define function for getting the number of in particular seller ,maker ,body . as you want

In [1]:
```
#(data['make'].value_counts()[0:6:]).plot(kind='pie',autopct="%0.1f%%")
import pandas as pd
```

In [2]:
```
df=pd.read_csv("C:/Users/ROCKSTAR/Desktop/final_1.csv")
```

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

Out[3]:

	Unnamed: 0	year	make	model	trim	body	transmission	state	condition	odometer	c
0	0	2014	Kia	sorento	LX	suv	automatic	ca	5.0	16639.0	v
1	1	2014	Kia	sorento	LX	suv	automatic	ca	5.0	9393.0	v
2	2	2015	Volvo	s60	T5	sedan	automatic	ca	41.0	14282.0	v
3	3	2014	Nissan	altima	2.5 S	sedan	automatic	ca	1.0	5554.0	
4	4	2014	Kia	optima	LX	sedan	automatic	ca	48.0	2034.0	

In [112]:
```
#df[df['year'].isnull()]
```

In [113]:
```
#df['month'] = df['saledate'].str.split(' ').str[1] to create new column
```

In [114]:
```
#df['year'] = df['year'].astype(int) converting data type
```

In [ ]:
```

```

In [115]:
```
#def sell_maker_body(maker,model,Body):
#    data = df[(df['make']==maker) & (df['model']==model) & (df['body']==Bo
3   # return data.shape
```

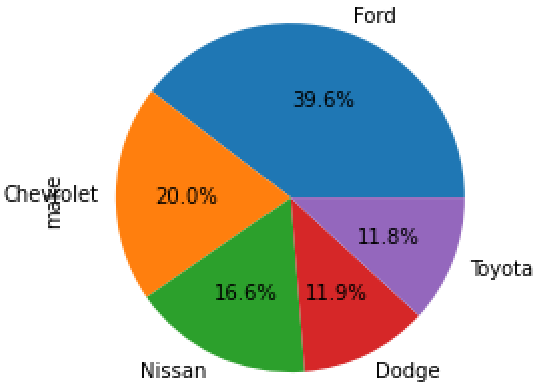
```
  Input In [115]
    3    return data.shape
         ^
SyntaxError: invalid syntax
```

```
In [116]: #sell_maker_body('Kia','sorento','suv')
```

top five maker

```
In [4]: (df['make'].value_counts()[0:5:]).plot(kind='pie',autopct="%0.1f%%")
```

```
Out[4]: <AxesSubplot:ylabel='make'>
```



```
In [5]: df['make'].value_counts()[0:5:]
```

```
Out[5]: Ford         42819
        Chevrolet    21688
        Nissan       18001
        Dodge        12911
        Toyota       12756
        Name: make, dtype: int64
```

top five model

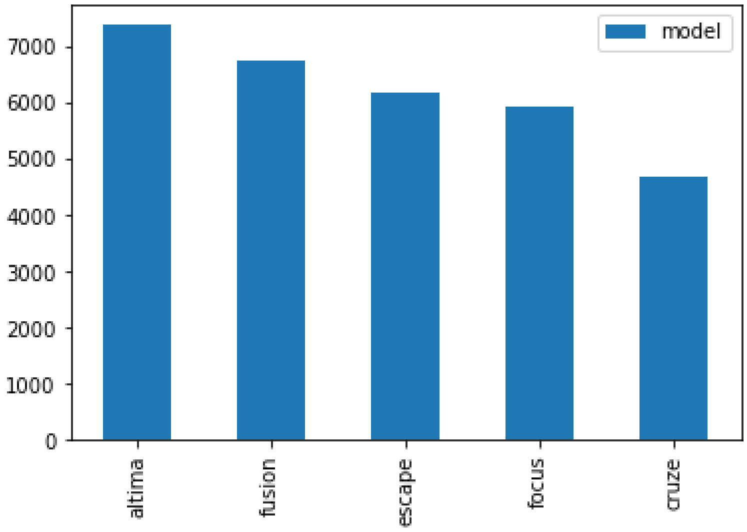
```
In [6]: df['model'].value_counts()[0:5:]
```

```
Out[6]: altima    7388
        fusion    6741
        escape    6172
        focus     5923
        cruze     4706
        Name: model, dtype: int64
```

