# Team Project - Part 1

Miguel A. Guijarro Martínez A01612042

José Rafael Delgado Dib A01611951

Gerardo Ramos Méndez A01611772

September 19th, 2022

Management and Process Improvement

Elvia Guadalupe Castro Félix

# Index

# 1. General Analysis

## Problem Identification

The question to be answered is ¿What is the right price for a used car? During 2022, car prices had increased by 15%. The automotive sector has faced a complex global landscape since the Covid-19 pandemic began. First, due to the shortage in the production lines during 2020; then, due to the lack of semiconductors that have been reduced as the manufacture of electronic equipment, such as cell phones and tablets, has increased. Finally, the increase in the cost of raw materials and sea freight has raised prices considerably.

These scenarios led to the production of 9,580,911 vehicles globally throughout 2021, according to the international consulting firm IHS Markit. This reduction in supply increased the price of available units and has impacted the availability to acquire new cars at a reasonable price. (Tzuaro, 2022).

Due to this situation, a quantitative and qualitative approach will be used to respond to our initial question. To achieve this objective, different tools and methods will be used such as web scraping, data analytics, and statistical models to bring this vision into reality.

## Organization

It was decided to analyze the data available on olxautos.com.mx. OLX Autos is the leading platform in Latin America for buying and selling used cars. This company already operated in Mexico under the brand VendeTuAuto and Autobastas. OLX Autos is present in 13 countries, including the United States, Colombia, Argentina, Chile, Ecuador, and Peru, and also leads the Latin American market. OLX has integrated a team of more than 280 collaborators, distributed in 60 service points in Mexico, and has become one of the primary car vendors in the online market. This webpage was perfect to scrap the data and analyze it in an easy manner.

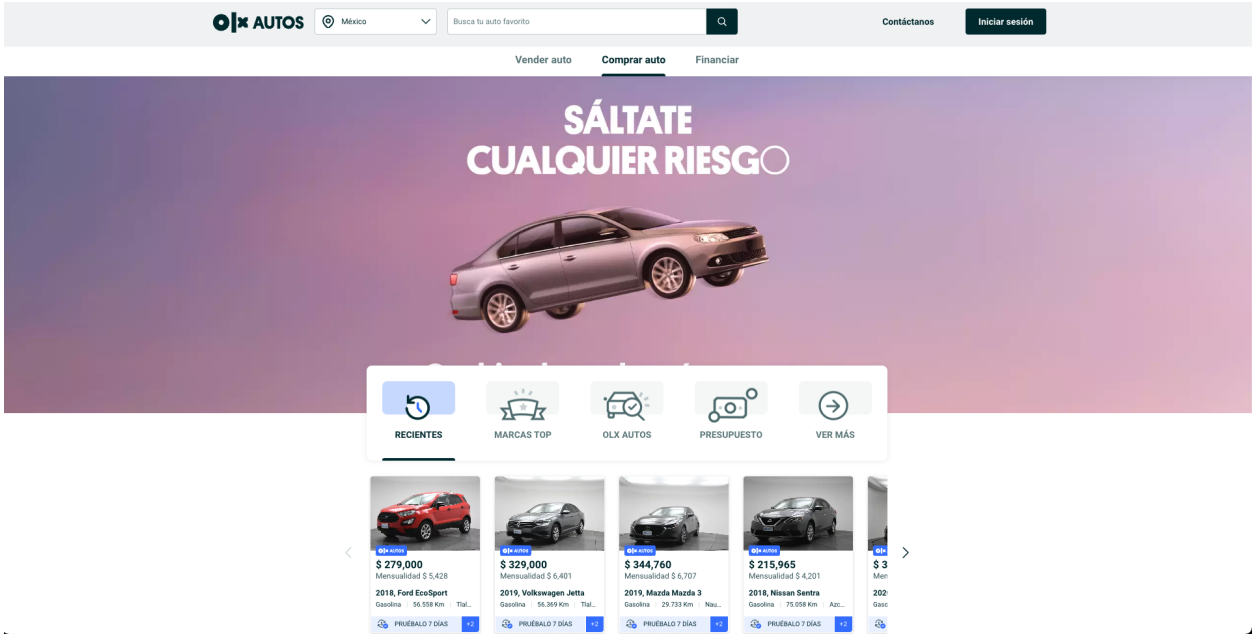


Figure 1.1: Homepage of OLX Autos

# Get Data

In order to obtain the data, Python was used as the main programming language. More specifically, we decided to use the selenium library. This decision was made after carefully reviewing other options. We decided to work with this library because of its robustness and the fact that it allows us to scrap the data available on the webpage while loading its dynamic content.

First, the user-agent parameters related to the web driver options were configured. This was made in order to avoid a *server rejection* status due to a crawler detection that some websites have. After doing this, the chrome web driver extension was installed and the first request to the olxautos website was made. (figure 1.2)

```
#Opciones del navegador
opts = Options()
opts.add_argument(
    "user-agent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)"
    " Ubuntu Chromium/71.0.3578.80 Chrome/71.0.3578.80 Safari/537.36")
driver = webdriver.Chrome(ChromeDriverManager().install(), chrome_options=opts)
driver.get("https://www.olxautos.com.mx/autos_c84")
```

