

Visualización de datos en R

Daniel Sánchez Pazmiño

2024-09

Visualización de datos para variables cuantitativas

Gráficos con R base

- R base proporciona una serie de funciones para crear gráficos básicos
- Algunas de las funciones más comunes son:
 - `plot()` : gráfico de dispersión
 - `hist()` : histograma
 - `boxplot()` : diagrama de caja y bigotes

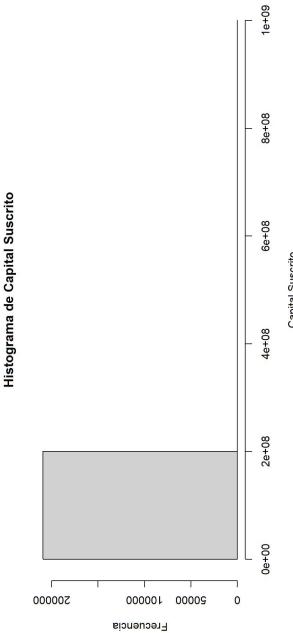
Ejemplo con R base: Histograma

- **Histograma:** muestra la distribución de una variable cuantitativa
- La distribución se divide en intervalos o “bins”, y se cuenta la frecuencia de datos en cada intervalo
- En R, se puede utilizar la función `hist()` para crear un histograma
- Se puede personalizar el número de intervalos con el argumento `breaks`

Creando un histograma de capital suscrito

- Utilizando la función `hist()` de R base

```
1 hist(supercias_limpio$capital_suscrito,  
2 breaks = 5,  
3 main = "Histograma de Capital Suscrito",  
4 xlab = "Capital Suscrito",  
5 ylab = "Frecuencia")
```



Creando un histograma de capital suscripto

- Una distribución con valores atípicos complica la visualización
- Se puede utilizar la escala logarítmica para mejorar la visualización

```
1 hist(log(supercias_limpio$capital_suscripto),  
2      breaks = 5,  
3      main = "Histograma de Capital Suscripto (log)",  
4      xlab = "Log (Capital Suscripto)",  
5      ylab = "Frecuencia")
```

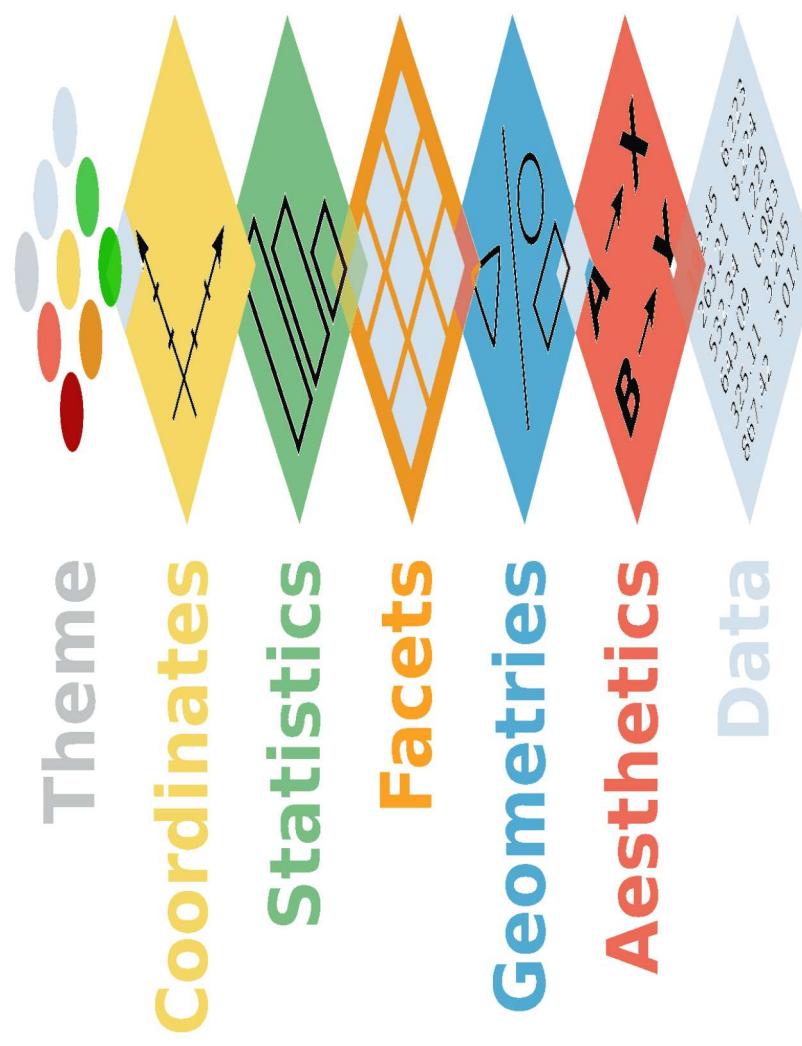
Gráficos con ggplot2

- `ggplot2` es una librería de R que permite crear gráficos de alta calidad y personalizables
- Utiliza el *Grammar of Graphics* para construir gráficos, definidos por capas diferentes

Gramática de Gráficos

- Todo gráfico, según el *Grammar of Graphics*, tiene siete posibles capas:
 - **Datos:** conjunto de datos a visualizar
 - **Estéticas:** mapeo de variables a atributos visuales (color, forma, tamaño, etc.)
 - **Geometrías:** tipo de gráfico (puntos, líneas, barras, etc.)
 - **Facetas:** subdivisión de los datos en subgráficos
 - **Estadísticas:** resumen de los datos (media, mediana, etc.)
 - **Coordenadas:** sistema de coordenadas (cartesiano, polar, etc.)
 - **Temas:** aspecto visual del gráfico (colores, fuentes, etc.)

