In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [88]:
```
df = pd.read_csv("skoda.csv")
print(df)
#print  first 5
print(df.head())
```

```
              model  year  price transmission  mileage fuelType   mpg  \
0           Octavia  2017  10550       Manual    25250   Petrol  54.3
1            Citigo  2018   8200       Manual     1264   Petrol  67.3
2           Octavia  2019  15650    Automatic     6825   Diesel  67.3
3      Yeti Outdoor  2015  14000    Automatic    28431   Diesel  51.4
4            Superb  2019  18350       Manual    10912   Petrol  40.9
...             ...   ...    ...          ...      ...      ...   ...
6262           Yeti  2014  11440    Semi-Auto    14569   Petrol  44.8
6263        Octavia  2014  10990    Semi-Auto    49999   Petrol  56.5
6264          Fabia  2017   9500    Semi-Auto    17131   Petrol  61.4
6265         Citigo  2016   5999       Manual    21747   Petrol  62.8
6266          Fabia  2017   9232    Semi-Auto    42530   Petrol  60.1

      engineSize
0            1.4
1            1.0
2            2.0
3            2.0
4            1.5
...          ...
6262         1.2
6263         1.4
6264         1.0
6265         1.0
6266         1.2

[6267 rows x 8 columns]
           model  year  price transmission  mileage fuelType   mpg  engineSize
0        Octavia  2017  10550       Manual    25250   Petrol  54.3         1.4
1         Citigo  2018   8200       Manual     1264   Petrol  67.3         1.0
2        Octavia  2019  15650    Automatic     6825   Diesel  67.3         2.0
3   Yeti Outdoor  2015  14000    Automatic    28431   Diesel  51.4         2.0
4         Superb  2019  18350       Manual    10912   Petrol  40.9         1.5
```

In [89]:
```
# Display First 5 and Last 5
display(df)

# Display First 10
display(df.head(10))
```

	model	year	price	transmission	mileage	fuelType	mpg	engineSize
0	Octavia	2017	10550	Manual	25250	Petrol	54.3	1.4
1	Citigo	2018	8200	Manual	1264	Petrol	67.3	1.0
2	Octavia	2019	15650	Automatic	6825	Diesel	67.3	2.0
3	Yeti Outdoor	2015	14000	Automatic	28431	Diesel	51.4	2.0
4	Superb	2019	18350	Manual	10912	Petrol	40.9	1.5
...	...	...	...	...	...	...	...	...
6262	Yeti	2014	11440	Semi-Auto	14569	Petrol	44.8	1.2
6263	Octavia	2014	10990	Semi-Auto	49999	Petrol	56.5	1.4
6264	Fabia	2017	9500	Semi-Auto	17131	Petrol	61.4	1.0
6265	Citigo	2016	5999	Manual	21747	Petrol	62.8	1.0
6266	Fabia	2017	9232	Semi-Auto	42530	Petrol	60.1	1.2

6267 rows × 8 columns

	model	year	price	transmission	mileage	fuelType	mpg	engineSize
0	Octavia	2017	10550	Manual	25250	Petrol	54.3	1.4
1	Citigo	2018	8200	Manual	1264	Petrol	67.3	1.0
2	Octavia	2019	15650	Automatic	6825	Diesel	67.3	2.0
3	Yeti Outdoor	2015	14000	Automatic	28431	Diesel	51.4	2.0

4	Superb	2019	18350	Manual	10912	Petrol	40.9	1.5
5	Yeti Outdoor	2017	13250	Automatic	47005	Diesel	51.4	2.0
6	Superb	2019	15250	Manual	14850	Petrol	40.9	1.5
7	Octavia	2019	18950	Automatic	5850	Diesel	50.4	2.0
8	Kodiaq	2019	29900	Automatic	2633	Petrol	31.4	2.0
9	Octavia	2017	18990	Manual	20000	Petrol	43.5	2.0

In [16]:

```
# Check missing value, data type of columns
display(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6267 entries, 0 to 6266
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   model         6267 non-null   object 
 1   year          6267 non-null   int64  
 2   price         6267 non-null   int64  
 3   transmission  6267 non-null   object 
 4   mileage       6267 non-null   int64  
 5   fuelType      6267 non-null   object 
 6   mpg           6267 non-null   float64
 7   engineSize    6267 non-null   float64
dtypes: float64(2), int64(3), object(3)
memory usage: 391.8+ KB
None
```

