

Módulo 5: Visualización de Datos con ggplot2

02 de July de 2025

Introducción a ggplot2

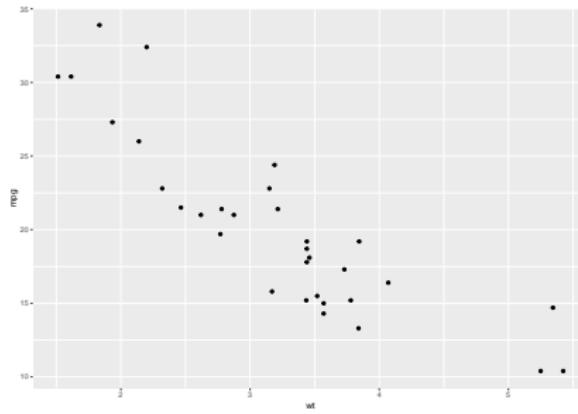
- ggplot2 es un sistema para crear gráficos de forma declarativa, basado en la “Gramática de Gráficos”. Esto significa que construyes tus visualizaciones especificando componentes en lugar de los pasos exactos para dibujarlos.
- Gramática de gráficos:
 - Datos (data): El data frame con la información que quieras visualizar.
 - Mapeos estéticos (aes()): Define cómo las variables de tus datos se traducen en propiedades visuales (ejes x, y, color, tamaño, forma, etc.).
 - Geometrías (geom_*) : El tipo de gráfico que quieras dibujar (puntos, líneas, barras, etc.).
 - Facetas, escalas, temas: Para personalizar y dividir el gráfico.

Estructura Básica de ggplot2

- La estructura fundamental de un gráfico en ggplot2 es simple: empiezas con la función `ggplot()`, especificas los datos y los mapeos estéticos, y luego añades capas usando `+`.

```
ggplot(data = df, aes(x = var1, y = var2)) + geom_point()
```

- Todo lo que vaya unido con `+` agrega capas o mejoras.



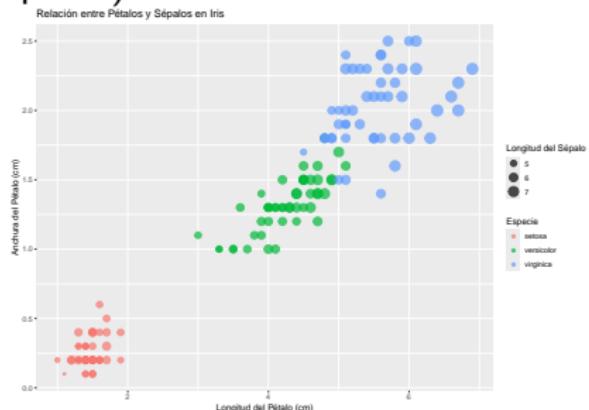
Gráfica de Dispersion (geom_point())

- Mapea variables continuas en los ejes x e y.
- Se pueden añadir color, tamaño o forma mapeando variables a estos estéticos.
- alpha controla la transparencia de los puntos.

Tipos de Gráficas Básicas

Gráfica de Dispersion (geom_point())

- Usamos el dataset 'iris'
- ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species, size = Sepal.Length)) + geom_point(alpha = 0.7) + labs(title = "Relación entre Pétalos y Sépalos en Iris", x = "Longitud del Pétalo (cm)", y = "Anchura del Pétalo (cm)", color = "Especie", size = "Longitud del Sépalo")



Tipos de Gráficas Básicas

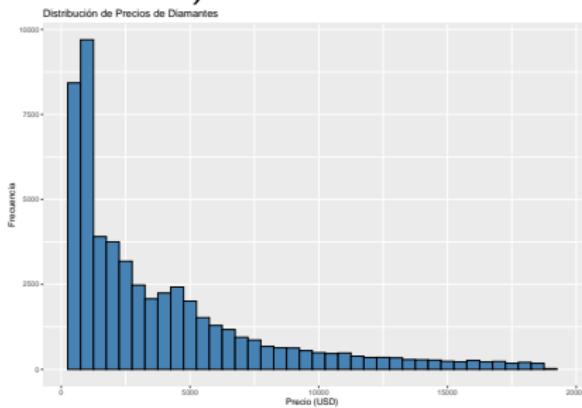
Histograma (geom_histogram()) y Densidad (geom_density())

- Histograma: Muestra la distribución de una variable numérica dividiendo sus valores en “bins” y contando las observaciones en cada uno.
- Densidad: Estima la función de densidad de probabilidad de una variable, suavizando la distribución.

Tipos de Gráficas Básicas

Histograma (geom_histogram()) y Densidad (geom_density())

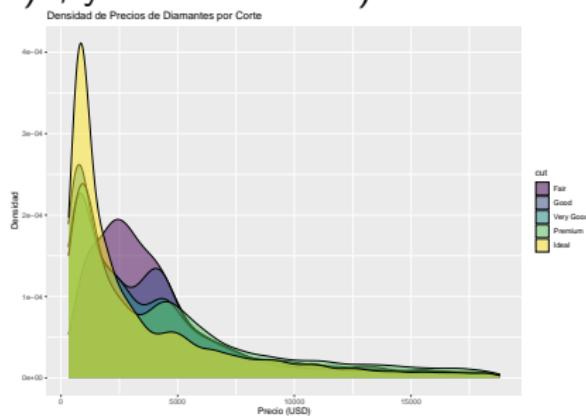
- Usamos el dataset 'diamonds'
- Histograma del precio ggplot(diamonds, aes(x = price)) +
geom_histogram(binwidth = 500, fill = "steelblue", color = "black")
+ labs(title = "Distribución de Precios de Diamantes", x = "Precio
(USD)", y = "Frecuencia")



Tipos de Gráficas Básicas

Histograma (geom_histogram()) y Densidad (geom_density())

- Gráfico de densidad del precio, diferenciado por el tipo de corte (cut)
`ggplot(diamonds, aes(x = price, fill = cut)) + geom_density(alpha = 0.5) + labs(title = "Densidad de Precios de Diamantes por Corte", x = "Precio (USD)", y = "Densidad")`



Tipos de Gráficas Básicas

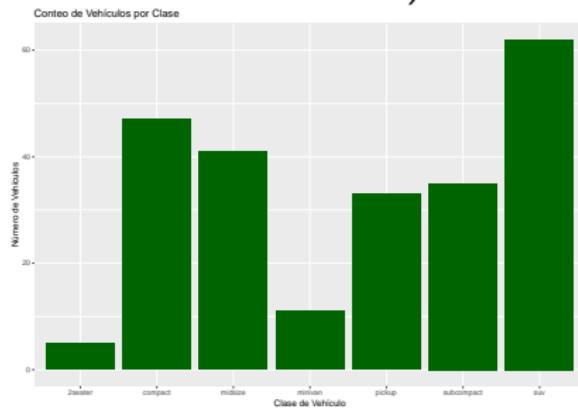
Histograma (geom_histogram()) y Densidad (geom_density())

- Gráfico de Barras (geom_bar() y geom_col())
 - geom_bar(): Cuenta el número de casos en cada categoría de una variable discreta. Útil para frecuencias.
 - geom_col(): Se usa cuando ya tienes una variable numérica que representa la altura de las barras (valores precalculados).

Tipos de Gráficas Básicas

Histograma (geom_histogram()) y Densidad (geom_density())

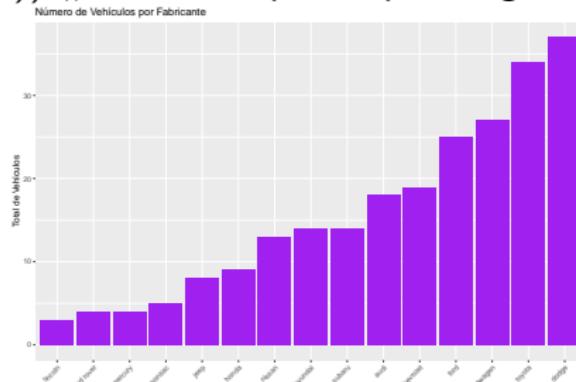
- Usamos el dataset 'mpg'
- Conteo de vehículos por tipo de clase
- `ggplot(mpg, aes(x = class)) + geom_bar(fill = "darkgreen") + labs(title = "Conteo de Vehículos por Clase", x = "Clase de Vehículo", y = "Número de Vehículos")`



