Visualización de Datos

Visualización de datos en R

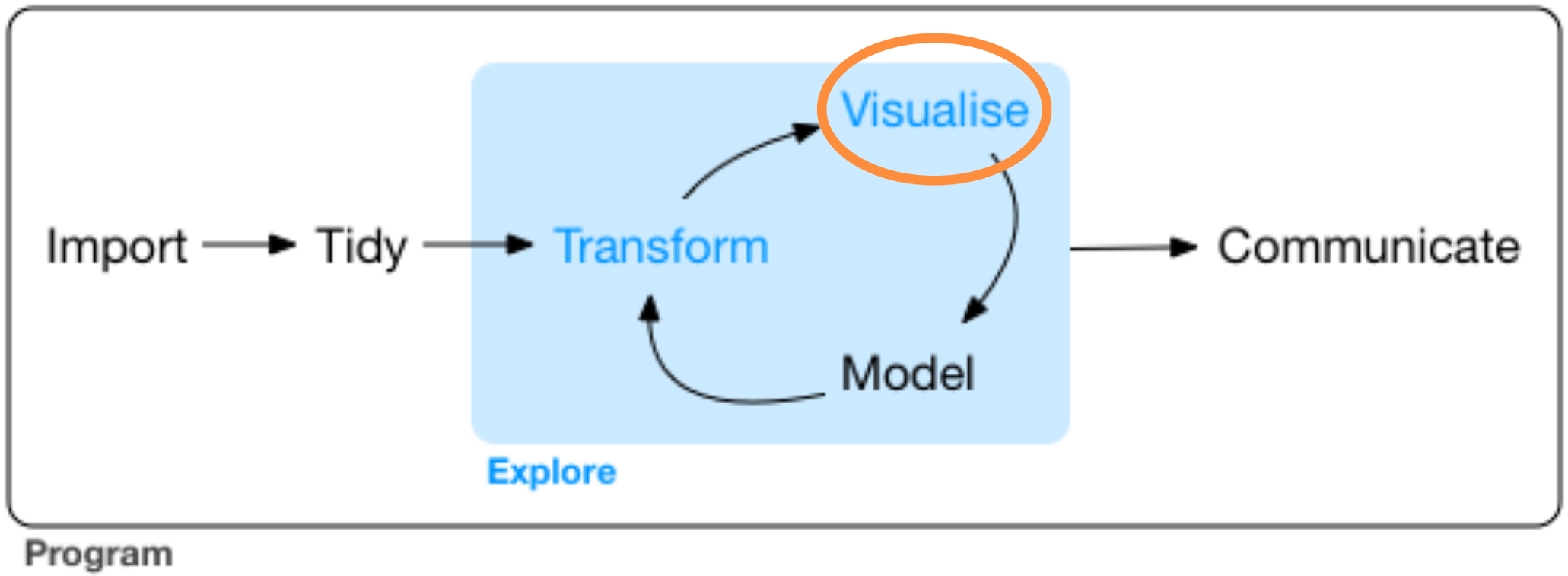
Juvenal Campos - 23 de Julio, 2021 - LNPP Tech

Segunda parte

La primera parte de esta presentación se puede consultar en este enlace:

https://juvecampos.github.io/presentaciones/lnpp_tech/presentacion_dataviz.pdf

Flujo de trabajo con datos



Hadley Wickham - R for data science
https://r4ds.had.co.nz/explore-intro.html

Visualización de datos en R

* **El lenguaje de programación *R* provee librerías y herramientas para visualizar datos.** (Al igual que muchos otros lenguajes de programación, como Python o Julia o programas como Excel o Tableau).

* Las **ventajas** que ofrece realizar visualizaciones de R consisten en:

* Alta personalización de todos los elementos de nuestras gráficas.

* La posibilidad de trabajar todo en un solo programa (importación, exploración, limpieza, modelación, visualización y comunicación).