In [18]:

```
#Number of unique value in column year
year_count = df['year'].value_counts()
display(year_count)
```

```
2019    2114
2017    1539
2018     874
2016     840
2015     285
2020     276
2014     183
2013      93
2012      17
2011      14
2010      10
2009       6
2008       6
2007       4
2006       3
2005       2
2004       1
Name: year, dtype: int64
```

In [22]:

```
#Number of unique value in  fuel type
fuel_type = df['fuelType'].value_counts()
display(fuel_type)
```

```
Petrol    4171
Diesel    2069
Hybrid      18
Other        9
Name: fuelType, dtype: int64
```

In [23]:

```
#Number of unique value in transmission
transmission_count = df['transmission'].value_counts()
display(transmission_count)
```

```
Manual       3754
Semi-Auto    1408
```

```
Automatic    1104
Other           1
Name: transmission, dtype: int64
```

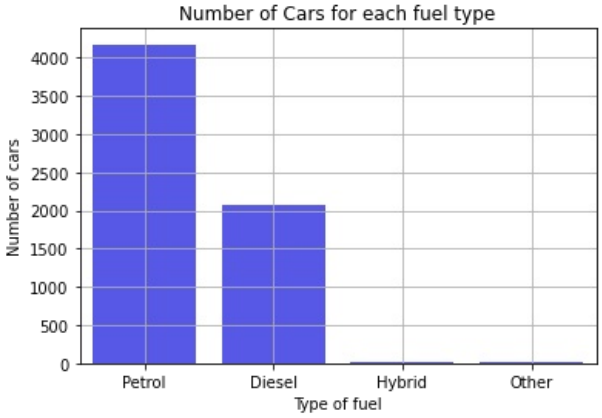
In [38]:

```
#Percentage of unique values present in the fuelType column

#Number of unique value in transmission
fuelType_count = df['fuelType'].value_counts()

#Taking into Dataframe
fuelType_count = pd.DataFrame(fuelType_count)

#Resettng index
fuelType_count = fuelType_count.reset_index()

#Renaming columns
fuelType_count = fuelType_count.rename(columns={'index':'fuel_type','fuelType':'No_of_cars'})

#Introducing new column for %
fuelType_count['% of cars'] = np.round((fuelType_count['No_of_cars']/fuelType_count['No_of_cars'].sum()*100),

#display(fuelType_count)
#display(fuelType_count['No_of_cars'].sum())
display(fuelType_count)
```

	fuel_type	No_of_cars	% of cars
0	Petrol	4171	66.55
1	Diesel	2069	33.01
2	Hybrid	18	0.29
3	Other	9	0.14

In [47]:

```
# Barplot using seaborn for fuleType column

sns.barplot(x='fuel_type', y = 'No_of_cars', data = fuelType_count, color ='blue', alpha = .75)

#title of the figure
plt.title("Number of Cars for each fuel type")

# X-axis and Y-axis label
plt.xlabel("Type of fuel")
plt.ylabel("Number of cars")

#showing on grid
plt.grid()

# to remove the top writings
#plt.show()
```



In [50]:

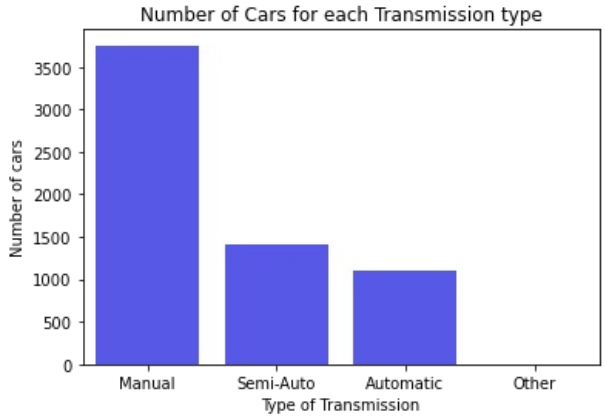
```
#Percentage of unique values present in the transmission column

#Number of unique value in transmission
transmission_count = df['transmission'].value_counts()

#Taking into Dataframe
transmission_count = pd.DataFrame(transmission_count)

#Resettng index
transmission_count = transmission_count.reset_index()
```

```
#Renaming columns
transmission_count = transmission_count.rename(columns={'index':'transmission','transmission':'No_of_cars'})

#Introducing new column for %
transmission_count['% of cars'] = np.round((transmission_count['No_of_cars']/transmission_count['No_of_cars']

#display(model_count)
#display(model_count['No_of_cars'].sum())
display(transmission_count)
```

	transmission	No_of_cars	% of cars
0	Manual	3754	59.90
1	Semi-Auto	1408	22.47
2	Automatic	1104	17.62
3	Other	1	0.02