Tipos de Gráficas Básicas

Histograma (geom_histogram()) y Densidad (geom_density())

- Si ya tenemos una variable numérica para la altura, usar geom_col()
- df_resumen_manuf <- mpg %>% group_by(manufacturer) %>% summarise(n = n(), .groups = "drop")
- ggplot(df_resumen_manuf, aes(x = reorder(manufacturer, n), y = n)) + # Reordenar por frecuencia geom_col(fill = "purple") + labs(title = "Número de Vehículos por Fabricante", x = "Fabricante", y = "Total de Vehículos") + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotar etiquetas para legibilidad



Tipos de Gráficas Básicas

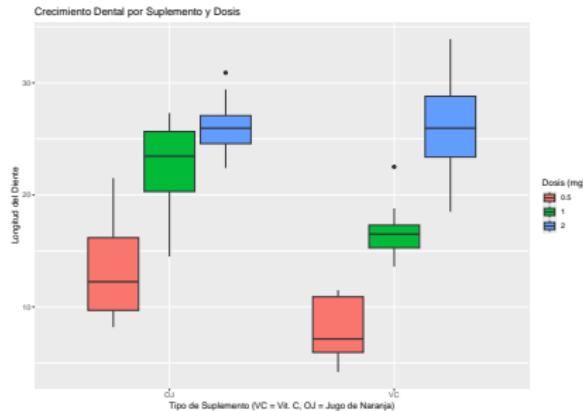
Boxplot (geom_boxplot())

- Compara la distribución de una variable numérica entre diferentes categorías.
- Muestra la mediana, cuartiles y posibles valores atípicos.

Tipos de Gráficas Básicas

Boxplot (geom_boxplot())

- Usamos el dataset 'ToothGrowth'
- ggplot(ToothGrowth, aes(x = factor(supp), y = len, fill = factor(dose))) + geom_boxplot() + labs(title = "Crecimiento Dental por Suplemento y Dosis", x = "Tipo de Suplemento (VC = Vit. C, OJ = Jugo de Naranja)", y = "Longitud del Diente", fill = "Dosis (mg)")



Tipos de Gráficas Básicas

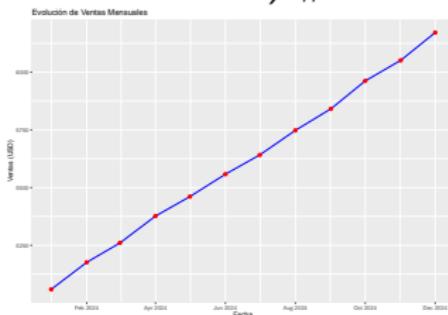
Líneas (geom_line())

- Ideal para representar series temporales o la evolución de una variable a lo largo de otra continua ordenada.

Tipos de Gráficas Básicas

Crear un df de ejemplo con datos de ventas simulados

- df_ventas_simulado <- tibble(Fecha = seq(as.Date("2024-01-01"), by = "month", length.out = 12), Ventas = cumsum(rnorm(12, mean = 100, sd = 20)) + 5000)
- ggplot(df_ventas_simulado, aes(x = Fecha, y = Ventas)) + geom_line(color = "blue", size = 1) + geom_point(color = "red", size = 2) + labs(title = "Evolución de Ventas Mensuales", x = "Fecha", y = "Ventas (USD)") + scale_x_date(date_breaks = "2 months", date_labels = "%b %Y") # Formato de fecha

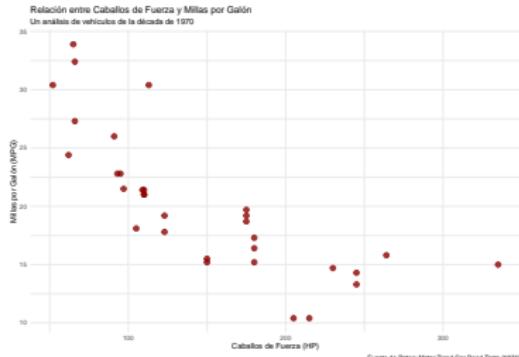


Personalización

Etiquetas y Títulos (labs())

labs() permite añadir y modificar fácilmente títulos, subtítulos, pies de página y etiquetas de ejes.

