Transmission 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 (https://www.kaggle.com/datasets/akshaydattatraykhare/car-details-dataset/code)](https://www.kaggle.com/datasets/akshaydattatraykhare/car-details-dataset/code)

In [7]:
```
# 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 [2]:
```
car_df = pd.read_csv('CAR DETAILS FROM CAR DEKHO.csv')
car_df.head()
```

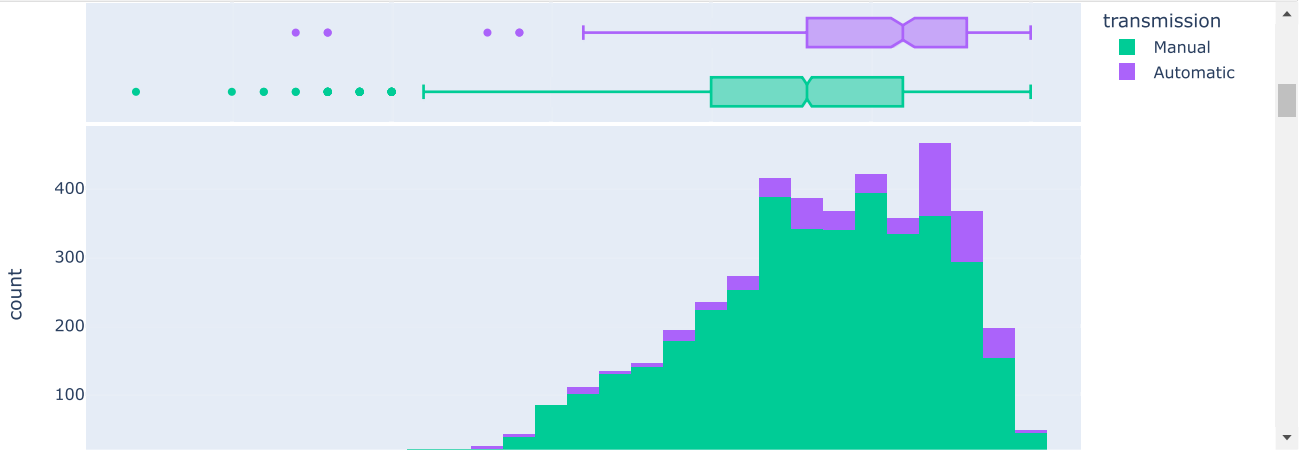
Out[2]:

	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

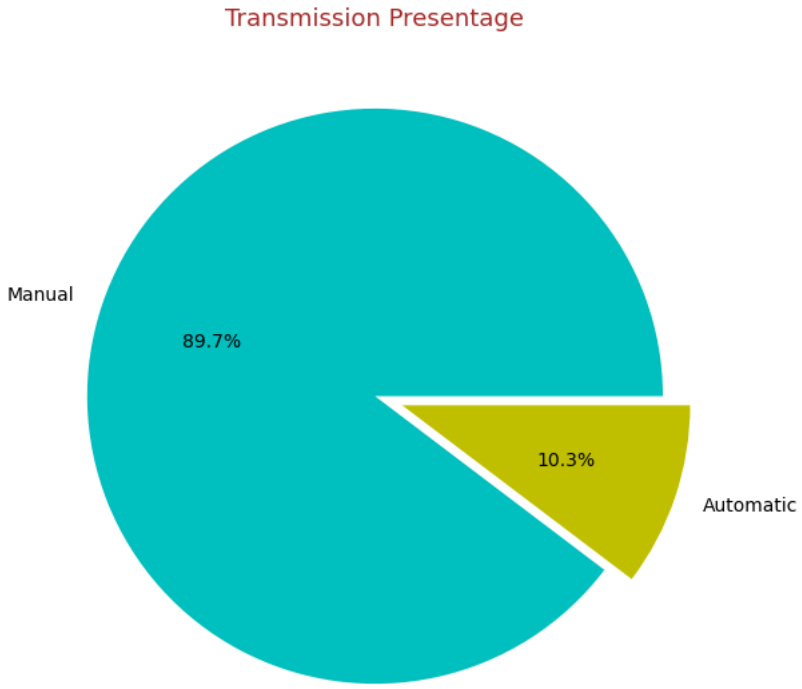
In [3]:
```
car_df.columns
```

Out[3]:
```
Index(['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
       'transmission', 'owner'],
      dtype='object')
```

In [4]:
```
col = ['name', 'year', 'selling_price', 'km_driven', 'fuel', 'seller_type',
       'transmission', 'owner']
for col in col:
    fig = px.histogram(car_df, x = col ,color = 'transmission', marginal='box' , title = col + ' vs Car Engine Transmission' , co
    fig.show()
```



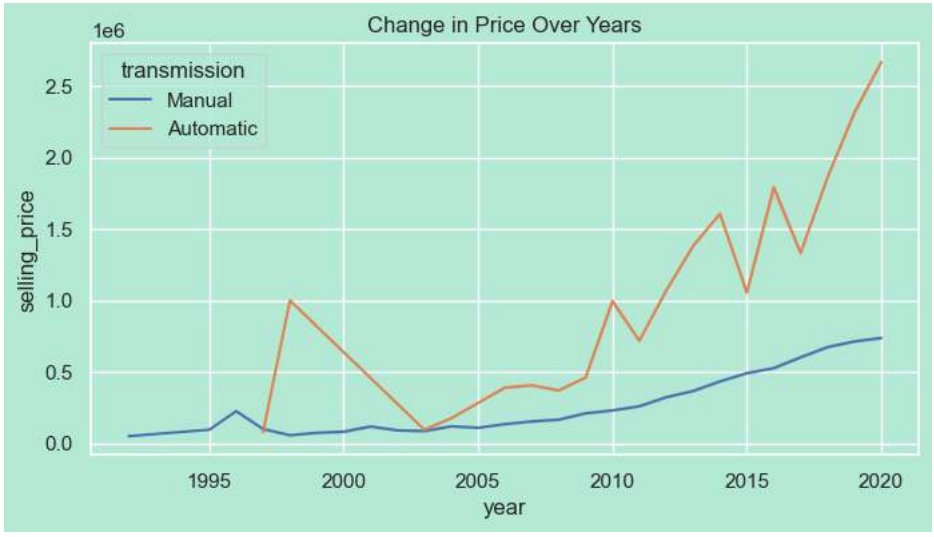
In [5]:
```
labels=car_df['transmission'].value_counts().index
colors=['c','y']
explode=[0,0.1]
values=car_df['transmission'].value_counts().values

#visualization
plt.figure(figsize=(7,7))
plt.pie(values,explode=explode,labels=labels,colors=colors,autopct='%1.1f%%')
plt.title('Transmission Presentage',color='brown',fontsize=13)
plt.show()
```



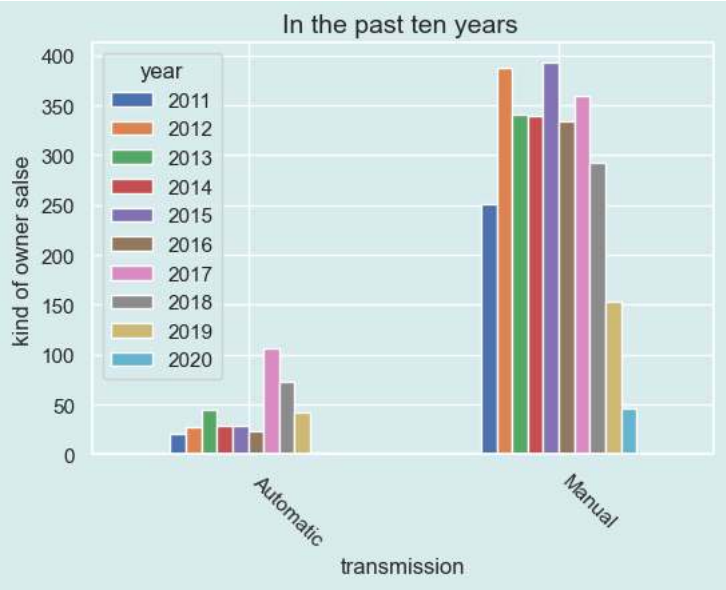
In [19]:
```
sns.set(rc={"axes.facecolor":"#b4e9d6","figure.facecolor":"#b4e9d6"})
pallet = ["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78", "#F3AB60"]
cmap = colors.ListedColormap(["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78", "#F3AB60"])
```

In [9]:
```
plt.figure(figsize=(8,4))
sns.lineplot(data=car_df, x="year", y="selling_price", hue="transmission", ci=False)
plt.title("Change in Price Over Years")
```

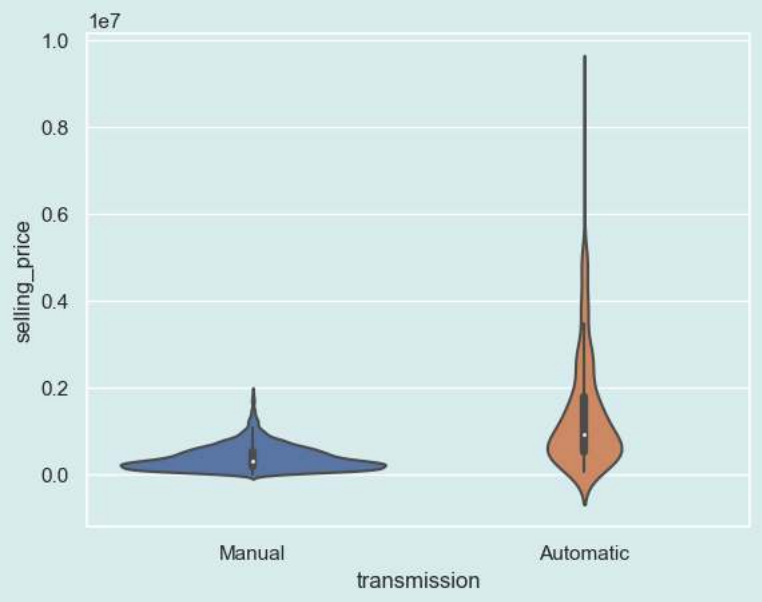
Out[9]:
```
Text(0.5, 1.0, 'Change in Price Over Years')
```



In [11]:
```
sns.set(rc={"axes.facecolor":"#d8ecee","figure.facecolor":"#d8ecee"})
pallet = ["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78", "#F3AB60"]
cmap = colors.ListedColormap(["#682F2F", "#9E726F", "#D6B2B1", "#B9C0C9", "#9F8A78", "#F3AB60"])

df_owner = car_df.query('year > 2010')
df_owner = (
    df_owner.groupby("year")["transmission"]
    .value_counts()
    .sort_index()
    .unstack()
)
ax = df_owner.T.plot(kind="bar", figsize=(6, 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()
```



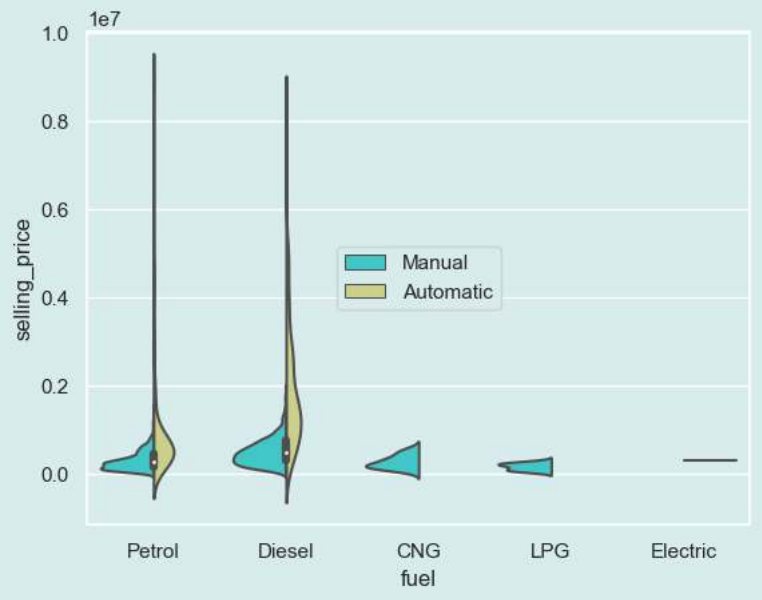
In [12]:
```
sns.violinplot(x=car_df['transmission'],y=car_df['selling_price'])
plt.show()
```



In [13]:
```
plt.figure(figsize=(14,6))
sns.violinplot(x='owner', y='selling_price',hue='transmission',data=car_df,palette='rainbow',split=True)
plt.title('Violin plot of selling Price of Car');
```

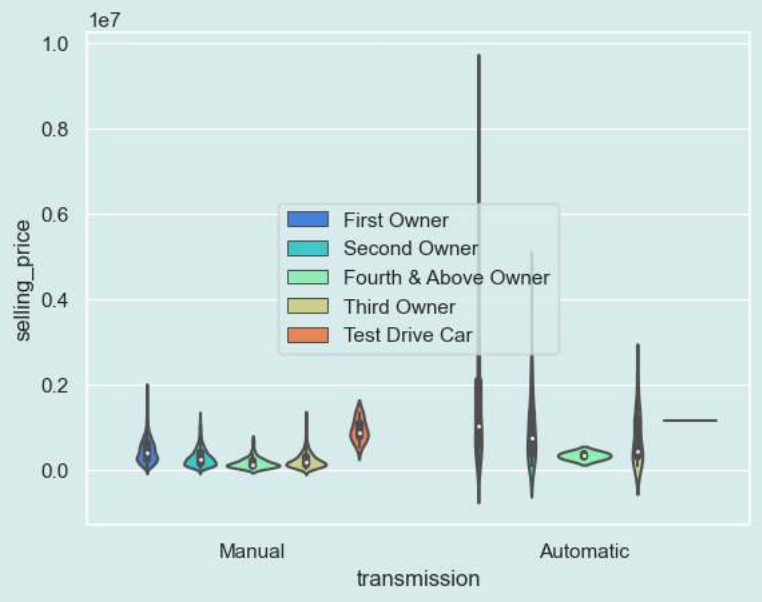
In [14]:
```
sns.violinplot(car_df['fuel'],car_df['selling_price'],
               hue=car_df['transmission'],palette='rainbow',split=True)
plt.legend(loc=10)
plt.show()
```

```
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 p
