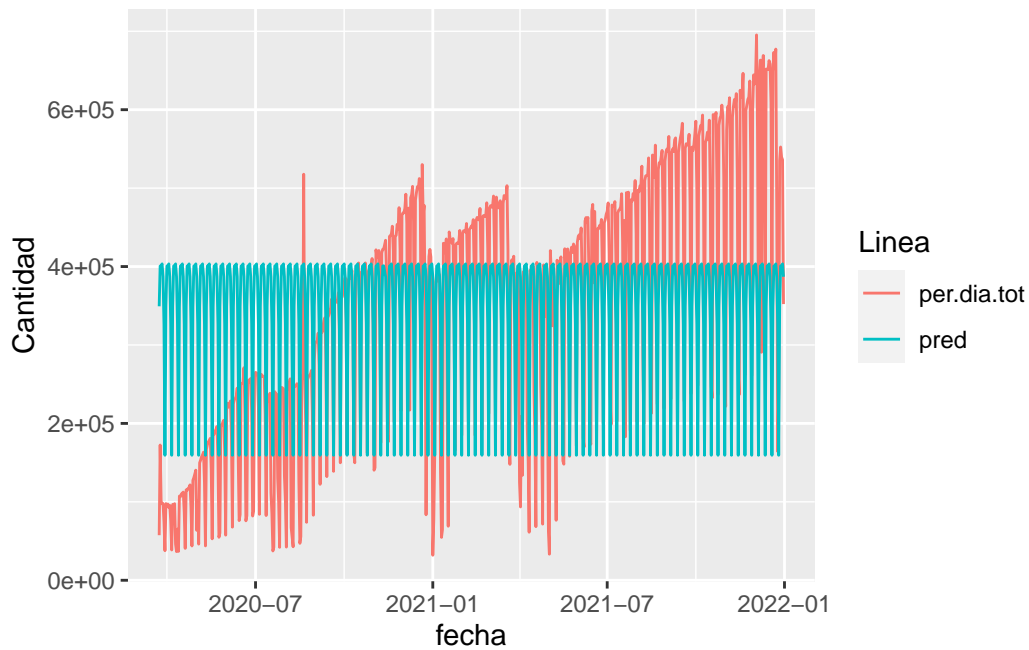
```
model.lin.A.p <- lm(per.dia.tot ~ dia.sem, data = df.lin.A.p)
kable(xtable(summary(model.lin.A.p)))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	386976.97	3388.259	114.211146	0.0000000
dia.semMonday	-37648.75	4791.722	-7.857040	0.0000000
dia.semSaturday	-60998.05	4831.488	-12.625105	0.0000000
dia.semSunday	-227567.77	4804.725	-47.363324	0.0000000
dia.semThursday	16461.42	4804.725	3.426090	0.0006142
dia.semTuesday	13694.76	4791.722	2.858004	0.0042700
dia.semWednesday	14057.38	4791.722	2.933679	0.0033556

```
df.lin.A.p$pred <- predict(model.lin.A.p, newdata = df.lin.A.p)
df.lin.A.p$per.dia.tot <- as.numeric(df.lin.A.p$per.dia.tot)
df.lin.A.p$pred <- as.numeric(df.lin.A.p$pred)
```

```
ajust <- df.lin.A.p %>% gather(key="Linea", value="Cantidad", per.dia.tot, pred)
```

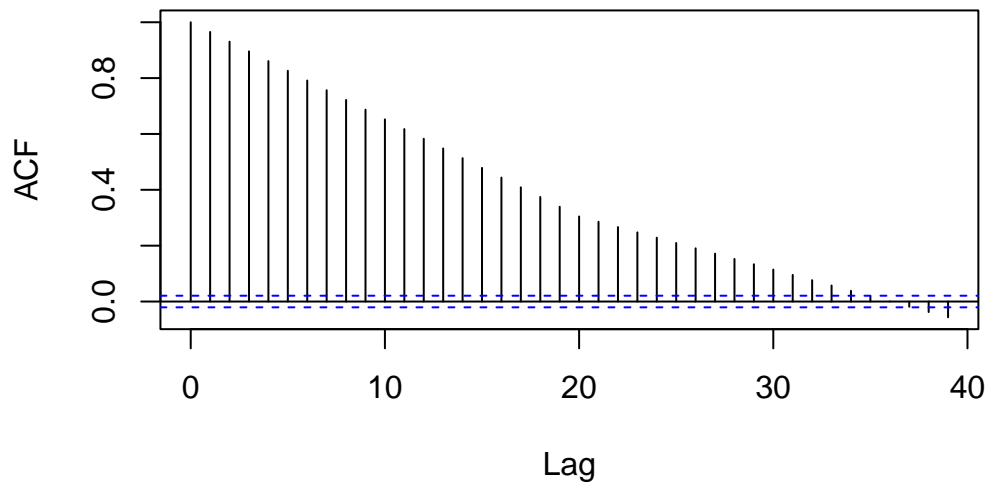
```
ajust %>% ggplot(aes(x=fecha, y=Cantidad,  
col=Linea))+  
geom_line()
```



Modelo linea B antes de pandemia:

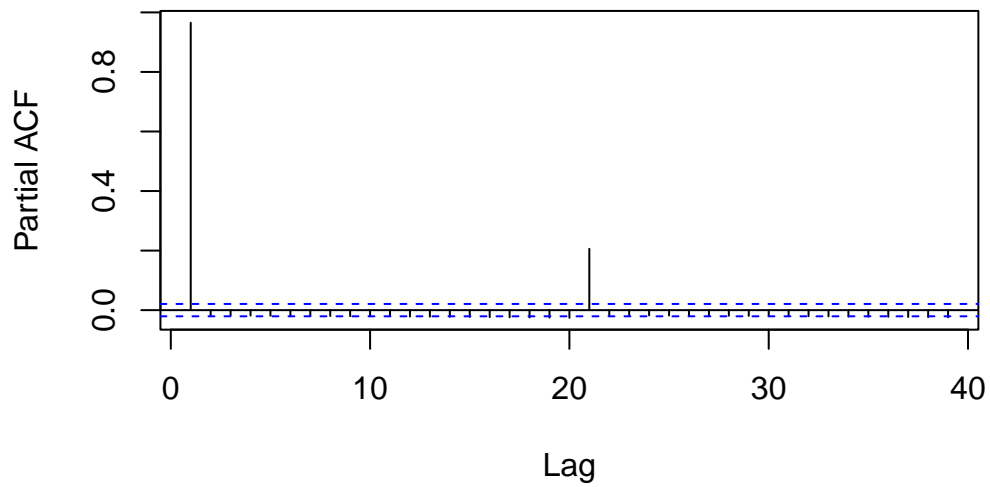
```
df.lin.B <- dat_lin_B |> filter(pandemia == "no")  
df.lin.B$dia.sem <- as.factor(df.lin.B$dia.sem)  
  
acf(df.lin.B$per.dia.tot)
```


Series df.lin.B\$per.dia.tot



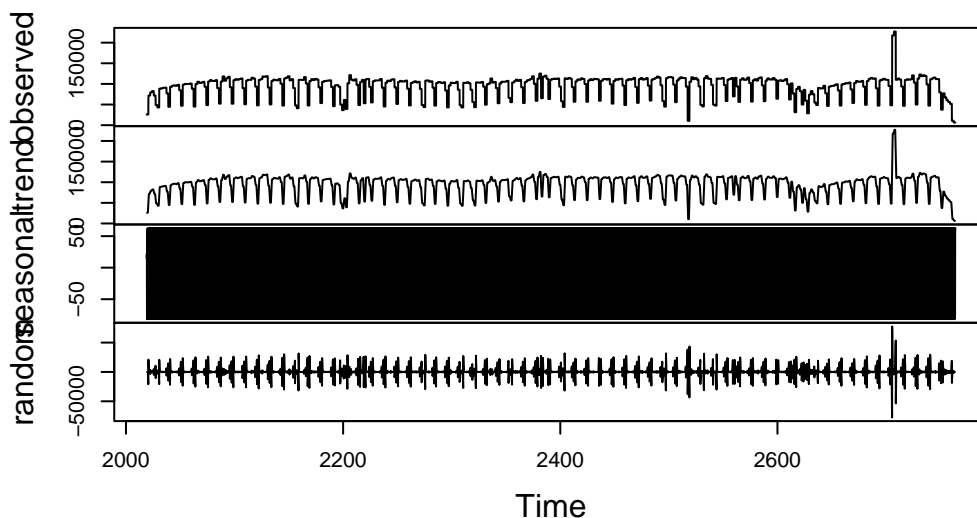
```
pacf(df.lin.B$per.dia.tot)
```

Series df.lin.B\$per.dia.tot



```
serie <- ts(df.lin.B$per.dia.tot, start = 2019, frequency = 12)
decom <- decompose(serie)
plot(decom)
```

Decomposition of additive time series



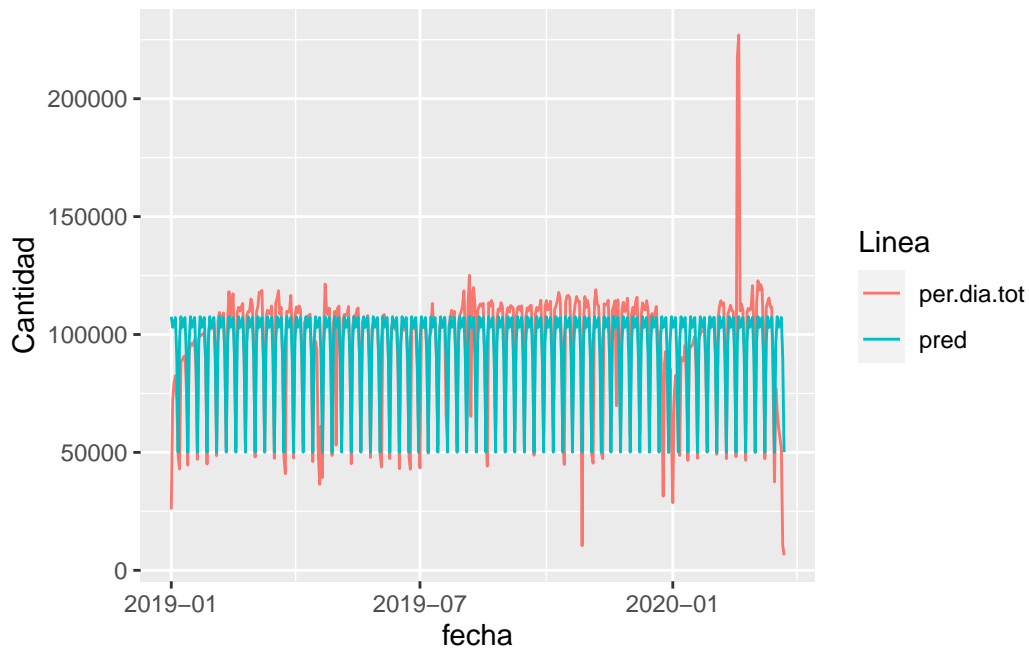
```
model.lin.B <- lm(per.dia.tot ~ dia.sem, data = df.lin.B)
kable(xtable(summary(model.lin.B)))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	107207.5938	492.7545	217.5679444	0.0000000
dia.semMonday	-10398.9906	699.6200	-14.8637694	0.0000000
dia.semSaturday	-16794.4531	696.8602	-24.1001768	0.0000000
dia.semSunday	-57116.0625	696.8602	-81.9620142	0.0000000
dia.semThursday	-2037.1562	696.8602	-2.9233358	0.0034717
dia.semTuesday	342.7969	696.8602	0.4919163	0.6227906
dia.semWednesday	-4182.1250	696.8602	-6.0013834	0.0000000

```
df.lin.B$pred <- predict(model.lin.B, newdata = df.lin.B)
df.lin.B$per.dia.tot <- as.numeric(df.lin.B$per.dia.tot)
df.lin.B$pred <- as.numeric(df.lin.B$pred)
```

```
ajust <- df.lin.B %>% gather(key="Linea", value="Cantidad", per.dia.tot, pred)
```

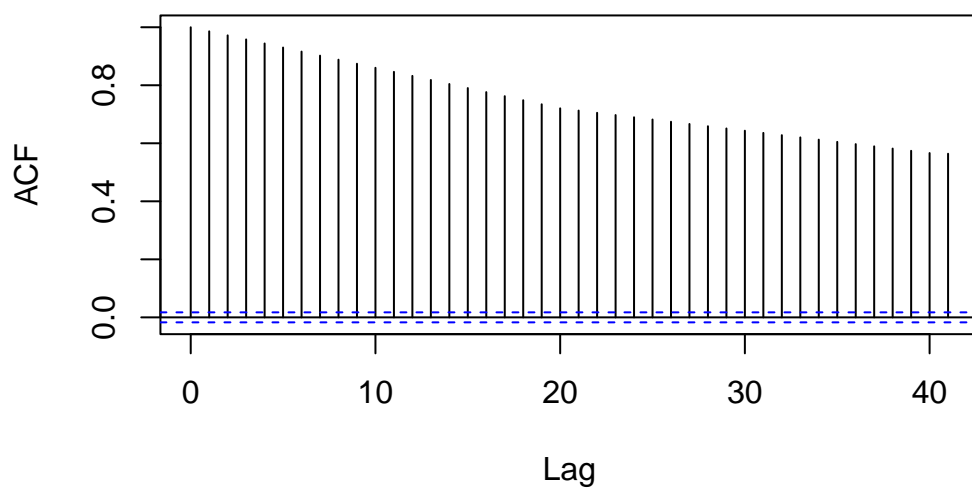
```
ajust %>% ggplot(aes(x=fecha, y=Cantidad,  
col=Linea))+  
geom_line()
```



Modelo linea B despues de pandemia:

