ESCUELA POLITECNICA NACIONAL

INGENIERIA EN CIENCIAS DE LA COMPUTACION

Data Mining y Machine Learning

3 Hands On: Data Exploration

Daniel Samaniego Zapata

GR2CC_2023-1

1 Summarization

Cargue el conjunto de datos carIns final. Ya tiene la imputación de valores faltantes.

1. Usando el paquete dplyr, responda las siguientes preguntas:

(a) Obtenga el número de automóviles por estilo de carrocería.

```
library('dplyr')
```

```
## Warning: package 'dplyr' was built under R version 4.3.1
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Carga el archivo .Rdata
df <- load("C:\\Users\\Dany\\Documents\\R\\Hand 03\\carIns_final.Rdata")
#df <- load("Hand On 03\\data\\carIns_final.Rdata")

#carIns_final$bodyStyle

#unique(carIns_final$bodyStyle)

carIns_final %>% group_by(bodyStyle) %>% count()
```

```
## # A tibble: 5 x 2
## # Groups:   bodyStyle [5]
##   bodyStyle       n
##   <fct>       <int>
## 1 convertible     6
```

```
## 2 hardtop         8
## 3 hatchback      70
## 4 sedan          96
## 5 wagon          25
```

(b) Obtenga el número de automóviles por bodyStyle y fuelType.

```
carIns_final%>% group_by(bodyStyle, fuelType) %>% count()
```

```
## # A tibble: 9 x 3
## # Groups:   bodyStyle, fuelType [9]
##   bodyStyle   fuelType     n
##   <fct>       <fct>    <int>
## 1 convertible gas          6
## 2 hardtop     diesel       1
## 3 hardtop     gas          7
## 4 hatchback   diesel       1
## 5 hatchback   gas         69
## 6 sedan       diesel      15
## 7 sedan       gas         81
## 8 wagon       diesel       3
## 9 wagon       gas         22
```

(c) Obtenga la media y la desviación estándar del atributo cityMpg por bodyStyle en orden ascendente.

```
carIns_final %>% group_by(bodyStyle) %>% summarise(MediaCityMpg = mean(cityMpg), DesviacionCityMpg = sd
```

```
## # A tibble: 5 x 3
##   bodyStyle   MediaCityMpg DesviacionCityMpg
##   <fct>              <dbl>             <dbl>
## 1 convertible         20.5              3.39
## 2 hardtop             21.6              5.42
## 3 hatchback           26.3              7.17
## 4 sedan               25.3              6.60
## 5 wagon               24.0              4.22
```

(d) También por body Style, y para los atributos city Mpg and highway Mpg, obtenga la media, la desviación estándar, la mediana y el rango intercuartil

```
carIns_final %>% group_by(bodyStyle) %>% summarise(Media_CityMpg = mean(cityMpg), Desviacion_CityMpg = s
```

```
## # A tibble: 5 x 9
##   bodyStyle   Media_CityMpg Desviacion_CityMpg Mediana_CityMpg Rango_CityMpg
##   <fct>               <dbl>              <dbl>           <dbl>         <dbl>
## 1 convertible          20.5               3.39              21          5.25
## 2 hardtop              21.6               5.42              23          7
## 3 hatchback            26.3               7.17              26         12
## 4 sedan                25.3               6.60              25         11.2
## 5 wagon                24.0               4.22              24          5
## # i 4 more variables: Media_highwayMpg <dbl>, Desviacion_highwayMpg <dbl>,
## #   Mediana_highwayMpg <dbl>, Rango_highwayMpg <dbl>
```

2 Visualization

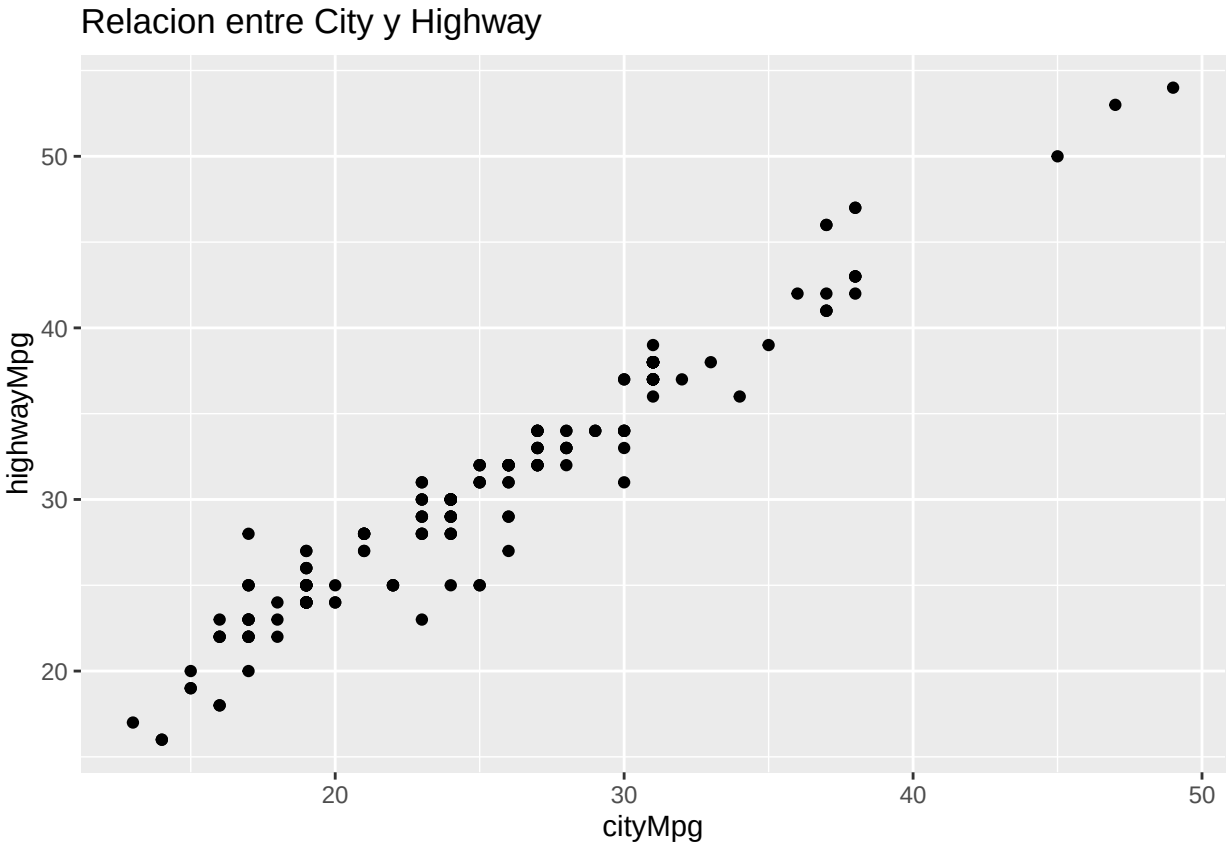
2. Con el paquete ggplot2, cree gráficos que le parezcan adecuados para responder a las siguientes preguntas.

(e) Muestre la relación entre los atributos cityMpg y highwayMpg

```
library("ggplot2")
```

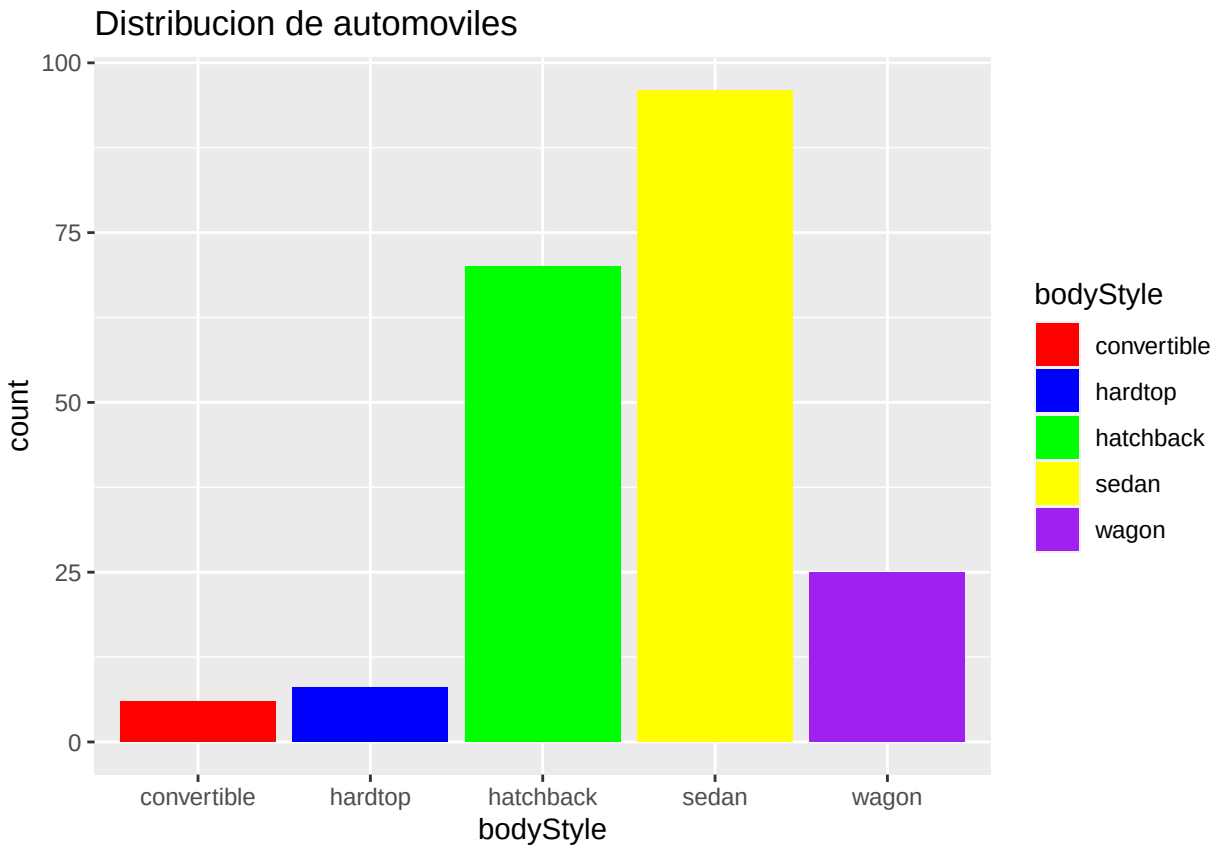
```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