Gramática de Gráficos



Creando un histograma con
graplotz

Replicando el histograma de capital suscrito con ggplot2

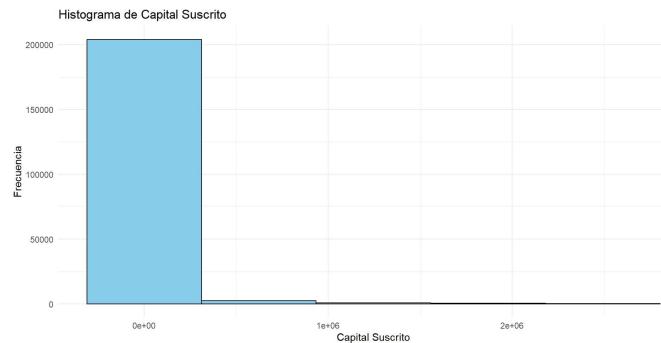
```
1 library(ggplot2)
2
3 ggplot(supercias_limpio, aes(x = capital_suscripto) ) +
4   geom_histogram(bins = 5) +
5   labs(title = "Histograma de Capital Suscrito",
6        x = "Capital Suscrito",
7        y = "Frecuencia")
```

Mejorando el histograma con ggplot2

- Este paquete permite una mayor personalización de los gráficos que R base
- Definimos la longitud de los bins, el color de las barras, el color de fondo, etc.
- El uso del pipe `%>%` permite encadenar funciones de manera más sencilla, combinando con `dplyr`

Histograma de capital suscrito mejorado con ggplot2

```
1 supercias_limpio %>%
2 filter(capital_suscrito > 0, capital_suscrito < 2500000) %>%
3 ggplot(aes(x = capital_suscrito)) +
4 geom_histogram(bins = 5, fill = "skyblue", color = "black") +
5 labs(title = "Histograma de Capital Suscrito",
6      x = "Capital Suscrito",
7      y = "Frecuencia") +
8 theme_minimal()
```



Eliminar valores atípicos con ggplot2

- Se pueden filtrar los valores atípicos para mejorar la visualización
- Se utiliza el rango intercuartil (IQR) para identificar los valores atípicos

$$[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$$

Histograma de capital suscrito sin valores atípicos

```
1 supercias_limpio %>%
2   filter(capital_suscripto > quantile(capital_suscripto, 0.25, na.rm = TRUE) < quantile(capital_suscripto, 0.75, na.rm = TRUE))
3   ggplot(aes(x = capital_suscripto)) +
4     geom_histogram(bins = 5, fill = "skyblue", color = "black") +
5     labs(title = "Histograma de Capital Suscrito",
6          x = "Capital Suscrito",
7          y = "Frecuencia") +
8          theme_minimal()
```

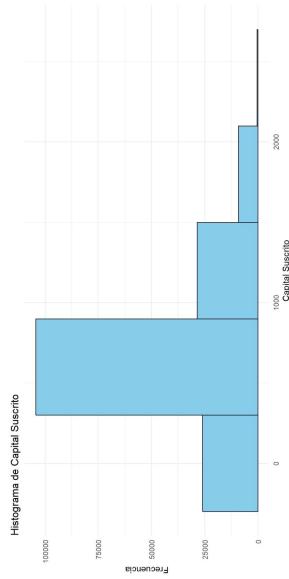
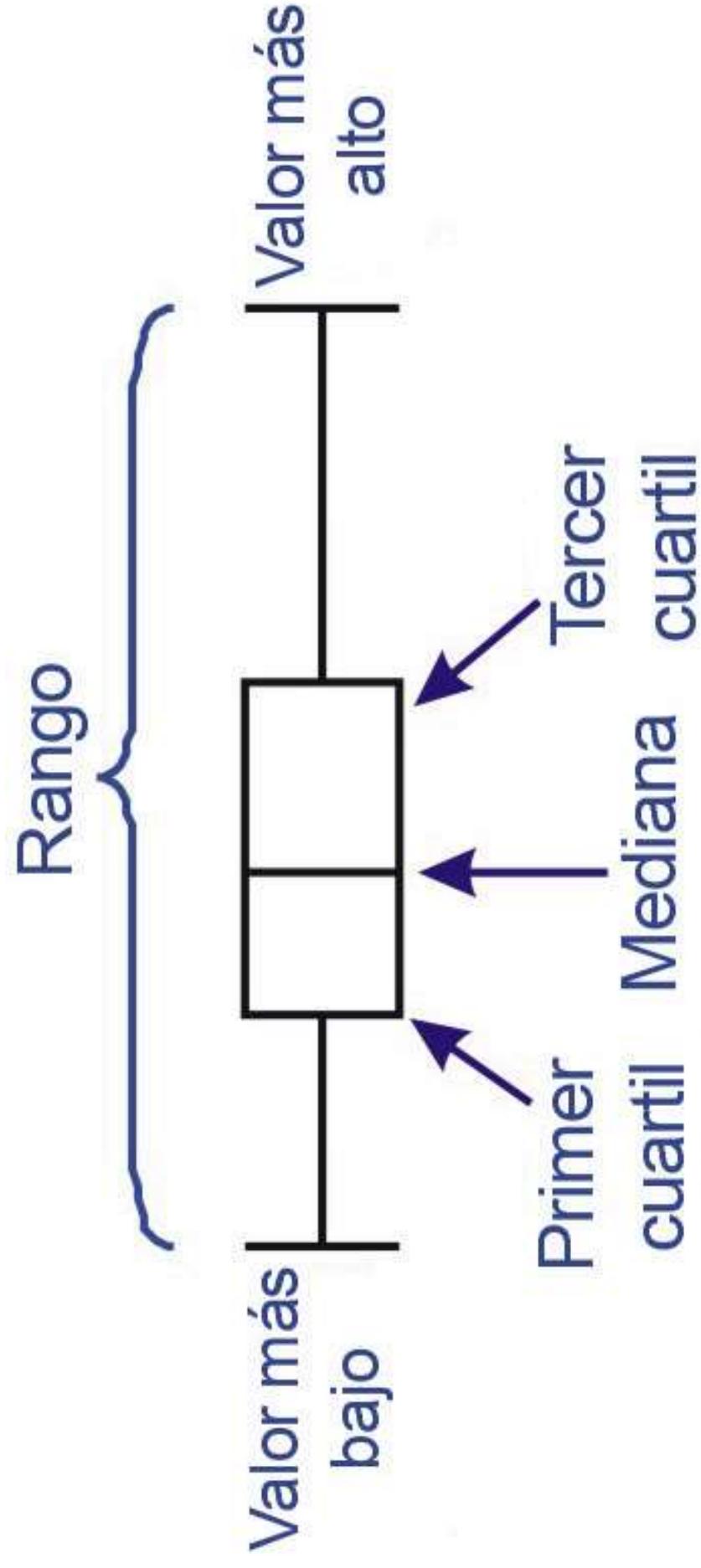


Diagrama de caja (caja y bigotes)