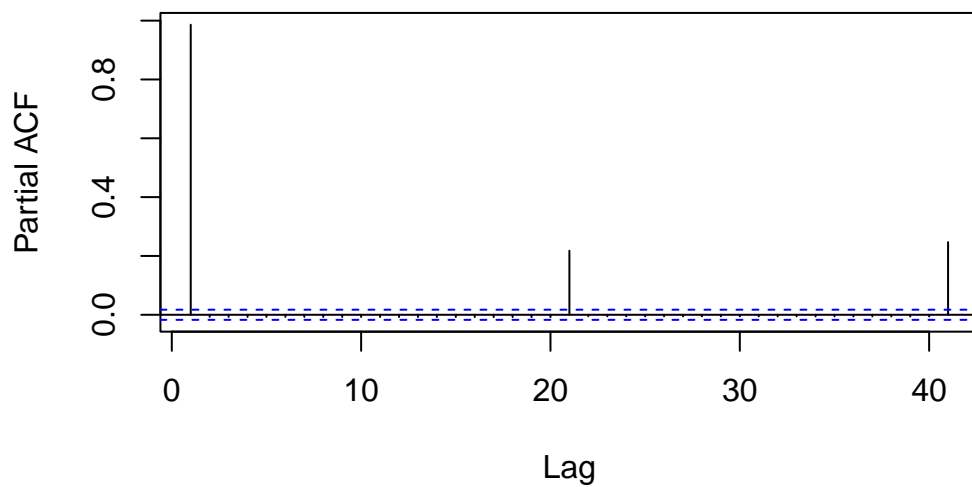
```
df.lin.B.p <- dat_lin_B |> filter(pandemia == "si")  
df.lin.B.p$dia.sem <- as.factor(df.lin.B.p$dia.sem)  
  
acf(df.lin.B.p$per.dia.tot)
```

Series df.lin.B.p\$per.dia.tot



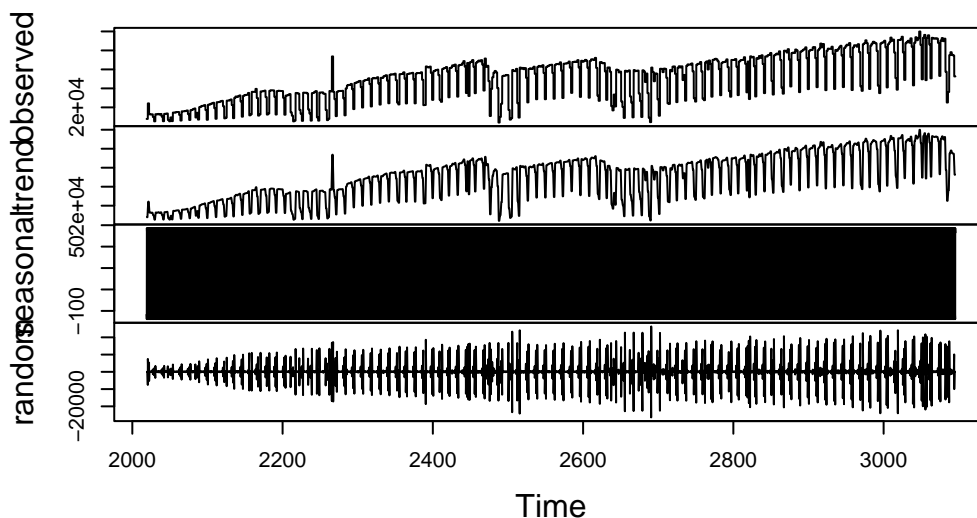
```
pacf(df.lin.B.p$per.dia.tot)
```

Series df.lin.B.p\$per.dia.tot



```
serie <- ts(df.lin.B.p$per.dia.tot, start = 2019, frequency = 12)
decom <- decompose(serie)
plot(decom)
```

Decomposition of additive time series



```
model.lin.B.p <- lm(per.dia.tot ~ dia.sem, data = df.lin.B.p)
kable(xtable(summary(model.lin.B.p)))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	55437.935	505.2586	109.721898	0.0000000
dia.semMonday	-4778.548	714.5436	-6.687553	0.0000000
dia.semSaturday	-7388.347	720.4735	-10.254848	0.0000000
dia.semSunday	-30073.718	716.4827	-41.974104	0.0000000
dia.semThursday	3028.119	716.4827	4.226367	0.0000239
dia.semTuesday	2507.774	714.5436	3.509617	0.0004503
dia.semWednesday	2696.419	714.5436	3.773625	0.0001616

```
df.lin.B.p$pred <- predict(model.lin.B.p, newdata = df.lin.B.p)
df.lin.B.p$per.dia.tot <- as.numeric(df.lin.B.p$per.dia.tot)
df.lin.B.p$pred <- as.numeric(df.lin.B.p$pred)
```

```
ajust <- df.lin.B.p %>% gather(key="Linea", value="Cantidad", per.dia.tot, pred)
```

```
ajust %>% ggplot(aes(x=fecha, y=Cantidad,  
col=Linea))+  
geom_line()
```

