Fuel Type Analyst of Selling Car

By : Ika Lulus Yuliatin

Car Dekho is an Indian auto portal that helps its users with car research, finance, insurance, used cars, and any other aspect of car buying and selling. The company has tie-ups with many auto manufacturers, car dealers, and numerous financial institutions to facilitate the purchase of vehicles.

In this report, we will do data visualization analysis from 2 kinds of variable continue and 6 kinds of variable discrete The details of variables included in the dataset are :

1) Car Name 2) Year 3) Selling Price 4) Kms driven 5) Fuel 6) Seller type 7) Transmission 8) Owner

This report is a continuation of the previous detailed report which carried out further visual analysis of the fuel in the selling cars

Source of Data : https://www.kaggle.com/datasets/akshaydattatraykhare/car-details-dataset/code

In [9]:

```
# Common
import os
import numpy as np
import pandas as pd

# Data Visualization
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
from matplotlib import colors
```

In [14]:

```
import pandas as pd

car_df = pd.read_csv('CAR DETAILS FROM CAR DEKHO.csv')
car_df.head()
```

Out[14]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner
0	Maruti 800 AC	2007	60000	70000	Petrol	Individual	Manual	First Owner
1	Maruti Wagon R LXI Minor	2007	135000	50000	Petrol	Individual	Manual	First Owner
2	Hyundai Verna 1.6 SX	2012	600000	100000	Diesel	Individual	Manual	First Owner
3	Datsun RediGO T Option	2017	250000	46000	Petrol	Individual	Manual	First Owner
4	Honda Amaze VX i-DTEC	2014	450000	141000	Diesel	Individual	Manual	Second Owner

In [15]:

```
car_df.columns
```

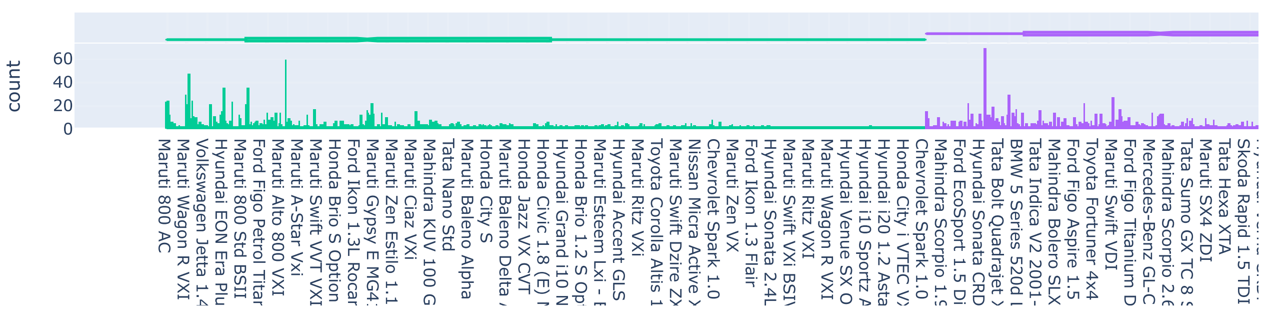
Out[15]:

```
Index(['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
       'transmission', 'owner'],
      dtype='object')
```

In [16]:

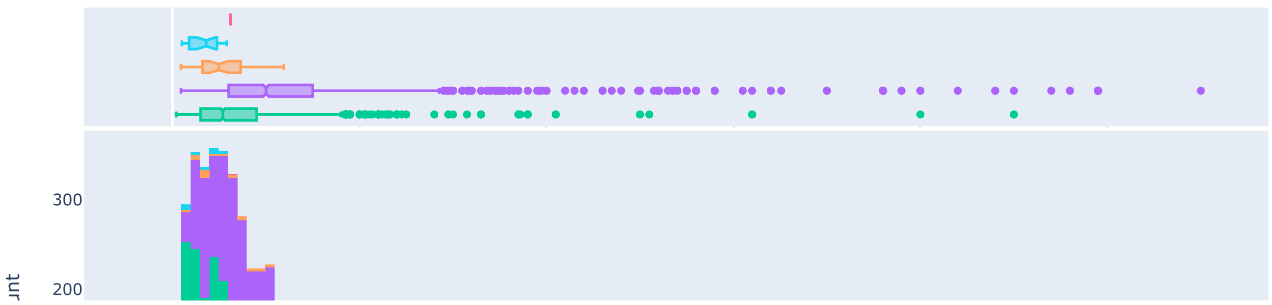
```
col = ['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
       'transmission', 'owner']
for col in col:
    fig = px.histogram(car_df, x = col ,color = 'fuel', marginal='box' , title = col + ' vs Fuel of Car' , color_discrete_map={1:'crimson
    fig.show()
```

name vs Fuel of Car

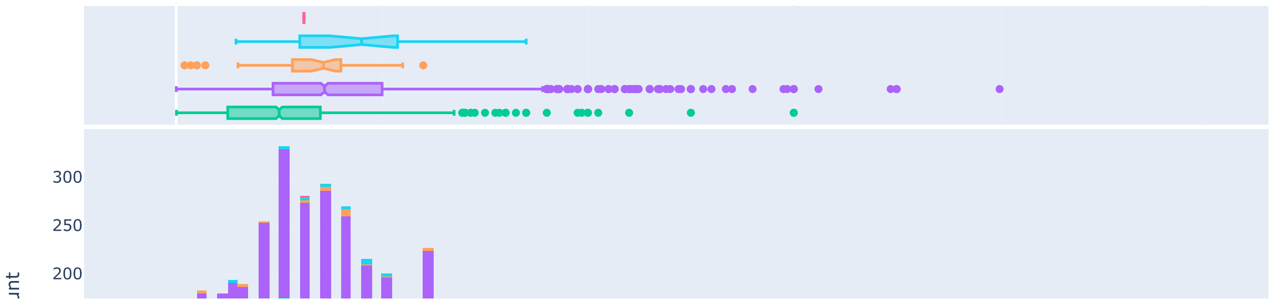


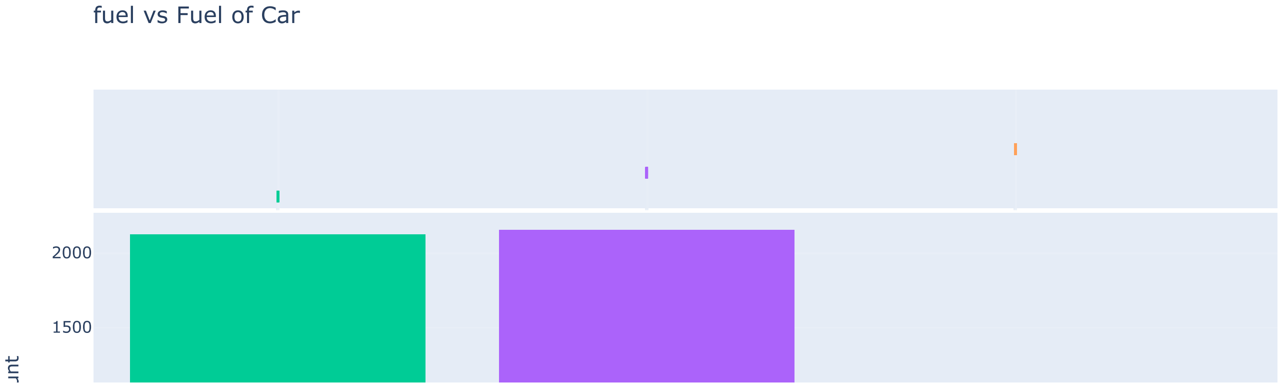
year vs Fuel of Car

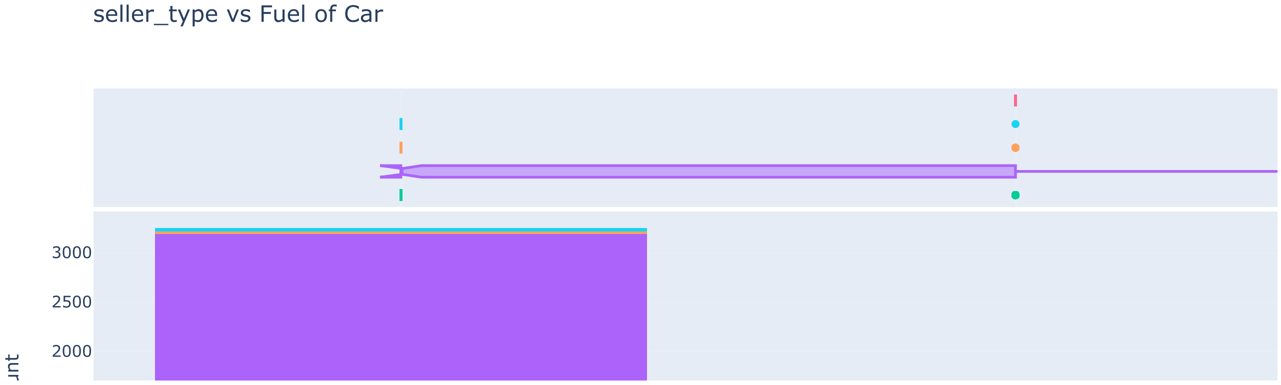
selling_price vs Fuel of Car

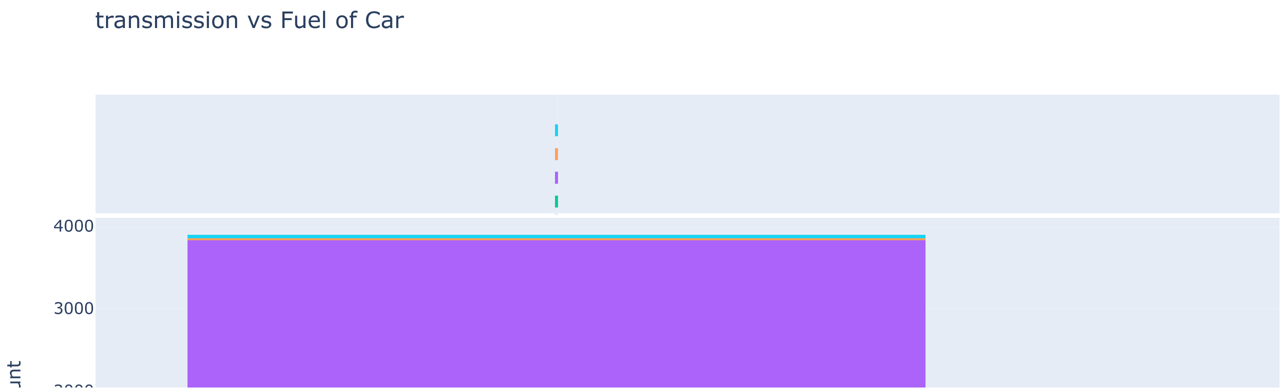


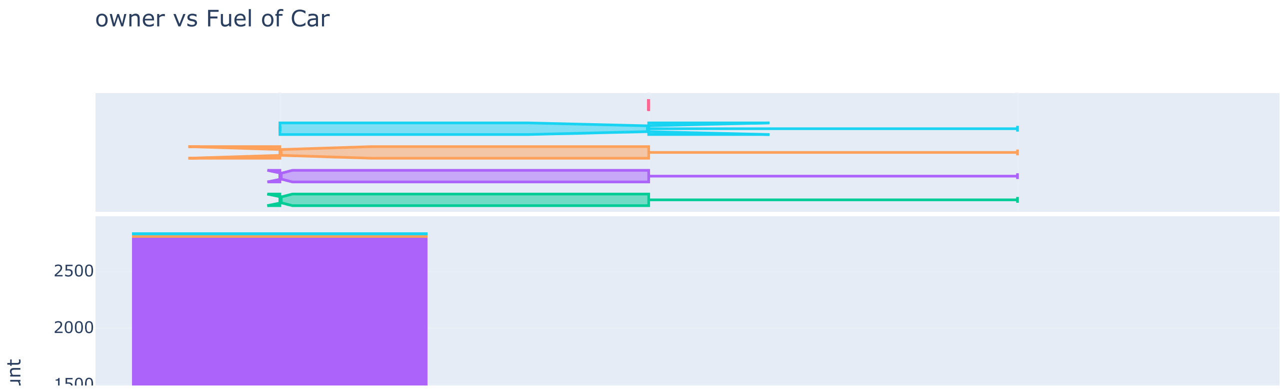
km_driven vs Fuel of Car



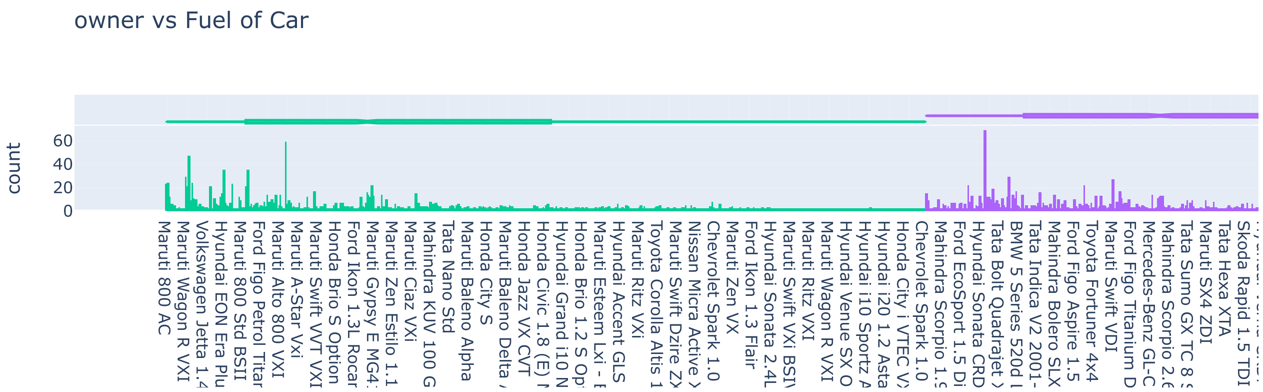

fuel vs Fuel of Car
2000
1500
unt

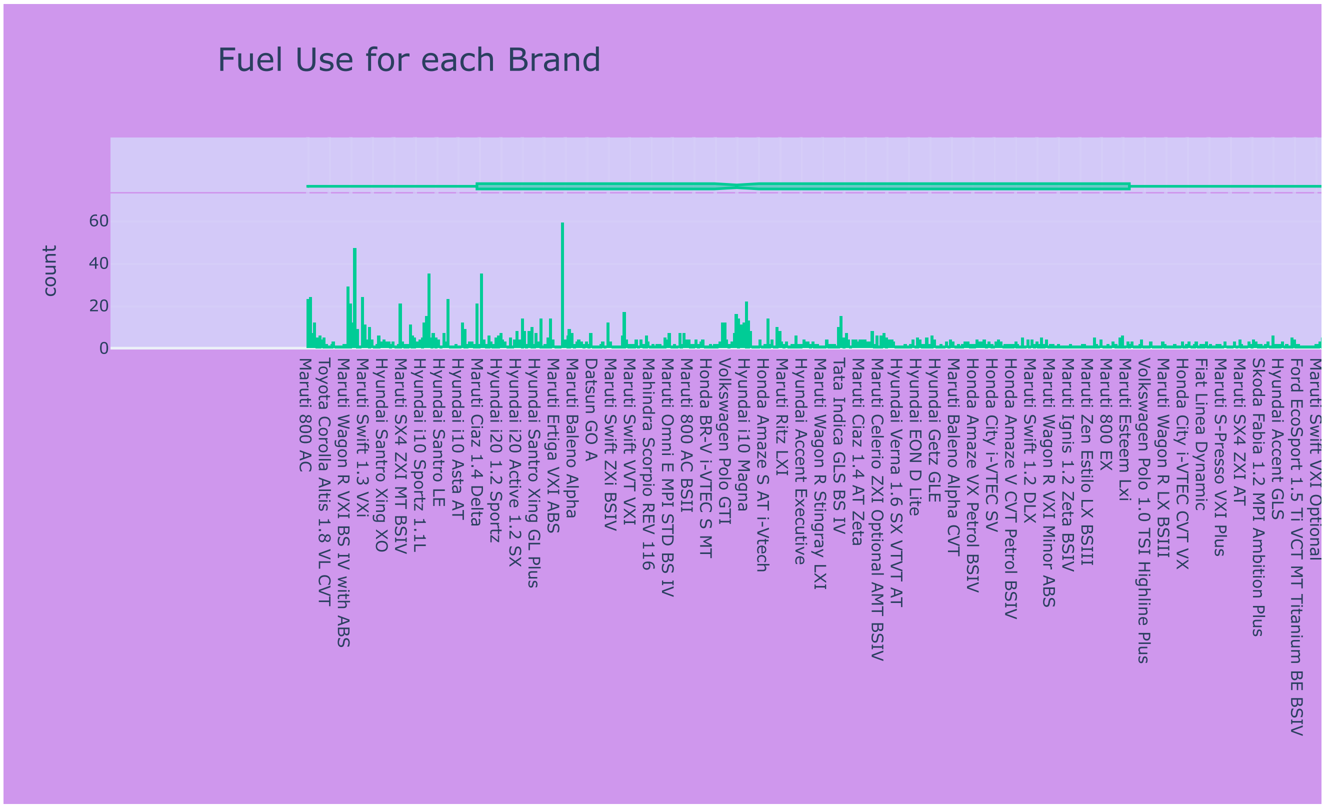

seller_type vs Fuel of Car
3000
2500
2000
unt



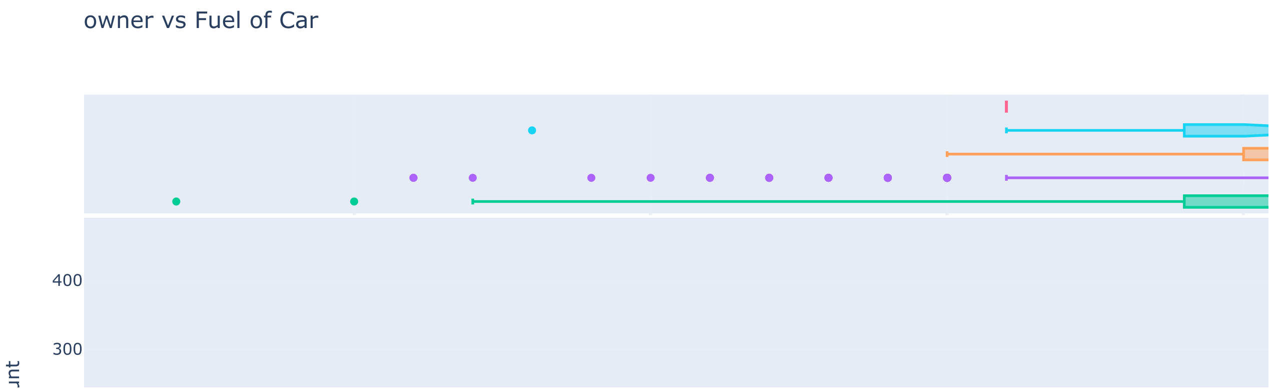


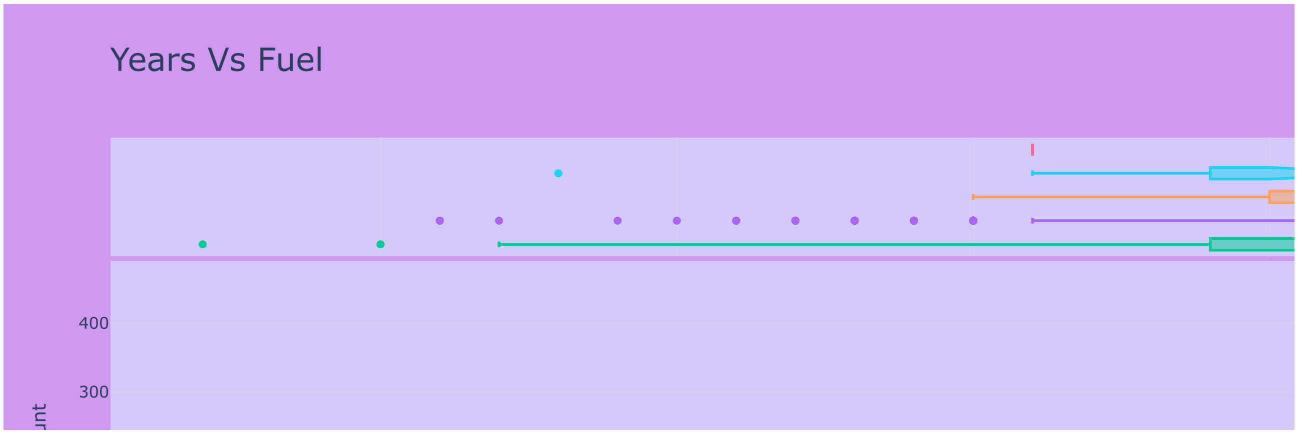
In [17]:
```
fig = px.histogram(car_df, x = 'name' ,color = 'fuel', marginal='box' , title = col + ' vs Fuel of Car' , color_discrete_map={1:'crimson'
fig.show()

fig.update_layout(title='Fuel Use for each Brand',
                  titlefont={'size': 24},
                  template='plotly_white',
                  paper_bgcolor="#cf97ed",
                  plot_bgcolor="#d3c9f8",
                  width=3200,
                  height=600
                 )
```



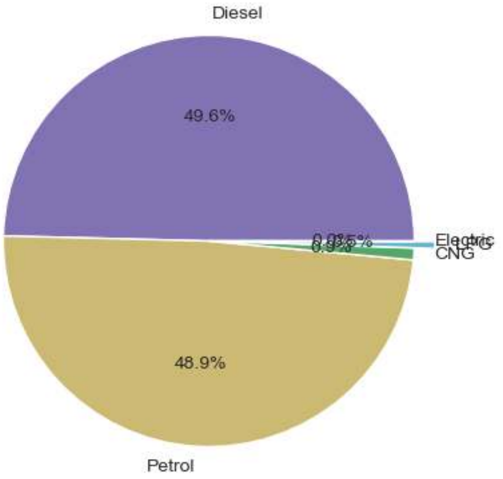


In [18]:
```
fig = px.histogram(car_df, x = 'year' ,color = 'fuel', marginal='box' , title = col + ' vs Fuel of Car' , color_discrete_map={1:'crimson'
fig.show()

fig.update_layout(title='Years Vs Fuel',
                 titlefont={'size': 24},
                 template='plotly_white',
                 paper_bgcolor="#cf97ed",
                 plot_bgcolor="#d3c9f8",
                )
```





In [19]:
```
sns.set()
labels=car_df['fuel'].value_counts().index
colors=['m','y','g','c','r']
explode=[0,0,0,0.1,0]
values=car_df['fuel'].value_counts().values
textprops = {"fontsize":10}

#visualization
plt.figure(figsize=(5,5))
plt.pie(values,explode=explode,labels=labels,colors=colors,autopct='%1.1f%%',textprops=textprops)
plt.show()
```



In [20]:

```
sns.set(rc={"axes.facecolor":"#cf97ed","figure.facecolor":"#cf97ed"})
pallet = ["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78", "#F3AB60"]
cmap = colors.ListedColormap(["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78", "#F3AB60"])
plt.figure(figsize=(8,4))
sns.lineplot(data=car_df, x="year", y="selling_price", hue="fuel", ci=False)
plt.title("Change in Price Over Years")
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_5380\1738330669.py in <module>
      1 sns.set(rc={"axes.facecolor":"#cf97ed","figure.facecolor":"#cf97ed"})
      2 pallet = ["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78", "#F3AB60"]
----> 3 cmap = colors.ListedColormap(["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78", "#F3AB60"])
      4 plt.figure(figsize=(8,4))
      5 sns.lineplot(data=car_df, x="year", y="selling_price", hue="fuel", ci=False)

AttributeError: 'list' object has no attribute 'ListedColormap'
```

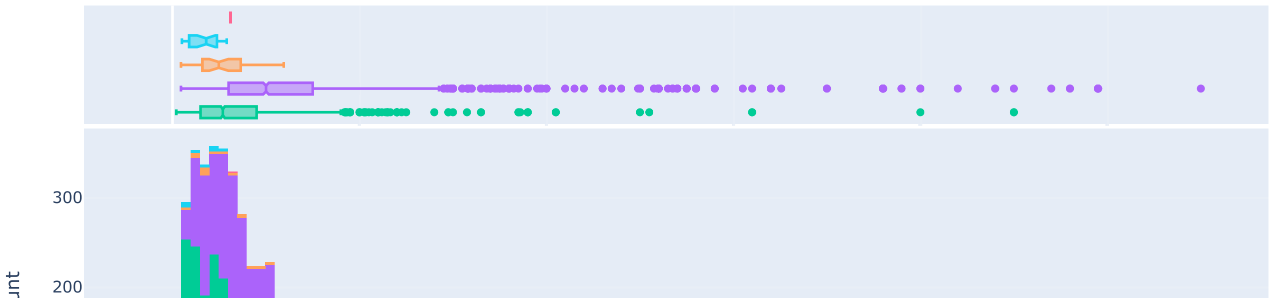
In [ ]:

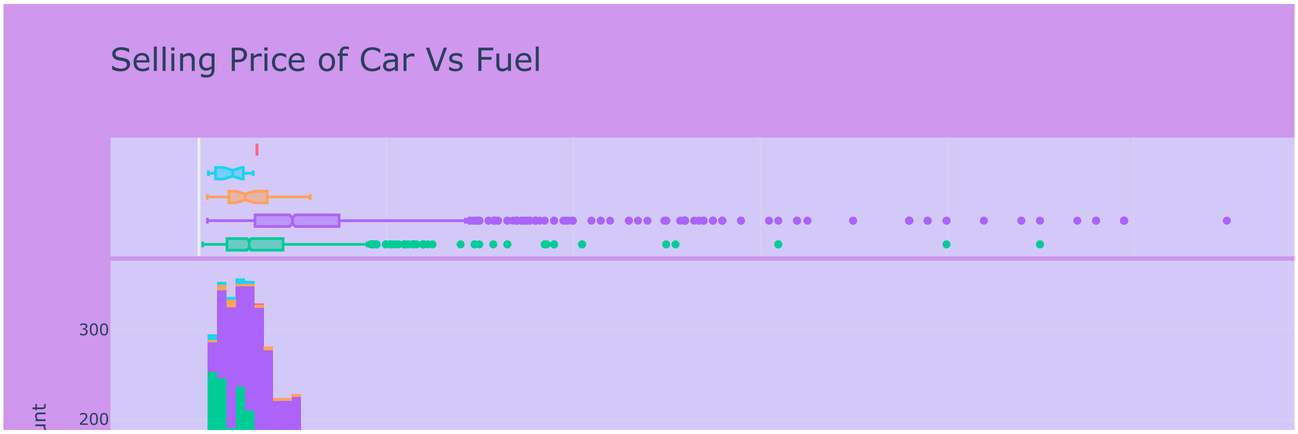
```
df_owner = car_df.query('year > 2010')
df_owner = (
    df_owner.groupby("year")["fuel"]
    .value_counts()
    .sort_index()
    .unstack()
)
ax = df_owner.T.plot(kind="bar", figsize=(8, 4), )
ax.set_title("In the past ten years ", fontsize=14)
ax.set_ylabel("kind of owner salse")
plt.xticks(rotation=-45, ha="left")
plt.show()
```

In [21]:

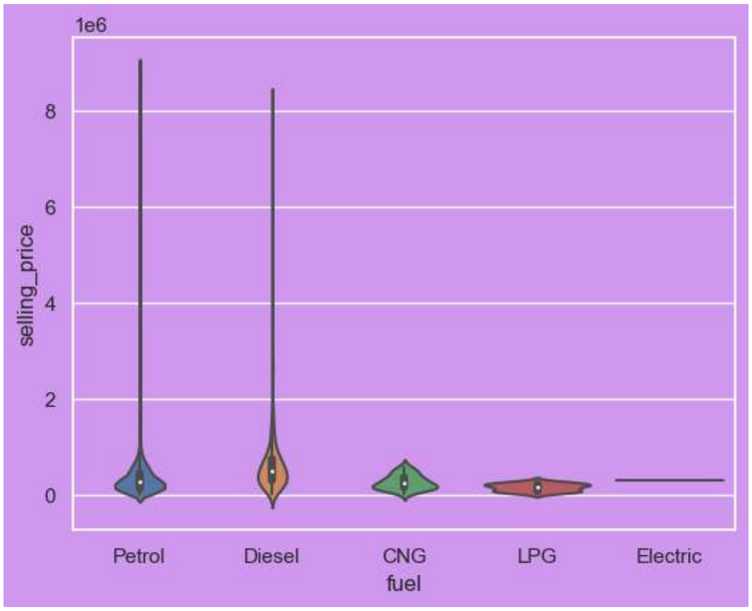
```
fig = px.histogram(car_df, x = 'selling_price' ,color = 'fuel', marginal='box' , title = col + ' vs Fuel of Car' , color_discrete_map={1:
fig.show()

fig.update_layout(title='Selling Price of Car Vs Fuel',
                  titlefont={'size': 24},
                  template='plotly_white',
                  paper_bgcolor="#cf97ed",
                  plot_bgcolor="#d3c9f8",
                 )
```

owner vs Fuel of Car

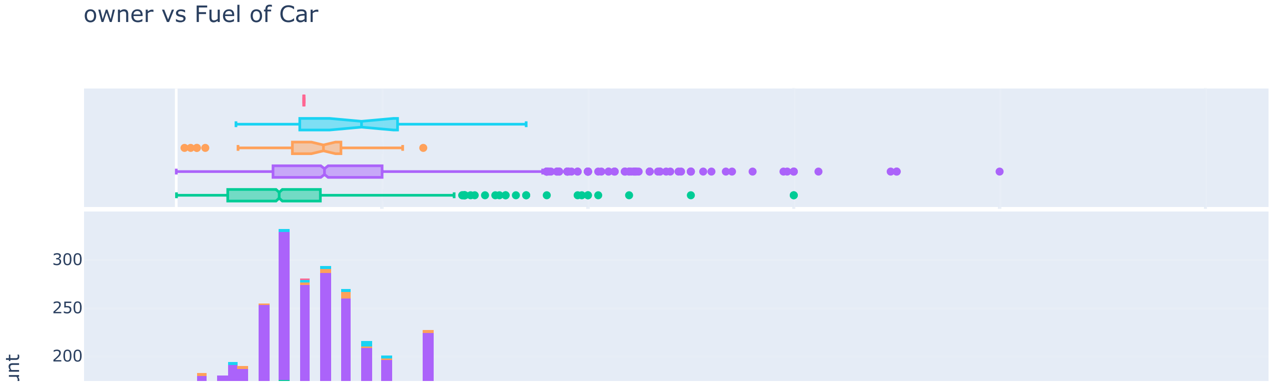


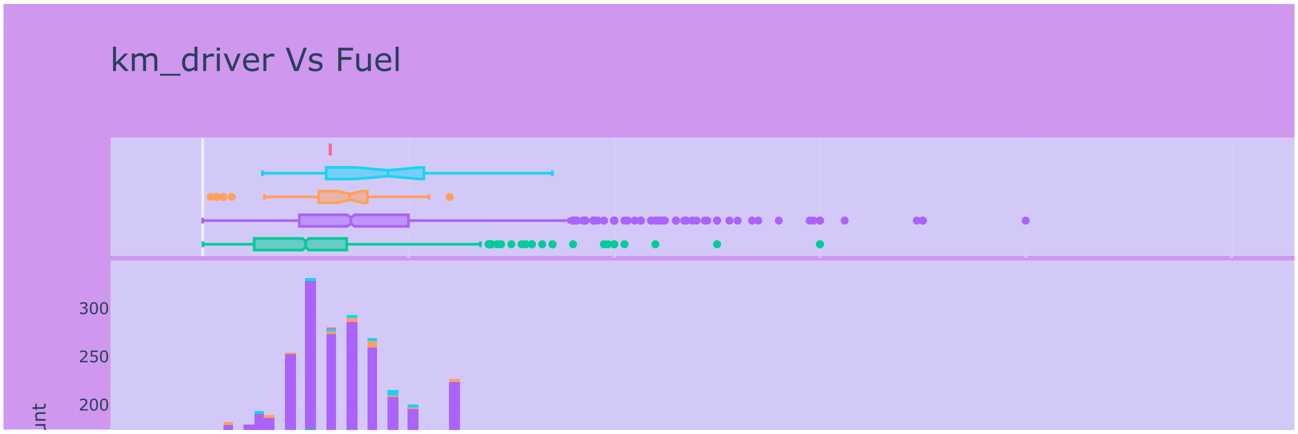

Selling Price of Car Vs Fuel

In [22]:
```
sns.violinplot(x=car_df['fuel'],y=car_df['selling_price'])
plt.show()
```

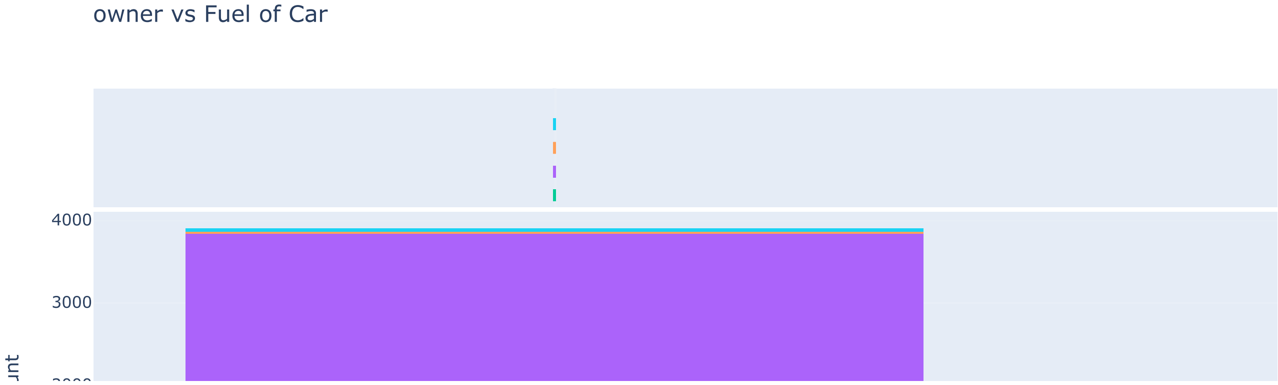


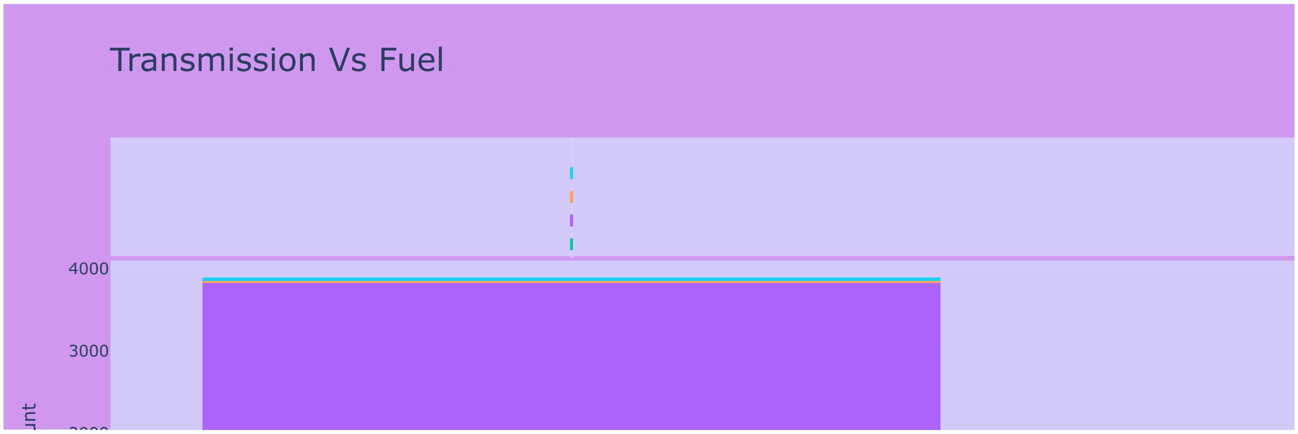
In [23]:
```
fig = px.histogram(car_df, x = 'km_driven' ,color = 'fuel', marginal='box' , title = col + ' vs Fuel of Car' , color_discrete_map={1:'cri
fig.show()

fig.update_layout(title='km_driver Vs Fuel',
                  titlefont={'size': 24},
                  template='plotly_white',
                  paper_bgcolor="#cf97ed",
                  plot_bgcolor="#d3c9f8",
                 )
```





In [24]:
```
fig = px.histogram(car_df, x = 'transmission' ,color = 'fuel', marginal='box' , title = col + ' vs Fuel of Car' , color_discrete_map={1:'
fig.show()

fig.update_layout(title='Transmission Vs Fuel',
                  titlefont={'size': 24},
                  template='plotly_white',
                  paper_bgcolor="#cf97ed",
                  plot_bgcolor="#d3c9f8",
                 )
```

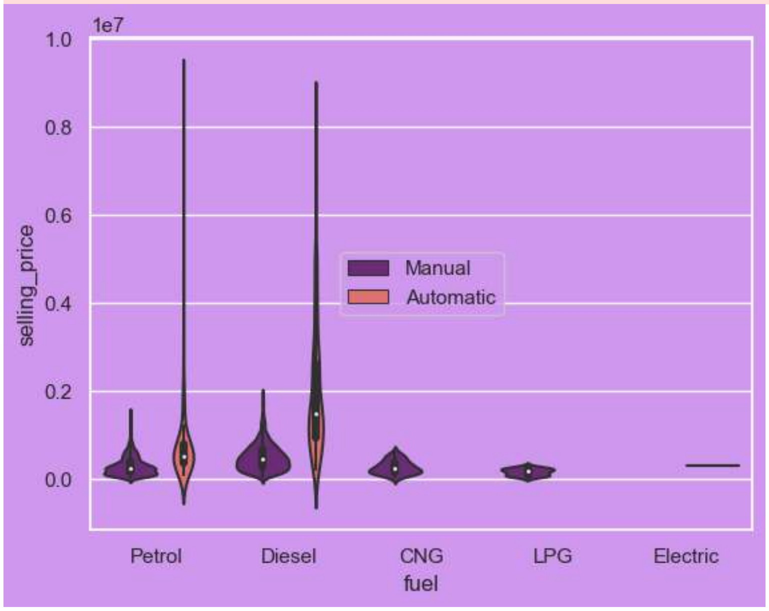




In [25]:

```
sns.violinplot(car_df['fuel'],y=car_df['selling_price'],hue=car_df['transmission'],palette='magma')
plt.legend(loc=10)
plt.show()
```

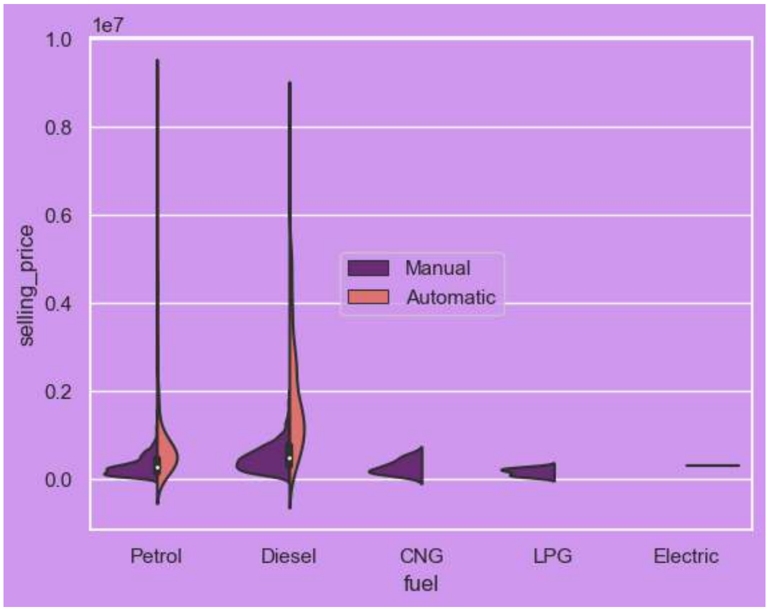
```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpretation.
```



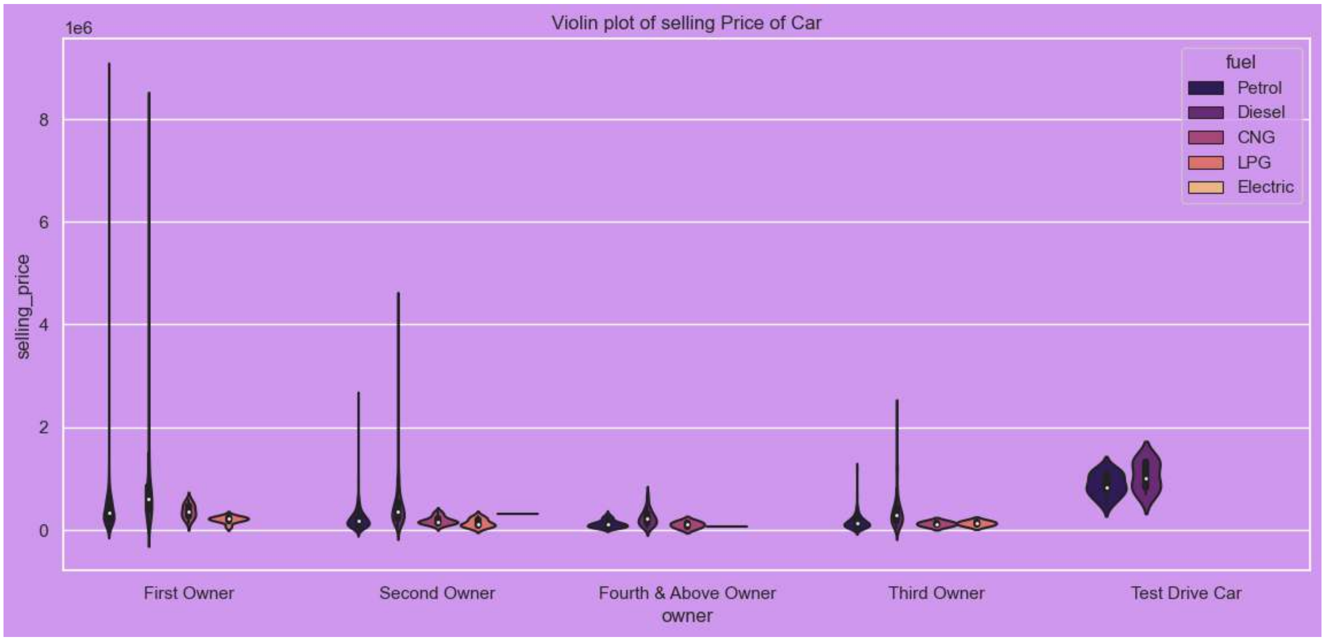
In [26]:

```
sns.violinplot(car_df['fuel'],car_df['selling_price'],
               hue=car_df['transmission'],palette='magma',split=True)
plt.legend(loc=10)
plt.show()
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
```

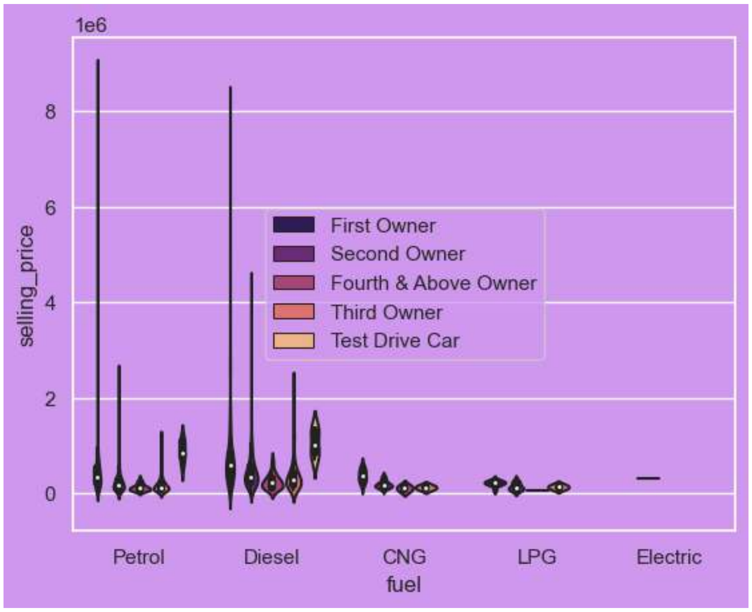


In [27]:
```
plt.figure(figsize=(14,6))
sns.violinplot(x='owner', y='selling_price',hue='fuel',data=car_df,palette='magma',split=False)
plt.title('Violin plot of selling Price of Car');
```

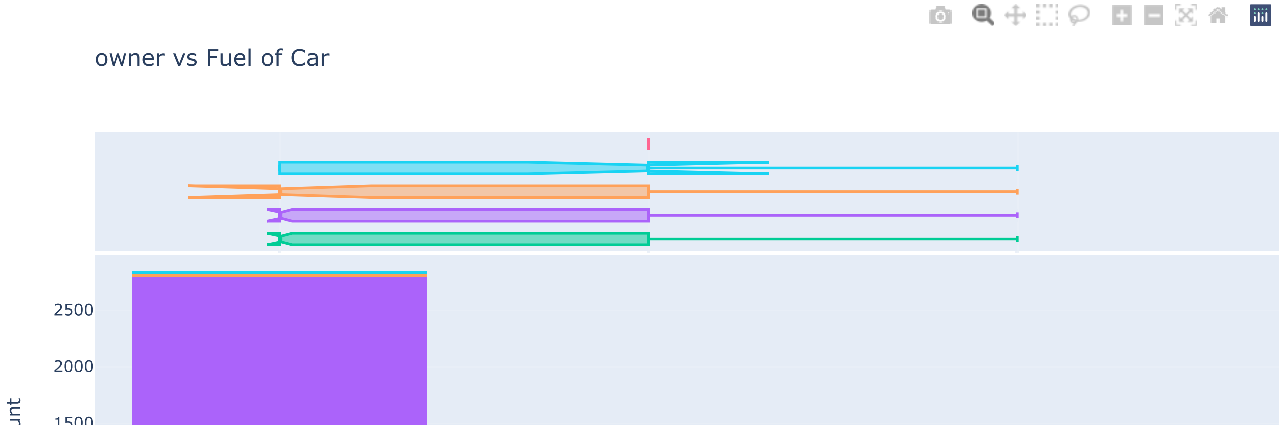


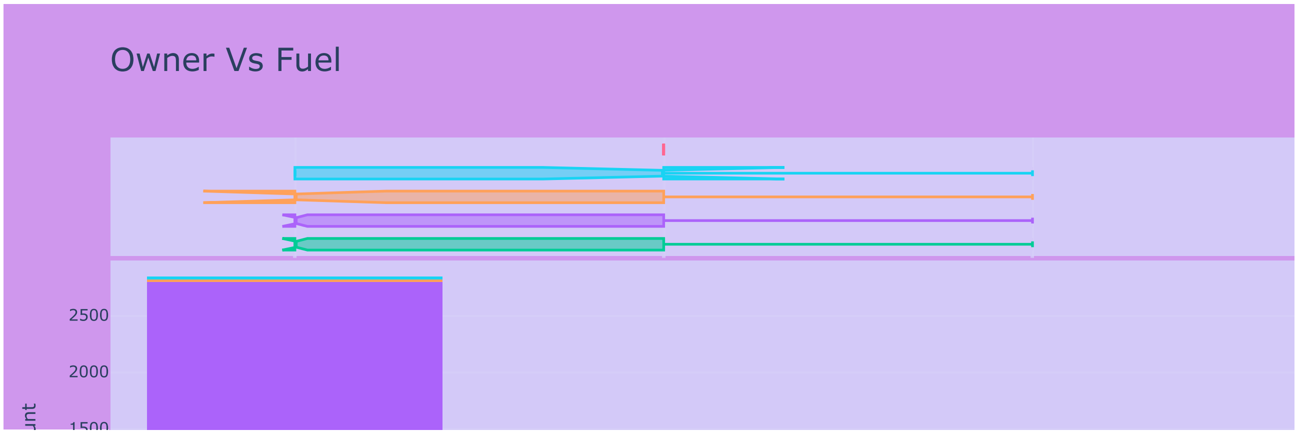
In [28]:
```
sns.violinplot(car_df['fuel'],y=car_df['selling_price'],hue=car_df['owner'],palette='magma')
plt.legend(loc=10)
plt.show()
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpretation.
```

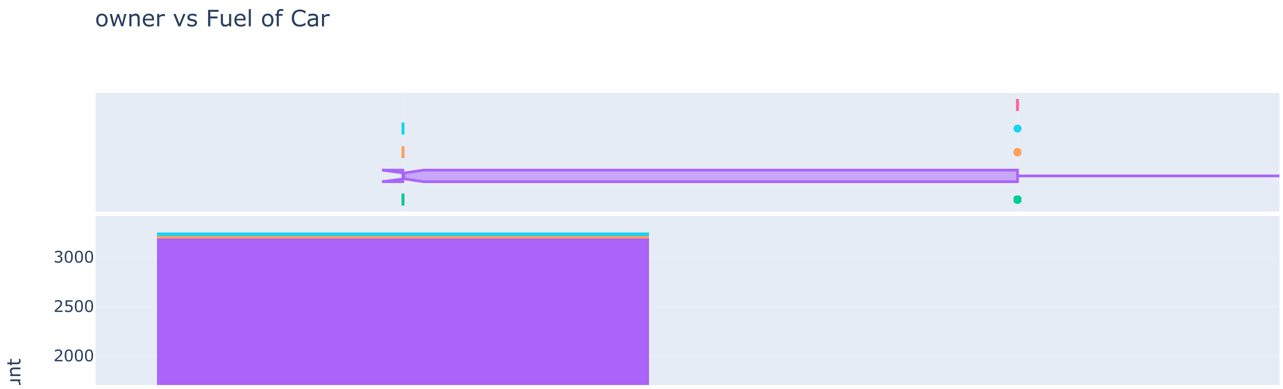


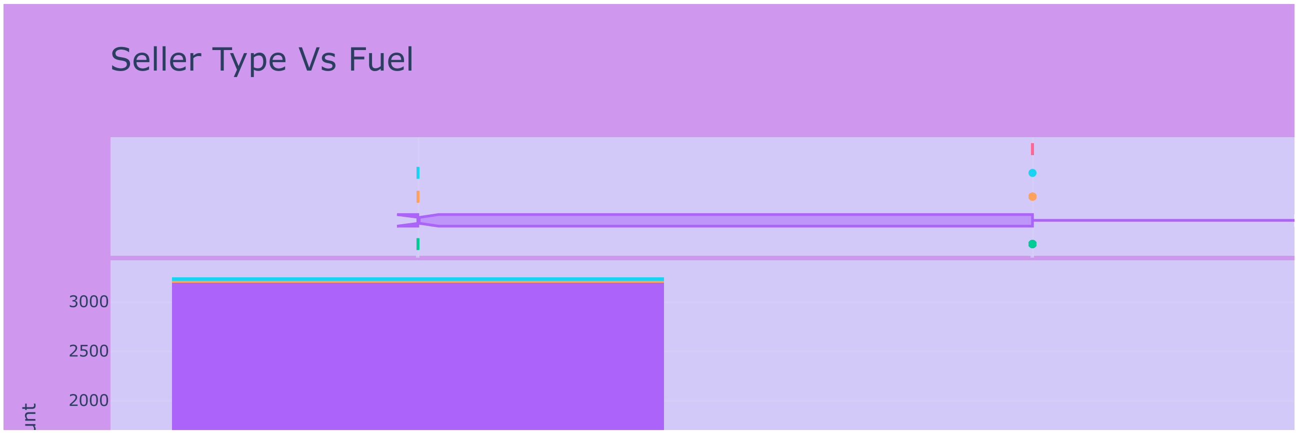
In [29]:
```
fig = px.histogram(car_df, x = 'owner' ,color = 'fuel', marginal='box' , title = col + ' vs Fuel of Car' , color_discrete_map={1:'crimson
fig.show()

fig.update_layout(title='Owner Vs Fuel',
                  titlefont={'size': 24},
                  template='plotly_white',
                  paper_bgcolor="#cf97ed",
                  plot_bgcolor="#d3c9f8",
                 )
```





In [30]:
```
fig = px.histogram(car_df, x = 'seller_type' ,color = 'fuel', marginal='box' , title = col + ' vs Fuel of Car' , color_discrete_map={1:'c
fig.show()

fig.update_layout(title='Seller Type Vs Fuel',
                  titlefont={'size': 24},
                  template='plotly_white',
                  paper_bgcolor="#cf97ed",
                  plot_bgcolor="#d3c9f8",
                 )
```

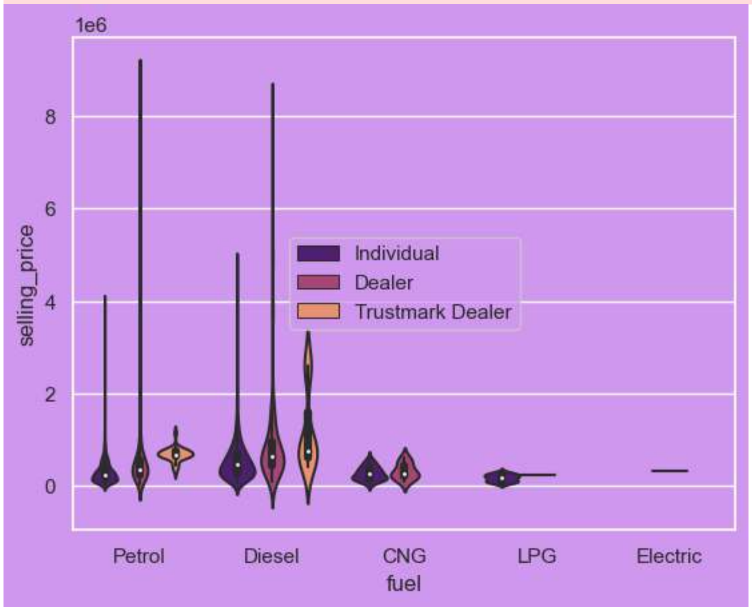




In [31]:

```
sns.violinplot(car_df['fuel'],y=car_df['selling_price'],hue=car_df['seller_type'],palette='magma')
plt.legend(loc=10)
plt.show()
```

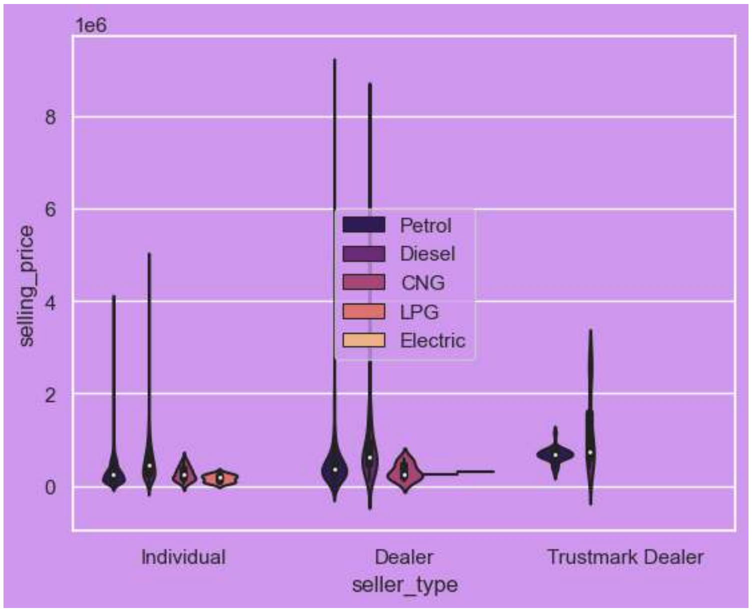
```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpretation.
```



In [32]:

```
sns.violinplot(car_df['seller_type'],car_df['selling_price'],
               hue=car_df['fuel'],palette='magma',split=False)
plt.legend(loc=10)
plt.show()
```

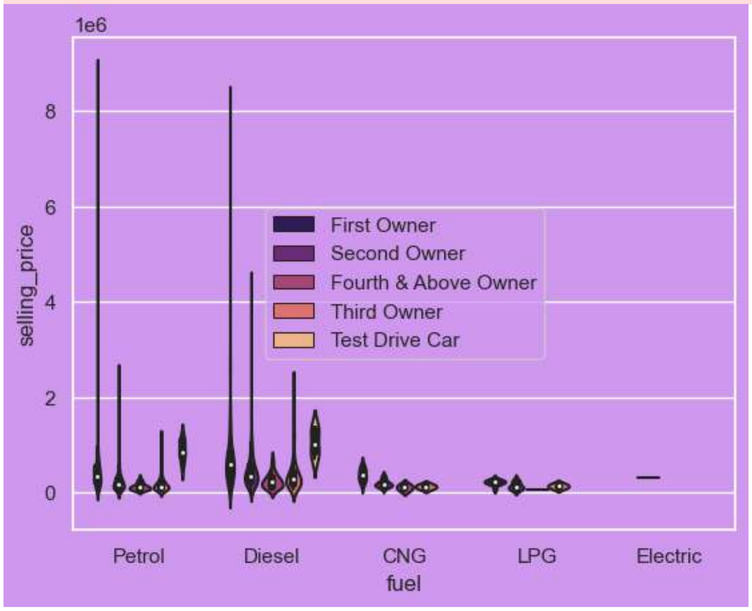
```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misinterpretation.
```



In [33]:

```
sns.violinplot(car_df['fuel'],y=car_df['selling_price'],hue=car_df['owner'],palette='magma')
plt.legend(loc=10)
plt.show()
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpretation.
```


1e6
selling_price
First Owner
Second Owner
Fourth & Above Owner
Third Owner
Test Drive Car
Petrol
Diesel
CNG
LPG
Electric
fuel