In [117]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder
```

In [4]:
```
df=pd.read_csv('Car details v3.csv')
df.head()
```

Out[4]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.4 kmpl
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14 kmpl
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.7 kmpl
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.0 kmpl
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.1 kmpl

In [5]:
```
df.describe()
```

Out[5]:

	year	selling_price	km_driven	seats
count	8128.000000	8.128000e+03	8.128000e+03	7907.000000
mean	2013.804011	6.382718e+05	6.981951e+04	5.416719
std	4.044249	8.062534e+05	5.655055e+04	0.959588
min	1983.000000	2.999900e+04	1.000000e+00	2.000000
25%	2011.000000	2.549990e+05	3.500000e+04	5.000000
50%	2015.000000	4.500000e+05	6.000000e+04	5.000000
75%	2017.000000	6.750000e+05	9.800000e+04	5.000000
max	2020.000000	1.000000e+07	2.360457e+06	14.000000

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           8128 non-null   object
 5   seller_type    8128 non-null   object
 6   transmission   8128 non-null   object
 7   owner          8128 non-null   object
 8   mileage        7907 non-null   object
 9   engine         7907 non-null   object
 10  max_power      7913 non-null   object
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 825.6+ KB
```

In [7]:
```
df1=pd.read_csv('car dekho.csv')
df1.head()
```

Out[7]:

	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 [8]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4340 entries, 0 to 4339
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   name           4340 non-null   object
 1   year           4340 non-null   int64 
 2   selling_price  4340 non-null   int64 
 3   km_driven      4340 non-null   int64 
 4   fuel           4340 non-null   object
 5   seller_type    4340 non-null   object
 6   transmission   4340 non-null   object
 7   owner          4340 non-null   object
dtypes: int64(3), object(5)
memory usage: 271.4+ KB
```

In [9]:
```
df1.describe()
```

Out[9]:

	year	selling_price	km_driven
count	4340.000000	4.340000e+03	4340.000000
mean	2013.090783	5.041273e+05	66215.777419
std	4.215344	5.785487e+05	46644.102194
min	1992.000000	2.000000e+04	1.000000
25%	2011.000000	2.087498e+05	35000.000000
50%	2014.000000	3.500000e+05	60000.000000
75%	2016.000000	6.000000e+05	90000.000000
max	2020.000000	8.900000e+06	806599.000000

In [10]:
```
df2=pd.read_csv('car data.csv')
df2.head()
```

Out[10]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmis
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Ma
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Ma
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Ma
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Ma
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Ma

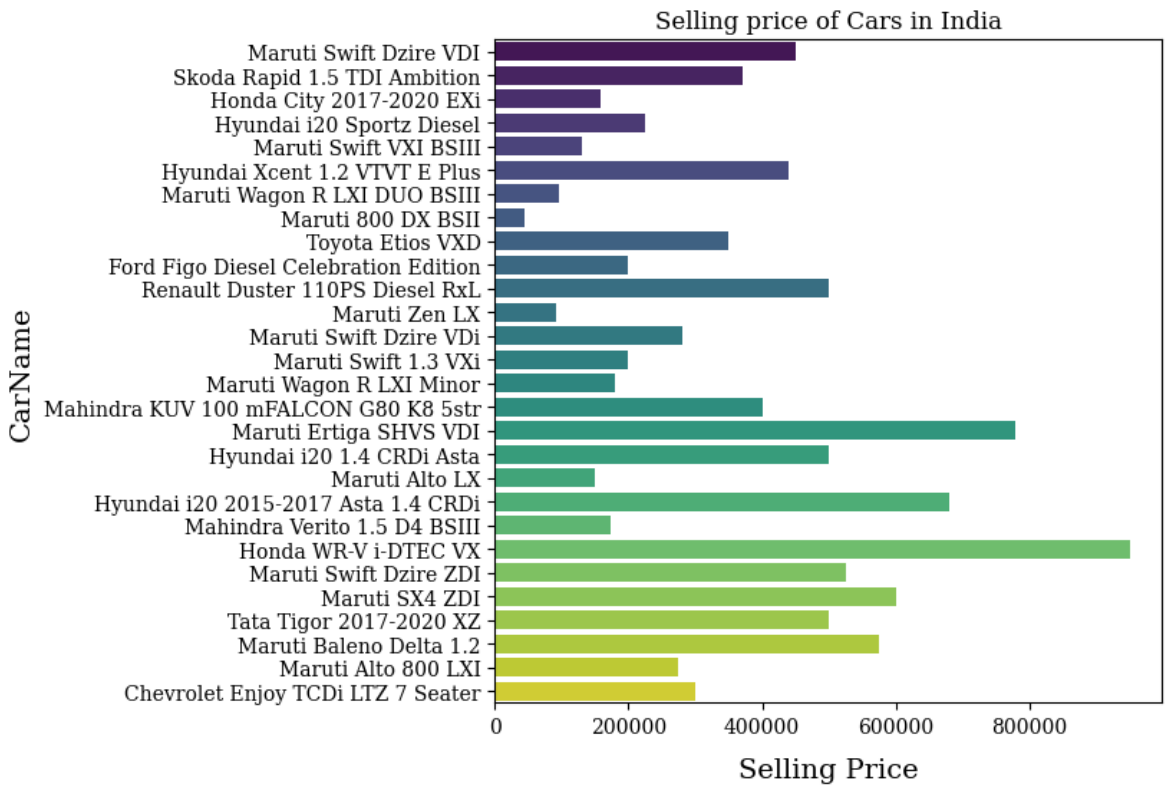
In [11]:
```
print('df shape: ',df.shape)
print('df1 shape: ',df1.shape)
print('df2 shape: ',df2.shape)
```

```
df shape:  (8128, 13)
df1 shape:  (4340, 8)
df2 shape:  (301, 9)
```

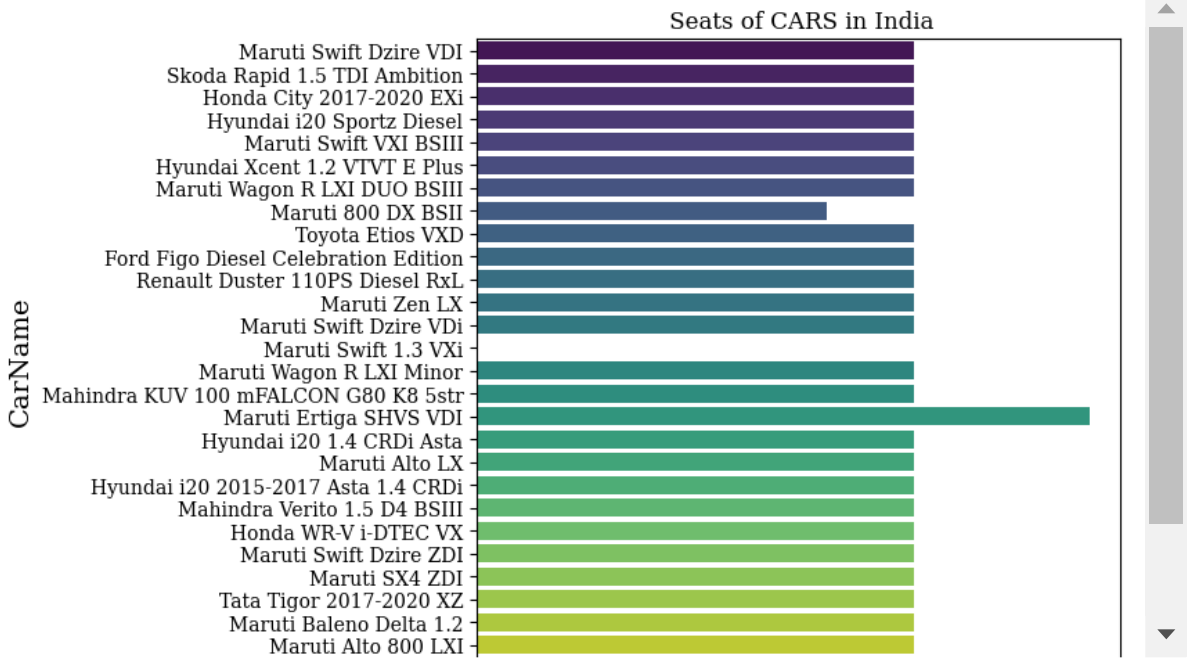
In [37]:
```
subset=df[:28]

