Kernel: Python (ds_env)

Used Car Catalog Analysis

The dataset was collected from Kaggle's "Used-cars-catalog" dataset which was web-scraped from a used car website in Belarus (eastern Europe) in December 2019.

In [1]:
```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [2]:
```
cars_df = pd.read_csv('data/car_data.csv')
```

In [3]:
```
cars_df.head(5)
```

Out[3]:

	manufacturer_name	model_name	transmission	color	odometer_value	year_produ
0	Subaru	Outback	automatic	silver	190000	2010
1	Subaru	Outback	automatic	blue	290000	2002
2	Subaru	Forester	automatic	red	402000	2001
3	Subaru	Impreza	mechanical	blue	10000	1999
4	Subaru	Legacy	automatic	black	280000	2001

5 rows × 30 columns

In [4]:
```
cars_df.info()
```

Out[4]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38531 entries, 0 to 38530
Data columns (total 30 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   manufacturer_name  38531 non-null  object 
 1   model_name         38531 non-null  object 
 2   transmission       38531 non-null  object 
 3   color              38531 non-null  object 
 4   odometer_value     38531 non-null  int64  
 5   year_produced      38531 non-null  int64  
 6   engine_fuel        38531 non-null  object 
 7   engine_has_gas     38531 non-null  bool   
 8   engine_type        38531 non-null  object 
 9   engine_capacity    38521 non-null  float64
 10  body_type          38531 non-null  object 
 11  has_warranty       38531 non-null  bool   
 12  state              38531 non-null  object 
 13  drivetrain         38531 non-null  object 
 14  price_usd          38531 non-null  float64
 15  is_exchangeable    38531 non-null  bool   
 16  location_region    38531 non-null  object 
 17  number_of_photos   38531 non-null  int64  
 18  up_counter         38531 non-null  int64  
 19  feature_0          38531 non-null  bool   
 20  feature_1          38531 non-null  bool   
 21  feature_2          38531 non-null  bool   
 22  feature_3          38531 non-null  bool   
 23  feature_4          38531 non-null  bool   
 24  feature_5          38531 non-null  bool   
 25  feature_6          38531 non-null  bool   
 26  feature_7          38531 non-null  bool
```

```
 27  feature_8          38531 non-null  bool
 28  feature_9          38531 non-null  bool
 29  duration_listed    38531 non-null  int64
dtypes: bool(13), float64(2), int64(5), object(10)
memory usage: 5.5+ MB
```

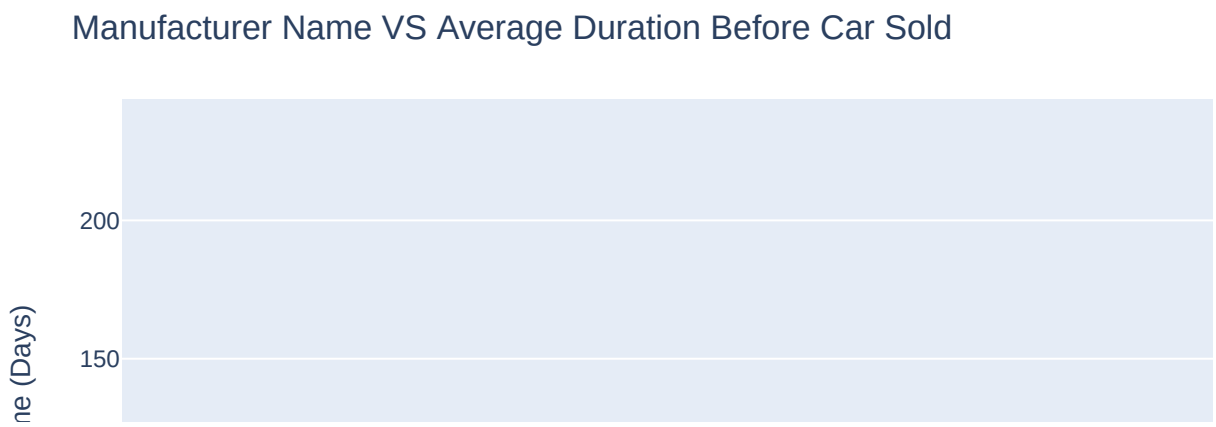
In [5]:
```
#List of columns to drop
columns_to_drop = ['feature_0', 'feature_1', 'feature_2',
       'feature_3', 'feature_4', 'feature_5', 'feature_6', 'feature_7',
       'feature_8', 'feature_9', 'engine_fuel', 'up_counter']

#Columns are dropped
cars_df.drop(columns_to_drop, axis=1, inplace = True) #axis = 1 lets pandas know we
are dropping columns, not rows.

#Translating russian to english
cars_df["location_region"].replace({"Минская обл.":"Minsk","Гомельская обл.":"Gomel",
"Витебская обл.":"Vitebsk", "Брестская обл.":"Brest", "Могилевская обл.":"Mogilev",
"Гродненская обл.":"Grodno"}, inplace=True)
```

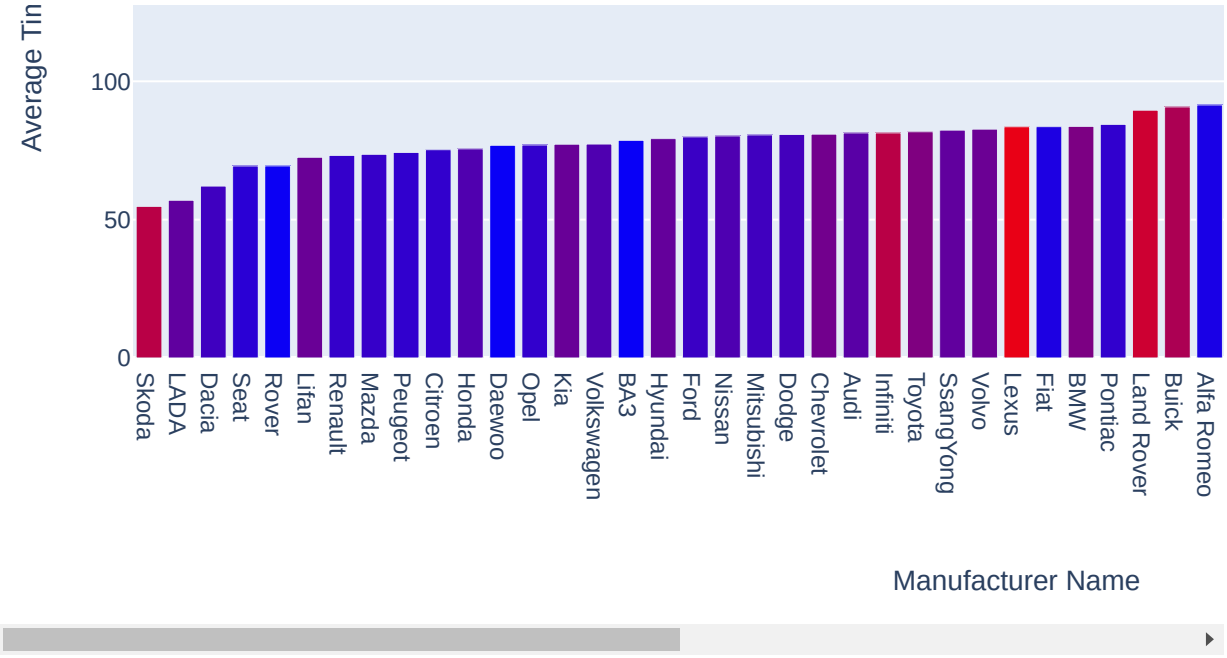
Relationship Between Car Features and Duration Listed