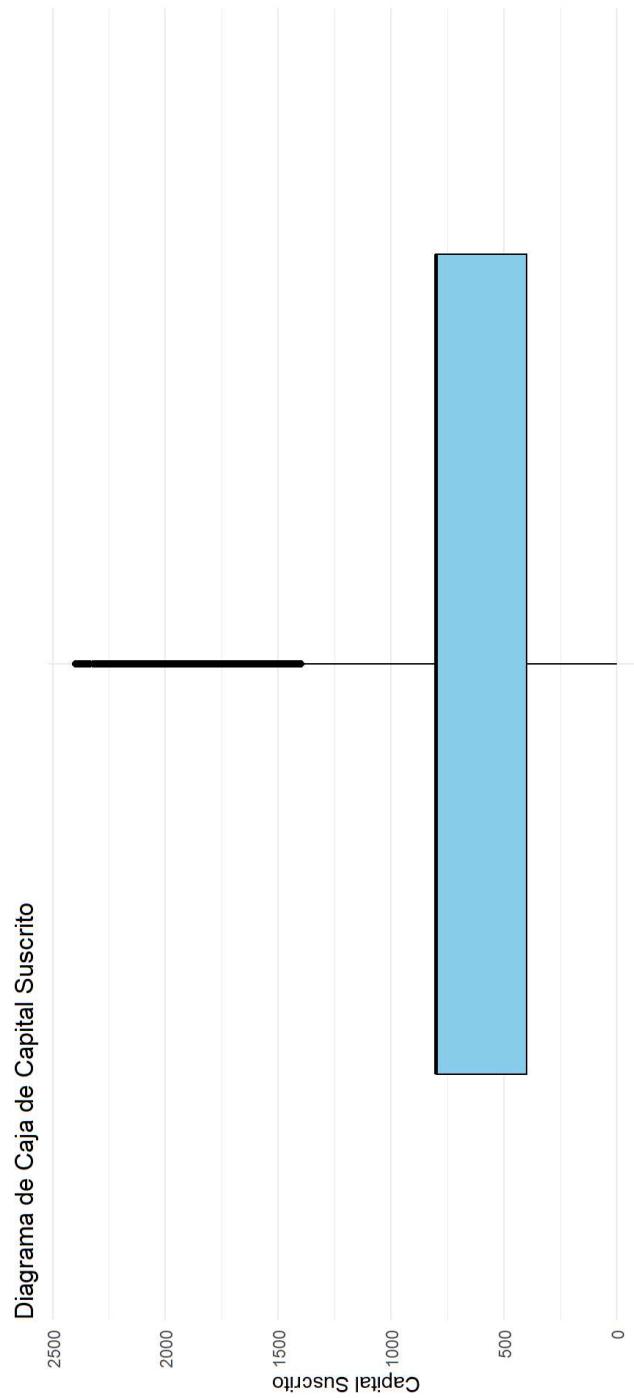
- El diagrama de caja y bigotes muestra la distribución de una variable cuantitativa
- La caja representa el rango intercuartil (IQR), y los bigotes se extienden hasta los valores extremos
- Se pueden identificar valores atípicos

Diagrama de caja genérico



Creando un diagrama de caja con ggplot2

- Utilizando la función `geom_boxplot()` de `ggplot2`



Creando un diagrama de caja con ggplot2

Geometrías para diferentes tipos de gráficos

- `geom_point()` : gráfico de dispersión
- `geom_line()` : gráfico de líneas
- `geom_bar()` : gráfico de barras
- `geom_boxplot()` : diagrama de caja y bigotes
- `geom_histogram()` : histograma
- `geom_density()` : densidad
- `geom_violin()` : violín
- `geom_text()` : texto
- `geom_label()` : etiquetas
- Se puede usar en combinación (ej. `geom_point() + geom_line()`)

Distribuciones acumuladas

Implementación en R

- Se puede generar una tabla de frecuencias con la función `cut()` y un workflow `dplyr` con `group_by()` y `summarise()`
- Se necesita decidir el número de intervalos o bins mediante `breaks`
- `cut` divide los datos en intervalos, y `table` cuenta la frecuencia de datos en cada intervalo

Para capital suscrito

```
1 tabla_frecuencias <- supercias_filtrado %>%
2   mutate(intervalo = cut(capital_suscripto, breaks = 5)) %>%
3   group_by(intervalo) %>% # Agrupar por intervalo
4   summarise(frecuencia = n()) %>% # Contar frecuencia
5   mutate(frecuencia_acumulada = cumsum(frecuencia), # Frecuencia ac
6   frecuencia_relativa = frecuencia / sum(frecuencia)) # Frec
7
8 tabla_frecuencias
9 ## # A tibble: 5 × 4
10 ##   intervalo
11 ##   <fct>
12 ##   1 (-2.36,480]
13 ##   2 (480,960]
14 ##   3 (960,1.44e+03]
15 ##   4 (1.44e+03,1.92e+03]
16 ##   5 (1.92e+03,2.4e+03]
```

Gráfico de la distribución acumulada

- Utilizando la función `geom_line()` de `ggplot2`, se puede crear la ojiva graficando la tabla calculada previamente

```
1 tabla_frecuencias %>%
2 ggplot(aes(x = intervalo, y = frecuencia_acumulada, group = 1)) +
  geom_line(color = "skyblue") +
  labs(title = "Ojiva de Capital Suscrito",
       x = "Intervalo",
       y = "Frecuencia Acumulada") +
  theme_minimal()
```

Análisis de datos categóricos

- Generalmente, nos interesa la frecuencia de ocurrencia de categorías
 - La frecuencia relativa o porcentaje de cada categoría es útil para comparar entre categorías
 - Utilizamos tablas de frecuencias y gráficos para visualizar la distribución de categorías
 - Calculamos frecuencias con workflows `dplyr` con `group_by()` y `summarise()`
- Para variables categóricas, se pueden utilizar diferentes tipos de gráficos
 - Algunos de los gráficos más comunes son:
 - Gráfico de barras

Gráfico de barras

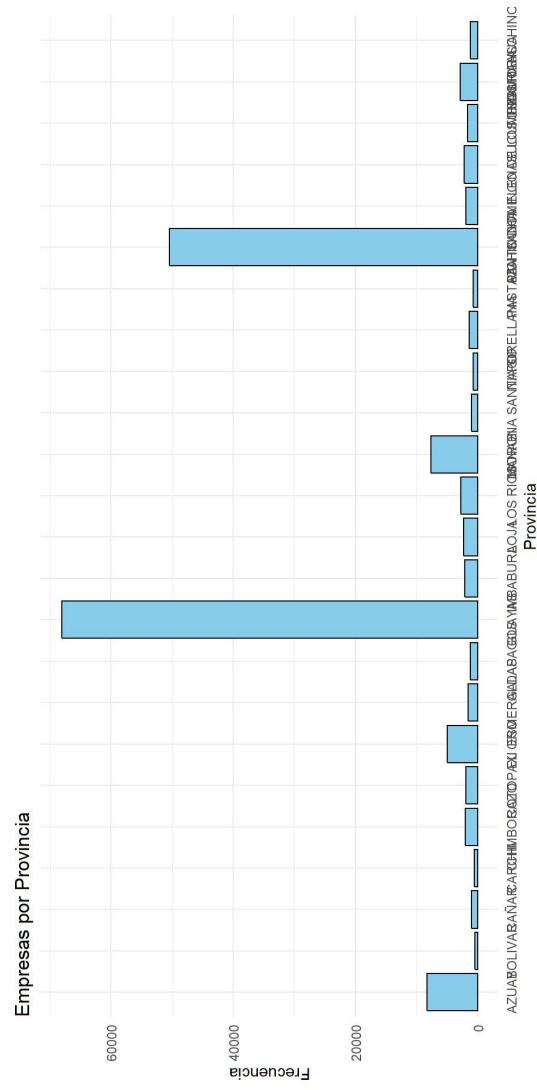
- El gráfico de barras es una forma sencilla de visualizar la frecuencia de una variable categórica
- En el eje x se muestran las categorías, y en el eje y la frecuencia
- En algunos casos, es válido apilar barras para mostrar la frecuencia de subcategorías
- Generalmente los gráficos de barras son verticales, pero también pueden ser horizontales
- Utiles con un número moderado de categorías

Creando un gráfico de barras con ggplot2

- Utilizando la función `geom_bar()` de `ggplot2`
- Se puede personalizar el color, el orden de las barras, las etiquetas, etc.
- La función `geom_bar()` cuenta automáticamente la frecuencia de cada categoría

Empresas por provincia

```
1 supercias_filtrado %>%
2   ggplot(aes(x = provincia)) +
3     geom_bar(fill = "skyblue", color = "black") +
4     labs(title = "Empresas por Provincia",
5       x = "Provincia",
6       y = "Frecuencia") +
7     theme_minimal()
```



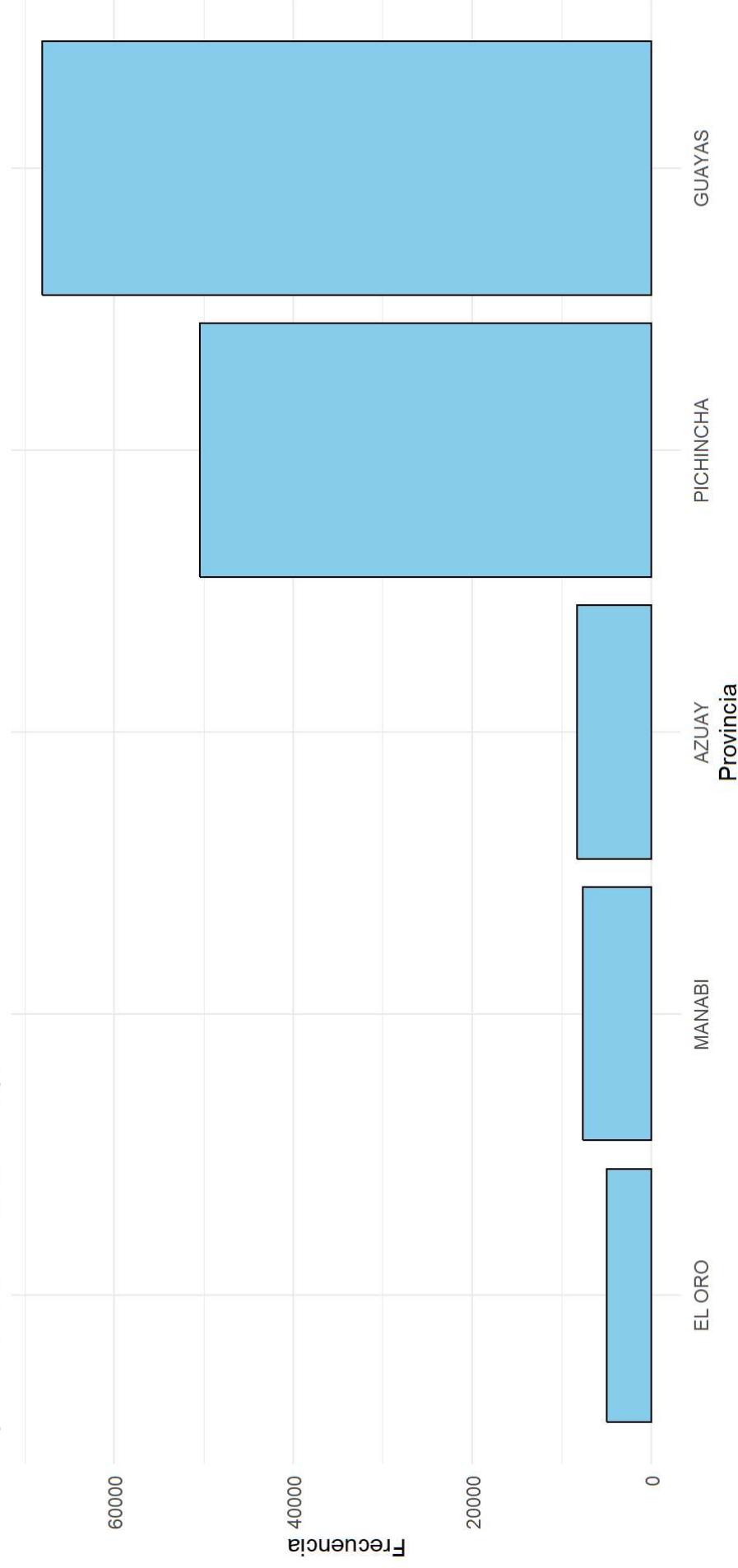
Empresas por provincia

- La función `top_n()` de `dplyr` permite seleccionar las `n` categorías más frecuentes
- Se puede utilizar para filtrar las 5 provincias con más empresas

```
1 library(forcats)
2
3 supercias_filtrado %>%
4 count(provincia) %>% # Count rápidamente cuenta la frecuencia de
5 top_n(5, n) %>%
6 ggplot(aes(x = fct_reorder(provincia, n), y = n)) +
7 geom_bar(stat = "identity", fill = "skyblue", color = "black") +
8 labs(title = "Top 5 Provincias con más Empresas",
9 x = "Provincia",
```

Empresas por provincia

Top 5 Provincias con más Empresas



geom_bar() y geom_col()

- `geom_bar()` es una abreviatura de `geom_bar(stat = "count")`, que cuenta la frecuencia de cada categoría
- `geom_col()` es una abreviatura de `geom_bar(stat = "identity")`, que utiliza los valores de la variable y para la altura de las barras
- Se puede utilizar `geom_col()` para gráficos de barras con valores precalculados, es decir, cuando ya se tiene la frecuencia de cada categoría
- Es más flexible que `geom_bar()`, ya que permite utilizar valores precalculados, no solo frecuencias

Columnas apiladas

- En algunos casos, es útil apilar las barras para mostrar la frecuencia de subcategorías
- Se puede utilizar el argumento `fill` en `aes()` para apilar las barras, utilizando una variable categórica adicional
- Se recomienda utilizar `geom_col()` y adecuadamente prepara los datos en formato largo (long format) - realizar un `pivot_longer()` con `tidyR` si es necesario

Empresas por provincia y sector (CIIU)

- Ordenamos las primeras 5 provincias con más empresas y los CIIU de nivel 1

```
1 supercias_filtrado %>%
2   group_by(provincia, ciu_nivel_1) %>%
3     summarise(frecuencia = n()) %>%
4       filter(provincia %in% c("EL ORO", "MANABI", "AZUAY", "PICHINCHA",
5         ggplot(aes(x = provincia, y = frecuencia, fill = ciu_nivel_1)) +
6           geom_col(position = "stack") +
7             labs(title = "Empresas por Provincia Y Sector (CIIU)",
8               x = "Provincia",
9               y = "Frecuencia",
10              fill = "Sector (CIIU)") +
11                theme_minimal()
```

Empresas por provincia y sector (CIIU)

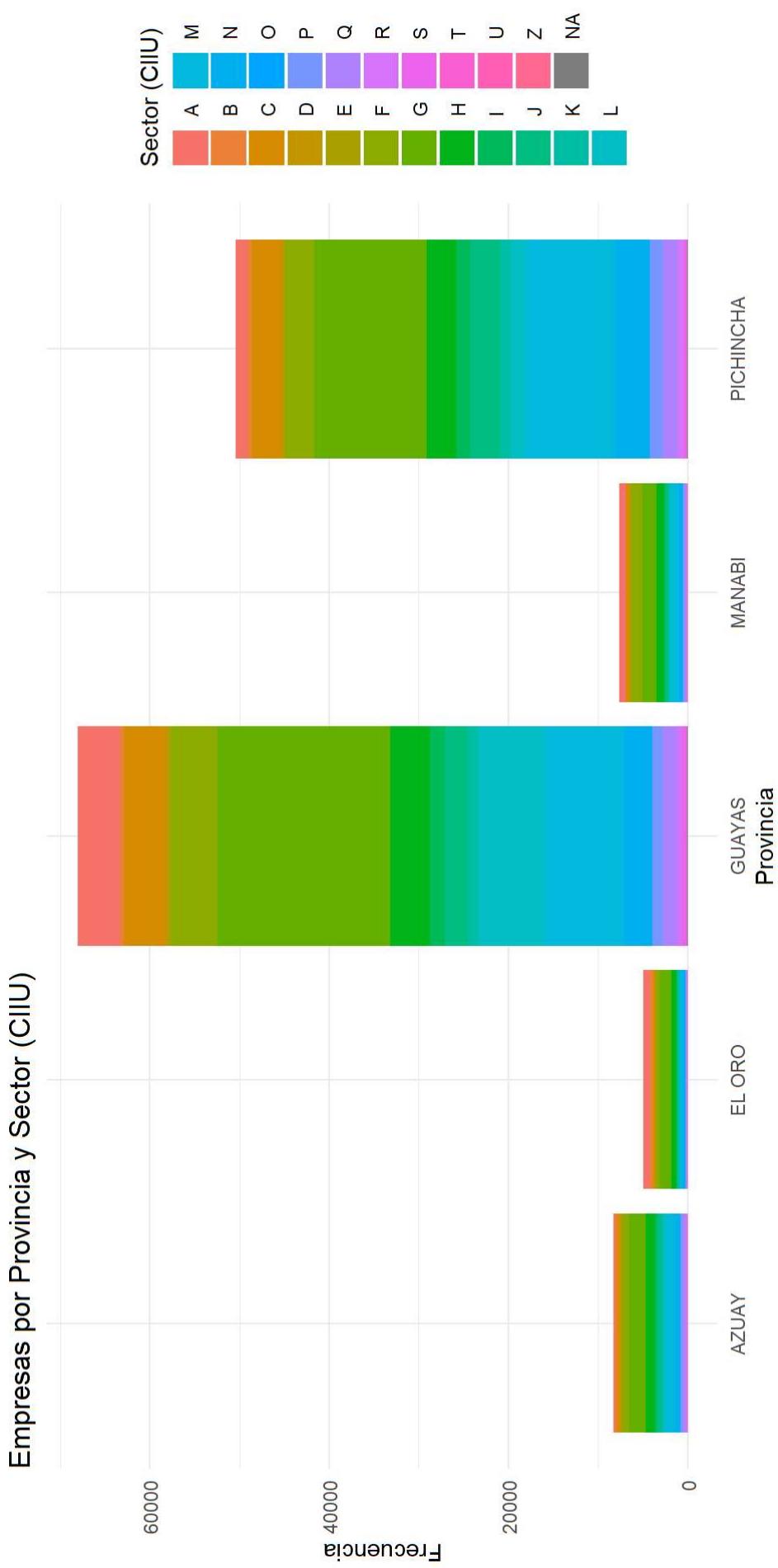


Gráfico de Pastel

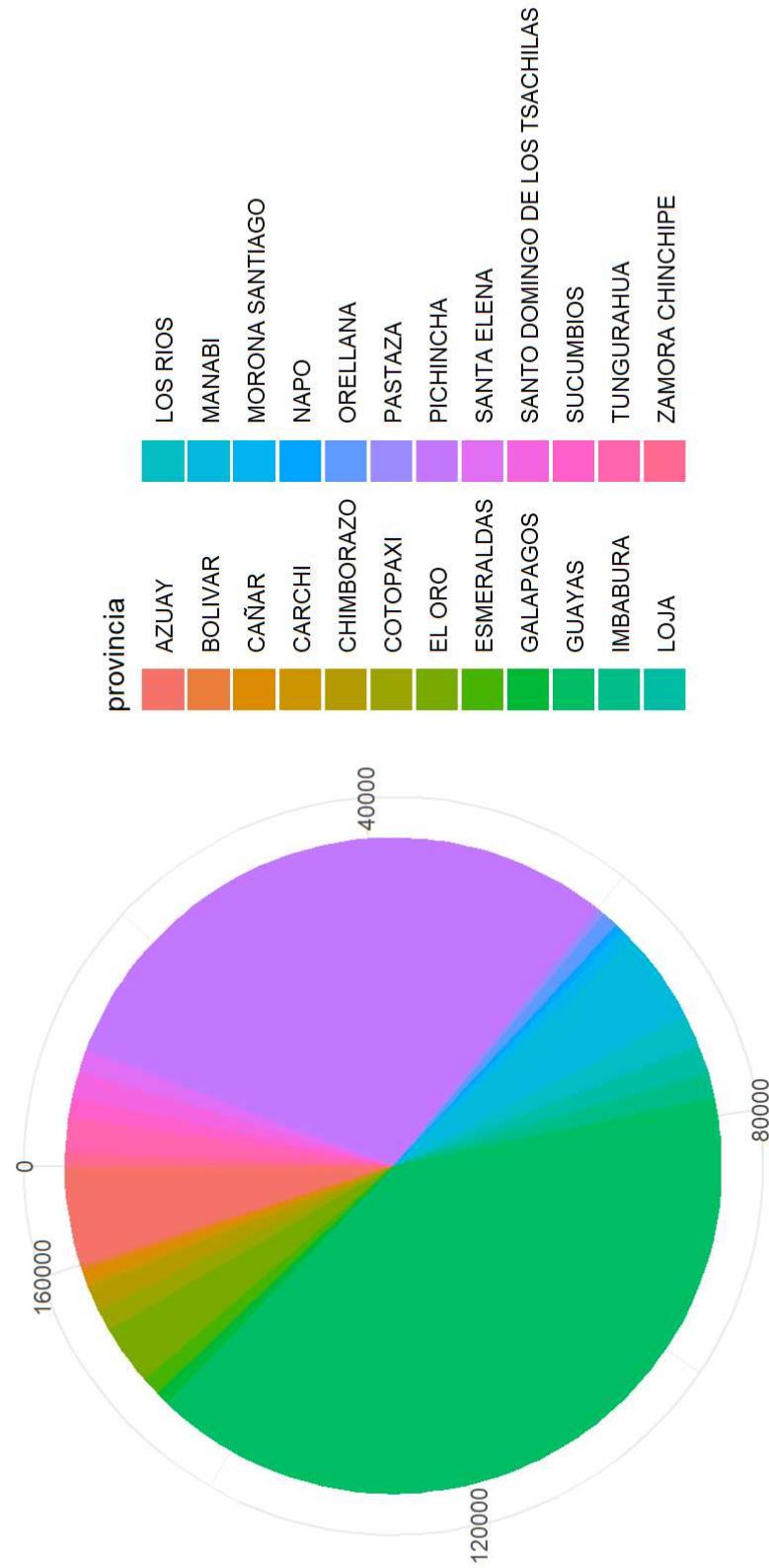
- El gráfico de pastel es una forma sencilla de visualizar la proporción de cada categoría en un conjunto de datos
- Se utiliza para mostrar la distribución de categorías en un conjunto de datos
- En R, se puede utilizar la función `geom_bar()` con el argumento `coord_polar()` para crear un gráfico de pastel
- Sin embargo, las buenas prácticas de visualización sugieren que los gráficos de barras son más efectivos para comparar categorías

Empresas por Provincia

```
1 supercias_filtrado %>%
2   count(provincia) %>%
3   ggplot(aes(x = " ", y = n, fill = provincia)) +
4     geom_bar(stat = "identity") +
5     coord_polar("y") +
6     labs(title = "Empresas por Provincia",
7           x = " ",
8           y = " ") +
9     theme_minimal()
```

Empresas por provincia

Empresas por Provincia



Facetas o paneles

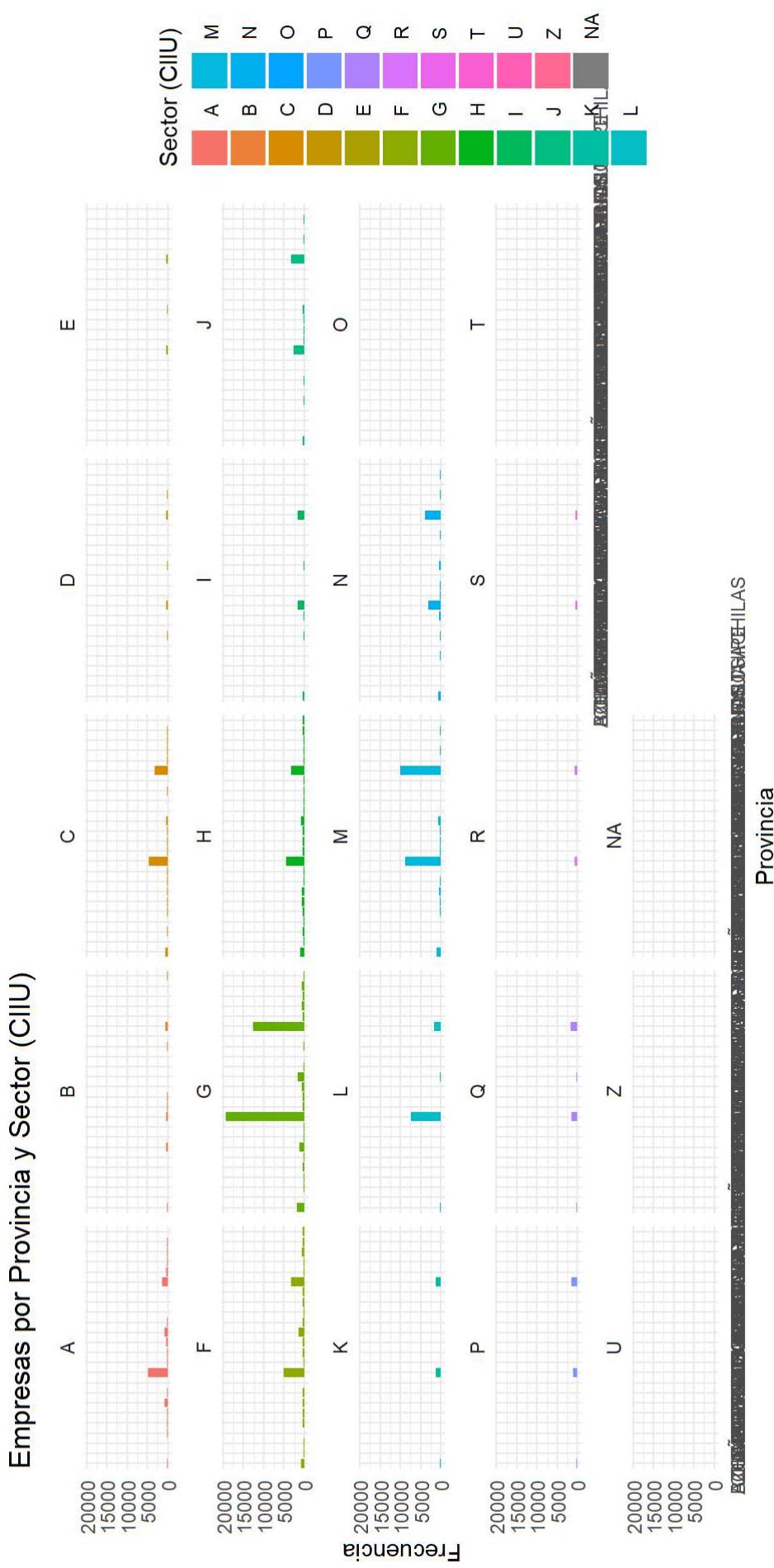
- Las facetas o paneles permiten dividir un gráfico en subgráficos según una variable categórica
- Se pueden utilizar para comparar la distribución de una variable en diferentes categorías
- En `ggplot2`, se puede utilizar la función `facet_wrap()` o `facet_grid()` para crear facetas
- Se especifica la variable categórica que se utilizará para dividir el gráfico en subgráficos

Empresas por provincia y sector (CIIU)

- Utilizamos facetas para mostrar la distribución de empresas por provincia y sector (CIIU) en subgráficos

```
1 supercias_filtrado %>%
2   group_by(provincia, ciu_nivel_1) %>%
3   summarise(frecuencia = n()) %>%
4   ggplot(aes(x = provincia, y = frecuencia, fill = ciu_nivel_1)) +
5   geom_col(position = "stack") +
6   labs(title = "Empresas por Provincia y Sector (CIIU)",
7        x = "Provincia",
8        y = "Frecuencia",
9        fill = "Sector (CIIU)") +
10  facet_wrap(~ciu_nivel_1) +
11  theme_minimal()
```

Empresas por provincia y sector (CIIU)



PROVINCIA

PROVINCIA

Editando gráficos con ggplot2:

theme()

- La función `theme()` permite personalizar la apariencia de un gráfico
- Se pueden modificar aspectos como el color de fondo, los ejes, las etiquetas, etc.

Estructura de theme()

- Se estructura a partir de cuatro tipos de elementos:
`element_text()`, `element_line()`,
`element_rect()`, `element_blank()`
- `element_text()` : textos en el gráfico
- `element_line()` : líneas en el gráfico
- `element_rect()` : rectángulos en el gráfico (por ejemplo, el fondo o los ejes)
- `element_blank()` : elementos en blanco, para eliminarlos

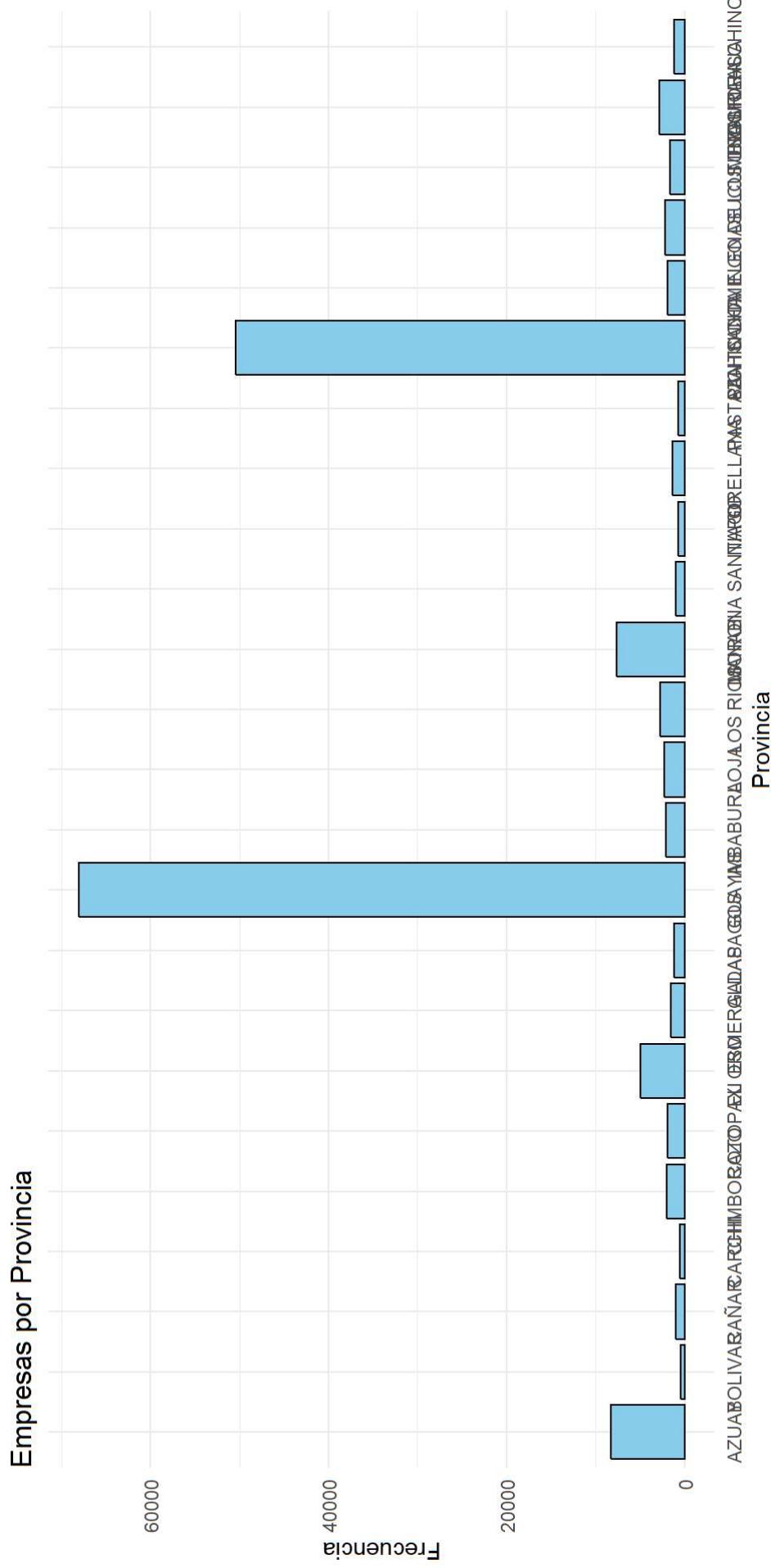
Ejemplo: modificar la leyenda de un gráfico

- Se puede modificar la apariencia de la leyenda con `theme()`
- Puesto que la leyenda es un elemento de texto, se utiliza `element_text()`
- Se puede modificar el tamaño, el color, la fuente, etc.

Modificar la leyenda de un gráfico

```
1 supercias_filtrado %>%
2   ggplot(aes(x = provincia)) +
3     geom_bar(fill = "skyblue", color = "black") +
4     labs(title = "Empresas por Provincia",
5       x = "Provincia",
6       y = "Frecuencia") +
7     theme_minimal() +
8     theme(legend.text = element_text(size = 10, color = "blue", face
```

Modificar la leyenda de un gráfico



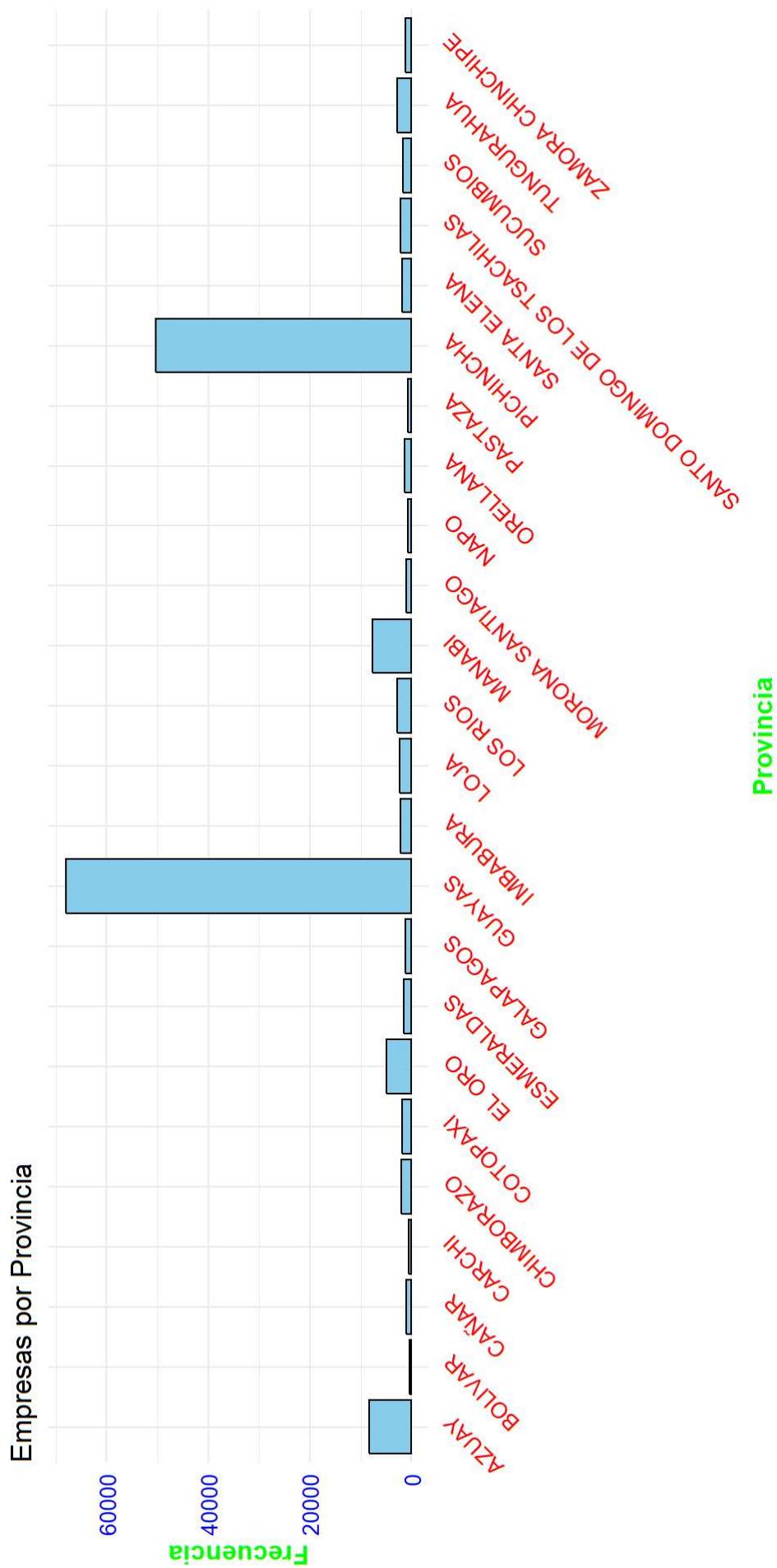
Modificar la apariencia del texto del eje

- Se puede modificar la apariencia del texto del eje con `theme()`
- Se utiliza `axis.text` para modificar el texto de los ejes, también `axis.title` para los títulos de los ejes
- Se puede modificar el tamaño, el color, la fuente, etc.

Modificar la apariencia del texto del eje

```
1 supercias_filtrado %>%
2   ggplot(aes(x = provincia)) +
3     geom_bar(fill = "skyblue", color = "black") +
4     labs(title = "Empresas por Provincia",
5       x = "Provincia",
6       y = "Frecuencia") +
7     theme_minimal() +
8     theme(axis.text.x = element_text(size = 10, color = "red", angle
9       axis.text.y = element_text(size = 10, color = "blue"),
10      axis.title = element_text(size = 12, color = "green", face
```

Modificar la apariencia del texto del eje



Provincia