



CS2101 Computer Programing I

Department of Information Technology

Vincent School Mary of Science and Technology

Pandas Report Project

By

Teerapat Poonpattanasuk 6035120

Submitted to Asst. Prof. THANACHAI THUMTHAWATWORN (1/2018)

Overview

This Extended Solution to the Inventory Management module adds the Inventory Usage Report and the Product Sales Analysis Report. The Inventory Usage Report prints stock and usage information for standard items and component items from unexploded bills in a one line per item format. The Product Sales Analysis Report prints one line per item for the total sales of the item to all customers. This basic sales report/forecast tool is for a small or new business which does not yet have a computerized full management information system, which would normally integrate sales reporting with other business processes.

```
In [4]: import pandas as pd
import numpy as np
import pylab as pl
import matplotlib as plt
%matplotlib inline

df = pd.read_csv("/SalesJan2009.csv")
df.head(16)
```

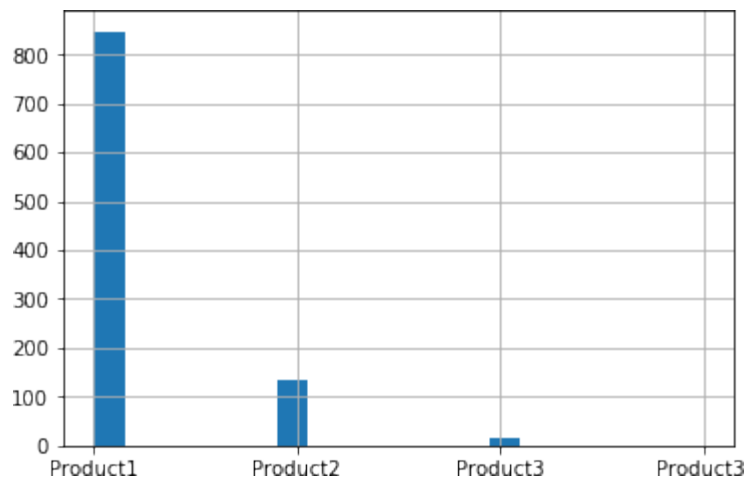
Out[4]:

	Transaction_date	Product	Price	Payment_Type	Name	City	State	C
0	1/2/09 6:17	Product1	1200	Mastercard	carolina	Basildon	England	Unite Kingd
1	1/2/09 4:53	Product1	1200	Visa	Betina	Parkville	MO	Unite State
2	1/2/09 13:08	Product1	1200	Mastercard	Federica e Andrea	Astoria	OR	Unite State
3	1/3/09 14:44	Product1	1200	Visa	Gouya	Echuca	Victoria	Austr
4	1/4/09 12:56	Product2	3600	Visa	Gerd W	Cahaba Heights	AL	Unite State
5	1/4/09 13:19	Product1	1200	Visa	LAURENCE	Mickleton	NJ	Unite State
6	1/4/09 20:11	Product1	1200	Mastercard	Fleur	Peoria	IL	Unite State
7	1/2/09 20:09	Product1	1200	Mastercard	adam	Martin	TN	Unite State
8	1/4/09 13:17	Product1	1200	Mastercard	Renee Elisabeth	Tel Aviv	Tel Aviv	Israe
9	1/4/09 14:11	Product1	1200	Visa	Aidan	Chatou	Ile-de-France	Fran
10	1/5/09 2:42	Product1	1200	Diners	Stacy	New York	NY	Unite State
11	1/5/09 5:39	Product1	1200	Amex	Heidi	Eindhoven	Noord-Brabant	Neth
12	1/2/09 9:16	Product1	1200	Mastercard	Sean	Shavano Park	TX	Unite State

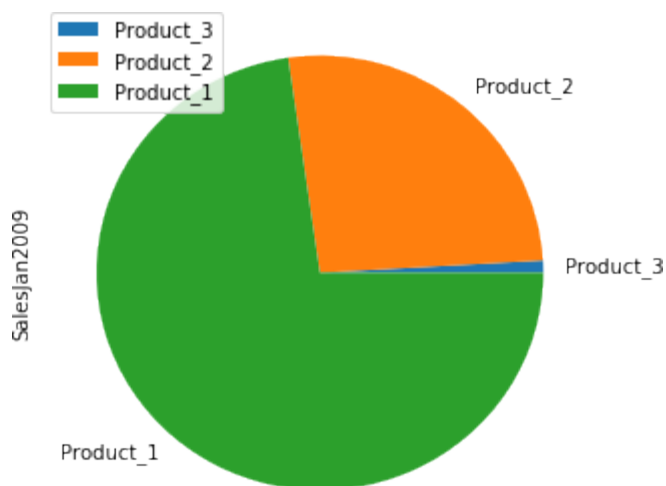
15	1/4/09 1:05	Product1	1200	Diners	Leanne	Julianstown	Meath	Ireland
----	-------------	----------	------	--------	--------	-------------	-------	---------

```
In [5]: df['Product'].hist(bins=20)
```

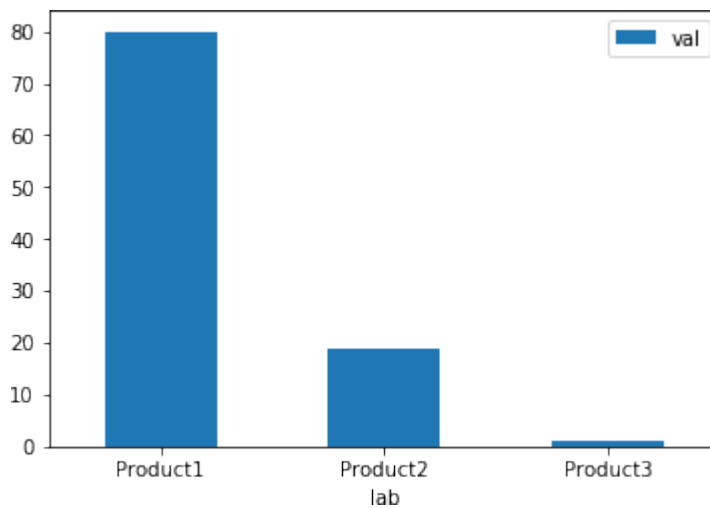
```
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x1549d39dd68>
```



```
In [51]: df = pd.DataFrame({'SalesJan2009': [1, 29, 80],
                             'radius': [2439.7, 6051.8, 6378.1]},
                             index=['Product_3', 'Product_2', 'Product_1'])
plot = df.plot.pie(y='SalesJan2009', figsize=(5, 5))
```



```
In [38]: df = pd.DataFrame({'lab': ['Product1', 'Product2', 'Product3'], 'val': [80,
                                          19, 1]})
ax = df.plot.bar(x='lab', y='val', rot=0)
```



```
In [31]: df = pd.read_csv("/SalesJan2009.csv")
df['Product'].value_counts()
```

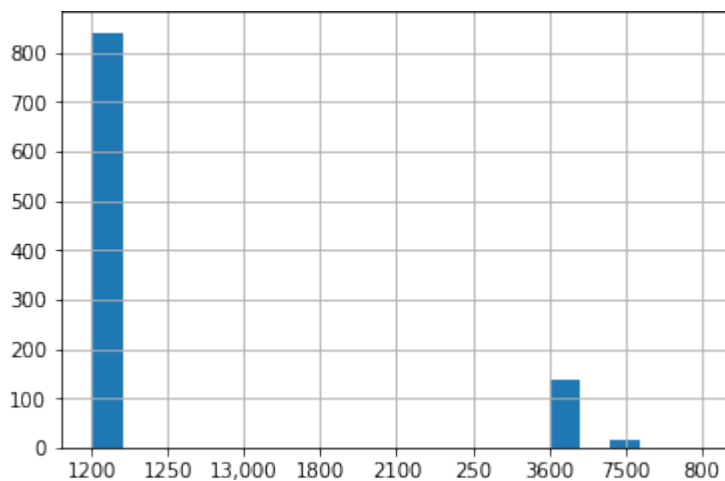
```
Out[31]: Product1      847
Product2      136
Product3       14
Product3         1
Name: Product, dtype: int64
```

```
In [32]: df['Price'].value_counts()
```

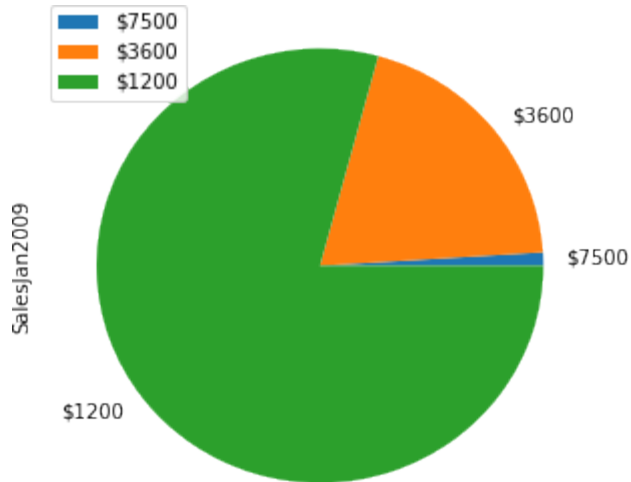
```
Out[32]: 1200      841
3600      136
7500       15
250         1
2100        1
800         1
13,000       1
1250         1
1800         1
Name: Price, dtype: int64
```

```
In [33]: df['Price'].hist(bins=20)
```

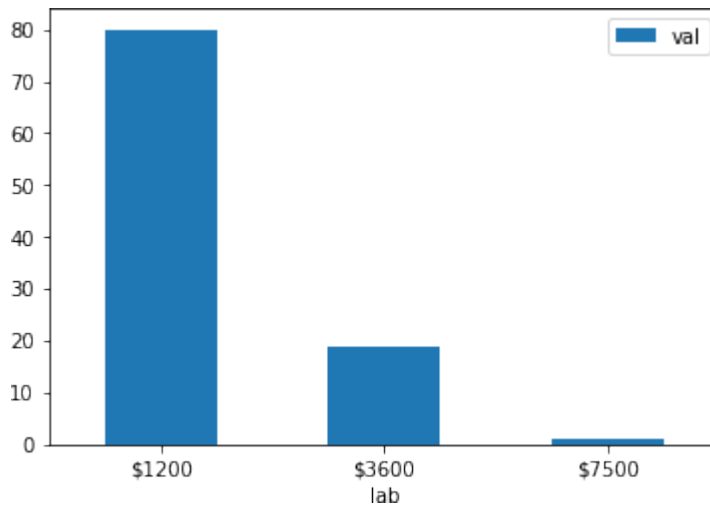
```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x1549f813320>
```



```
In [35]: df = pd.DataFrame({'SalesJan2009': [1, 20, 80],
                             'radius': [2439.7, 6051.8, 6378.1]},
                             index=['$7500', '$3600', '$1200'])
plot = df.plot.pie(y='SalesJan2009', figsize=(5, 5))
```



```
In [36]: df = pd.DataFrame({'lab': ['$1200', '$3600', '$7500'], 'val': [80, 19, 1]})
ax = df.plot.bar(x='lab', y='val', rot=0)
```

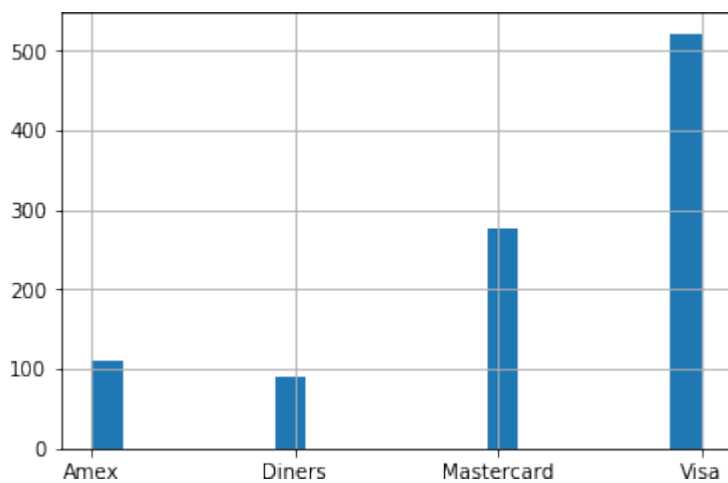


```
In [41]: df = pd.read_csv("/SalesJan2009.csv")
df['Payment_Type'].value_counts()
```

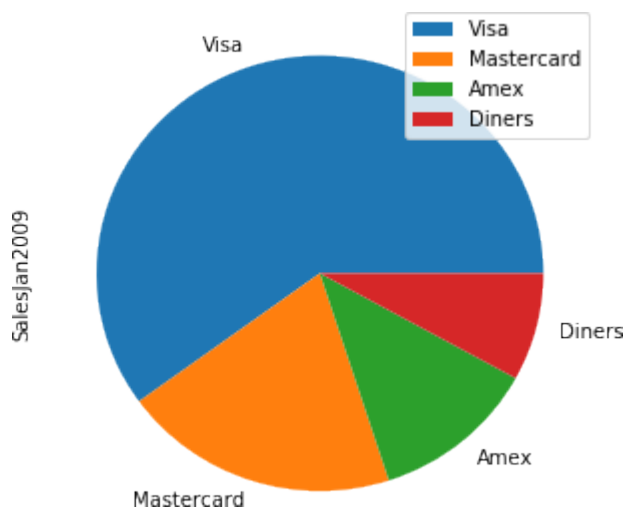
```
Out[41]: Visa          522
Mastercard    277
Amex          110
Diners        89
Name: Payment_Type, dtype: int64
```

```
In [42]: df['Payment_Type'].hist(bins=20)
```

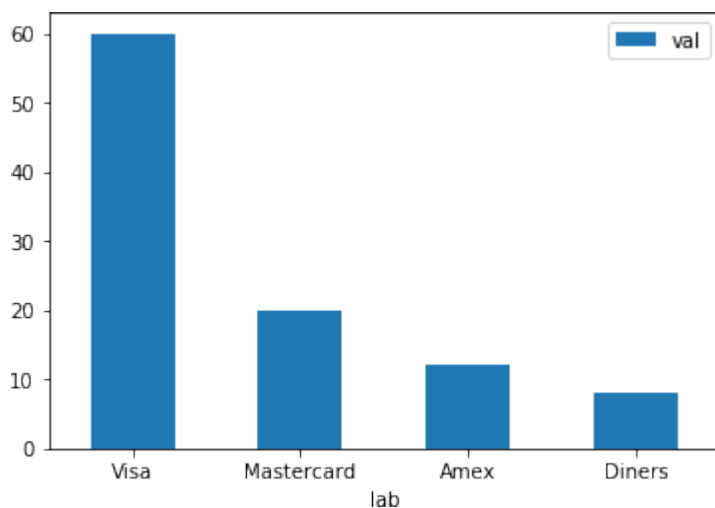
```
Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x1549f53e470>
```



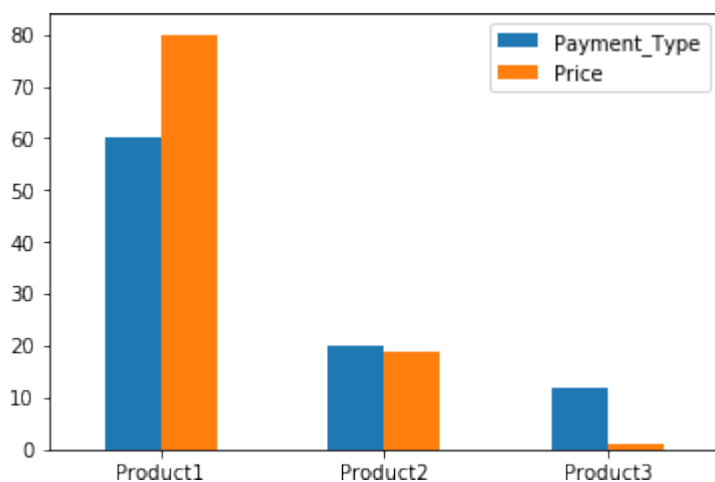
```
In [43]: df = pd.DataFrame({'SalesJan2009': [60, 20, 12, 8 ],
                             'radius': [2439.7, 6051.8, 6378.1, 2439.7]},
                             index=['Visa', 'Mastercard', 'Amex', 'Diners'])
plot = df.plot.pie(y='SalesJan2009', figsize=(5, 5))
```



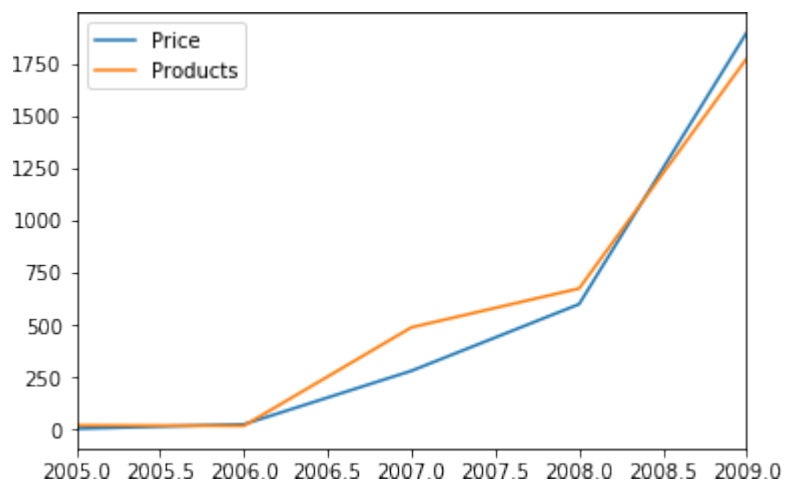
```
In [44]: df = pd.DataFrame({'lab': ['Visa', 'Mastercard', 'Amex', 'Diners'], 'val': [60, 20, 12, 8]})
ax = df.plot.bar(x='lab', y='val', rot=0)
```



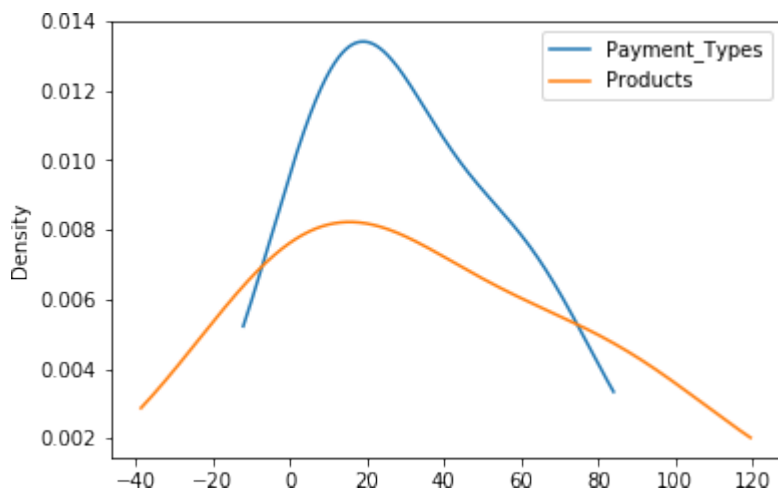
```
In [50]: Price = [80,19,1]
Payment_Type = [60,20,12]
index = ['Product1', 'Product2', 'Product3']
df = pd.DataFrame({'Price': Price,
                  'Payment_Type': Payment_Type}, index=index)
ax = df.plot.bar(rot=0)
```



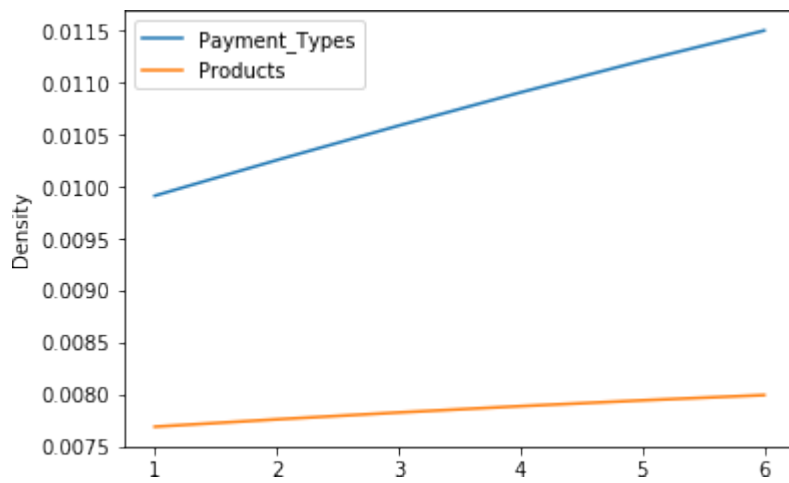
```
In [68]: df = pd.DataFrame({
    'Products': [20, 18, 489, 675, 1776],
    'Price': [4, 25, 281, 600, 1900]
}, index=[2005, 2006, 2007, 2008, 2009])
lines = df.plot.line()
```



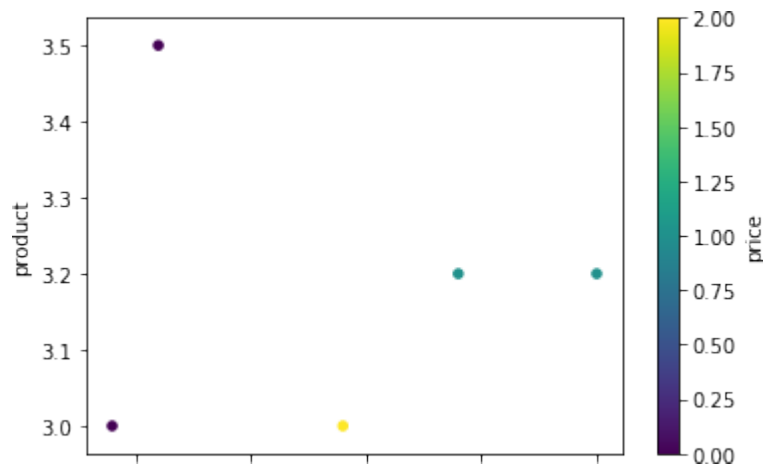
```
In [74]: df = pd.DataFrame({
    'Products': [80, 19, 1],
    'Payment_Types': [60, 20, 12,],
})
ax = df.plot.kde()
```



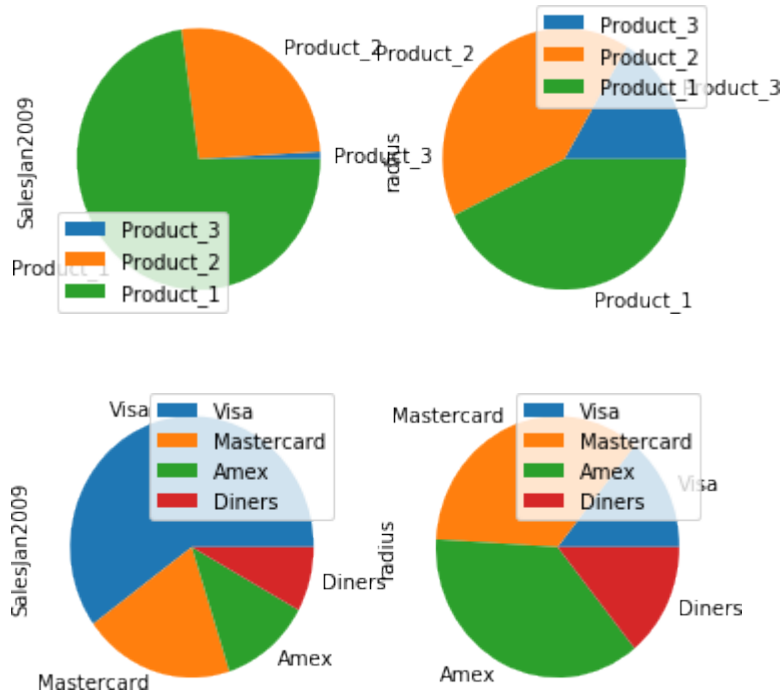
```
In [75]: df = pd.DataFrame({
    'Products': [80, 19, 1],
    'Payment_Types': [60, 20, 12,],
})
ax = df.plot.kde(ind=[1, 2, 3, 4, 5, 6])
```

```
In [82]: >>> df = pd.DataFrame([[5.1, 3.5, 0], [4.9, 3.0, 0], [7.0, 3.2, 1],
...                          [6.4, 3.2, 1], [5.9, 3.0, 2]],
...                          columns=['PaymentType', 'product', 'price'])
>>> ax2 = df.plot.scatter(x='PaymentType',
...                       y='product',
...                       c='price',
...                       colormap='viridis')
```



```
In [87]: df = pd.DataFrame({'SalesJan2009': [1, 29, 80],
...                          'radius': [2439.7, 6051.8, 6378.1]},
...                          index=['Product_3', 'Product_2', 'Product_1'])
plot = df.plot.pie(subplots=True, figsize=(6, 3))
df = pd.DataFrame({'SalesJan2009': [60, 20, 12, 8],
...               'radius': [2439.7, 6051.8, 6378.1, 2439.7]},
...               index=['Visa', 'Mastercard', 'Amex', 'Diners'])
plot = df.plot.pie(subplots=True, figsize=(6, 3))
```



```
In [94]: df = pd.read_csv("/SalesJan2009.csv")
df.apply(lambda x: sum(x.isnull()),axis=0)
```

```
Out[94]: Transaction_date    0
Product                    0
Price                      0
Payment_Type               0
Name                       0
City                       0
State                      1
Country                    0
Account_Created            0
Last_Login                 0
Latitude                   0
Longitude                   0
dtype: int64
```

```
In [95]: df['State'].value_counts()
```

```
Out[95]: England            86
CA                          66
NY                           41
TX                           37
VA                           30
FL                           29
Ontario                      27
Dublin                       22
GA                           21
NJ                           19
British Columbia             18
Alberta                      18
IL                           16
MD                           15
```

WA	14
MA	13
CO	11
MI	11
TN	11
PA	11
New South Wales	11
Cork	10
AZ	10
Zuid-Holland	10
MN	10
Queensland	10
CT	9
Stockholm	9
HI	9
Victoria	9
	..
Kiev	1
Yukon Territory	1
Budapest	1
Minas Gerais	1
Attiki	1
Kuala Lumpur	1
Pest	1
Westmeath	1
NM	1
Kerry	1
Manitoba	1
Andalucia	1
Mazowieckie	1
Hainaut	1
NH	1
Sao Paulo	1
Abu Zaby	1
Skane	1
Pays de la Loire	1
Black River	1
Phuket	1
Tasmania	1
Buenos Aires	1
Tyrol	1
Antwerpen	1
Guangdong	1
Heredia	1
Ceara	1
Noord-Brabant	1
Saarland	1

Name: State, Length: 204, dtype: int64

```
In [97]: df['State'].fillna('No', inplace=True)
```

```
In [98]: df.apply(lambda x: sum(x.isnull()), axis=0)
```

```
Out[98]: Transaction_date    0
Product                    0
Price                      0
```

```
Payment_Type      0
Name              0
City             0
State            0
Country          0
Account_Created  0
Last_Login       0
Latitude         0
Longitude        0
dtype: int64
```

```
In [99]: df.dtypes
```

```
Out[99]: Transaction_date      object
Product                       object
Price                         object
Payment_Type                  object
Name                          object
City                          object
State                         object
Country                       object
Account_Created               object
Last_Login                    object
Latitude                      float64
Longitude                     float64
dtype: object
```

```
In [105]: df['State'].value_counts()
```

```
Out[105]: England      86
CA                    66
NY                    41
TX                    37
VA                    30
FL                    29
Ontario              27
Dublin               22
GA                   21
NJ                   19
British Columbia    18
Alberta              18
IL                   16
MD                   15
WA                   14
MA                   13
CO                   11
MI                   11
TN                   11
PA                   11
New South Wales     11
Cork                 10
AZ                   10
Zuid-Holland        10
MN                   10
Queensland          10
CT                   9
Stockholm            9
```

```

HI 9
Victoria 9
..
Yukon Territory 1
Minas Gerais 1
Staden Kobenhavn 1
Alsace 1
Saarland 1
MO 1
Pays de la Loire 1
Kerry 1
Manitoba 1
Andalucia 1
Mazowieckie 1
No 1
Hainaut 1
NH 1
Sao Paulo 1
Abu Zaby 1
Skane 1
Black River 1
Pest 1
Phuket 1
Tasmania 1
Buenos Aires 1
Tyrol 1
Antwerpen 1
Guangdong 1
Heredia 1
Ceara 1
Noord-Brabant 1
Westmeath 1
NM 1
Name: State, Length: 205, dtype: int64

```

```
In [106]: df['State'].fillna('England',inplace = True)
```

```
In [107]: df.apply(lambda x: sum(x.isnull()),axis=0)
```

```
Out[107]: Transaction_date    0
Product                    0
Price                      0
Payment_Type               0
Name                       0
City                       0
State                      0
Country                    0
Account_Created            0
Last_Login                 0
Latitude                   0
Longitude                   0
dtype: int64
```

```
In [123]: from sklearn.preprocessing import LabelEncoder
var_mod = ['Price', 'Product', 'Payment_Type', 'Name', 'City', 'State', 'Country',
', 'Account_Created', 'Last_Login', 'Latitude', 'Longitude']
```

```
le = LabelEncoder()  
for i in var_mod:  
    df[i] = le.fit_transform(df[i])  
df.dtypes
```

```
Out[123]: Transaction_date    object  
Product                    int64  
Price                      int64  
Payment_Type              int64  
Name                      int64  
City                      int64  
State                    int64  
Country                  int64  
Account_Created          int64  
Last_Login               int64  
Latitude                 int64  
Longitude                int64  
dtype: object
```

Conclusion

Outdoor Equipment Ltd is not in a very secure financial position. Improvements in every area of the company are needed if the company is, in the first instance, to survive and then grow. The key areas of reform are the liquidity of the company and the quantity and quality of working capital, profitability, and financial stability. Management must address these areas simultaneously if the company is to overcome its present poor record. This case highlights the importance of understanding consumer behavior and how it is influenced by many factors. There are many factors that affect consumer and business buyer behavior, so it is important use common buyer patterns in your marketing campaign. This company has done this in its marketing for key products and services.