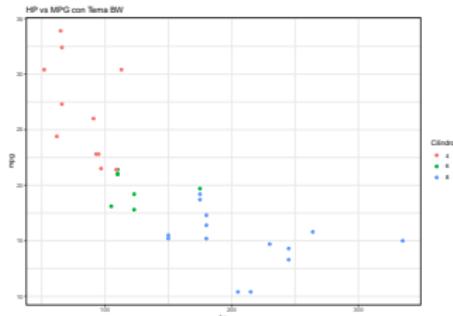
```
ggplot(mtcars, aes(x = hp, y = mpg)) + geom_point(color = "darkred",  
size = 3, alpha = 0.8) + labs( title = "Relación entre Caballos de Fuerza  
y Millas por Galón", subtitle = "Un análisis de vehículos de la década de  
1970", x = "Caballos de Fuerza (HP)", y = "Millas por Galón (MPG)",  
caption = "Fuente de Datos: Motor Trend Car Road Tests (1974)" ) +  
theme_minimal()
```



Personalización

Temas (theme_*())

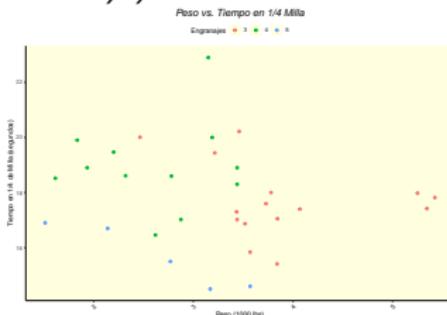
- Temas predefinidos: theme_minimal(), theme_bw(), theme_classic(), theme_light(), etc., cambian rápidamente la apariencia general del gráfico.
- Ajustes manuales con theme(): Permite un control granular sobre cada elemento del tema (texto, líneas, fondos, leyendas).
- Ejemplo con tema predefinido
- `ggplot(mtcars, aes(x = hp, y = mpg)) + geom_point(aes(color = factor(cyl))) + theme_bw() + # Tema en blanco y negro labs(title = "HP vs MPG con Tema BW", color = "Cilindros")`



Personalización

Temas (theme_*())

- Ajustes manuales con theme()
- ```
ggplot(mtcars, aes(x = wt, y = qsec)) + geom_point(aes(color = factor(gear))) + labs(title = "Peso vs. Tiempo en 1/4 Milla", x = "Peso (1000 lbs)", y = "Tiempo en 1/4 de Milla (segundos)", color = "Engranajes") + theme_classic() + theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10, face = "bold"), plot.title = element_text(size = 16, face = "italic", hjust = 0.5), legend.position = "top", panel.background = element_rect(fill = "lightyellow", color = NA))
```



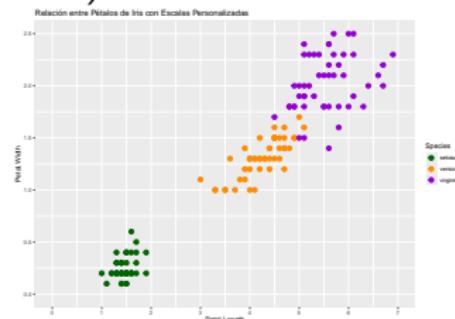
## Escalas (scale\_\*)()

- Las escalas controlan cómo los datos se mapean a los estéticos.
- Ejes: `scale_x_continuous()`, `scale_y_log10()`, `scale_x_date()`, etc., para controlar rangos, breaks, transformaciones.
- Colores/Rellenos: `scale_color_manual()`, `scale_fill_brewer()`, `scale_fill_viridis_d()`, etc., para definir paletas de colores o asignaciones manuales.