* La posibilidad de generar visualizaciones interactivas traduciendo código de R a HTML-CSS-JS y de generar aplicaciones web tipo shiny o documentos Markdown.

https://www.data-to-viz.com/caveats.html

¿Por qué usar código para visualizar?

Las **ventajas** que trae realizar visualizaciones con código son las siguientes:

***Replicabilidad:** Al registrar todos los pasos necesarios para realizar una gráfica, esta se puede replicar.

***Podemos reciclar trabajo:** Muchos de los elementos utilizados para realizar una visualización (principalmente elementos estéticos) se pueden reciclar y adjuntar de manera sencilla en otras visualizaciones.

* Podemos adjuntar nuestro proceso en un **loop** y generar o modificar múltiples gráficas de manera rápida.

* Podemos adaptar el código a **conjuntos de datos que van cambiando en el tiempo.**

* Al no ser una herramienta de apuntar/dar click, **no requerimos memorizarnos todos los pasos** para realizar una gráfica.

https://www.northeastern.edu/graduate/blog/r-vs-excel/

Paquetes de R para visualización

Los paquetes de R para visualización pueden incluir (mas no limitarse) a los siguientes:

Visualización estática:

- **base:** la librería base de R para hacer gráficas.
- **lattice:** librería que mejora las capacidades de r-base.
- **ggplot2:** la librería del tidyverse para hacer gráficas altamente personalizables.

Visualización interactiva:

- **plotly**, para gráficos interactivos básicos y compatible con ggplot2
- **Highcharter**, port de highcharts.js para R.
- **Amcharts**, visualización interactiva.
- **htmlwidgets**, familia de librerías para hacer visualización interactiva.
- **dygraphs,** para gráficas interactivas (especialmente Series de tiempo).
- **r2d3**, librería para hacer algunas gráficas de D3.js en R.

Mapas:

- **leaflet:** Mapas interactivos de leaflet.js
- También se pueden usar ggplot2, plotly o highcharter

Colores:

- **RColorBrewer,** paletas de colores.
- **viridis**, paletas de colores en R para personas con daltonismo.

Tablas:

- **DT:** Interface de la librería DataTable.js para R.
- **kableExtra:** Generar tablas con formato en HTML
- **formattable:** Tablas para presentación.
- **reactable:** Tablas con gráficas

Redes:

- **Igraph,** para análisis y visualización básica de redes.
- **networkD3,** para redes interactivas y sankeys.

ggplot2

* Es la librería del *tidyverse* más utilizada para generar gráficas en R.

* Utiliza la gramática de gráficas para estandarizar la forma de generar gráficas.

La programación de las gráficas requiere lo siguiente:

1) Un objeto tibble/dataframe (base de datos tabular) para darle la información.

2) Inicializar el lienzo con la función ggplot()

3) Definir los elementos estéticos (canales)

4) Definir la(s) geometría(s) o el tipo de gráfica a utilizar (marcas)

5) Modificar elementos de la gráfica, como etiquetas, formato de los ejes, sistema de coordenadas, límites de los valores, etc.

6) Opcional - Definir temas y añadidos estéticos para que la gráfica se vea bonita.

1. Objeto tibble/dataframe

1) **Un objeto tibble/dataframe (base de datos tabular) para darle la información.**

* Este objeto debe tener exactamente los datos que se requieren para la gráfica.

* Para llegar a estos datos, hay que limpiar y procesar las bases de datos originales.

* A veces es necesario generar columnas de colores o de etiquetas con formato.

2. Inicializar el lienzo

2) Inicializar el lienzo con la función ggplot()

* Le pasamos el tibble del paso 1) a la función ggplot() para que se inicie el lienzo.

* A partir de este punto las capas de la gráfica se van ligando con el símbolo de "+".

* Dentro de ggplot() pueden irse incorporando los elementos estéticos.

3. Definir los elementos estéticos

3) Definir los elementos estéticos (canales)

* En este punto definimos las variables que van a regular la apariencia de las *marcas* (o los elementos de las gráficas).

* Solo regulamos las variables que van a definir la apariencia, no como la van a definir (esto se hace en un paso más adelante).

* Los elementos estéticos a modificar son (pero no se limitan a) **la posición (coordenada *x* o *y*), el color, el relleno de una forma, el grosor de línea, el tamaño del punto, la transparencia, la forma que toma un punto o una línea, el grupo al que pertenece, etc.**

* Solo podemos modificar o definir los elementos estéticos que modifican al tipo de gráfica que podemos utilizar.

* Se pueden definir desde la función ggplot() si van a afectar a todas las geometrías, o dentro de las geometrías si solo van a afectar a la geometría particular.

4. Definir geometrías o marcas

4) Definir la geometría o el tipo de gráfica a utilizar (marcas)

* En este paso se decide qué marca utilizar (que geometría utilizar).

* Cada geometría requiere los elementos estéticos mínimos para poder funcionar (por ejemplo, si vas a usar puntos en 2D se requiere que al menos definas la coordenada ***x*** y ***y***), si no, no va a funcionar.

* Las geometrías más comunes se pueden consultar acá:

    * https://ggplot2.tidyverse.org/reference/

5. Modificar elementos de la gráfica

5) Modificar elementos de la gráfica, como etiquetas, formato de los ejes, sistema de coordenadas, límites de los valores, etc.

* Existen funciones para modificar estos elementos de la gráfica.

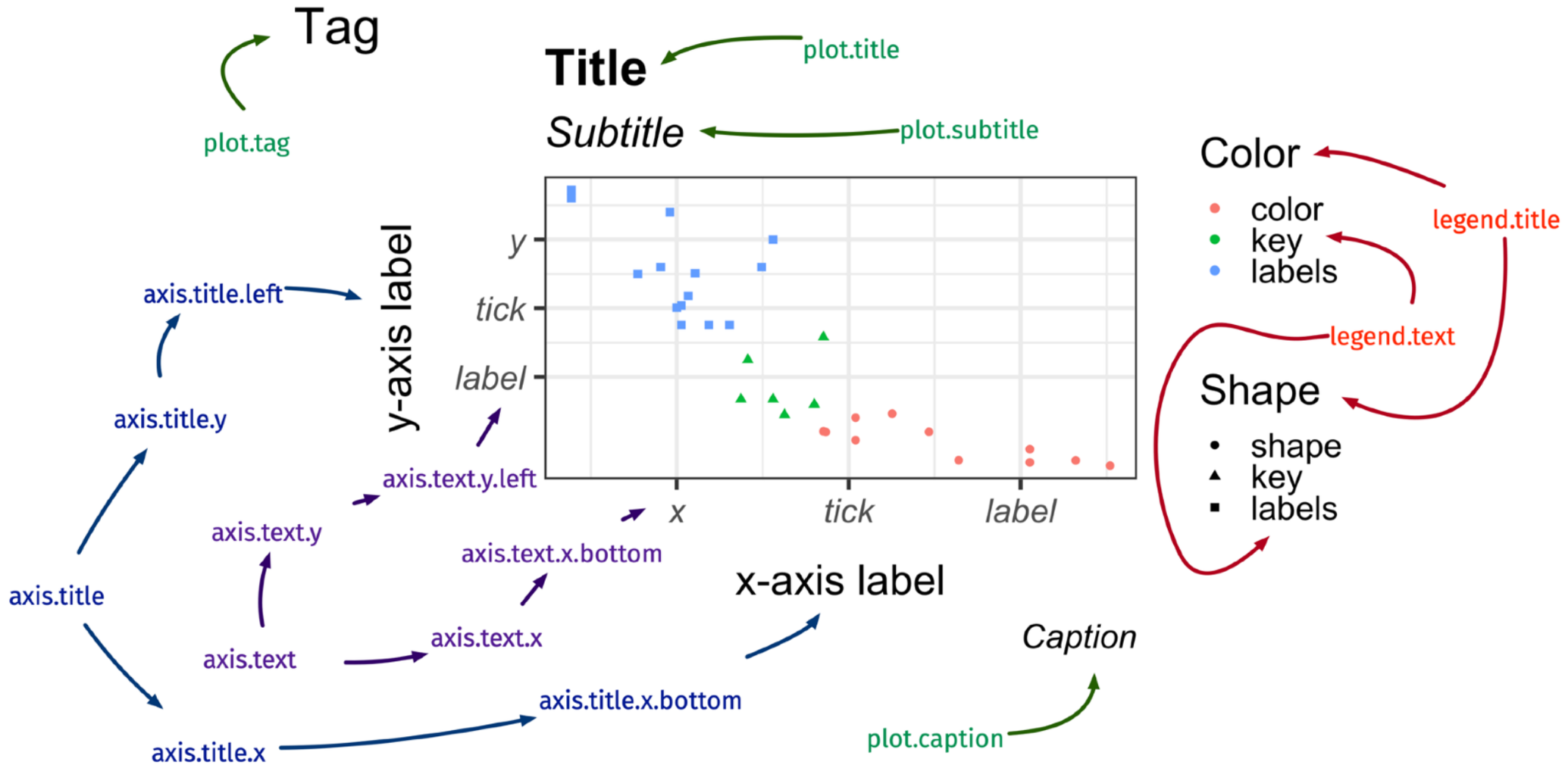
* Para modificar los límites o los textos del eje x utilizamos las funciones scale_x_*()

* Para modificar los colores, scale_fill/color_*()

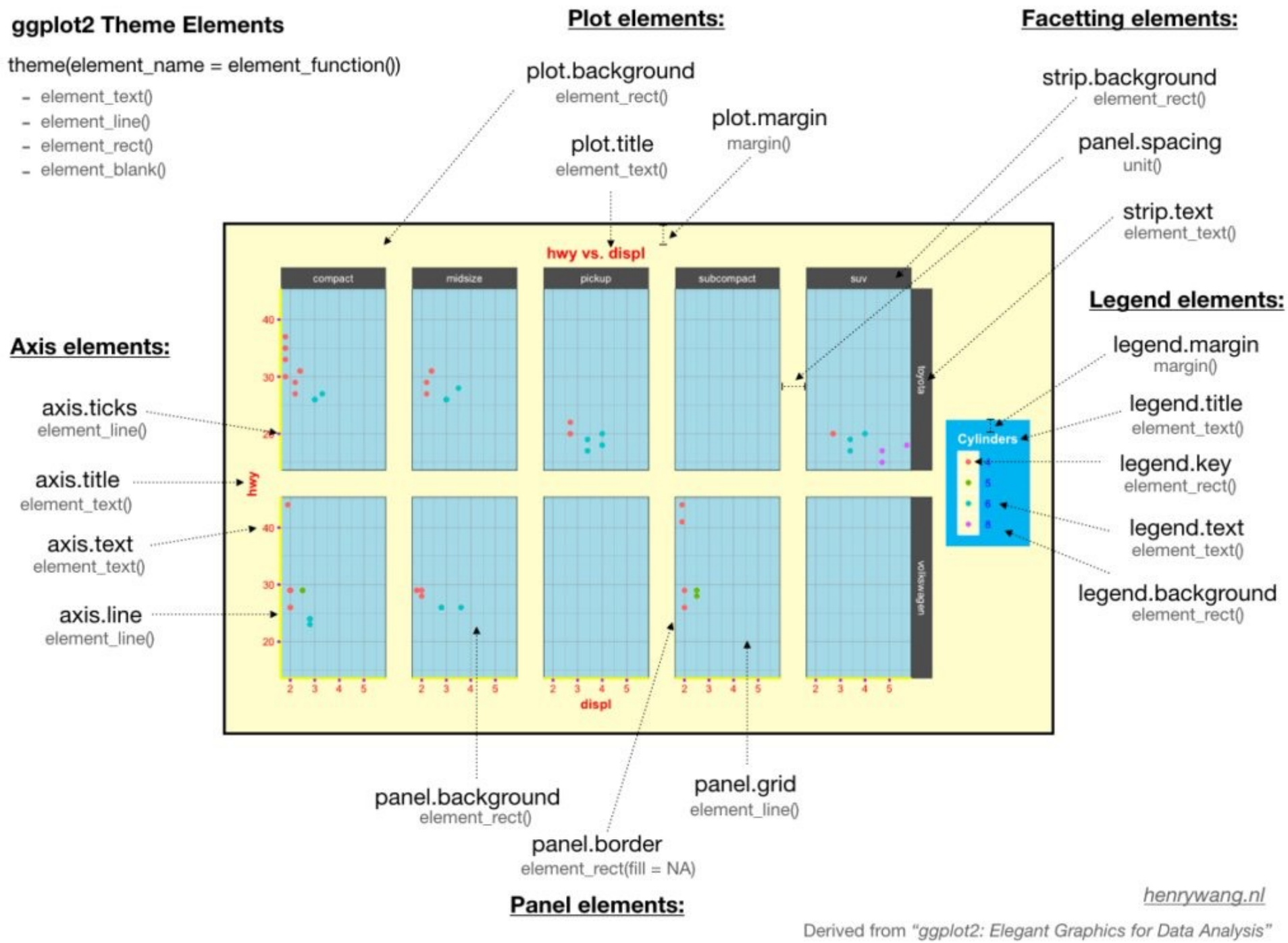
* Para añadir etiquetas, utilizamos la función ***labs()*** para modificar títulos, subtítulos, tags, pie de página, títulos de los ejes y de las leyendas.

6.Temas: lo que se puede modificar

6) Opcional - Definir temas y añadidos estéticos para que la gráfica se vea bonita.



6.Temas: lo que se puede modificar



Tips

1. **Define bien tus colores** y tus paletas de colores.

2. **Ten a la mano los iconos que utilices más a menudo**, y dedícale el tiempo necesario a tener tu carpeta de ícenos más utilizados (logos de tu empresa, de empresas con las que trabajes, partidos políticos, administraciones estatales, etc.)

3. Igualmente, **ten un repositorio de archivos que uses comúnmente.** Por ejemplo, para mapas yo tengo este repositorio de archivos: https://github.com/JuveCampos/Shapes_Resiliencia_CDMX_CIDE listos para crear mapas.

4. **Aprende como generar y subir páginas** en las cuales montar tus visualizaciones interactivas (Rmarkdown, Github Pages, shiny).

5. Trata de evitarlo, pero si no puedes, **hecha mano de la post-producción para editar tus gráficas** (por ejemplo, photoshop, illustrator o power point).

Más tips

1. Un buen ejercicio para empezar a practicar es el **replicar otras gráficas de otros programadores y usuarios.**

2. No te tienen que salir perfectas; recuerda que lo que te falte por hacer lo puedes corregir o añadir utilizando post-producción.

3. Si estas en ceros en R, te recomiendo que sepas: r-base, manejo de factores o variables categóricas, manejo de datos con el tidyverse, colores (rgb, hexadecimales) y pivoteo de bases (formato ancho y largo).