Figure 1.2: The web driver options were defined and the Webdriver was installed

Secondly, a for loop with a determined number of iterations (in our case we decided to use 10 iterations in order to obtain a significant amount of data ) was programmed to click the "Cargar más" button by using Xpath in order to find it. Within this loop, a timer that waits up to 40 seconds for the load of the button within the page was added in order to allow the page to load completely and to prevent some errors related to selenium trying to click on the button before it appears. Furthermore, after the click of the button, another timer was created in order to load all posts within the new section. (Figure 1.3). The whole code snippet handles errors gracefully, due to the fact that the code inside the loop's body is contained within a try-except block in order to capture any errors.

```
#clickear botones un numero especifico de veces
for i in range(10):
    try:
        boton = WebDriverWait(driver, 40).until(
            EC.presence_of_element_located((By.XPATH, '//button[@data-aut-id= "btnLoadMore"]'))
        )
        boton.click()
        WebDriverWait(driver, 20).until(
            EC.presence_of_all_elements_located((By.XPATH, '//li[@data-aut-id="itemBox"]//span[@class="_1OBW-"]'))
        )
    except:
        break
```

Figure 1.3: For loop including the click

After the end of the for loop, all the links of each car post were obtained and added to a list called links_paginas. Additionally, the header of the CSV file was created and added to a list called lista_de_autos (figure 1.4).

```
links_autos = driver.find_elements(By.XPATH, '//li[@data-aut-id="itemBox"]//a')
links_paginas = []
#colocar todos los links en una lista
for href in links_autos:
    links_paginas.append(href.get_attribute('href'))

headercsv = ["marca", "modelo", "año", "precio", "mensualidad", "meses", "enganche", "co
             "kilometraje", "transmision", "numero de puertas", "ubicacion"]
lista_de_autos = [headercsv]
fallos = 0
```

Figure 1.4: For loop including the click

Subsequently, a for loop was added in order to access each link within the list links_paginas, inside this for loop the driver accessed the link and waits between 3 and 9 seconds with a uniform distribution in order to emulate human action, after this interaction, the web driver checks whether the title has loaded and if not it waits for a maximum of 60 seconds and if no the web driver does not succeed and an exception is triggered. Once the title loads all the data is scrapped and cleansed in order to avoid future errors while processing the file. After this part, the data extracted is stored on the lista_de_autos as a sublist in order to process it for the CSV

conversion. As soon as the data is stored on the list the web driver returns to the previous page in order to emulate a human interaction and another iteration occurs. (Figure 1.5)

It is important to remark that the for loop contains a while loop inside that runs all the time, this is due to the fact that sometimes the website detects too many requests and an error page is shown, this problem is solved by just reloading the page, and that is why the while loop only gets interrupted when either no errors occurred within the same for loop iteration or the number of failures within the same link exceeds the amount of three. Finally, once all the data is added to the list, a CSV file is created adding all the elements within the list (Figure 1.6)

```
for link in links_paginas:
    while True:
        try:
            #Entrar al link
            driver.get(link)
            #Esperar de acuerdo a una distribucion uniforme entre 3 y 9 segundos
            time.sleep(random.uniform(3,9))
            WebDriverWait(driver, 60).until(
                EC.presence_of_all_elements_located((By.XPATH, '//div[@data-aut-id="itemTitle"]'))
            )
            #Obtencion y limpieza de datos
            modelo_año = driver.find_element(By.XPATH, '//div[@data-aut-id="itemTitle"]').text
            marca, modelo, año = modelo_año.split(" ", 2)
            año = año.replace("(", "").replace(")", "")
            print(marca)
            print(modelo)
            print(año)
            precio = driver.find_element(By.XPATH, '//div[@data-aut-id="itemPrice"]').text
            precio = precio.replace("$ ", "").replace(",", "")
            print(precio)
            mensualidad = driver.find_element(By.XPATH, "//div[@data-aut-id='itemEmi']").text
            mensualidad, meses = mensualidad.replace("Mensualidad $ ", "").replace(",", "").split(" X ")
            print(mensualidad)
            print(meses)
            enganche = driver.find_element(By.XPATH, "//div[@data-aut-id='itemHitchAmount']//span[@class='_21VvJ']").text
            enganche = enganche.replace("$ ", "").replace(",", "")
            print(enganche)
            combkilotrans = driver.find_element(By.XPATH, "//div[@class='aOxkz']").text
            combustible, kilometraje, transmision = combkilotrans.split("\n")
            kilometraje= kilometraje.replace(" KM", "")
            print(combustible)
            print(kilometraje)
            print(transmision)
            componentes = driver.find_element(By.XPATH, "//div[@class='_3tLee']").text
            puertas = componentes.find(" pts")
            npuertas = componentes[puertas - 1:puertas]
            print(npuertas)
            WebDriverWait(driver, 10).until(
                EC.presence_of_all_elements_located((By.XPATH, '//div[@class="_1idEV"]//div[@class="_1gasz"]'))
            )
            varios = driver.find_elements(By.XPATH, '//div[@class="_1idEV"]//div[@class="_1gasz"]')
            ubicacion = varios[1].text
            print(ubicacion)
```

Figure 1.4: Scraping the data

```
            # Agregar datos obtenidos a lista
            lista_de_autos.append(
                [marca, modelo, año, precio, mensualidad, meses, enganche, combustible, kilometraje, transmision,
                 npuertas,
                 ubicacion])
            driver.back()

        except:
            fallos += 1
            if fallos > 3:
                fallos = 0
                break
            print("fallo")
            continue
        break

    # escribir datos de la lista en csv
with open("/Users/jose/PycharmProjects/selenium/autos_seminuevos.csv", 'w', encoding="UTF-8") as f:
    writer = csv.writer(f)
    writer.writerows(lista_de_autos)
print(fallos)
```

Figure 1.5: Exception handling and file writing

# Quantitative and Qualitative variables

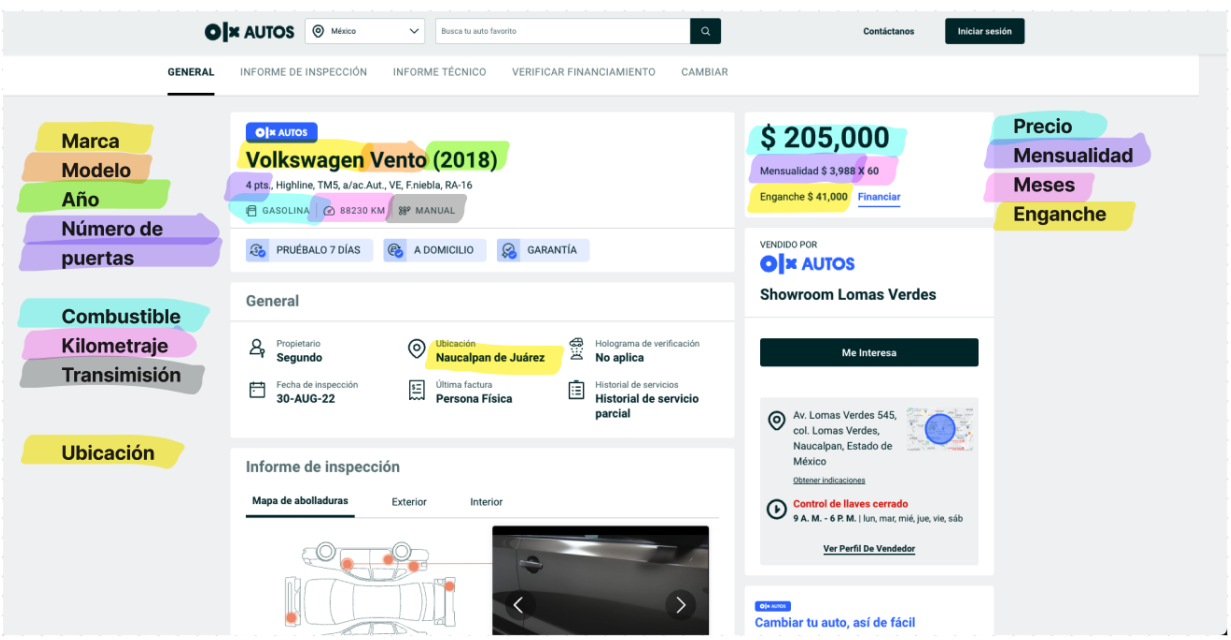


Figure 1..6 - Scoping of data available on the webpage.

The data obtained from the scrapping stage contains 12 variables that represent the state of each car. The selected variables are these ones:

- Marca
- Modelo
- Año
- Precio
- Mensualidad
- Meses
- Enganche
- Combustible
- Kilometraje
- Transmisión
- Número de puertas
- Ubicación

The variables can be divided into the next categories

| **Quantitative** | **Qualitative** |
|---|---|
| • Número de puertas<br>• Kilometraje<br>• Precio<br>• Mensualidad<br>• Meses<br>• Enganche<br>• Año | • Marca<br>• Modelo<br>• Combustible<br>• Transmisión<br>• Ubicación |

Table 1.7 - Distribution of the Quantitative and Qualitative variables

# Data Exploration

In order to start with the data exploration, it was decided to use R since it is a simple statistical programming language that allows graphs and statistics models. Firstly a cleaning of the data was made by converting each data point to its respective type (as factors for the qualitative and as numeric for the quantitative) after doing this, the columns mensualidad and año were affected, and thus there were some NAs on these columns (1 Na for the mensualidad column and 27 NAs for the year). It was decided to omit the rows with NAs in order to make the exploration easier. However, while doing this around 12% of the rows were eliminated. (Figure n.n)

```r
#Miguel Angel Guijarro
#Gerardo Ramos
#Jose Rafael Delgado Dib
#install.packages("ltm")
library(ltm)
library(tidyverse)
library(car)
data <- read.csv("autos_seminuevos.csv")
seed(20)
data$mensualidad <-as.numeric(data$mensualidad)

summary(data)
data$marca <- as.factor(data$marca)
data$modelo <- as.factor(data$modelo)
summary(data)
data$transmision <- as.factor(data$transmision)
summary(data)
data$combustible <- as.factor(data$combustible)
data$año<-as.numeric(data$año)
summary(data)
data$ubicacion <- as.factor(data$ubicacion)
summary(data)
data <- na.omit(data)
summary(data)
```