print(subset)
```

```
8   Diesel  Individual       Manual   First Owner   23.59 kmpl  1364 CC
9   Diesel  Individual       Manual   First Owner    20.0 kmpl  1399 CC
10  Diesel  Individual       Manual  Second Owner   19.01 kmpl  1461 CC
11  Petrol  Individual       Manual  Second Owner    17.3 kmpl   993 CC
12  Diesel  Individual       Manual  Second Owner    19.3 kmpl  1248 CC
13  Petrol  Individual       Manual  Second Owner         NaN       NaN
14  Petrol  Individual       Manual  Second Owner    18.9 kmpl  1061 CC
15  Petrol  Individual       Manual   First Owner   18.15 kmpl  1198 CC
16  Diesel  Individual       Manual  Second Owner   24.52 kmpl  1248 CC
17  Diesel  Individual       Manual  Second Owner    23.0 kmpl  1396 CC
18  Petrol  Individual       Manual  Second Owner    19.7 kmpl   796 CC
19  Diesel  Individual       Manual   First Owner   22.54 kmpl  1396 CC
20  Diesel  Individual       Manual  Second Owner    21.0 kmpl  1461 CC
21  Diesel  Individual       Manual   First Owner    25.5 kmpl  1498 CC
22  Diesel  Individual       Manual   First Owner   26.59 kmpl  1248 CC
23  Diesel  Individual       Manual   First Owner    21.5 kmpl  1248 CC
24  Petrol  Individual       Manual   First Owner    20.3 kmpl  1199 CC
25  Petrol  Individual       Manual   First Owner    21.4 kmpl  1197 CC
26  Petrol  Individual       Manual   First Owner    24.7 kmpl   796 CC
27  Diesel  Individual       Manual   First Owner    18.2 kmpl  1248 CC
```

In [38]:
```
plt.figure(figsize=(6,6))
sns.barplot(data=subset, y='name',x='selling_price',palette='viridis')
plt.ylabel('CarName',fontsize=14,family='serif')
plt.xlabel('Selling Price',family='serif',fontsize=14,labelpad=10)
plt.xticks(family='serif')
plt.yticks(family='serif')
plt.title(label='Selling price of Cars in India',weight=200,family='serif')
plt.show()
```



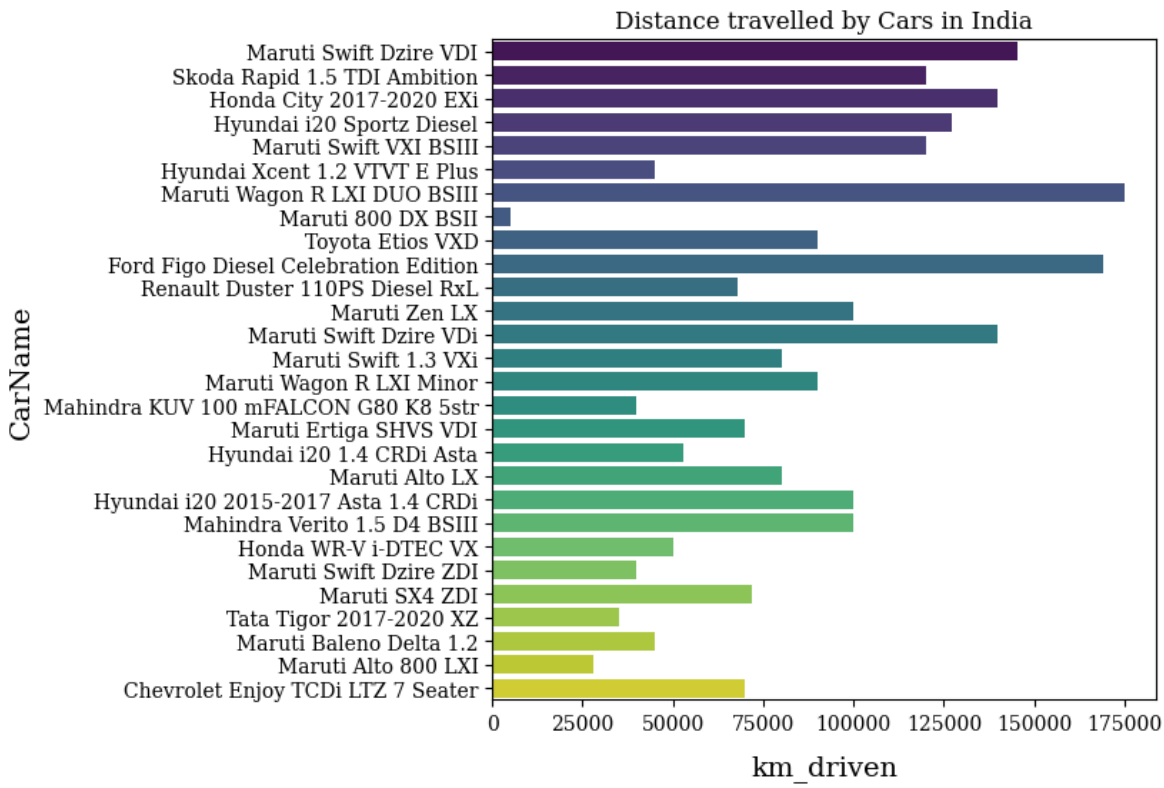
In [42]:
```
plt.figure(figsize=(6,6))
sns.barplot(data=subset, y='name',x='seats',palette='viridis')
plt.ylabel('CarName',fontsize=14,family='serif')
plt.xlabel('Seats',family='serif',fontsize=14,labelpad=10)
plt.xticks(family='serif')
plt.yticks(family='serif')
plt.title(label=' Seats of CARS in India',weight=200,family='serif')
plt.show()
```



In [46]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           8128 non-null   object
 5   seller_type    8128 non-null   object
 6   transmission   8128 non-null   object
 7   owner          8128 non-null   object
 8   mileage        7907 non-null   object
 9   engine         7907 non-null   object
 10  max_power      7913 non-null   object
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 825.6+ KB
```

In [52]:
```
plt.figure(figsize=(6,6))
sns.barplot(data=subset, y='name',x='km_driven',palette='viridis')
plt.ylabel('CarName',fontsize=14,family='serif')
plt.xlabel('km_driven',family='serif',fontsize=14,labelpad=10)
plt.xticks(family='serif')
plt.yticks(family='serif')
plt.title(label='Distance travelled by Cars in India',weight=200,family='se
plt.show()
```