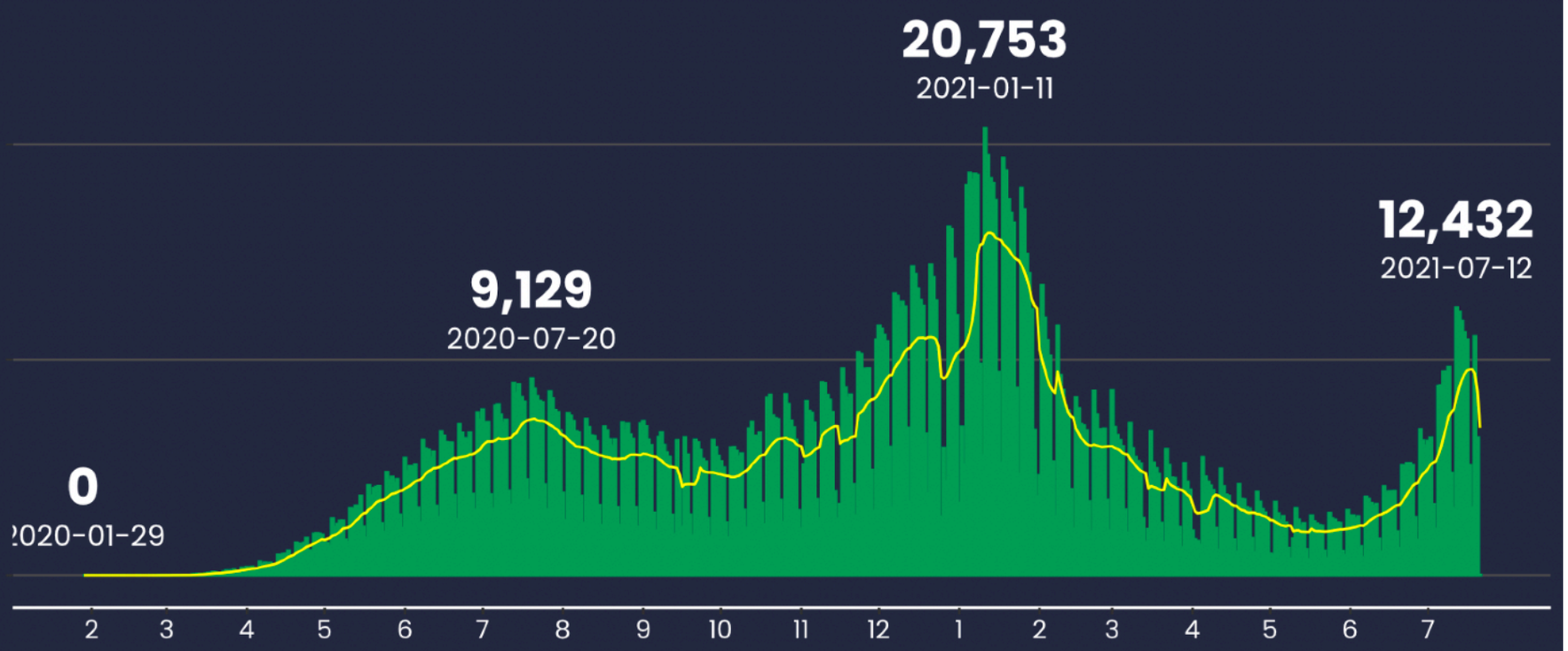
4. Si requieres **replicar una fuente** (tipo de letra) dada, usa la página https://www.myfonts.com/WhatTheFont/ para guiarte y Google Fonts para descargar letras.

Numero de casos nuevos por dia de coronavirus en México

Covid-19 en México | NUMERO DE CASOS NUEVOS POR DÍA

Acumulados: 2’693,495

Defunciones: 237,207



Nota: "La Secretaría de Salud agregó el 5 de octubre datos históricos rezagados desde el inicio de la pandemia en México, tras incorporar la confirmación por dictaminación y asociación epidemiológica para el diagnóstico de Covid-19. 2 El de junio de 2021, la Secretaría de Salud agregó datos pendientes dictaminación técnica para el diagnóstico de coronavirus. En el caso de decesos, reportó 3,924 correspondientes a 2020 y 348 a semanas recientes.

FUENTE: SECRETARÍA DE SALUD

Laboratorio Nacional de Políticas Públicas

Gracias :3