In [52]:

```
# Barplot using seaborn for transmission column
sns.barplot(x='transmission', y = 'No_of_cars', data = transmission_count, color ='blue', alpha = .75)

#title of the figure
plt.title("Number of Cars for each transmission type")

# X-axis and Y-axis label
plt.xlabel("Type of transmission")
plt.ylabel("Number of cars")

# plt.grid()
# to remove the top writings
plt.show()
```



In [57]:

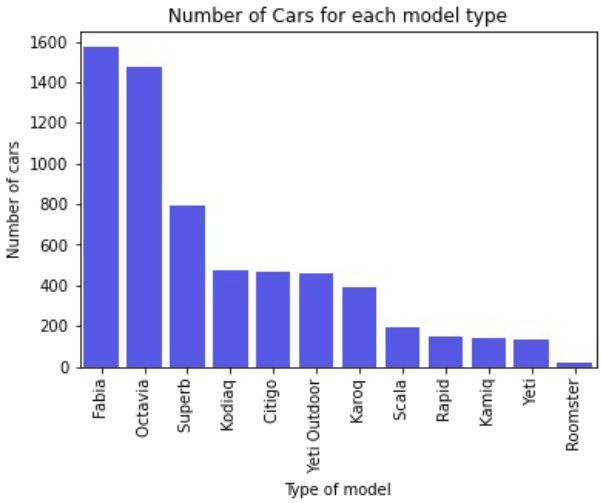
```
#Percentage of unique values present in the model column

#Number of unique value in transmission
model_count = df['model'].value_counts()

#Taking into Dataframe
model_count = pd.DataFrame(model_count)

#Resettng index
model_count = model_count.reset_index()

#Renaming columns
model_count = model_count.rename(columns={'index':'model','model':'No_of_cars'})

#Introducing new column for %
model_count['% of cars'] = np.round((model_count['No_of_cars']/model_count['No_of_cars'].sum()*100),2)

#display(model_count)
#display(model_count['No_of_cars'].sum())
display(model_count)
```

	model	No_of_cars	% of cars
0	Fabia	1571	25.07
1	Octavia	1477	23.57
2	Superb	791	12.62
3	Kodiaq	472	7.53
4	Citigo	470	7.50

5	Yeti Outdoor	458	7.31
6	Karoq	390	6.22
7	Scala	192	3.06
8	Rapid	152	2.43
9	Kamiq	141	2.25
10	Yeti	136	2.17
11	Roomster	17	0.27

In [59]:

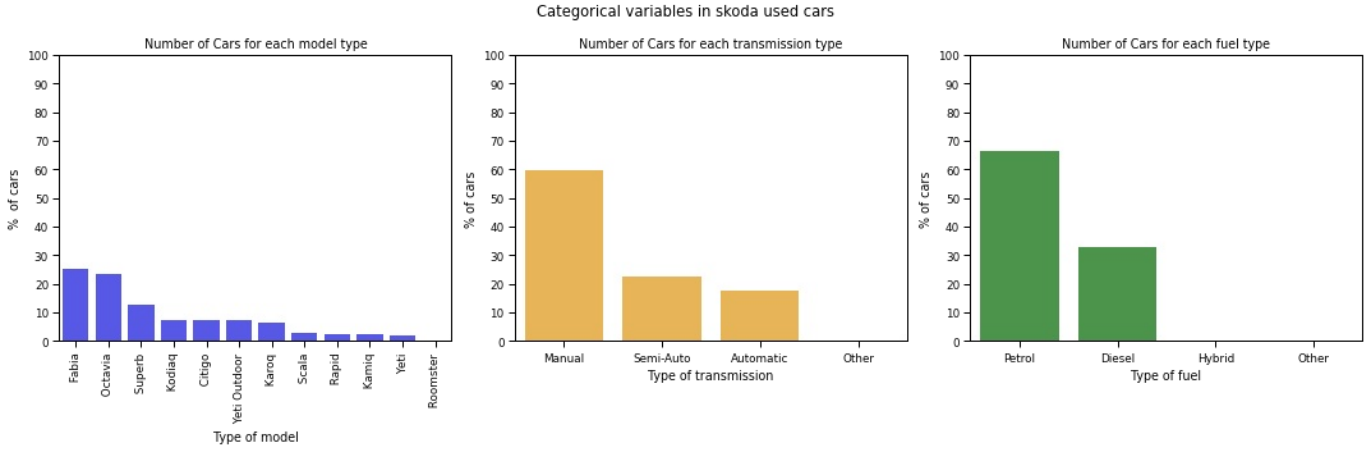
```
# Barplot using seaborn for model column

sns.barplot(x='model', y = 'No_of_cars', data = model_count, color ='blue', alpha = .75)

#title of the figure
plt.title("Number of Cars for each model type")

# X-axis and Y-axis label
plt.xlabel("Type of model")
plt.ylabel("Number of cars")

# the rotation of xticks
plt.xticks(rotation = 90)

#plt.grid()
# to remove the top writings
plt.show()
```



In [74]:

```
sns.set_context('paper')
#subplot using matplotlib and presenting all chart in one figure
plt.figure(figsize=(15,5))


#(1,1)
plt.subplot(1,3,1)
sns.barplot(x='model', y ='% of cars', data = model_count, color ='blue', alpha = 0.75)
#title of the plot
plt.title("Number of Cars for each model type")
# X-axis and Y-axis label
plt.xlabel("Type of model") plt.ylabel("%  of cars")
# rotation the xticks
plt.xticks(rotation = 90)
# range for the yticks
plt.yticks(np.arange(0,101,10).tolist())


#(1,2)
plt.subplot(1,3,2)
sns.barplot(x='transmission', y ='% of cars', data = transmission_count, color ='orange', alpha = 0.75)
#title of the plot
plt.title("Number of Cars for each transmission type")
# X-axis and Y-axis label
plt.xlabel("Type of transmission") plt.ylabel("% of cars")
# range for the yticks
plt.yticks(np.arange(0,101,10).tolist())


#(1,3)
```

```
    plt.subplot(1,3,3)
    sns.barplot(x='fuel_type', y ='% of cars', data = fuelType_count, color ='green', alpha = 0.75)
    #title of the plot
    plt.title("Number of Cars for each fuel type")
    # X-axis and Y-axis label
    plt.xlabel("Type of fuel") plt.ylabel("% of cars")
    # range for the yticks
    plt.yticks(np.arange(0,101,10).tolist())


    #overall title of all plot
    plt.suptitle("Categorical variables in skoda used cars")

    #Keeping plots separated
    plt.tight_layout()

    #shoing the figure
    plt.show()
```



In [84]:
```
def unique_val_count(data, column):

    #counting the unique falue
    df_count = data[column].value_counts()

    #Taking into dataframe
    df_count = pd.DataFrame(df_count)

    #Resettng index
    df_count = df_count.reset_index()

    #Renaming columns
    df_count = df_count.rename(columns={'index':column, column:'No_of_cars'})

    #Introducing new column for %
    df_count['% of cars'] = np.round((df_count['No_of_cars']/df_count['No_of_cars'].sum()*100), 2)

    return df_count
```

In [79]:
```
    model_count = unique_val_count(df, 'model')
    year_count = unique_val_count(df, 'year')
    transmission_count = unique_val_count(df, 'transmission')
    fuelType_count = unique_val_count(df, 'fuelType')
```

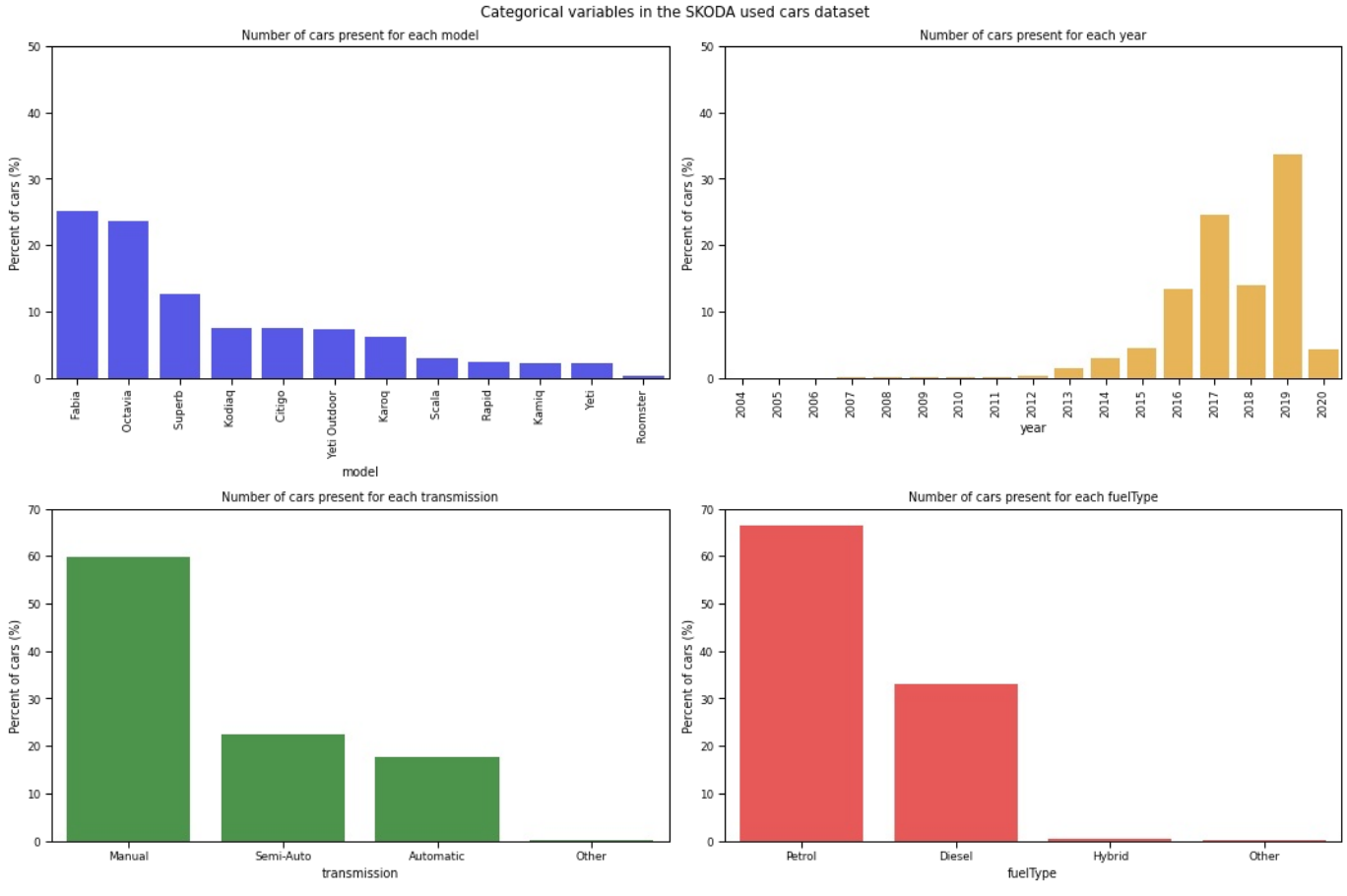
In [80]:
```
    # sort the year_count DataFrame based on year
    year_count = year_count.sort_values(by='year')
```

In [81]:
```
def barplot(data, column_x, color, rotation, yticks):

    #  barplot using seaborn
    sns.barplot(x=column_x, y='% of cars', data=data, color=color, alpha=0.75)

    # title of plot
    plt.title("Number of cars present for each " + column_x)

    # lebel for the x and y axis
    plt.xlabel(column_x)
    plt.ylabel("Percent of cars (%)")

    # rotation the xticks
    plt.xticks(rotation=rotation)

    # range for the yticks
```

```
    plt.yticks(yticks)
```

In [87]:

```
    sns.set_context('paper')

    plt.figure(figsize=(15,10))

    #(1,1)
    plt.subplot(2,2,1)
    barplot(model_count, 'model', 'blue', 90, np.arange(0,51,10))

    #(1,2)
    plt.subplot(2,2,2)
    barplot(year_count, 'year', 'orange', 90, np.arange(0,51,10))

    #(2,1)
    plt.subplot(2,2,3)
    barplot(transmission_count, 'transmission', 'green', 0, np.arange(0,71,10))

    #(2,2)
    plt.subplot(2,2,4)
    barplot(fuelType_count, 'fuelType', 'red', 0, np.arange(0,71,10))

    #title for all the plots
    plt.suptitle("Categorical variables in the SKODA used cars dataset")

    # individual plots  are separated
    plt.tight_layout()

    # display all the plots
    plt.show()
```



In [ ]: