In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv('car data.csv')
df
```

Out[2]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer
4	swift	2014	4.60	6.87	42450	Diesel	Dealer
...	...	...	...	...	...	...	...
296	city	2016	9.50	11.60	33988	Diesel	Dealer
297	brio	2015	4.00	5.90	60000	Petrol	Dealer
298	city	2009	3.35	11.00	87934	Petrol	Dealer
299	city	2017	11.50	12.50	9000	Diesel	Dealer
300	brio	2016	5.30	5.90	5464	Petrol	Dealer

301 rows × 9 columns

In [3]:
```
df.dtypes
```

Out[3]:
```
Car_Name          object
Year               int64
Selling_Price    float64
Present_Price    float64
Kms_Driven         int64
Fuel_Type         object
Seller_Type       object
Transmission      object
Owner              int64
dtype: object
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['Car_Name', 'Year', 'Selling_Price', 'Present_Price', 'Kms_Driven',
       'Fuel_Type', 'Seller_Type', 'Transmission', 'Owner'],
      dtype='object')
```

In [7]:
```
df.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	
Year	301.0	2013.627907	2.891554	2003.00	2012.0	2014.0	2016.0	
Selling_Price	301.0	4.661296	5.082812	0.10	0.9	3.6	6.0	
Present_Price	301.0	7.628472	8.644115	0.32	1.2	6.4	9.9	
Kms_Driven	301.0	36947.205980	38886.883882	500.00	15000.0	32000.0	48767.0	5(
Owner	301.0	0.043189	0.247915	0.00	0.0	0.0	0.0	

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Car_Name         0
Year             0
Selling_Price    0
Present_Price    0
Kms_Driven       0
Fuel_Type        0
Seller_Type      0
Transmission     0
Owner            0
dtype: int64
```

In [10]:
```
df.head(2)
```

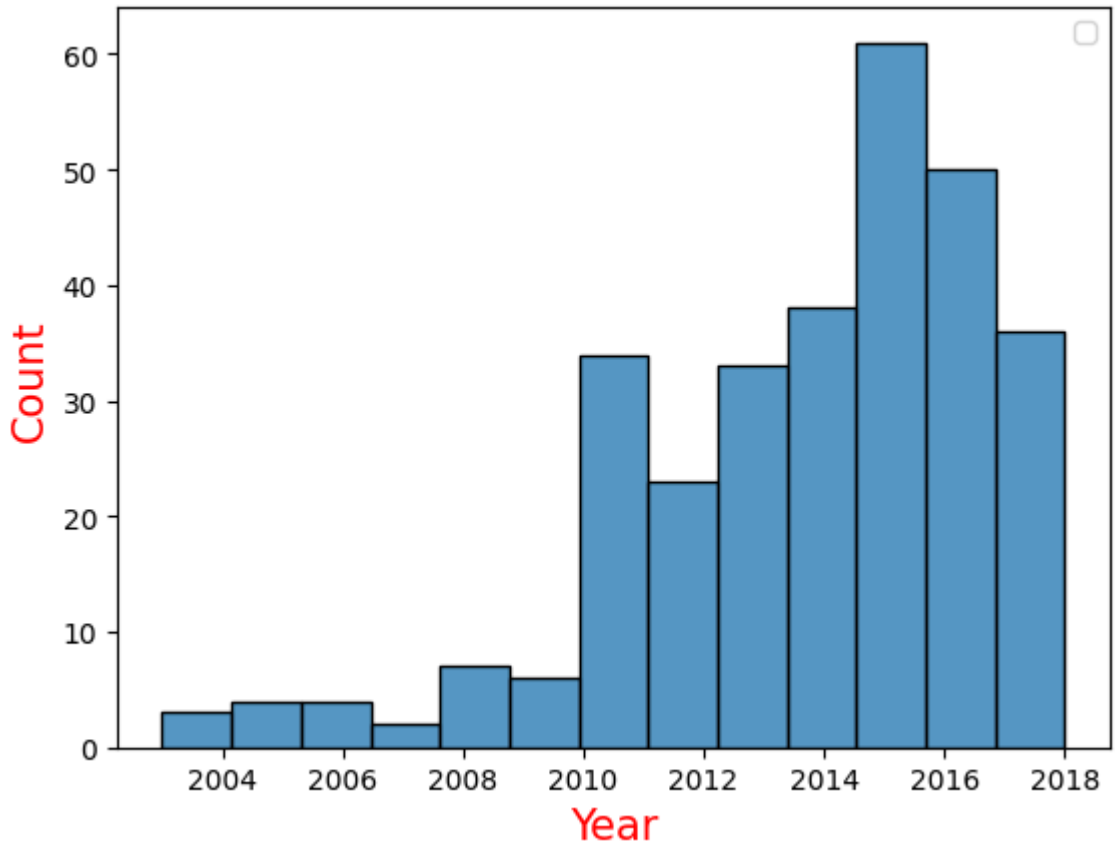
Out[10]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Tr
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	

In [40]:
```
sns.histplot(x='Year',data=df)
plt.xlabel('Year',c='r',size=15)
plt.ylabel('Count',c='r',size=15)
plt.legend()
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_5848\1485868473.py:4: UserWarning: No ar
tists with labels found to put in legend.  Note that artists whose label start wi
th an underscore are ignored when legend() is called with no argument.
  plt.legend()
```

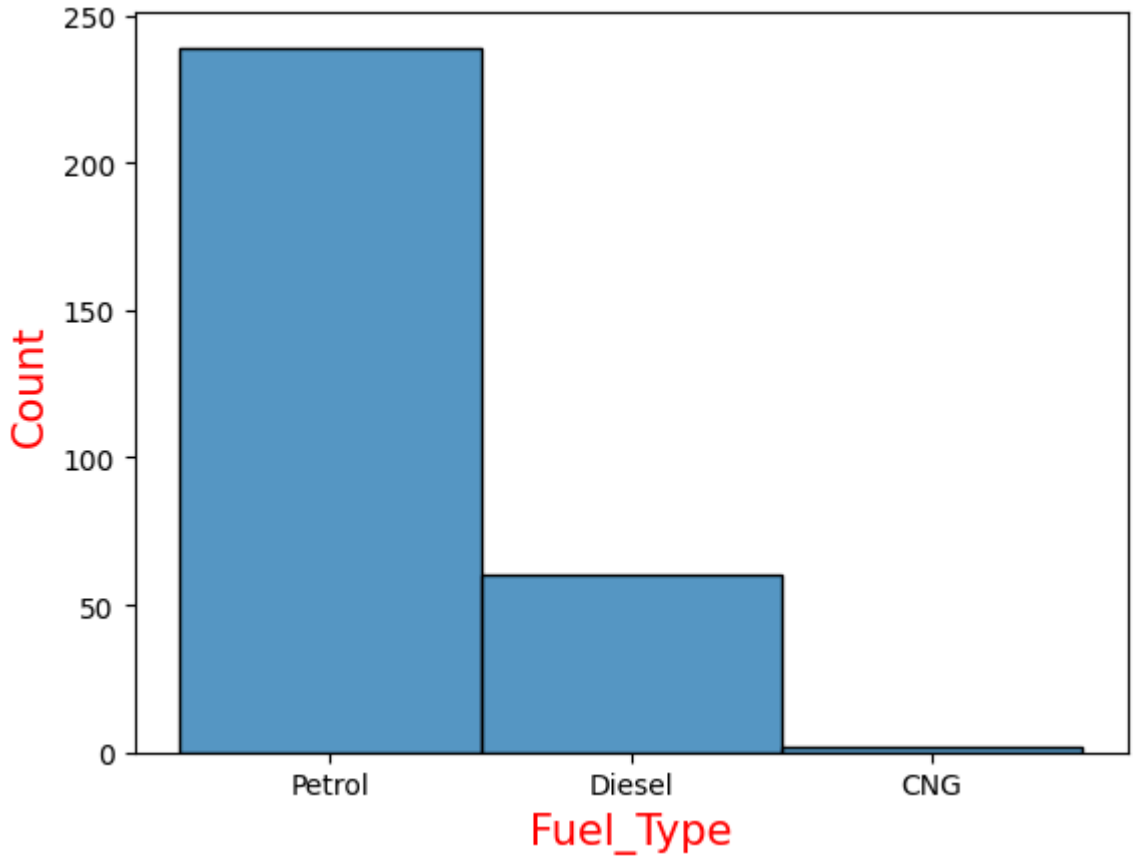
Out[40]:
```
<matplotlib.legend.Legend at 0x2bdcf55f610>
```



In [41]:
```
sns.histplot(x='Fuel_Type',data=df,palette="Set1")
plt.xlabel('Fuel_Type',c='r',size=15)
plt.ylabel('Count',c='r',size=15)
```

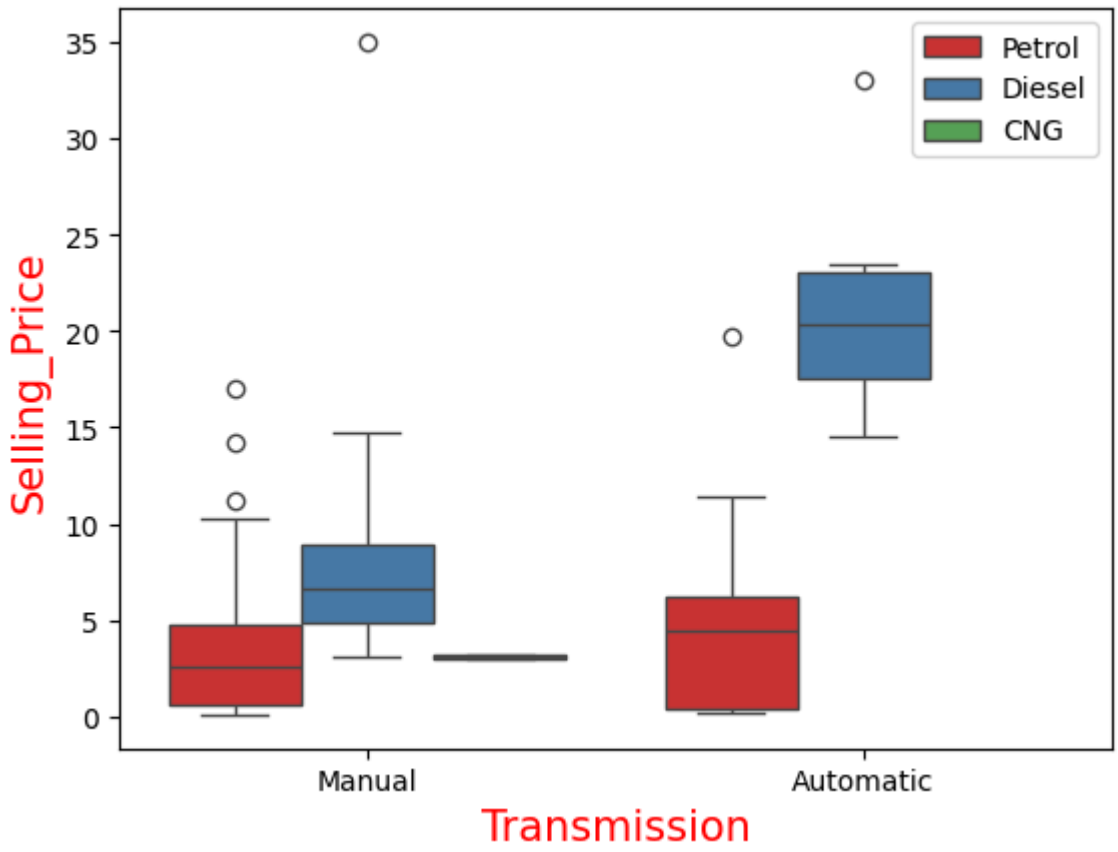
```
C:\Users\hp\AppData\Local\Temp\ipykernel_5848\887738274.py:1: UserWarning: Ignori
ng `palette` because no `hue` variable has been assigned.