This data study is centered on the question: How do car features, like manufacturer or transmission, affect the duration of the car's listing before getting sold? We investigated how features like transmission, drivetrain, price or odometer value have an affect on the duration of the specific car's listing. We hypothesize that there will be differences among the car features that affect the duration of the car's listing.

In [6]:
```
x = cars_df['manufacturer_name'].value_counts()
y, y2=[], []
for name in x.index:
    y.append(cars_df[cars_df["manufacturer_name"] == name]['duration_listed'].mean())
    y2.append(cars_df[cars_df["manufacturer_name"] == name]['price_usd'].mean())
z={
    "Manufacturer Name":x.index,
    "Average Time (Days)":y,
    "Average Price":y2
}
avgPrice_df = pd.DataFrame(data=z)
avgPrice_df.sort_values(by="Average Time (Days)", ascending=True, inplace=True)

box_manufacturer = px.bar(avgPrice_df, x="Manufacturer Name", y="Average Time
(Days)", color="Average Price", color_continuous_scale="Bluered")
box_manufacturer.update_layout(
    title = 'Manufacturer Name VS Average Duration Before Car Sold',
    width=1100
)

box_manufacturer.show()
```

Out[6]:

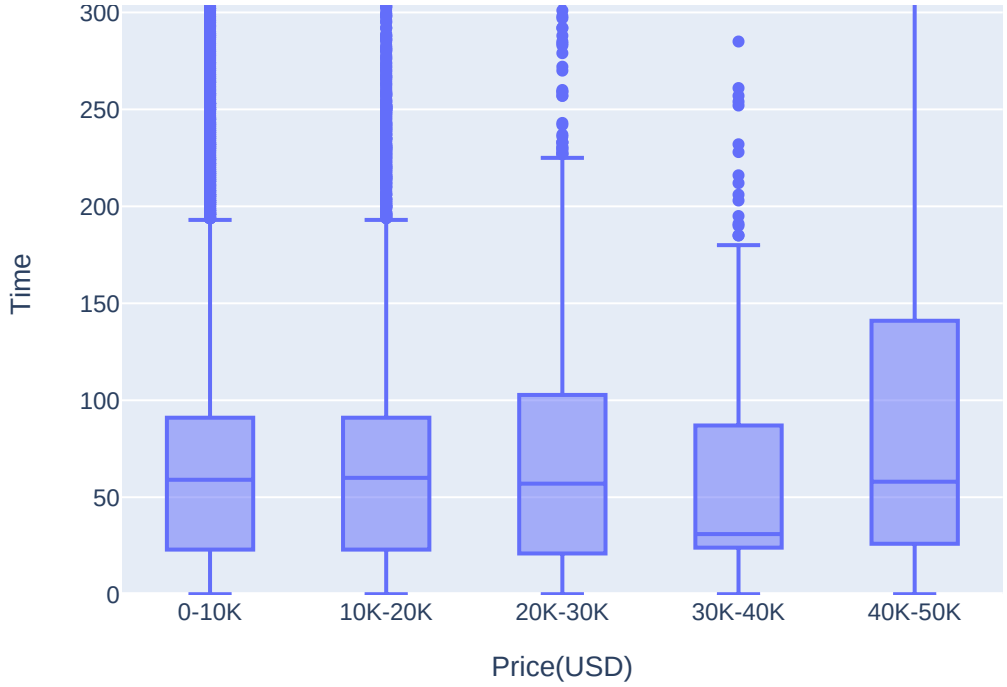




This figure represents the average duration of each manufacturer before the car got sold and the average price. As shown in the figure the average time to sell the a car is dependent on the manufacturer. One factor we believe influences the differences in manufacturer sale times is the average price. Cheaper cars like Mazda, Rover and Honda have a less average listing duration than expensive cars such as Jaguars, Jeep and Lincoln. There are a few exceptions, such as expensive cars like Skoda being sold a lot quicker than a cheaper car, like Lancia.

In [7]:
```
y=[]

for i in cars_df["price_usd"]:
    if(i<=10000):
        y.append("0-10K")
    elif(i<=20000):
        y.append("10K-20K")
    elif(i<=30000):
        y.append("20K-30K")
    elif(i<=40000):
        y.append("30K-40K")
    elif(i<=50000):
        y.append("40K-50K")

z={
    "Price(USD)":y,
    "Time":cars_df['duration_listed'],
    "Price":cars_df['price_usd']
}
avgPrice_df = pd.DataFrame(data=z)
avgPrice_df.sort_values(by="Price(USD)", ascending=True, inplace=True)
box_manufacturer = px.box(avgPrice_df, x="Price(USD)", y="Time", range_y=[0,320])

box_manufacturer.update_layout(
    width = 600,
    height = 450,
    title = 'Price(USD) VS Time(Days)',
    showlegend = False
)

box_manufacturer.show()
```

