Importing necessary modules

In [1]:
```
import pandas as pd
```

In [2]:
```
car=pd.read_csv(r"K:\Desktop\NIIT\tables\DS1_C4_S4_Car_Prices_Data_Challenge.csv")
car
```

Out[2]:

	year	make	model	trim	body	transmission	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice	saledate
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5.0	16639.0	white	black	kia motors america, inc	20500	21500	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
1	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5.0	9393.0	white	beige	kia motors america, inc	20800	21500	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
2	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	4.5	1331.0	gray	black	financial services remarketing (lease)	31900	30000	Thu Jan 15 2015 04:30:00 GMT-0800 (PST)
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	4.1	14282.0	white	black	volvo na rep/world omni	27500	27750	Thu Jan 29 2015 04:30:00 GMT-0800 (PST)
4	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	4.3	2641.0	gray	black	financial services remarketing (lease)	66000	67000	Thu Dec 18 2014 12:30:00 GMT-0800 (PST)
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
558832	2015	Kia	K900	Luxury	Sedan	NaN	knalw4d4xf6019304	in	4.5	18255.0	silver	black	avis corporation	35300	33000	Thu Jul 09 2015 07:00:00 GMT-0700 (PDT)
558833	2012	Ram	2500	Power Wagon	Crew Cab	automatic	3c6td5et6cg112407	wa	5.0	54393.0	white	black	i -5 uhlmann rv	30200	30800	Wed Jul 08 2015 09:30:00 GMT-0700 (PDT)
558834	2012	BMW	X5	xDrive35d	SUV	automatic	5uxzw0c58cl668465	ca	4.8	50561.0	black	black	financial services remarketing (lease)	29800	34000	Wed Jul 08 2015 09:30:00 GMT-0700 (PDT)
558835	2015	Nissan	Altima	2.5 S	sedan	automatic	1n4al3ap0fc216050	ga	3.8	16658.0	white	black	enterprise vehicle exchange / tra / rental / t...	15100	11100	Thu Jul 09 2015 06:45:00 GMT-0700 (PDT)
558836	2014	Ford	F-150	XLT	SuperCrew	automatic	1ftfw1et2eke87277	ca	3.4	15008.0	gray	gray	ford motor credit company,llc pd	29600	26700	Thu May 28 2015 05:30:00 GMT-0700 (PDT)

558837 rows × 16 columns

Task1: To identify cars made in 2014 and lowest price car makers

In [3]:
```
pd.pivot_table(car[(car.year==2014)],index="make",values=["sellingprice"]
               ,aggfunc="mean").sort_values("sellingprice").head()
```

Out[3]:

make	sellingprice
smart	8620.000000
FIAT	12585.109290
Mitsubishi	13502.323838
Kia	14341.490037
Volkswagen	14840.196850

Task2: To help compare bmw cars sold in 2010 and 2015 which have condition score above 4.5 and find relative percentage in sold numbers

In [4]:
```
BMW_1510=car[(car.make=="BMW")&(car.condition>=4.5)&((car.year==2010)|(car.year==2015))]
sales=pd.pivot_table(BMW_1510,index="year",values=["sellingprice"]).sort_values("sellingprice",ascending=False)
relative_percent=(sales.iloc[0,:]-sales.iloc[1,:])/(sales.iloc[1,:])*100
print("The relative percent of increase in sales in 2015 is ",round(relative_percent,2)[0],"%")
print()
sales
```

```
The relative percent of increase in sales in 2015 is  85.22 %
```

Out[4]:

year	sellingprice
2015	43096.770601
2010	23267.441860

Task3: To compare the average wholesale price mmr of cars with grey and black color interior

In [5]:
```
interior=car[(car.interior=="black")|(car.interior=="gray")]
int_bg=pd.pivot_table(interior,index="interior",values=["mmr"])
print("The difference in mmr for black and gray color interior cars :",(int_bg.iloc[0,:]-int_bg.iloc[1,:])[0])
int_bg
```

```
The difference in mmr for black and gray color interior cars : 4668.743510266328
```

Out[5]:

interior	mmr
black	15858.520055
gray	11189.776545

Task4: To find total cars sold based on color from 2013 to 2015

In [6]:
```
color_b=car[(car.year>=2013)&(car.year<=2015)]
pd.crosstab(color_b.make,color_b.color)
```

Out[6]:

color / make	beige	black	blue	brown	burgundy	charcoal	gold	gray	green	lime	off-white	orange	pink	purple	red	silver	turquoise	white	yellow	—
Acura	0	170	19	5	20	2	0	107	1	0	3	0	0	1	8	61	0	98	0	3
Audi	2	402	78	10	1	0	1	181	0	0	0	0	0	0	29	183	0	313	0	85
BMW	34	1235	252	99	1	1	8	710	4	0	2	14	0	1	139	190	0	1101	0	297
Bentley	0	5	3	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	2
Buick	21	157	55	25	12	1	19	189	0	0	11	0	0	6	75	235	0	308	0	13
Cadillac	55	298	36	12	2	0	50	85	1	0	11	0	12	1	48	196	0	141	0	13
Chevrolet	421	3570	1582	153	167	36	354	2933	160	1	24	52	13	25	2154	4474	1	5143	75	343
Chrysler	175	1599	848	56	232	18	334	1343	2	0	4	0	0	0	788	860	0	1657	0	183
Dodge	129	2412	1722	15	100	40	311	2304	16	1	18	135	1	14	1530	1251	1	2601	3	300
FIAT	7	123	37	12	0	9	0	91	55	0	0	7	0	0	88	45	0	116	23	17
Ford	55	7102	2209	550	444	12	70	5076	848	2	87	162	0	36	4474	5101	13	9969	18	6465
GMC	39	438	91	31	30	5	37	214	8	0	6	0	0	2	127	338	0	1150	0	29
Honda	25	859	363	137	156	2	28	859	15	0	10	3	0	14	173	513	3	600	0	60
Hyundai	129	1666	1502	324	83	4	95	1835	32	1	14	48	0	10	1227	2059	0	1736	10	186
Infiniti	3	1424	300	20	49	0	1	1452	35	0	14	0	0	32	101	371	0	1098	0	351
Jaguar	0	47	13	0	2	1	0	8	3	0	0	0	0	0	13	14	0	38	0	8
Jeep	34	1105	338	45	91	14	28	813	66	3	1	59	0	1	504	365	1	912	21	108
Kia	97	1964	571	495	429	2	29	1706	288	0	30	0	0	87	969	1754	0	2027	89	161
Land Rover	3	134	10	8	1	0	1	45	4	0	0	1	0	0	8	21	0	116	0	10
Lexus	37	798	114	74	41	12	60	426	0	0	28	0	0	5	166	420	0	515	0	84
Lincoln	13	443	51	61	29	1	6	177	19	0	30	3	0	4	181	192	0	287	0	571
MINI	5	84	119	17	0	0	1	47	54	0	10	6	0	0	111	31	0	144	1	27
Maserati	0	10	3	2	0	0	0	2	0	0	0	0	0	0	1	3	0	8	0	2
Mazda	1	517	231	4	13	9	1	821	74	0	3	1	0	0	325	228	3	392	0	47
Mercedes-Benz	0	1410	73	5	1	1	0	538	2	0	0	0	0	2	132	401	0	971	0	685
Mitsubishi	25	246	148	22	4	0	0	205	18	0	0	3	3	4	119	217	2	280	0	83
Nissan	170	3357	1015	298	316	6	62	2953	16	0	54	2	0	126	1409	2704	5	3683	4	1796
Porsche	0	115	32	19	0	0	0	47	1	0	0	0	0	1	29	26	0	85	4	7
Ram	2	567	164	52	39	1	2	271	5	0	0	26	0	0	350	325	0	815	0	49
Rolls-Royce	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Scion	2	79	44	4	8	0	0	79	0	0	3	12	0	4	58	51	0	80	0	9
Subaru	52	157	492	22	40	2	43	370	118	0	5	18	0	3	207	334	0	421	1	60
Suzuki	2	4	8	3	0	0	0	10	0	0	0	0	0	0	2	9	0	14	0	0
Tesla	0	5	1	0	0	0	0	8	1	0	0	0	0	0	2	2	0	3	0	1
Toyota	93	2074	796	51	103	7	153	2634	250	0	4	28	0	13	1393	2437	6	2357	1	351
Volkswagen	44	685	338	30	69	0	38	739	0	0	1	0	0	4	242	549	0	605	9	354
Volvo	11	226	65	10	3	1	12	164	0	0	2	3	0	1	23	154	0	219	0	22
buick	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
chevrolet	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
dodge	0	15	6	0	0	0	0	3	0	0	0	0	0	0	1	4	0	12	0	0
jeep	0	11	3	0	0	0	0	1	0	0	0	0	0	0	4	0	0	8	0	0
land rover	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
lincoln	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
nissan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
smart	0	31	9	1	0	0	0	13	0	0	0	0	0	0	18	23	0	30	3	5
volkswagen	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
vw	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	10

Task5: To find total sales of cars on particular dates and body type as SUV

In [7]:
```
suv=car[(car.body=="SUV")&((car.saledate=="Fri Dec 19 2014 09:00:00 GMT-0800 (PST)")|(car.saledate=="Thu Jan 15 2015 04:30:00 GMT-0800 (PST)"))]
print("The total sales of SUV cars on corresponding dates = ",len(suv))
y2014=len(suv[suv.saledate=="Fri Dec 19 2014 09:00:00 GMT-0800 (PST)"])
y2015=len(suv[suv.saledate=="Thu Jan 15 2015 04:30:00 GMT-0800 (PST)"])
print("Relative difference percent = ",round((y2015-y2014)/y2015 * 100,2),"%")
```

```
The total sales of SUV cars on corresponding dates =  879
Relative difference percent =  -83.55 %
```

Task6: To create a column discounted price and give 20% for chevrolet and ford and 10% for others

In [8]:
```
car["discounted_price"]=car[(car.make=="Ford")|(car.make=="chevrolet")].sellingprice*0.80
car.discounted_price=car.discounted_price.fillna(car.sellingprice*0.90)
car.iloc[:,16:17]
```

Out[8]:

	discounted_price
0	19350.0
1	19350.0
2	27000.0
3	24975.0
4	60300.0
...	...
558832	29700.0
558833	27720.0
558834	30600.0
558835	9990.0
558836	21360.0

558837 rows × 1 columns

Task7: To drop saledate and vin and replace all missing values of dataframe

In [9]:
```
    car=car.drop(["saledate","vin"],axis=1)
    print(car.columns)
    car.isna().sum()
    for item in car.columns:
        car[item].fillna(str(car[item].mode()),inplace=True)
    car.isna().sum()
```

```
Index(['year', 'make', 'model', 'trim', 'body', 'transmission', 'state',
       'condition', 'odometer', 'color', 'interior', 'seller', 'mmr',
       'sellingprice', 'discounted_price'],
      dtype='object')
```

Out[9]:
```
year               0
make               0
model              0
trim               0
body               0
transmission       0
state              0
condition          0
odometer           0
color              0
interior           0
seller             0
mmr                0
sellingprice       0
discounted_price   0
dtype: int64
```