In [7]:

```
data=df['model'].value_counts()[0:5:]
df= pd.DataFrame(data)
df.plot(kind='bar')
plt.show()
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Input In [7], in <cell line: 4>()
      2 df= pd.DataFrame(data)
      3 df.plot(kind='bar')
----> 4 plt.show()

NameError: name 'plt' is not defined
```



top five body

```
In [9]: df['body']
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3621, in In
dex.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._
libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._
libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'body'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
Input In [9], in <cell line: 1>()
----> 1 df['body']

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3505, in DataFram
e.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3623, in In
dex.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'body'
```

```
In [125]: #(df['body'].value_counts()[0:5:]).plot(kind='pie',autopct="0.1f%%")
```

In [127]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = df['body'].value_counts()[0:5:]

# Create DataFrame
df = pd.DataFrame(data)



# Plot bar chart
df.plot(kind='bar')

# Show the plot
plt.show()
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3621, in In
dex.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._
libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._
libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'body'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
Input In [127], in <cell line: 5>()
      2 import matplotlib.pyplot as plt
      4 # Sample data
----> 5 data = df['body'].value_counts()[0:5:]
      7 # Create DataFrame
      8 df = pd.DataFrame(data)

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3505, in DataFram
e.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3623, in In
dex.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'body'
```

top five state

In [ ]:

```
```

In [10]:

```
df['state'].value_counts()[0:5:]
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3621, in In
dex.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._
libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._
libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'state'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
Input In [10], in <cell line: 1>()
----> 1 df['state'].value_counts()[0:5:]

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3505, in DataFram
e.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3623, in In
dex.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'state'
```

year

In [128]:

```
df['year'].value_counts()
```

Out[128]:

```
2015    176548
2014     12091
0           36
Name: year, dtype: int64
```

top five rating

In [129]:

```
df['condition'].value_counts()
```

Out[129]:

```
44.0    13652
4.0     12768
43.0    12554
42.0    11467
41.0    10035
49.0     9317
37.0     9178
5.0      8849
48.0     8336
39.0     8126
35.0     7667
46.0     7630
36.0     7529
38.0     7077
47.0     7043
45.0     6992
34.0     4423
3.0      4397
29.0     3594
2.0      3512
28.0     3350
33.0     2631
27.0     2501
32.0     2325
19.0     2052
31.0     2037
1.0      1893
26.0     1733
25.0     1711
24.0     1097
23.0      876
21.0      844
0.0       781
22.0      636
17.0       14
18.0       14
16.0       10
15.0        8
11.0        5
14.0        5
13.0        3
12.0        3
Name: condition, dtype: int64
```

top exterior color

```
In [11]: df['color'].value_counts()[0:5:]
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3621, in In
dex.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._
libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._
libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'color'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
Input In [11], in <cell line: 1>()
----> 1 df['color'].value_counts()[0:5:]

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3505, in DataFram
e.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3623, in In
dex.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'color'
```

top interior color

```
In [131]: df['interior'].value_counts()[0:5:]
```

```
Out[131]: black    109331
          gray      46852
          beige     14936
          tan        9713
          —          3522
          Name: interior, dtype: int64
```

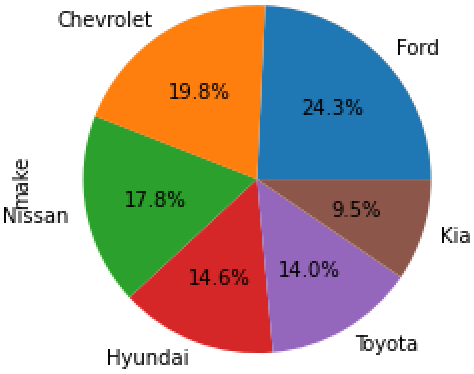
checking the which top body type (sedan) with automatic transmission bulid by which maker and its count

```
In [132]: data=df[(df['body']=='sedan') & (df['transmission']=='automatic')]
          data['make'].value_counts()[0:4:]
```

```
Out[132]: Ford         13013
          Chevrolet    10639
          Nissan        9553
          Hyundai       7810
          Name: make, dtype: int64
```

```
In [133]: (data['make'].value_counts()[0:6:]).plot(kind='pie',autopct="%0.1f%%")
```

```
Out[133]: <AxesSubplot:ylabel='make'>
```



knowing the top makers and top 5 body type for which maker build highest particular body type with automatic transmission

```
In [134]: data=df[(df['body']=='sedan') & (df['transmission']=='automatic')]         #
          data['make'].value_counts()[0:2:]
```

```
Out[134]: Ford         13013
          Chevrolet    10639
          Name: make, dtype: int64
```

```
In [135]: data=df[(df['body']=='suv') & (df['transmission']=='automatic')]         #Fo
          data['make'].value_counts()[0:2:]                                        #se
```

```
Out[135]: Ford         12900
          Chevrolet     4590
          Name: make, dtype: int64
```

In [136]:

```
data=df[(df['body']=='hatchback') & (df['transmission']=='automatic')]
data['make'].value_counts()[0:2:]                                             #th
```

Out[136]:

```
Ford      2710
Nissan    1548
Name: make, dtype: int64
```

In [137]:

```
data=df[(df['body']=='minivan') & (df['transmission']=='automatic')]
data['make'].value_counts()[0:2:]                                              #f
```

Out[137]:

```
Dodge       3771
Chrysler    3153
Name: make, dtype: int64
```

In [138]:

```
data=df[(df['body']=='wagon') & (df['transmission']=='automatic')]          #
data['make'].value_counts()[0:2:]                                           #fit
```

Out[138]:

```
Kia     1560
Ford     994
Name: make, dtype: int64
```

knowing which makers making top 3 model with automatic transmission rating >30

In [139]:

```
#altima    ,fusion ,escape

data=df[(df['model']=='altima') & (df['transmission']=='automatic') & (df['
data['make'].value_counts()[0:2:]
```

Out[139]:

```
Nissan    4541
Name: make, dtype: int64
```

In [140]:

```
data=df[(df['model']=='fusion') & (df['transmission']=='automatic') & (df['
data['make'].value_counts()[0:2:]
```

Out[140]:

```
Ford    5204
Name: make, dtype: int64
```

In [141]:

```
data=df[(df['model']=='escape') & (df['transmission']=='automatic') & (df['
data['make'].value_counts()[0:2:]
```

Out[141]:

```
Ford    4818
Name: make, dtype: int64
```

checking year wise which brand is highest maker and its highest body type

In [143]:

```
data=df[df['year']==2014]                         #in 2014 the highest maker(seller
data['make'].value_counts()[0:3:]
```

Out[143]:

```
Ford         2379
Chevrolet    1296
Nissan       1244
Name: make, dtype: int64
```

```
In [145]: data=df[(df['year']==2014) & (df['make']=='Ford')]
          data['body'].value_counts()[0:3:]

Out[145]: suv          817
          sedan        725
          hatchback    164
          Name: body, dtype: int64
```

```
In [146]: data=df[df['year']==2014]                 #in 2015 the highest maker(seller
          data['make'].value_counts()[0:3:]

Out[146]: Ford         2379
          Chevrolet    1296
          Nissan       1244
          Name: make, dtype: int64
```

```
In [148]: data=df[(df['year']==2014) & (df['make']=='Ford')]
          data['body'].value_counts()[0:3:]

Out[148]: suv          817
          sedan        725
          hatchback    164
          Name: body, dtype: int64
```

top model and its tanssmission automatic and know the maker

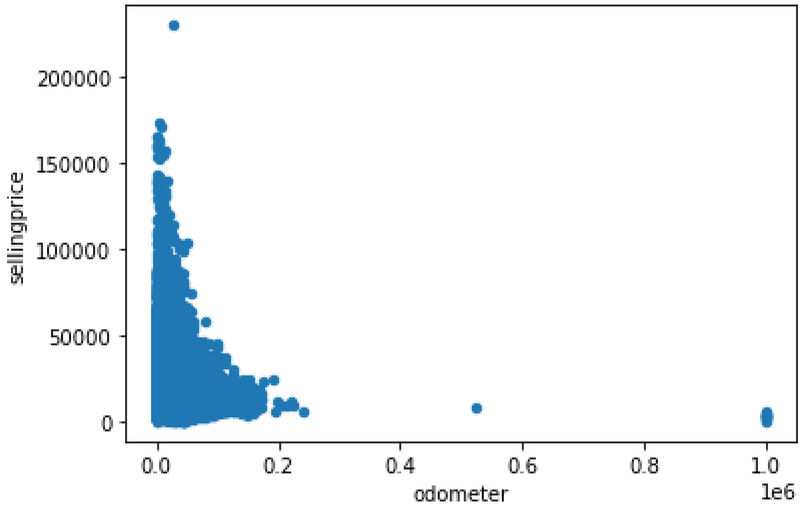
```
In [149]: df['model'].value_counts()[0:2:]     #top model   altima    7388    and the

Out[149]: altima    7388
          fusion    6741
          Name: model, dtype: int64
```

```
In [150]: data=df[(df['model']=='altima') & (df['transmission']=='automatic')]
          data['make'].value_counts()[0:2:]

Out[150]: Nissan    6074
          Name: make, dtype: int64
```

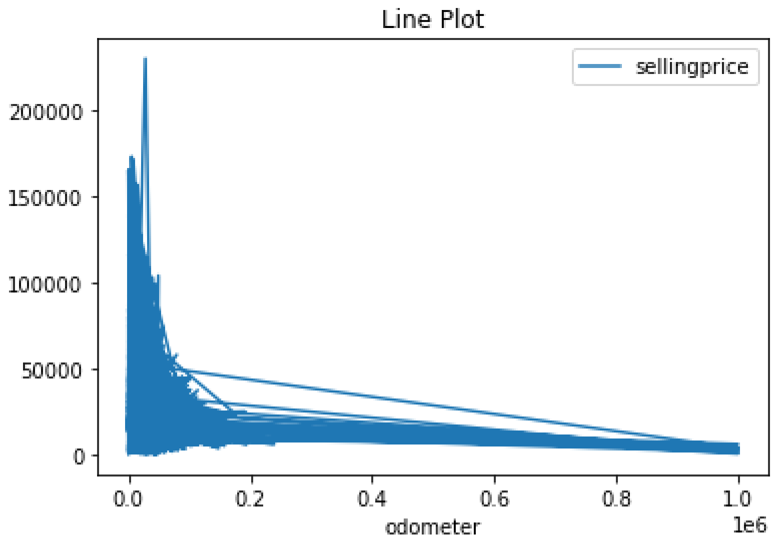
```
In [ ]:
```

```
In [151]: import pandas as pd
```

```
In [152]: import matplotlib.pyplot as plt
In [153]: df.plot(kind='scatter', x='odometer', y='sellingprice', marker='o')
          plt.show()
```



```
In [154]: df.plot(kind='line', x='odometer', y='sellingprice', title='Line Plot')
          plt.show()
```



which exterior color choose by the top body or maker

```
In [155]: data=df[df['make']=='Ford']            #1st  white        9969
                                                  # 2nd  black        7102
          data['color'].value_counts()[0:2:]
```

```
Out[155]: white     9969
          black     7102
          Name: color, dtype: int64
```

```
In [ ]:
```

which interior color choose by the top maker

In [156]:

```
data=df[df['make']=='Ford']
data['interior'].value_counts()[0:3:]          #1st black    22881
                                               #2nd gray     12650
```

Out[156]:

```
black    22881
gray     12650
beige     3232
Name: interior, dtype: int64
```

in top 3 state which seller is highest in there particular state

In [157]:

```
df['state'].value_counts()[0:3:]       #top 3 three state
                                        #fl    -   29348
                                        #ca   -  23609
                                        #pa   -    16410
```

Out[157]:

```
fl    29348
ca    23609
pa    16410
Name: state, dtype: int64
```

In [158]:

```
#1st fl state and thre highest seller  #the seller is the hertz corporation
data=df[df['state']=='fl']
data['seller'].value_counts()[0:2:]
```

Out[158]:

```
the hertz corporation    1613
gm remarketing           1584
Name: seller, dtype: int64
```

In [159]:

```
#2nd ca state and thre highest seller  #the seller is the hertz corporation
data=df[df['state']=='ca']
data['seller'].value_counts()[0:2:]
```

Out[159]:

```
the hertz corporation                                   1759
enterprise vehicle exchange / tra / rental / tulsa     1483
Name: seller, dtype: int64
```

In [160]:

```
##3rd pa state and thre highest seller  #the seller is the hertz corporatio
data=df[df['state']=='pa']
data['seller'].value_counts()[0:2:]
```

Out[160]:

```
the hertz corporation    1522
avis corporation          936
Name: seller, dtype: int64
```

highest selling price and whci maker are greater than average selling price

In [162]:

```
df['sellingprice'].median()
```

Out[162]:

```
16500.0
```

In [163]:
```
data=df[df['sellingprice']>df['sellingprice'].mean()]
data['make'].value_counts()[0:3:]            #Ford - 17791
```

Out[163]:
```
Ford         17791
Chevrolet     5732
Infiniti      5037
Name: make, dtype: int64
```

how many maker build only few model type not other maker build that body type

In [164]:
```
df['model'].value_counts()[-4::]
```

Out[164]:
```
range                 1
q3                    1
grand cherokee srt    1
ghost                 1
Name: model, dtype: int64
```

these are the unique model #1)range land rover #2)q3 Audi #3)grand cherokee srt Jeep #4)ghost Rolls-Royce

In [171]:
```
df[df['model']=='range']
```

Out[171]:

	Unnamed: 0	year	make	model	trim	body	transmission	state	condition	odometer
19119	19119	2014	land rover	range	r spt v6 hse	0	automatic	tx	43.0	18620.0

In [166]:
```
df[df['model']=='q3']
```

Out[166]:

	Unnamed: 0	year	make	model	trim	body	transmission	state	condition	odomet
6315	6315	2015	Audi	q3	Premium Plus	suv	automatic	il	5.0	1404

In [19]:
```
df[df['model']=='ghost']
```

Out[19]:

	Unnamed: 0	year	make	model	trim	body	transmission	state	condition	odomet
186858	186858	2013	Rolls-Royce	ghost	Base	sedan	automatic	fl	42.0	7852

```
In [167]: df[df['model']=='grand cherokee srt']
```

Out[167]:

	Unnamed: 0	year	make	model	trim	body	transmission	state	condition	odomete
5956	5956	2015	Jeep	grand cherokee srt	Base	suv	automatic	ga	47.0	6846.

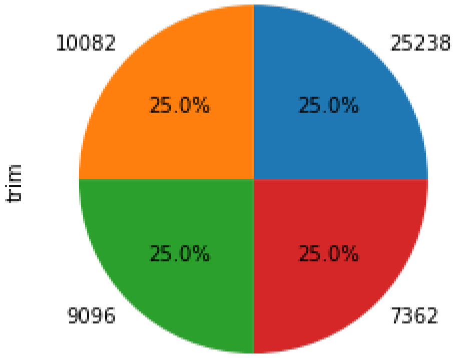
top trim #SE 25238 -----Ford 16712 #Base 10082------Lexus 739 #LX 9096 -----Kia 7025 #Limited 7362 -----Ford 3790

```
In [52]: df['trim'].value_counts()[0:4:]
```

```
Out[52]: 25238    1
10082    1
9096     1
7362     1
Name: trim, dtype: int64
```

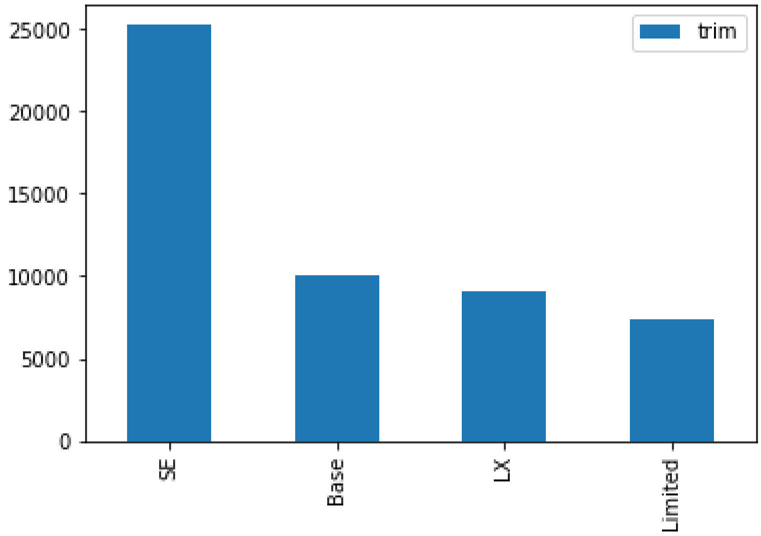
```
In [51]: (df['trim'].value_counts()[0:4:]).plot(kind='pie',autopct="%0.1f%%")
```

```
Out[51]: <AxesSubplot:ylabel='trim'>
```



In [50]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = df['trim'].value_counts()[0:4:]

# Create DataFrame
df = pd.DataFrame(data)



# Plot bar chart
df.plot(kind='bar')

# Show the plot
plt.show()
```



In [185]:

```
data=df[df['trim']=='Limited']
```

In [187]:

```
data['make'].value_counts()[0:4:]
```

Out[187]:

```
Ford        3790
Hyundai     1079
Jeep         859
Chrysler     676
Name: make, dtype: int64
```

most rarer color use by the maker in exterior

In [190]:

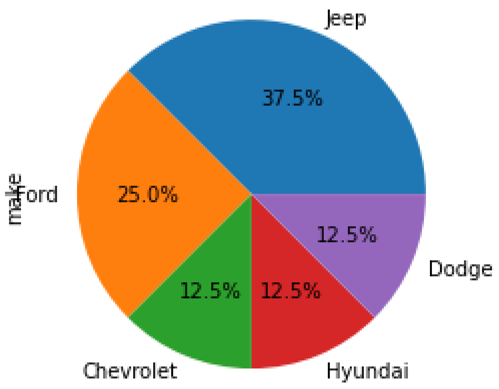
```
df['color'].value_counts()      #pink  29  top maker Chevrolet     13
                                  #lime  8    top maker Jeep            3
```

Out[190]:

```
white        40237
black        35742
gray         29538
silver       26220
red          17312
blue         13803
–            12857
brown         2684
burgundy      2489
green         2096
gold          1750
beige         1691
orange         585
purple         397
off-white      376
0              351
yellow         262
charcoal       187
turquoise       35
pink            29
lime             8
1167             1
9410             1
5001             1
9562             1
2846             1
18561            1
5705             1
2817             1
6158             1
721              1
20627            1
20379            1
9837             1
9887             1
18384            1
339              1
6864             1
2711             1
11034            1
16633            1
6388             1
15719            1
12655            1
14872            1
4802             1
2172             1
Name: color, dtype: int64
```

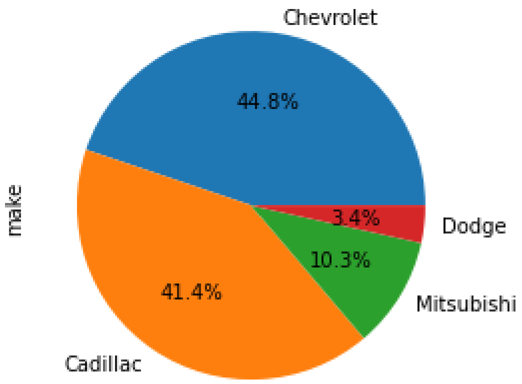
```
In [191]: data=df[df['color']=='lime']#how many use lime color exterior
          data['make'].value_counts()
          (data['make'].value_counts()[0:6:]).plot(kind='pie',autopct="%0.1f%%")
```

```
Out[191]: <AxesSubplot:ylabel='make'>
```



```
In [194]: data=df[df['color']=='pink'] #how many use ink color exterior
          data['make'].value_counts()
          (data['make'].value_counts()[0:6:]).plot(kind='pie',autopct="%0.1f%%")
```

```
Out[194]: <AxesSubplot:ylabel='make'>
```



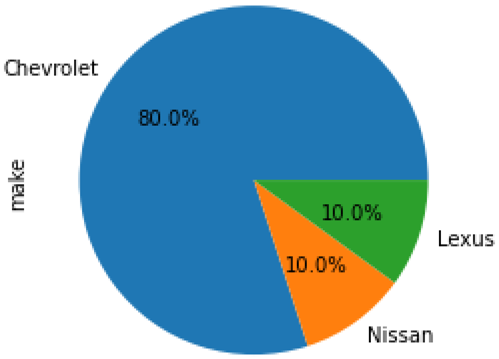
most rarer color use by the maker in interiors

```
In [195]: data=df[df['interior']=='yellow']  #orange-28  top maker is  Nissan -5  BMW
          data['make'].value_counts()        #yellow -10 top maker Chevrolet -  8
```

```
Out[195]: Chevrolet    8
          Nissan       1
          Lexus        1
          Name: make, dtype: int64
```

```
In [72]: (data['make'].value_counts()[0:6:]).plot(kind='pie',autopct="%0.1f%%")#to v
```

```
Out[72]: <AxesSubplot:ylabel='make'>
```

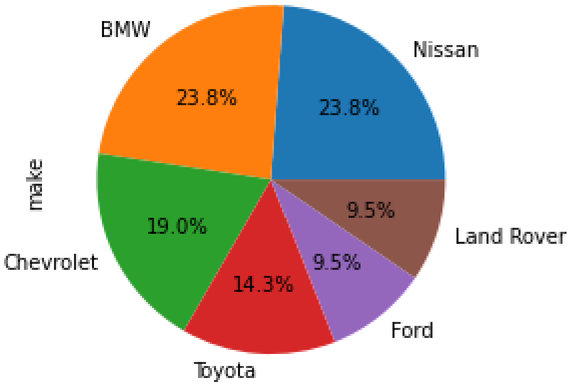


```
In [196]: data=df[df['interior']=='orange']  #orange-28  top maker is  Nissan -5  BMW
          data['make'].value_counts()
```

```
Out[196]: Nissan        5
          BMW           5
          Chevrolet     4
          Toyota        3
          Ford          2
          Land Rover    2
          Lexus         2
          Kia           1
          Honda         1
          Lincoln       1
          FIAT          1
          Mazda         1
          Name: make, dtype: int64
```

```
In [197]: (data['make'].value_counts()[0:6:]).plot(kind='pie',autopct="%0.1f%%")#to v
```

```
Out[197]: <AxesSubplot:ylabel='make'>
```



```
In [107]: #df.drop(columns=['Unnamed: 0'],inplace=True)#its index column in pandas we
```

In [87]:

```
#df.drop(columns=['saledate'],inplace=True)#because we take data required d
```

In [108]:

```
df.head()
```

Out[108]:

	year	make	model	trim	body	transmission	state	condition	odometer	color	interic
0	2014	Kia	sorento	LX	suv	automatic	ca	5.0	16639.0	white	blac
1	2014	Kia	sorento	LX	suv	automatic	ca	5.0	9393.0	white	beig
2	2015	Volvo	s60	T5	sedan	automatic	ca	41.0	14282.0	white	blac
3	2014	Nissan	altima	2.5 S	sedan	automatic	ca	1.0	5554.0	gray	blac
4	2014	Kia	optima	LX	sedan	automatic	ca	48.0	2034.0	red	ta

In [110]:

```
#df.to_csv('C:/Users/ROCKSTAR/Desktop/py notes/final_1.csv')
```

In [115]:

```
df['month'].unique()
```

Out[115]:

```
array(['Dec', 'Jan', 'Jul', 'Feb', 'Jun', 'Mar', 'May', 'Apr', '0'],
      dtype=object)
```

In [116]:

```
df['year'].unique()
```

Out[116]:

```
array([2014, 2015,    0], dtype=int64)
```

in particular year which month has highest sell and its seller name and most selling body type in that month

In [222]:

```
data=df[df['year']==2014]
```

In [223]:

```
data['month'].value_counts()#in 2014 december has highest sell
```

Out[223]:

```
Dec    11962
Jan      128
Feb        1
Name: month, dtype: int64
```

In [243]:

```
data=df[df['month']=='Dec']
data['seller'].value_counts()[0:5:]#in 2014 december has highest sell and s
                                    #1st ford motor credit company llc
                                    # body type suv    258
```

Out[243]:

```
ford motor credit company llc                         550
enterprise veh exchange/rental                        540
avis corporation                                      487
toyota motor sales usa inc/program                    381
enterprise vehicle exchange / tra / rental / tulsa    377
Name: seller, dtype: int64
```

In [244]:

```
data1=df[(df['year']==2014) & (df['month']=='Dec') & (df['seller']=='ford m
```

In [245]:

```
data1['body'].value_counts()[0:5:]
```

Out[245]:

```
suv          258
sedan        206
hatchback     33
supercrew     26
wagon         11
Name: body, dtype: int64
```

```
In [19]: data=df[df['year']==2015]
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3621, in In
dex.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._
libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._
libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'year'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
Input In [19], in <cell line: 1>()
----> 1 data=df[df['year']==2015]

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3505, in DataFram
e.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3623, in In
dex.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'year'
```

In [213]:

```
data['month'].value_counts()#in 2015 feb has highest sell
```

Out[213]:

```
Feb    54240
Jan    42812
Jun    41851
May    20612
Mar    16062
Jul      532
Apr      439
Name: month, dtype: int64
```

In [246]:

```
data=df[df['month']=='Feb']
data['seller'].value_counts()[0:5:]#in 2015 Feb has highest sell and seller
                                    #1st avis corporation          4847
                                    #sedan
```

Out[246]:

```
avis corporation                    4847
the hertz corporation               3578
ford motor credit company llc       2861
enterprise veh exchange/rental      1923
kia motors america  inc             1646
Name: seller, dtype: int64
```

```
In [18]: data2=df[(df['year']==2015) & (df['month']=='Feb') & (df['seller']=='avis c
```

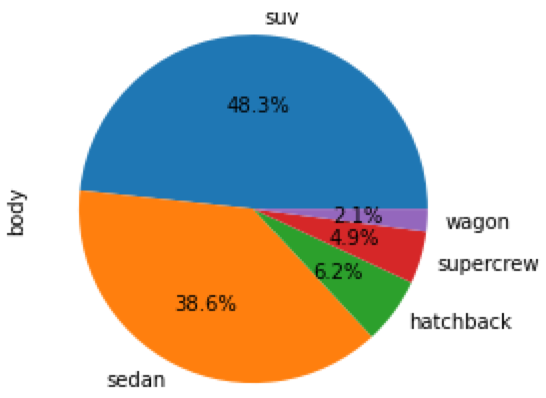
```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3621, in In
dex.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._
libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._
libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'year'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
Input In [18], in <cell line: 1>()
----> 1 data2=df[(df['year']==2015) & (df['month']=='Feb') & (df['seller']
=='avis corporation')]

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3505, in DataFram
e.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3623, in In
dex.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'year'
```

```
In [252]: data2['body'].value_counts()[:5:]#sedan
```

```
Out[252]: sedan        2636
          suv          1202
          minivan       455
          hatchback     297
          coupe         100
          Name: body, dtype: int64
```

```
In [256]: (data1['body'].value_counts()[:5:]).plot(kind='pie',autopct="%0.1f%%") #yea
```

```
Out[256]: <AxesSubplot:ylabel='body'>
```



```
In [17]: (data2['body'].value_counts()[:5:]).plot(kind='pie',autopct="%0.1f%%") #yea
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Input In [17], in <cell line: 1>()
----> 1 (data2['body'].value_counts()[:5:]).plot(kind='pie',autopct="%0.1
f%%")

NameError: name 'data2' is not defined
```

top seller

In [ ]:

```
# Sample data
data = df['body'].value_counts()[0:5:]

# Create DataFrame
df = pd.DataFrame(data)



# Plot bar chart
df.plot(kind='bar')

# Show the plot
plt.show()
```

In [8]:

```
df['seller'].value_counts()[0:5:]
```

Out[8]:

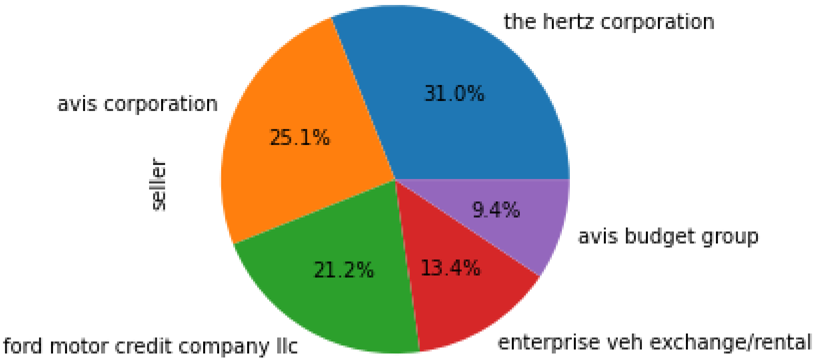
```
the hertz corporation              15306
avis corporation                   12381
ford motor credit company llc      10469
enterprise veh exchange/rental      6598
avis budget group                   4637
Name: seller, dtype: int64
```

In [10]:

```
(df['seller'].value_counts()[0:5:]).plot(kind='pie',autopct="%0.1f%%")
```

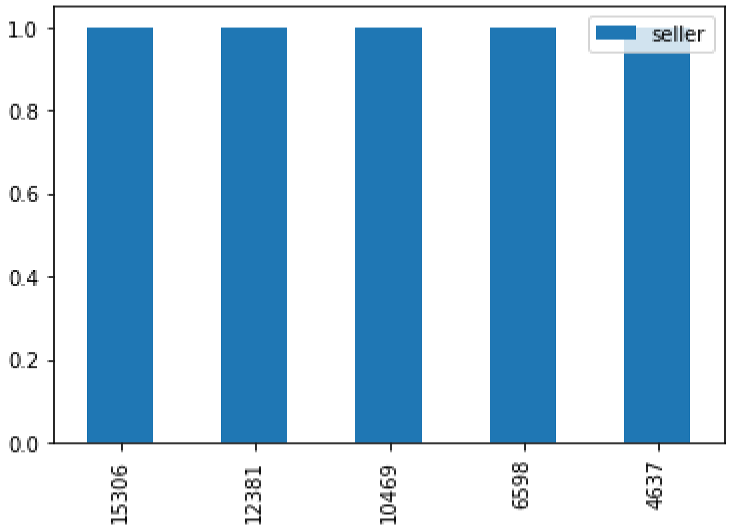
Out[10]:

```
<AxesSubplot:ylabel='seller'>
```



In [16]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = df['seller'].value_counts()[0:5:]

# Create DataFrame
df = pd.DataFrame(data)



# Plot bar chart
df.plot(kind='bar')

# Show the plot
plt.show()
```



In [14]:

```
df.head(10)
```

Out[14]:

	model
altima	7388
fusion	6741
escape	6172
focus	5923
cruze	4706

```
In [13]: df['trim']
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3621, in In
dex.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._
libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._
libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'trim'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
Input In [13], in <cell line: 1>()
----> 1 df['trim']

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3505, in DataFram
e.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3623, in In
dex.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'trim'
```

```
In [ ]:
```