Data visualisation using Plotly express

The plots made using Plotly Express are dynamic in nature, so they wont render in sites like Github directly.

In [1]:

```
import pandas as pd
import numpy as np
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
data = pd.read_csv('./Car_sales.csv', parse_dates=['Latest_Launch'])
data.head(10)
```

Out[2]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thou
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
5	Audi	A6	18.780	23.555	Passenger	
6	Audi	A8	1.380	39.000	Passenger	
7	BMW	323i	19.747	NaN	Passenger	
8	BMW	328i	9.231	28.675	Passenger	
9	BMW	528i	17.527	36.125	Passenger	

In [3]:

```
data.isnull().mean()*100
```

Out[3]:

```
Manufacturer            0.000000
Model                   0.000000
Sales_in_thousands      0.000000
__year_resale_value    22.929936
Vehicle_type            0.000000
Price_in_thousands      1.273885
Engine_size             0.636943
Horsepower              0.636943
Wheelbase               0.636943
Width                   0.636943
Length                  0.636943
Curb_weight             1.273885
Fuel_capacity           0.636943
Fuel_efficiency         1.910828
Latest_Launch           0.000000
Power_perf_factor       1.273885
dtype: float64
```

In [4]:

```
data.dropna(inplace=True)
data.head()
```

Out[4]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thou
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
5	Audi	A6	18.780	23.555	Passenger	

In [5]:

```
data['Year'] = data.Latest_Launch.dt.year
```

In [6]:

```
data['Month'] =  data.Latest_Launch.dt.month
```

In [7]:

```
data.Month.replace({1:"January",
2: "February",
3:"March",
4: "April",
5: "May",
6: "June",
7: "July",
8 :"August",
9 : "September",
10 :"October",
11: "November",
12:"December" }, inplace=True)
```

In [8]:

```
data['Month_Number'] = data.Latest_Launch.dt.month
```

In [9]:

```
data.head()
```

Out[9]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thou
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	
5	Audi	A6	18.780	23.555	Passenger	

In [10]:

```
data.sort_values(by=['Year', 'Month_Number'], inplace=True)
data.head(10)
```

Out[10]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_
90	Mercury	Mountaineer	27.609	20.430	Car	
91	Mercury	Villager	20.380	14.795	Car	
131	Saturn	SW	5.223	10.790	Passenger	
145	Volkswagen	Golf	9.761	11.425	Passenger	
111	Oldsmobile	Aurora	14.690	19.890	Passenger	
137	Toyota	Camry	247.994	13.245	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
95	Mercedes-B	SL-Class	3.311	58.600	Passenger	
104	Nissan	Quest	27.308	15.380	Car	
130	Saturn	SC	24.546	10.590	Passenger	

In [11]:

```
data2011 = data.query("Year == 2011")
```

In [12]:

```
data2011.head()
```

Out[12]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_th
131	Saturn	SW	5.223	10.790	Passenger	
145	Volkswagen	Golf	9.761	11.425	Passenger	
111	Oldsmobile	Aurora	14.690	19.890	Passenger	
137	Toyota	Camry	247.994	13.245	Passenger	
3	Acura	RL	8.588	29.725	Passenger	

In [13]:

```
Year_month_Sales = data.groupby(['Year', 'Month'])['Sales_in_thousands'].sum()
Year_month_Sales = Year_month_Sales.reset_index()
```

In [14]:

```
Year_month_Sales
```

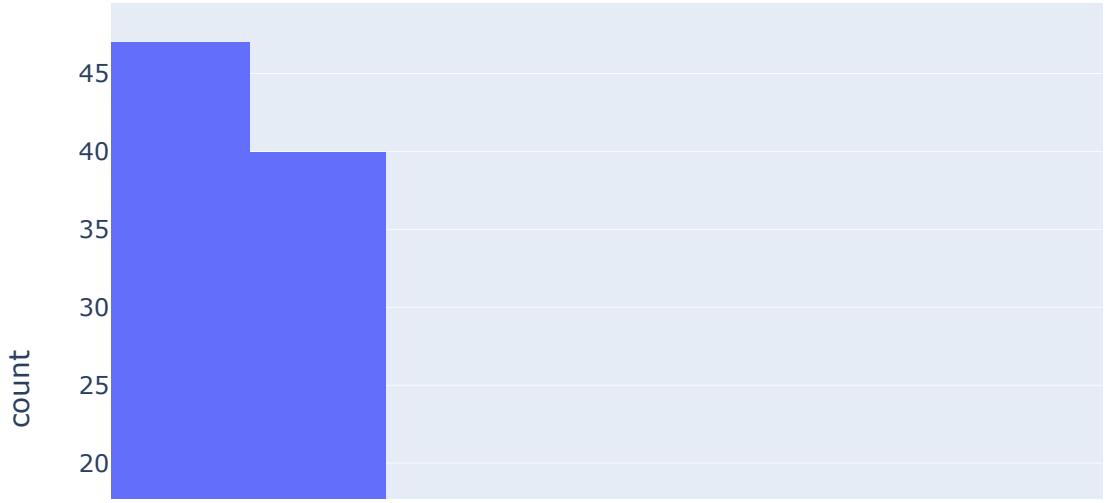
Out[14]:

	Year	Month	Sales_in_thousands
0	2008	February	27.609
1	2009	October	20.380
2	2011	April	240.939
3	2011	August	527.609
4	2011	December	521.482
5	2011	February	262.684
6	2011	January	14.984
7	2011	July	112.725
8	2011	June	80.519
9	2011	March	132.164
10	2011	May	176.214
11	2011	November	314.117
12	2011	October	349.976
13	2011	September	400.412
14	2012	April	348.547
15	2012	August	675.758
16	2012	December	48.900
17	2012	February	349.355
18	2012	January	486.469
19	2012	July	141.018
20	2012	June	129.530
21	2012	March	595.400
22	2012	May	314.357
23	2012	November	173.638
24	2012	October	236.877
25	2012	September	234.478

Histogram

In [15]:

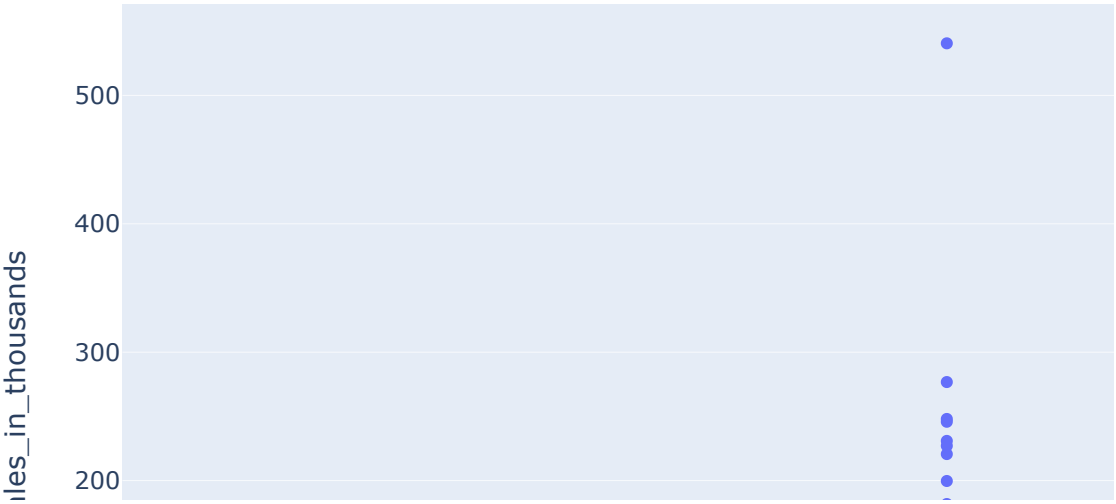
```
fig = px.histogram(data_frame=data, x='Sales_in_thousands')
fig.show()
```



Box plots

In [16]:

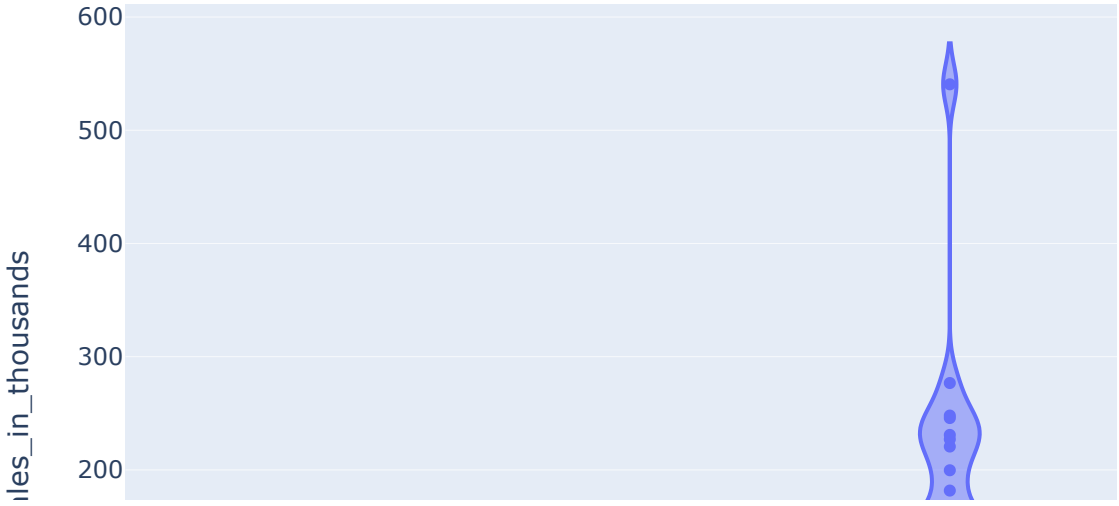
```
fig = px.box(data_frame=data, y='Sales_in_thousands')
fig.show()
```



