



# R4Finance

Joining 2 & Workshop

J. Gibrán Peniche

Versión 0.0.1

2020/07/16

 [jgpeniche](https://github.com/jgpeniche)

 [PenicheGibran](https://twitter.com/PenicheGibran)

[G.jgpeniche@gmail.com](mailto:G.jgpeniche@gmail.com)

# Recap

- Aprendimos a utilizar `left_join()`, `right_join()`, `full_join()`, `anti_join()` para "consolidar" bases de datos
- Aprendimos la estructura básica para construir una visualización partiendo de un `df` en formato *tidy*

```
df %>%  
  ggplot2() +  
  aes() +  
  geom_something() +  
  labs() +  
  theme()
```

- Aprendimos como cambiar algunos atributos de color básicos, y a usar los temas de la librería `ggthemes`

# Agenda

1. Relación entre df's en formato *long* y las visualizaciones
2. Facetas
3. Temas personalizados
4. Caso de Estudio

1

# Relación entre df's en formato *long* y las visualizaciones

# Variables cualitativas

- Existen dos tipos de información sobre las variables
  1. Cuantitativa: Que se relaciona con magnitudes (velocidad, peso, precio, retornos, etc)
  2. Cualitativa: Que se relaciona con atributos no medibles (color, tipo, condición física, etc)
- Como ya habíamos comentado el tipo de objeto en R que nos permite codificar texto como variables cualitativas son los factores

Esto tiene relevancia ya que ggplot2 tiene un tratamiento especial para este tipo de variables

# Un ejemplo práctico

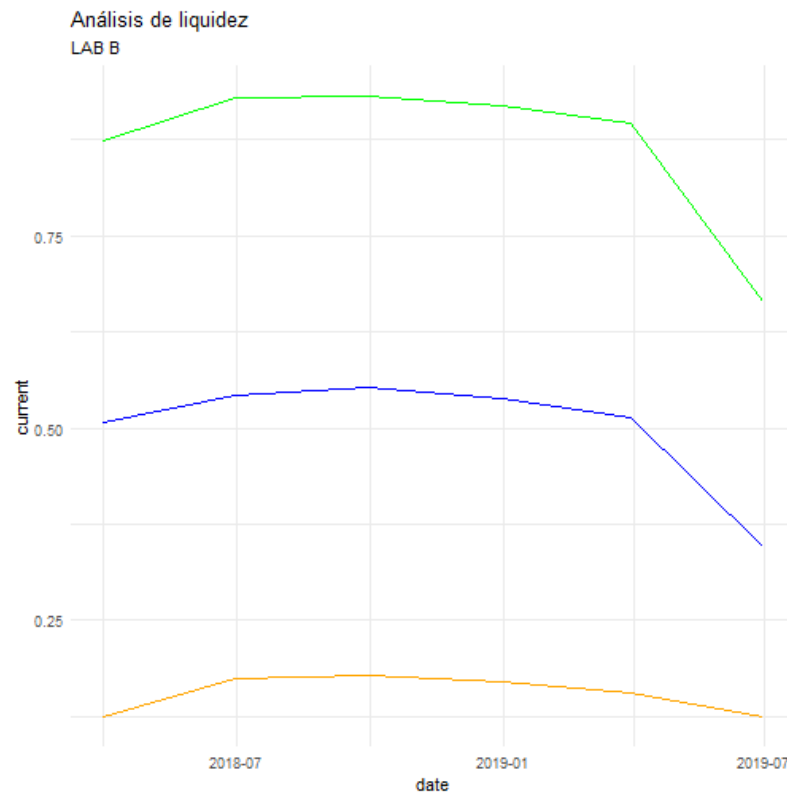
- Consideremos el siguiente df que contiene el análisis de liquidez para *LAB B* de la semana pasada

```
## Rows: 6
## Columns: 6
## $ date          <date> 2018-03-31, 2018-06-30, 2018-09-30, 2018-12-31, 2...
## $ current       <dbl> 0.8744778, 0.9292701, 0.9310768, 0.9200394, 0.8964...
## $ acid_test     <dbl> 0.5067134, 0.5436615, 0.5525045, 0.5382059, 0.5140...
## $ cash2assets   <dbl> 0.1246119, 0.1743048, 0.1774276, 0.1690932, 0.1544...
## $ cash2liabilities <dbl> 0.10897030, 0.16197624, 0.16519876, 0.15557242, 0....
## $ analisis      <chr> "Liquidéz", "Liquidéz", "Liquidéz", "Liquidéz", "L..."
```

# Un ejemplo práctico

- A partir de esta información quisieramos contruir un gráfico de líneas con la evolución histórica de las razones de liquidez

```
liq %>%  
  ggplot() +  
    aes(x = date) +  
    geom_line( aes(y = current), col = 'green') +  
    geom_line( aes(y = acid_test), col = 'blue') +  
    geom_line( aes(y = cash2assets), col = 'orange') +  
    labs(title = 'Análisis de liquidez',  
          subtitle = 'LAB B') +  
    theme_minimal()
```



# Un ejemplo práctico

**PREGUNTA:** ¿Existe alguna manera más sencilla de crear esta visualización y además agregar una leyenda automática?

**R:** Consideren el siguiente long df

```
liq_long <- liq %>%  
  pivot_longer(-c(date, analisis), names_to = 'ratio') %>%  
  mutate(ratio = factor(ratio))  
  
liq_long %>% glimpse()
```

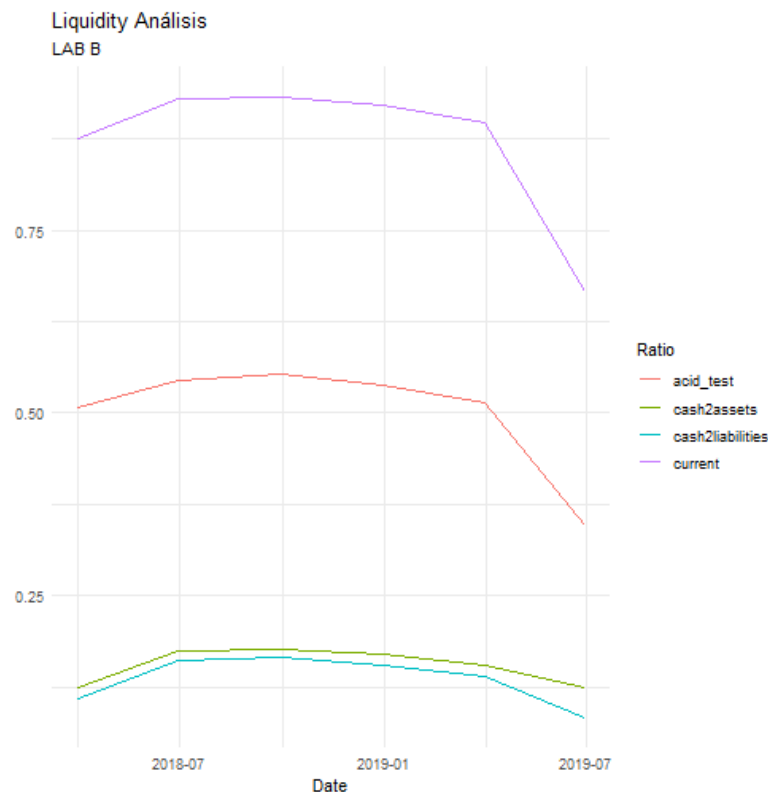
```
## Rows: 24  
## Columns: 4  
## $ date      <date> 2018-03-31, 2018-03-31, 2018-03-31, 2018-03-31, 2018-06-3...  
## $ analisis  <chr> "Liquidéz", "Liquidéz", "Liquidéz", "Liquidéz", "Liquidéz"...  
## $ ratio     <fct> current, acid_test, cash2assets, cash2liabilities, current...  
## $ value     <dbl> 0.87447779, 0.50671339, 0.12461185, 0.10897030, 0.92927005...
```



# Un ejemplo práctico

y ahora con ggplot

```
liq_long %>%  
  ggplot() +  
  aes(x = date, y = value, col = ratio) +  
  geom_line() +  
  labs(title = 'Liquidity Análisis',  
        subtitle = 'LAB B',  
        x = 'Date',  
        y = '',  
        col = 'Ratio') +  
  theme_minimal()
```



# Ejemplo práctico

De esta manera, utilizando datos en formato long (que en esta ocasión coinciden con el formato tidy), podemos generar legends automáticas

2

Facetas

# Más usos para las variables cualitativas

- Consideremos los objetos `rot_long` y `lev_long` contruidos de la misma manera que `liq_long`

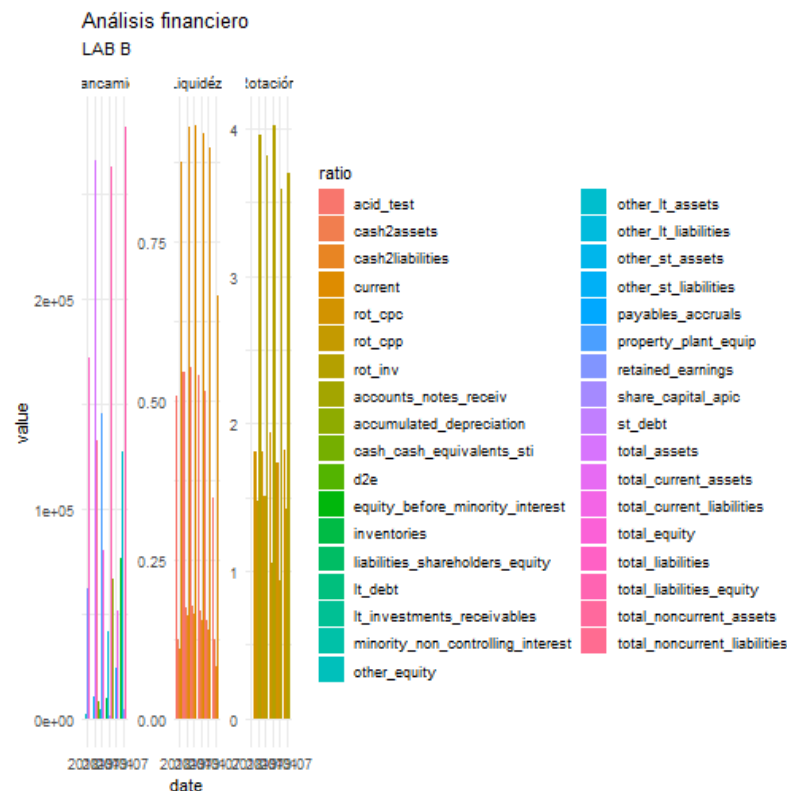
```
ratios <- liq_long %>%  
  bind_rows(rot_long) %>%  
  bind_rows(lev_long)  
  
ratios %>% glimpse()
```

```
## Rows: 210  
## Columns: 4  
## $ date      <date> 2018-03-31, 2018-03-31, 2018-03-31, 2018-03-31, 2018-06-3...  
## $ analisis  <chr> "Liquidéz", "Liquidéz", "Liquidéz", "Liquidéz", "Liquidéz"...  
## $ ratio     <fct> current, acid_test, cash2assets, cash2liabilities, current...  
## $ value     <dbl> 0.87447779, 0.50671339, 0.12461185, 0.10897030, 0.92927005...
```

- Estaría increíble crear un sola visualización con todos los análisis, esto se realiza con `facet_wrap()` utilizando la variable cualitativa *análisis*

# Más usos para las variables cualitativas

```
ratios %>%  
  ggplot() +  
  aes(x = date, y = value, fill = ratio) +  
  geom_bar(position = 'dodge', stat = 'identity') +  
  facet_wrap(~ analisis, scales = 'free') +  
  labs(title = 'Análisis financiero',  
        subtitle = 'LAB B') +  
  theme_minimal()
```



# patchwork

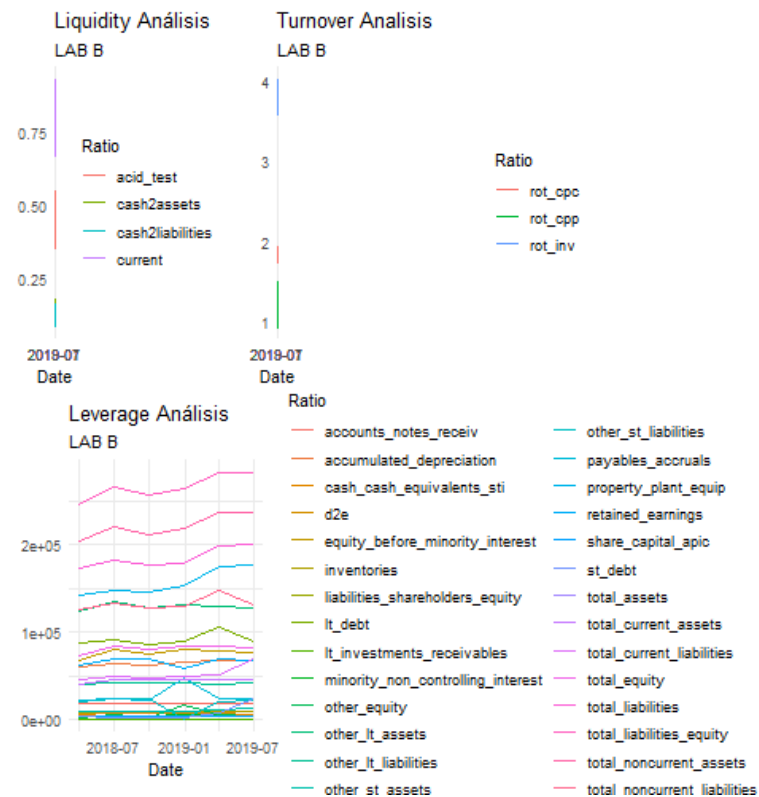
Otra opción sería guardar cada visualización en un objeto y luego desplegarlas en cierto orden, para eso usamos la librería patchwork

```
p1 <- liq_long %>%  
  ggplot() +  
  aes(x = date, y = value, col = ratio) +  
  geom_line() +  
  labs(title = 'Liquidity Análisis',  
        subtitle = 'LAB B',  
        x = 'Date',  
        y = '',  
        col = 'Ratio') +  
  theme_minimal()
```

```
p2 <- rot_long %>%  
  ggplot() +  
  aes(x = date, y = value, col = ratio) +  
  geom_line() +  
  labs(title = 'Turnover Analisis',  
        subtitle = 'LAB B',  
        x = 'Date',  
        y = '',
```

# patchwork

```
library(patchwork)
(p1 + p2)/p3
```



4

## Caso de estudio



# Caso de Estudio

1. Crear una nueva carpeta en tus documentos llamada `analisis_solvenscia_tu_nombre`
  - Dentro de esta carpeta crea un nuevo folder llamado `data` y copia en este directorio el archivo `template.xlsx` con el que hemos estado trabajando
2. Abre RStudio y crea un nuevo proyecto en la carpeta que acabas de crear que se llame `analisis_solvenscia`
3. En RStudio File > New File > RMarkdown, esto abrirá un diálogo
  - Title: 'Análisis de Solvenscia Genoma Lab'
  - Author: Tu nombre
  - Verifica que la opción **HTML** esté seleccionada
  - Click en *Ok*

1. Esto va a generar un nuevo archivo en el editor, oprime la opción **knit** y guardalo en la misma carpeta del proyecto con el nombre **reporte\_labb**. Nota que al guardar se genera una ventana nueva donde se despliega un documento
2. Regresa al editor de RStudio e inspecciona el código del archivo
3. Importa las pestañas `solv`, `solv2` y `solv3` a R, pásalas a formato long (`tidyr::pivot_wider( )`) y trnasforma los datos al tipo de información correcto (fecha a objetos tipo *date* y variables cualitativas a tipo *factor*). Para checar cual es el key-shortcut para crear code-chunks: Tools > Keyboard Shortcut help y busca `insert chunk`
4. Ya que los tres objetos estén en formato long, crea un nuevo objeto llamado `solvency` utilizando `bind_rows`
5. Redacta una breve introducción al documento de que es un análisis de solvencia de acuerdo al capítulo 12 del libro que les compartí
6. Realiza una visulización de los márgenes de solvencia para Genoma Lab en formato ejecutivo. Ojo la leyenda debe estar en el fondo, y los colores del título y subtítulo deben de ser diferentes (HINT: googlealo 'change title color ggplot2')

1. Vuelve a oprimir *knit* y verifica que no haya ningún error, ojo el documento html **unicamente debe incluir la introducción y el gráfico**, no debe aparecer todo el código de limpieza, ni el que generó el plot
2. Comprime la carpeta y envíame el .zip