In [54]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4340 entries, 0 to 4339
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           4340 non-null   object
 1   year           4340 non-null   int64
 2   selling_price  4340 non-null   int64
 3   km_driven      4340 non-null   int64
 4   fuel           4340 non-null   object
 5   seller_type    4340 non-null   object
 6   transmission   4340 non-null   object
 7   owner          4340 non-null   object
dtypes: int64(3), object(5)
memory usage: 271.4+ KB
```

In [57]:
```
df2.head()
```

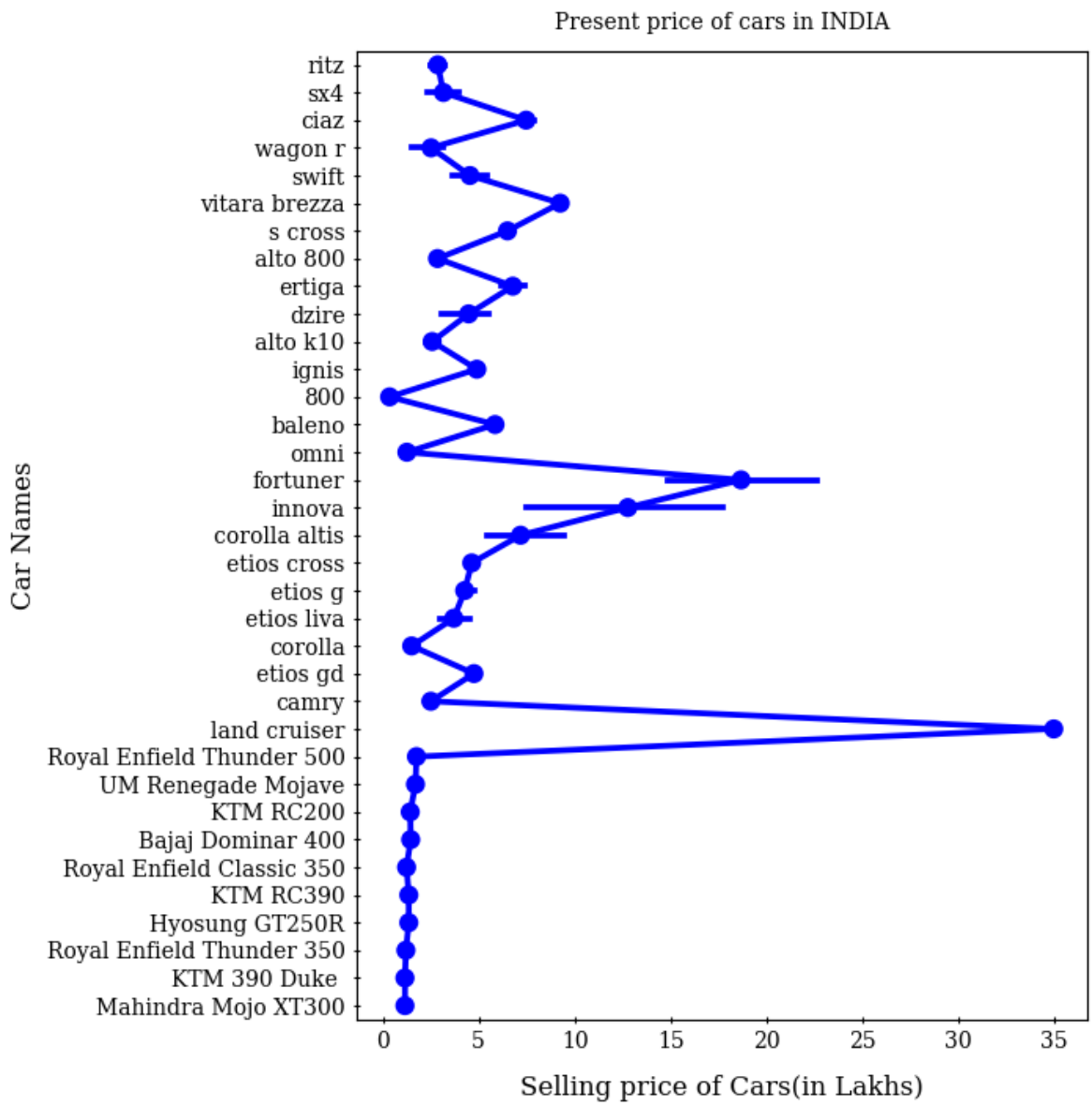
Out[57]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmis
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Ma
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Ma
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Ma
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Ma
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Ma

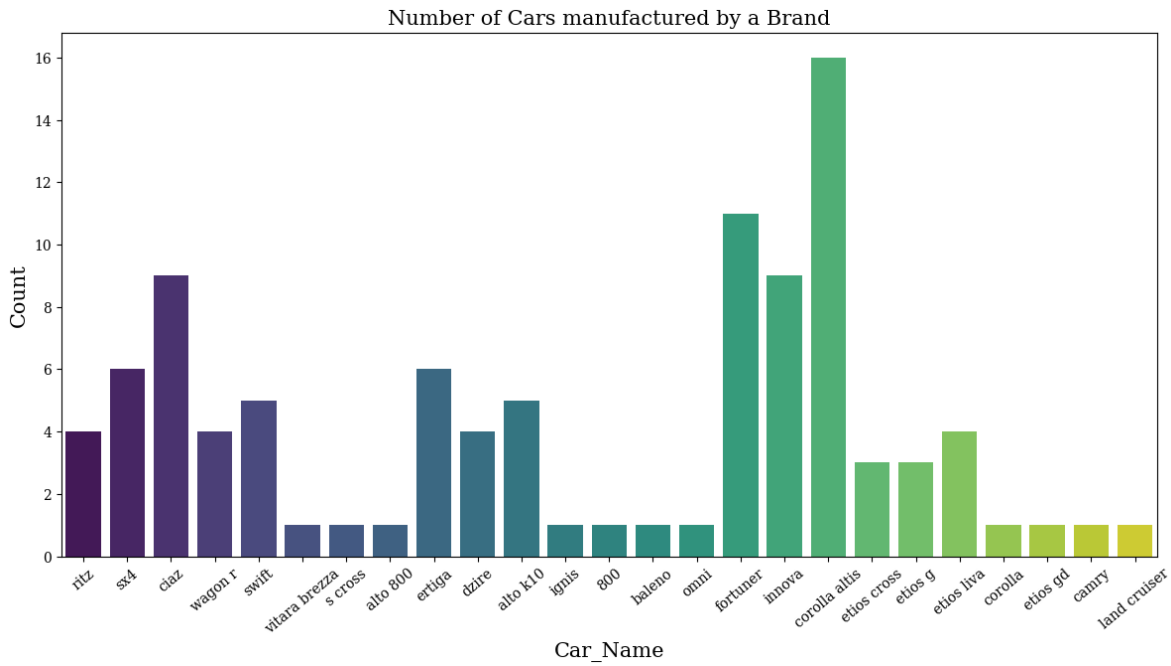
In [84]:
```
subset=df2[:100]

print(subset)
```

```
         Car_Name  Year  Selling_Price  Present_Price  Kms_Driven Fuel_Ty
pe  \
0            ritz  2014           3.35           5.59       27000    Petr
ol
1             sx4  2013           4.75           9.54       43000    Dies
el
2            ciaz  2017           7.25           9.85        6900    Petr
ol
3         wagon r  2011           2.85           4.15        5200    Petr
ol
4           swift  2014           4.60           6.87       42450    Dies
el
..            ...   ...            ...            ...         ...
...
95  corolla altis  2012           5.85          18.61       72000    Petr
ol
96         innova  2016          20.75          25.39       29000    Dies
el
97  corolla altis  2017          17.00          18.64        8700    Petr
ol
98  corolla altis  2013           7.05          18.61       45000    Petr
ol
99       fortuner  2010           9.65          20.45       50024    Dies
el

   Seller_Type Transmission  Owner
0       Dealer       Manual      0
1       Dealer       Manual      0
2       Dealer       Manual      0
3       Dealer       Manual      0
4       Dealer       Manual      0
..         ...          ...    ...
95      Dealer       Manual      0
96      Dealer    Automatic      0
97      Dealer       Manual      0
98      Dealer       Manual      0
99      Dealer       Manual      0

[100 rows x 9 columns]
```

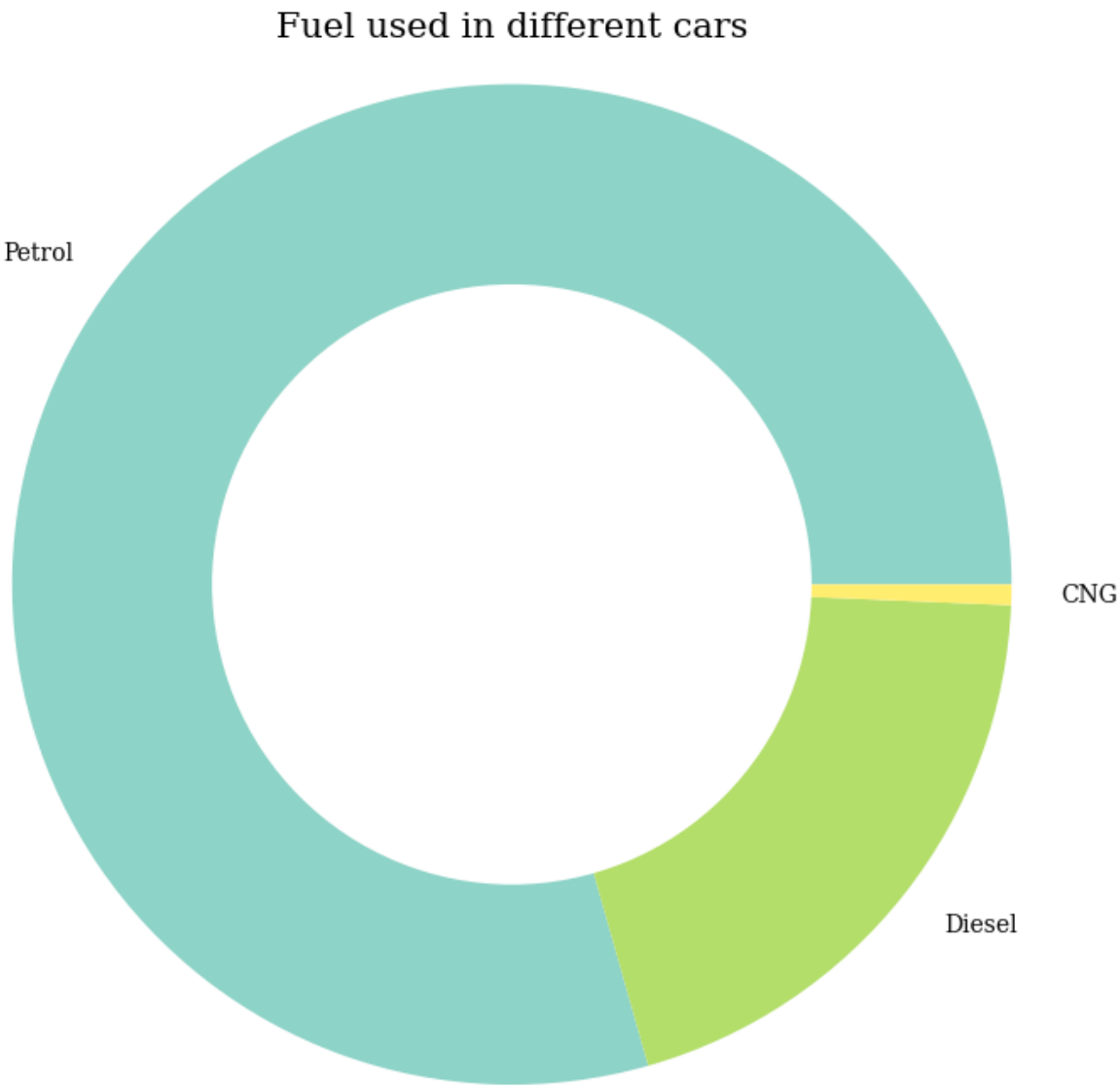
In [81]:
```
plt.figure(figsize=(6,8))
sns.pointplot(data=subset,y='Car_Name',x='Selling_Price',color='blue')
plt.xlabel('Selling price of Cars(in Lakhs)',family='serif',size=12,labelpa
plt.ylabel('Car Names',family='serif',size=12)
plt.tick_params(direction='inout')
plt.xticks(family='serif',size=10)
plt.yticks(family='serif',size=10)
plt.title(label='Present price of cars in INDIA',weight=200,family='serif',
plt.show()
```



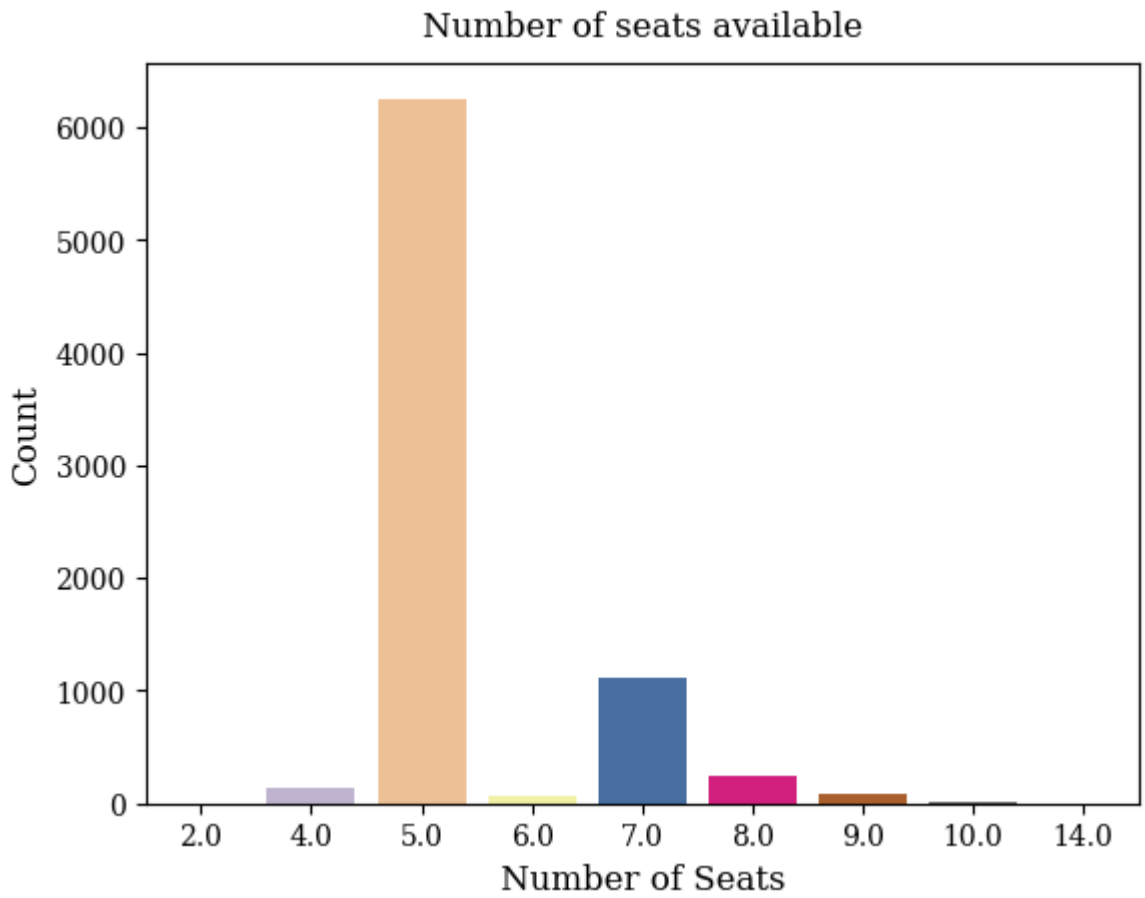
In [86]:
```
sns.catplot(data=subset,x='Car_Name',kind='count',palette='viridis',height=
sns.despine(right=False,top=False)
plt.tick_params(axis='x',rotation=40)
plt.xlabel('Car_Name',family='serif',size=15)
plt.ylabel('Count',family='serif',size=15)
plt.xticks(family='serif')
plt.yticks(family='serif')