Figure 1.8 - Main code

```
       marca        modelo          año           precio        mensualidad        meses        enganche
 Chevrolet :38   Trax    : 10   Min.   :2016   Min.   :145000   Min.   : 2821   Min.   :60   Min.   : 29000
 Nissan    :26   Versa   : 10   1st Qu.:2017   1st Qu.:249500   1st Qu.: 4854   1st Qu.:60   1st Qu.: 49900
 Volkswagen:22   Beat    :  8   Median :2018   Median :318000   Median : 6187   Median :60   Median : 63600
 Honda     :21   Forte   :  7   Mean   :2018   Mean   :328677   Mean   : 6394   Mean   :60   Mean   : 65735
 Kia       :20   X-Trail:  7   3rd Qu.:2019   3rd Qu.:394500   3rd Qu.: 7675   3rd Qu.:60   3rd Qu.: 78900
 Seat      :14   A3      :  6   Max.   :2022   Max.   :799800   Max.   :15560   Max.   :60   Max.   :159960
 (Other)   :72   (Other):165
                        combustible   kilometraje            transmision  numero.de.puertas
 DIÉSEL                       :  2   Min.   : 1397   AUTOMÁTICO    :155   Min.   :1.000
 DIÉSEL / ELÉCTRICO           :  1   1st Qu.:27488   MANUAL        : 56   1st Qu.:4.000
 GASOLINA                     :209   Median :48625   SEMIAUTOMÁTICO:  1   Median :5.000
 GASOLINA / ELÉCTRICA (HIBRIDO):  1   Mean   :48730   TRIPTONIC     :  1   Mean   :4.516
                                      3rd Qu.:66795                        3rd Qu.:5.000
                                      Max.   :99168                        Max.   :5.000

              ubicacion
 Naucalpan de Juárez:45
 Tlalnepantla de Baz:31
 Zapopan            :25
 Monterrey          :24
 Azcapotzalco       :23
 Coyoacán           :23
 (Other)            :42
```

Figure 1.9 - Summary of the data

After the cleansing, a statistical summary was obtained by using the summary function in which more detailed information can be seen regarding each of the data points, such as the mean the median and the quartiles in the case of the numeric variables, and all the different factors (levels) for the qualitative data points.

A correlation matrix was also made taking into account all the numerical values.It can be seen that the variables precio, enganche and mensualidad are correlated, which means that we face a multicollinearity problem, and thus both the mensualidad and the enganche need to be taken out from the dataset. On the other hand, also the variables meses needs to be taken out because all of it has a zero mean deviation i.e all the values are the same.

| | precio | año | numero.de.puertas | meses | enganche | mensualidad |
|---|---|---|---|---|---|---|
| precio | 1.0000000 | 0.26425565 | 0.13282692 | NA | 1.0000000 | 1.0000000 |
| año | 0.2642557 | 1.00000000 | -0.09707275 | NA | 0.2642557 | 0.2642501 |
| numero.de.puertas | 0.1328269 | -0.09707275 | 1.00000000 | NA | 0.1328269 | 0.1328146 |
| meses | NA | NA | NA | 1 | NA | NA |
| enganche | 1.0000000 | 0.26425565 | 0.13282692 | NA | 1.0000000 | 1.0000000 |
| mensualidad | 1.0000000 | 0.26425013 | 0.13281463 | NA | 1.0000000 | 1.0000000 |

Additionally, it was decided to plot almost every data point vs the prince in order to see not only patterns and distributions on the data but also the dispersion between numerical values. Finally, it was decided to use the ggplot library in order to compare 3 different variables (2 numericals: *precio & kilometraje* and 1 categorical: *transmision*). This has allowed us to understand the relationship between those variables and make preliminary deductions about the data. For example, cars with automatic transmissions tend to have a higher price than manual ones. These graphs can be found in the annex section.

# Conclusions

The development of the first part of the project involved different techniques that included not only the scrap of the data but also an exploratory analysis which allowed us to understand the data and deepen into their relationships and structure. For the second part of the project, it is expected that this exploratory analysis will allow us to build some models to explain the data and also obtain more updated data by improving the web scraping model.

# Annex

## Code:

The github repo of the used code can be found on the following link:

https://github.com/joserdd2205/data_analytics_project

## Graphs:

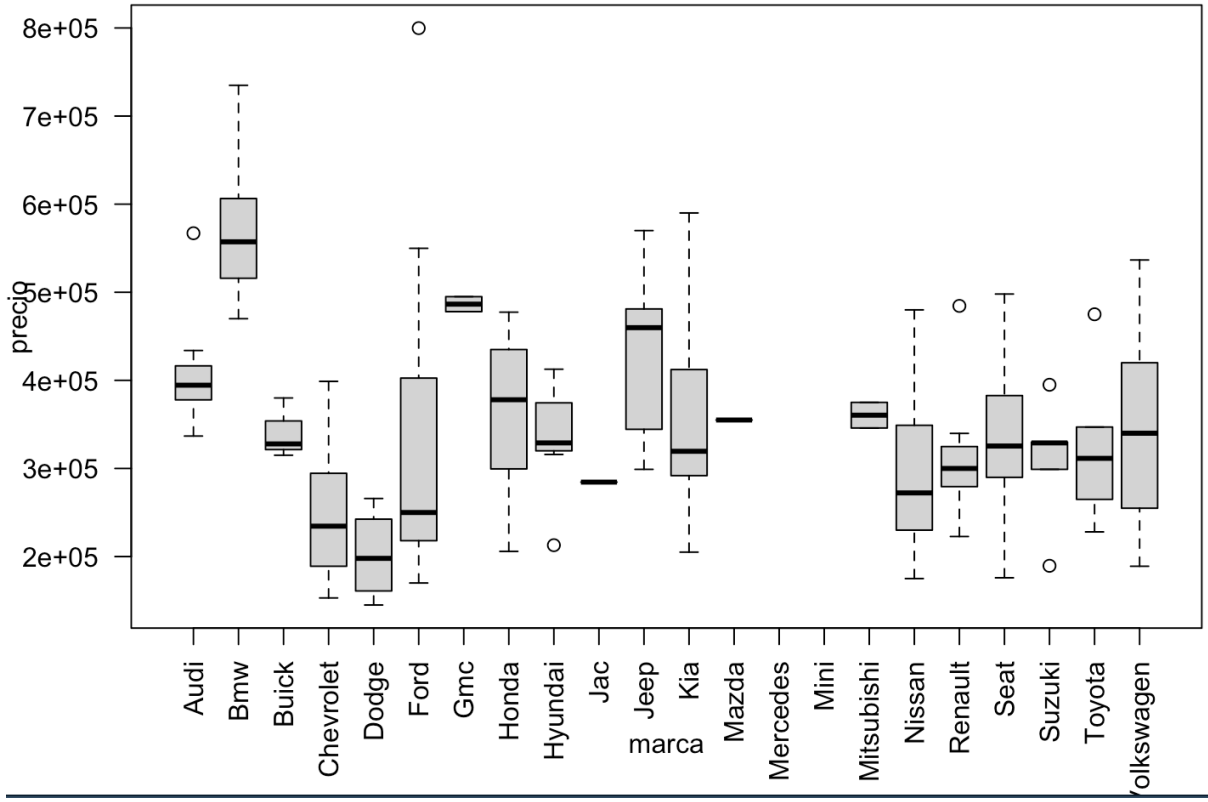


Figure 2.1

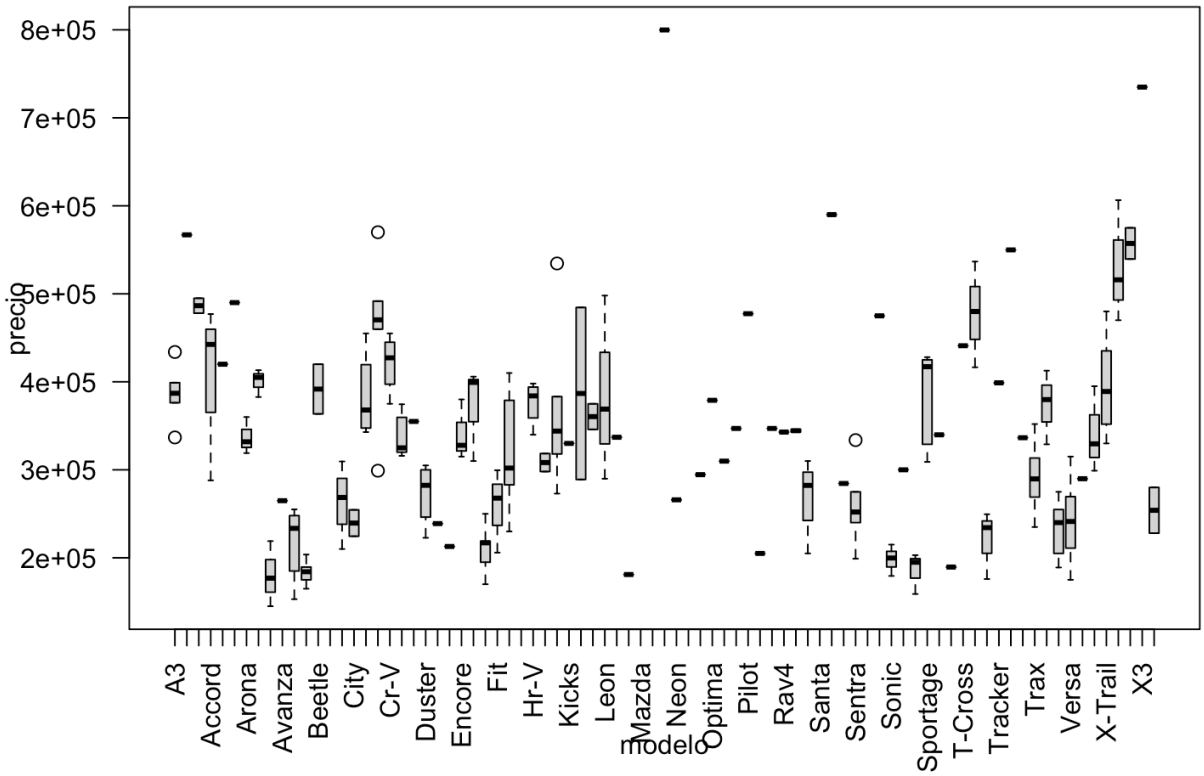


Figure 2.2: car model vs price

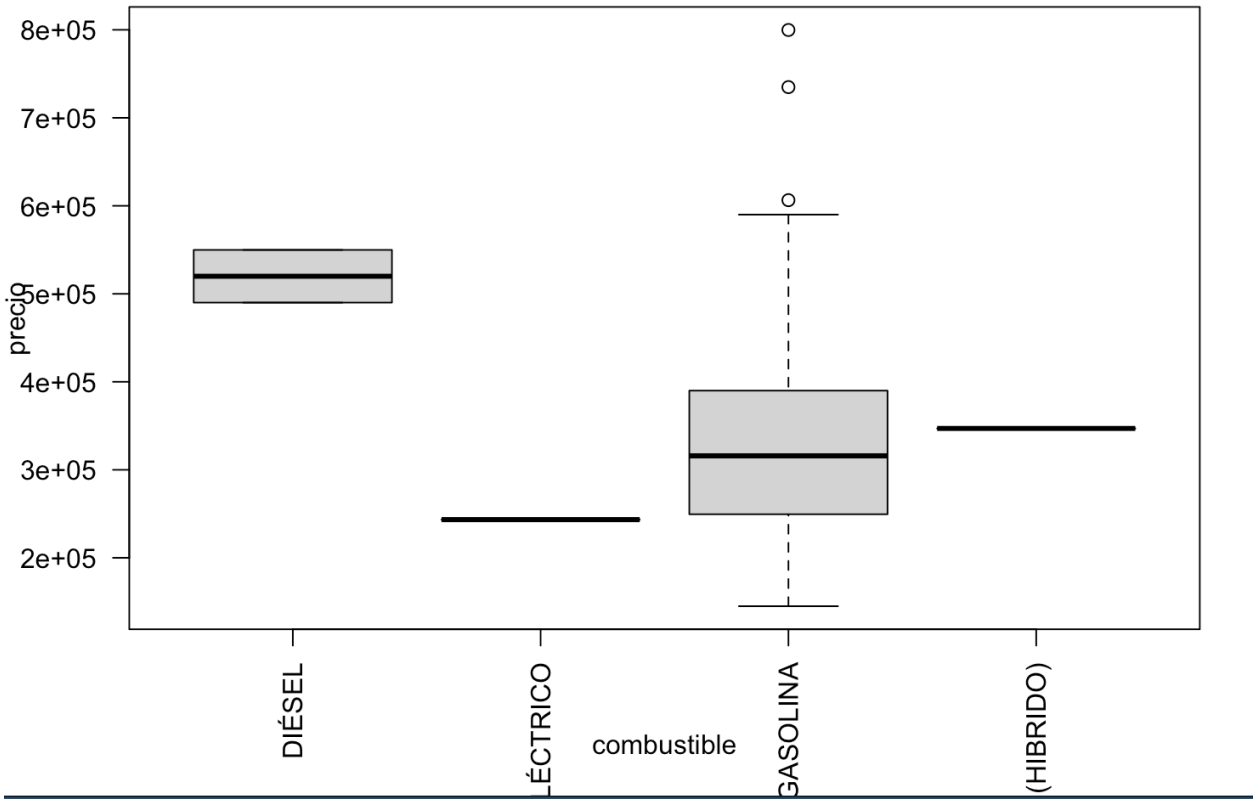