  sns.histplot(x='Fuel_Type',data=df,palette="Set1")
```

Out[41]:
```
Text(0, 0.5, 'Count')
```



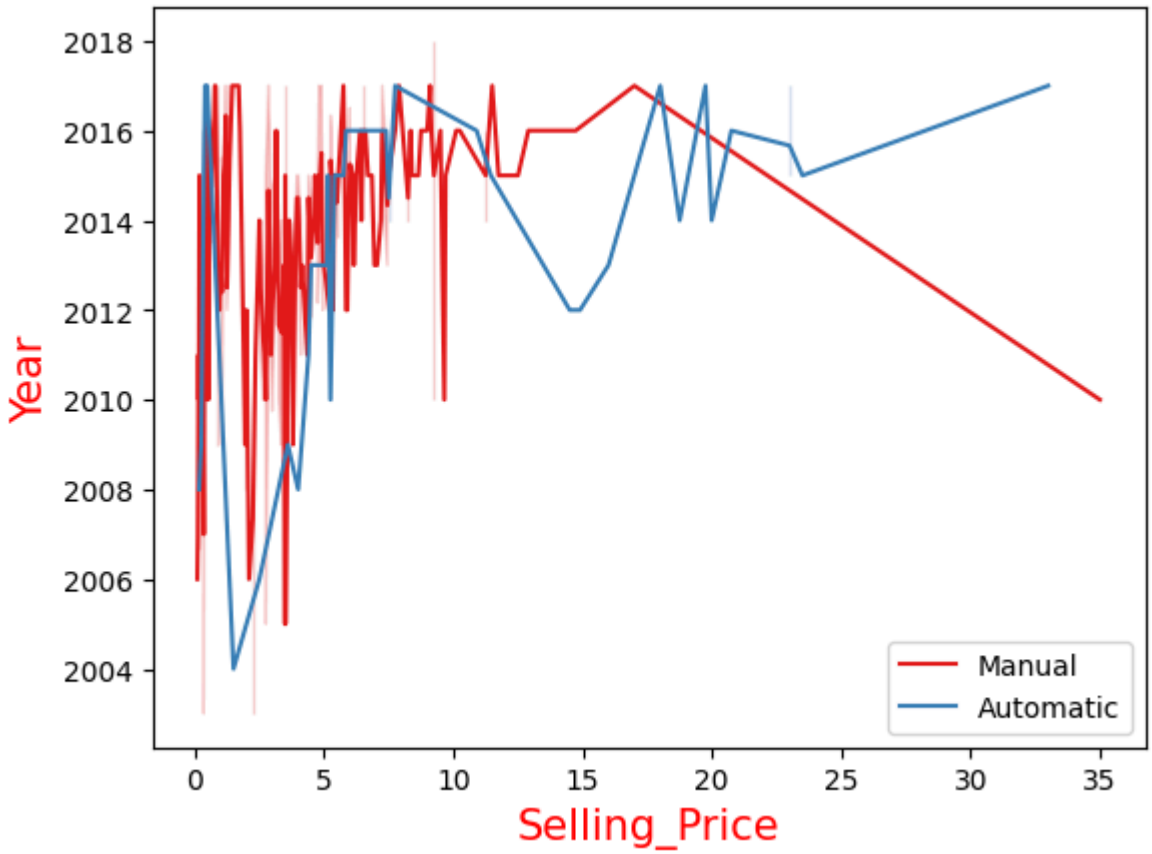
In [42]:
```
sns.boxplot(y='Selling_Price',x='Transmission',hue='Fuel_Type',data=df,palette="
plt.xlabel('Transmission',c='r',size=15)
plt.ylabel('Selling_Price',c='r',size=15)
plt.legend()
```

Out[42]: <matplotlib.legend.Legend at 0x2bdd067bd90>

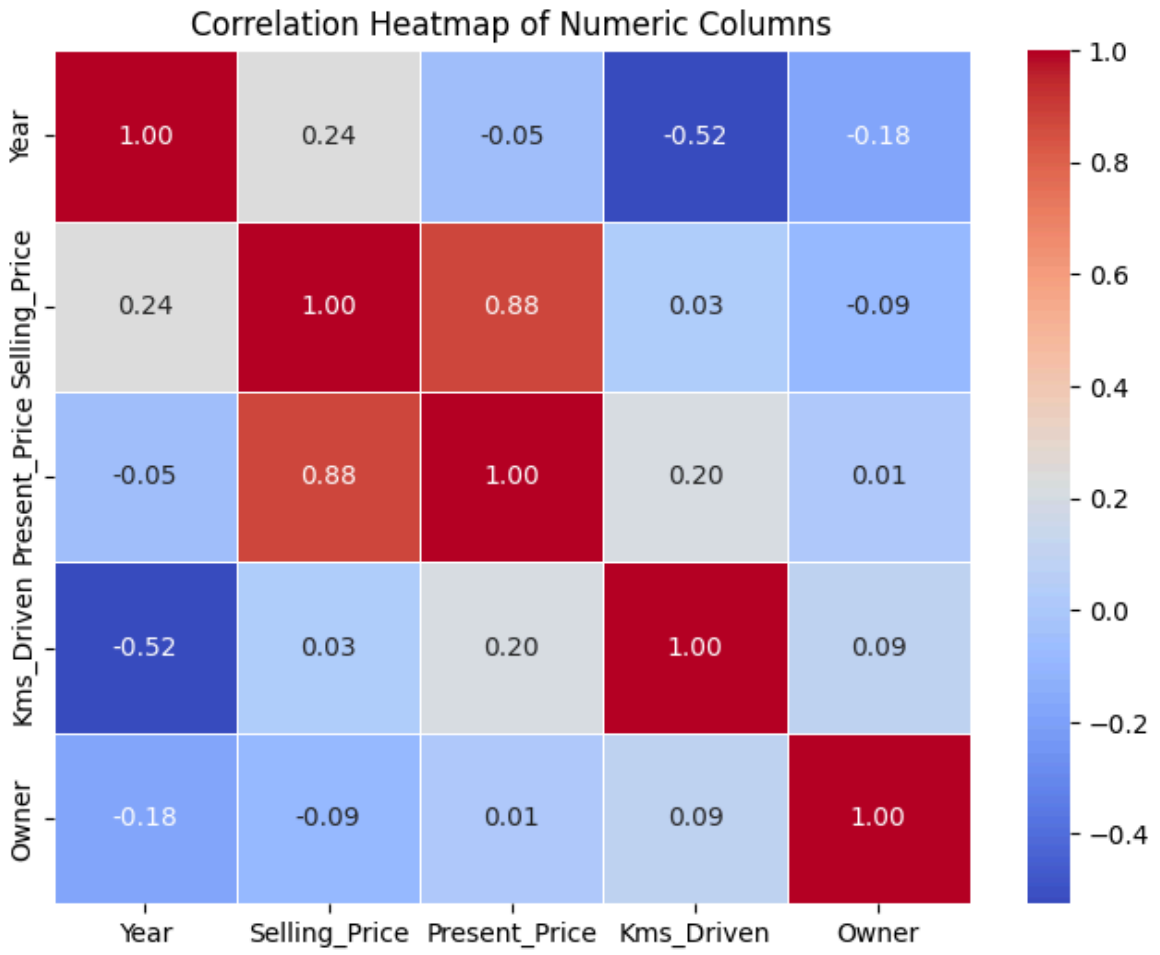


In [49]:
```
sns.lineplot(x='Selling_Price',y='Year',hue='Transmission',data=df,palette="Set1
plt.xlabel('Selling_Price',c='r',size=15)
plt.ylabel('Year',c='r',size=15)
plt.legend()
```

Out[49]: <matplotlib.legend.Legend at 0x2bdd2dcc410>



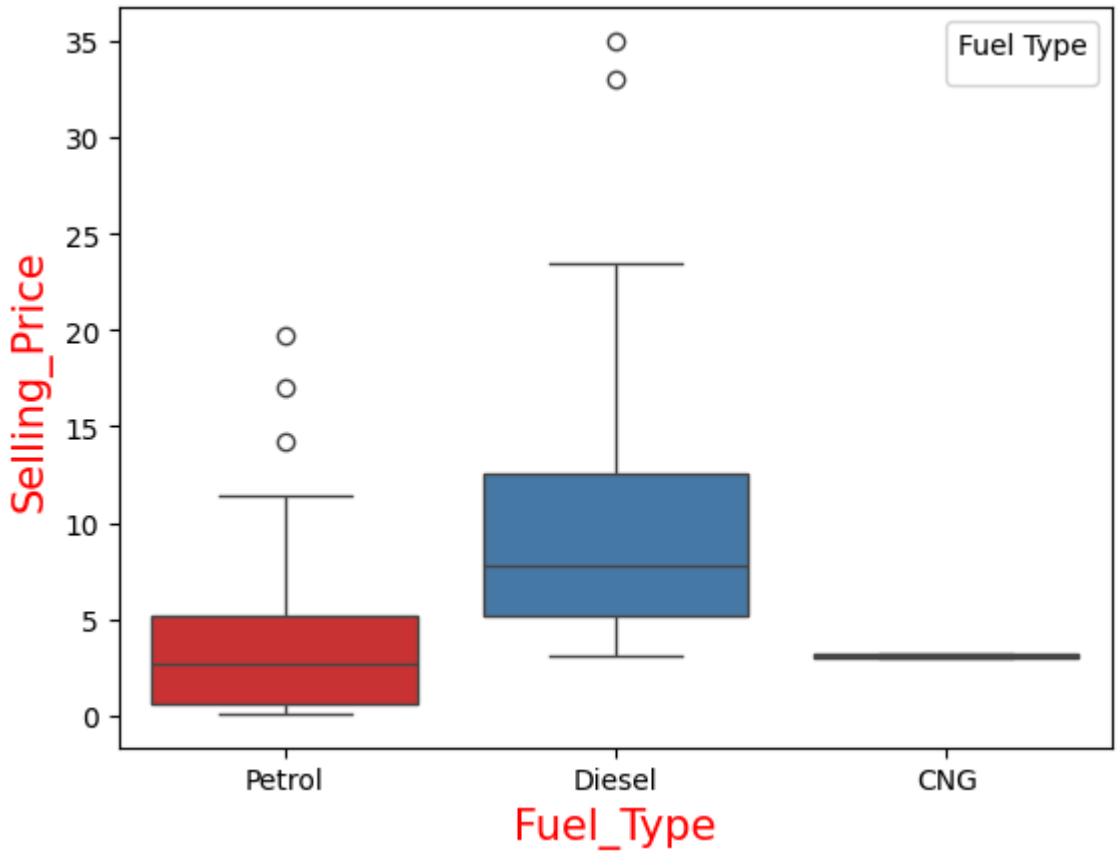
In [51]:
```