plt.title('Number of Cars manufactured by a Brand',family='serif',size=15)
plt.show()
```



In [98]:
```
x=df2['Fuel_Type'].value_counts().plot.pie(radius=2,cmap='Set3',startangle=
plt.pie(x=[1],radius=1.2,colors='white')
plt.title(label='Fuel used in different cars',family='serif',size=15,pad=10
plt.ylabel('')
plt.show()
```



In [107]:
```
sns.countplot(data=df,x='seats',palette='Accent')
plt.xlabel('Number of Seats', family='serif',size=12)
plt.ylabel('Count',family='serif',size=12)
plt.xticks(family='serif')
plt.yticks(family='serif')
plt.title(label='Number of seats available',family='serif',size=12,pad=10)
plt.show()
```



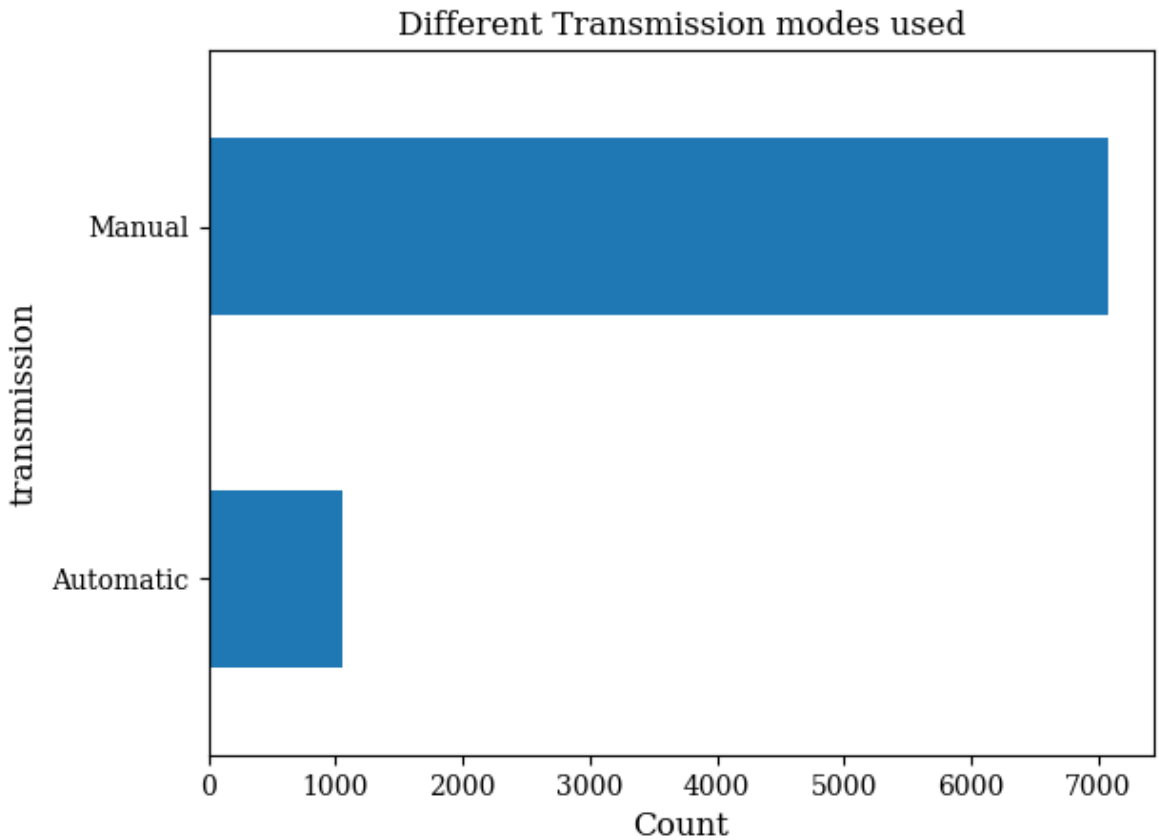
In [111]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   name           8128 non-null   object 
 1   year           8128 non-null   int64  
 2   selling_price  8128 non-null   int64  
 3   km_driven      8128 non-null   int64  
 4   fuel           8128 non-null   object 
 5   seller_type    8128 non-null   object 
 6   transmission   8128 non-null   object 
 7   owner          8128 non-null   object 
 8   mileage        8128 non-null   object 
 9   engine         7907 non-null   object 
 10  max_power      7913 non-null   object 
 11  torque         7906 non-null   object 
 12  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 825.6+ KB
```

