

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
```

```
In [2]: data = pd.read_csv("D:/Softwares/Python/Material/car_sales.csv")
```

```
In [3]: data.head()
```

Out[3]:

Unnamed: 0	Manufacturer	Model	Type	Min.Price	Price	Max.Price	AirBags	...	Passengers	Length	Wheelbase	Width	Turn.circle	Rear.seat.room	Luggage.room	Weight	Origin	Make
0	1	Acura Integra	Small	12.9	15.9	18.8	5	...	5	177	102	68	37	26.5	11	2705	non-USA	Acura Integra
1	2	Acura Legend	Midsize	29.2	33.9	38.7	5	...	5	195	115	71	38	30.0	15	3560	non-USA	Acura Legend
2	3	Audi 90	Compact	25.9	29.1	32.3	5	...	5	180	102	67	37	28.0	14	3375	non-USA	Audi 90
3	4	Audi 100	Midsize	30.8	37.7	44.6	5	...	6	193	106	70	37	31.0	17	3405	non-USA	Audi 100
4	5	BMW 535i	Midsize	23.7	30.0	36.2	4	...	4	186	109	69	39	27.0	13	3640	non-USA	BMW 535i

5 rows × 28 columns

```
In [4]: print(data.head())
```

Unnamed: 0	Manufacturer	Model	Type	Min.Price	Price	Max.Price	AirBags	...	Passengers	Length	Wheelbase	Width	Turn.circle	Rear.seat.room	Luggage.room	Weight	Origin	Make
0	1	Acura Integra	Small	12.9	15.9	18.8	5	...	5	177	102	68	37	26.5	11	2705	non-USA	Acura Integra
1	2	Acura Legend	Midsize	29.2	33.9	38.7	5	...	5	195	115	71	38	30.0	15	3560	non-USA	Acura Legend
2	3	Audi 90	Compact	25.9	29.1	32.3	5	...	5	180	102	67	37	28.0	14	3375	non-USA	Audi 90
3	4	Audi 100	Midsize	30.8	37.7	44.6	5	...	6	193	106	70	37	31.0	17	3405	non-USA	Audi 100
4	5	BMW 535i	Midsize	23.7	30.0	36.2	4	...	4	186	109	69	39	27.0	13	3640	non-USA	BMW 535i

[5 rows x 28 columns]

In [5]: data.describe()

Out[5]:

	Unnamed: 0	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	DriveTrain	Cylinders	EngineSize	Horsepower	RPM	Rev.per.mile	Man.trans.avail	Fuel.tank.capacity	Passengers	Length	Wheelbase	Width	Turn.circle	Rear.seat.room	Luggage.room	Weight	Origin	Make
