# Intro to Data Science_ Plotting Data 01

March 25, 2020

[82]:
```
# This Python 3 environment comes with many helpful analytics libraries␣
 ↪installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/
 ↪docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list␣
 ↪all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.
```

```
/kaggle/input/sample-sales-data/sales_data_sample.csv
```

This is the post about **Introduction to Data Scinece**, I am going to write about* handling tabular/dataframe* data in python3.

## 0.1 ### Plotting

A Plot is a way of representing data visually. It can describe relationships between variables. Plots can give insight and information about the data that may not have been easily derived by looking at the data as a table. There are two types of plotting avvailable

- Univariate : single variable plotting
- Bivariate : plot bwteen two variable, it shows relationship between two variables.

In this section we will use this sales data https://www.kaggle.com/kyanyoga/sample-sales-data

### 0.1.1 Import Data

[83]:
```
df = pd.read_csv('/kaggle/input/sample-sales-data/sales_data_sample.csv',␣
↪encoding='unicode_escape')
display(df.head())
```

```
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70
4        10159               49     100.00               14  5205.27

          ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0   2/24/2003 0:00  Shipped       1         2     2003  ...
1    5/7/2003 0:00  Shipped       2         5     2003  ...
2    7/1/2003 0:00  Shipped       3         7     2003  ...
3   8/25/2003 0:00  Shipped       3         8     2003  ...
4  10/10/2003 0:00  Shipped       4        10     2003  ...

                    ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0         897 Long Airport Avenue           NaN            NYC    NY
1              59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3              78934 Hillside Dr.           NaN       Pasadena    CA
4                 7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0      10022     USA       NaN              Yu             Kwai    Small
1      51100  France      EMEA         Henriot             Paul    Small
2      75508  France      EMEA        Da Cunha            Daniel   Medium
3      90003     USA       NaN           Young             Julie   Medium
4        NaN     USA       NaN           Brown             Julie   Medium

[5 rows x 25 columns]
```

## 0.2 ### Plotting individual columns

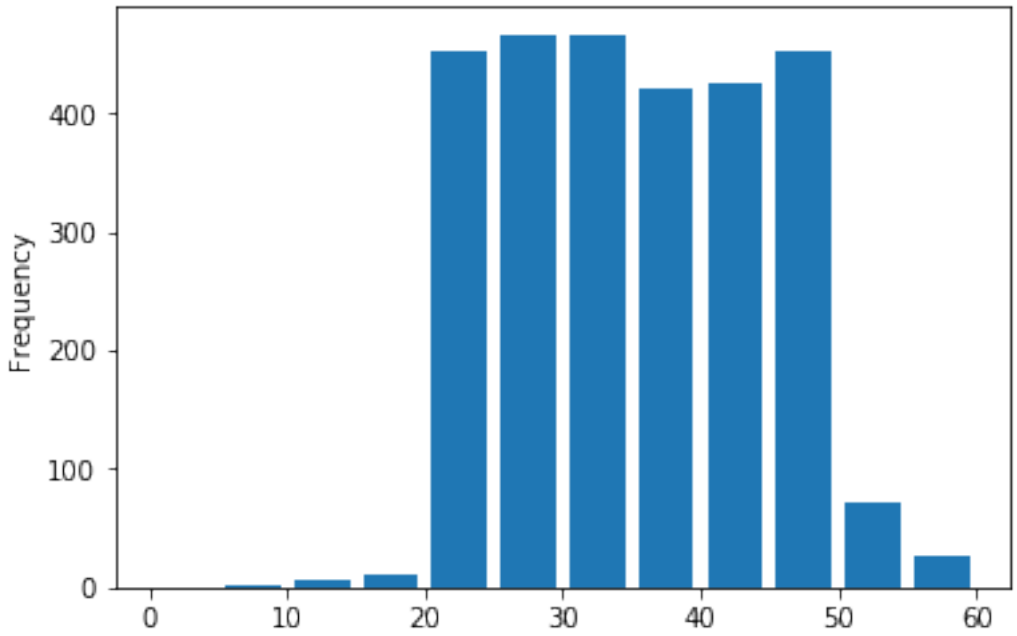
we will use **histogram**. Histograms are plots that inform us of the *underlying frequency distribution* of the data. They group numbers into ranges and tells us how many values of a variable lie in every range.

[84]:
```
import matplotlib.pyplot as plt

print('Minimum quantity ordered : ', df['QUANTITYORDERED'].min())
print('Maximum quantity ordered : ', df['QUANTITYORDERED'].max())
#print(df['QUANTITYORDERED'].value_counts())
```

```
#plot
df['QUANTITYORDERED'].plot(kind='hist',␣
 ↪bins=[0,5,10,15,20,25,30,35,40,45,50,55,60], rwidth = 0.8)
plt.show()
```

```
Minimum quantity ordered :  6
Maximum quantity ordered :  97
```

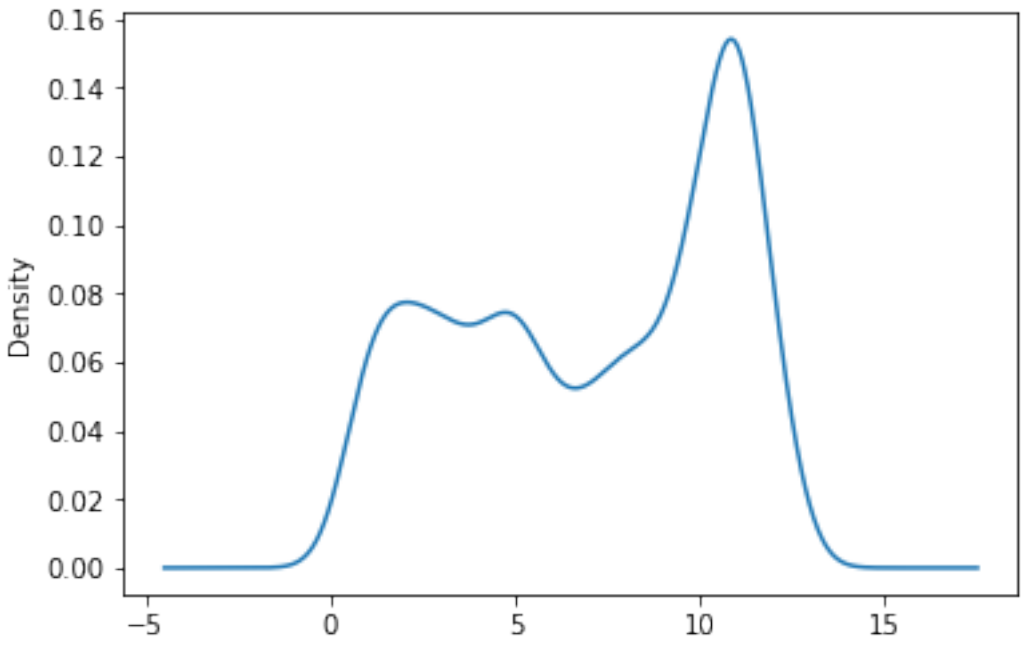



As we can see here the firm received more than 400 orders in range 20 to 50.

### 0.3 ### Density plots

A density plot is a representation of the distribution of a numeric variable. It uses a density estimate to show the probability density function of the variable. It is a smoothed version of the histogram and is used in the same concept.

[85]:
```
df['MONTH_ID'].plot(kind='density')
plt.show()
```

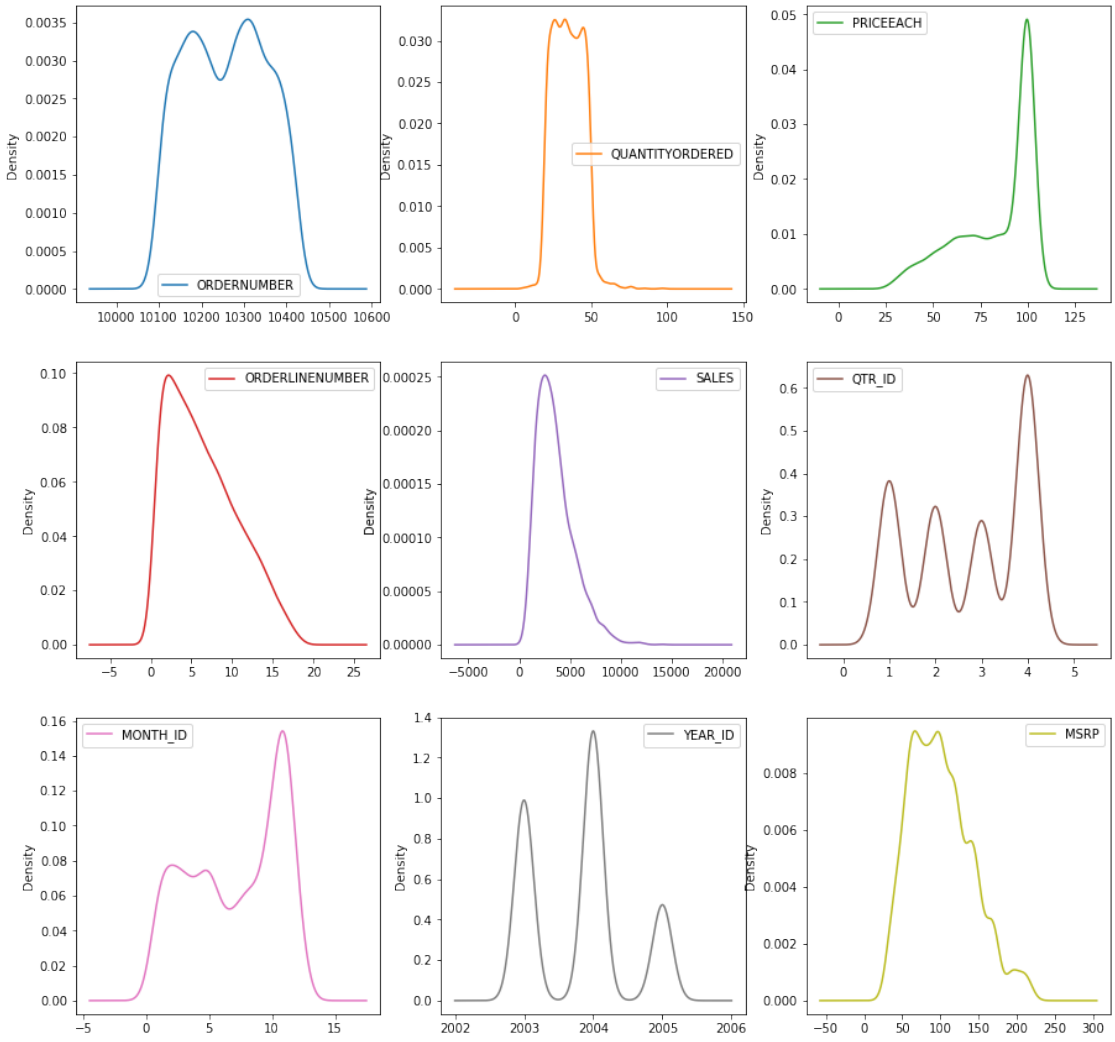



It tells us how much data we have in each month. We see a peak at months 10 and 11, which tells us that most orders were made in these months.

## 0.4 ### Plotting multiple columns

we can plot distribution for all columns that has **single numeric value**. By using onlt sinlge line code.

[86]:
```
df.plot(kind='density', subplots=True, sharex=False, sharey=False,␣
↪layout=(3,3),figsize=(15,15))
plt.show()
```




we can see here that by using single line code we can plot all the column that has numeric value. Let's explain few plot,

1. **QUANTITYORDERED** plot : we can see value between 30 to 50 has most order frequency.
2. **MONTH_ID** : we can see that month between 10 and 11 has most orders than other months