# Personalización

## Escalas (scale\_\*)()

- Usamos el dataset 'iris'
- ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species))  
+ geom\_point(size = 3) + scale\_x\_continuous(limits = c(0, 7),  
breaks = seq(0, 7, 1)) + # Limites y ticks en eje x  
scale\_y\_continuous(limits = c(0, 2.5), breaks = seq(0, 2.5, 0.5)) +  
# Limites y ticks en eje y scale\_color\_manual(values = c("setosa" =  
"darkgreen", "versicolor" = "darkorange", "virginica" = "darkviolet"))  
+ # Colores manuales labs(title = "Relación entre Pétalos de Iris con  
Escalas Personalizadas")



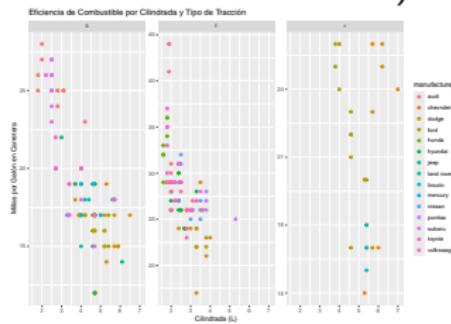
## Facetas (`facet_wrap()` y `facet_grid()`)

- Dividen el gráfico en múltiples paneles (o “facetas”) basados en una o dos variables categóricas.
  - `facet_wrap(~ var)`: Una dimensión, envuelve los paneles en filas/columnas.
  - `facet_grid(var_fila ~ var_columna)`: Dos dimensiones, crea una cuadrícula de paneles.

# Personalización

## Facetas (facet\_wrap() y facet\_grid())

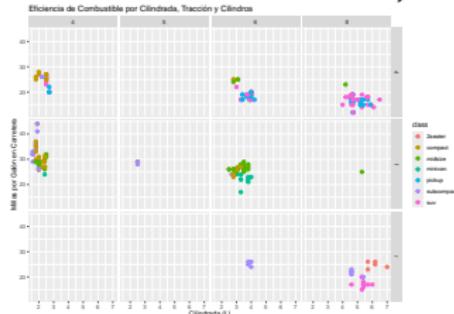
- Usamos el dataset 'mpg'
- Faceta unidimensional: relación displ-hwy por tipo de tracción (drv)
- ggplot(mpg, aes(x = displ, y = hwy)) + geom\_point(aes(color = manufacturer)) + facet\_wrap(~ drv, scales = "free\_y") + # 'scales = "free\_y"' permite escalas y en cada faceta labs(title = "Eficiencia de Combustible por Cilindrada y Tipo de Tracción", x = "Cilindrada (L)", y = "Millas por Galón en Carretera")



# Personalización

## Facetas (facet\_wrap() y facet\_grid())

- Faceta bidimensional: filas ~ drv, columnas ~ cyl (cilindros)
- `ggplot(mpg, aes(x = displ, y = hwy)) + geom_point(aes(color = class), size = 2) + facet_grid(drv ~ cyl) + labs(title = "Eficiencia de Combustible por Cilindrada, Tracción y Cilindros", x = "Cilindrada (L)", y = "Millas por Galón en Carretera")`



## Mapa de Calor (geom\_tile())

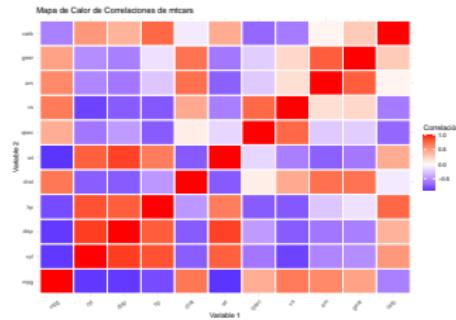
- Para variables de dos dimensiones ( $x$ ,  $y$ ) donde el color indica una tercera magnitud.
- Comúnmente usado para matrices de correlación o datos de tablas dinámicas.