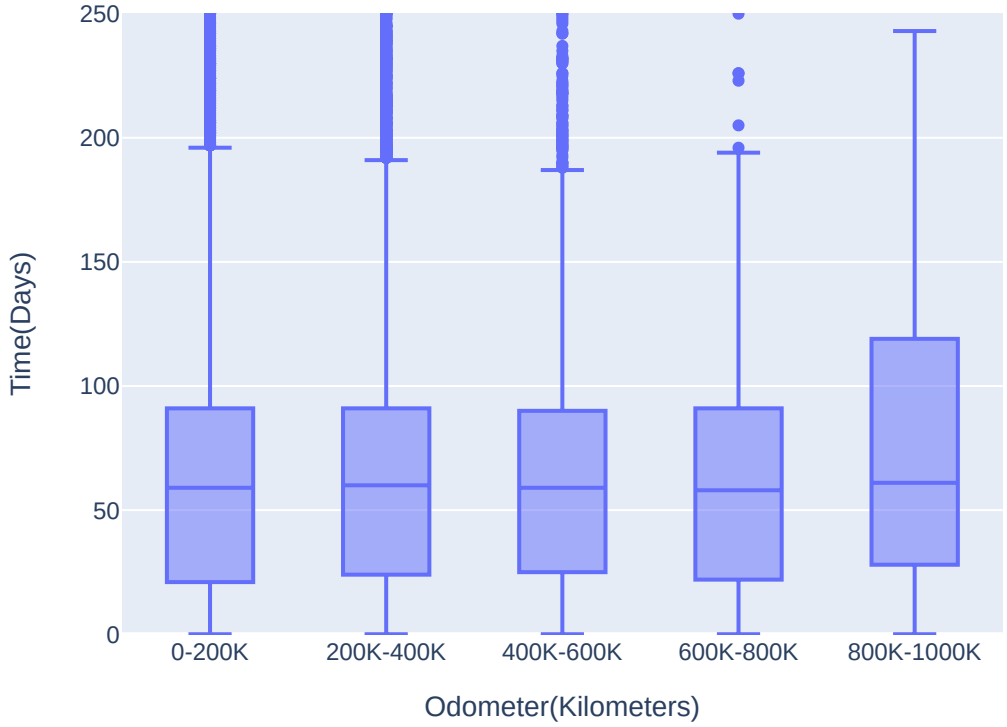
Out[7]:



This figure represents the listing duration before the car was sold and the price. The figures show that the duration before being sold is not very dependent on price. Most of the price ranges have similar 1st and 3rd quartiles. An exception is 40K to 50K price range which had a larger 3rd quartile, representing that a large proportion of cars in this price range took longer to sell.

In [8]:
```
k=[]

for i in cars_df["odometer_value"]:
    if (i<=200000):
        k.append("0-200K")
    elif (i<=400000):
        k.append("200K-400K")
    elif (i<=600000):
        k.append("400K-600K")
    elif (i<=800000):
        k.append("600K-800K")
    elif (i<=1000000):
        k.append("800K-1000K")

z={
    "Odometer(Kilometers)":k,
    "Time(Days)":cars_df['duration_listed'],
    "Odometer(KM)":cars_df['odometer_value']
}
avgPrice_df = pd.DataFrame(data=z)
avgPrice_df.sort_values(by="Odometer(KM)", ascending=True, inplace=True)
box_manufacturer = px.box(avgPrice_df, x="Odometer(Kilometers)", y="Time(Days)",
range_y=[0,250])

box_manufacturer.update_layout(
    width = 600,
    height = 450,
    title = 'Odometer VS Duration Before Car Sold'
)
box_manufacturer.show()
```

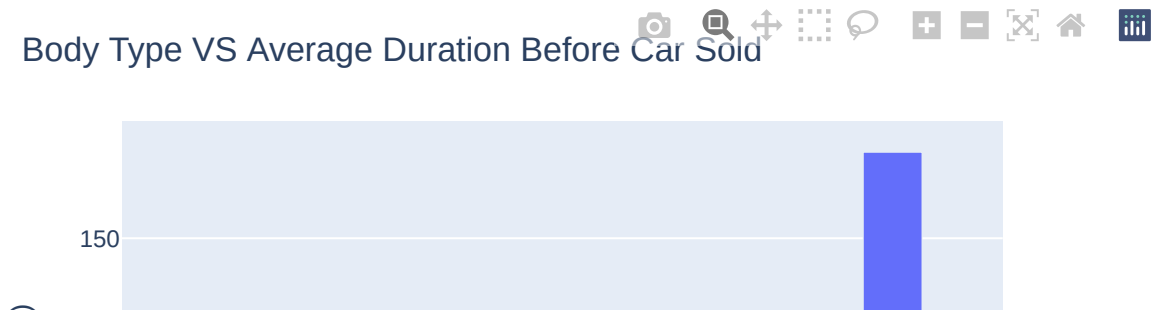
Out[8]:

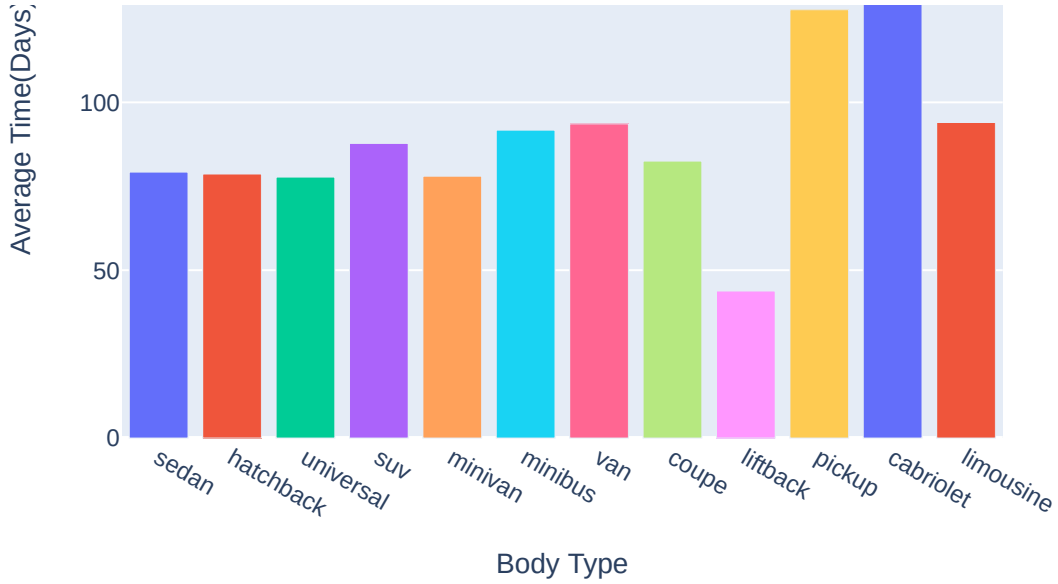


This figure represents the duration of the listing before the car got sold and the odometer meaning distance traveled in kilometers. In general, the time before being sold is not very dependent on the odometer value. This is surprising because we expected people to buy cars that have less odometer value as the car will last longer. However, the box plot shows cars with 600K-800K was sold around the same median time of 0-200K odometer value. There is a larger 3rd quartile range for the 800K-1000K group, but it's not as significant as we expected it to be.