Violin Plot

In [17]:

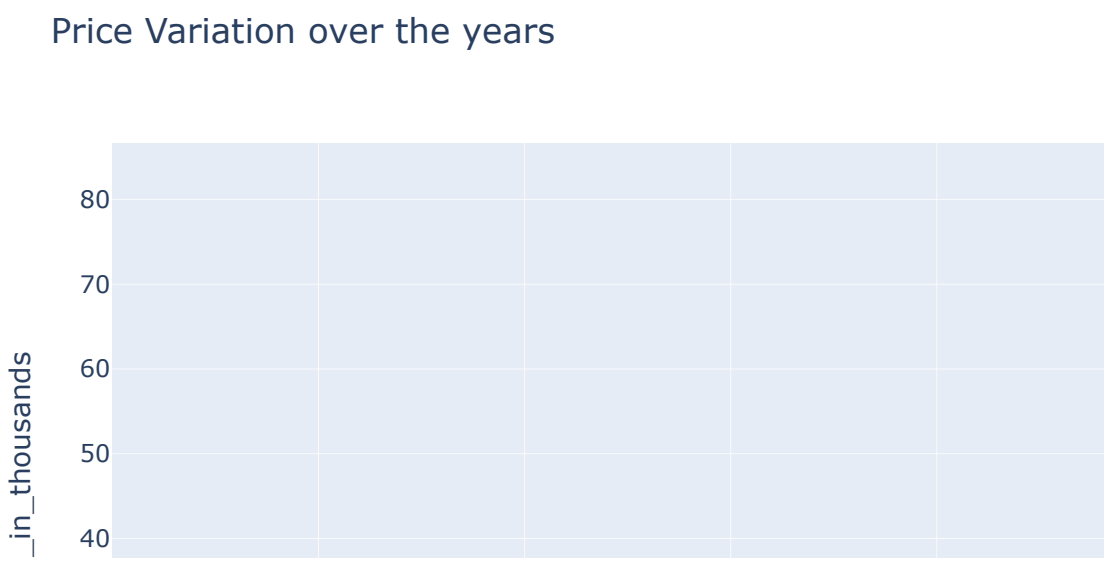
```
fig = px.violin(data_frame=data, y='Sales_in_thousands')
fig.show()
```



Line plots

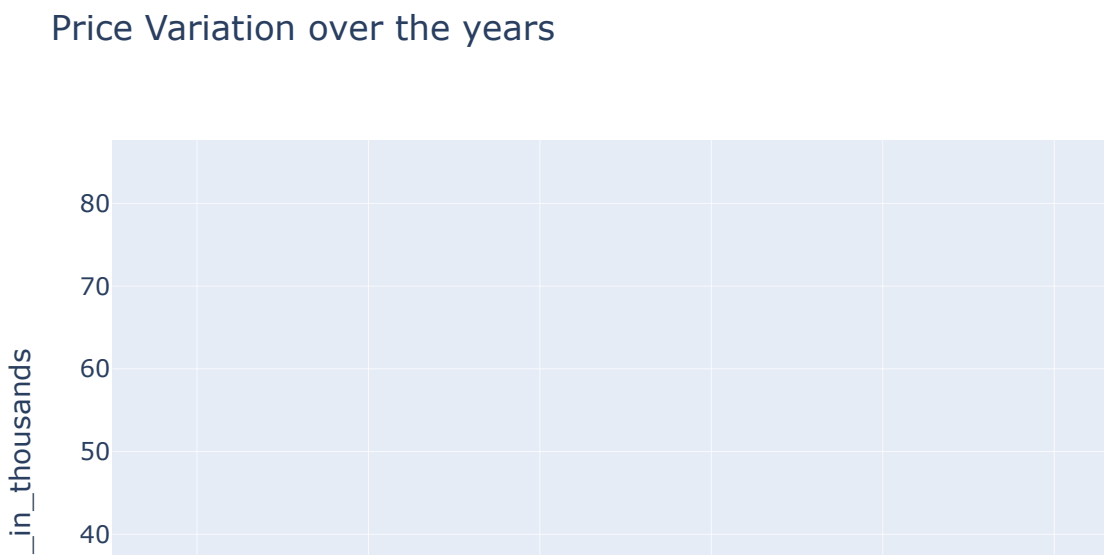
In [18]:

```
fig = px.line(data_frame=data, x='Year', y='Price_in_thousands', title='Price Variation o
fig.show()
```



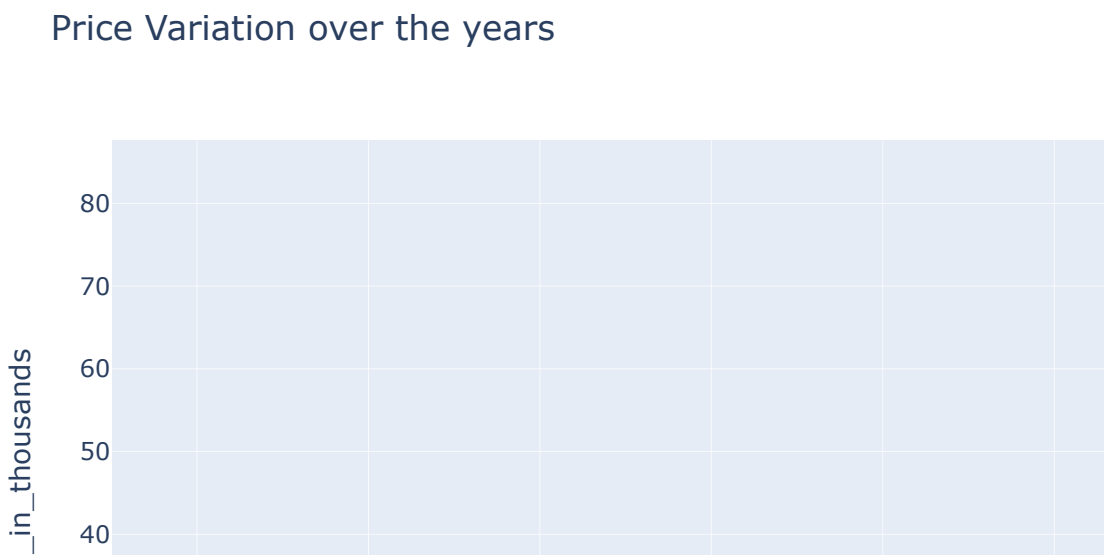
In [19]:

```
fig = px.line(data_frame=data, x='Year', y='Price_in_thousands',
              title='Price Variation over the years',
              color='Vehicle_type', text='Year')
fig.show()
```



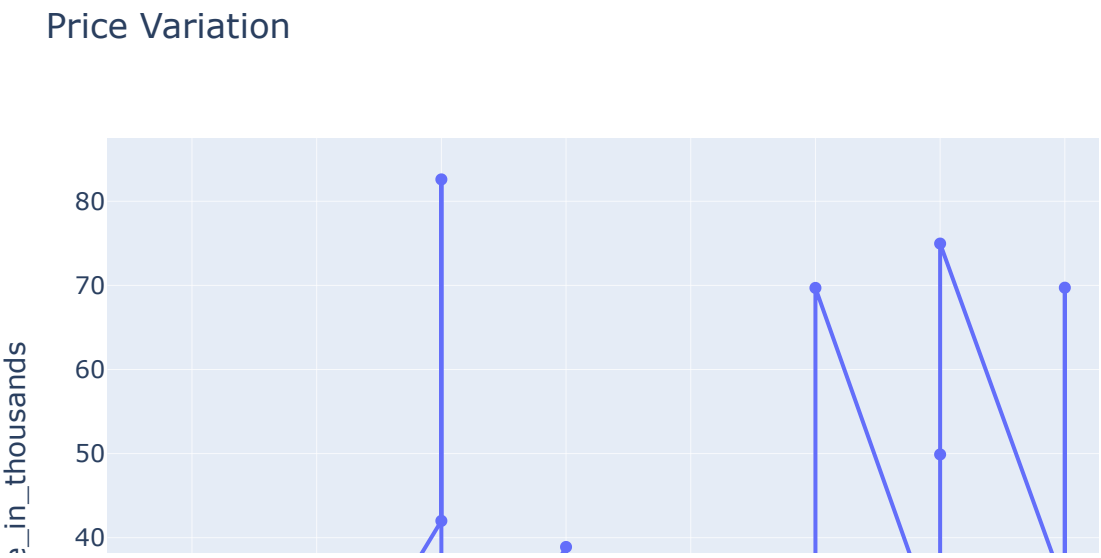
In [20]:

```
fig = px.line(data_frame=data, x='Year', y='Price_in_thousands',
              title='Price Variation over the years',
              color='Vehicle_type', markers=True)
fig.show()
```



In [21]:

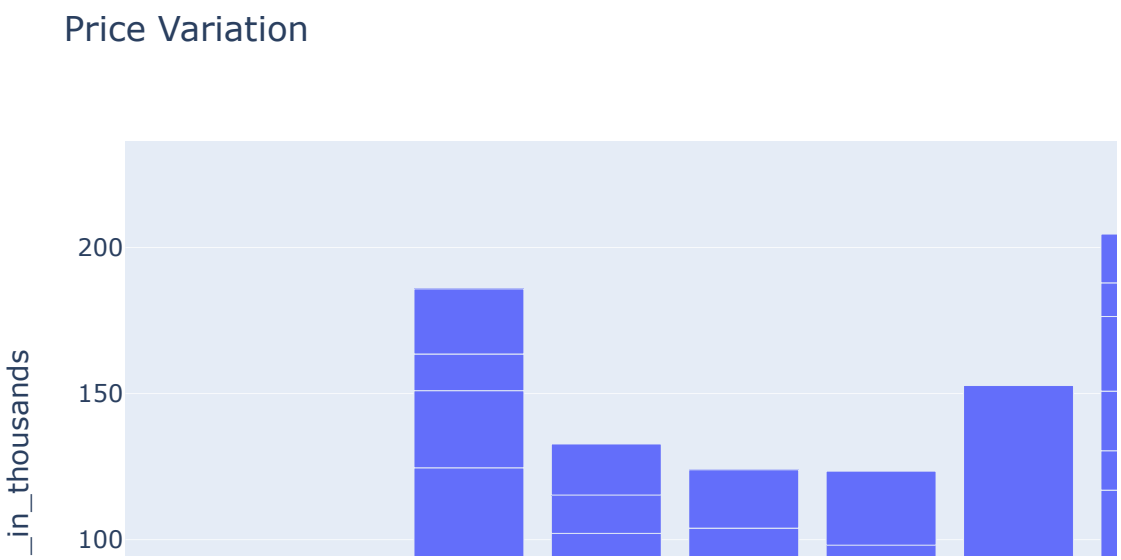
```
fig = px.line(data_frame=data2011, x='Month', y='Price_in_thousands',
              title='Price Variation',
              color='Vehicle_type', markers=True)
fig.show()
```



Bar chart

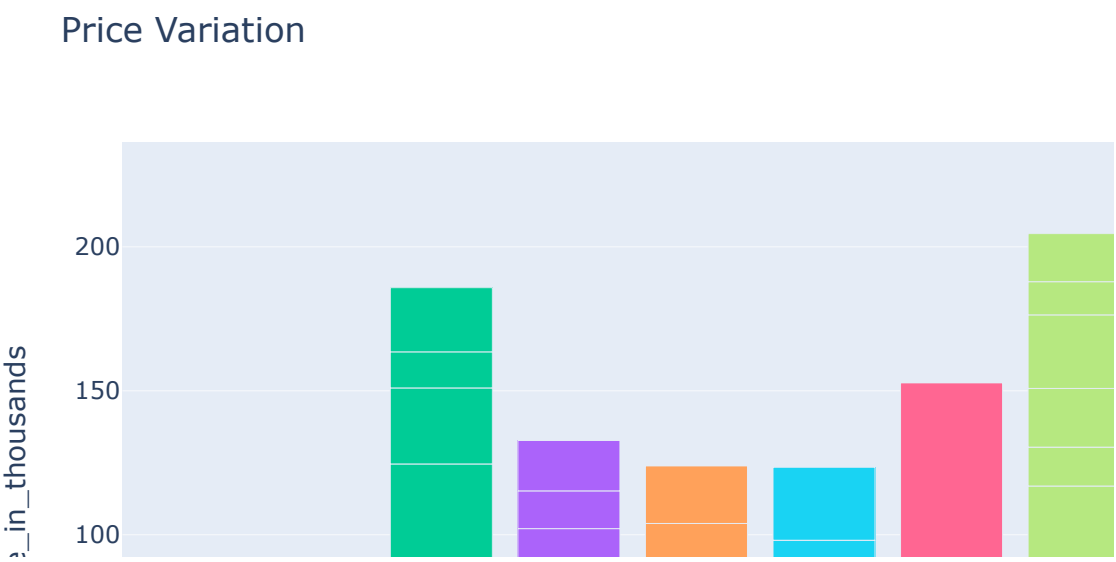
In [22]:

```
fig = px.bar(data_frame=data2011, x='Month', y='Price_in_thousands',
             title='Price Variation')
fig.show()
```



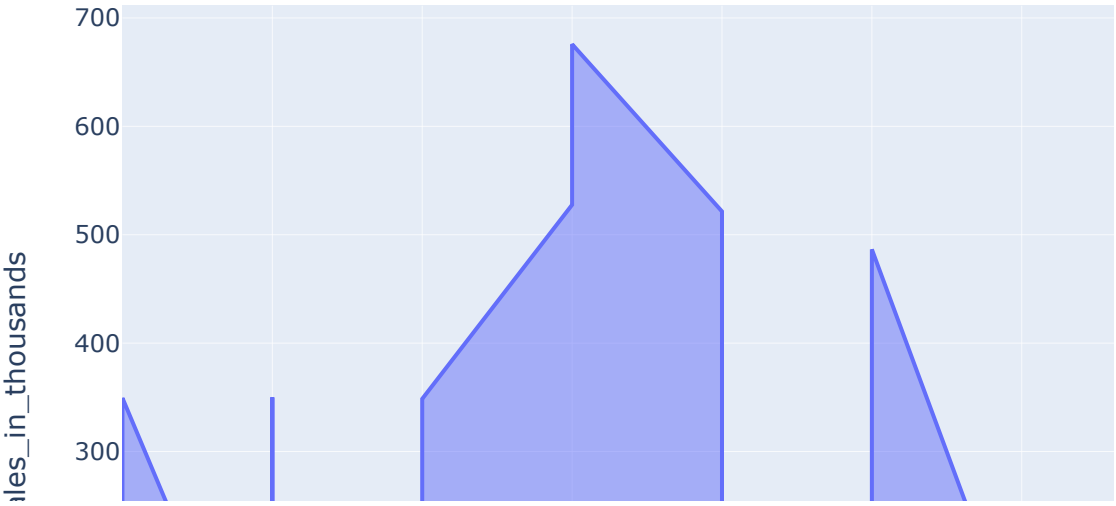
In [23]:

```
fig = px.bar(data_frame=data2011, x='Month', y='Price_in_thousands',
             color='Month',title='Price Variation')
fig.show()
```



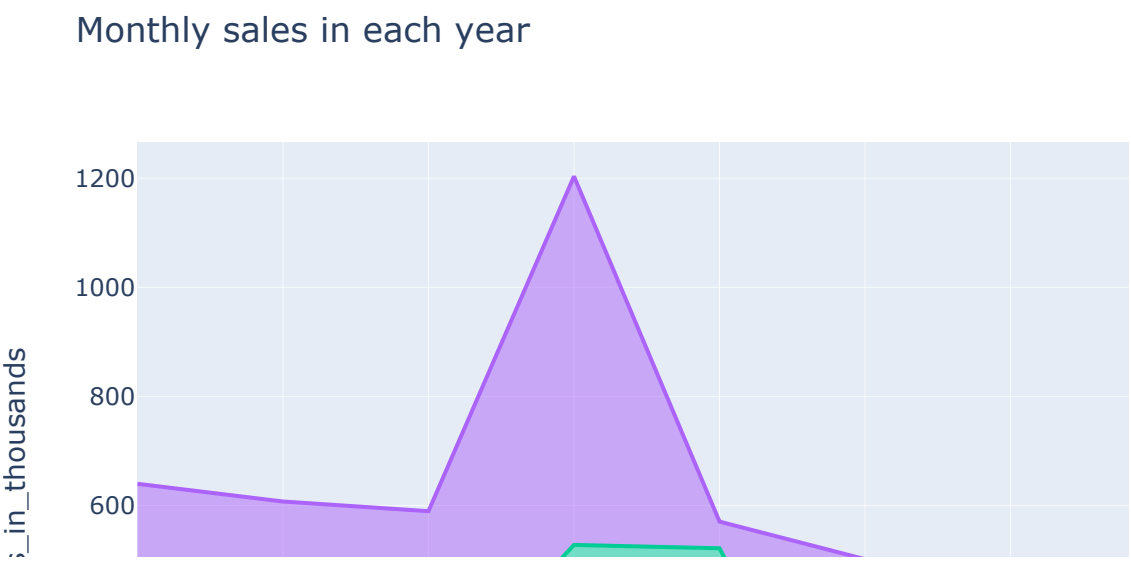
In [24]:

```
fig = px.area(data_frame=Year_month_Sales, x= 'Month', y='Sales_in_thousands')
fig.show()
```



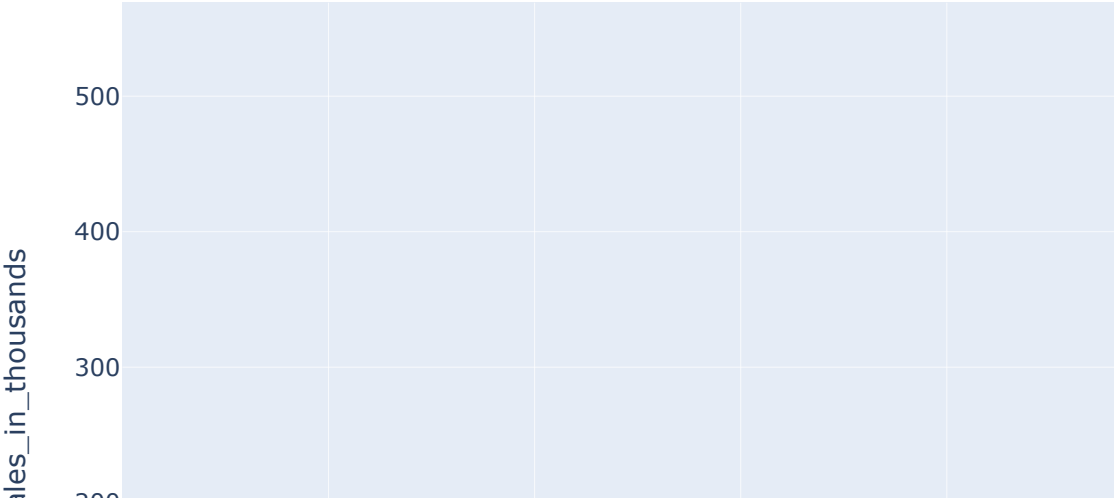
In [25]:

```
fig = px.area(data_frame=Year_month_Sales, x= 'Month', y='Sales_in_thousands', color='Yea
             title='Monthly sales in each year')
fig.show()
```



In [26]:

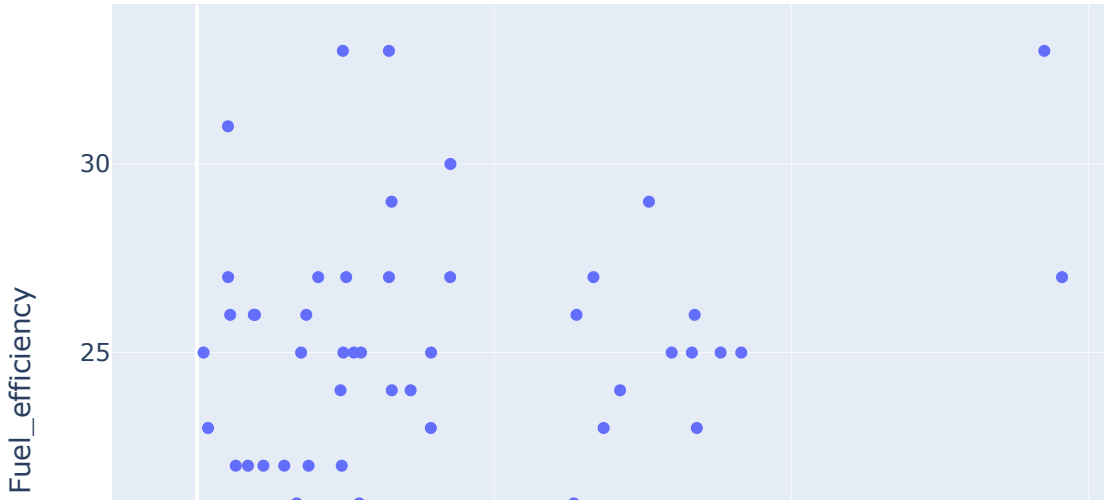
```
fig = px.area(data_frame=data, x= 'Year', y='Sales_in_thousands')
fig.show()
```



Scatter Plots

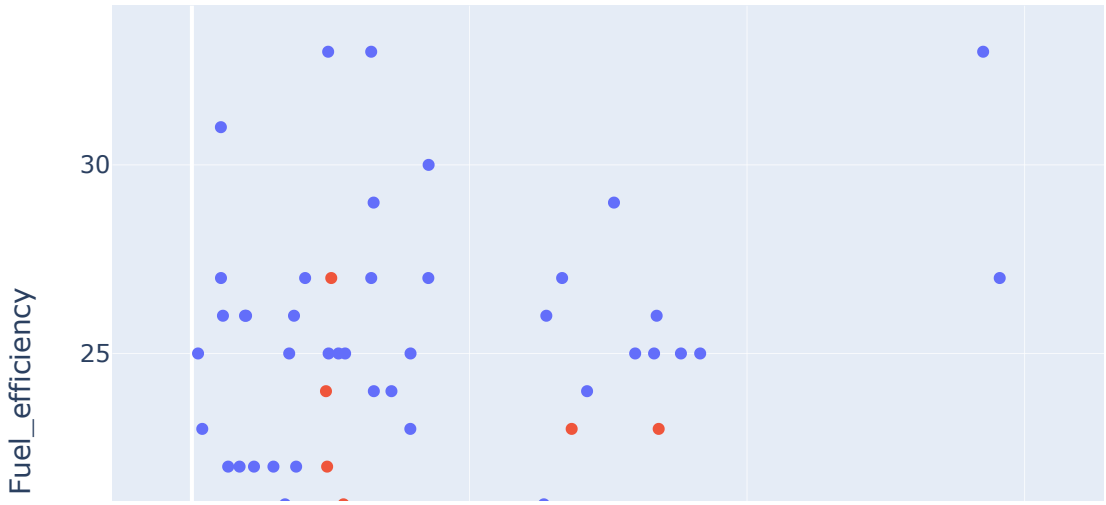
In [27]:

```
fig = px.scatter(data_frame=data2011, x='Sales_in_thousands', y='Fuel_efficiency')
fig.show()
```



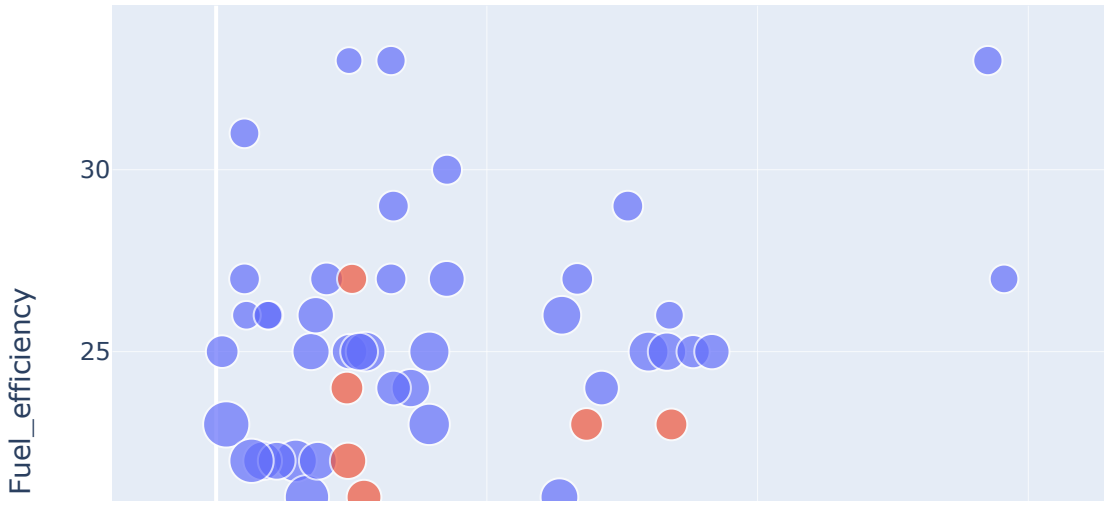
In [28]:

```
fig = px.scatter(data_frame=data2011, x='Sales_in_thousands', y='Fuel_efficiency', color=
fig.show()
```



In [29]:

```
fig = px.scatter(data_frame=data2011, x='Sales_in_thousands', y='Fuel_efficiency', color=
fig.show()
```



Pie Chart

In [30]:

```
fig = px.pie(data_frame=data2011, names='Vehicle_type',values='Sales_in_thousands')
fig.show()
```