```
# ggplot crea graficos para presentar los resultados
ggplot(carIns_final, aes(x = cityMpg, y = highwayMpg)) +
  geom_point() +
  ggtitle("Relacion entre City y Highway")  # Agrega un titulo
```



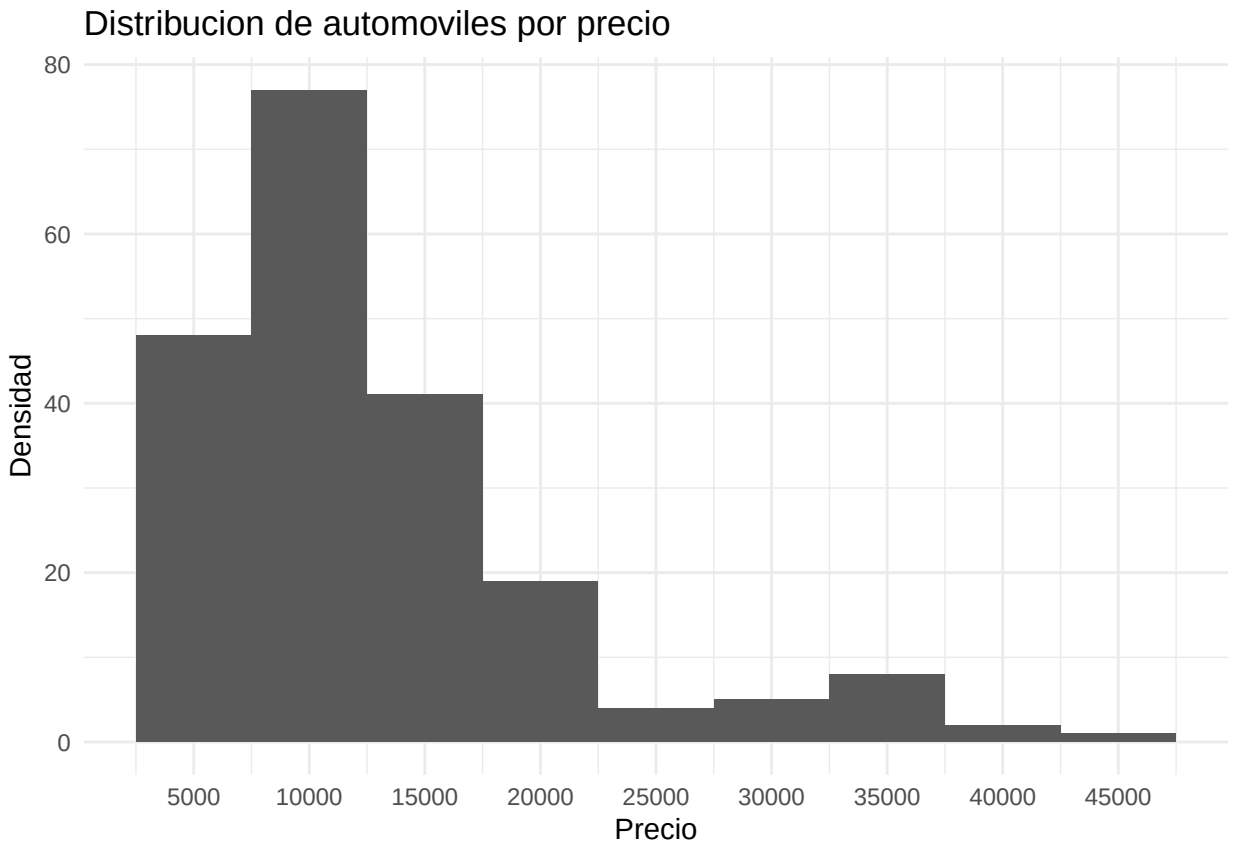
(f) Mostrar la distribución de automóviles por bodyStyle.

