In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from warnings import filterwarnings
filterwarnings("ignore")
import seaborn as sb
```

In [2]:
```
Y1=pd.read_csv("C:/Users/ASUS/OneDrive/Desktop/Datasets in python/Cars93.csv")
Y1
```

Out[2]:

	id	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	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	25	31	NaN	...	5	177	102	68	37	26.5	11.0	2705	non-USA	Acura Integra
1	2	Acura	Legend	Midsize	29.2	33.9	38.7	18	25	Driver & Passenger	...	5	195	115	71	38	30.0	15.0	3560	non-USA	Acura Legend
2	3	Audi	90	Compact	25.9	29.1	32.3	20	26	Driver only	...	5	180	102	67	37	28.0	14.0	3375	non-USA	Audi 90
3	4	Audi	100	Midsize	30.8	37.7	44.6	19	26	NaN	...	6	193	106	70	37	31.0	17.0	3405	non-USA	Audi 100
4	5	BMW	535i	Midsize	23.7	30.0	36.2	22	30	Driver only	...	4	186	109	69	39	27.0	13.0	3640	non-USA	BMW 535i
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
88	89	Volkswagen	Eurovan	Van	16.6	19.7	22.7	17	21	NaN	...	7	187	115	72	38	34.0	NaN	3960	non-USA	Volkswagen Eurovan
89	90	Volkswagen	Passat	Compact	17.6	20.0	22.4	21	30	NaN	...	5	180	103	67	35	31.5	14.0	2985	non-USA	Volkswagen Passat
90	91	Volkswagen	Corrado	Sporty	22.9	23.3	23.7	18	25	NaN	...	4	159	97	66	36	26.0	15.0	2810	non-USA	Volkswagen Corrado
91	92	Volvo	240	Compact	21.8	22.7	23.5	21	28	Driver only	...	5	190	104	67	37	29.5	14.0	2985	non-USA	Volvo 240
92	93	Volvo	850	Midsize	24.8	26.7	28.5	20	28	Driver & Passenger	...	5	184	105	69	38	30.0	15.0	3245	non-USA	Volvo 850

93 rows × 28 columns

In [3]:
```
Y1.head()
```

Out[3]:

	id	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	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	25	31	NaN	...	5	177	102	68	37	26.5	11.0	2705	non-USA	Acura Integra
1	2	Acura	Legend	Midsize	29.2	33.9	38.7	18	25	Driver & Passenger	...	5	195	115	71	38	30.0	15.0	3560	non-USA	Acura Legend
2	3	Audi	90	Compact	25.9	29.1	32.3	20	26	Driver only	...	5	180	102	67	37	28.0	14.0	3375	non-USA	Audi 90
3	4	Audi	100	Midsize	30.8	37.7	44.6	19	26	NaN	...	6	193	106	70	37	31.0	17.0	3405	non-USA	Audi 100
4	5	BMW	535i	Midsize	23.7	30.0	36.2	22	30	Driver only	...	4	186	109	69	39	27.0	13.0	3640	non-USA	BMW 535i

5 rows × 28 columns

In [4]:
```
from preprocessing import replacer
replacer(Y1)
```

In [5]:
```
Y1.isna().sum()
```

Out[5]:
```
id                  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 [6]:
```
Y2=pd.read_csv("C:/Users/ASUS/OneDrive/Desktop/Datasets in python/Credit.csv")
Y2
```

Out[6]:

	Unnamed: 0	id	Income	Limit	Rating	Cards	Age	Education	Gender	Student	Married	Ethnicity	Balance
0	1	1	14.891	3606	283	2	34	11	Male	No	Yes	Caucasian	333
1	2	2	106.025	6645	483	3	82	15	Female	Yes	Yes	Asian	903
2	3	3	104.593	7075	514	4	71	11	Male	No	No	Asian	580
3	4	4	148.924	9504	681	3	36	11	Female	No	No	Asian	964
4	5	5	55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331
...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	396	396	12.096	4100	307	3	32	13	Male	No	Yes	Caucasian	560
396	397	397	13.364	3838	296	5	65	17	Male	No	No	African American	480
397	398	398	57.872	4171	321	5	67	12	Female	No	Yes	Caucasian	138
398	399	399	37.728	2525	192	1	44	13	Male	No	Yes	Caucasian	0
399	400	400	18.701	5524	415	5	64	7	Female	No	No	Asian	966

400 rows × 13 columns

In [7]:
```
Y2.head()
```

Out[7]:

	Unnamed: 0	id	Income	Limit	Rating	Cards	Age	Education	Gender	Student	Married	Ethnicity	Balance
0	1	1	14.891	3606	283	2	34	11	Male	No	Yes	Caucasian	333
1	2	2	106.025	6645	483	3	82	15	Female	Yes	Yes	Asian	903
2	3	3	104.593	7075	514	4	71	11	Male	No	No	Asian	580
3	4	4	148.924	9504	681	3	36	11	Female	No	No	Asian	964
4	5	5	55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331

In [8]:
```
from preprocessing import replacer
replacer(Y2)
```

In [9]:
```
Y2.isna().sum()
```

Out[9]:
```
Unnamed: 0    0
id            0
Income        0
Limit         0
Rating        0
Cards         0
Age           0
Education     0
Gender        0
Student       0
Married       0
Ethnicity     0
Balance       0
dtype: int64
```

In [10]:
```
c=pd.merge(Y1,Y2,on="id",how="inner")
```

In [11]:
```
c
```

Out[11]:

	id	Manufacturer	Model	Type	Min.Price	Price	Max.Price	MPG.city	MPG.highway	AirBags	...	Limit	Rating	Cards	Age	Education	Gender	Student	Married	Ethnicity	Balance
0	1	Acura	Integra	Small	12.9	15.9	18.8	25	31	Driver only	...	3606	283	2	34	11	Male	No	Yes	Caucasian	333
1	2	Acura	Legend	Midsize	29.2	33.9	38.7	18	25	Driver & Passenger	...	6645	483	3	82	15	Female	Yes	Yes	Asian	903
2	3	Audi	90	Compact	25.9	29.1	32.3	20	26	Driver only	...	7075	514	4	71	11	Male	No	No	Asian	580
3	4	Audi	100	Midsize	30.8	37.7	44.6	19	26	Driver only	...	9504	681	3	36	11	Female	No	No	Asian	964
4	5	BMW	535i	Midsize	23.7	30.0	36.2	22	30	Driver only	...	4897	357	2	68	16	Male	No	Yes	Caucasian	331
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
88	89	Volkswagen	Eurovan	Van	16.6	19.7	22.7	17	21	Driver only	...	4307	352	4	57	8	Male	No	Yes	African American	579
89	90	Volkswagen	Passat	Compact	17.6	20.0	22.4	21	30	Driver only	...	7518	543	3	52	9	Female	No	No	African American	1176
90	91	Volkswagen	Corrado	Sporty	22.9	23.3	23.7	18	25	Driver only	...	5797	431	4	42	16	Male	No	Yes	African American	1023
91	92	Volvo	240	Compact	21.8	22.7	23.5	21	28	Driver only	...	6040	456	3	47	16	Male	No	Yes	Caucasian	812
92	93	Volvo	850	Midsize	24.8	26.7	28.5	20	28	Driver & Passenger	...	2832	249	4	51	13	Male	No	No	Caucasian	0

93 rows × 40 columns

In [12]:
```
cat=[]
con=[]
for i in c.columns:
    if(c[i].dtypes=="object"):
        cat.append(i)
    else:
        con.append(i)
```

In [13]:
```
cat
```

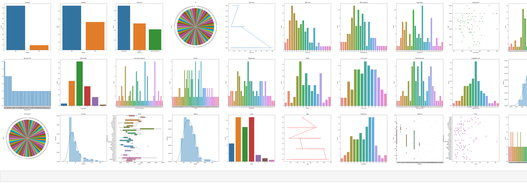
Out[13]:
```
['Manufacturer',
 'Model',
 'Type',
 'AirBags',
 'DriveTrain',
 'Cylinders',
 'Man.trans.avail',
 'Origin',
 'Make',
 'Gender',
 'Student',
 'Married',
 'Ethnicity']
```

In [14]:
```
con
```

Out[14]:
```
['id',
 'Min.Price',
 'Price',
 'Max.Price',
 'MPG.city',
 'MPG.highway',
 'EngineSize',
 'Horsepower',
 'RPM',
 'Rev.per.mile',
 'Fuel.tank.capacity',
 'Passengers',
 'Length',
 'Wheelbase',
 'Width',
 'Turn.circle',
 'Rear.seat.room',
 'Luggage.room',
 'Weight',
 'Unnamed: 0',
 'Income',
 'Limit',
 'Rating',
 'Cards',
 'Age',
 'Education',
 'Balance']
```

In [15]:
```
plt.figure(figsizer=(100,100))
plt.subplot(10,10,1)
sb.countplot(data=c,x="Manufacturer")
plt.title("Manufacturer")
plt.subplot(10,10,2)
c.Model.value_counts().plot(kind="pie")
plt.title("Model")
plt.subplot(10,10,3)
sb.countplot(data=c,x="Type")
plt.title("Type")
plt.subplot(10,10,4)
sb.countplot(data=c,x="AirBags")
plt.title("Airbags")
plt.subplot(10,10,5)
sb.countplot(data=c,x="DriveTrain")
plt.title("DriveTrain")
plt.subplot(10,10,6)
sb.countplot(data=c,x="Cylinders")
plt.title("Cylinders")
plt.subplot(10,10,7)
sb.countplot(data=c,x="Man.trans.avail")
plt.title("Man.trans.avail")
plt.subplot(10,10,8)
sb.countplot(data=c,x="Origin")
plt.title("Origin")
plt.subplot(10,10,9)
c.Make.value_counts().plot(kind="pie")
plt.title("Make")
plt.subplot(10,10,10)
c.Gender.hist()
plt.title("Gender")
plt.subplot(10,10,11)
sb.countplot(data=c,x="Student")
plt.title("Student")
plt.subplot(10,10,12)
sb.countplot(data=c,x="Married")
plt.title("Married")
plt.subplot(10,10,13)
sb.countplot(data=c,x="Ethnicity")
plt.title("Ethnicity")
plt.subplot(10,10,14)
c.id.value_counts().plot(kind="pie")
plt.title("id")
plt.subplot(10,10,15)
c["Min.Price"].value_counts().plot(kind="line")
plt.title("Min.Price")
plt.subplot(10,10,16)
sb.countplot(x=c['MPG.city'],data=c)
plt.title("MPG.city")
plt.subplot(10,10,17)
sb.countplot(x=c['MPG.highway'],data=c)
plt.title("MPG.highway")
plt.subplot(10,10,18)
sb.countplot(data=c,x="EngineSize")
plt.title("EngineSize")
plt.subplot(10,10,19)
sb.countplot(data=c,x="Horsepower",y="RPM",c="green")
plt.title("Horsepower")
plt.subplot(10,10,20)
sb.countplot(data=c,x="RPM")
plt.title("RPM")
plt.subplot(10,10,21)
c["Rev.per.mile"].value_counts().plot(kind="bar")
plt.title("Rev.per.mile")
plt.subplot(10,10,22)
sb.countplot(data=c,x="Passengers")
plt.title("Passengers")
plt.subplot(10,10,23)
sb.countplot(x=c['Fuel.tank.capacity'],data=c)
plt.title("Fuel.tank.capacity")
plt.subplot(10,10,24)
sb.countplot(data=c,x="Length")
plt.title("Length")
plt.subplot(10,10,25)
sb.countplot(data=c,x="Wheelbase")
plt.title("Wheelbase")
plt.subplot(10,10,26)
sb.countplot(data=c,x="Width")
plt.title("Width")
plt.subplot(10,10,27)
sb.countplot(x=c['Turn.circle'],data=c)
plt.title("Turn.circle")
plt.subplot(10,10,28)
sb.countplot(x=c['Rear.seat.room'],data=c)
plt.title("Rear.seat.room")
plt.subplot(10,10,29)
sb.countplot(x=c['Luggage.room'],data=c)
plt.title("Luggage.room")
plt.subplot(10,10,30)
sb.distplot(c.weight)
plt.title("weight")
plt.subplot(10,10,31)
c["Unnamed: 0"].value_counts().plot(kind="pie")
plt.title("Unnamed:0")
plt.subplot(10,10,32)
sb.distplot(c.Income)
plt.title("Income")
plt.subplot(10,10,33)
sb.boxplot(data=c,x="Limit",y="Manufacturer")
plt.title("Limit")
plt.subplot(10,10,34)
sb.distplot(c.Rating)
plt.title("Rating")
plt.subplot(10,10,35)
sb.countplot(data=c,x="Cards")
plt.title("Cards")
plt.subplot(10,10,36)
c.Age.value_counts().plot(kind="line",c="red")
plt.title("Age")
plt.subplot(10,10,37)
sb.countplot(data=c,x="Education")
plt.title("Education")
plt.subplot(10,10,38)
sb.boxplot(data=c,x="Balance",y="Age")
plt.title("Balance")
plt.subplot(10,10,39)
sb.scatterplot(data=c,x="Price",y="Manufacturer",c="purple")
plt.title("Price")
plt.subplot(10,10,40)
sb.countplot(x=c['Max.Price'],data=c)
plt.title("Max.Price")
```

Out[16]:
```
Text(0.5, 1.0, 'Max.Price')
```



In [ ]: