

```
In [15]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [16]: cs=pd.read_csv('Car Sales.xlsx - car_data.csv')
```

Inroduction of Data

```
In [17]: cs
Out[17]:
```

	Car_id	Date	Customer Name	Gender	Annual Income	Dealer_Name	Company	Model	Engine	Transmission	Color	Price (\$)	Dealer_No	Body Style	Phone	Dealer_Region
0	C_CND_000001	12/2022	Geradine	Male	13500	Buddy Storbeck's Diesel Service Inc	Ford	Expedition	DoubleA Overhead Camshaft	Auto	Black	26000	06457-3834	SLV	8254678	Middletown
1	C_CND_000002	12/2022	Gia	Male	140000	C & M Motors Inc	Dodge	Durango	DoubleA Overhead Camshaft	Auto	Black	19000	60504-7114	SLV	6848189	Aurora
2	C_CND_000003	12/2022	Gianna	Male	1035000	Capitol KIA	Cadillac	Eldorado	Overhead Camshaft	Manual	Red	31500	38701-8847	Passenger	7298798	Greenville
3	C_CND_000004	12/2022	Giselle	Male	13500	Chrysler of Tri-Cities	Toyota	Celica	Overhead Camshaft	Manual	Pale White	14000	99301-3882	SLV	6257567	Pasco
4	C_CND_000005	12/2022	Grace	Male	1465000	Chrysler Plymouth	Acura	TL	DoubleA Overhead Camshaft	Auto	Red	24500	53546-9427	Hatchback	7081483	Janeville
...
23901	C_CND_023902	12/31/2023	Martin	Male	13500	C & M Motors Inc	Plymouth	Voyager	Overhead Camshaft	Manual	Red	12000	60504-7114	Passenger	8583598	Pasco
23902	C_CND_023903	12/31/2023	Jimmy	Female	900000	Ryder Truck Rental and Leasing	Chevrolet	Prizm	DoubleA Overhead Camshaft	Auto	Black	16000	06457-3834	Hardtop	7914229	Middletown
23903	C_CND_023904	12/31/2023	Emma	Male	705000	Chrysler of Tri-Cities	BMW	328i	Overhead Camshaft	Manual	Red	21000	89301-3882	Sedan	7659127	Scottsdale
23904	C_CND_023905	12/31/2023	Victore	Male	13500	Chrysler Plymouth	Chevrolet	Metro	DoubleA Overhead Camshaft	Auto	Black	31000	53546-9427	Passenger	6030764	Austin
23905	C_CND_023906	12/31/2023	Donovan	Male	1225000	Pars Auto Sales	Lexus	ES300	DoubleA Overhead Camshaft	Auto	Pale White	27500	38701-8847	Hardtop	7020564	Middletown

23906 rows × 16 columns

```
In [18]: cs.isnull().sum()
Out[18]:
Car_id      0
Date         0
Customer Name  0
Gender       0
Annual Income  0
Dealer Name  0
Company      0
Model        0
Engine       0
Transmission  0
Color        0
Price ($)    0
Dealer_No    0
Body Style   0
Phone        0
Dealer_Region 0
dtype: int64
```

```
In [19]: cs.dtypes
Out[19]:
Car_id      object
Date        object
Customer Name  object
Gender       object
Annual Income  int64
Dealer Name  object
Company      object
Model        object
Engine       object
Transmission  object
Color        object
Price ($)    int64
Dealer_No    object
Body Style   object
Phone        int64
Dealer_Region  object
dtype: object
```

```
In [24]: cs.rename(columns={'Price ($)': 'Price'}, inplace=True)
In [25]: cs.columns
Out[25]:
Index(['Car_id', 'Date', 'Customer Name', 'Gender', 'Annual Income', 'Dealer_Name', 'Company', 'Model', 'Engine', 'Transmission', 'Color', 'Price', 'Dealer_No', 'Body Style', 'Phone', 'Dealer_Region'],
      dtype='object')
```

Data Analysis

```
In [26]: cs.describe()
Out[26]:
```

	Annual Income	Price	Phone
count	2.390600e+04	23906.000000	2.390600e+04
mean	8.308403e+05	28090.247846	7.497741e+06
std	7.200604e+05	14788.857608	8.674930e+05
min	1.000000e+04	1200.000000	6.000000e+06
25%	3.880000e+05	10000.000000	6.746456e+06
50%	7.350000e+05	23000.000000	7.496158e+06
75%	1.175750e+06	34000.000000	8.248146e+06
max	1.120000e+07	85800.000000	8.999579e+06

```
In [27]: avg_prices['Price'].mean()
Out[27]:
28090.247845729107
```

```
In [28]: ##Avg_car_price = 28090
In [29]: cs.columns
```

```
Index(['Car_id', 'Date', 'Customer Name', 'Gender', 'Annual Income', 'Dealer_Name', 'Company', 'Model', 'Engine', 'Transmission', 'Color', 'Price', 'Dealer_No', 'Body Style', 'Phone', 'Dealer_Region'],
      dtype='object')
```

```
In [30]: ## How many cars have manual and automatic transmission?
In [31]: cs.groupby(['Model'])['Transmission'].value_counts()
```

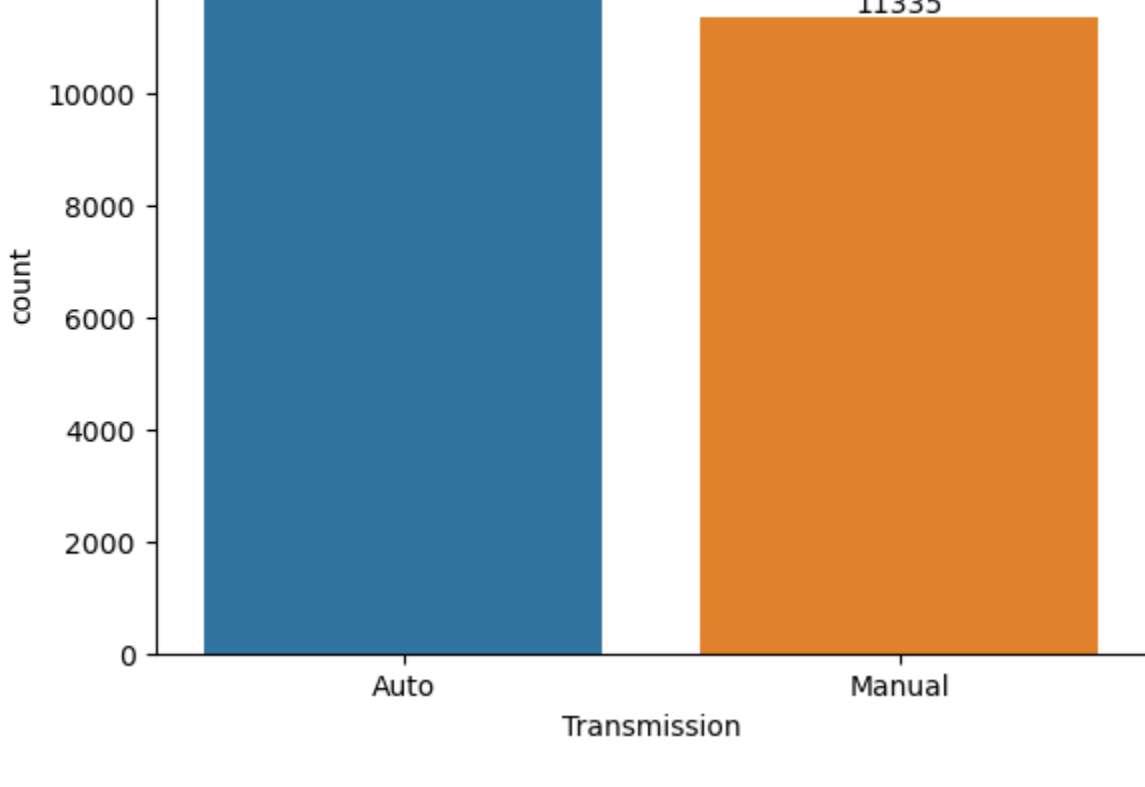
```
Out[31]:
Model      Transmission      count
3-Sep      Auto              171
3888GT      Auto              156
Engine      Manual           147
3888M       Auto             243
323L        Auto             223

Windstar    Manual           22
Wrangler     Auto             25
Wrangler     Manual           69
Xterra       Auto             74
Xterra       Manual           18
Name: Transmission, Length: 261, dtype: int64
```

```
In [32]: transmission_counts = cs['Transmission'].value_counts()
transmission_counts
```

```
Out[32]:
Auto      12571
Manual    11335
Name: Transmission, dtype: int64
```

```
In [33]: ax=sns.countplot(data=cs,x='Transmission')
For bars in ax.containers:
ax.bar_label(bars)
```

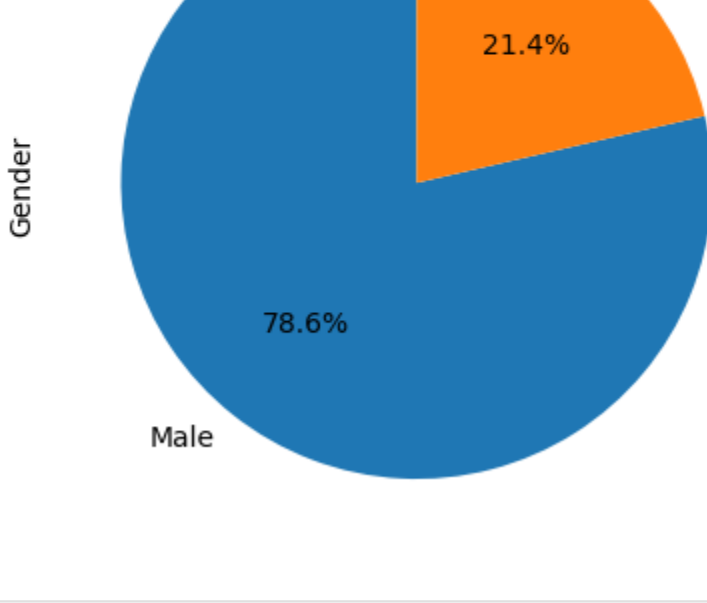


```
In [34]: ## Gender count.
```

```
In [35]: gender_counts =cs['Gender'].value_counts()
gender_counts
```

```
Out[35]:
Male      18798
Female    5108
Name: Gender, dtype: int64
```

```
In [36]: gender_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90)
plt.title('Male vs Female')
plt.show()
```



```
In [37]: ## Top 10 companies
```

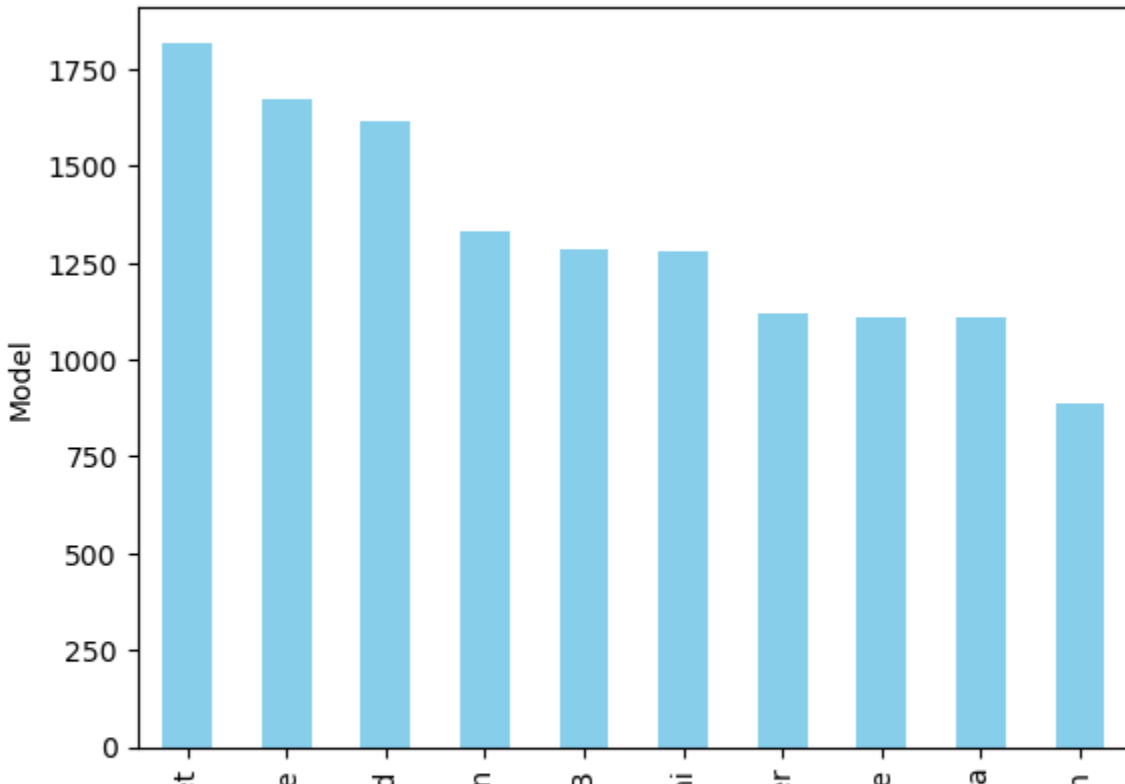
```
In [38]: cs.columns
Out[38]:
Index(['Car_id', 'Date', 'Customer Name', 'Gender', 'Annual Income', 'Dealer_Name', 'Company', 'Model', 'Engine', 'Transmission', 'Color', 'Price', 'Dealer_No', 'Body Style', 'Phone', 'Dealer_Region'],
      dtype='object')
```

```
In [39]: top_selling=cs.groupby(['Company'], as_index=False)['Model'].count().sort_values(by='Model',ascending=False).head(10)
top_selling
```

```
Out[39]:
```

	Company	Model
5	Chevrolet	1819
7	Dodge	1671
8	Ford	1614
28	Volkswagen	1333
16	Mercedes-B	1295
18	Mitsubishi	1277
6	Chrysler	1120
30	Oldsmobile	1111
17	Toyota	1110
19	Nissan	886

```
In [40]: top_selling.plot(kind='bar', x='Company', y='Model', legend=False, color='skyblue')
plt.xlabel('Company')
plt.ylabel('Model')
plt.title('Total Sales by Car Model')
plt.show()
```



```
In [41]: ## how many different car models are there in the dataset?
```

```
In [42]: model=cs['Model'].nunique()
model
```

```
Out[42]:
154
```

```
In [43]: ## Top 10 regions with most number of sales?
```

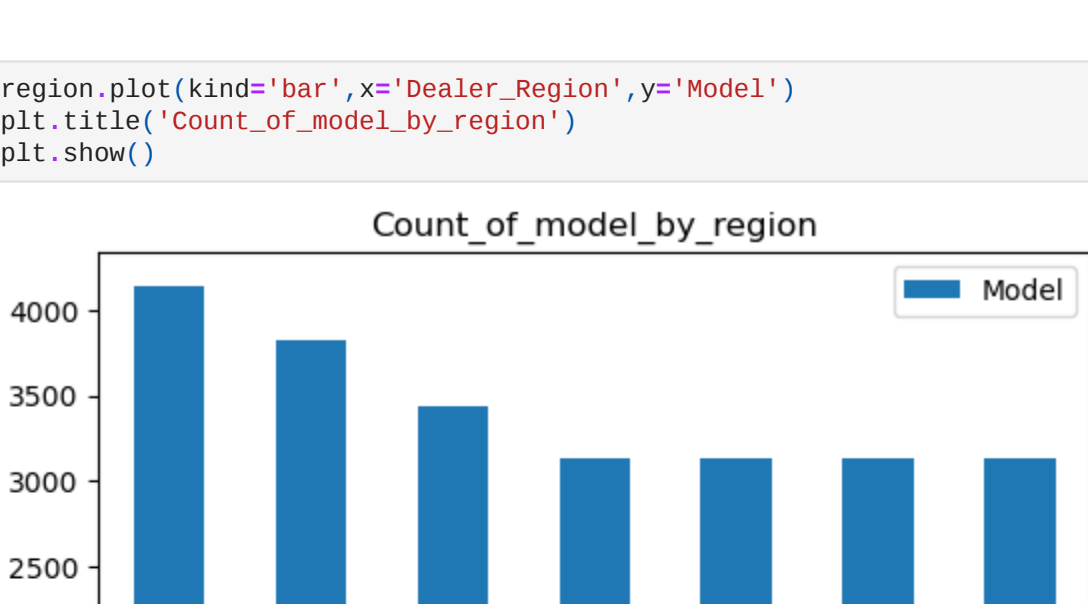
```
In [44]: cs.columns
Out[44]:
Index(['Car_id', 'Date', 'Customer Name', 'Gender', 'Annual Income', 'Dealer_Name', 'Company', 'Model', 'Engine', 'Transmission', 'Color', 'Price', 'Dealer_No', 'Body Style', 'Phone', 'Dealer_Region'],
      dtype='object')
```

```
In [45]: regionlms.groupby(['Dealer_Region'],as_index=False)['Model'].count().sort_values(by='Model',ascending=False)
In [46]: region
```

```
Out[46]:
```

	Dealer_Region	Model
1	Austin	4135
3	Janesville	3821
6	Scottsdale	3433
5	Pasco	3131
9	Aurora	3130
2	Greenville	3128
4	Middletown	3128

```
In [47]: region.plot(kind='bar',x='Dealer_Region',y='Model')
plt.title('Count_of_model_by_region')
plt.show()
```



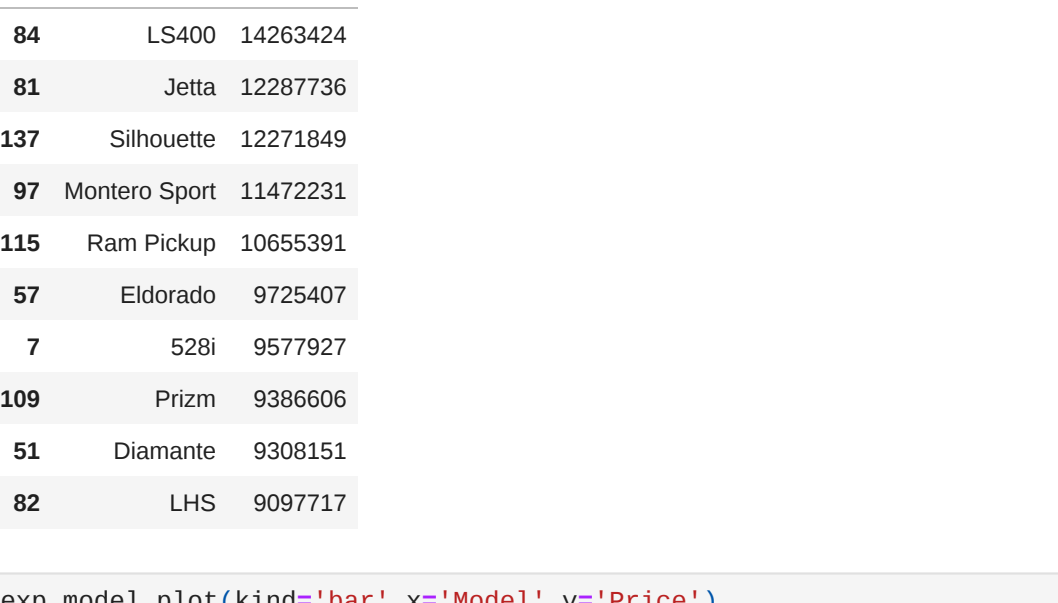
```
In [48]: cs.columns
Out[48]:
Index(['Car_id', 'Date', 'Customer Name', 'Gender', 'Annual Income', 'Dealer_Name', 'Company', 'Model', 'Engine', 'Transmission', 'Color', 'Price', 'Dealer_No', 'Body Style', 'Phone', 'Dealer_Region'],
      dtype='object')
```

```
In [49]: ## Top 10 expensive car model?
In [50]: exp_model=cs.groupby(['Model'], as_index = False)['Price'].sum().sort_values(by='Price', ascending=False).head(10)
exp_model
```

```
Out[50]:
```

	Model	Price
84	LS400	14053404
81	Jetta	12087736
137	Silhouette	12215949
87	Montero Sport	11472231
115	Ram Pickup	10655391
57	Eldorado	9725407
7	S28	9577927
108	Prizm	9386606
51	Diamante	9308151
82	LHS	9097717

```
In [51]: exp_model.plot(kind='bar',xs='Model',y='Price')
plt.title('Most expensive model')
plt.show()
```

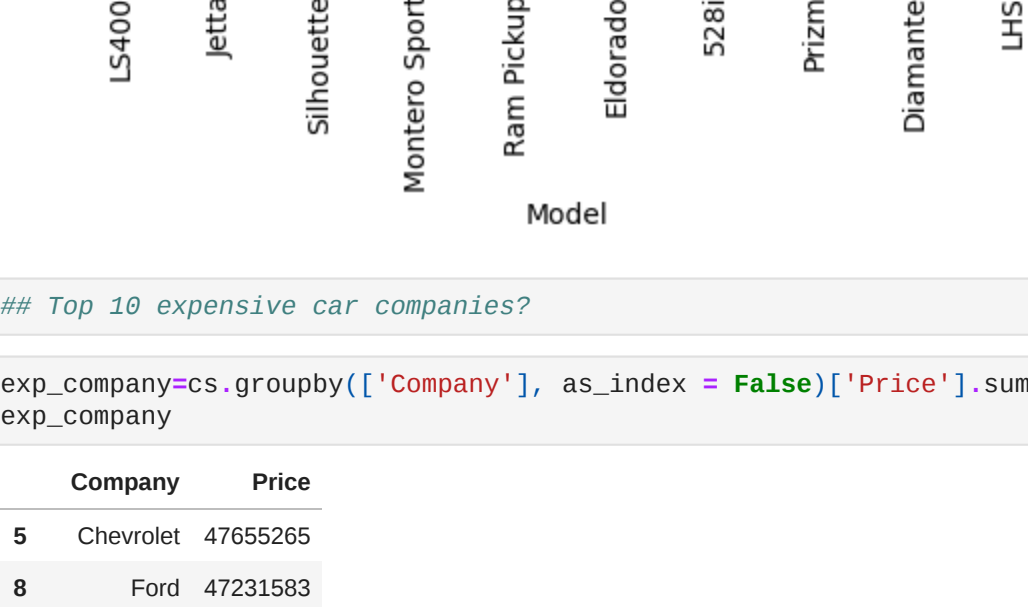


```
In [52]: ## Top 10 expensive car companies?
In [53]: exp_company=cs.groupby(['Company'], as_index = False)['Price'].sum().sort_values(by='Price', ascending=False).head(10)
exp_company
```

```
Out[53]:
```

	Company	Price
5	Chevrolet	47655265
8	Ford	47223583
7	Dodge	44124966
20	Oldsmobile	35434512
16	Mercedes-B	34624123
28	Volkswagen	34062981
18	Mitsubishi	34062566
27	Toyota	32545644
6	Chrysler	29141873
14	Lexus	27287703

```
In [54]: exp_company.plot(kind='bar',xs='Company',y='Price')
plt.title('Most expensive car company')
plt.show()
```



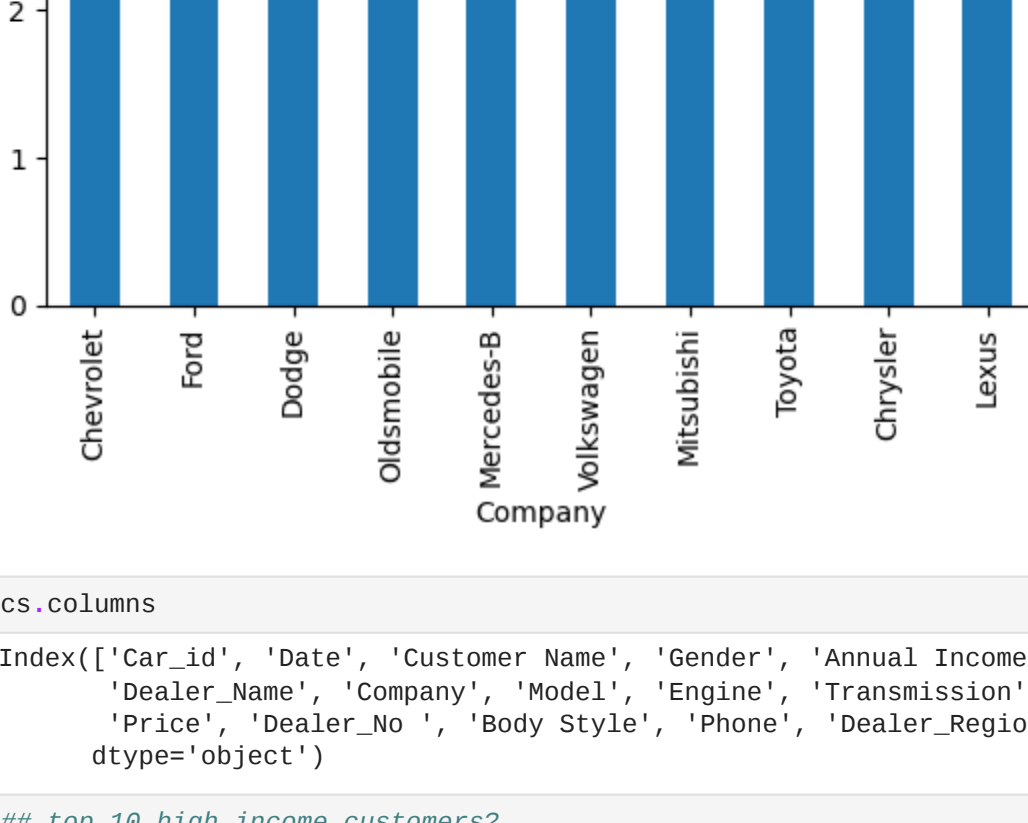
```
In [55]: cs.columns
Out[55]:
Index(['Car_id', 'Date', 'Customer Name', 'Gender', 'Annual Income', 'Dealer_Name', 'Company', 'Model', 'Engine', 'Transmission', 'Color', 'Price', 'Dealer_No', 'Body Style', 'Phone', 'Dealer_Region'],
      dtype='object')
```

```
In [56]: ## top 10 high income customers?
In [60]: high_income=cs.groupby(['Customer Name'], as_index=False)['Annual Income'].sum().sort_values(by='Annual Income',ascending=False).head(10)
high_income
```

```
Out[60]:
```

	Customer Name	Annual Income
862	Emma	78304500
1790	Lucas	74465688
2209	Nathan	69170113
2769	Thomas	69126000
1778	Louis	67201700
1665	Leo	66647388
1644	Lee	65280000
598	Chloe	59854750
115	Alexis	59847000
2552	Sarah	57486750

```
In [63]: high_income.plot(kind='bar',xs='Customer Name',y='Annual Income')
plt.show()
```



```
In [ ]:
```