assing other arguments without an explicit keyword will result in an error or misinterpretation.
```



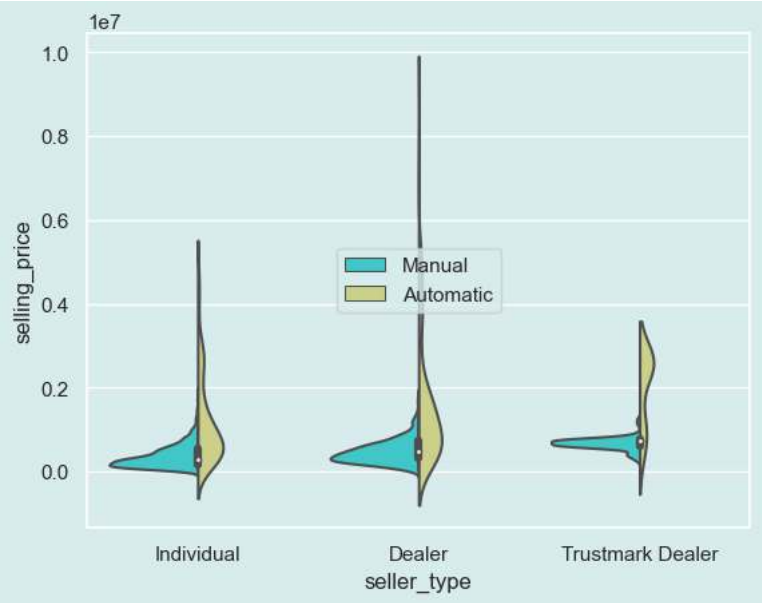
In [15]:
```
sns.violinplot(car_df['transmission'],y=car_df['selling_price'],hue=car_df['owner'],palette='rainbow')
plt.legend(loc=10)
plt.show()
```

```
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 pass
ing other arguments without an explicit keyword will result in an error or misinterpretation.
```



In [17]:
```
sns.violinplot(car_df['seller_type'],car_df['selling_price'],
               hue=car_df['transmission'],palette='rainbow',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 p
assing other arguments without an explicit keyword will result in an error or misinterpretation.
```



In [18]:
```
sns.violinplot(car_df['transmission'],y=car_df['selling_price'],hue=car_df['seller_type'],palette='rainbow')
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 pass
ing other arguments without an explicit keyword will result in an error or misinterpretation.
```