Series Temporales

Armando Ocampo

Librerías de trabajo

Antes de comenzar a trabajar, debemos llamar a nuestras librerias de trabajo. Si no cuentas con alguna de estas librerias, puedes descargarla mediante la función *install.packages()*, y el nombre de la paquetería entre comillas. Por ejemplo, *install.packages('dplyr')*.

```
library(readr)
library(dplyr)
library(ggplot2)
library(ggpubr)
```

Dataset de trabajo

De la misma manera, llamaremos a nuestro conjunto de datos a utilizar en el desarrollo del proyecto. Este dataset contiene información de diferentes variables asociadas con la pandemia de COVID-19, tomando solo información de México. Presenta un rango del 01 de enero del 2020 al 02 de agosto del 2023. Para más información, puedes acceder al sitio de descarga de los datos crudos, **Our World in Data**.

```
covid_mexico <- readRDS('../data/covid_mexico.RDS')
```

Series Temporales

Las series de tiempo de caracterizan por presentar información en un intervalo de tiempo definido. Este proceso permite comparar tendencias en los datos. Así como sitios de estancamiento. Para su exploración, es posible utilizar funciones como *min()*, *max()*, *length()* y *deltat()*. Esta última función arroja el intervalo de tiempo en días presente en el dataset.

```
min(covid_mexico$date)
```

```
## [1] "2020-01-01"
```

```
max(covid_mexico$date)
```

```
## [1] "2023-08-02"
```

```
length(covid_mexico$date)
```

```
## [1] 1310
```

```
1310/365  #3.5 años
```

```
## [1] 3.589041
```

```
deltat(covid_mexico$date) # intervalo de tiempo en días
```

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

Otra de las funciones que permiten describir el dataset es la función *diff()*, de la paquetería *base*. Esta función describe la diferencia de cada unidad con el valor previo en el rango de fechas establecidos.

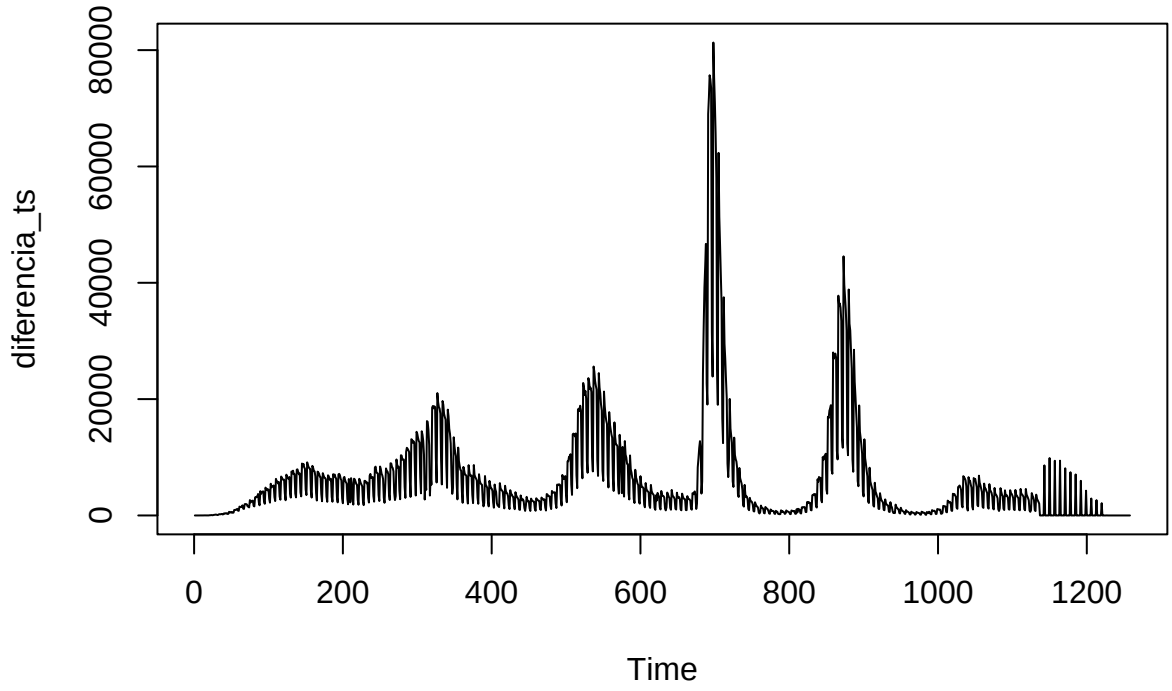
```
diferencia_casos <- diff(covid_mexico$total_cases) %>%
  na.omit()

head(diferencia_casos,20)
```

```
##  [1]  0  0  0  0  0  0  0  3  2  2  2  5  9 10 11  6  7  8  5 17
```

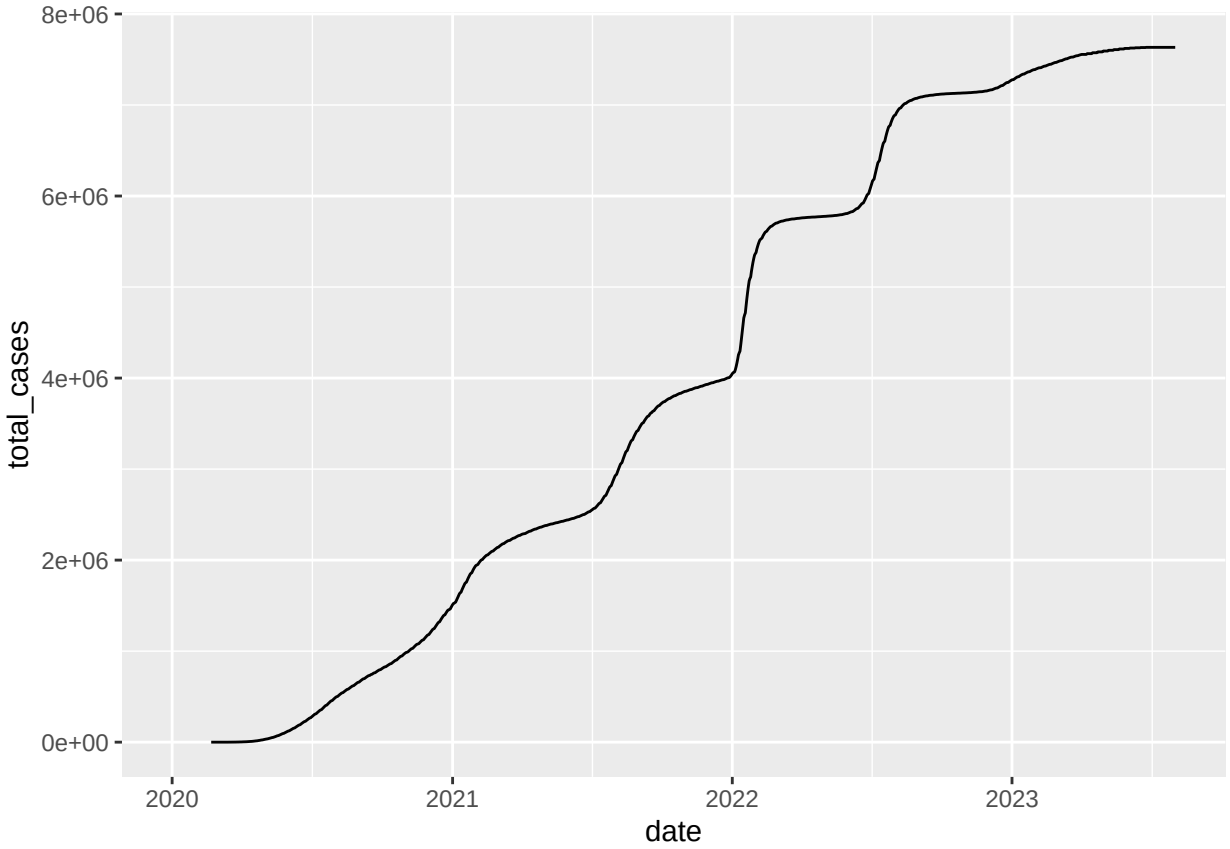
Mediante la función *ts()* de la paquetería *stats* es porible convertir esta diferencia en una serie de tiempo y graficar los valores dentro del periodo de tiempo marcado. De esta manera es posible identificar los días con ascenso, descenso o estancamiento de la información.

```
diferencia_ts <- ts(diferencia_casos)
plot(diferencia_ts)
```



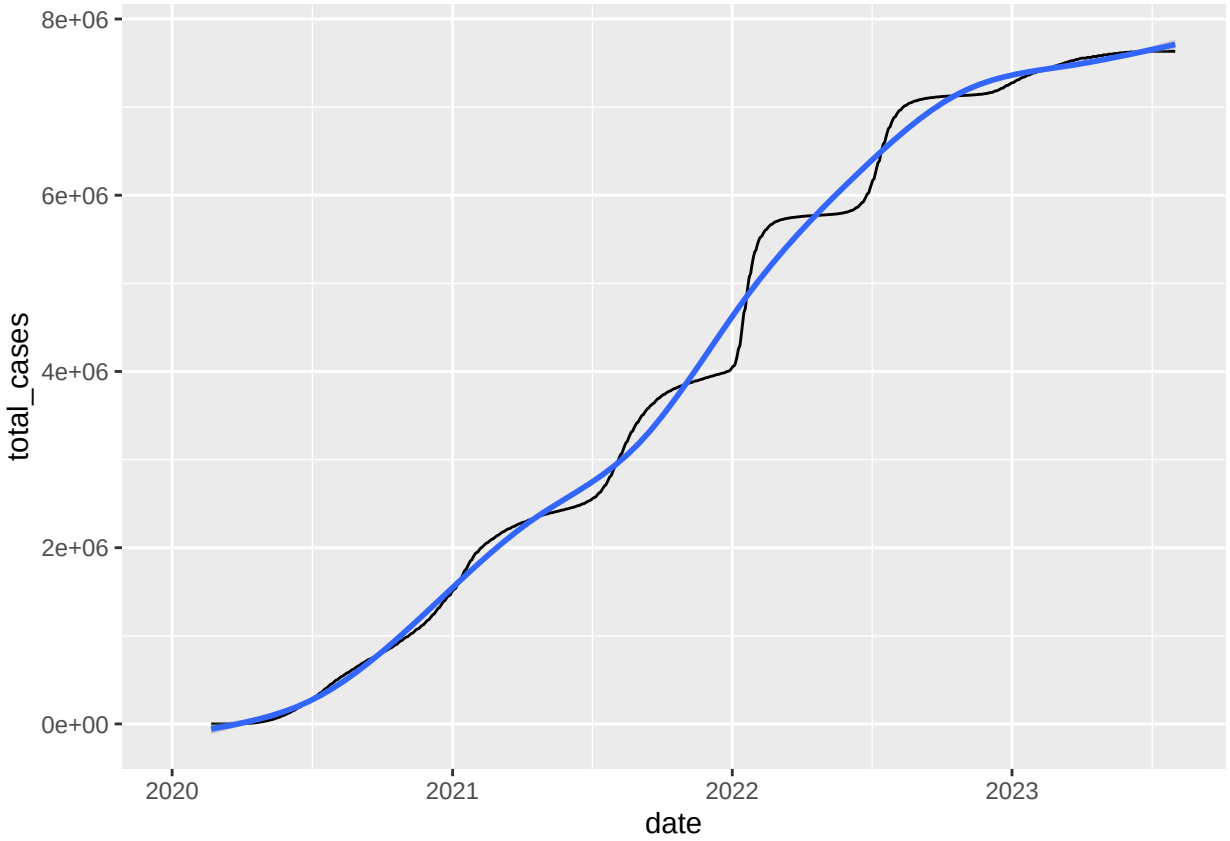
Mediante la función *ggplot()* de la paquetería *ggplot2*, podemos evaluar la tendencia de los datos utilizando como base una fecha establecida. Esto mediante un gráfico de líneas. A continuación se muestra un ejemplo, en el cual eje 'x' se conforma por la fecha, y el eje 'y' por el total de casos por COVID-19.

```
ggplot(covid_mexico, aes(x = date, y = total_cases)) +
  geom_line()
```



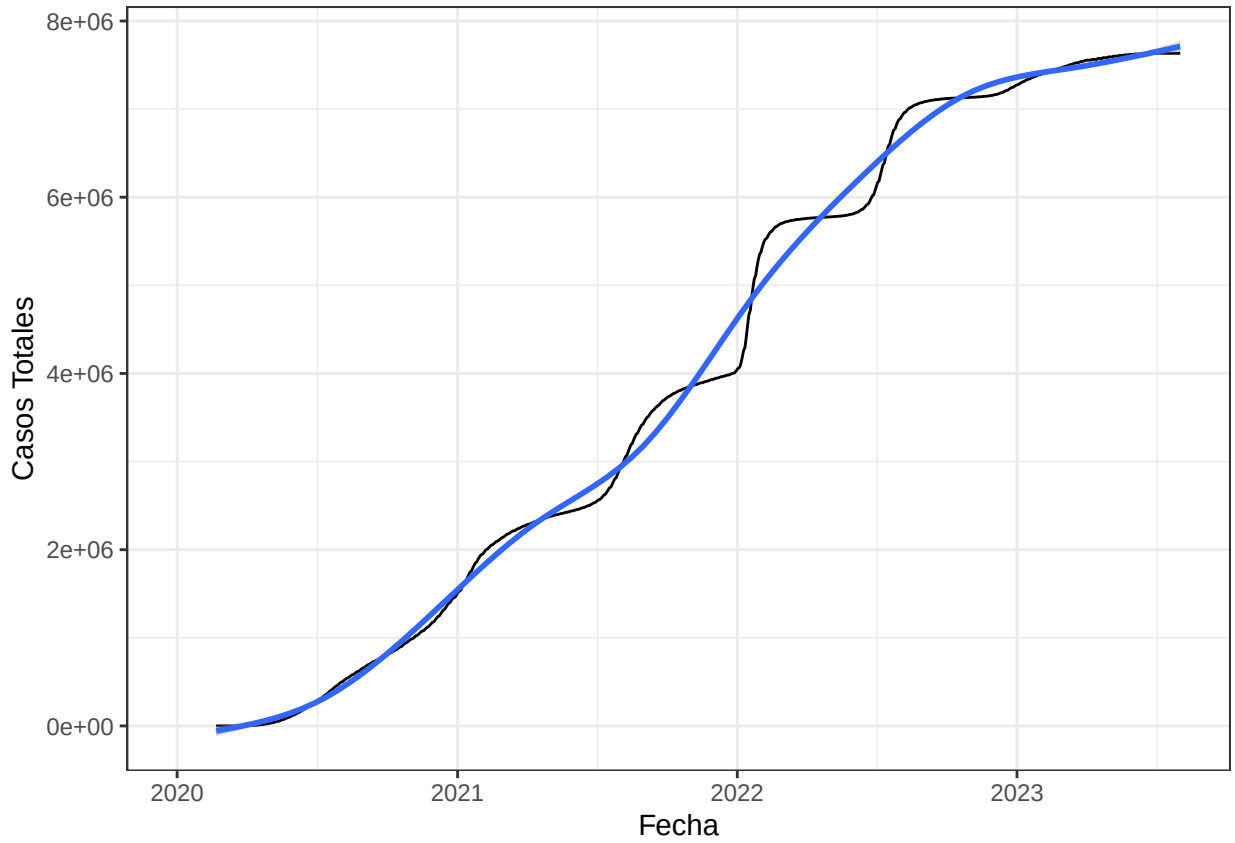
Tomando este gráfico como base, podemos agregar la función *geom_smooth()* para graficar la líneas de tendencia de los datos.

```
ggplot(covid_mexico, aes(x = date, y = total_cases)) +
  geom_line() +
  geom_smooth()
```



Por último, agregaremos algunos estéticos al gráfico.

```
ggplot(covid_mexico, aes(x = date, y = total_cases)) +
  geom_line() +
  geom_smooth() +
  xlab('Fecha') +
  ylab('Casos Totales') +
  theme_bw()
```



Uno de los puntos a detallar es la modificación del formato del eje de las equis. Siendo posible agregar los años, meses y días al gráfico. Para esto, se utiliza la función *scale_x_date()* compañado del siguiente código de formato para la fecha:

%d: Días como un número del 1 al

%a: Abreviatura del día de la semana. 'Lun'

%A: Día de la semana sin abreviatura. 'Lunes'

%m: Mes con número del 1 al 12

%b: Abreviatura del mes. 'Ene'

%B: Mes sin abreviatura. 'Enero'

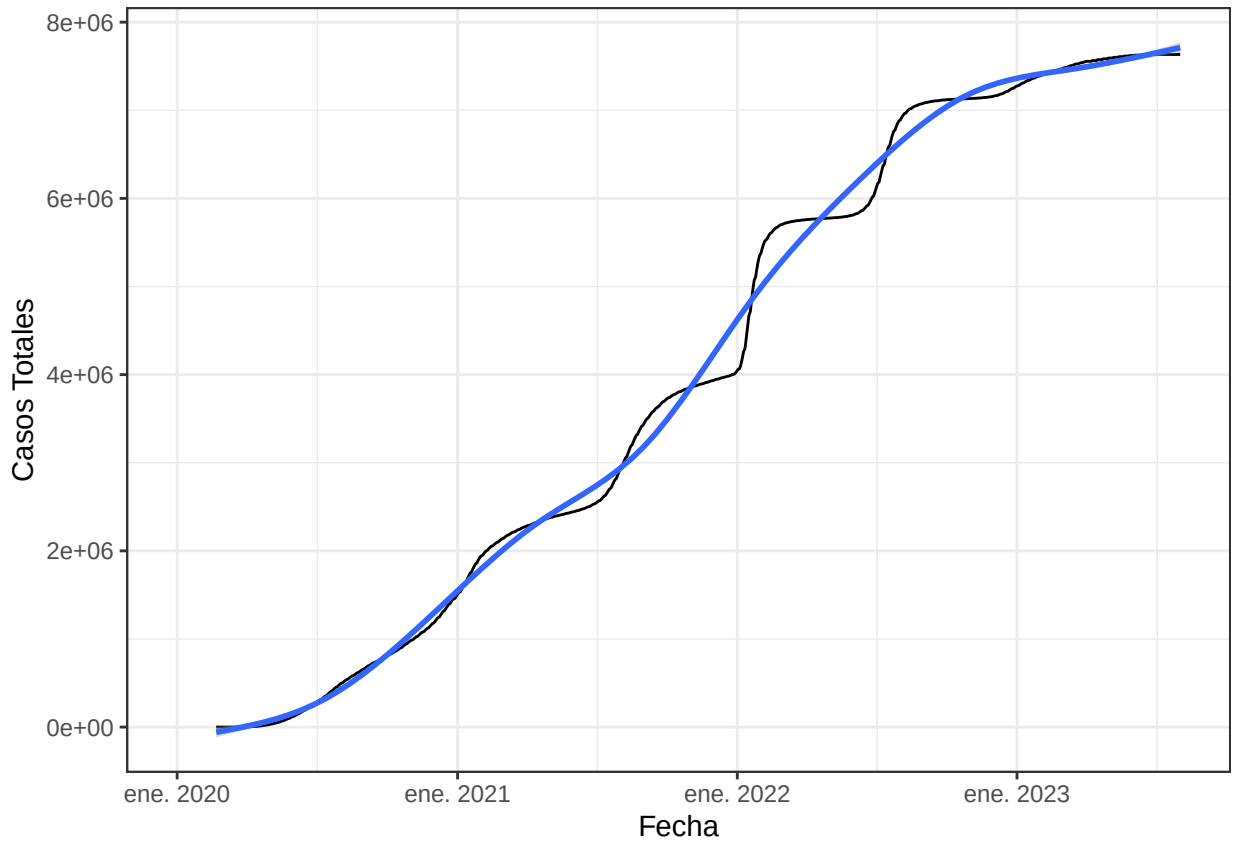
%y: Año con dos dígitos. '23'

%Y: Año con 4 dígitos. '2023'

%W: Semana del año con un número entre 1 y 2.

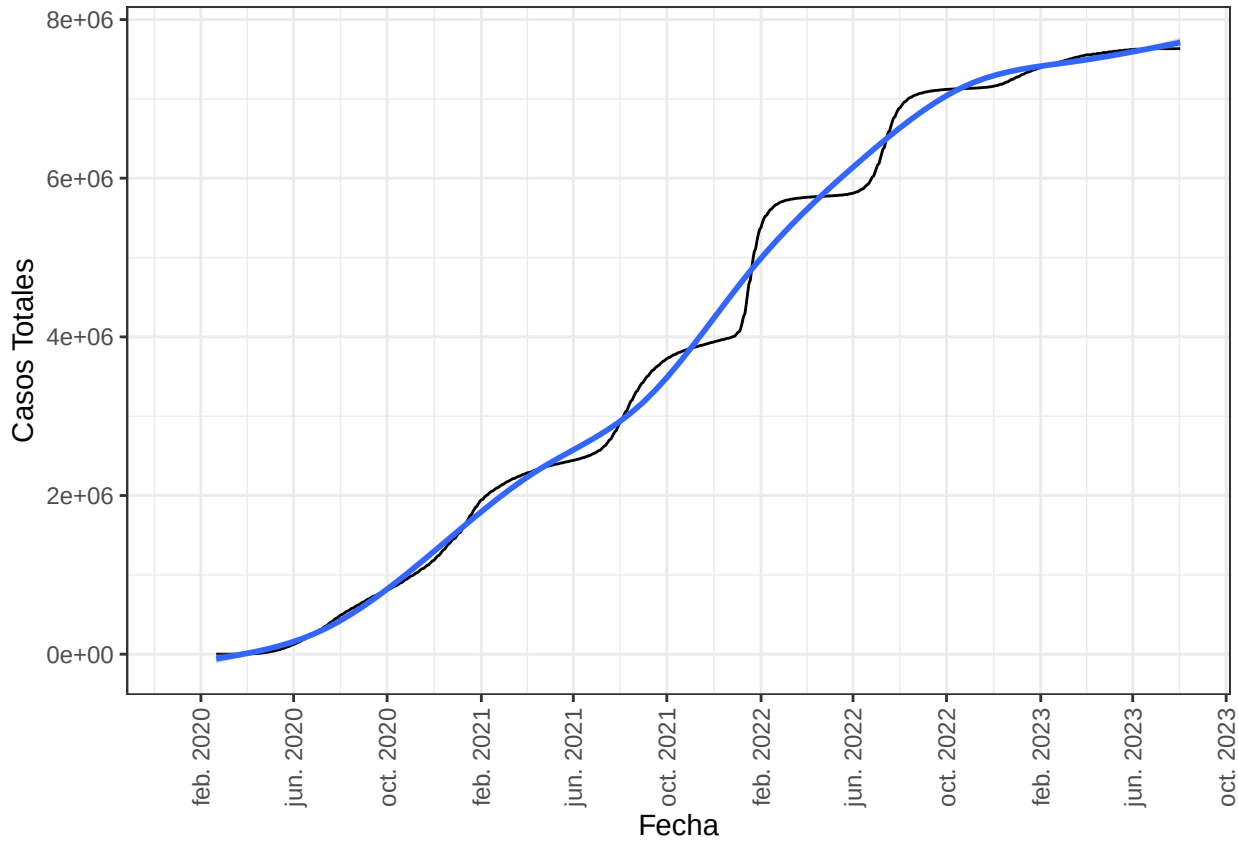
En el siguiente ejemplo se modificará el eje de las equis, colocando el nombre del mes de forma abreviada y el año en formato de 4 digitos.

```
ggplot(covid_mexico, aes(x = date, y = total_cases)) +
  geom_line() +
  geom_smooth() +
  xlab('Fecha') +
  ylab('Casos Totales') +
  scale_x_date(date_labels = '%b %Y') +
  theme_bw()
```



Asimismo, en la función *scale_x_date()* es posible agregar el argumento *date_breaks=* para generar intervalos de tiempo en el eje. A continuación, se realizarán cortes en la fecha cada 4 meses.

```
ggplot(covid_mexico, aes(x = date, y = total_cases)) +
  geom_line() +
  geom_smooth() +
  xlab('Fecha') +
  ylab('Casos Totales') +
  theme_bw() +
  scale_x_date(date_labels = '%b %Y',
               date_breaks = '4 month') +
  theme(axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust=1))
```



En ocasiones no se requiere trabajar con todo el dataset, sino intervalos definidos de tiempo. Para ello, es necesario filtrar la información. A continuación, se muestra el intervalo de tiempo del conjunto de datos.

```
min(covid_mexico$date) # 01 enero 2020
```

```
## [1] "2020-01-01"
```

```
max(covid_mexico$date) # 02 agosto 2023
```

```
## [1] "2023-08-02"
```

Supongamos que solo necesitamos información 2021. Para esto necesitamos filtrar la variable *date*, mediante la función *filter()* de la paquetería *dplyr*. En este filtro es necesario colocar dos condiciones, una para el intervalo menor y la segunda para el intervalo mayor.

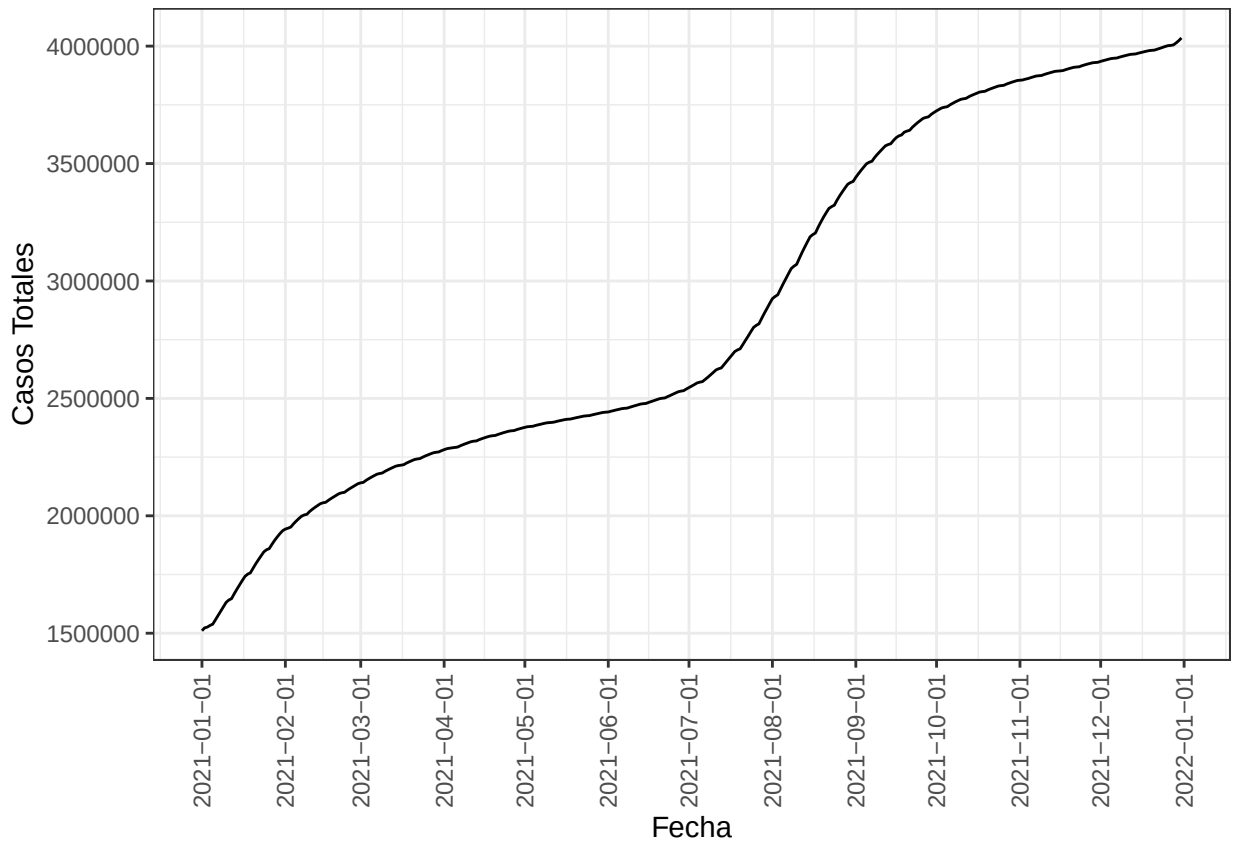
```
covid_mexico %>%
  filter(date >= '2021-01-01',
         date <= '2021-12-31')
```

```
## # A tibble: 365 x 67
##    iso_code continent locat~1 date       total~2 new_c~3 new_c~4 total~5 new_d~6
##    <chr>    <chr>     <chr>   <date>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 MEX      North Am~ Mexico  2021-01-01 1510795   14728  10085.  148569     946
##  2 MEX      North Am~ Mexico  2021-01-02 1522878   12083  10314.  149455     886
##  3 MEX      North Am~ Mexico  2021-01-03 1526291    3413  10411   150442     987
##  4 MEX      North Am~ Mexico  2021-01-04 1533239    6948  10556   151435     993
##  5 MEX      North Am~ Mexico  2021-01-05 1538513    5274  10681.  152472    1037
##  6 MEX      North Am~ Mexico  2021-01-06 1557069   18556  11014.  153584    1112
##  7 MEX      North Am~ Mexico  2021-01-07 1575890   18821  11403.  154653    1069
```

```
##  8 MEX      North Am~ Mexico  2021-01-08 1594299   18409  11929.  155813    1160
##  9 MEX      North Am~ Mexico  2021-01-09 1613065   18766  12884.  156877    1064
## 10 MEX      North Am~ Mexico  2021-01-10 1631666   18601  15054.  158074    1197
## # ... with 355 more rows, 58 more variables: new_deaths_smoothed <dbl>,
## #   total_cases_per_million <dbl>, new_cases_per_million <dbl>,
## #   new_cases_smoothed_per_million <dbl>, total_deaths_per_million <dbl>,
## #   new_deaths_per_million <dbl>, new_deaths_smoothed_per_million <dbl>,
## #   reproduction_rate <dbl>, icu_patients <dbl>,
## #   icu_patients_per_million <dbl>, hosp_patients <dbl>,
## #   hosp_patients_per_million <dbl>, weekly_icu_admissions <dbl>, ...
```

La ventaja de trabajar con *dplyr* es el desarrollo de flujos de trabajo que permiten graficar la información de manera directa. En este apartado graficaremos los datos de total de casos en 2021. Utilizando una escala de 1 mes para el eje de las equis.

```
covid_mexico %>%
  filter(date >= '2021-01-01',
         date <= '2021-12-31') %>%
  ggplot(aes(x = date, y = total_cases)) +
  geom_line() +
  scale_x_date(date_breaks = '1 month') +
  theme_bw()+
  theme(axis.text.x = element_text(angle = 90,
                                   vjust = 0.5, hjust=1))+
  xlab('Fecha') +
  ylab('Casos Totales')
```



La evalución de variables con diferente escala suele ser un problema en estadística. No obstante, mediante la evaluación de la línea de tendencia es posible identificar la relación de estos procesos. Mediante la función *ggarrange()* de la paquetería *ggpubr* es posible pegar diferentes gráficos y comparar las líneas de tendencia. Antes de utilizar la función generaremos 3 gráficos, siendo esto el total de casos para COVID-19, el total de muertes y el total de pruebas de detección en 2021. A cada uno de estos gráficos se le asignará un nombre.

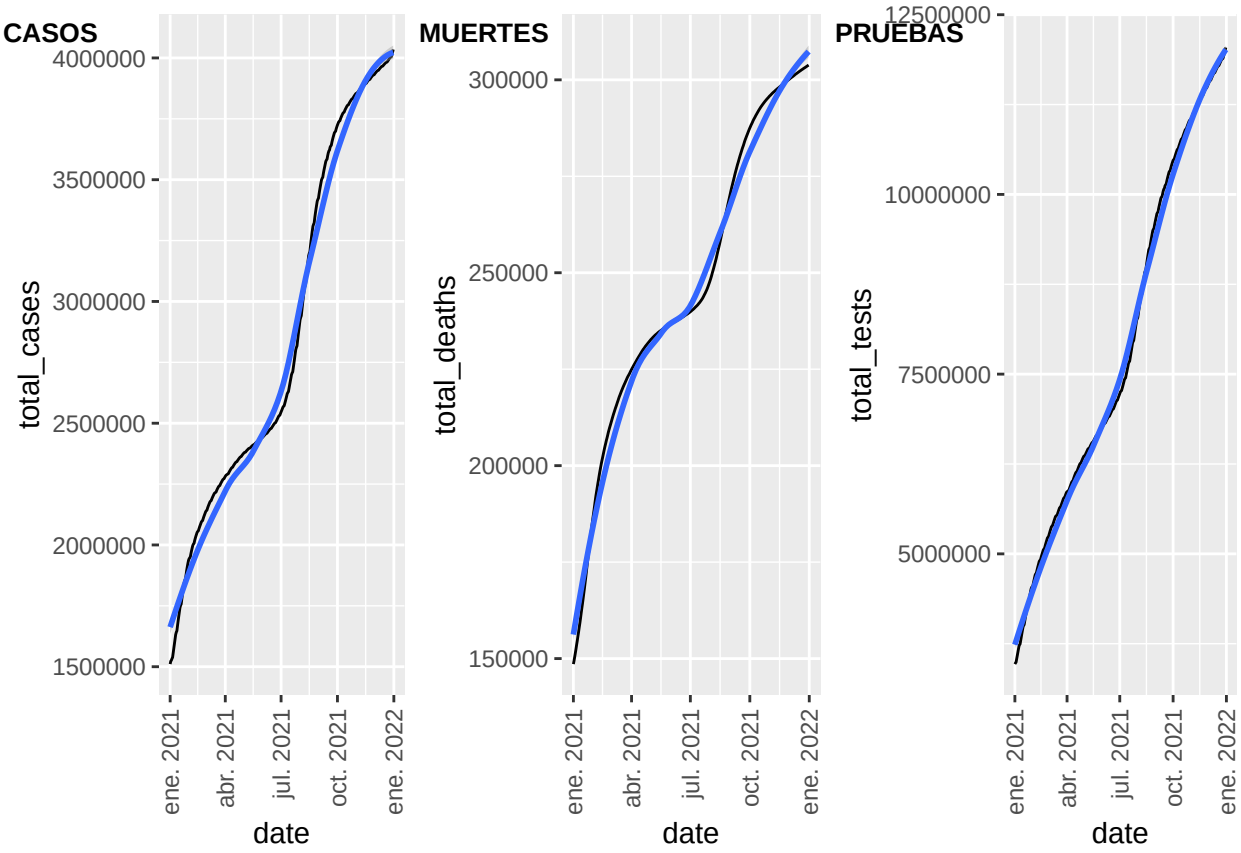
```
a <- covid_mexico %>%
  filter(date >= '2021-01-01',
         date <= '2021-12-31') %>%
  select(total_cases, date) %>%
  ggplot(aes(y = total_cases, x = date)) +
  geom_line() +
  geom_smooth() +
  theme(axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust=1))

b <- covid_mexico %>%
  filter(date >= '2021-01-01',
         date <= '2021-12-31') %>%
  select(total_deaths, date) %>%
  ggplot(aes(y = total_deaths, x = date)) +
  geom_line() +
  geom_smooth()+
  theme(axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust=1))

c <- covid_mexico %>%
  filter(date >= '2021-01-01',
         date <= '2021-12-31') %>%
  select(total_tests, date) %>%
  ggplot(aes(y = total_tests, x = date)) +
  geom_line() +
  geom_smooth()+
  theme(axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust=1))
```

A continuación, se utilizará la función *ggarrange()* para ordenar estos elementos.

```
ggarrange(a, b, c,
          labels = c('CASOS', 'MUERTES', 'PRUEBAS'),
          ncol = 3, nrow = 1,
          font.label = list(size = 10),
          hjust = 0, vjust = 2.2)
```



De esta manera, se visualiza que las tres variables presentan una línea de tendencia similar durante el 2021.