```
ggplot(carIns_final, aes(x = bodyStyle, fill = bodyStyle)) +
  geom_bar() +
  scale_fill_manual(values = c('red', 'blue', 'green', 'yellow', 'purple')) +
  ggtitle("Distribucion de automoviles")
```



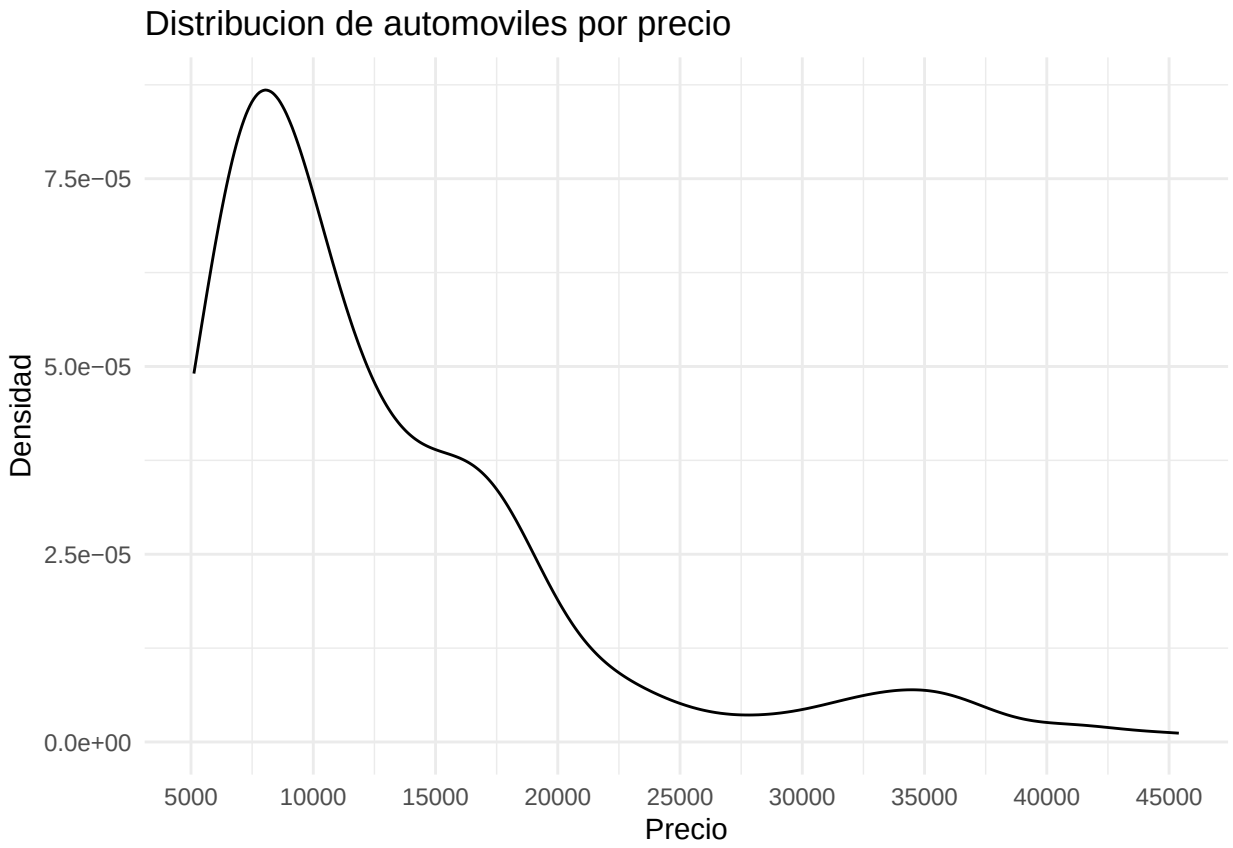
(g) Mostrar la distribución de automóviles por precio. Sugerencia: cree contenedores de ancho igual a 5000.

```r
ggplot(carIns_final, aes(x = price)) +
  #geom_density(adjust = 1) +   # Crea grafico de densidad
  geom_histogram(binwidth = 5000) +
  # Ajusta los contenedores de ancho igual a 5000
  scale_x_continuous(breaks = seq(0, max(carIns_final$price), 5000)) +
  labs(x = 'Precio', y = 'Densidad', title = 'Distribucion de automoviles por precio') +
  theme_minimal()
```



(h) Agregue la información de la estimación de densidad al gráfico anterior.

```
ggplot(carIns_final, aes(x = price)) +
  geom_density() +   # Crea grafico de densidad
  # Ajusta los contenedores de ancho igual a 5000
  scale_x_continuous(breaks = seq(0, max(carIns_final$price), 5000)) +
  labs(x = 'Precio', y = 'Densidad', title = 'Distribucion de automoviles por precio') +
  theme_minimal()
```



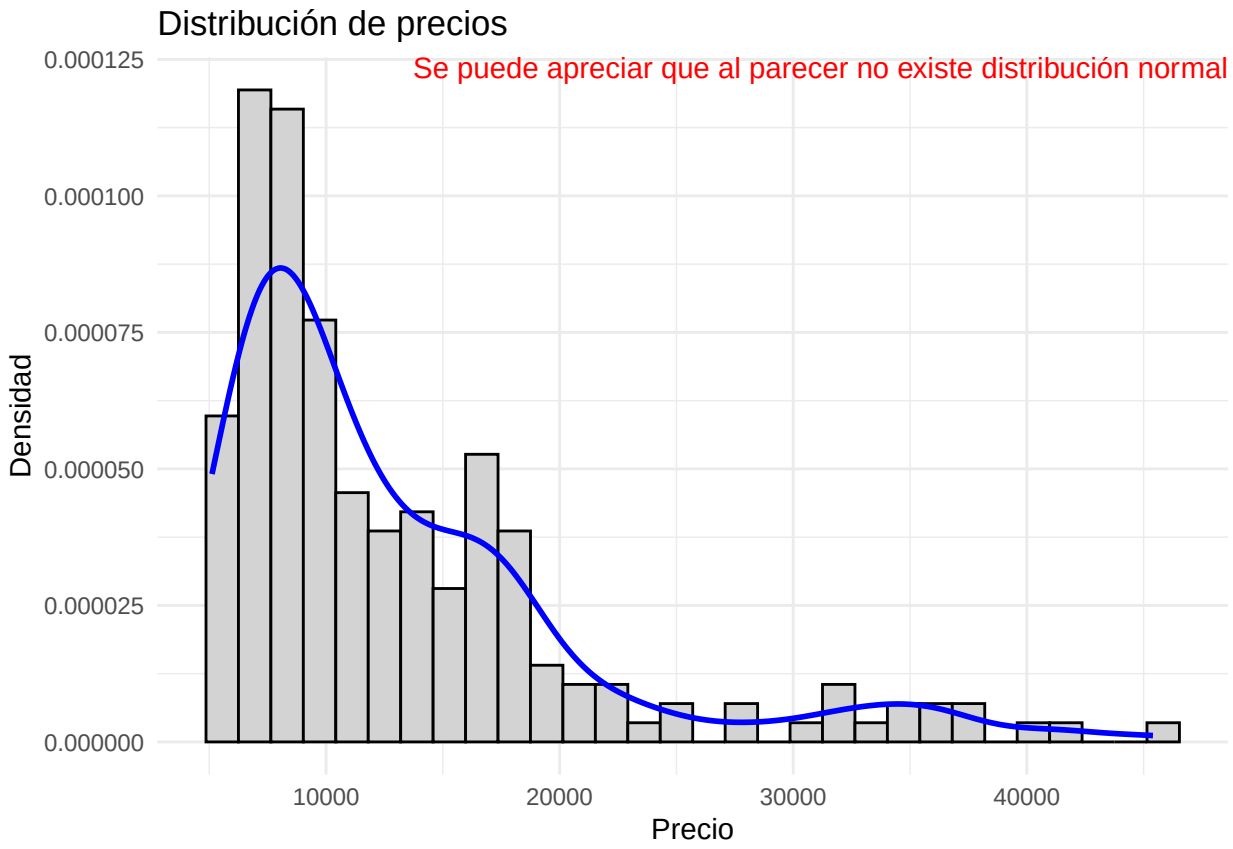
(i) Compruebe (visualmente) si es plausible considerar que el precio sigue una distribución normal.

```
library(ggplot2)

# Histograma de precios
ggplot(carIns_final, aes(x = price)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "lightgray", color = "black") +
  labs(x = "Precio", y = "Densidad", title = "Distribución de precios") +
  theme_minimal() +

  # Curva de densidad teórica
  geom_density(color = "blue", size = 1) +

  # Anotación al final del gráfico
  annotate("text", x = Inf, y = Inf, label = "Se puede apreciar que al parecer no existe distribución n
           hjust = 1, vjust = 1, size = 4, color = "red")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

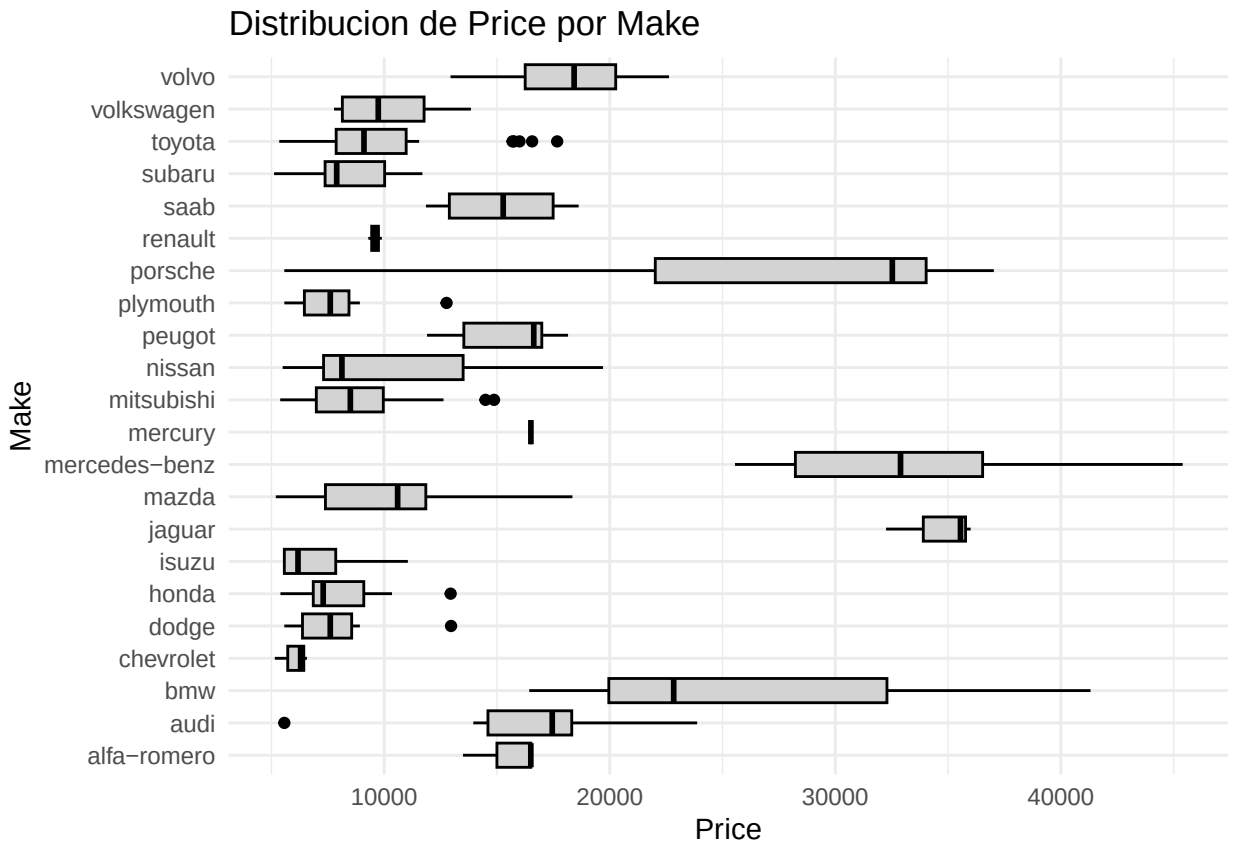
```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



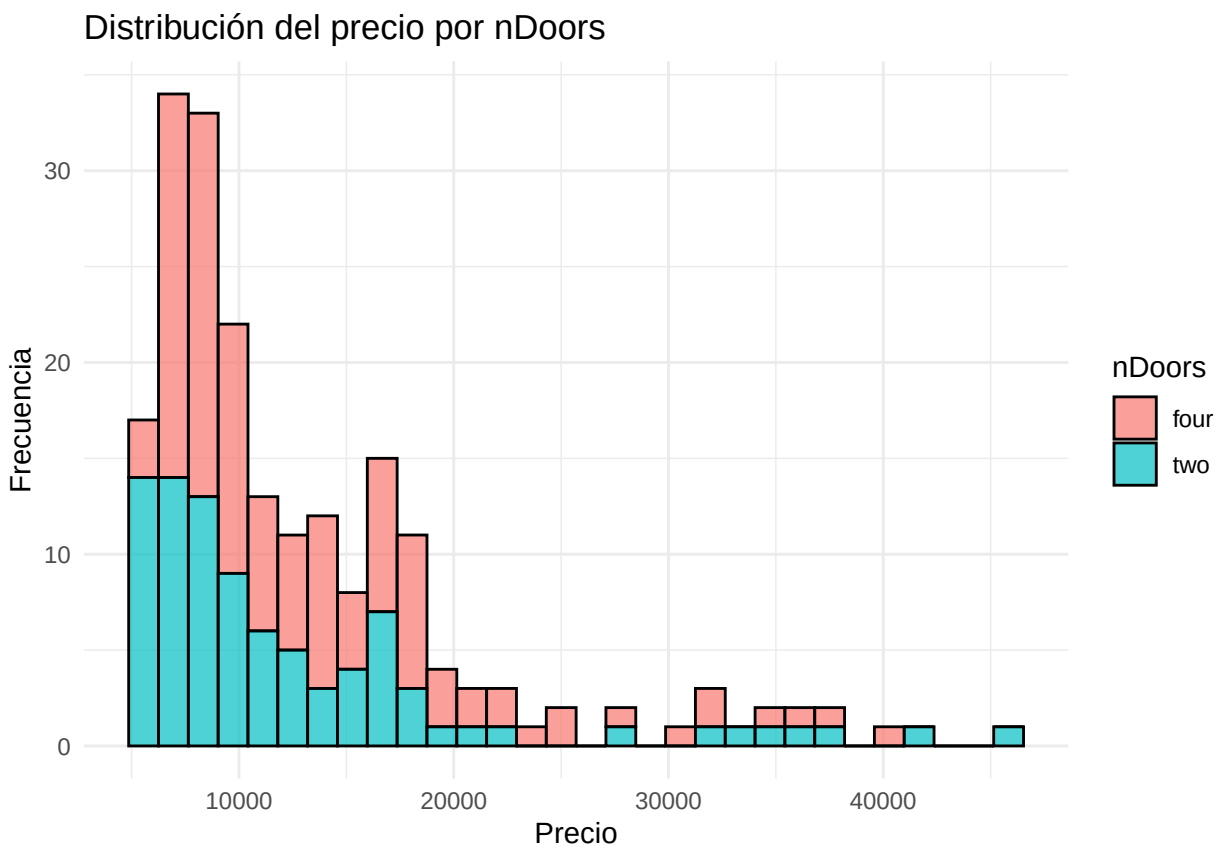
(j) Mostrar la distribución del price por atributo make. Sugerencia: use diagramas de caja y la función coord_flip().

```r
library(ggplot2)

# Diagrama de caja de distribucion de precio por make
ggplot(carIns_final, aes(x = make, y= price)) +
  # Traza los diagramas de caja
  geom_boxplot(fill = "lightgray", color = "black") +  # fill = color de relleno, color = indica el col
  coord_flip() +    # El grafico se muestra horizontalmente
  labs(x = "Make", y = "Price", title = "Distribucion de Price por Make") +
  theme_minimal()
```



(k) Muestre la distribución del price por atributo en nDoors. Sugerencia: utilice histogramas.

```r
library(ggplot2)

# Histograma de la distribución del precio por indoors
ggplot(carIns_final, aes(x = price, fill = nDoors)) +
  geom_histogram(bins = 30, color = "black", alpha = 0.7) +
  # bins indica el numero de intervalos del histograma
  # alpha defina la transparencia de las barras
  labs(x = "Precio", y = "Frecuencia", title = "Distribución del precio por nDoors") +
  theme_minimal()
```



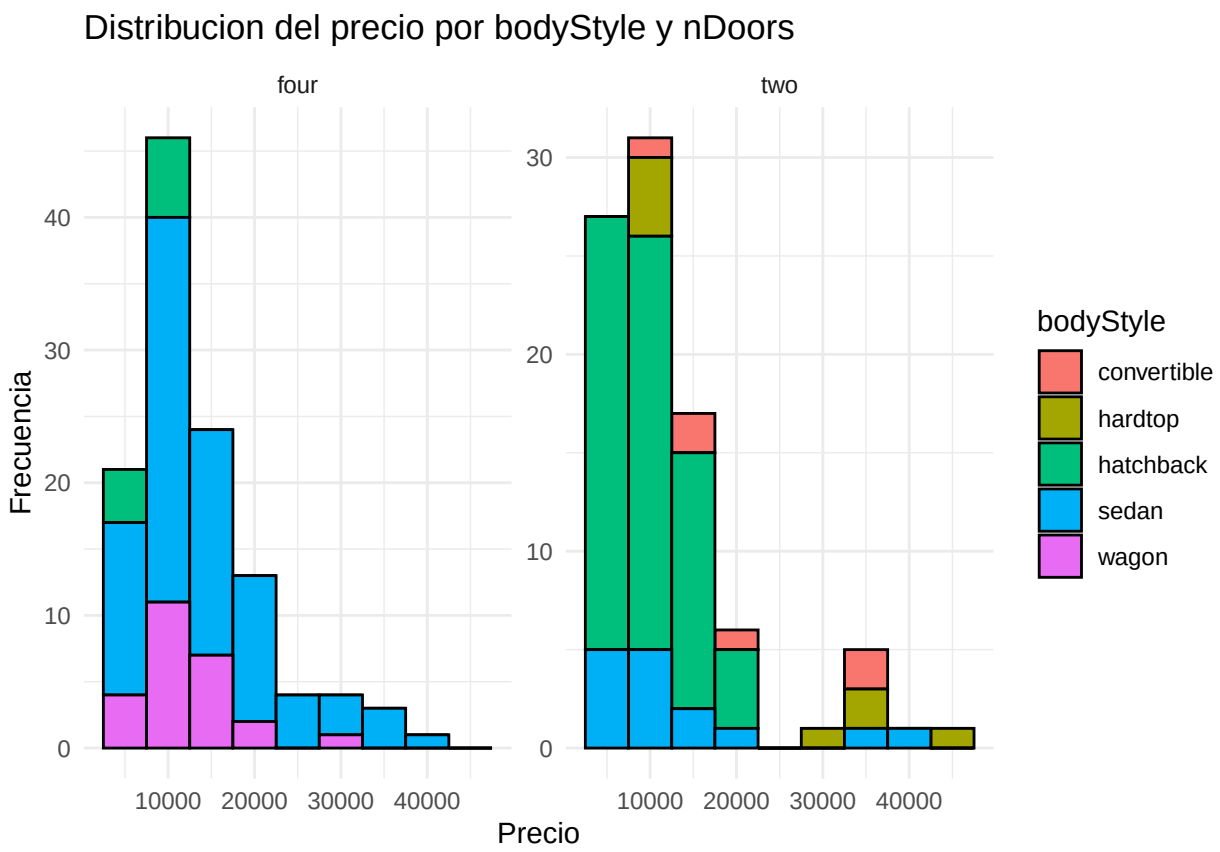
(l) Muestre la distribución del price por bodyStyle y el atributos nDoors. Sugerencia: utilice histogramas.

```
library(ggplot2)

# Histograma de la distribucion del precio por bodyStyle y nDoors
ggplot(carIns_final, aes(x = price, fill = bodyStyle)) +
  geom_histogram(position = "fill", bins = 30, color = "black", alpha = 0.7) +
  # position = "fill" muestra las barras proporcionales al total de cada grupo
  # bins = 30 establece el numero de intervales del histograma
  facet_wrap(~nDoors) +     # Agrega paneles separados por los valores unicos del atributo "nDoors"
  labs(x = "Precio", y = "Frecuencia", title = "Distribucion del precio por bodyStyle y nDoors") +
  theme_grey()
```

```
## Warning: Removed 60 rows containing missing values (`geom_bar()`).
```

(m) Agregue el parámetro scales="free_y" a la función de faceta en el gráfico anterior.

```
library(ggplot2)

ggplot(carIns_final, aes(x = price, fill = bodyStyle)) +
  geom_histogram(binwidth = 5000, color = "black", aes(y = after_stat(count))) +
  # aes(y = ..count..) muestra el conteo de observaciones en el eje y del grafico de histograma
  facet_wrap(~nDoors, scales = "free_y") +
  # Al agregar scales = "free_y" a facet_wrap, cada panel del grafico tiene su propia escala en el eje
  labs(x = "Precio", y = "Frecuencia", title = "Distribucion del precio por bodyStyle y nDoors") +
  theme_minimal()
```




```
# two y four son las categorias de la columna nDoors
```