3 Hands On: Data Exploration

1.Summarization

Load the data set carIns final. It already has the imputation of missing values.

(a) Obtain the number of cars by bodyStyle.

```
library(dplyr)
```

```
##
## 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
```

```
datos <- load("C:/Proyectos ML/Mineria/Hands On_3 Data Exploration/data/carIns_final.Rdata")
data <- as.data.frame(carIns_final)
leerDato<- data%>%group_by(bodyStyle)%>% count()


library(flextable)
```

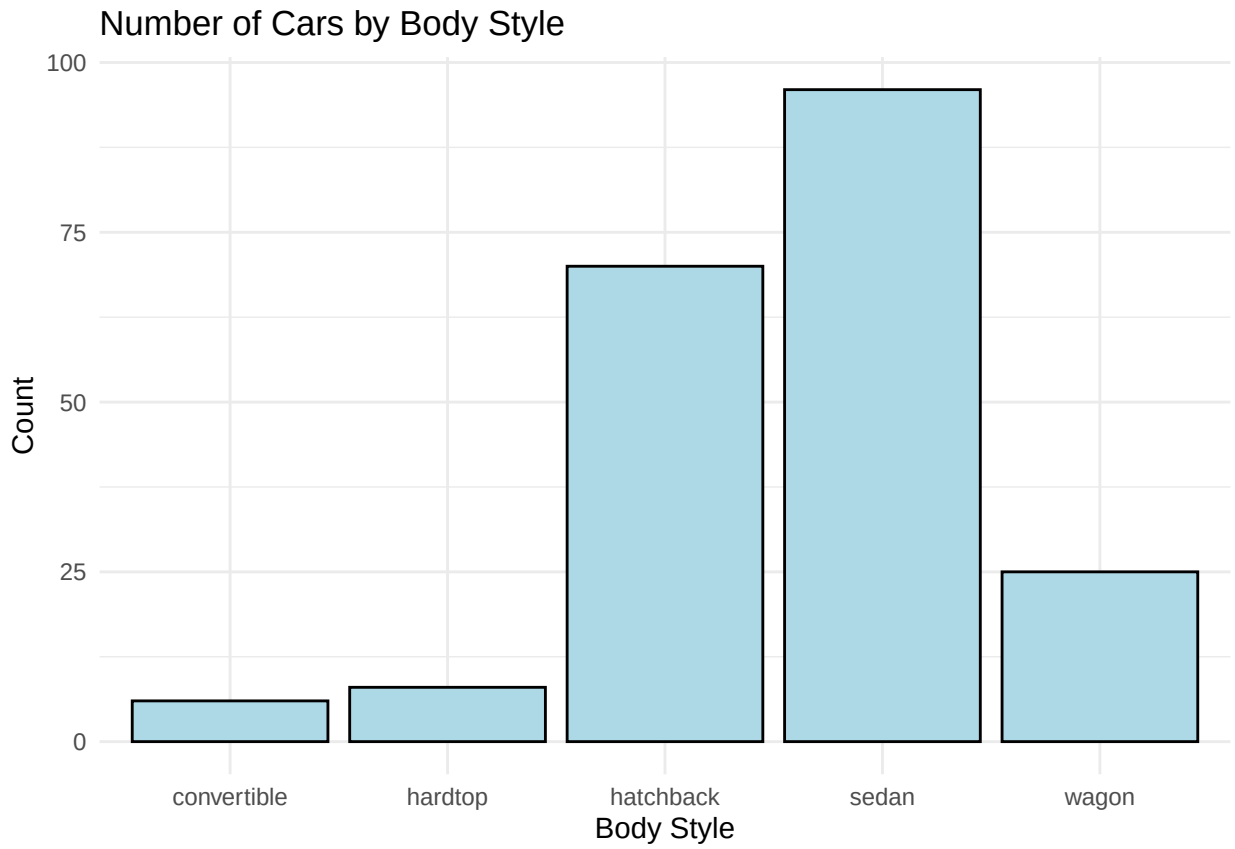
```
## Warning: package 'flextable' was built under R version 4.3.1
```

```
# Crear una tabla utilizando flextable()
tabla <- flextable(leerDato)

# Ajustar automáticamente el ancho de las columnas
tabla <- autofit(tabla)

# Imprimir la tabla
#print(tabla)
print(leerDato)
```

```
## # 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
```

```
library(ggplot2)

# Create a bar plot of car counts by bodyStyle
ggplot(data, aes(x = bodyStyle)) +
  geom_bar(fill = "lightblue", color = "black") +
  labs(x = "Body Style", y = "Count") +
  ggtitle("Number of Cars by Body Style") +
  theme_minimal()
```



(b) Obtain the number of cars by bodyStyle and fuelType

```
data %>% 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) Obtain the mean and the standard deviation of the attribute cityMpg by bodyStyle in ascending order.

```
library(ggplot2)


valores<-carIns_final %>% group_by(bodyStyle)%>%summarise(cityMpg.mean = mean(cityMpg),
          cityMpg.sd = sd(cityMpg) )%>% arrange(cityMpg.mean)

print(valores)
```

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

(d) Also by bodyStyle, and for the attributes cityMpg and highwayMpg, obtain the mean, the standard deviation, the median and the inter-quartile range.

```
library(ggplot2)
  carIns_final %>% group_by(bodyStyle)%>% summarise(cityMpg.mean = mean(cityMpg),
  cityMpg.sd = sd(cityMpg),carretera = IQR(highwayMpg))
```

```
## # A tibble: 5 x 4
##   bodyStyle   cityMpg.mean cityMpg.sd carretera
##   <fct>              <dbl>      <dbl>     <dbl>
## 1 convertible         20.5       3.39       3
## 2 hardtop             21.6       5.42       5
## 3 hatchback           26.3       7.17      11.8
## 4 sedan               25.3       6.60      11.2
## 5 wagon               24.0       4.22       7
```

```
library(flextable)
tabla <- flextable(carIns_final)
tabla <- autofit(tabla)
print(tabla)
```

```
## a flextable object.
## col_keys: ‘symb‘, ‘normLoss‘, ‘make‘, ‘fuelType‘, ‘aspiration‘, ‘nDoors‘, ‘bodyStyle‘, ‘driveWheels‘
## header has 1 row(s)
```

```
## body has 205 row(s)
## original dataset sample:
##   symb normLoss        make fuelType aspiration nDoors   bodyStyle driveWheels
## 1    3      161 alfa-romero      gas        std    two convertible         rwd
## 2    3      161 alfa-romero      gas        std    two convertible         rwd
## 3    1      161 alfa-romero      gas        std    two   hatchback         rwd
## 4    2      164        audi      gas        std   four       sedan         fwd
## 5    2      164        audi      gas        std   four       sedan         4wd
##   engineLocation wheelBase length width height curbWeight engineType nrCylinds
## 1          front      88.6  168.8  64.1   48.8       2548       dohc      four
## 2          front      88.6  168.8  64.1   48.8       2548       dohc      four
## 3          front      94.5  171.2  65.5   52.4       2823       ohcv       six
## 4          front      99.8  176.6  66.2   54.3       2337        ohc      four
## 5          front      99.4  176.6  66.4   54.3       2824        ohc      five
##   engineSize fuelSystem bore stroke compressionRatio horsePower peakRpm cityMpg
## 1        130       mpfi 3.47   2.68                9        111    5000      21
## 2        130       mpfi 3.47   2.68                9        111    5000      21
## 3        152       mpfi 2.68   3.47                9        154    5000      19
## 4        109       mpfi 3.19   3.40               10        102    5500      24
## 5        136       mpfi 3.19   3.40                8        115    5500      18
##   highwayMpg price
## 1         27 13495
## 2         27 16500
## 3         26 16500
## 4         30 13950
## 5         22 17450
```

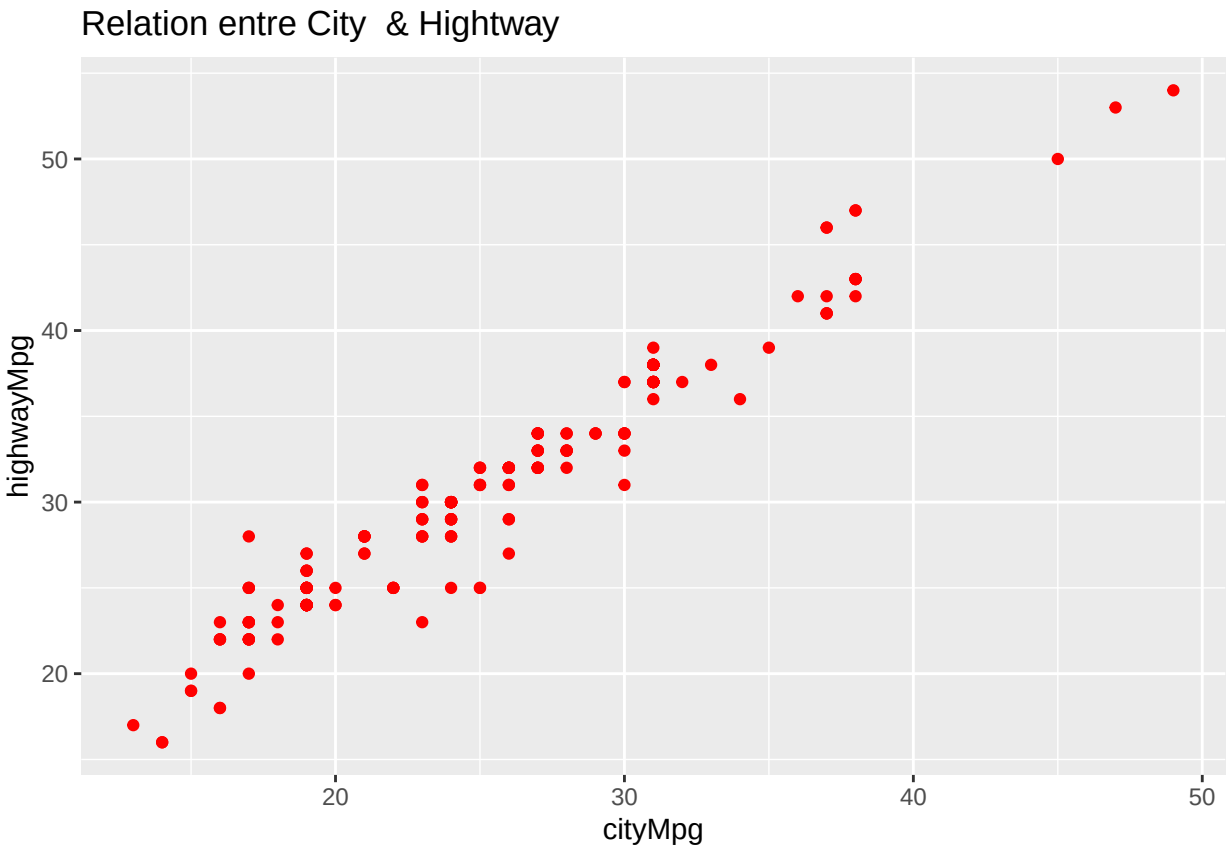
```
#   library(gt)
#carIns_final %>% gt()
```

2. Visualization

Using the package ggplot2, create graphs that you find adequate to answer the following questions

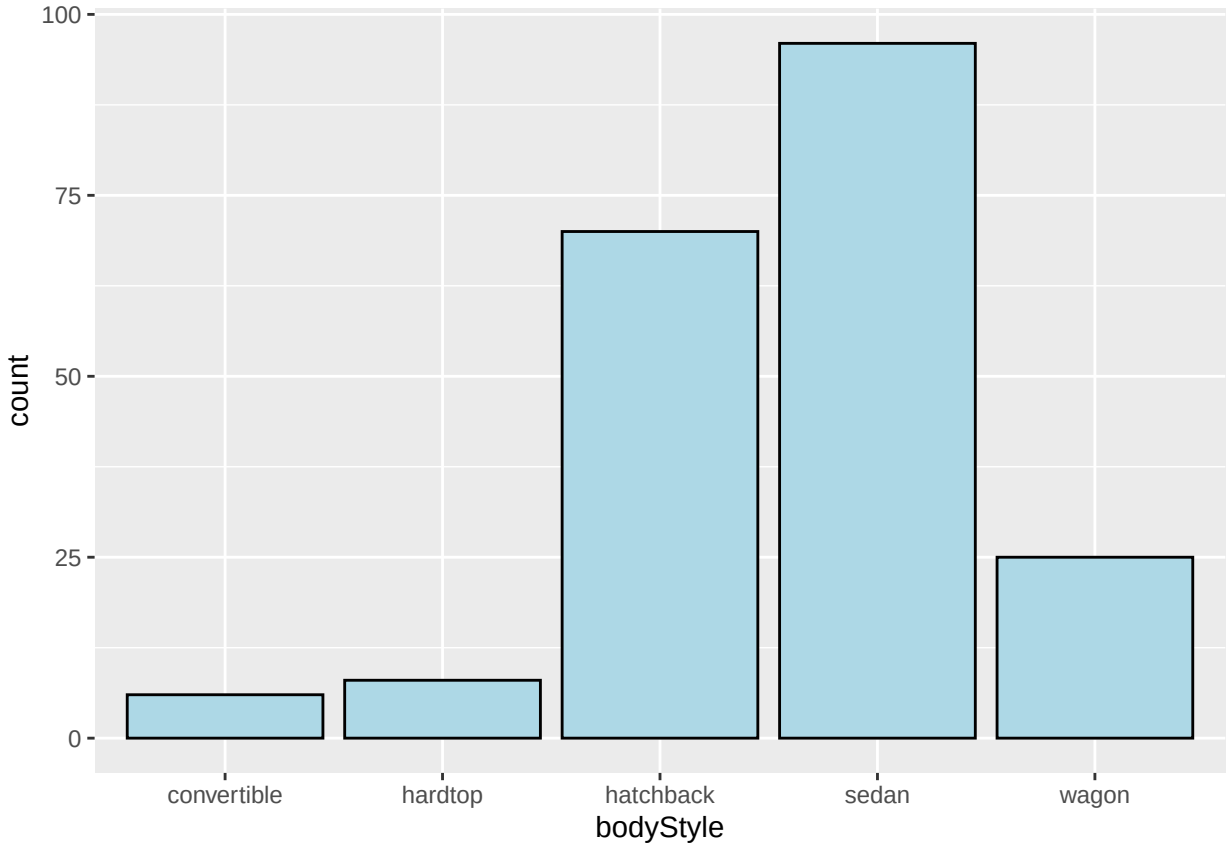
(e) Show the relationship between the attributes cityMpg and highwayMpg

```
ggplot(carIns_final,aes(x = cityMpg, y = highwayMpg))+geom_point( color = "red")+
  ggtitle("Relation entre City  & Hightway")
```



(f) Show the distribution of cars by bodyStyle.

```
ggplot(carIns_final,aes(x = bodyStyle))+ geom_bar(fill = "lightblue", color = "black")
```

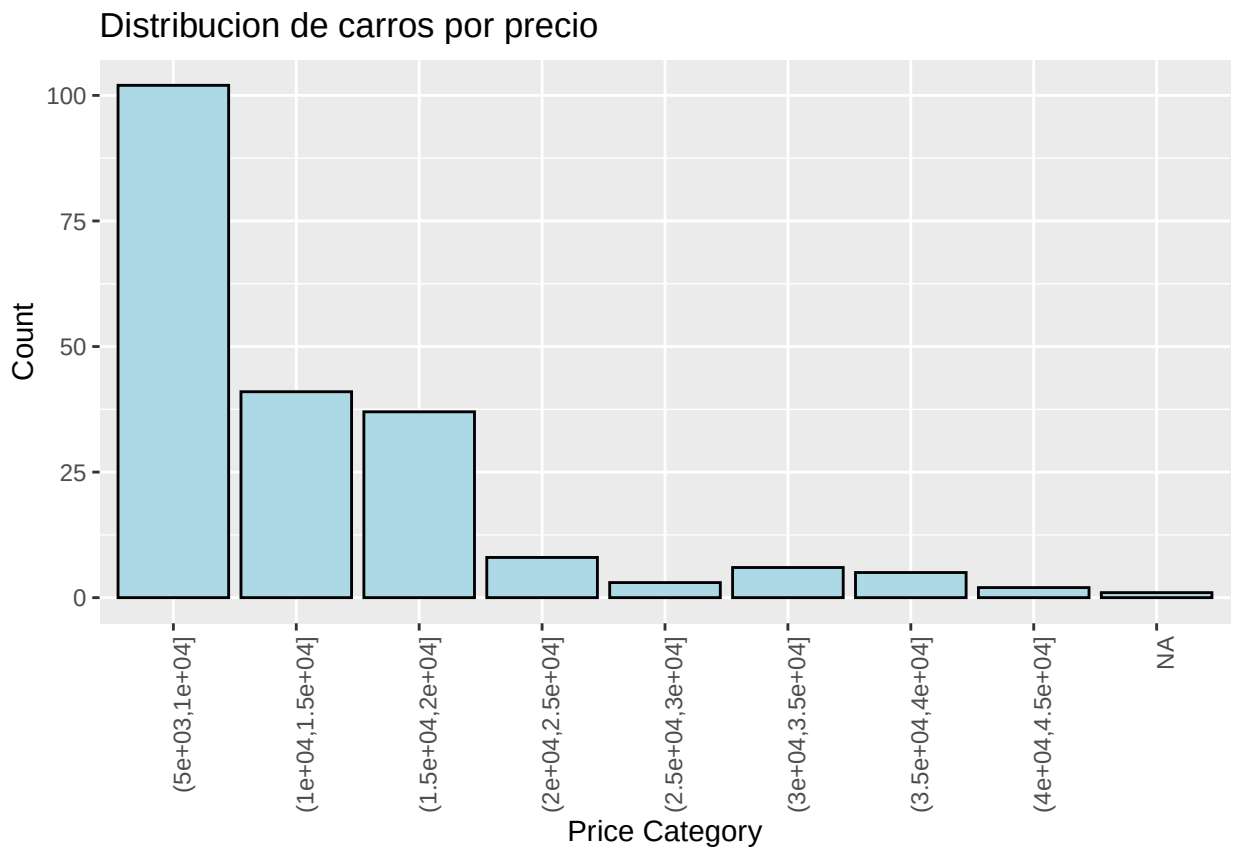


(g) Show the distribution of cars by price. Suggestion: create bins of width equal to 5000.

```
# Create price bins with a width of 5000
price_bins <- seq(0, max(data$price), by = 5000)

print(price_bins)
```

```
##  [1]     0  5000 10000 15000 20000 25000 30000 35000 40000 45000
```

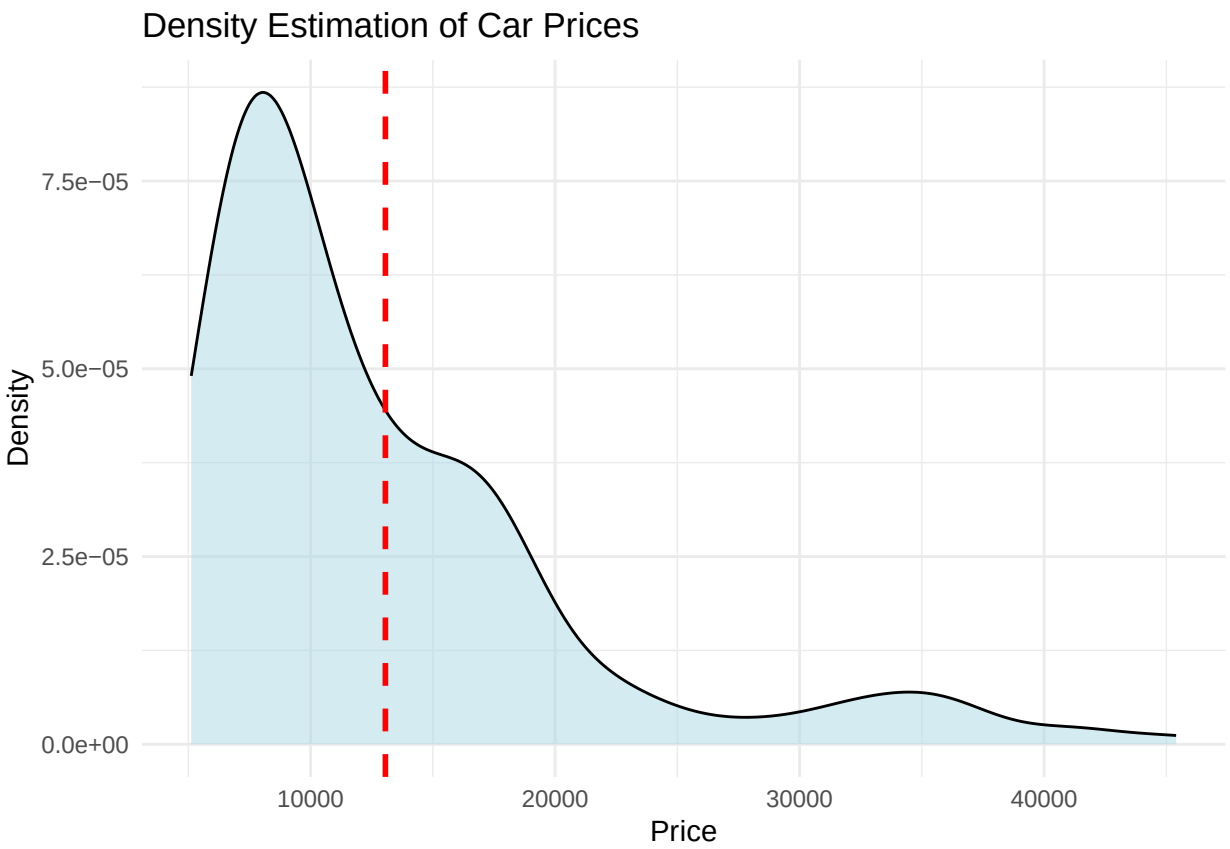
```
# Cut the prices into bins
price_categories <- cut(data$price, breaks = price_bins, include.lowest = TRUE)
ggplot(carIns_final,aes(x=price_categories))+geom_bar(fill = "lightblue", color = "black")+labs(x = "Pr
  ggtitle("Distribucion de carros por precio")+theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



(h) Add the information of the density estimation to the previous graph

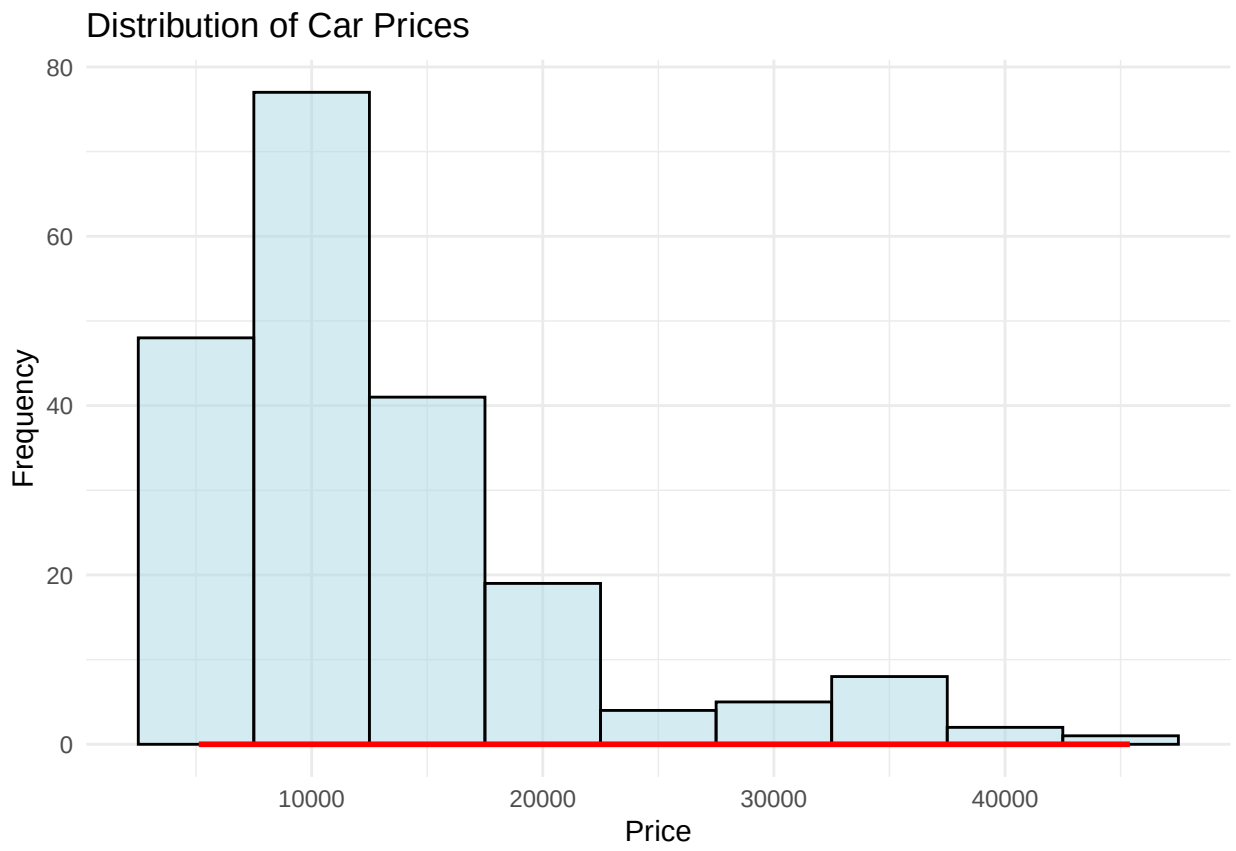
```
library(ggplot2)

# Create price bins with a width of 5000
price_bins <- seq(0, max(carIns_final$price), by = 5000)


# Cut the prices into bins
data$price_category <- cut(data$price, breaks = price_bins, include.lowest = TRUE)

# Create a density plot of car prices
ggplot(data, aes(x = price)) +
  geom_density(fill = "lightblue", alpha = 0.5) +
  geom_vline(aes(xintercept = mean(price)), color = "red", linetype = "dashed", size = 1) +
  labs(x = "Price", y = "Density") +
  ggtitle("Density Estimation of Car Prices") +
  theme_minimal()
```

```
## 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.
```



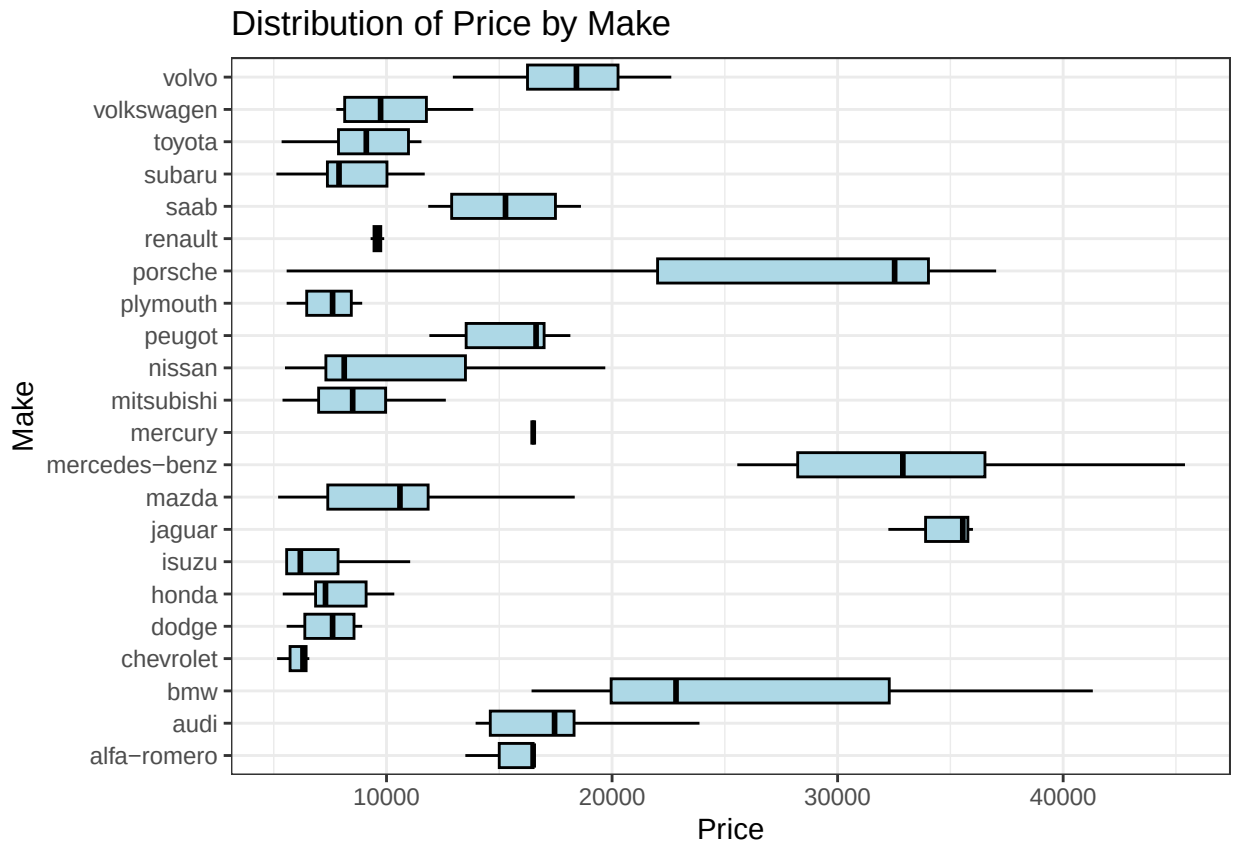
(i) Check (visually) if it is plausible to consider that price follows a normal distribution.

```r
library(ggplot2)
library(dplyr)
library(ggfittext)

# Create a histogram with density curve
ggplot(data, aes(x = price)) +
  geom_histogram(binwidth = 5000, fill = "lightblue", color = "black", alpha = 0.5) +
  stat_function(fun = dnorm, args = list(mean = mean(data$price), sd = sd(data$price)), color = "red", s
  labs(x = "Price", y = "Frequency") +
  ggtitle("Distribution of Car Prices") +
  theme_minimal()
```



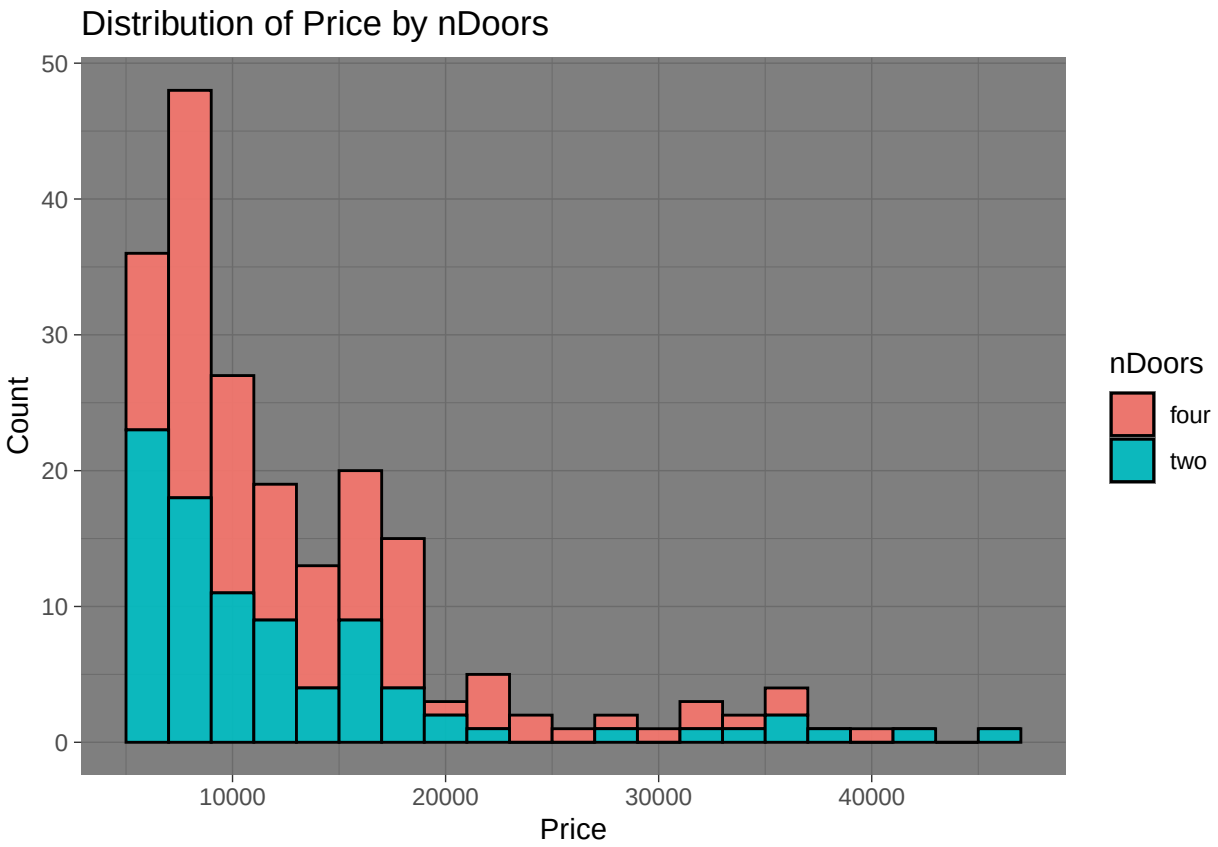
(j) Show the distribution of price by make attribute. Suggestion: use boxplots and the function coord_flip().

```
library(ggplot2)
# Create a boxplot of price by make
ggplot(data, aes(x = make, y = price)) +
  geom_boxplot(fill = "lightblue", color = "black", outlier.shape = NA) +
  coord_flip() +
  labs(x = "Make", y = "Price") +
  ggtitle("Distribution of Price by Make") +
  theme_bw()
```



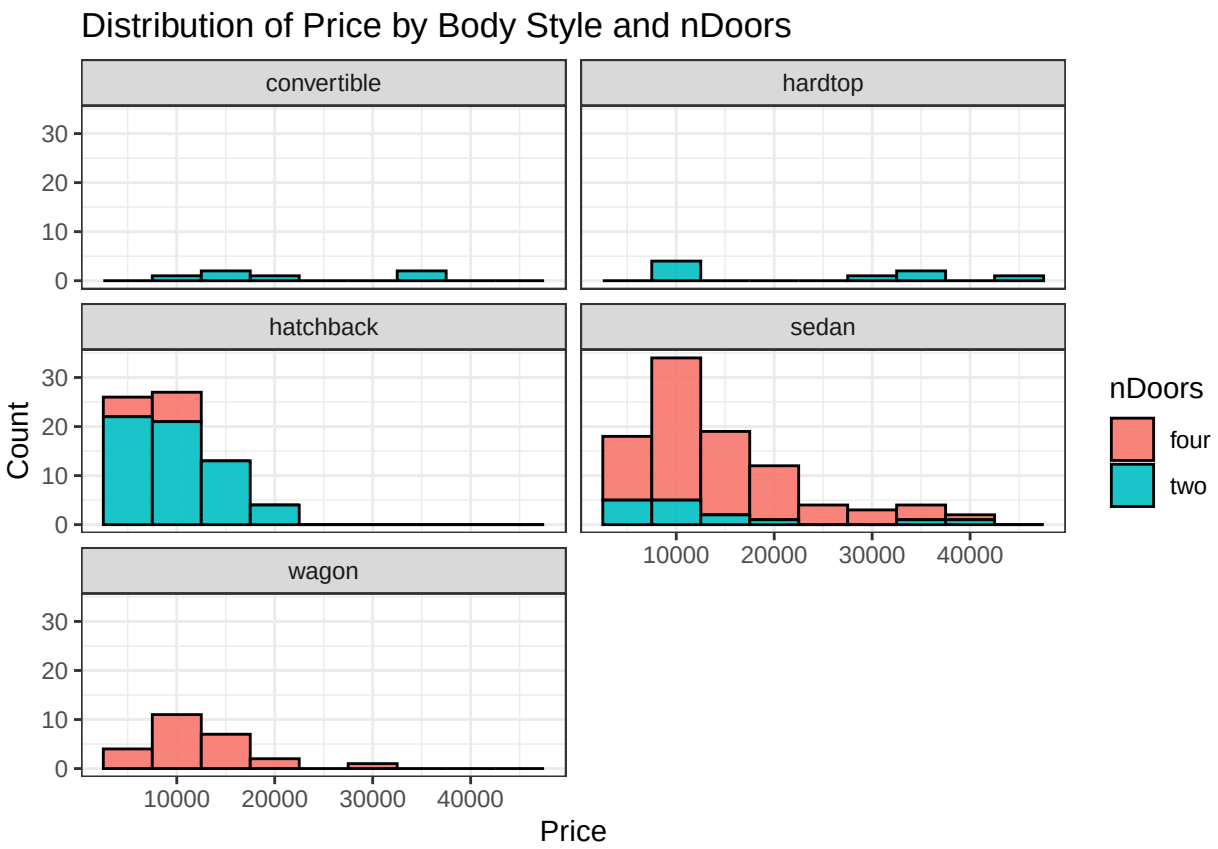
(k) Show the distribution of price by nDoors attribute. Suggestion: use histograms.

```
library(ggplot2)

# Create a histogram of price by nDoors
ggplot(data, aes(x = price, fill = nDoors)) +
  geom_histogram(binwidth = 2000, color = "black", alpha = 0.9) +
  labs(x = "Price", y = "Count") +
  ggtitle("Distribution of Price by nDoors") + theme_dark()
```



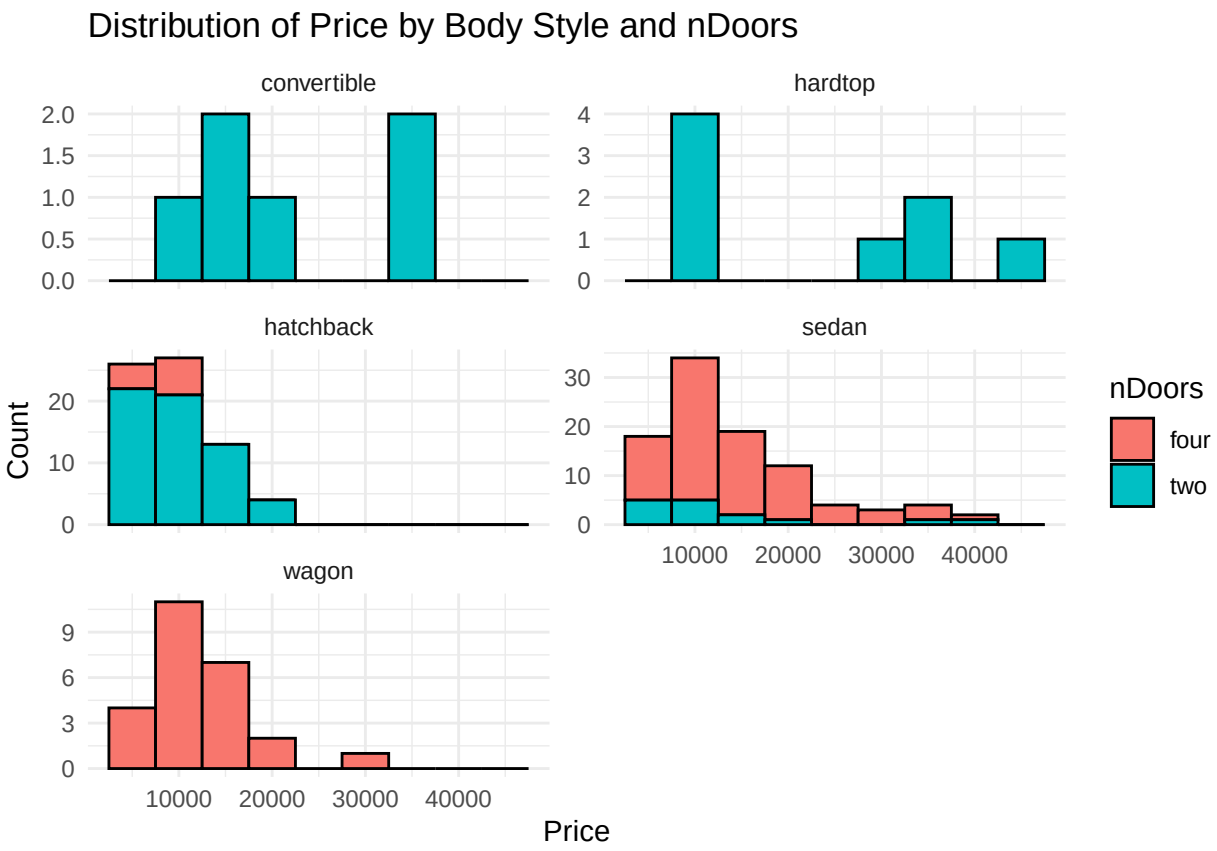
(l) Show the distribution of price by bodyStyle and nDoors attributes. Suggestion: use histograms

```
library(ggplot2)

# Create a histogram of price by bodyStyle and nDoors
ggplot(data, aes(x = price, fill = nDoors)) +
  geom_histogram(binwidth = 5000, color = "black", alpha = 0.9) +
  facet_wrap(~ bodyStyle, ncol = 2) +
  labs(x = "Price", y = "Count") +
  ggtitle("Distribution of Price by Body Style and nDoors") +
  theme_bw()
```



(m) Add the parameter scales="free_y" to the facet function in the previous graph.

```
library(ggplot2)

# Create a histogram of price by bodyStyle and nDoors
ggplot(data, aes(x = price, fill = nDoors)) +
  geom_histogram(binwidth = 5000, color = "black", alpha = 1) +
  facet_wrap(~ bodyStyle, scales = "free_y", ncol = 2) +
  labs(x = "Price", y = "Count") +
  ggtitle("Distribution of Price by Body Style and nDoors") +
  theme_minimal()
```


Distribution of Price by Body Style and nDoors
convertible
hardtop
hatchback
sedan
wagon
Count
Price
nDoors
four
two