numeric_cols = ["Year", "Selling_Price", "Present_Price", "Kms_Driven", "Owner"]

# Correlation matrix
corr_matrix = df[numeric_cols].corr()

# Plot heatmap
plt.figure(figsize=(8,6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", linewidths=0.5, fmt=".2f")
plt.title("Correlation Heatmap of Numeric Columns")
plt.show()
```



In [88]:
```
sns.boxplot(x='Fuel_Type',y='Selling_Price',legend=True,data=df,palette="Set1")
plt.xlabel('Fuel_Type',c='r',size=15)
plt.ylabel('Selling_Price',c='r',size=15)
plt.legend(title="Fuel Type")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_5848\1310071511.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.boxplot(x='Fuel_Type',y='Selling_Price',legend=True,data=df,palette="Set1")
C:\Users\hp\AppData\Local\Temp\ipykernel_5848\1310071511.py:4: UserWarning: No ar
tists with labels found to put in legend.  Note that artists whose label start wi
th an underscore are ignored when legend() is called with no argument.
  plt.legend(title="Fuel Type")
```

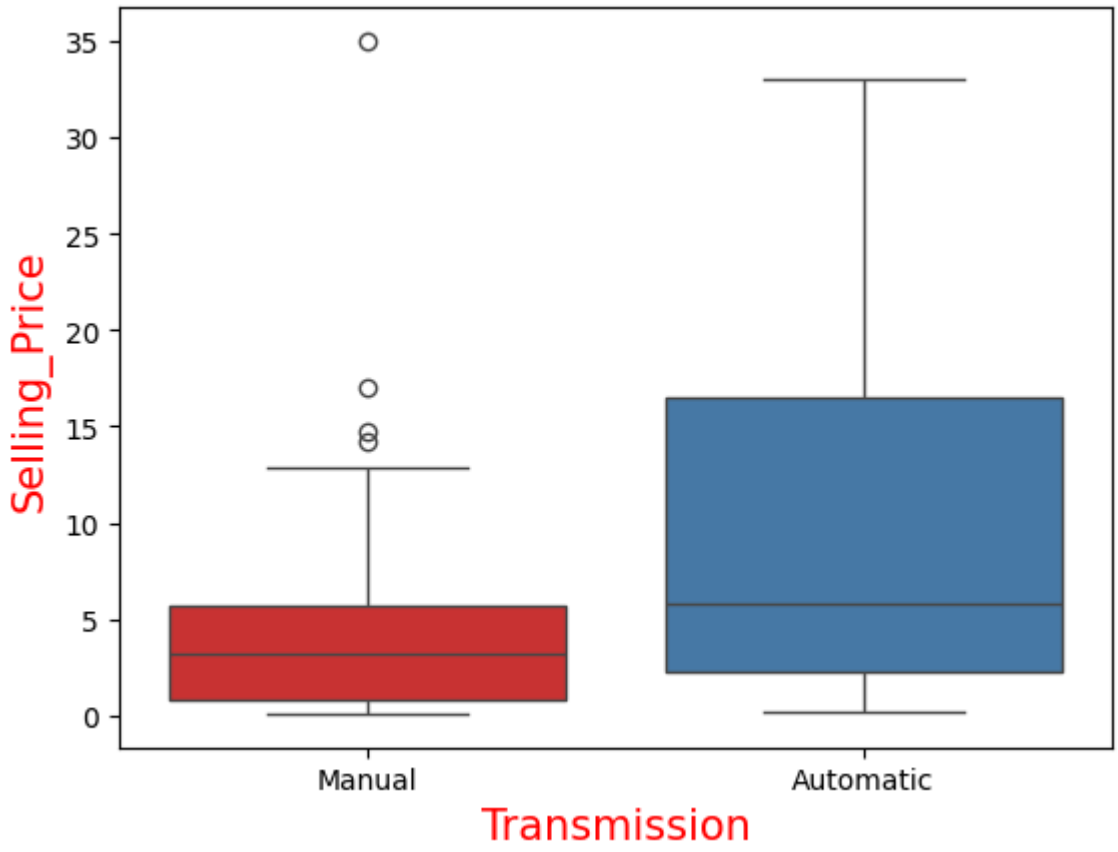
Out[88]:
```
<matplotlib.legend.Legend at 0x2bddad6a0d0>
```



In [89]:
```
sns.boxplot(x='Transmission',y='Selling_Price',legend=True,data=df,palette="Set1
plt.xlabel('Transmission',c='r',size=15)
plt.ylabel('Selling_Price',c='r',size=15)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_5848\2713549537.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.boxplot(x='Transmission',y='Selling_Price',legend=True,data=df,palette="Set
1")
```

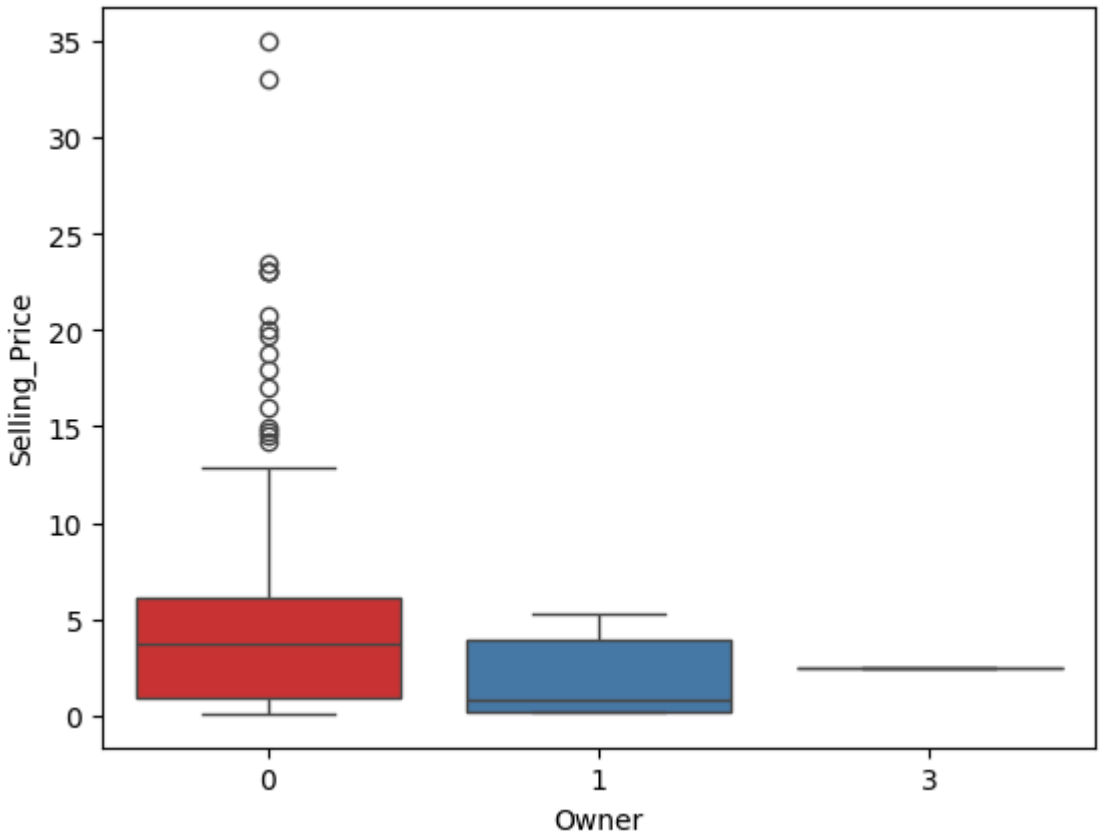
Out[89]:
```
Text(0, 0.5, 'Selling_Price')
```



In [90]:
```
sns.boxplot(x='Owner',y='Selling_Price',legend=True,data=df,palette="Set1")
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_5848\1477934305.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.boxplot(x='Owner',y='Selling_Price',legend=True,data=df,palette="Set1")
```

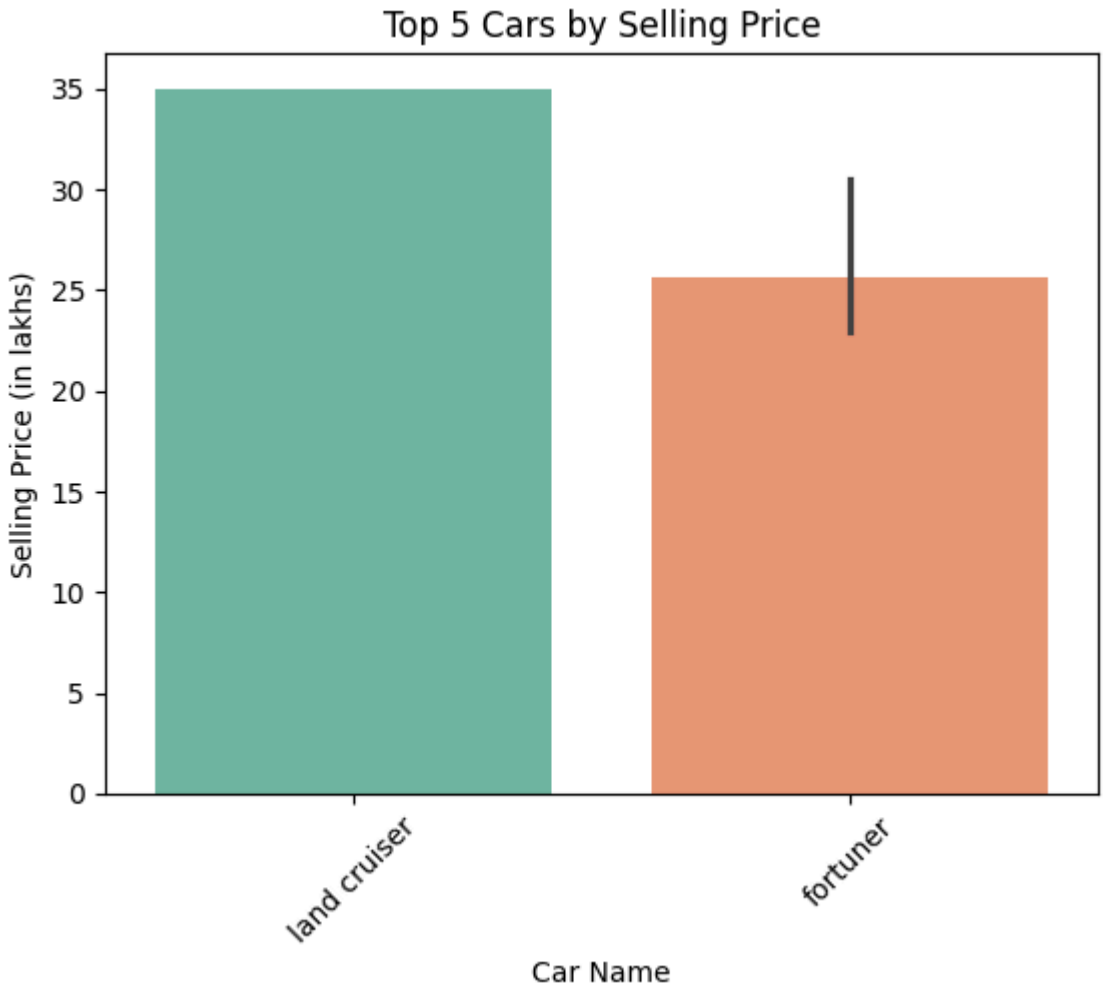
Out[90]:
```
<Axes: xlabel='Owner', ylabel='Selling_Price'>
```



In [91]:
```
top5_cars = df[['Car_Name', 'Selling_Price']].sort_values(by='Selling_Price', as
print(top5_cars)
sns.barplot(x='Car_Name', y='Selling_Price',legend=True, data=top5_cars, palette
plt.title('Top 5 Cars by Selling Price')
plt.ylabel('Selling Price (in lakhs)')
plt.xlabel('Car Name')
plt.xticks(rotation=45)
plt.show()
```

```
        Car_Name  Selling_Price
86  land cruiser           35.0
64      fortuner           33.0
63      fortuner           23.5
51      fortuner           23.0
93      fortuner           23.0
```

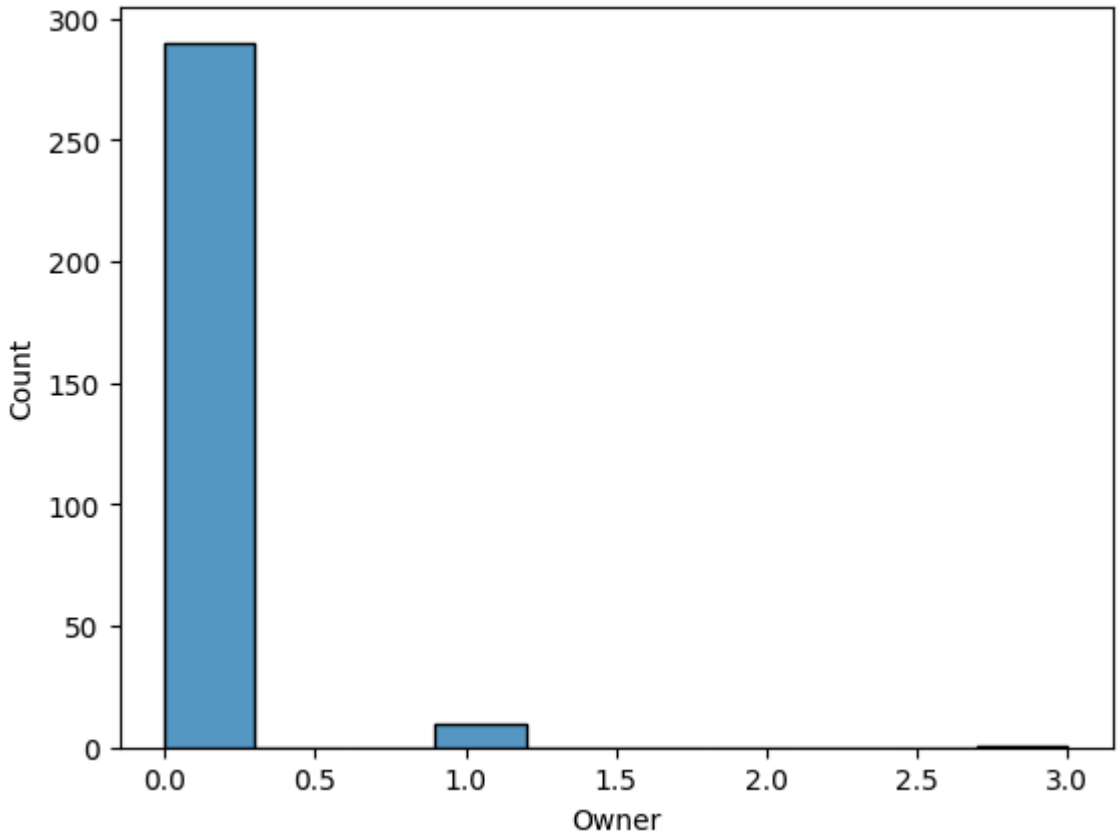
```
C:\Users\hp\AppData\Local\Temp\ipykernel_5848\2580649709.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x='Car_Name', y='Selling_Price',legend=True, data=top5_cars, palett
e='Set2')
```



In [92]:
```
sns.histplot(x='Owner',data=df,palette="Set1")
```

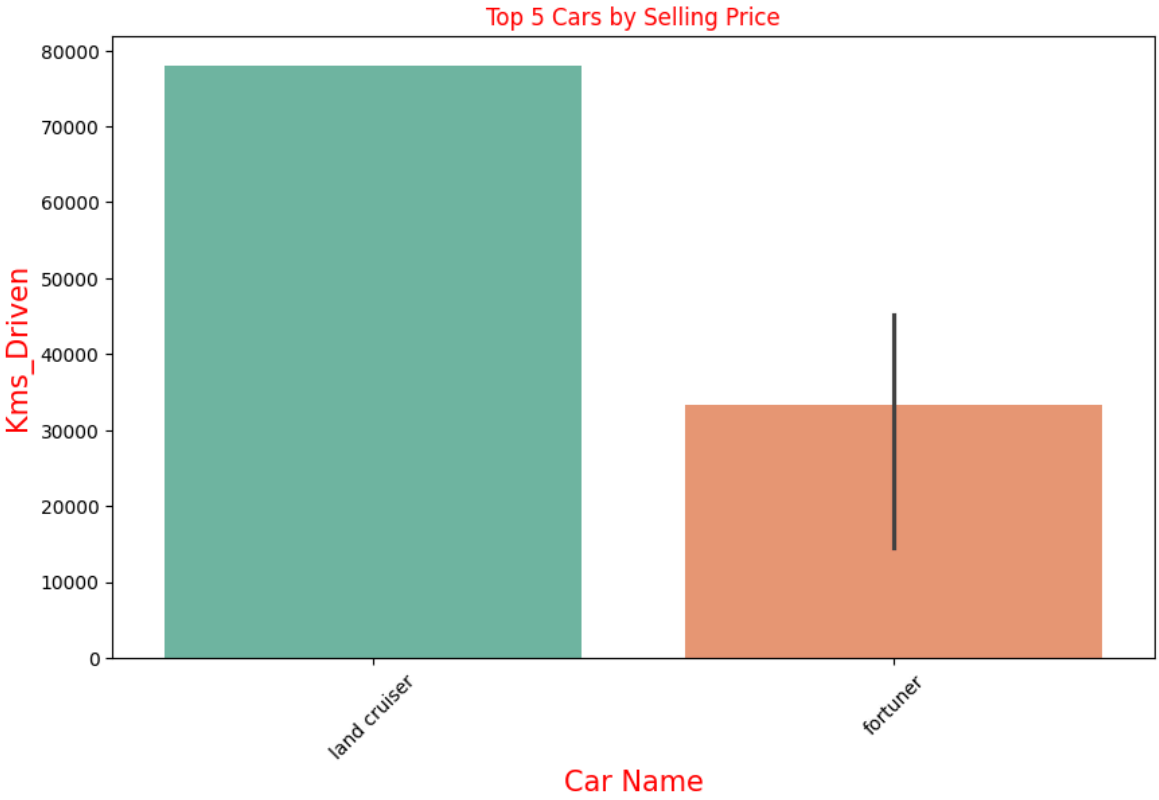
```
C:\Users\hp\AppData\Local\Temp\ipykernel_5848\4240418213.py:1: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
  sns.histplot(x='Owner',data=df,palette="Set1")
```

Out[92]:
```
<Axes: xlabel='Owner', ylabel='Count'>
```



In [111…

```
top5_cars = df[['Car_Name', 'Selling_Price', 'Kms_Driven']].sort_values(by='Sell

# Scatter plot: Kms_Driven vs Selling_Price
plt.figure(figsize=(10,6))
sns.barplot(x='Car_Name', y='Kms_Driven', data=top5_cars,hue='Car_Name',palette=
plt.title('Top 5 Cars Selling Price (Kms Driven not shown)',c='r',size=15)
plt.title('Top 5 Cars by Selling Price')
plt.ylabel('Kms_Driven',c='r',size=15)
plt.xlabel('Car Name',c='r',size=15)
plt.xticks(rotation=45)
plt.show()
plt.show()
```


Top 5 Cars by Selling Price
Kms_Driven
80000
70000
60000
50000
40000
30000
20000
10000
0
land cruiser
fortuner
Car Name