In [9]:
```
x = cars_df['body_type'].value_counts()
y=[]
for name in x.index:
    y.append(cars_df[cars_df["body_type"] == name]['duration_listed'].mean())
z={
    "Body Type":x.index,
    "Average Time(Days)":y
}
avgPrice_df = pd.DataFrame(data=z)
#avgPrice_df.sort_values(by="Average Time(Days)", ascending=True, inplace=True)
box_manufacturer = px.bar(avgPrice_df, x="Body Type", y="Average Time(Days)",
color="Body Type")
box_manufacturer.update_layout(
    title = 'Body Type VS Average Duration Before Car Sold',
    width = 600,
    height = 450,
    showlegend=False
)

box_manufacturer.show()
```

Out[9]:

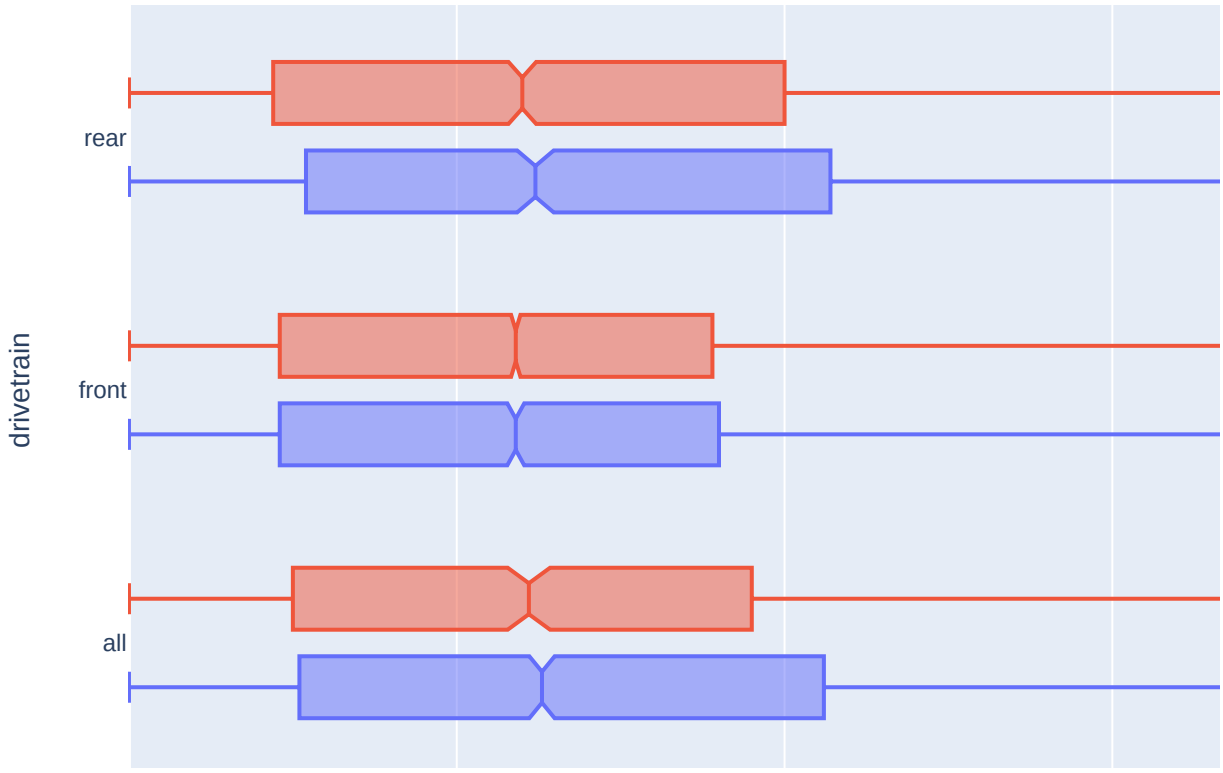


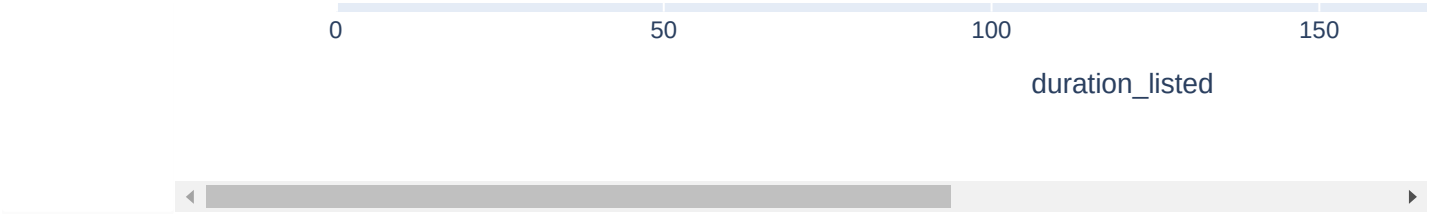


This figure represents the average duration of the listing before the different body-type cars got sold. As shown by the figure, the demand for most body size cars is similar, typically ranging between being sold in 80 to 90 days. However, some notable exceptions are liftback, pickup, and cabriolet cars. Liftback were sold much quicker than other cars, while pickup and cabriolet cars took longer to sell.

In [11]:
```
sc_yeardrive = px.box(cars_df, x="duration_listed", y="drivetrain",
color="transmission", range_x=[0,240], points='suspectedoutliers', notched=True)

sc_yeardrive.update_layout(
        width=1000
)

sc_yeardrive.show()
```

Out[11]:





This figure represents the duration of the listing before the cars got sold and the specifications. As shown in the graph, the drivetrain and transmission type do not largely affect the duration . However, automatic cars with all drivetrain or rear drivetrain took a little more time to sell than the rest.

Conclusion - Relationship Between Car Features and Duration Listed

The big question is, "Does a car's specification or manufacture affect the duration of the car's listing before getting sold?" From all the visuals and data we have seen, we can conclude that the manufacturer name has the greatest influence on the duration of a listing. Other features such as transmission, drivetrain, body type, or price did not greatly affect the duration time before being sold. There are are some exceptions in each case, but they are not very pronounced.

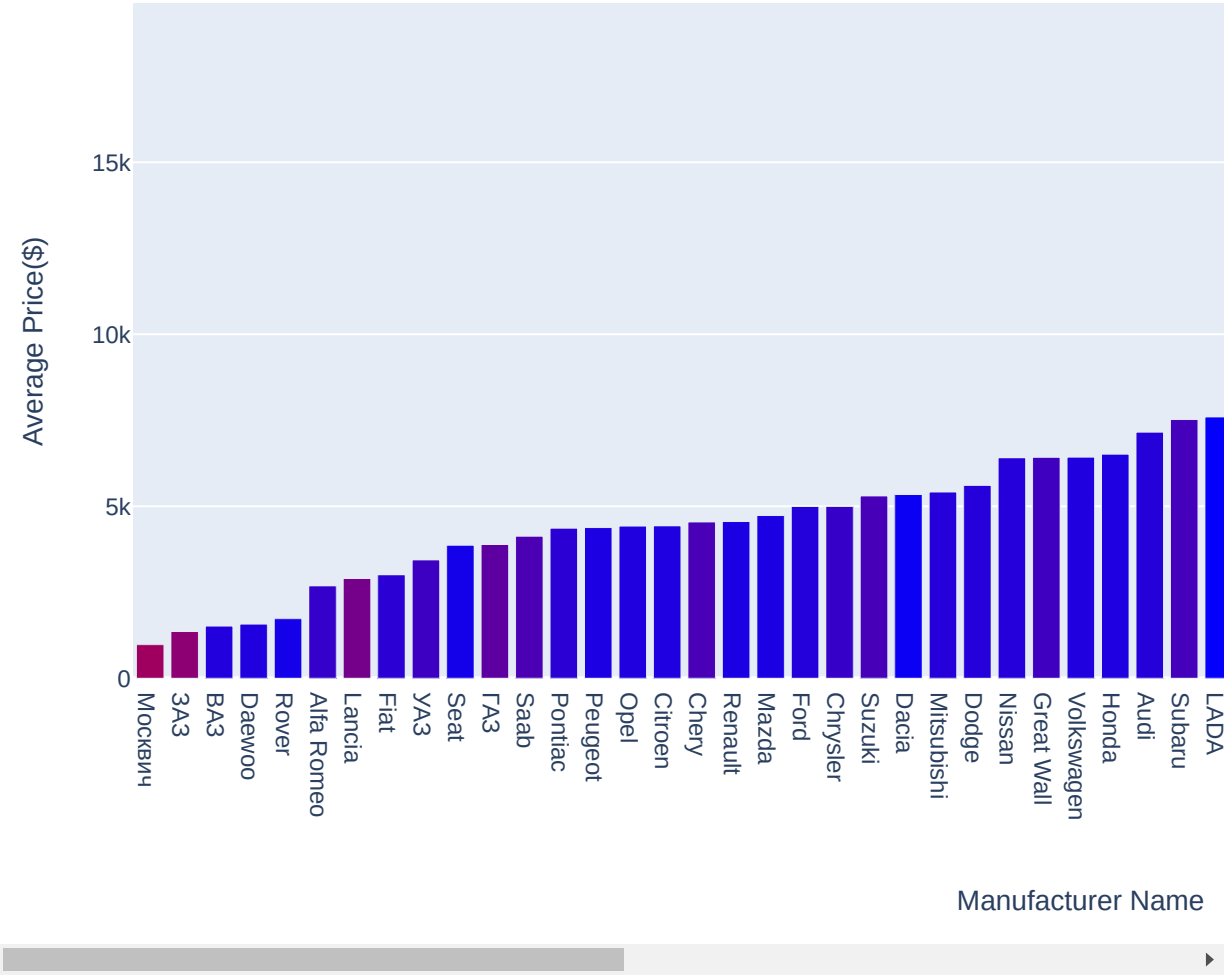
Relationship Between Car Features and Price

This part of the study was centralized on the question: Do different features affect the listing price of cars? We hypothesized that some features, like transmission and body type, may affect the price of the cars drastically. In addition, we hypothesized factors such as region and color would not really have that big of an impact on the price. To test this, we analyzed the different features listed in the dataset and compared how the color, manufacturer's name, manufactured region, and other factors may affect the listing price.

In [19]:
```
x = cars_df['manufacturer_name'].value_counts()

y, y2 = [], []
for name in x.index:
    y.append(cars_df[cars_df["manufacturer_name"] == name]['price_usd'].mean())
    y2.append(cars_df[cars_df["manufacturer_name"] == name]
['duration_listed'].mean())
z={
    "Manufacturer Name":x.index,
    "Average Price($)":y,
    "Avg Duration Listed":y2
}
avgPrice_df = pd.DataFrame(data=z)

avgPrice_df.sort_values(by="Average Price($)", ascending=True, inplace=True)

box_manufacturer = px.bar(avgPrice_df, x="Manufacturer Name", y="Average Price($)",
color="Avg Duration Listed", color_continuous_scale="Bluered")
box_manufacturer.update_layout(
    title="Average Price of Each Manufacturer's Car",
    width=1200
)

box_manufacturer.show()
```

Out[19]:

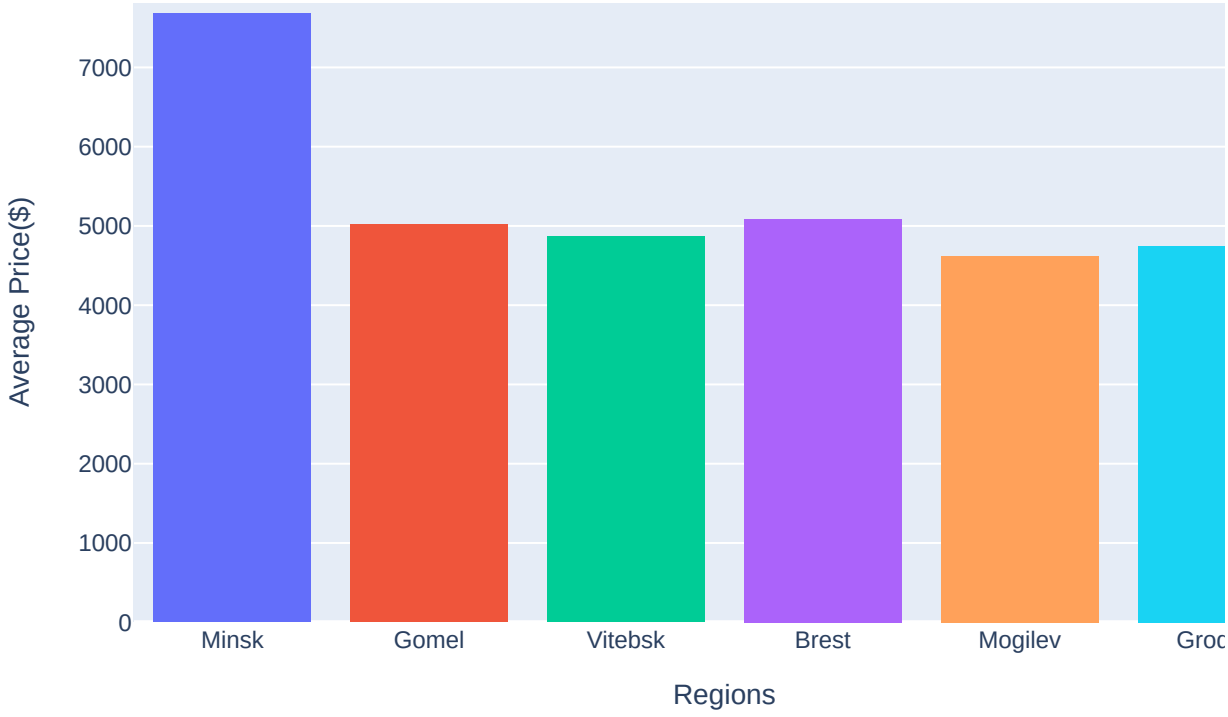
Average Price of Each Manufacturer's Car



This figure represents the average car price of each manufacturer. As shown by the figure, the average price is highly dependent on the manufacturer. For example, the average price of Porsche, Jaguar, Lexus, and Land Rover are among the highest average car prices. The difference in prices reflects how manufacturers create different quality products. In addition, it should be noted that most cars are sold before 150 days, excluding Lincoln and the two cheapest car brands.

In [20]:
```
locs = cars_df["location_region"].value_counts()
locx = locs.index
locy = []
for location in locx:
    mean = cars_df[cars_df["location_region"] == location]["price_usd"].mean()
    locy.append(mean)

hist_location = px.histogram(x=locx, y=locy,
                             labels={
                                 "x":"Regions"
                             },
                              color=locx,
                              title="Average Price of Cars in Each Region")
hist_location.update_layout(height=500, width=750, yaxis_title="Average Price($)",
showlegend=False)
hist_location.show()
```

Out[20]:

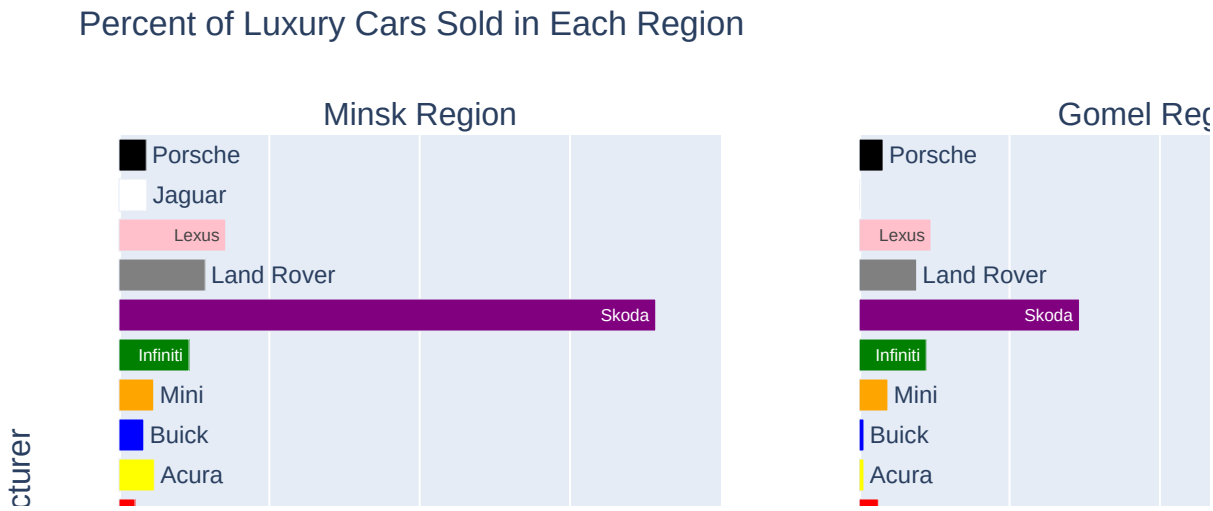


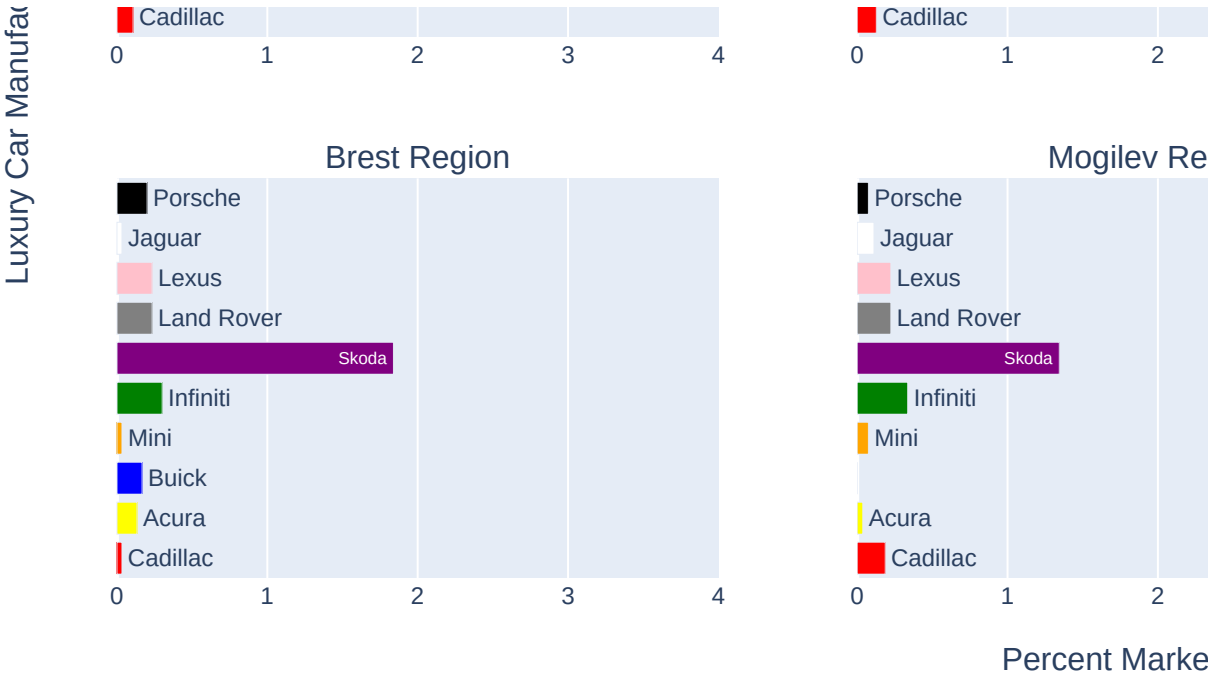
This figure represents the average prices of cars in the regions they were sold in. As shown by the figure, the average price in Minsk Region is significantly higher than the average price of cars sold in the other regions. To understand why there are differences in average prices in each region, we further analyze the types of cars being sold in each region in the next plot.

In [21]:
```
locs = cars_df["location_region"].value_counts()
locx = locs.index
locy = []
for location in locx:
    mean = cars_df[cars_df["location_region"] == location]["price_usd"].mean()
    locy.append(mean)

percent_df = pd.DataFrame(columns=["Manufacturer Name", "Minsk Region", "Gomel
Region", "Vitebsk Region", "Brest Region", "Mogilev Region", "Grodno Region"])

maxPrice_df = avgPrice_df["Manufacturer Name"].tail(10)

for i in range(10):
    car = maxPrice_df.iloc[i]
    loclist = [car]
    j = 0
    for loc in locx:
        num = len(cars_df[(cars_df["manufacturer_name"] == car) &
(cars_df["location_region"] == loc)].index)
        num /= locs.iloc[j]
        num *= 100
        loclist.append(num)
        j += 1
    percent_df.loc[i] = loclist

car_colors = {'Cadillac':'red', 'Acura':'yellow', 'Buick':'blue', 'Mini':'orange',
'Infiniti':'green', 'Skoda':'purple', 'Land Rover':'grey',
       'Lexus':'pink', 'Jaguar':'white', 'Porsche':'black'}

def mapColors():
    colors = []
    for i in range(10):
```

```
        colors.append(car_colors[percent_df["Manufacturer Name"].iloc[i]])
    return colors



color_list = mapColors()

fig_carlocs = make_subplots(rows=2, cols=3,
                            subplot_titles=("Minsk Region", "Gomel Region", "Vitebsk
Region", "Brest Region", "Mogilev Region", "Grodno Region"),
                            vertical_spacing = 0.15,
                            x_title="Percent Market Share",
                            y_title="Luxury Car Manufacturer",
                            )

trace0 = go.Bar(x=percent_df["Minsk Region"], marker=dict(color = color_list),
text=percent_df["Manufacturer Name"], hoverlabel=dict(namelength=0))
trace1 = go.Bar(x=percent_df["Gomel Region"], marker=dict(color = color_list),
text=percent_df["Manufacturer Name"], hoverlabel=dict(namelength=0))
trace2 = go.Bar(x=percent_df["Vitebsk Region"], marker=dict(color = color_list),
text=percent_df["Manufacturer Name"], hoverlabel=dict(namelength=0))
trace3 = go.Bar(x=percent_df["Brest Region"], marker=dict(color = color_list),
text=percent_df["Manufacturer Name"], hoverlabel=dict(namelength=0))
trace4 = go.Bar(x=percent_df["Mogilev Region"], marker=dict(color = color_list),
text=percent_df["Manufacturer Name"], hoverlabel=dict(namelength=0))
trace5 = go.Bar(x=percent_df["Grodno Region"], marker=dict(color = color_list),
text=percent_df["Manufacturer Name"], hoverlabel=dict(namelength=0))

fig_carlocs.append_trace(trace0, 1, 1)
fig_carlocs.append_trace(trace1, 1, 2)
fig_carlocs.append_trace(trace2, 1, 3)
fig_carlocs.append_trace(trace3, 2, 1)
fig_carlocs.append_trace(trace4, 2, 2)
fig_carlocs.append_trace(trace5, 2, 3)


fig_carlocs.update_layout(
    title="Percent of Luxury Cars Sold in Each Region",
    width=1200,
    height=650,
    showlegend=False,
)


for i in range(1,7):
    fig_carlocs['layout'][f'yaxis{i}']['showticklabels'] = False
fig_carlocs.update_xaxes(range=[0, 4.])

fig_carlocs.show()
```

Out[21]:

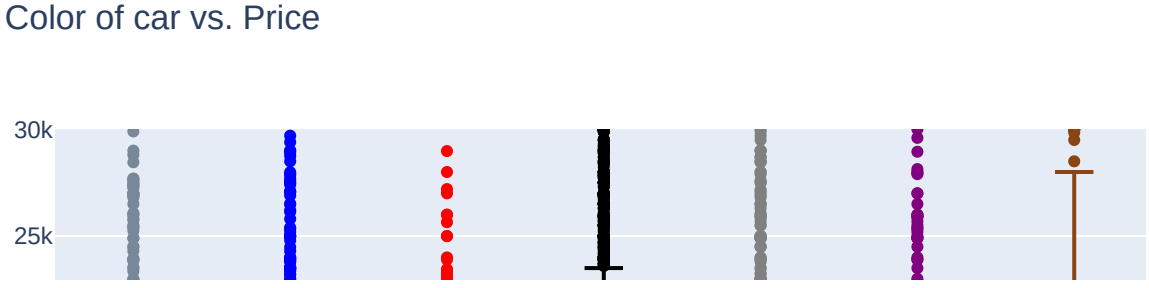


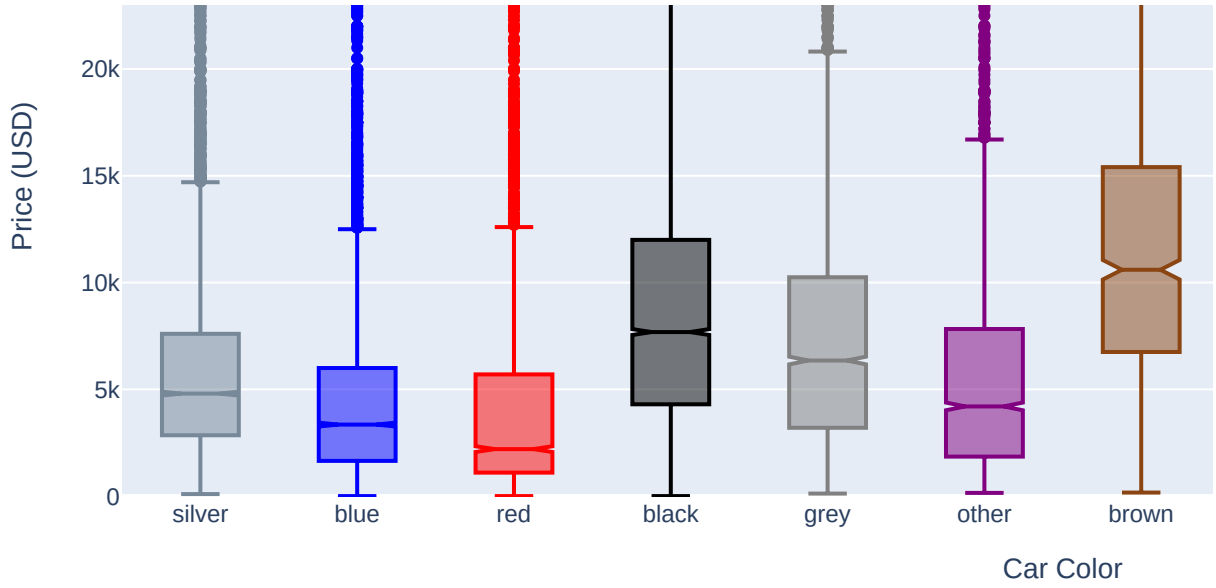


This figure represents the proportion of luxury cars sold as a percentage of the total amount of cars sold in each region. In the graph, there is a higher proportion of luxury cars sold in the Minsk Region than in the others. For example, Skoda cars are sold in a higher proportion in the Minsk Region than in any other region. We believe the higher average price in Minsk is due to the fact that a higher percentage of luxury cars are sold in Minsk.

In [15]:
```
box_color = px.box(cars_df, x="color", y="price_usd",
                   labels={"color":"Car Color","price_usd":"Price (USD)"},
                   title="Color of car vs. Price", range_y=[0,30000],
                   notched = True,
                   color='color',
                   color_discrete_map=
{'silver':'lightslategrey','blue':'blue','red':'red', 'black':'black', 'grey':'grey',
'other':'purple',
                                       'brown':'saddlebrown',
'white':'lightsteelblue', 'green':'green', 'violet':'violet', 'orange':'orange',
'yellow':'yellow'
                                           }
                  )

box_color.update_layout(
    width = 1100,
    height = 500,
    showlegend = False
)

box_color.show()
```

Out[15]:

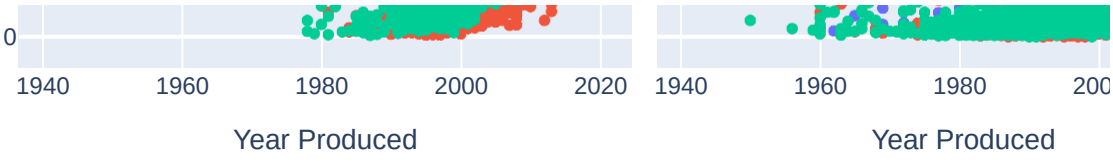




This figure represents the prices of cars based on their color. Some colors like brown, black, white, and grey have slightly higher median prices, but most of the colors have a similar distribution of prices. This matches our hypothesis that color is not a major factor in determining a car's price.

In [16]:
```
sc_yeardrive = px.scatter(cars_df, x="year_produced", y="price_usd",
color="drivetrain", facet_col ="transmission",
                          labels={
                              "price_usd":"Price (USD)",
                              "year_produced":"Year Produced",
                              "drivetrain":"Drivetrain"
                          },
                          title="Year produced vs. Price", marginal_x='histogram')
sc_yeardrive.update_layout(height=500, width=800)
sc_yeardrive.show()
```

Out[16]:



This figure represents the relation between the price of a car and the year it was produced. Cars are further separated by transmission and drivetrain, and the graph shows that cars with mechanical transmissions were produced earlier and sell for lower prices than cars with automatic transmissions. Additionally, cars with an all drivetrain have a higher price range. This is more noticeable on the graph of cars with automatic transmission, while cars with mechanical transmission have a lower range with a few outliers. The next highest range is cars with a rear drivetrain. Cars with a front drivetrain have a lower price range, despite being the most common type of car, as shown on the histograms above.

In [17]:
```
box_engine = px.box(cars_df, x="body_type", y="price_usd", color="engine_type",
                    labels={
                        "body_type":"Body Type",
                        "engine_type":"Engine Type",
                        "price_usd":"Price (USD)"
                    }, title="Body and engine type vs. Price")
box_engine.update_layout(
    width = 1200
)

box_engine.show()
```

Out[17]:

This figure represents the relationship between the body and engine type of a car and its price. According to the chart, the prices between gasoline and diesel engines do not differ significantly. However, cars with electric engines have a significantly higher price than their respective gasoline or engine counterparts. The prices ranges do not vary greatly, they lie between $5k to $10k depending on body type. However, SUVs and pickups are the most expensive with a median price around $10k.

Conclusion - Relationship Between Car Features and Price

Our central question is, "Do different features affect the listing price of cars?" From our figures and data, we have concluded that various features influence the price of a car, such as the manufacturer, transmission, and year produced. Unexpectedly, we also found that cars sold in a certain region had a higher average price. However, some features like the body type and color did not influence the price as much.

In [0]: