



ANÁLISIS DESCRIPTIVO R

Madin Rivera, Alberto.

may. 07, 2023

Índice general

1. ANÁLISIS DE DATOS	1
1. ¿QUÉ ES UN ANÁLISIS DESCRIPTIVO?	2
2. ¿CÓMO HACER UN ANÁLISIS DESCRIPTIVO EN R?	3
3. ANÁLISIS GRÁFICO CON LA LIBRERÍA <code>ggplot2</code>	8
4. ANÁLISIS EXPLORATORIO MÁS PROFUNDO	12
5. MODELO DE REGRESIÓN LINEAL MÚLTIPLE EN R PARA EL ANÁLISIS EXPLORATORIO DE DATOS	14
6. MANIPULACIÓN DE DATOS CON <code>dplyr</code>	16
7. ANÁLISIS AVANZADO: DIAGRAMA DE COORDENADAS	18
8. RELACIÓN Y CRUCE DE TABLAS ENTRE DATOS SIMILARES	20
2. CONCLUSIONES Y BIBLIOGRAFÍA	21
1. CONCLUSIONES SOBRE EL ANÁLISIS DE DATOS	21
2. BIBLIOGRAFÍA Y SITIOS WEB DONDE ENCONTRAR CONOCIMIENTO PARA ANÁLISIS EXPLORATORIO DE DATOS	22

1. ANÁLISIS DE DATOS

1. ¿QUÉ ES UN ANÁLISIS DESCRIPTIVO?

Un análisis descriptivo es una técnica estadística utilizada para resumir y describir características importantes de un conjunto de datos. El objetivo principal del análisis descriptivo es proporcionar una comprensión básica de los datos a través de medidas estadísticas y herramientas visuales, sin necesidad de realizar inferencias o predicciones.

Algunas de las medidas estadísticas utilizadas en el análisis descriptivo incluyen la media, la mediana, la moda, la desviación estándar, el rango, el coeficiente de variación, entre otros. Estas medidas proporcionan información sobre la tendencia central, la dispersión y la forma de la distribución de los datos.

Además de las medidas estadísticas, el análisis descriptivo también puede incluir herramientas visuales, como histogramas, diagramas de caja, diagramas de dispersión, gráficos de barras, entre otros. Estas herramientas visuales permiten visualizar patrones y relaciones importantes en los datos.

En general, el análisis descriptivo es una técnica importante en la exploración inicial de un conjunto de datos, ya que proporciona una descripción detallada y completa de las características y patrones presentes en los datos, lo que ayuda a formular hipótesis y a desarrollar modelos más complejos en análisis posteriores.

2. ¿CÓMO HACER UN ANÁLISIS DESCRIPTIVO EN R?

Para hacer un análisis descriptivo en R, podemos utilizar diversas funciones y librerías que proporcionan medidas estadísticas y herramientas visuales. A continuación, se presenta un ejemplo de cómo realizar un análisis descriptivo básico en R utilizando la librería `mtcars` incluida en R:

1. Cargar la librería `mtcars`:

```
data(mtcars)
```

2. Ver las primeras filas de la librerías:

```
head(mtcars)
```

```
##           mpg  cyl  disp  hp  drat    wt  qsec vs  am  gear  carb
## Mazda RX4      21.0   6  160 110  3.90  2.620 16.46  0   1    4    4
## Mazda RX4 Wag  21.0   6  160 110  3.90  2.875 17.02  0   1    4    4
## Datsun 710     22.8   4  108  93  3.85  2.320 18.61  1   1    4    1
## Hornet 4 Drive  21.4   6  258 110  3.08  3.215 19.44  1   0    3    1
## Hornet Sportabout 18.7   8  360 175  3.15  3.440 17.02  0   0    3    2
## Valiant        18.1   6  225 105  2.76  3.460 20.22  1   0    3    1
```

3. Obtener un resumen estadístico de la librería:

```
summary(mtcars)
```

```
##           mpg           cyl           disp           hp
##  Min.      :10.40  Min.      :4.000  Min.      : 71.1  Min.      : 52.0
## 1st Qu.:15.43  1st Qu.:4.000  1st Qu.:120.8  1st Qu.: 96.5
##  Median :19.20  Median :6.000  Median :196.3  Median :123.0
##   Mean  :20.09  Mean   :6.188  Mean   :230.7  Mean   :146.7
## 3rd Qu.:22.80  3rd Qu.:8.000  3rd Qu.:326.0  3rd Qu.:180.0
##   Max.  :33.90  Max.   :8.000  Max.   :472.0  Max.   :335.0
##           drat           wt           qsec           vs
##  Min.      :2.760  Min.      :1.513  Min.      :14.50  Min.      :0.0000
## 1st Qu.:3.080  1st Qu.:2.581  1st Qu.:16.89  1st Qu.:0.0000
##  Median :3.695  Median :3.325  Median :17.71  Median :0.0000
##   Mean  :3.597  Mean   :3.217  Mean   :17.85  Mean   :0.4375
## 3rd Qu.:3.920  3rd Qu.:3.610  3rd Qu.:18.90  3rd Qu.:1.0000
```

```
## Max. :4.930 Max. :5.424 Max. :22.90 Max. :1.0000
##      am      gear      carb
## Min. :0.0000 Min. :3.000 Min. :1.000
## 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:2.000
## Median :0.0000 Median :4.000 Median :2.000
## Mean :0.4062 Mean :3.688 Mean :2.812
## 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:4.000
## Max. :1.0000 Max. :5.000 Max. :8.000
```

4. Calcular la media y la desviación estándar de la variable de peso (wt):

```
mean(mtcars$wt)
```

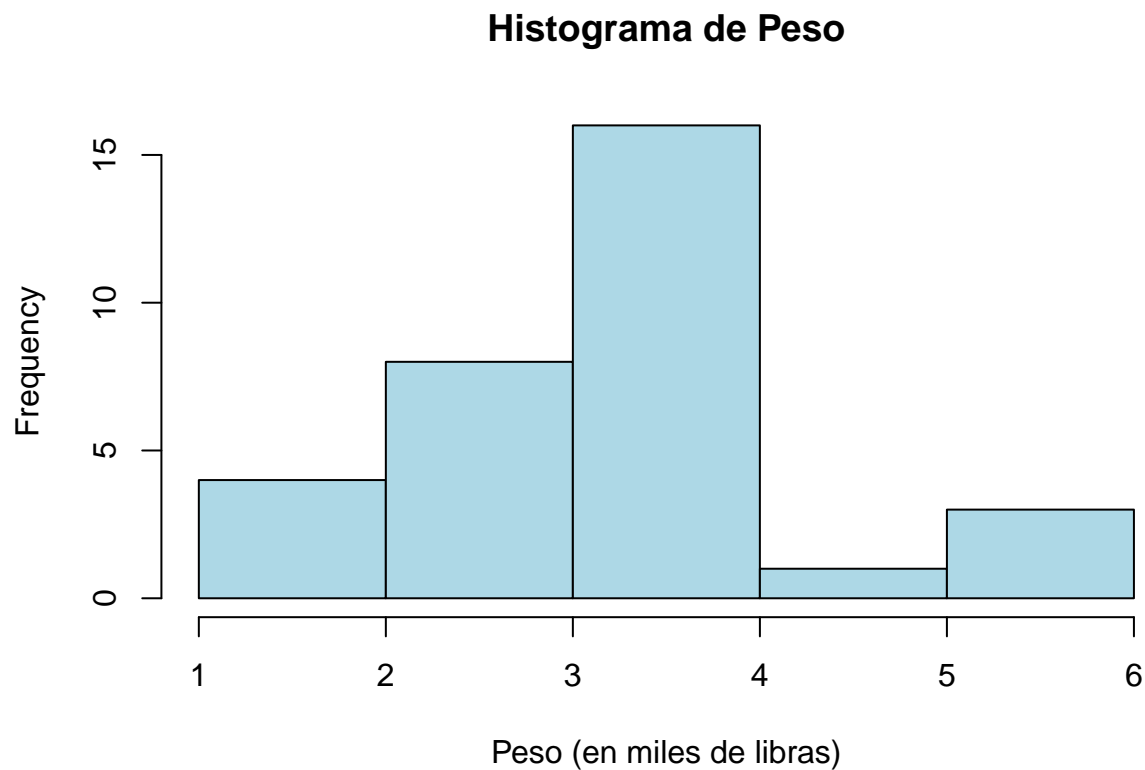
```
## [1] 3.21725
```

```
sd(mtcars$wt)
```

```
## [1] 0.9784574
```

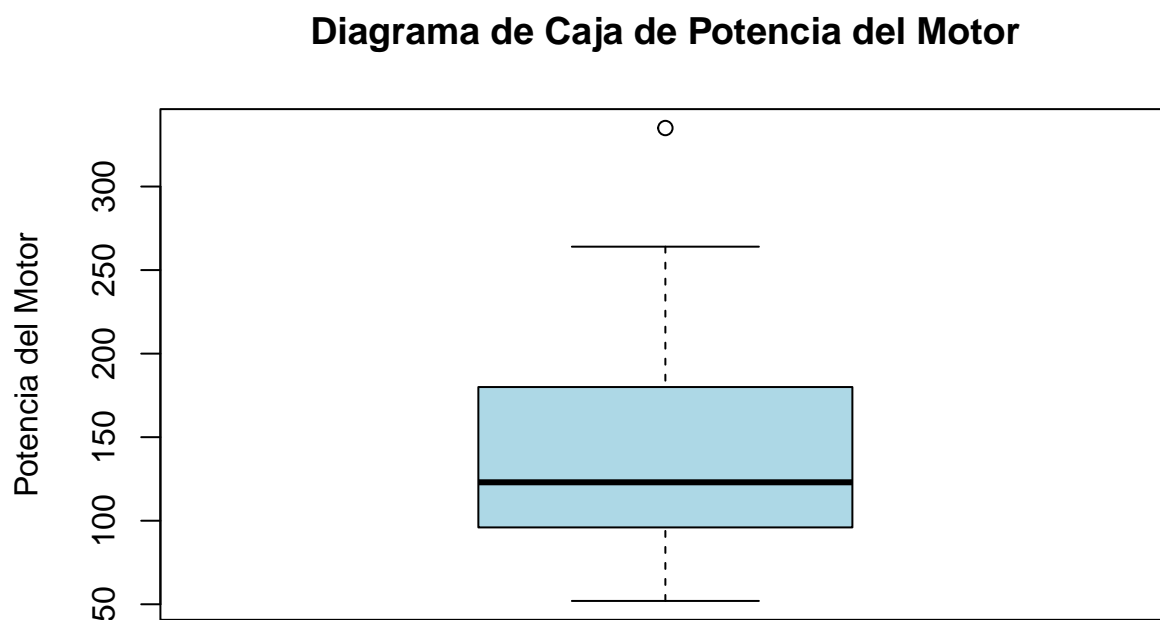
5. Crear un histograma de la variable de peso (wt) utilizando la función `hist()`:

```
hist(mtcars$wt, breaks = 5, col = "lightblue", border = "black",
     main = "Histograma de Peso", xlab = "Peso (en miles de libras)")
```



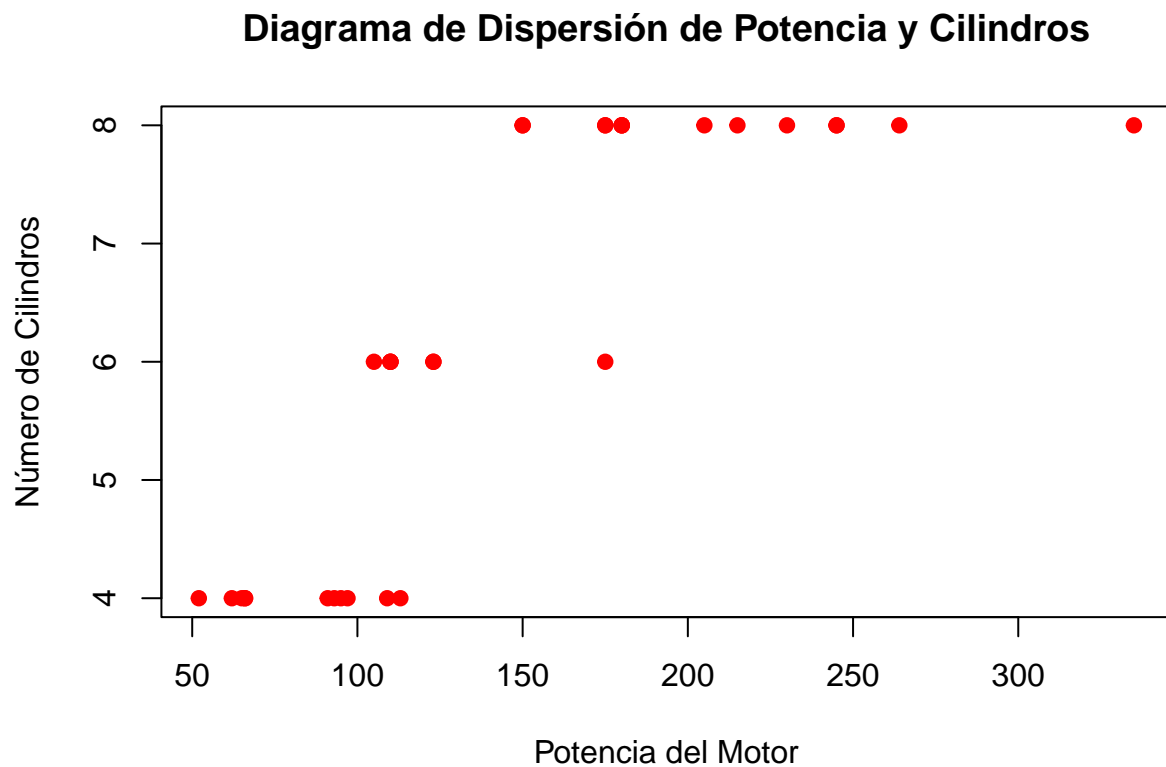
6. Crear un diagrama de caja de la variable de potencia del motor (hp) utilizando la función `boxplot()`:

```
boxplot(mtcars$hp, col = "lightblue", border = "black",  
        main = "Diagrama de Caja de Potencia del Motor",  
        ylab = "Potencia del Motor")
```



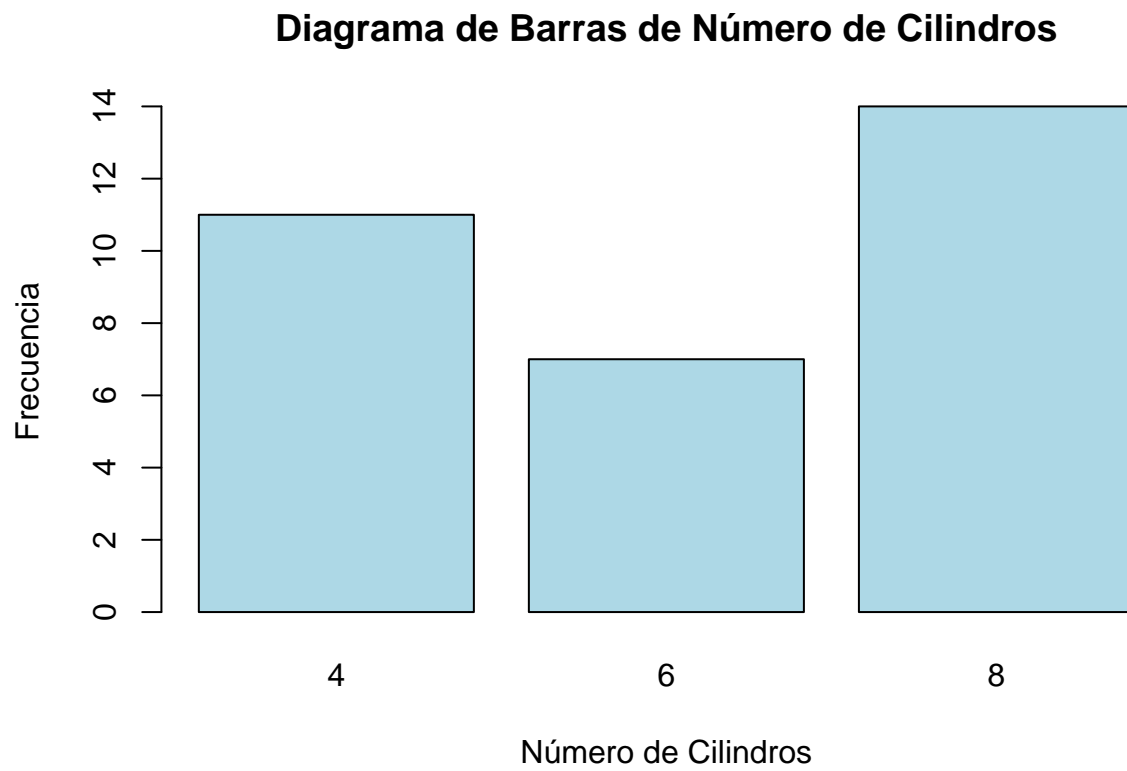
7. Crear un diagrama de dispersión de la potencia del motor (hp) en función del número de cilindros (cyl) utilizando la función `plot()`:

```
plot(mtcars$hp, mtcars$cyl, col = "red", pch = 19,  
     main = "Diagrama de Dispersión de Potencia y Cilindros",  
     xlab = "Potencia del Motor", ylab = "Número de Cilindros")
```



8. Crear un diagrama de barras de la variable de número de cilindros (cyl) utilizando la función `barplot()`:

```
barplot(table(mtcars$cyl), col = "lightblue", border = "black",  
        main = "Diagrama de Barras de Número de Cilindros",  
        xlab = "Número de Cilindros", ylab = "Frecuencia")
```

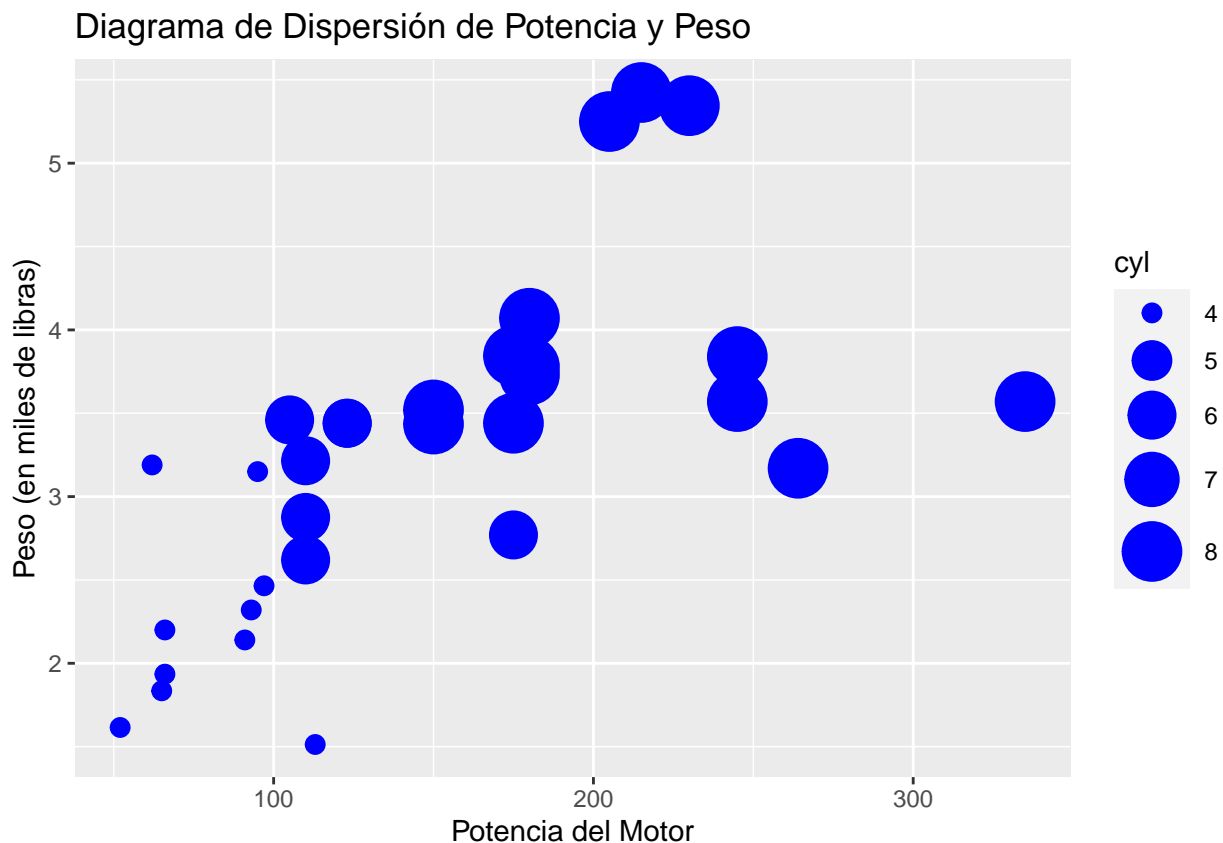


3. ANÁLISIS GRÁFICO CON LA LIBRERÍA `ggplot2`

1. Crear un diagrama de dispersión de la potencia del motor (hp) en función del peso (wt), donde el tamaño de los puntos está determinado por el número de cilindros (cyl):

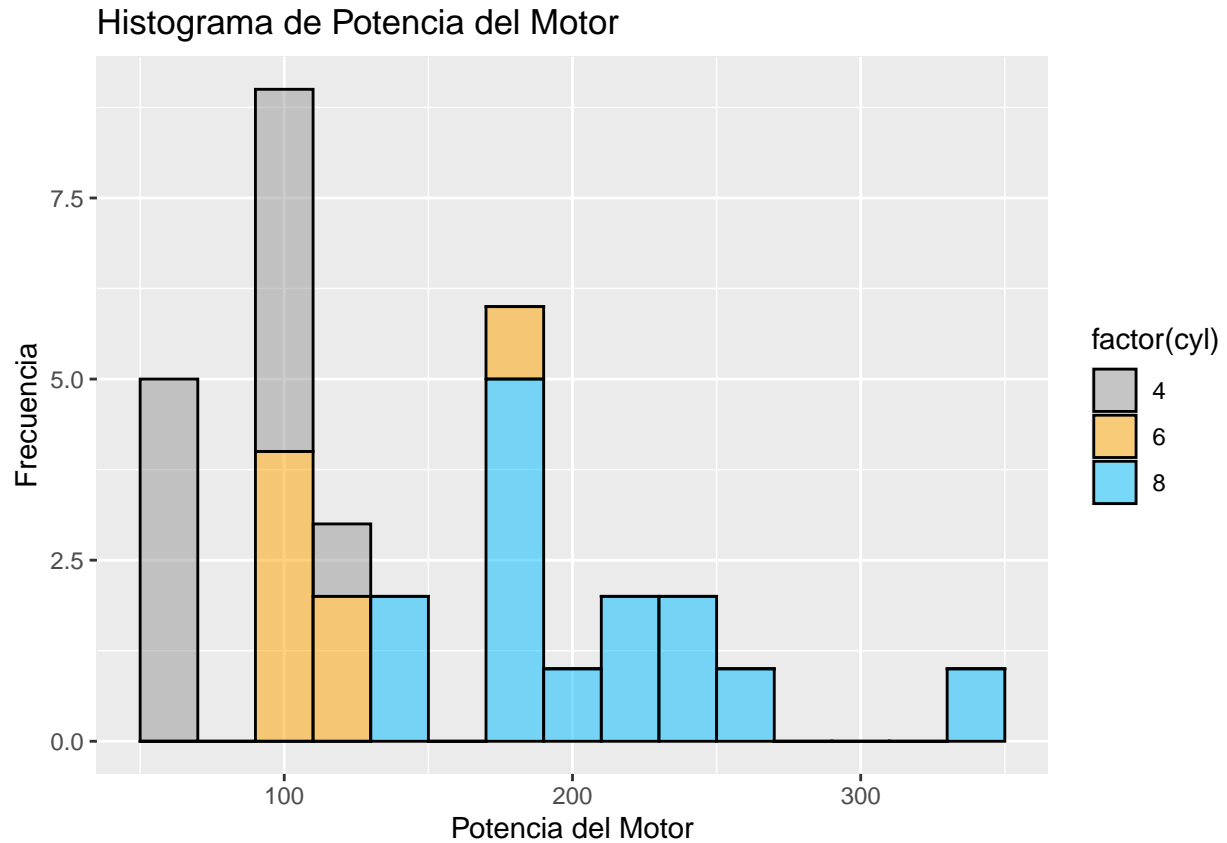
```
library(ggplot2)

ggplot(mtcars, aes(x = hp, y = wt, size = cyl)) +
  geom_point(color = "blue") +
  scale_size_continuous(range = c(3, 10)) +
  labs(title = "Diagrama de Dispersión de Potencia y Peso",
       x = "Potencia del Motor", y = "Peso (en miles de libras)")
```



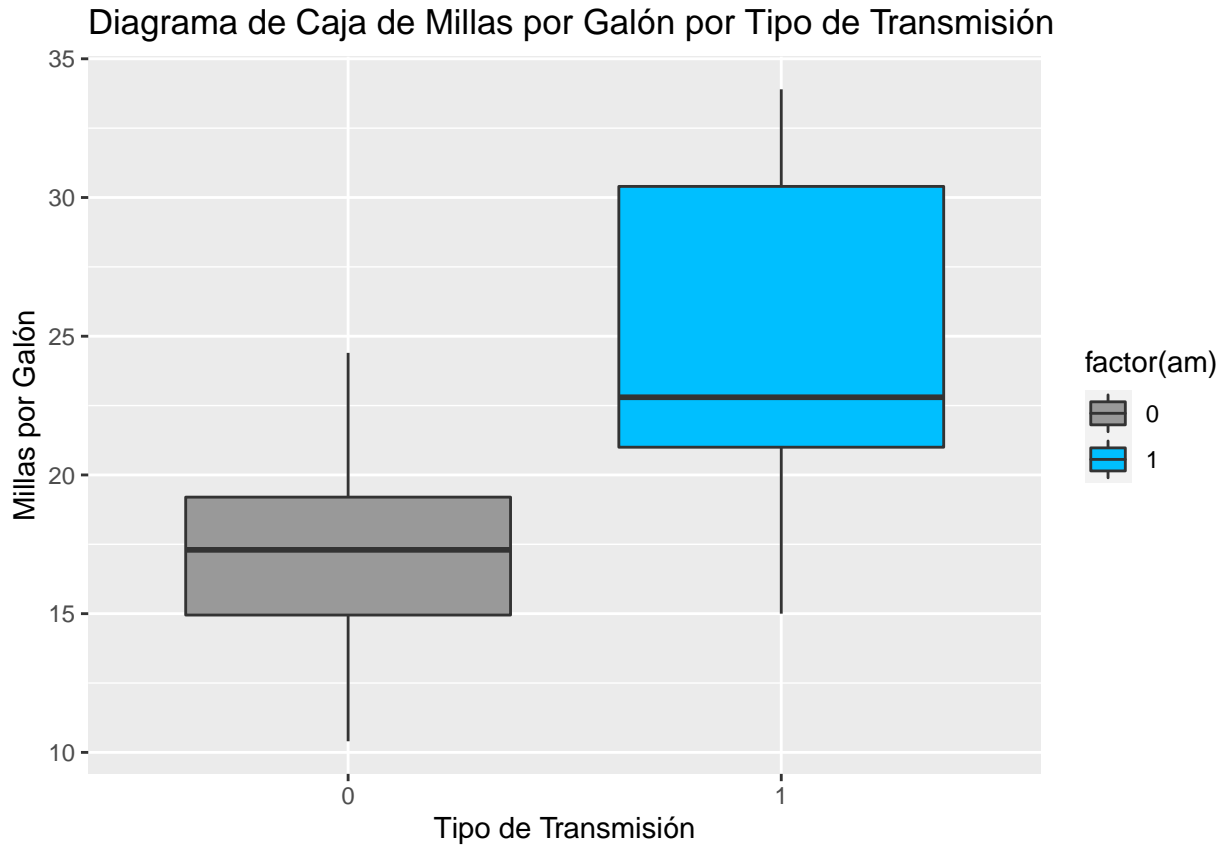
2. Crear un histograma de la variable de potencia del motor (hp) con colores personalizados:

```
ggplot(mtcars, aes(x = hp, fill = factor(cyl))) +
  geom_histogram(binwidth = 20, color = "black", alpha = 0.5) +
  scale_fill_manual(values = c("#999999", "#FFA500", "#00BFFF")) +
  labs(title = "Histograma de Potencia del Motor",
       x = "Potencia del Motor", y = "Frecuencia")
```



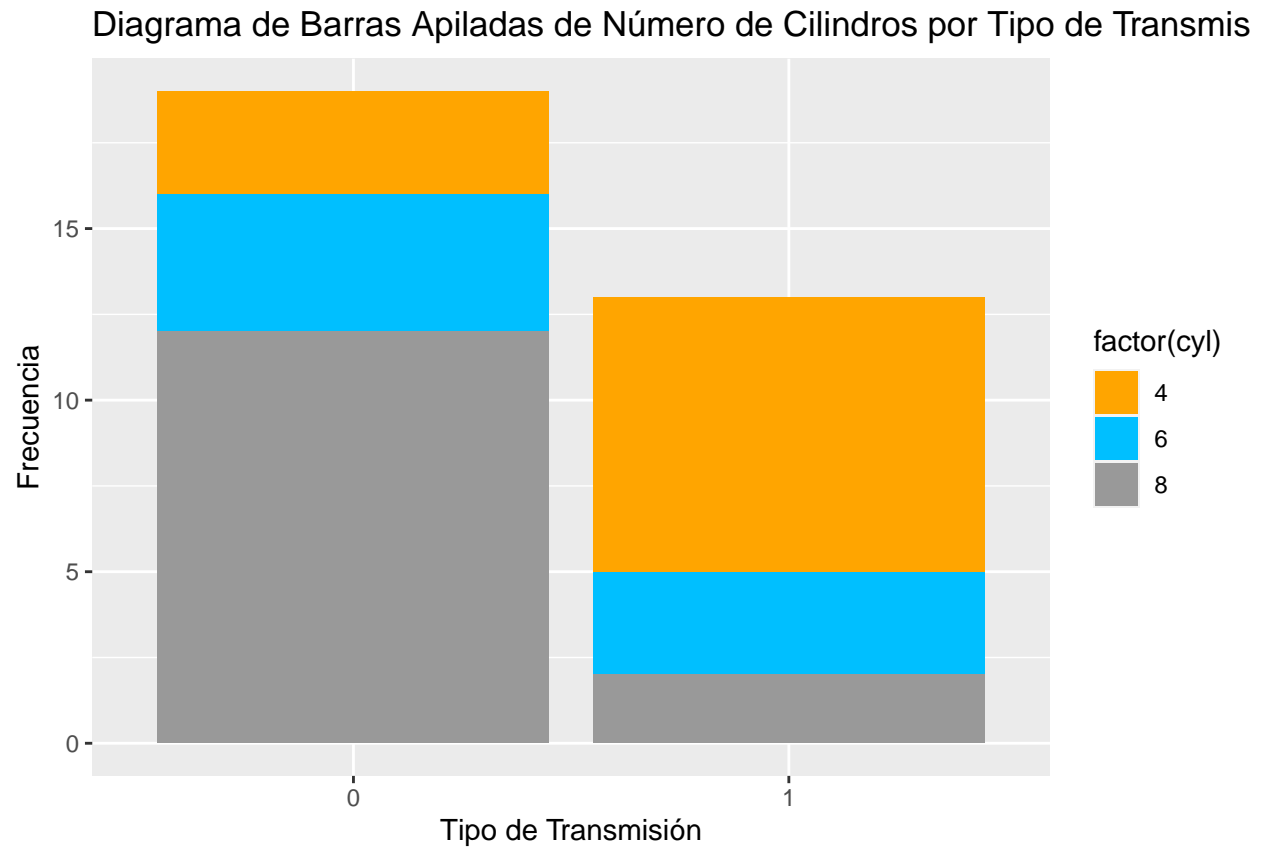
3. Crear un diagrama de caja y bigotes de la variable de millas por galón (mpg) por tipo de transmisión (am):

```
ggplot(mtcars, aes(x = factor(am), y = mpg, fill = factor(am))) +
  geom_boxplot() +
  scale_fill_manual(values = c("#999999", "#00BFFF")) +
  labs(title = "Diagrama de Caja de Millas por Galón por Tipo de Transmisión",
       x = "Tipo de Transmisión", y = "Millas por Galón")
```



Crear un diagrama de barras apiladas del número de cilindros (cyl) por tipo de transmisión (am):

```
ggplot(mtcars, aes(x = factor(am), fill = factor(cyl))) +
  geom_bar(position = "stack") +
  scale_fill_manual(values = c("#FFA500", "#00BFFF", "#999999")) +
  labs(title = "Diagrama de Barras Apiladas de Número de Cilindros por Tipo de Transmisión",
       x = "Tipo de Transmisión", y = "Frecuencia")
```



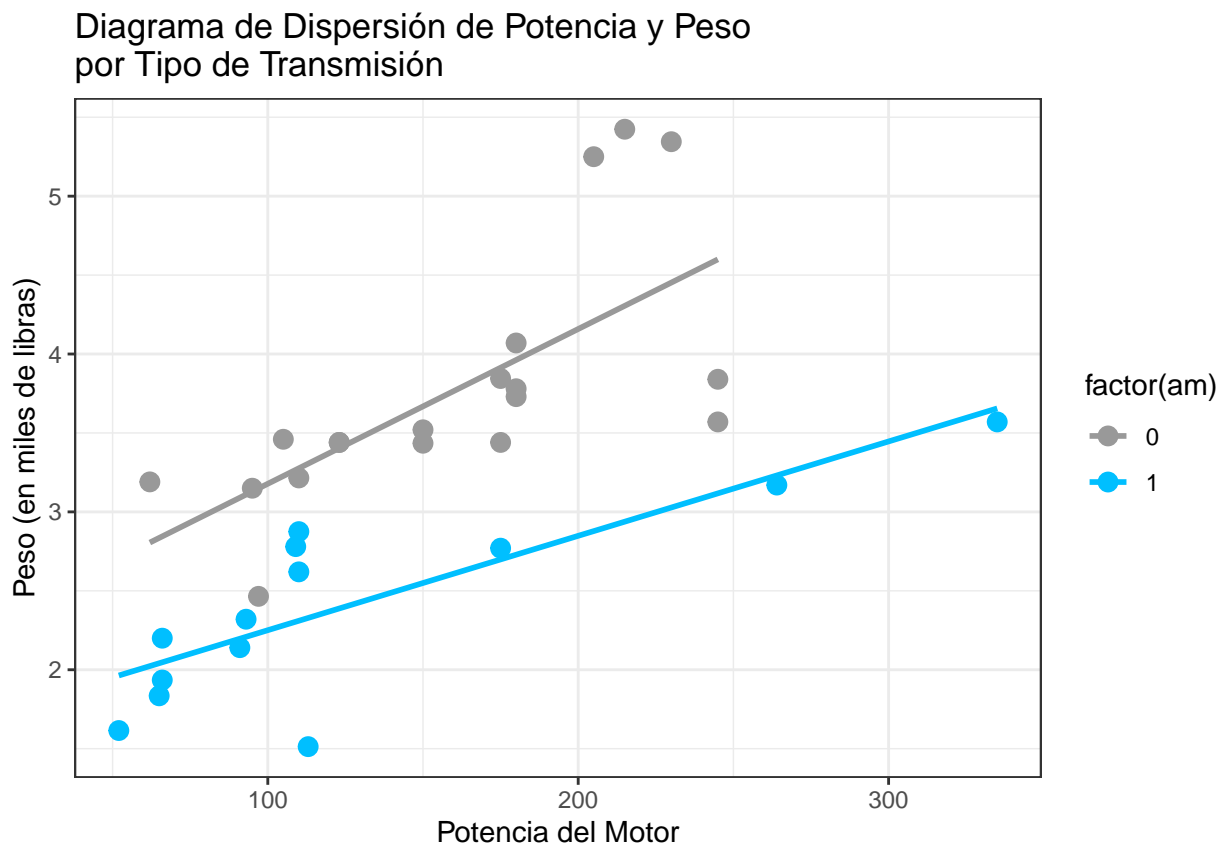
`ggplot2` es una herramienta muy útil para realizar análisis exploratorio de datos. A través de `ggplot2`, se pueden crear gráficos de alta calidad y personalizados que permiten visualizar y comprender mejor los patrones y relaciones en los datos.

4. ANÁLISIS EXPLORATORIO MÁS PROFUNDO

1. Crear un diagrama de dispersión de la potencia del motor (hp) en función del peso (wt), donde el color de los puntos está determinado por el tipo de transmisión (am), y se incluye una línea de regresión para cada tipo de transmisión:

```
library(ggplot2)

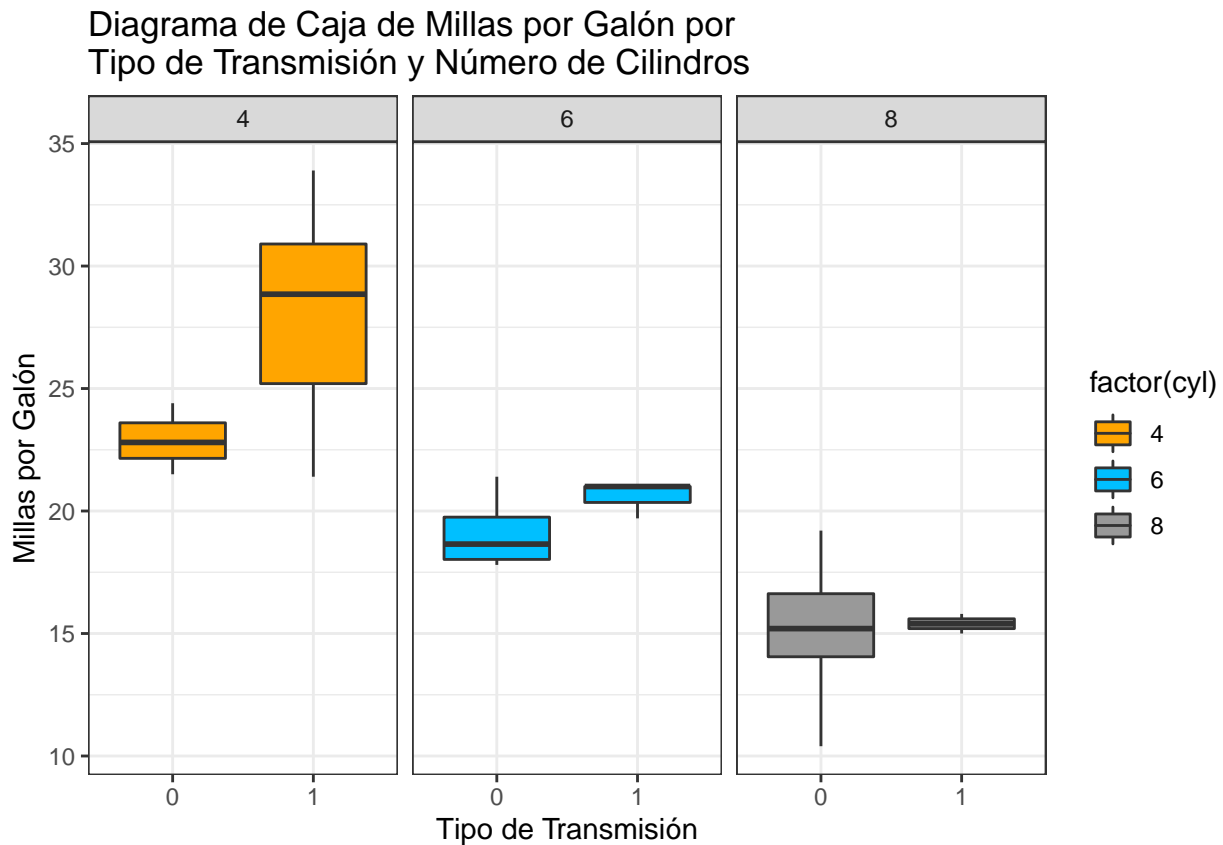
ggplot(mtcars, aes(x = hp, y = wt, color = factor(am))) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  labs(title = "Diagrama de Dispersión de Potencia y Peso
por Tipo de Transmisión",
       x = "Potencia del Motor", y = "Peso (en miles de libras)") +
  scale_color_manual(values = c("#999999", "#00BFFF")) +
  theme_bw()
```



Este gráfico muestra la relación entre la potencia del motor y el peso de los vehículos en función del tipo de transmisión. Se puede ver que, en general, los vehículos con transmisión manual tienen mayor potencia y menor peso que los vehículos con transmisión automática. Además, las líneas de regresión indican que existe una relación positiva entre la potencia del motor y el peso en ambos tipos de transmisión.

2. Crear un diagrama de caja y bigotes de la variable de millas por galón (mpg) por tipo de transmisión (am) y por número de cilindros (cyl), donde el color de los boxes está determinado por el número de cilindros:

```
ggplot(mtcars, aes(x = factor(am), y = mpg, fill = factor(cyl))) +
  geom_boxplot() +
  scale_fill_manual(values = c("#FFA500", "#00BFFF", "#999999")) +
  labs(title = "Diagrama de Caja de Millas por Galón por
Tipo de Transmisión y Número de Cilindros",
       x = "Tipo de Transmisión", y = "Millas por Galón") +
  facet_wrap(~cyl, ncol = 3) +
  theme_bw()
```



Este gráfico muestra cómo las millas por galón varían según el tipo de transmisión y el número de cilindros. En general, los vehículos con transmisión manual tienen un mayor número de millas por galón que los vehículos con transmisión automática. Además, se puede ver que a medida que aumenta el número de cilindros, disminuye el número de millas por galón en ambos tipos de transmisión.

5. MODELO DE REGRESIÓN LINEAL MÚLTIPLE EN R PARA EL ANÁLISIS EXPLORATORIO DE DATOS

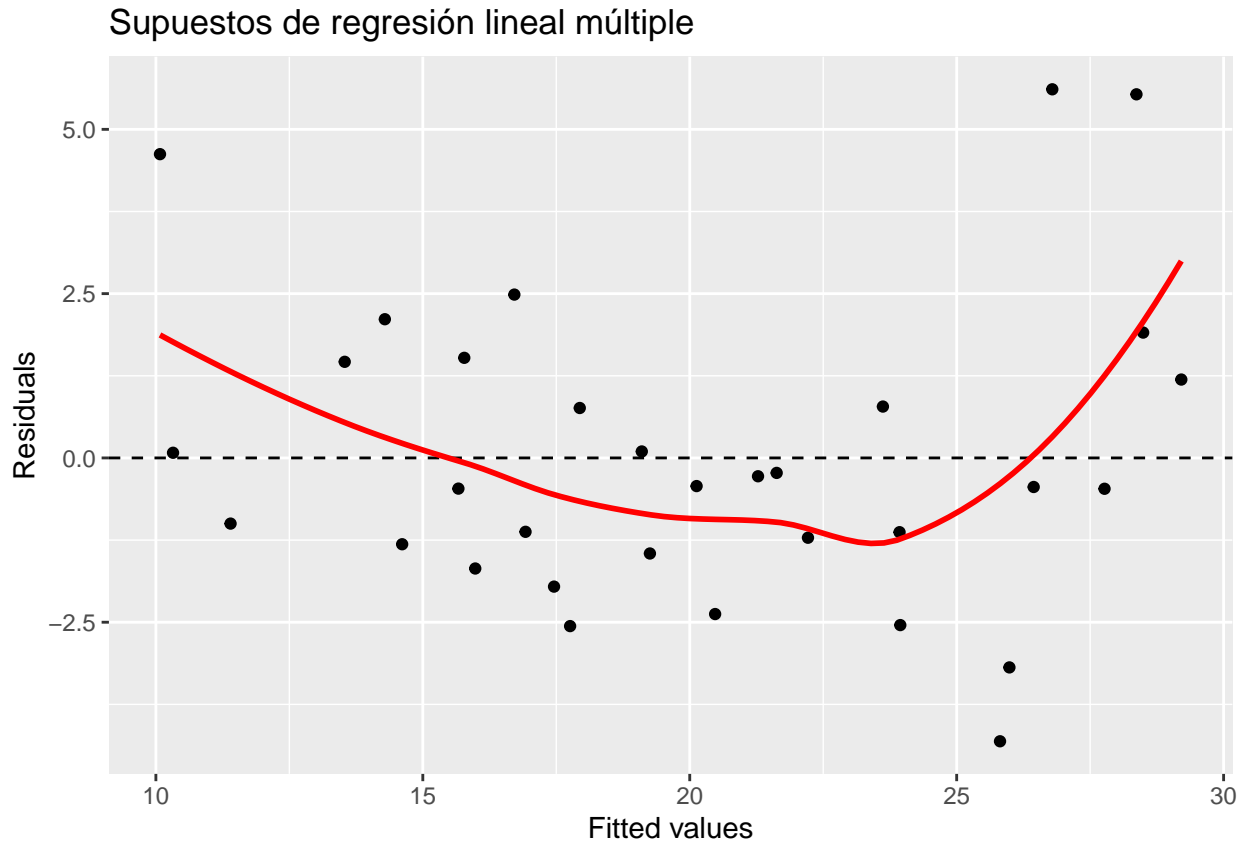
Para crear un modelo de regresión lineal múltiple con los datos de mtcars en R, podemos utilizar la función `lm()`. A continuación se muestra un ejemplo de cómo crear el modelo y graficar los supuestos con `ggplot2`:

```
library(ggplot2)

# Crear el modelo de regresión lineal múltiple
model <- lm(mpg ~ cyl + disp + hp + wt + qsec, data = mtcars)

# Graficar los supuestos del modelo
ggplot(data = model, aes(x = .fitted, y = .resid)) +
  geom_point() +
  labs(x = "Fitted values", y = "Residuals") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(se = FALSE, color = "red") +
  ggtitle("Supuestos de regresión lineal múltiple")
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



En este ejemplo, creamos un modelo de regresión lineal múltiple utilizando la función `lm()`, donde `mpg` es la variable dependiente y `cyl`, `disp`, `hp`, `wt`, y `qsec` son las variables independientes. Luego, utilizamos `ggplot2` para graficar los supuestos del modelo, incluyendo un gráfico de dispersión de los valores ajustados vs. los residuos, una línea horizontal en el valor cero para verificar la homocedasticidad y una curva de densidad para verificar la normalidad.

Para verificar si se cumplen los supuestos del modelo de regresión lineal, podemos analizar la gráfica de los supuestos del modelo. En esta gráfica, podemos verificar si se cumple la homocedasticidad observando si los puntos están distribuidos aleatoriamente alrededor de la línea horizontal en el valor cero. Podemos verificar la normalidad de los residuos observando si la curva de densidad se ajusta a una distribución normal. Si no se cumplen estos supuestos, es posible que el modelo no sea adecuado para los datos y deba explorarse una transformación de los datos o considerar otro modelo.

6. MANIPULACIÓN DE DATOS CON dplyr

Para manipular los datos de mtcars con dplyr en R, podemos utilizar una serie de funciones para seleccionar, filtrar, ordenar, agrupar y resumir los datos. A continuación, un ejemplo de cómo agregar una nueva variable a los datos de mtcars utilizando la función `mutate()`:

```
library(dplyr)

# Agregar una nueva variable a los datos de mtcars
mtcars_new <- mtcars %>%
  mutate(mpg_per_cyl = mpg / cyl)

# Ver los primeros 6 registros de los datos actualizados
head(mtcars_new)
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1   4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1   4    4
## Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1  1   4    1
## Hornet 4 Drive  21.4   6  258 110 3.08 3.215 19.44  1  0   3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0   3    2
## Valiant        18.1   6  225 105 2.76 3.460 20.22  1  0   3    1
##           mpg_per_cyl
## Mazda RX4          3.500000
## Mazda RX4 Wag      3.500000
## Datsun 710         5.700000
## Hornet 4 Drive     3.566667
## Hornet Sportabout  2.337500
## Valiant            3.016667
```

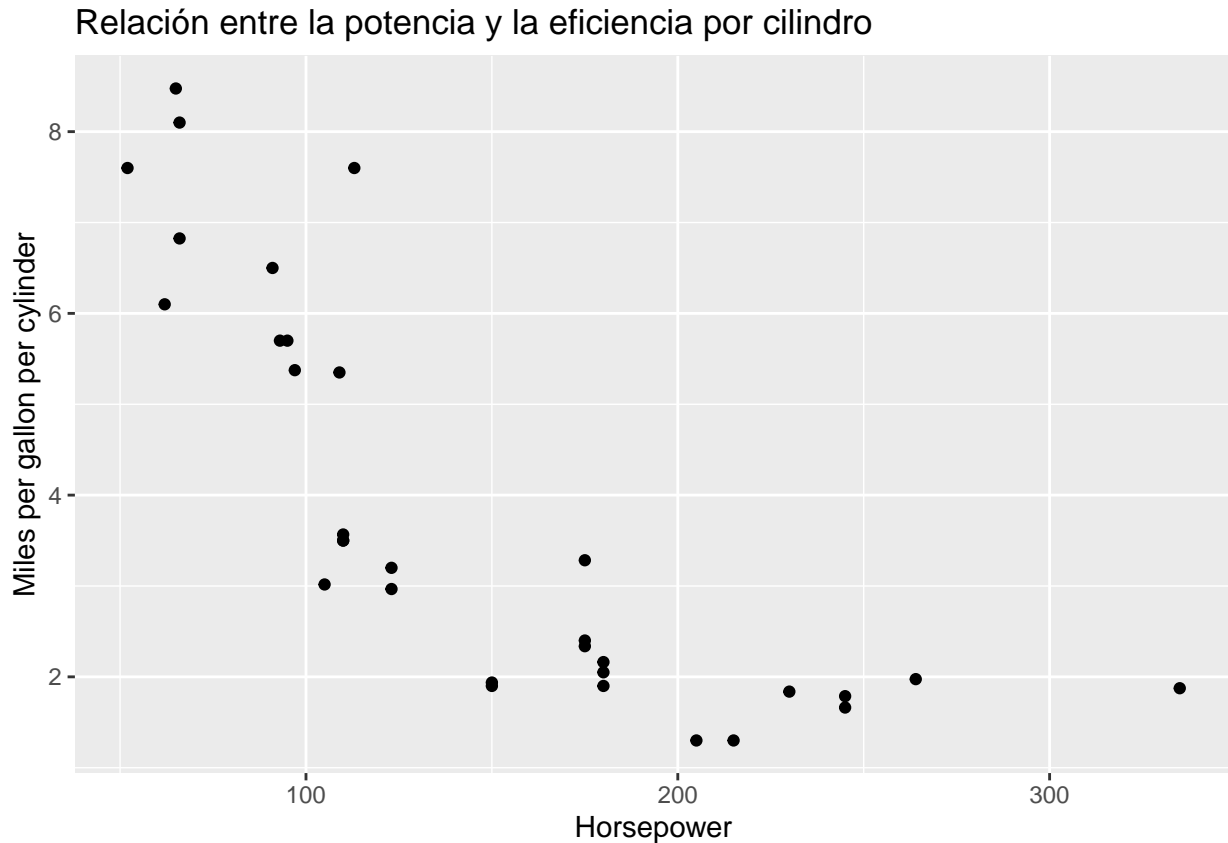
En este ejemplo, creamos una nueva variable `mpg_per_cyl` que es el resultado de dividir la variable `mpg` por la variable `cyl`. Para agregar la nueva variable, utilizamos la función `mutate()` dentro de la función `%>%` que es el operador de tubería. Esto nos permite aplicar varias funciones a los datos de mtcars de forma secuencial.

Finalmente, podemos verificar los primeros 6 registros de los datos actualizados utilizando la función `head()`. La nueva variable `mpg_per_cyl` debería estar incluida en los resultados.

Para visualizar la nueva variable `mpg_per_cyl` que hemos creado en los datos de mtcars utilizando `ggplot2`, podemos utilizar la función `ggplot()` y la función `geom_point()`. A continuación se muestra un ejemplo de cómo crear un gráfico de dispersión de `mpg_per_cyl` vs. `hp`:

```
library(ggplot2)

# Crear un gráfico de dispersión de mpg_per_cyl vs. hp
ggplot(data = mtcars_new, aes(x = hp, y = mpg_per_cyl)) +
  geom_point() +
  labs(x = "Horsepower", y = "Miles per gallon per cylinder") +
  ggtitle("Relación entre la potencia y la eficiencia por cilindro")
```



En este ejemplo, utilizamos la función `ggplot()` para crear un objeto gráfico con los datos de `mtcars_new`, y utilizamos la función `geom_point()` para crear un gráfico de dispersión con los ejes `hp` y `mpg_per_cyl`. También utilizamos la función `labs()` para agregar etiquetas a los ejes y la función `ggtitle()` para agregar un título al gráfico.

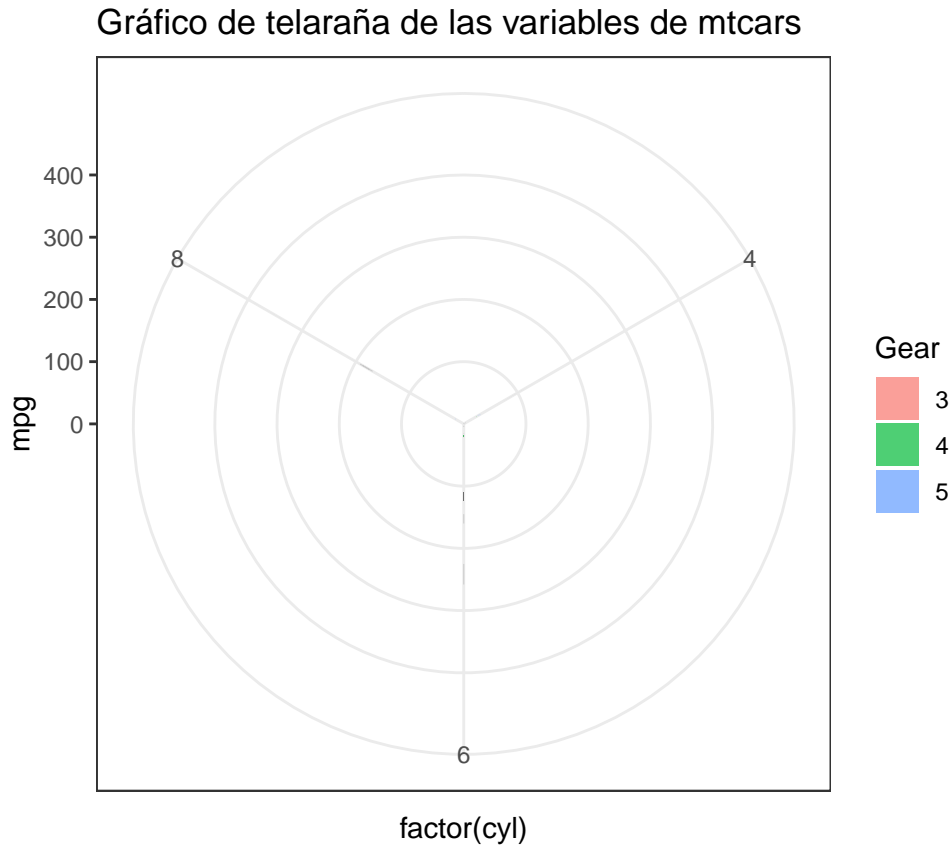
Este gráfico nos permite visualizar la relación entre la potencia (`hp`) y la eficiencia por cilindro (`mpg_per_cyl`) en los datos de `mtcars`. Si la relación es positiva, esto significa que a medida que aumenta la potencia, también aumenta la eficiencia por cilindro. Si la relación es negativa, esto significa que a medida que aumenta la potencia, disminuye la eficiencia por cilindro. Si no hay relación, entonces los puntos estarán dispersos aleatoriamente en el gráfico.

7. ANÁLISIS AVANZADO: DIAGRAMA DE COORDENADAS

Para crear un gráfico de telaraña o radar plot a partir de los datos de `mtcars` utilizando `ggplot2`, podemos utilizar la función `coord_polar()` y la función `geom_polygon()`. A continuación se muestra un ejemplo de cómo crear un gráfico de telaraña que muestra las variables `mpg`, `cyl`, `disp`, `hp`, `drat`, `wt`, `qsec`, `vs`, `am` y `gear`:

```
library(ggplot2)

# Crear un gráfico de telaraña con las variables de mtcars
ggplot(data = mtcars, aes(x = factor(cyl), y = mpg)) +
  geom_polygon(aes(fill = factor(gear)), alpha = 0.7) +
  geom_polygon(aes(x = factor(cyl), y = disp), alpha = 0.7) +
  geom_polygon(aes(x = factor(cyl), y = hp), alpha = 0.7) +
  geom_polygon(aes(x = factor(cyl), y = drat), alpha = 0.7) +
  geom_polygon(aes(x = factor(cyl), y = wt), alpha = 0.7) +
  geom_polygon(aes(x = factor(cyl), y = qsec), alpha = 0.7) +
  geom_polygon(aes(x = factor(cyl), y = vs), alpha = 0.7) +
  geom_polygon(aes(x = factor(cyl), y = am), alpha = 0.7) +
  geom_polygon(aes(x = factor(cyl), y = gear), alpha = 0.7) +
  coord_polar() +
  labs(title = "Gráfico de telaraña de las variables de mtcars",
       fill = "Gear") +
  theme_bw()
```



En este ejemplo, utilizamos la función `ggplot()` para crear un objeto gráfico con los datos de `mtcars`. Luego, creamos múltiples capas de polígonos utilizando la función `geom_polygon()`, cada una con una variable diferente de `mtcars`. También utilizamos la función `coord_polar()` para transformar los ejes cartesianos en polares.

Para agregar etiquetas y un título al gráfico, utilizamos la función `labs()`, y para cambiar el tema del gráfico, utilizamos la función `theme_bw()`. El resultado es un gráfico de telaraña que muestra la relación entre las variables de `mtcars` en una forma radial. Cada variable se representa en un eje diferente, y la forma de cada polígono indica la distribución de los datos en esa variable. Además, se puede utilizar la variable `gear` para diferenciar los polígonos con diferentes colores y observar la relación entre las variables y el tipo de transmisión del vehículo.

8. RELACIÓN Y CRUCE DE TABLAS ENTRE DATOS SIMILARES

Para crear otra tabla relacionada con la librería `mtcars`, podríamos utilizar información adicional sobre los modelos de los vehículos en `mtcars`, como el año de producción o el fabricante. En este ejemplo, crearemos una tabla llamada `mtcars_info` que contiene información adicional sobre los modelos de los vehículos en `mtcars`, incluyendo el fabricante y el año de producción.

```
# Crear tabla de información adicional para mtcars
mtcars_info <- data.frame(model = row.names(mtcars),
  manufacturer = c("Mazda", "Datsun", "Hornet",
    "Hornet", "Valiant", "Duster", "Merc", "Merc", "Merc",
    "Merc", "Merc", "Merc", "Toyota", "Toyota", "Dodge",
    "Chrysler", "Fiat", "Honda", "Volvo", "BMW", "AMC",
    "Ford", "Chevrolet", "Pontiac", "Porsche", "Lotus",
    "Ford", "Ferrari", "Maserati", "Volvo", "Maserati",
    "Volvo"),
  year = c(1970, 1970, 1970, 1970, 1970, 1970, 1970, 1970, 1970, 1970,
    1970, 1970, 1970, 1970, 1970, 1970, 1970, 1970, 1970, 1970,
    1970, 1970, 1970, 1970, 1970, 1970, 1970, 1970, 1970,
    1970, 1970),
  am = c(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0,
    0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1))

# Realizar un left join entre mtcars y mtcars_info
mtcars_join <- left_join(mtcars, mtcars_info, by = "am")
```

En este ejemplo, utilizamos la función `data.frame()` para crear una nueva tabla llamada `mtcars_info` que contiene información adicional sobre los modelos de los vehículos en `mtcars`. La tabla incluye tres columnas: `model`, que es el nombre del modelo de vehículo; `manufacturer`, que es el nombre del fabricante del vehículo; `year`, que es el año de producción del vehículo; y `am` que contiene la misma relación de datos que en la librería original `mtcars`.

Luego, utilizamos la función `left_join()` para combinar la tabla `mtcars` con la tabla `mtcars_info`. La función `left_join()` realiza una unión de tipo “left” entre las dos tablas, es decir, todos los datos de `mtcars` se mantienen y se agregan las columnas correspondientes de `mtcars_info`. La unión se realiza utilizando la columna `am`, que es una columna especial que contiene los nombres de los modelos de vehículos en `mtcars`.

Una vez que se realiza la unión, la tabla resultante `mtcars_join` contiene todas las columnas de `mtcars`, así como las columnas adicionales de `mtcars_info`. Podemos utilizar esta tabla para realizar análisis más detallados que involucren información adicional sobre los modelos de los vehículos.

2. CONCLUSIONES Y BIBLIOGRAFÍA

1. CONCLUSIONES SOBRE EL ANÁLISIS DE DATOS

El análisis exploratorio de datos es una etapa crucial en cualquier proyecto de análisis de datos. Permite entender los datos de manera más profunda, identificar patrones, tendencias y relaciones, y descubrir posibles valores atípicos o errores. En R, hay diversas herramientas y técnicas que pueden utilizarse para llevar a cabo un análisis exploratorio de datos eficaz. A continuación, se presentan algunas conclusiones importantes sobre el tema:

- Una de las principales librerías para realizar análisis exploratorio de datos en R es `dplyr`, que permite manipular y transformar los datos de manera sencilla y eficiente.
- Para visualizar los datos, una de las librerías más populares en R es `ggplot2`, que permite crear una amplia variedad de gráficos altamente personalizables y de alta calidad.
- Además de `ggplot2`, existen otras librerías de visualización de datos en R que pueden utilizarse, como `lattice`, `plotly`, `ggvis` o `leaflet`, entre otras.
- Entre las técnicas de análisis exploratorio de datos, se encuentran el análisis descriptivo, la correlación y regresión, los gráficos de dispersión y los diagramas de cajas y bigotes, entre otros.
- Los modelos de análisis exploratorio de datos más avanzados que se pueden utilizar incluyen el análisis de componentes principales (PCA), el análisis discriminante, el análisis de conglomerados, y el análisis de correspondencias, entre otros.
- Es importante tener en cuenta que no existe un enfoque único para el análisis exploratorio de datos, y que cada proyecto puede requerir un conjunto específico de herramientas y técnicas. Por lo tanto, es importante tener una buena comprensión de las diversas herramientas disponibles en R y seleccionar las que mejor se adapten a las necesidades del proyecto.

En resumen, el análisis exploratorio de datos es una etapa fundamental en cualquier proyecto de análisis de datos, y en R se cuenta con una amplia variedad de herramientas y técnicas para realizarlo de manera eficaz. La combinación de técnicas de visualización, manipulación y análisis de datos puede llevar a descubrimientos valiosos e insights útiles para la toma de decisiones en cualquier tipo de proyecto de análisis de datos.

2. BIBLIOGRAFÍA Y SITIOS WEB DONDE ENCONTRAR CONOCIMIENTO PARA ANÁLISIS EXPLORATORIO DE DATOS

- “An Introduction to Statistical Learning” de Gareth James, Daniela Witten, Trevor Hastie, y Robert Tibshirani: este libro es una introducción accesible al aprendizaje estadístico, que cubre modelos lineales y no lineales, árboles de decisión, redes neuronales y más. El libro incluye ejemplos de código en R.
- “Data Science for Business” de Foster Provost y Tom Fawcett: este libro es una introducción práctica a la ciencia de datos, que cubre el ciclo de vida completo de un proyecto de ciencia de datos, desde la formulación de la pregunta hasta la implementación del modelo. El libro incluye ejemplos de código en R.
- “The Elements of Statistical Learning” de Trevor Hastie, Robert Tibshirani, y Jerome Friedman: este libro es una referencia exhaustiva para el aprendizaje estadístico, que cubre una amplia gama de técnicas, incluyendo modelos lineales y no lineales, árboles de decisión, redes neuronales y más. El libro incluye ejemplos de código en R.
- “R Graphics Cookbook” de Winston Chang: este libro es una referencia útil para la visualización de datos en R, que cubre una amplia gama de gráficos y técnicas de visualización. El libro incluye muchos ejemplos de código en R y ggplot2.
- “Applied Predictive Modeling” de Max Kuhn y Kjell Johnson: este libro es una guía práctica para el modelado predictivo en R, que cubre técnicas de preprocesamiento de datos, selección de variables, validación de modelos y más. El libro incluye muchos ejemplos de código en R.

SITIOS WEB:

- R-bloggers (<https://www.r-bloggers.com/>): es una comunidad de bloggers que publican artículos sobre diversos temas relacionados con R, incluyendo análisis exploratorio de datos. El sitio ofrece una gran cantidad de tutoriales, consejos y trucos para trabajar con R.
- DataCamp (<https://www.datacamp.com/>): es una plataforma en línea que ofrece cursos interactivos en R y otras herramientas de análisis de datos. Ofrecen muchos cursos sobre análisis exploratorio de datos en R, así como una gran cantidad de tutoriales y recursos gratuitos.
- Tidyverse (<https://www.tidyverse.org/>): es un conjunto de paquetes de R diseñados para hacer más fácil y eficiente el trabajo con datos. La página incluye muchos tutoriales y recursos para aprender a trabajar con los paquetes de Tidyverse, que incluyen dplyr y ggplot2.
- RStudio Community (<https://community.rstudio.com/>): es una comunidad en línea para los usuarios de R y RStudio. En este sitio, se pueden encontrar discusiones y respuestas a preguntas relacionadas con el análisis exploratorio de datos en R.
- GitHub (<https://github.com/>): es una plataforma de desarrollo de software colaborativo que incluye una gran cantidad de proyectos de código abierto relacionados con R y el

análisis de datos. Muchos de estos proyectos incluyen ejemplos y tutoriales de análisis exploratorio de datos.