count	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
unique	1	15	15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
top	0	Toyota	Camry	Sedan	0	15000	25000	24	34	0	Front-wheel drive	4	2.5	158	5800	24	0	16.0	5	182.7	104.3	70.9	33.7	16.4	35.3	3100	USA	Toyota
mean	0	Toyota	Camry	Sedan	0	15000	25000	24	34	0	Front-wheel drive	4	2.5	158	5800	24	0	16.0	5	182.7	104.3	70.9	33.7	16.4	35.3	3100	USA	Toyota
std	0	Toyota	Camry	Sedan	0	15000	25000	24	34	0	Front-wheel drive	4	2.5	158	5800	24	0	16.0	5	182.7	104.3	70.9	33.7	16.4	35.3	3100	USA	Toyota
min	0	Toyota	Camry	Sedan	0	15000	25000	24	34	0	Front-wheel drive	4	2.5	158	5800	24	0	16.0	5	182.7	104.3	70.9	33.7	16.4	35.3	3100	USA	Toyota
max	0	Toyota	Camry	Sedan	0	15000	25000	24	34	0	Front-wheel drive	4	2.5	158	5800	24	0	16.0	5	182.7	104.3	70.9	33.7	16.4	35.3	3100	USA	Toyota
dtypes	int64	object	object	object	int64	int64	int64	int64	int64	int64	object	int64	float64	int64	int64	float64	int64	float64	int64	float64	float64	float64	float64	float64	float64	float64	float64	float64

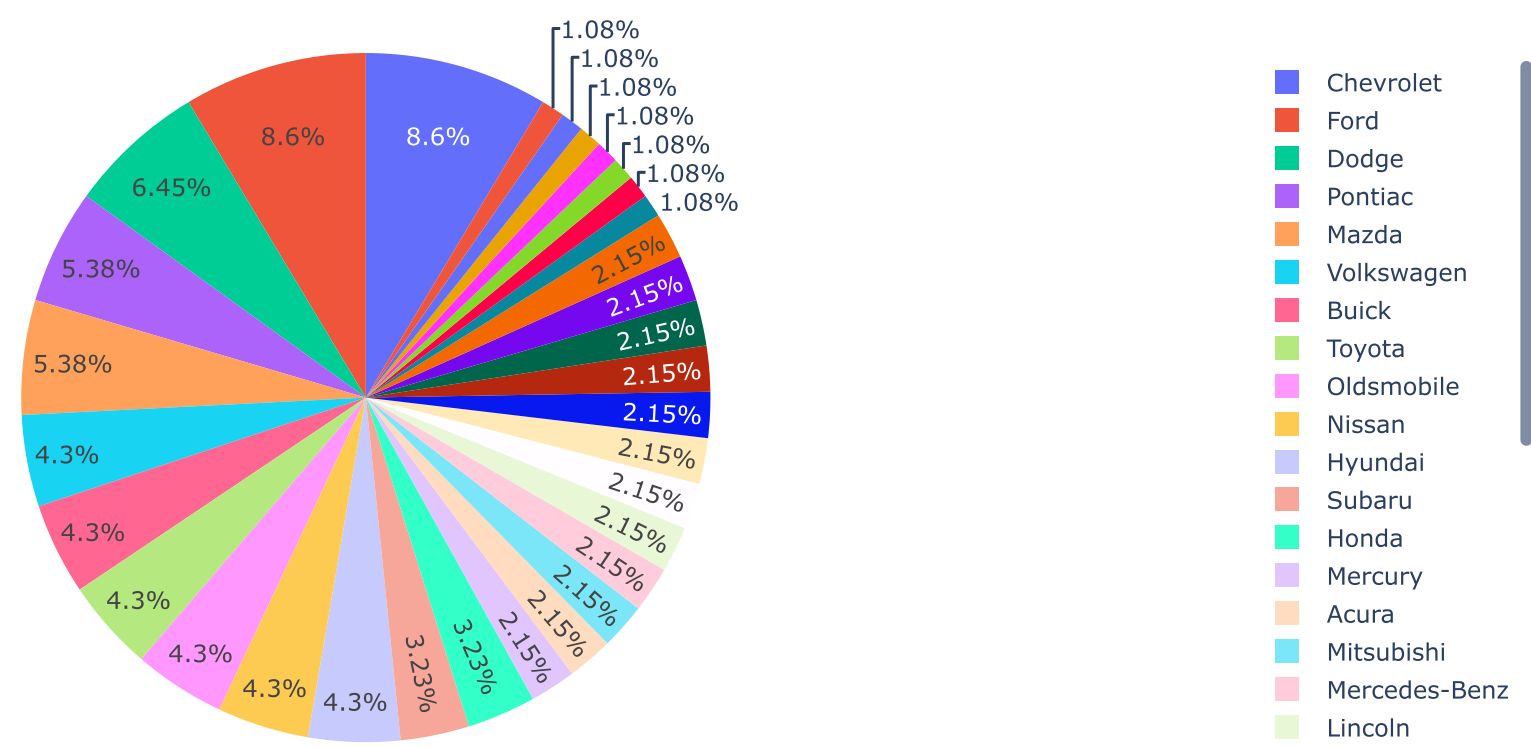
In [6]: print(data.isnull().sum())

Unnamed: 0 0
Manufacturer 0
Model 0
Type 0
Min.Price 0
Price 0
Max.Price 0
MPG.city 0
MPG.highway 0
AirBags 0
DriveTrain 0
Cylinders 0
EngineSize 0
Horsepower 0
RPM 0
Rev.per.mile 0
Man.trans.avail 0
Fuel.tank.capacity 0
Passengers 0
Length 0
Wheelbase 0
Width 0
Turn.circle 0
Rear.seat.room 0
Luggage.room 0
Weight 0
Origin 0
Make 0
dtype: int64

In [7]: Nos = data["Manufacturer"].value_counts()
counts = Nos.values
label = Nos.index

In [8]: fig = go.Figure(data=[go.Pie(labels = label, values = counts)])
fig.update_layout(title = "No of model provided by Manufacturer")
fig.show()

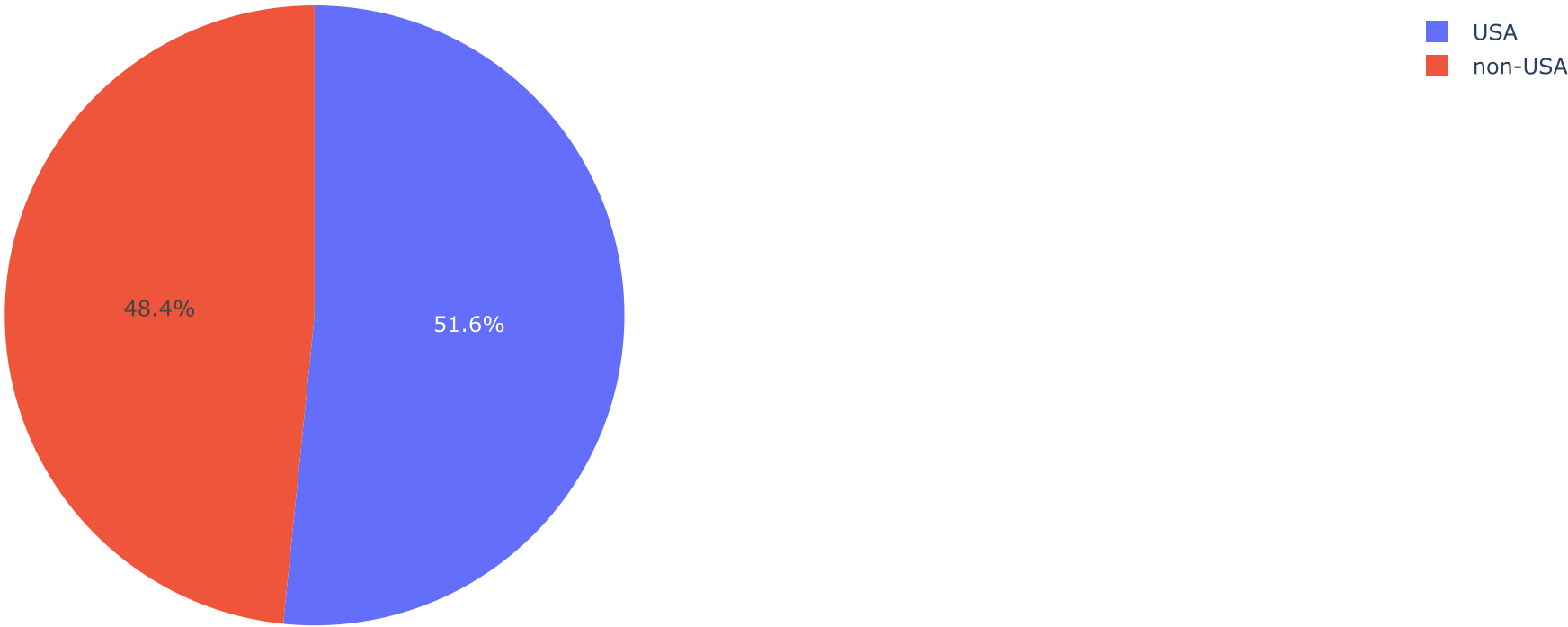
No of model provided by Manufacturer



```
In [9]: Nos = data["Origin"].value_counts()
counts = Nos.values
label = Nos.index

In [10]: fig = go.Figure(data=[go.Pie(labels = label, values = counts)])
fig.update_layout(title = "Foreign vs Local Manufacturer")
fig.show()
```

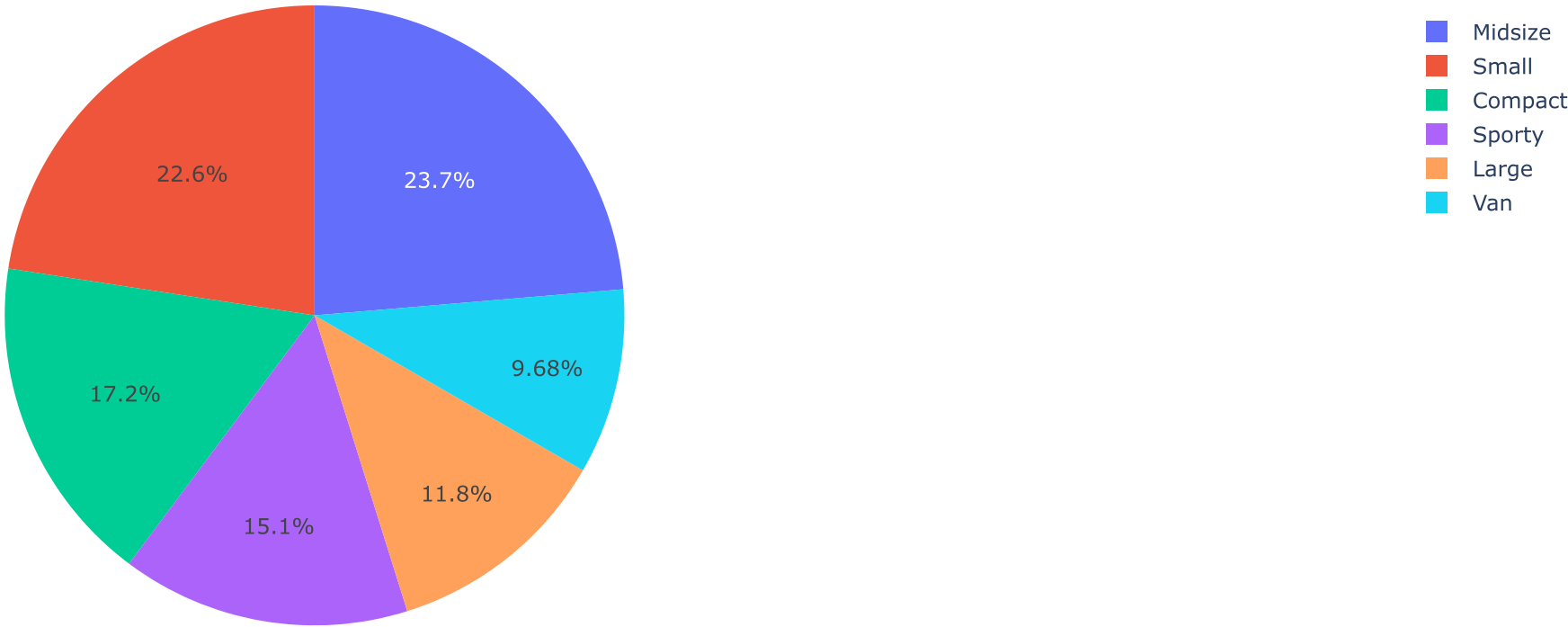
Foreign vs Local Manufacturer



```
In [11]: Nos = data["Type"].value_counts()
        label = Nos.index
        counts = Nos.values

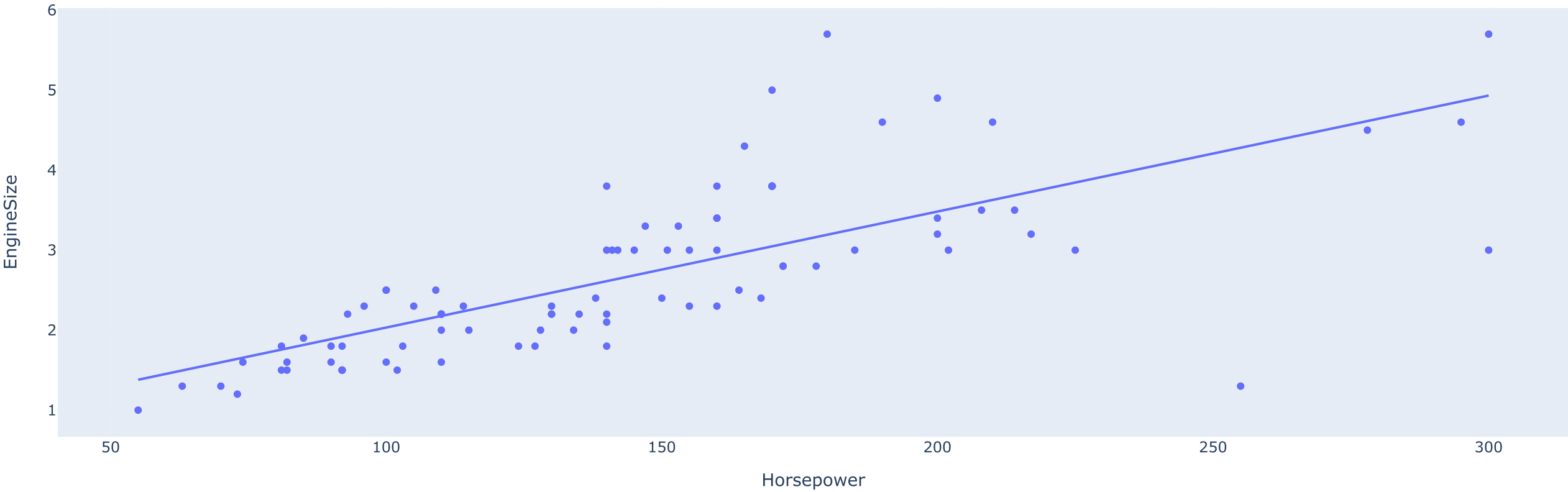
In [12]: fig = go.Figure(data=[go.Pie(labels = label, values = counts,)])
        fig.update_layout( title = "Vehicle Types")
        fig.show()
```

Vehicle Types



```
In [13]: Engine_Size = data["EngineSize"]
figure = px.scatter(data, x = "Horsepower", y = Engine_Size, trendline = "ols", title = "Engine Size to Horsepower")
figure.show()
```

Engine Size to Horsepower



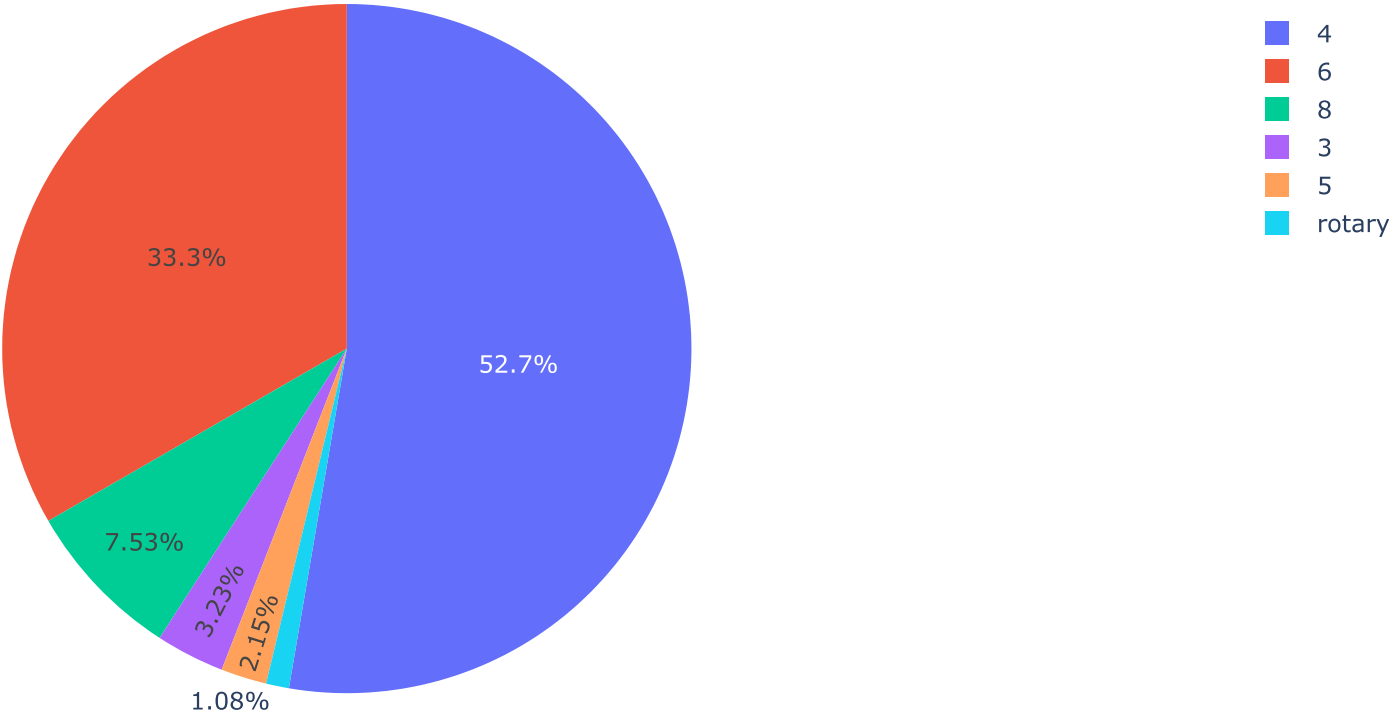
```
In [14]: data["Cylinders"].values
```

```
Out[14]: array(['4', '6', '6', '6', '4', '4', '6', '6', '6', '8', '8', '4', '4',  
        '6', '4', '6', '6', '8', '8', '6', '4', '6', '4', '4', '4', '6',  
        '4', '6', '4', '6', '4', '4', '4', '4', '4', '6', '6', '8', '3',  
        '4', '4', '4', '4', '4', '4', '4', '4', '8', '6', '6', '6', '8',  
        '4', '4', '4', '6', 'rotary', '4', '6', '4', '6', '4', '6', '4',  
        '4', '6', '6', '4', '4', '6', '6', '4', '4', '4', '6', '6', '6',  
        '4', '4', '3', '4', '4', '3', '4', '4', '4', '4', '4', '5', '4',  
        '6', '4', '5'], dtype=object)
```

```
In [15]: Ros = data["Cylinders"].value_counts()  
labelr = Ros.index  
countsr = Ros.values
```

```
In [16]: fig = go.Figure(data=[go.Pie(labels = labelr, values = countsr)])  
fig.update_layout(title = "Engines with no. of Cylinders")  
fig.show()
```

Engines with no. of Cylinders



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```