R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
install.packages("reshape2")
```

```
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the a
ppropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Edgar/AppData/Local/R/win-library/4.3’
(as ‘lib’ is unspecified)
also installing the dependency ‘plyr’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.3/plyr_1.8.8.zip'
Content type 'application/zip' length 1161886 bytes (1.1 MB)
downloaded 1.1 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.3/reshape2_1.4.4.zip'
Content type 'application/zip' length 454195 bytes (443 KB)
downloaded 443 KB
```

```
package ‘plyr’ successfully unpacked and MD5 sums checked
package ‘reshape2’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Edgar\AppData\Local\Temp\RtmpmwfnzO\downloaded_packages
```

```
library(reshape2)
```

```


Attaching package: ‘reshape2’

The following object is masked from ‘package:tidyr’:

    smiths
```

```
library(gridExtra)
library(ggplot2)
library(tidyverse)
library(dplyr)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

3 DATA EXPLORATION

1 Extracto

cargar los datos del data set carIns_final.RData la cual ya tiene descrtados los valores NA

1 usando el paquete de dplyr, responda las siguientes preguntas

a) Obtener el número de carros agrupados por bodystyle

```
load("C:/Users/Edgar/Documents/GitHub/DataMining_and_MachineLearning_EdgarV/data/HandsOn_Data/carIns_final.Rdat
a")
## ls()

df <- carIns_final

num_carros_por_bodystile <- df %>%
                            group_by(bodyStyle) %>%
                            count()
num_carros_por_bodystile
```

bodyStyle	n
<fctr>	<int>
convertible	6
hardtop	8
hatchback	70
sedan	96
wagon	25

5 rows

```
NA
```

b) Obtener el número de carros agrupados por bodystyle y ademas por fuelType

```
num_carros_por_bodystyle_fueltype <- df %>%
                                     group_by(bodyStyle,fuelType) %>%
                                     count()

## NOS MUESTRA EL NUMERO DE VEHÍCULOS SI ESTE ES A GASOLINA O A DIESEL Y POR CADA DISEÑO
num_carros_por_bodystyle_fueltype
```

bodyStyle	fuelType	n
<fctr>	<fctr>	<int>
convertible	gas	6
hardtop	diesel	1
hardtop	gas	7
hatchback	diesel	1
hatchback	gas	69
sedan	diesel	15
sedan	gas	81
wagon	diesel	3
wagon	gas	22

9 rows

```
NA
```

c) Obtener la media y la desviación estandar del atributo cityMpg agrupado por bodyStyle y en orden ascendente

```
media_st_cityMpg_por_bodystyle <- df %>%
                                  group_by(bodyStyle)%>%
                                  summarise(cityMpg_mean = mean(cityMpg),cityMpg_st = sd(cityMpg))%>%
                                  arrange(cityMpg_mean,cityMpg_st)
media_st_cityMpg_por_bodystyle
```

bodyStyle	cityMpg_mean	cityMpg_st
<fctr>	<dbl>	<dbl>
convertible	20.50000	3.391165
hardtop	21.62500	5.423165
wagon	24.04000	4.217819
sedan	25.32292	6.599035
hatchback	26.31429	7.169870

5 rows

d) Agrupelos por bodyStyle los atributos de cityMpg y highWayMpg, obtener la media, la mediana, desviación estandar y el rango inter-cuartil

```
resumen_cityMpg_highwayMpg <- df %>%
                              group_by(bodyStyle)%>%
                        summarise(cityMpg_mean=mean(cityMpg),highwayMpg_mean=mean(highwayMpg),
cityMpg_median=median(cityMpg),highwayMpg_median=median(highwayMpg),
                                  cityMpg_sd=sd(cityMpg),highwayMpg_sd=sd(highwayMpg),
                                  cityMpg_rec=IQR(cityMpg),highwayMpg_rec=IQR(highwayMpg))
resumen_cityMpg_highwayMpg
```

bodyStyle	cityMpg_mean	highwayMpg_mean	cityMpg_median	highwayMpg_median	cityMpg_sd ▸
<fctr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
convertible	20.50000	26.00000	21	27.0	3.391165
hardtop	21.62500	27.25000	23	27.5	5.423165
hatchback	26.31429	32.17143	26	31.5	7.169870
sedan	25.32292	30.83333	25	30.5	6.599035
wagon	24.04000	28.72000	24	29.0	4.217819

5 rows | 1-6 of 9 columns

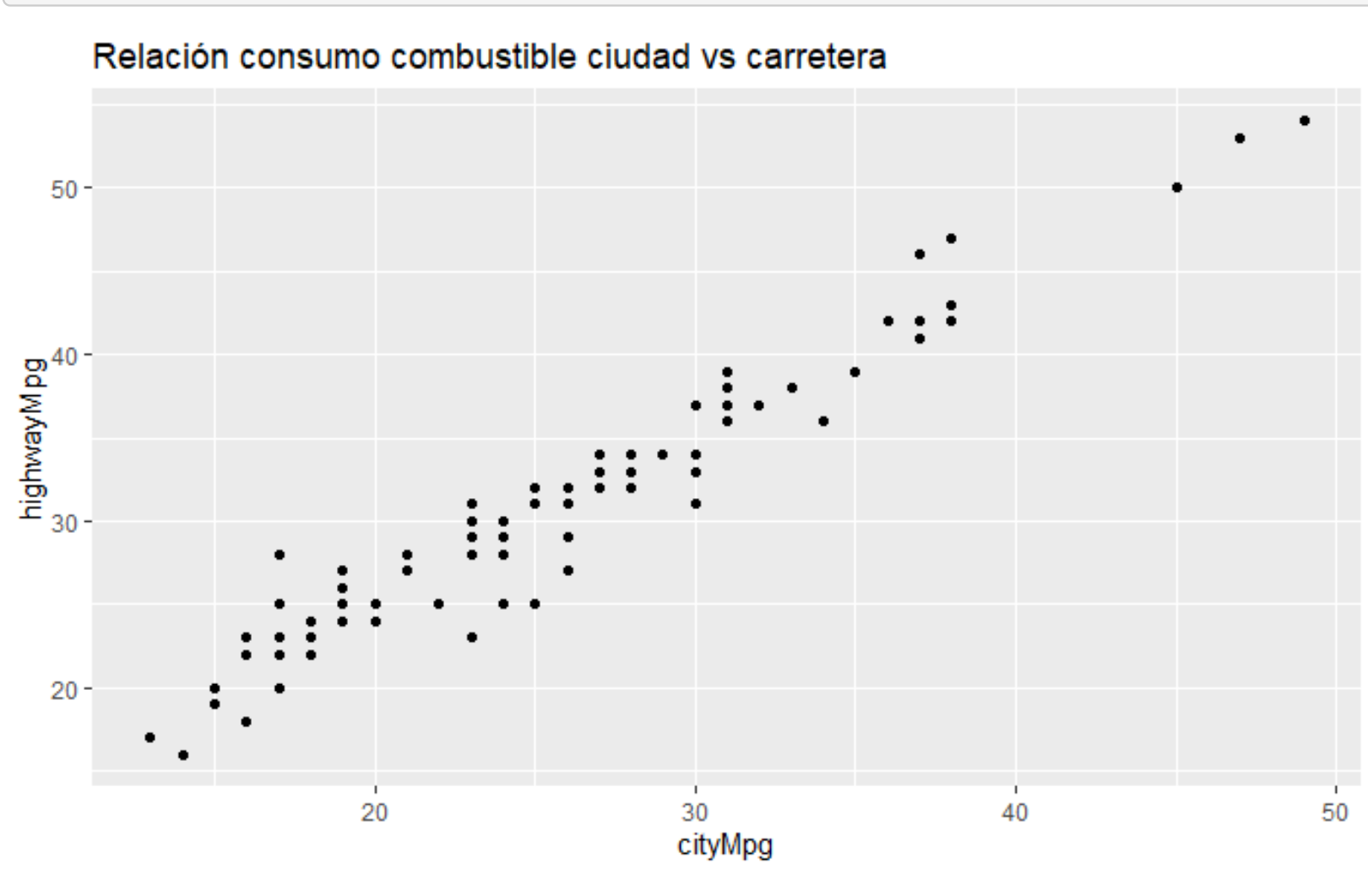
```
NA
```

2. VISUALIZACIÓN

2.- Usando el paquete ggplot2, crear el grafico que usted encuentre adecuado para responder las siguientes preguntas

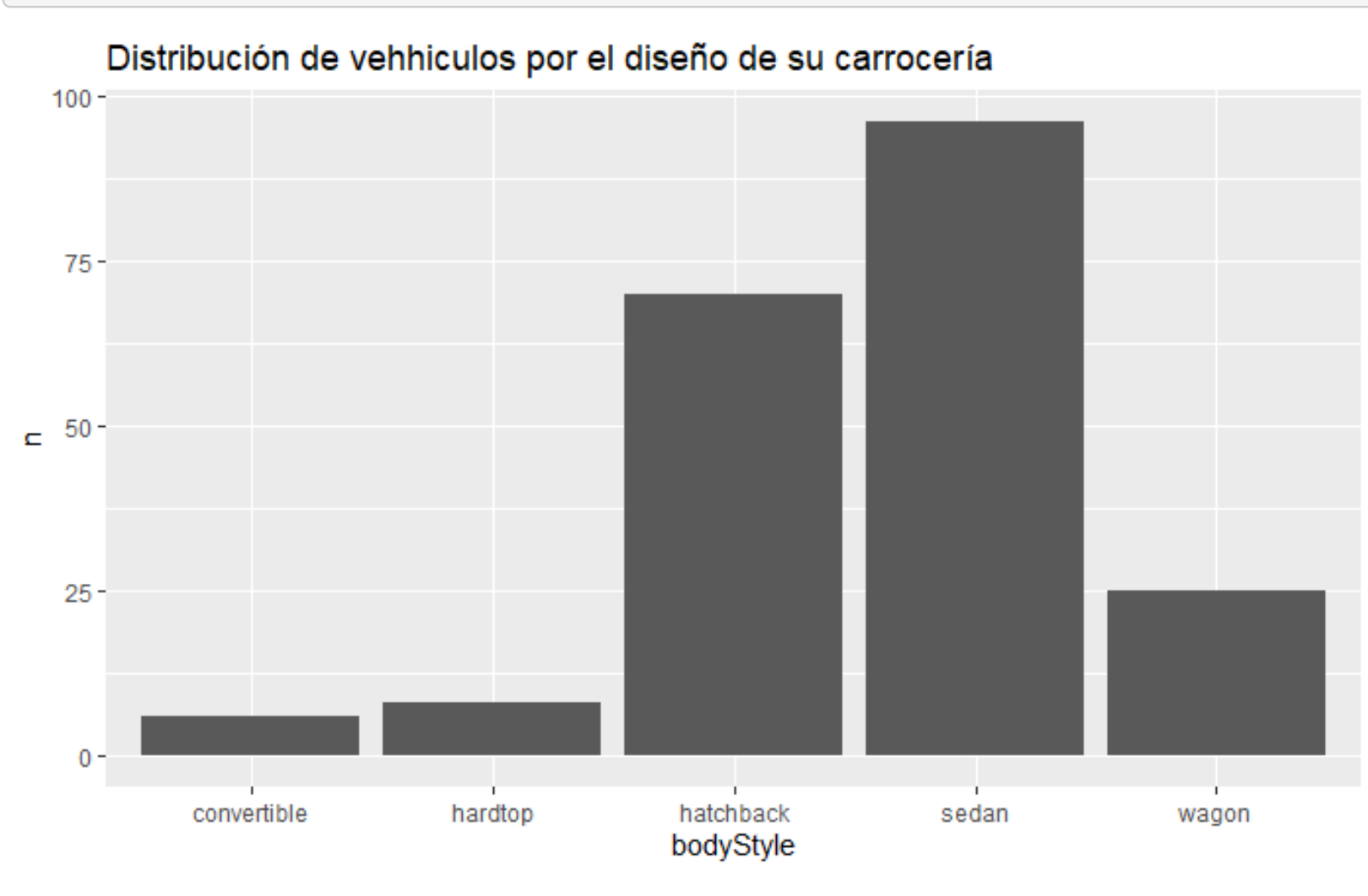
e) Mostrar la relación entre cityMpg y highwayMpg

```
ggplot(df, aes(x = cityMpg, y = highwayMpg)) +
  geom_point() + labs(title="Relación consumo combustible ciudad vs carretera")
```



f) Mostrar la distribución de carros por su diseño

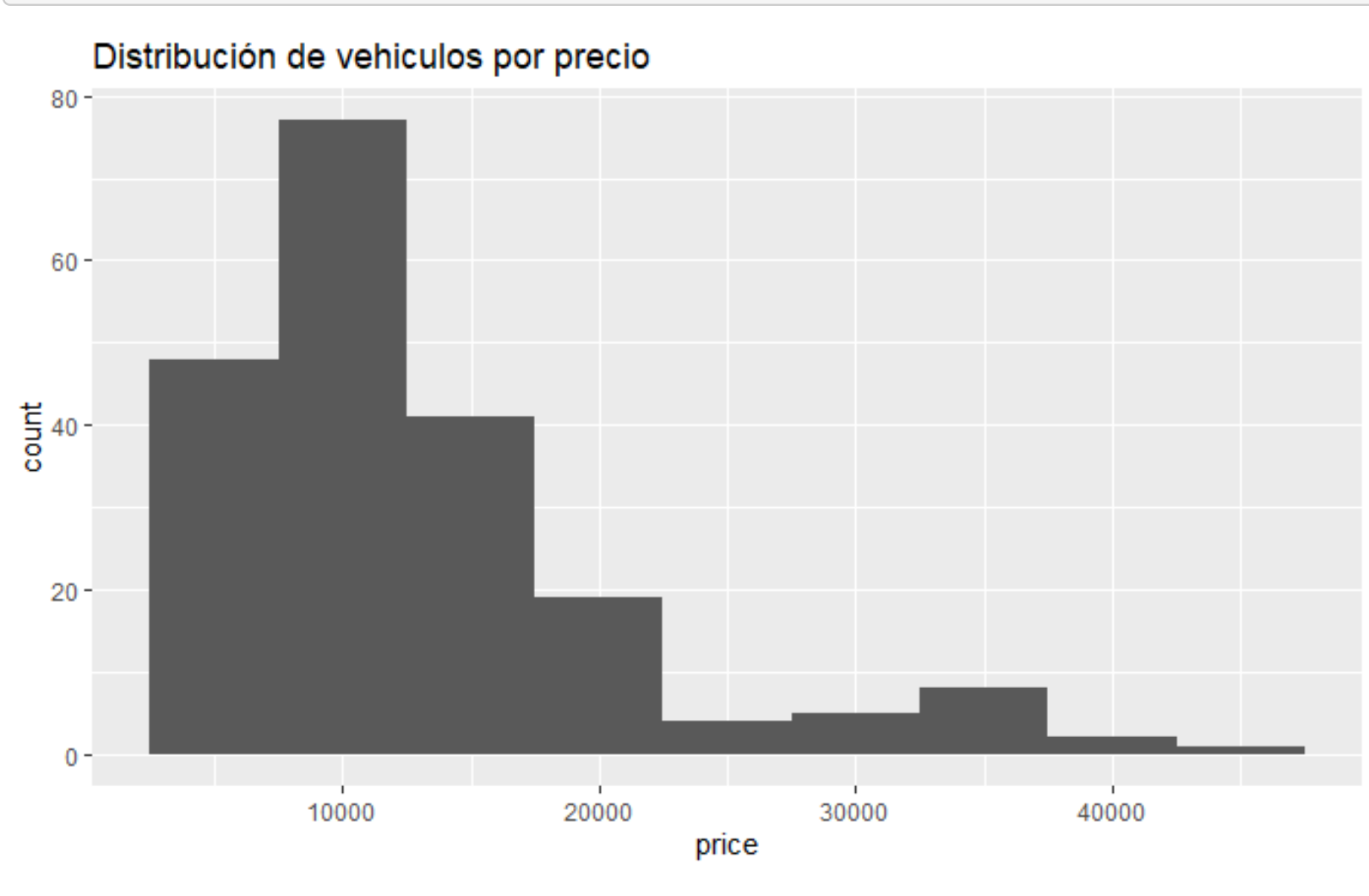
```
ggplot(num_carros_por_bodystile, aes(x = bodyStyle, y = n)) +
   geom_bar(stat = "identity") + labs(title="Distribución de vehhiculos por el diseño de su carrocería")
```



g) Mostrar la distribución de carros por su precio (Sugerencia: establecer el anchi de la barra en 5000)

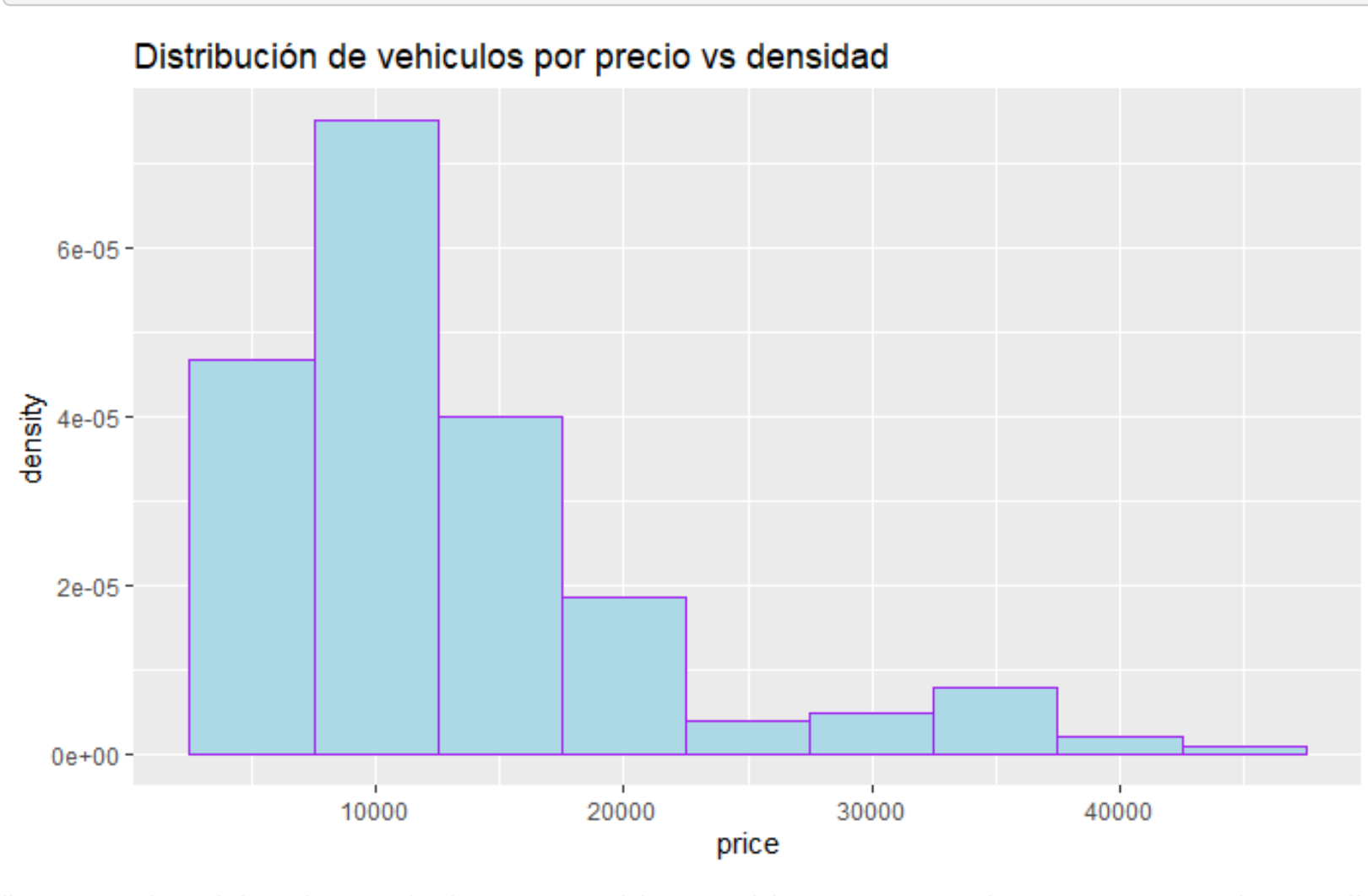
```
carros_por_precio <- df%>%
                    group_by(price)

ggplot(carros_por_precio, aes(x = price)) +
   geom_histogram(binwidth = 5000) + labs(title="Distribución de vehiculos por precio")
```



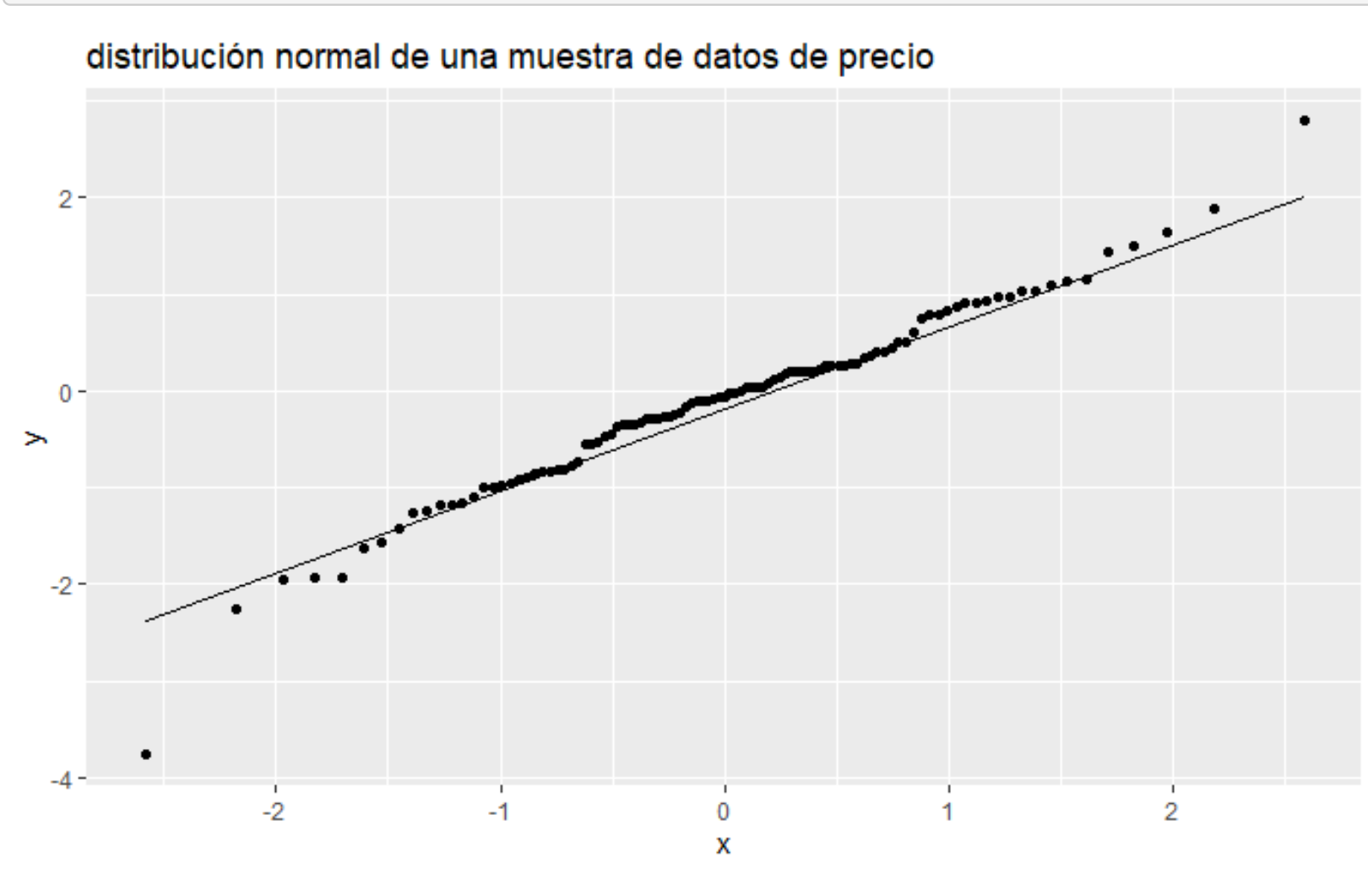
h) agregar la información de la estimación de densidad el grafico anterior

```
ggplot(carros_por_precio, aes(x = price)) +
   geom_histogram(aes(y = after_stat(density)), binwidth = 5000, color = "purple", fill = "lightblue") + labs(tit
le="Distribución de vehiculos por precio vs densidad")
```



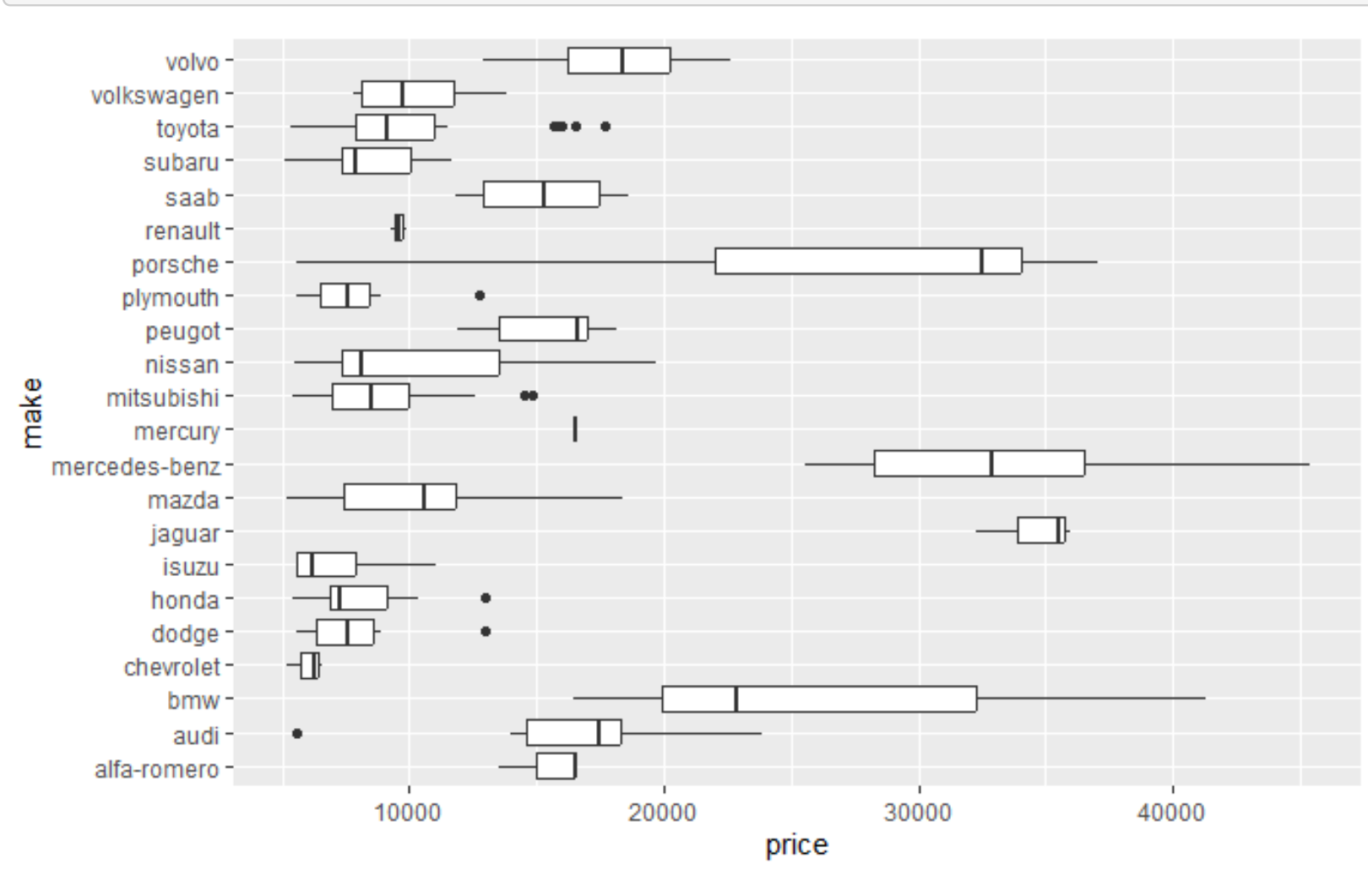
i) comprobar (visualmente) si es aceptable considerar ese precio para conseguir una distribución normal

```
carros_por_precio <- rnorm(103)
ggplot(data.frame(carros_por_precio), aes(sample = carros_por_precio)) +
  geom_qq() + labs(title = "distribución normal de una muestra de datos de precio") +
  geom_qq_line()
```



j) Mostrar la distribucion del precio por el atributo marca (Sugerencia: usar Boxplots y la función coord_flip())

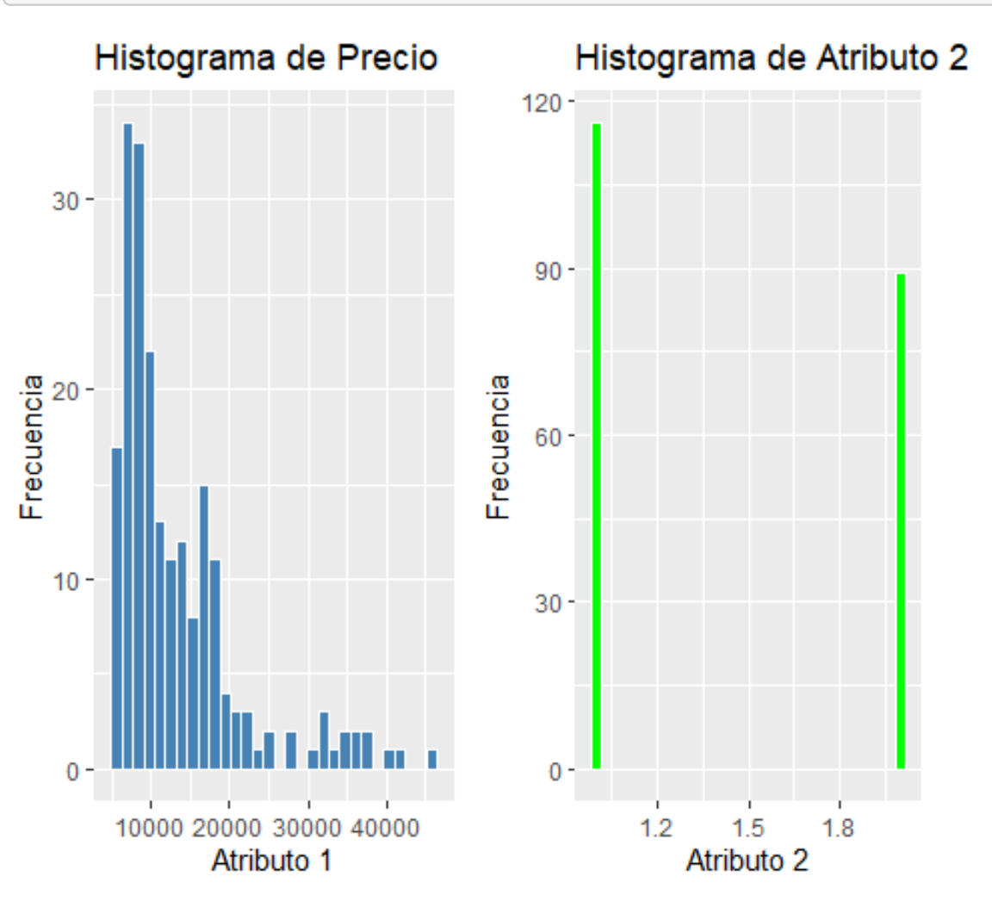
```
carros_por_precio1 <- df%>%
                    group_by(price)
ggplot(carros_por_precio1, aes(x = make, y = price)) +
  geom_boxplot() +
  coord_flip()
```



k) Mostrar la distribución de precios por elatributo nDors(numero de puertas) (Sugerencia: usar histogramas)

```
histograma1 <- ggplot(new_carro_por_precio, aes(x = price)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 30) +
  labs(x = "Atributo 1", y = "Frecuencia") +
  ggtitle("Histograma de Precio")

histograma2 <- ggplot(new_carro_por_precio, aes(x = nDoors)) +
  geom_histogram(fill = "green", color = "white", bins = 30) +
  labs(x = "Atributo 2", y = "Frecuencia") +
  ggtitle("Histograma de Atributo 2")
grid.arrange(histograma1, histograma2, ncol = 3)
```



N #### l) Mostrar la distribución del precio por bodyStyle y atributos nDoors, (Sugerencia: usar Histogramas)

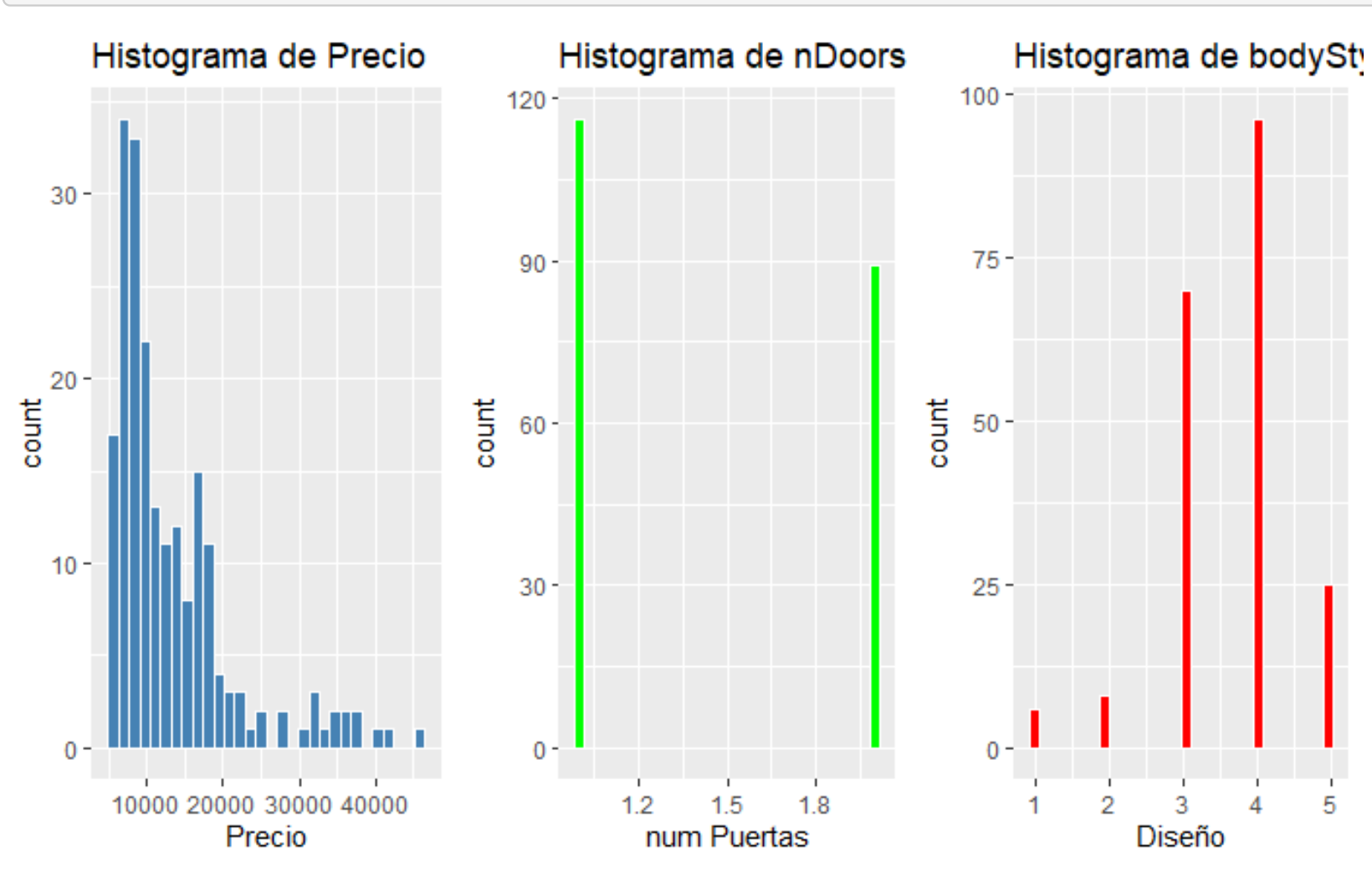
```

histograma1 <- ggplot(new_carro_por_precio, aes(x = price)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 30) +
  labs(x = "Precio", y = "count") +
  ggtitle("Histograma de Precio")

histograma2 <- ggplot(new_carro_por_precio, aes(x = nDoors)) +
  geom_histogram(fill = "green", color = "white", bins = 30) +
  labs(x = "num Puertas", y = "count") +
  ggtitle("Histograma de nDoors")

histograma3 <- ggplot(new_carro_por_precio, aes(x = bodyStyle)) +
  geom_histogram(fill = "red", color = "white", bins = 30) +
  labs(x = "Diseño", y = "count") +
  ggtitle("Histograma de bodyStyle")

grid.arrange(histograma1, histograma2, histograma3, ncol = 3)
```



m) agregar el parámetro free_y a la función de la faceta en el gráfico anterior

```
carros <- new_carro_por_precio %>%
          group_by(bodyStyle,nDoors)%>%
          select(price)
```

```
Adding missing grouping variables: `bodyStyle`, `nDoors`
```

```
carros
```

bodyStyle	nDoors	price
<int>	<int>	<int>
1	2	13495
1	2	16500
3	2	16500
4	1	13950
4	1	17450
4	2	15250
4	1	17710
5	1	18920
4	1	23875
3	2	5572

1-10 of 205 rows Previous 1 2 3 4 5 6 ... 21 Next

```
carros_melt <- reshape2::melt(carros)
```

```
No id variables; using all as measure variables
```

```
  histograma_final <- ggplot(carros_melt, aes(x = value)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 30) +
  facet_wrap(~ variable, scales = "free_y") +
  labs(x = "Valor", y = "Frecuencia") +
  ggtitle("Histograma de los Atributos")

  histograma_final
```