# Gráficos Avanzados

## Mapa de Calor (geom\_tile())

- Ejemplo: matriz de correlaciones del dataset 'mtcars'

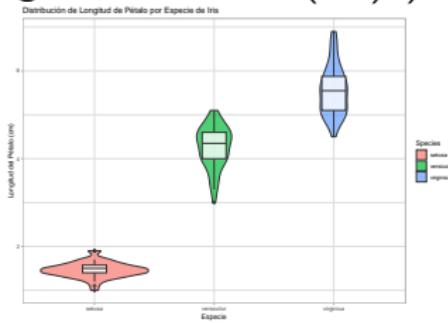
```
ggplot(df_cor, aes(x = Var1, y = Var2, fill = value)) + geom_tile(color = "white") + # Añadir bordes blancos para distinción
scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0, name = "Correlación") + labs(title = "Mapa de Calor de Correlaciones de mtcars", x = "Variable 1", y = "Variable 2") +
theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotar etiquetas
```



# Gráficos Avanzados

## Gráfico de Violín (geom\_violin())

- Visualiza la distribución de datos por categorías de forma similar a un boxplot, pero mostrando la densidad de los datos. Útil para ver la forma de la distribución.
- ```
ggplot(iris, aes(x = Species, y = Petal.Length, fill = Species)) +  
  geom_violin(alpha = 0.7) + geom_boxplot(width = 0.2, fill =  
  "white", alpha = 0.8) + # Añadir boxplot para ver cuartiles  
  labs(title = "Distribución de Longitud de Pétalo por Especie de Iris", x =  
  "Especie", y = "Longitud del Pétalo (cm)") + theme_light()
```



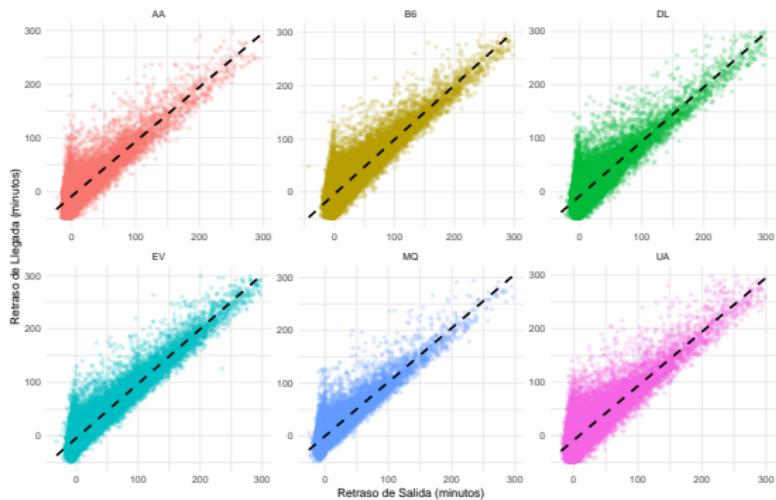
Tarea - Gráficos Avanzados

Gráfico de Áreas Apiladas (geom_area())

- Muestra la evolución de la contribución de diferentes categorías a un total a lo largo del tiempo o una variable continua.

Relación entre Retraso de Salida y Retraso de Llegada por Aerolínea

Datos de vuelos de NYC en 2013



Exportar Gráficos

Exportar Gráficos con ggsave()

- La función `ggsave()` es la forma más sencilla de guardar un gráfico de `ggplot2` directamente desde R.
- Guarda el último gráfico mostrado por defecto, o puedes especificar un objeto gráfico.

Argumentos clave de `ggsave()`:

- `filename`: Nombre del archivo de salida (ej. “mi_grafico.png”).
- `plot`: El objeto gráfico a guardar (por defecto, el último `ggplot` mostrado).
- `device`: Tipo de dispositivo (inferido por la extensión del `filename`, ej. “png”, “pdf”, “svg”, “jpeg”).
- `width, height`: Dimensiones del gráfico (por defecto en pulgadas).
- `dpi`: Resolución para formatos raster (PNG, JPG). Un valor común es 300 o 600.

¡Gracias!