Figure 2.3: Fuel type vs price

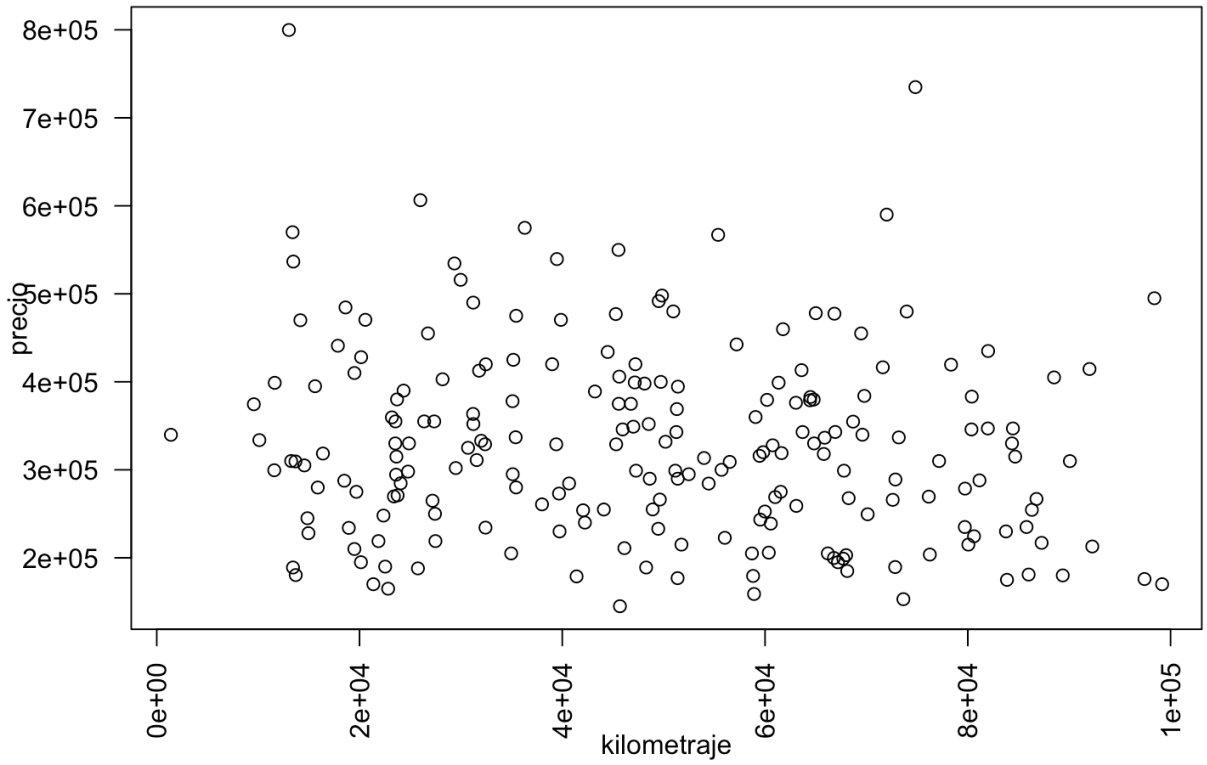


Figure 2.4: mileage vs price

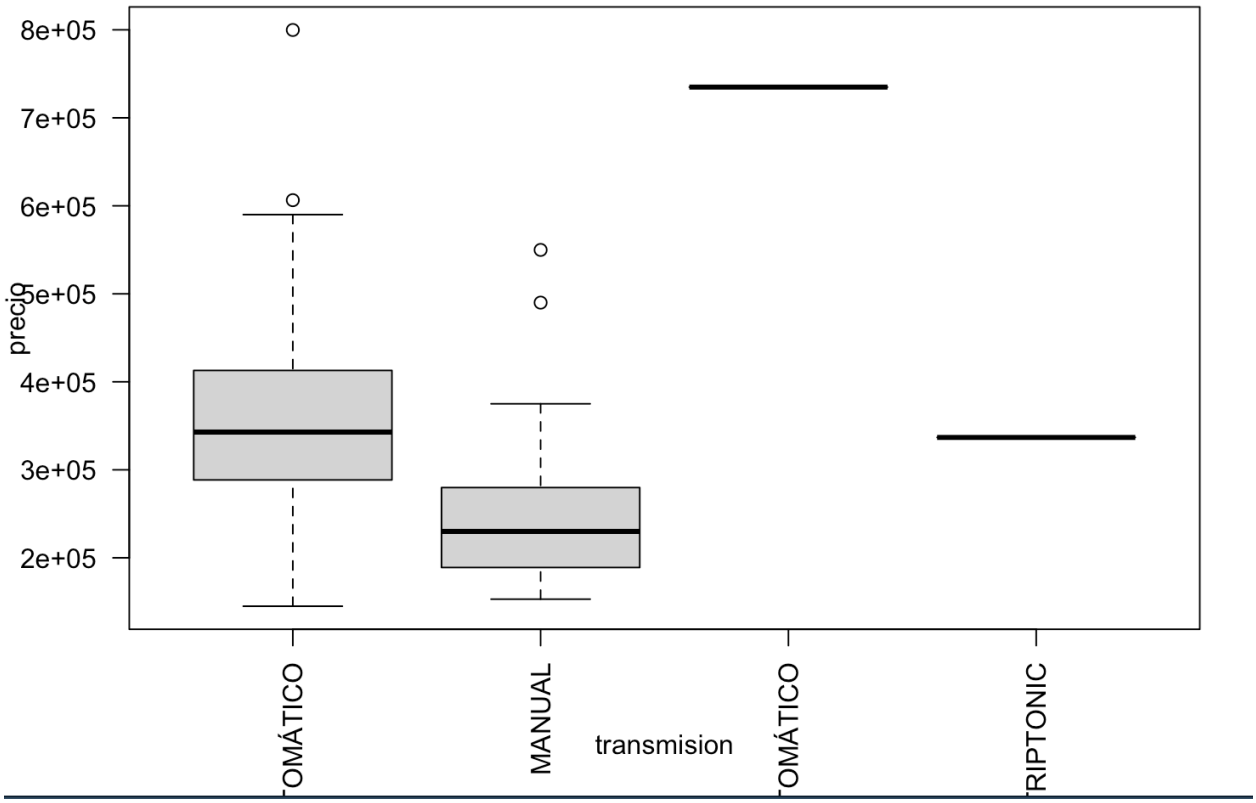


Figure2.5: Transmission type vs price

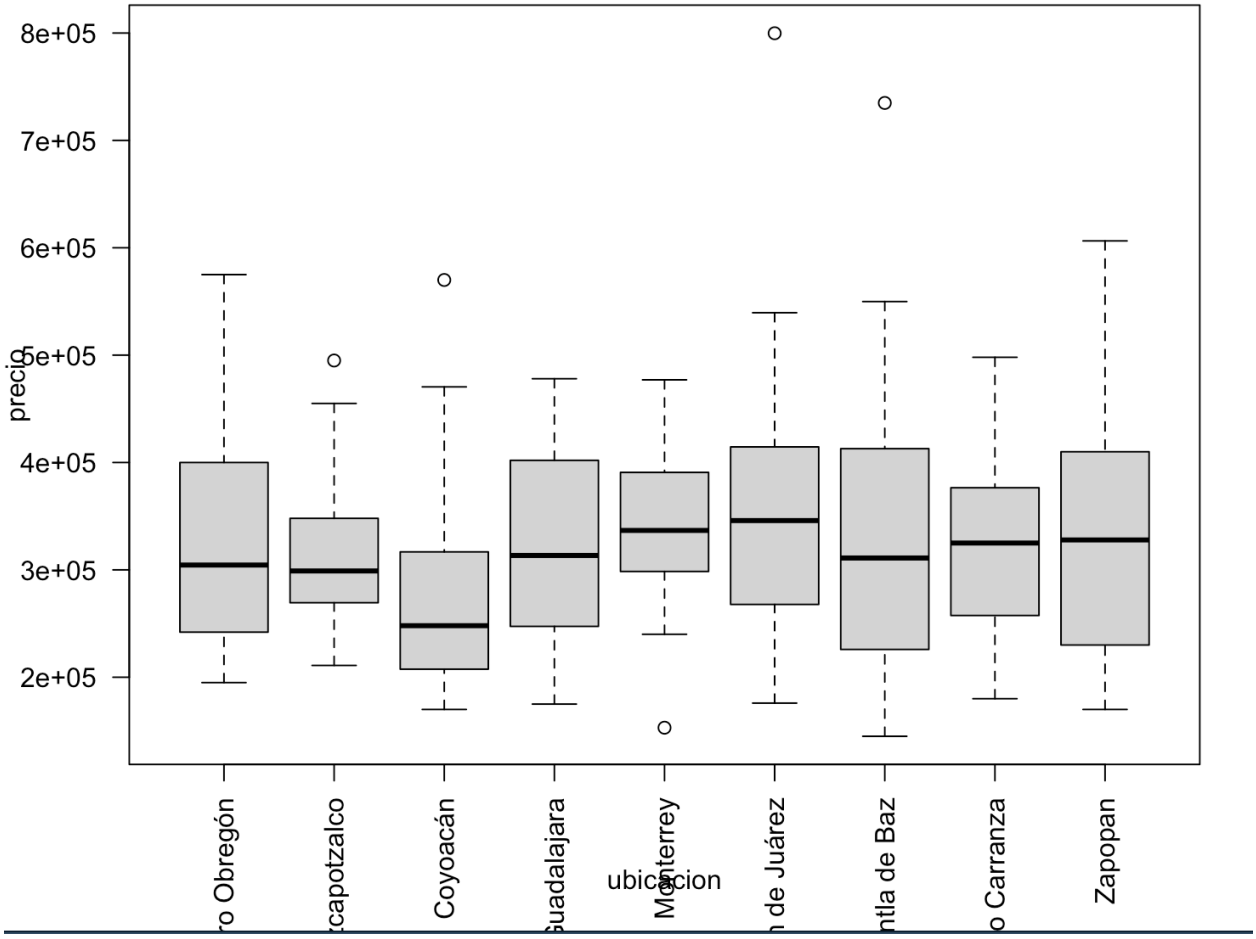


Figure 2.5: Location vs price

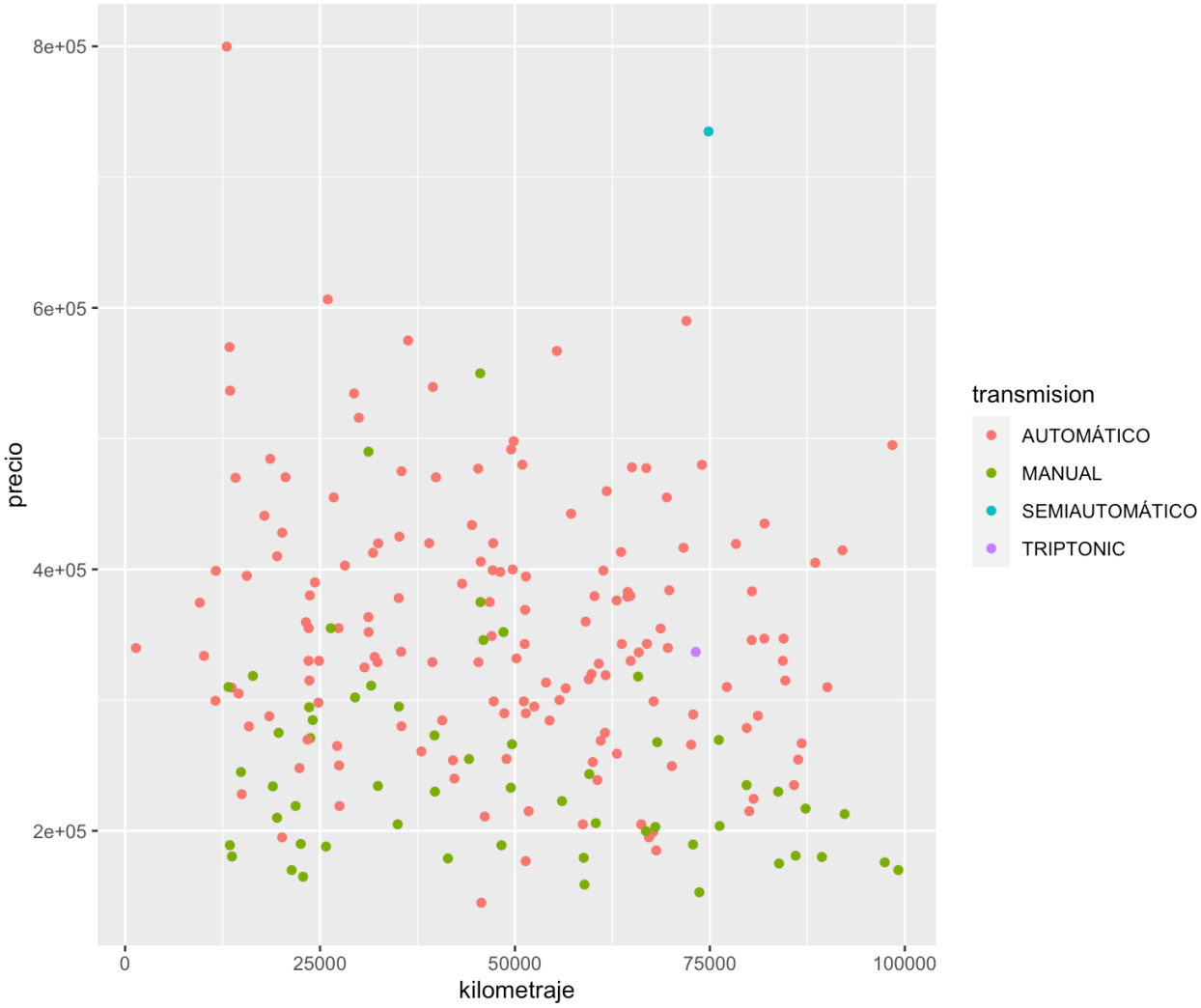


Figure 2.6: Mileage vs price grouped by transmission type

**Citas en Formato APA**

Tzuara De Luna @tzuaradeluna, 2022. ¿Por qué los autos usados Están subiendo de Precio en Lugar de Bajar? Expansión. Available at: https://expansion.mx/empresas/2022/08/03/por-que-autos-usados-suben-de-precio-en-lugar-de-bajar [Accessed September 19, 2022].