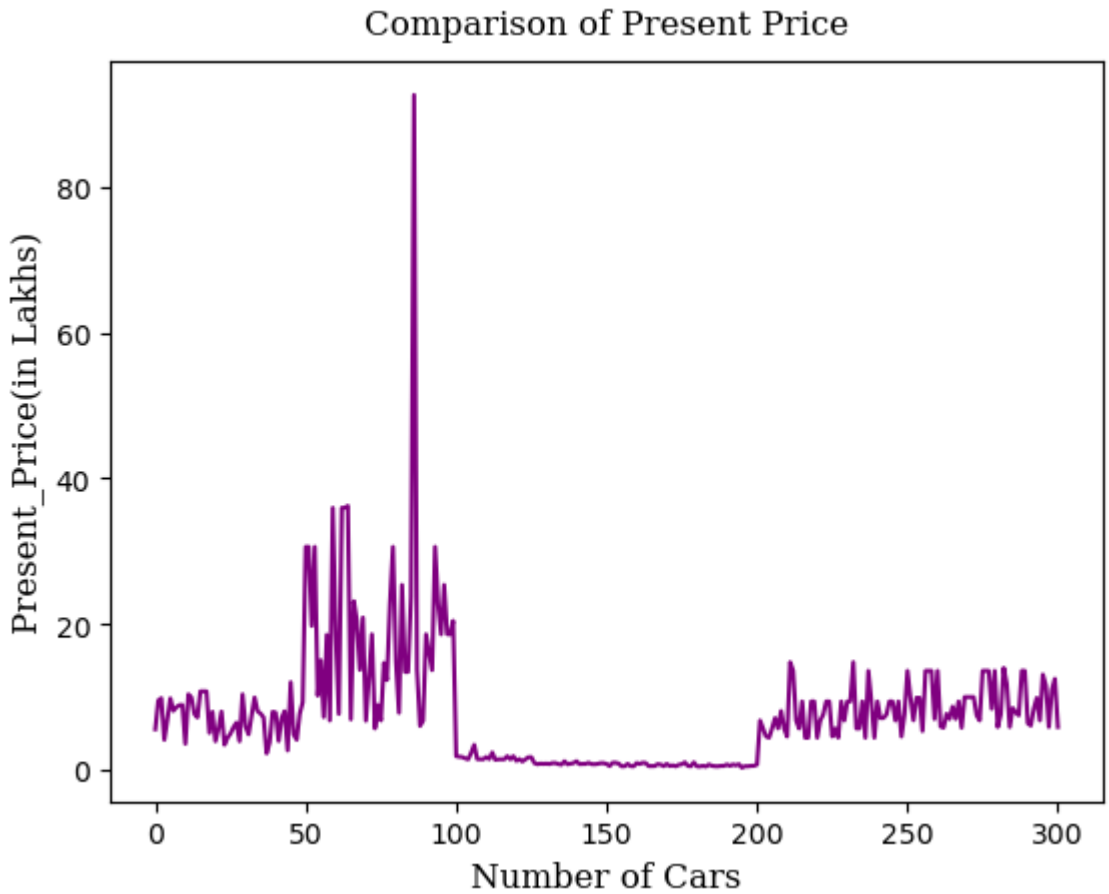
In [121]:
```
df['engine']=pd.to_numeric(df['engine'],errors='coerce')
print(df)
```

```
                              name  year  selling_price  km_driven  fuel
\
0           Maruti Swift Dzire VDI  2014         450000     145500   NaN
1     Skoda Rapid 1.5 TDI Ambition  2014         370000     120000   NaN
2         Honda City 2017-2020 EXi  2006         158000     140000   NaN
3        Hyundai i20 Sportz Diesel  2010         225000     127000   NaN
4           Maruti Swift VXI BSIII  2007         130000     120000   NaN
...                            ...   ...            ...        ...   ...
8123             Hyundai i20 Magna  2013         320000     110000   NaN
8124         Hyundai Verna CRDi SX  2007         135000     119000   NaN
8125        Maruti Swift Dzire ZDi  2009         382000     120000   NaN
8126               Tata Indigo CR4  2013         290000      25000   NaN
8127               Tata Indigo CR4  2013         290000      25000   NaN

     seller_type transmission                 owner mileage  engine  \
0     Individual       Manual           First Owner    mean     NaN
1     Individual       Manual          Second Owner    mean     NaN
2     Individual       Manual           Third Owner    mean     NaN
3     Individual       Manual           First Owner    mean     NaN
4     Individual       Manual           First Owner    mean     NaN
...          ...          ...                   ...     ...     ...
8123  Individual       Manual           First Owner    mean     NaN
8124  Individual       Manual  Fourth & Above Owner    mean     NaN
8125  Individual       Manual           First Owner    mean     NaN
8126  Individual       Manual           First Owner    mean     NaN
8127  Individual       Manual           First Owner    mean     NaN

       max_power                     torque  seats
0         74 bhp             190Nm@ 2000rpm    5.0
1     103.52 bhp        250Nm@ 1500-2500rpm    5.0
2         78 bhp      12.7@ 2,700(kgm@ rpm)    5.0
3         90 bhp   22.4 kgm at 1750-2750rpm    5.0
4       88.2 bhp      11.5@ 4,500(kgm@ rpm)    5.0
...          ...                        ...    ...
8123   82.85 bhp           113.7Nm@ 4000rpm    5.0
8124     110 bhp  24@ 1,900-2,750(kgm@ rpm)    5.0
8125    73.9 bhp             190Nm@ 2000rpm    5.0
8126      70 bhp        140Nm@ 1800-3000rpm    5.0
8127      70 bhp        140Nm@ 1800-3000rpm    5.0

[8128 rows x 13 columns]
```

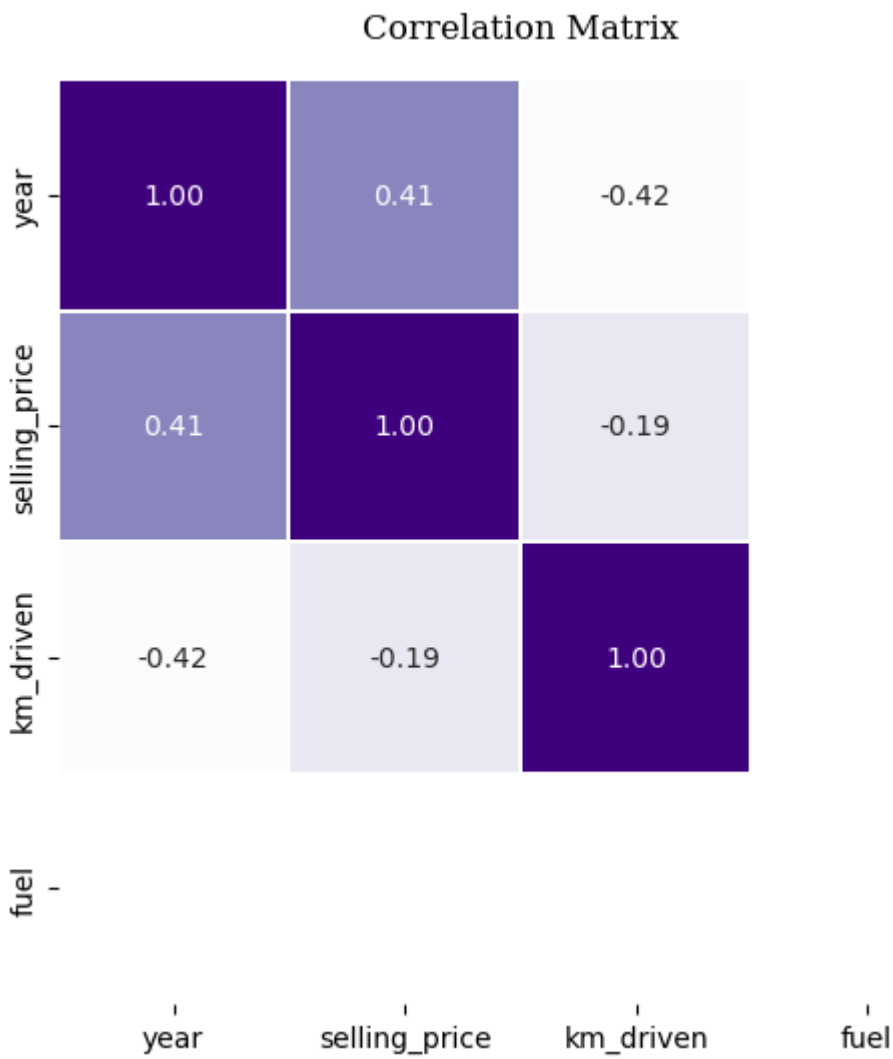
In [122]:
```
df['transmission'].value_counts().sort_values(ascending=True).plot.barh()
plt.xlabel('Count',family='serif',size=12)
plt.ylabel('transmission',family='serif',size=12)
plt.xticks(family='serif')
plt.yticks(family='serif')
plt.title('Different Transmission modes used',family='serif',size=12)
plt.show()
```



In [127]:
```
plt.plot(df2['Present_Price'],color='Purple')
plt.xlabel('Number of Cars',family='serif',size=12)
plt.ylabel('Present_Price(in Lakhs)',family='serif',size=12)
plt.title('Comparison of Present Price',family='serif',size=12,pad=10)
plt.show()
```



In [140]:
```
plt.figure(figsize=(7,6))
sns.heatmap(data=df1.corr(numeric_only=True),annot=True,cmap='Purples',cbar
plt.title('Correlation Matrix',family='serif',size=12,pad=15)
plt.show()
```



In [142]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   name           8128 non-null   object 
 1   year           8128 non-null   int64  
 2   selling_price  8128 non-null   int64  
 3   km_driven      8128 non-null   int64  
 4   fuel           0 non-null      float64
 5   seller_type    8128 non-null   object 
 6   transmission   8128 non-null   object 
 7   owner          8128 non-null   object 
 8   mileage        8128 non-null   object 
 9   engine         0 non-null      float64
 10  max_power      7913 non-null   object 
 11  torque         7906 non-null   object 
 12  seats          7907 non-null   float64
dtypes: float64(3), int64(3), object(7)
memory usage: 825.6+ KB
None
```

In [157]:
```
df['seller_type']=pd.to_numeric(df['seller_type'],errors='coerce').astype('
df.dtypes
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   name           8128 non-null   object 
 1   year           8128 non-null   int64  
 2   selling_price  8128 non-null   int64  
 3   km_driven      8128 non-null   int64  
 4   fuel           0 non-null      float64
 5   seller_type    0 non-null      Int64  
 6   transmission   8128 non-null   object 
 7   owner          8128 non-null   object 
 8   mileage        8128 non-null   object 
 9   engine         0 non-null      float64
 10  max_power      7913 non-null   object 
 11  torque         7906 non-null   object 
 12  seats          7907 non-null   float64
dtypes: Int64(1), float64(3), int64(3), object(6)
memory usage: 833.6+ KB
None
```

In [158]:
```
df['transmission']=pd.to_numeric(df['transmission'],errors='coerce').astype
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           0 non-null      float64
 5   seller_type    0 non-null      Int64
 6   transmission   0 non-null      Int64
 7   owner          8128 non-null   object
 8   mileage        8128 non-null   object
 9   engine         0 non-null      float64
 10  max_power      7913 non-null   object
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: Int64(2), float64(3), int64(3), object(5)
memory usage: 841.5+ KB
None
```

In [160]:
```
df['owner']=pd.to_numeric(df['owner'],errors='coerce').astype('Int64')
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           0 non-null      float64
 5   seller_type    0 non-null      Int64
 6   transmission   0 non-null      Int64
 7   owner          0 non-null      Int64
 8   mileage        8128 non-null   object
 9   engine         0 non-null      float64
 10  max_power      7913 non-null   object
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: Int64(3), float64(3), int64(3), object(4)
memory usage: 849.4+ KB
None
```

In [161]:
```
df['mileage']=pd.to_numeric(df['mileage'],errors='coerce').astype('Int64')
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           0 non-null      float64
 5   seller_type    0 non-null      Int64
 6   transmission   0 non-null      Int64
 7   owner          0 non-null      Int64
 8   mileage        0 non-null      Int64
 9   engine         0 non-null      float64
 10  max_power      7913 non-null   object
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: Int64(4), float64(3), int64(3), object(3)
memory usage: 857.4+ KB
None
```

In [163]:
```
df['max_power']=pd.to_numeric(df['max_power'],errors='coerce').astype('Int6
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           0 non-null      float64
 5   seller_type    0 non-null      Int64
 6   transmission   0 non-null      Int64
 7   owner          0 non-null      Int64
 8   mileage        0 non-null      Int64
 9   engine         0 non-null      float64
 10  max_power      6 non-null      Int64
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: Int64(5), float64(3), int64(3), object(2)
memory usage: 865.3+ KB
None
```

In [164]:
```
df['torque']=pd.to_numeric(df['torque'],errors='coerce').astype('Int64')
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           0 non-null      float64
 5   seller_type    0 non-null      Int64
 6   transmission   0 non-null      Int64
 7   owner          0 non-null      Int64
 8   mileage        0 non-null      Int64
 9   engine         0 non-null      float64
 10  max_power      6 non-null      Int64
 11  torque         0 non-null      Int64
 12  seats          7907 non-null   float64
dtypes: Int64(6), float64(3), int64(3), object(1)
memory usage: 873.2+ KB
None
```

In [181]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Car_Name       301 non-null    object
 1   Year           301 non-null    int64
 2   Selling_Price  301 non-null    float64
 3   Present_Price  301 non-null    float64
 4   Kms_Driven     301 non-null    int64
 5   Fuel_Type      301 non-null    object
 6   Seller_Type    301 non-null    object
 7   Transmission   301 non-null    object
 8   Owner          301 non-null    int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

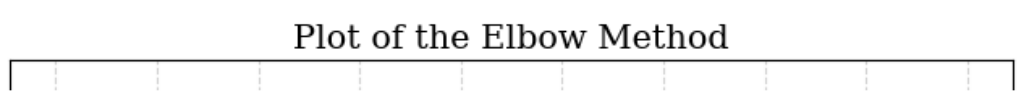
In [185]:
```
#encoding and selecting features
df2['Present_Price'].replace(to_replace=['RWD','FWD','AWD'],value=[0,1,2],i
df2['Selling_Price'].replace(to_replace=['no','yes'],value=[0,1],inplace=Tr

X= df2[['Year','Selling_Price','Present_Price','Kms_Driven','Owner']]

#feature scaling
scaler=StandardScaler()
X_scaled=scaler.fit_transform(X)

#PCA
pca=PCA(n_components=5)
X_pca=pca.fit_transform(X_scaled)
df3_pca=pd.DataFrame(X_pca,columns=['PC1','PC2','PC3','PC4','PC5'])
df3_pca.head()
```

Out[185]:

	PC1	PC2	PC3	PC4	PC5
0	-0.351939	-0.306730	-0.101146	-0.064044	0.054910
1	0.176722	0.219113	-0.228155	-0.102077	0.098614
2	0.562003	-1.373339	0.282492	0.185309	0.066949
3	-0.697317	0.028320	-0.114823	-1.131204	-0.194720
4	-0.029181	-0.058479	-0.191225	0.183703	-0.021023

In [190]:
```
#plotting the results

wcss=[]

for i in range(1,11):
    kmean = KMeans(n_clusters=i,init='k-means++', random_state=90)
    kmean.fit(X_pca)
    wcss.append(kmean.inertia_)

plt.figure(figsize=(8,6))
plt.title('Plot of the Elbow Method',size=15,family='serif')
plt.plot(range(1,11),wcss)
plt.xticks(range(1,11),family='serif')
plt.yticks(family='serif')
plt.xlabel('Number of Clusters (K)',family='serif')
plt.ylabel('WCSS',family='serif')
plt.grid()
plt.tick_params(axis='both',direction='inout',length=6,color='purple',grid_
plt.show()
```

```
the warning
  warnings.warn(
C:\Users\KeerthanaSEN\anaconda3\lib\site-packages\sklearn\cluster\_kmea
ns.py:1382: UserWarning: KMeans is known to have a memory leak on Windo
ws with MKL, when there are less chunks than available threads. You can
avoid it by setting the environment variable OMP_NUM_THREADS=2.
  warnings.warn(
C:\Users\KeerthanaSEN\anaconda3\lib\site-packages\sklearn\cluster\_kmea
ns.py:870: FutureWarning: The default value of `n_init` will change fro
m 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress
the warning
  warnings.warn(
C:\Users\KeerthanaSEN\anaconda3\lib\site-packages\sklearn\cluster\_kmea
ns.py:1382: UserWarning: KMeans is known to have a memory leak on Windo
ws with MKL, when there are less chunks than available threads. You can
avoid it by setting the environment variable OMP_NUM_THREADS=2.
  warnings.warn(
```



In [191]:
```
#training the model using k=4 from the above plot

kmean = KMeans(n_clusters=4,init='k-means++',random_state=90)
kmean.fit(X_pca)
```

```
C:\Users\KeerthanaSEN\anaconda3\lib\site-packages\sklearn\cluster\_kmean
s.py:870: FutureWarning: The default value of `n_init` will change from 1
0 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the
warning
  warnings.warn(
C:\Users\KeerthanaSEN\anaconda3\lib\site-packages\sklearn\cluster\_kmean
s.py:1382: UserWarning: KMeans is known to have a memory leak on Windows
with MKL, when there are less chunks than available threads. You can avoi
d it by setting the environment variable OMP_NUM_THREADS=2.
  warnings.warn(
```

Out[191]:
```
KMeans(n_clusters=4, random_state=90)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [192]:
```
print(kmean.labels_)
```

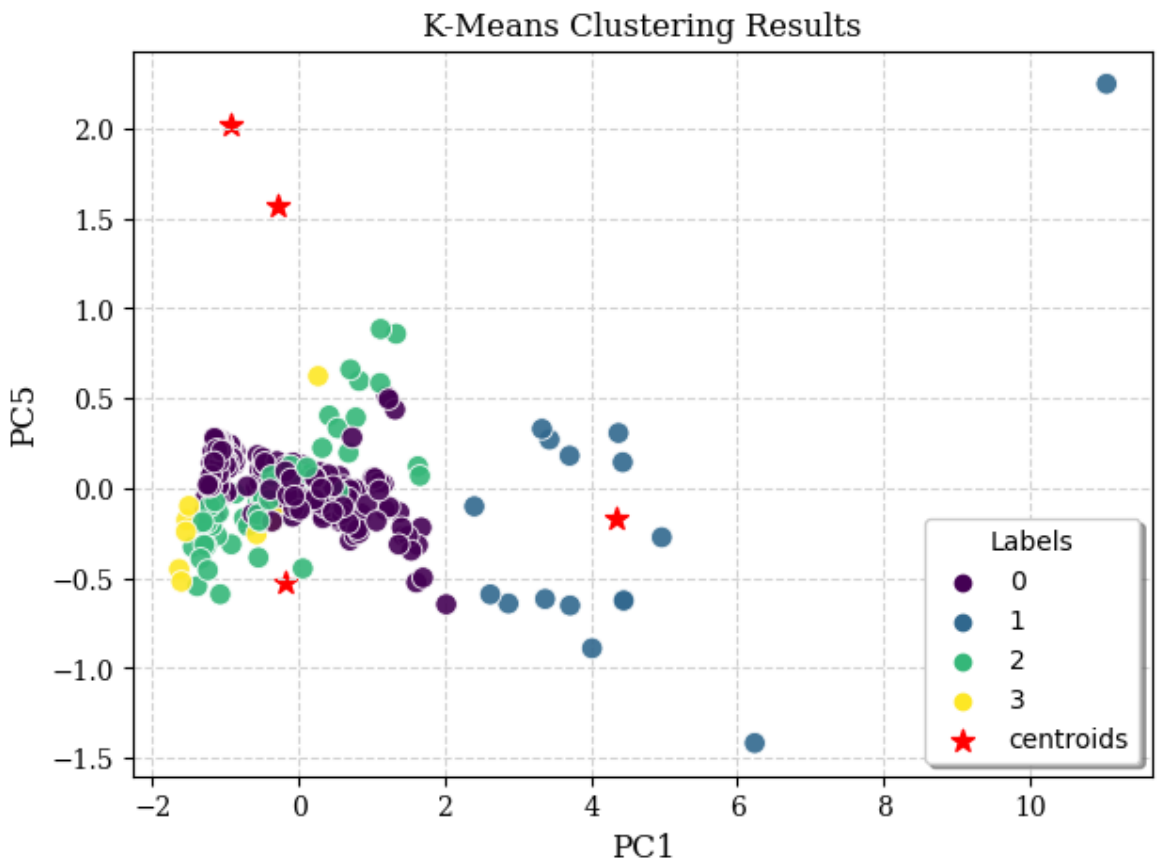
```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 2 0 0 2 0 0 0 0 0 2 0 2 2 0 0 0 2
0
 2 0 2 0 0 2 0 2 0 0 2 0 0 1 1 1 1 2 2 0 2 3 1 0 0 1 1 1 0 1 2 2 1 0 2 0
2
 0 0 2 2 2 1 0 0 1 0 2 3 1 2 0 0 2 0 2 1 2 2 1 1 0 2 0 0 0 0 0 0 3 0 0 0
0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0
 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 2 2 0 0 0
3
 2 2 0 0 2 2 3 3 3 2 0 2 2 3 2 2 3 0 2 0 3 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0
0
 0 0 0 0 0 2 2 2 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 2 0 0 0 0 0 0 0 0 0 2 0 0
0
 0 0 0 0 0 0 0 0 0 0 0 2 2 0 2 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 2 0
0
 0 0 2 0 0]
```

In [194]:
```
pd.Series(kmean.labels_).value_counts()
```

Out[194]:
```
0    215
2     59
1     16
3     11
dtype: int64
```

In [195]:
```
df2['clusters']=kmean.labels_
```

In [196]:
```
plt.figure(figsize=(7,5))
sns.scatterplot(data=df3_pca,x='PC1',y='PC5',s=70,hue=kmean.labels_,palette
plt.scatter(x=kmean.cluster_centers_[:,0],y=kmean.cluster_centers_[:,1],mar
plt.xlabel('PC1',family='serif',size=12)
plt.ylabel('PC5',family='serif',size=12)
plt.xticks(family='serif')
plt.yticks(family='serif')
plt.grid()
plt.tick_params(grid_color='lightgray',grid_linestyle='--',zorder=1)
plt.legend(title='Labels',fancybox=True,shadow=True)
plt.title('K-Means Clustering Results',family='serif',size=12)
plt.show()
```



In [ ]: