

Pandas6

March 3, 2023

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

```
[4]: data = {"a" : [1,2,3,4] ,  
            "b": [5,6,7,8],  
            "c": ["heramb", "varsha", "chetan", "patil"]}
```

```
[5]: df = pd.DataFrame(data)
```

```
[6]: df
```

```
[6]:   a  b      c  
0  1  5  heramb  
1  2  6  varsha  
2  3  7   chetan  
3  4  8   patil
```

```
[3]: df.set_index('a' , inplace = True)
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[3], line 1  
----> 1 df.set_index('a' , inplace = True)  
  
NameError: name 'df' is not defined
```

```
[11]: df
```

```
[11]:   b      c  
a  
1  5  heramb  
2  6  varsha  
3  7   chetan  
4  8   patil
```

```
[13]: df.reset_index(inplace = True)
```

```
[14]: df
```

```
[14]:      a  b      c
      0  1  5  heramb
      1  2  6  varsha
      2  3  7  chetan
      3  4  8   patil
```

```
[16]: data = {"a" : [1,2,3,4] ,
             "b": [5,6,7,8],
             "c": ["heramb","varsha","chetan","patil"]}
df1 = pd.DataFrame(data , index = ['a','b','c','d'])
```

```
[17]: df1
```

```
[17]:      a  b      c
      a  1  5  heramb
      b  2  6  varsha
      c  3  7  chetan
      d  4  8   patil
```

```
[18]: df1.reindex(['b','c','d','a'])
```

```
[18]:      a  b      c
      b  2  6  varsha
      c  3  7  chetan
      d  4  8   patil
      a  1  5  heramb
```

```
[19]: df1
```

```
[19]:      a  b      c
      a  1  5  heramb
      b  2  6  varsha
      c  3  7  chetan
      d  4  8   patil
```

```
[22]: for i, j in df1.iterrows():
      print(i, j)
```

```
a a      1
b      5
c  heramb
Name: a, dtype: object
b a      2
b      6
c  varsha
Name: b, dtype: object
c a      3
b      7
```

```
c    chetan
Name: c, dtype: object
d a      4
b      8
c    patil
Name: d, dtype: object
```

```
[21]: df1
```

```
[21]:   a  b    c
a  1  5 heramb
b  2  6 varsha
c  3  7 chetan
d  4  8 patil
```

```
[23]: print(j)
```

```
a      4
b      8
c    patil
Name: d, dtype: object
```

```
[24]: for i in df1.iteritems():
      print(i)
```

```
('a', a      1
b      2
c      3
d      4
Name: a, dtype: int64)
('b', a      5
b      6
c      7
d      8
Name: b, dtype: int64)
('c', a    heramb
b    varsha
c    chetan
d    patil
Name: c, dtype: object)
```

```
/tmp/ipykernel_95/549805685.py:1: FutureWarning: iteritems is deprecated and
will be removed in a future version. Use .items instead.
  for i in df1.iteritems():
```

```
[25]: list(df['a'])
```

```
[25]: [1, 2, 3, 4]
```

```
[26]: [i for i in df['a']]
```

```
[26]: [1, 2, 3, 4]
```

```
[28]: def test(x):  
      return x.sum()  
df1.apply(test)
```

```
[28]: a          10  
      b          26  
      c  herambvarshachetanpatil  
      dtype: object
```

```
[29]: df2 = df1[['a','b']]
```

```
[30]: df2
```

```
[30]:   a  b  
a   1  5  
b   2  6  
c   3  7  
d   4  8
```

```
[31]: df2.applymap(lambda x : x**3)
```

```
[31]:   a    b  
a   1  125  
b   8  216  
c  27  343  
d  64  512
```

```
[32]: df1
```

```
[32]:   a  b      c  
a   1  5  heramb  
b   2  6  varsha  
c   3  7  chetan  
d   4  8   patil
```

```
[33]: df
```

```
[33]:   a  b      c  
0   1  5  heramb  
1   2  6  varsha  
2   3  7  chetan  
3   4  8   patil
```

```
[34]: df.sort_values('c')
```

```
[34]:   a  b      c
      2  3  7  chetan
      0  1  5  heramb
      3  4  8   patil
      1  2  6  varsha
```

```
[36]: df.sort_index(ascending= False)
```

```
[36]:   a  b      c
      3  4  8   patil
      2  3  7  chetan
      1  2  6  varsha
      0  1  5  heramb
```

```
[20]: df3 = pd.DataFrame({"desc" : ["Data Science Masters course is highly curated,
↳and uniquely designed according to the latest industry standards. This
↳program instills students the skills essential to knowledge discovery
↳efforts to identify standard, novel, and truly differentiated solutions and
↳decision-making, including skills in managing, querying, analyzing,
↳visualizing, and extracting meaning from extremely large data sets. This
↳trending program provides students with the statistical, mathematical and
↳computational skills needed to meet the large-scale data science challenges
↳of today's professional world. You will learn all the stack required to work
↳in data science industry including cloud infrastructure and real-time
↳industry projects. This course will be taught in Hindi language.", "my name
↳chetan", "i learn data science"]})
```

```
[3]: df3
```

```
[3]:           desc
0  Data Science Masters course is highly curated ...
1                my name chetan
2                i learn data science
```

```
[4]: pd.set_option("display.max_colwidth", 1000)
```

```
[5]: df3
```

```
[5]:           desc
0  Data Science Masters course is highly curated and uniquely designed according
to the latest industry standards. This program instills students the skills
essential to knowledge discovery efforts to identify standard, novel, and truly
differentiated solutions and decision-making, including skills in managing,
querying, analyzing, visualizing, and extracting meaning from extremely large
data sets. This trending program provides students with the statistical,
```

mathematical and computational skills needed to meet the large-scale data science challenges of today's professional world. You will learn all the stack required to work in data science industry including cloud infrastructure and real-time industry projects. This course will be taught in Hindi language.

```
1
my name chetan
2
i learn data science
```

```
[51]: df3['len'] = df3['desc'].apply(len)
```

```
[52]: df3
```

```
[52]:          desc \
0  Data Science Masters course is highly curated and uniquely designed according
to the latest industry standards. This program instills students the skills
essential to knowledge discovery efforts to identify standard, novel, and truly
differentiated solutions and decision-making, including skills in managing,
querying, analyzing, visualizing, and extracting meaning from extremely large
data sets. This trending program provides students with the statistical,
mathematical and computational skills needed to meet the large-scale data
science challenges of today's professional world. You will learn all the stack
required to work in data science industry including cloud infrastructure and
real-time industry projects. This course will be taught in Hindi language.
```

```
1
my name chetan
2
i learn data science
```

```
      len
0  765
1   14
2   20
```

```
[53]: t = "i learn data science"
```

```
[54]: t.split()
```

```
[54]: ['i', 'learn', 'data', 'science']
```

```
[21]: df3['word_count'] = df3['desc'].apply(lambda x : len(x.split()))
```

```
[22]: df3
```

```
[22]:          desc \
0  Data Science Masters course is highly curated and uniquely designed according
to the latest industry standards. This program instills students the skills
```

essential to knowledge discovery efforts to identify standard, novel, and truly differentiated solutions and decision-making, including skills in managing, querying, analyzing, visualizing, and extracting meaning from extremely large data sets. This trending program provides students with the statistical, mathematical and computational skills needed to meet the large-scale data science challenges of today's professional world. You will learn all the stack required to work in data science industry including cloud infrastructure and real-time industry projects. This course will be taught in Hindi language.

```
1
my name chetan
2
i learn data science
```

```
word_count
0         104
1          3
2          4
```

```
[6]: df
```

```
[6]:   a  b      c
0  1  5  heramb
1  2  6  varsha
2  3  7   chetan
3  4  8   patil
```

```
[7]: df['a'][0]
```

```
[7]: 1
```

```
[8]: df['a'].mean()
```

```
[8]: 2.5
```

```
[9]: df['a'].median()
```

```
[9]: 2.5
```

```
[10]: df['a'].mode()
```

```
[10]: 0    1
      1    2
      2    3
      3    4
      Name: a, dtype: int64
```

```
[11]: df['a'].std()
```

```
[11]: 1.2909944487358056
```

```
[12]: df['a'].sum()
```

```
[12]: 10
```

```
[8]: df['a'].min()
```

```
[8]: 1
```

```
[9]: df['a'].max()
```

```
[9]: 4
```

```
[12]: df['a'].var()
```

```
[12]: 1.6666666666666667
```

```
[11]: df['a'].min()
```

```
[11]: 1
```

```
[17]: #Python Pandas Window Functions  
df4 = pd.DataFrame({'a' : [1,2,3,4,2,3,4,56,7]})
```

```
[16]: df4
```

```
[16]:
```

	a
0	1
1	2
2	3
3	4
4	2
5	3
6	4
7	56
8	7

```
[18]: df4['a'].rolling(window = 1).mean()
```

```
[18]:
```

0	1.0
1	2.0
2	3.0
3	4.0
4	2.0
5	3.0
6	4.0


```
7    56.0
8     7.0
Name: a, dtype: float64
```

```
[22]: df4['a'].rolling(window = 2).mean()
```

```
[22]: 0     NaN
      1     1.5
      2     2.5
      3     3.5
      4     3.0
      5     2.5
      6     3.5
      7    30.0
      8    31.5
      Name: a, dtype: float64
```

```
[23]: df4['a'].rolling(window = 5).mean()
```

```
[23]: 0     NaN
      1     NaN
      2     NaN
      3     NaN
      4     2.4
      5     2.8
      6     3.2
      7    13.8
      8    14.4
      Name: a, dtype: float64
```

```
[24]: df4['a'].rolling(window = 3).sum()
```

```
[24]: 0     NaN
      1     NaN
      2     6.0
      3     9.0
      4     9.0
      5     9.0
      6     9.0
      7    63.0
      8    67.0
      Name: a, dtype: float64
```

```
[26]: df4['a'].rolling(window = 4).min()
```

```
[26]: 0     NaN
      1     NaN
```

```
2    NaN
3    1.0
4    2.0
5    2.0
6    2.0
7    2.0
8    3.0
Name: a, dtype: float64
```

```
[27]: df4['a'].rolling(window = 4).max()
```

```
[27]: 0    NaN
      1    NaN
      2    NaN
      3    4.0
      4    4.0
      5    4.0
      6    4.0
      7   56.0
      8   56.0
      Name: a, dtype: float64
```

```
[28]: df4['a'].cumsum()
```

```
[28]: 0     1
      1     3
      2     6
      3    10
      4    12
      5    15
      6    19
      7    75
      8    82
      Name: a, dtype: int64
```

```
[31]: #Python Pandas Date functionality
      date = pd.date_range(start = '2023-02-28', end = '2023-04-30')
```

```
[32]: date
```

```
[32]: DatetimeIndex(['2023-02-28', '2023-03-01', '2023-03-02', '2023-03-03',
                    '2023-03-04', '2023-03-05', '2023-03-06', '2023-03-07',
                    '2023-03-08', '2023-03-09', '2023-03-10', '2023-03-11',
                    '2023-03-12', '2023-03-13', '2023-03-14', '2023-03-15',
                    '2023-03-16', '2023-03-17', '2023-03-18', '2023-03-19',
                    '2023-03-20', '2023-03-21', '2023-03-22', '2023-03-23',
                    '2023-03-24', '2023-03-25', '2023-03-26', '2023-03-27',
```

```

'2023-03-28', '2023-03-29', '2023-03-30', '2023-03-31',
'2023-04-01', '2023-04-02', '2023-04-03', '2023-04-04',
'2023-04-05', '2023-04-06', '2023-04-07', '2023-04-08',
'2023-04-09', '2023-04-10', '2023-04-11', '2023-04-12',
'2023-04-13', '2023-04-14', '2023-04-15', '2023-04-16',
'2023-04-17', '2023-04-18', '2023-04-19', '2023-04-20',
'2023-04-21', '2023-04-22', '2023-04-23', '2023-04-24',
'2023-04-25', '2023-04-26', '2023-04-27', '2023-04-28',
'2023-04-29', '2023-04-30'],
dtype='datetime64[ns]', freq='D')

```

```
[33]: df_date = pd.DataFrame({'date' : date})
```

```
[34]: df_date
```

```
[34]:
      date
0  2023-02-28
1  2023-03-01
2  2023-03-02
3  2023-03-03
4  2023-03-04
..      ...
57 2023-04-26
58 2023-04-27
59 2023-04-28
60 2023-04-29
61 2023-04-30

```

```
[62 rows x 1 columns]
```

```
[36]: df7 = pd.DataFrame({'date': ['2023-04-30', '2023-04-26', '2023-03-01']})
```

```
[37]: df7
```

```
[37]:
      date
0  2023-04-30
1  2023-04-26
2  2023-03-01

```

```
[39]: df7.dtypes
```

```
[39]: date    object
      dtype: object

```

```
[40]: df_date.dtypes
```

```
[40]: date      datetime64[ns]
      dtype: object
```

```
[42]: df7['updated_date'] = pd.to_datetime(df7['date'])
```

```
[43]: df7
```

```
[43]:      date updated_date
0  2023-04-30  2023-04-30
1  2023-04-26  2023-04-26
2  2023-03-01  2023-03-01
```

```
[45]: df7.dtypes
```

```
[45]: date      object
      updated_date  datetime64[ns]
      dtype: object
```

```
[46]: df7['year'] = df7['updated_date'].dt.year
```

```
[47]: df7
```

```
[47]:      date updated_date  year
0  2023-04-30  2023-04-30  2023
1  2023-04-26  2023-04-26  2023
2  2023-03-01  2023-03-01  2023
```

```
[52]: df7['day'] = df7['updated_date'].dt.day
```

```
[53]: df7
```

```
[53]:      date updated_date  year  day
0  2023-04-30  2023-04-30    4   30
1  2023-04-26  2023-04-26    4   26
2  2023-03-01  2023-03-01    3    1
```

```
[54]: df7['month'] = df7['updated_date'].dt.month
```

```
[55]: df7
```

```
[55]:      date updated_date  year  day  month
0  2023-04-30  2023-04-30    4   30     4
1  2023-04-26  2023-04-26    4   26     4
2  2023-03-01  2023-03-01    3    1     3
```

```
[58]: #Python Pandas - Time Delta
      pd.Timedelta(days = 1, hours = 5, minutes =30)
```

```
[58]: Timedelta('1 days 05:30:00')
```

```
[59]: dt = pd.to_datetime('2023-04-26')
```

```
[60]: td = pd.Timedelta(days = 1)
```

```
[61]: dt + td
```

```
[61]: Timestamp('2023-04-27 00:00:00')
```

```
[67]: #Python Pandas - Categorical Data  
data = ["heramb", "varsha", "chetan", "patil", "chetan", "heramb", "heramb"]
```

```
[68]: cat = pd.Categorical(data)
```

```
[69]: cat
```

```
[69]: ['heramb', 'varsha', 'chetan', 'patil', 'chetan', 'heramb', 'heramb']  
Categories (4, object): ['chetan', 'heramb', 'patil', 'varsha']
```

```
[70]: cat.value_counts()
```

```
[70]: chetan    2  
heramb    3  
patil     1  
varsha    1  
dtype: int64
```

```
[71]: #Python Pandas Visualization
```

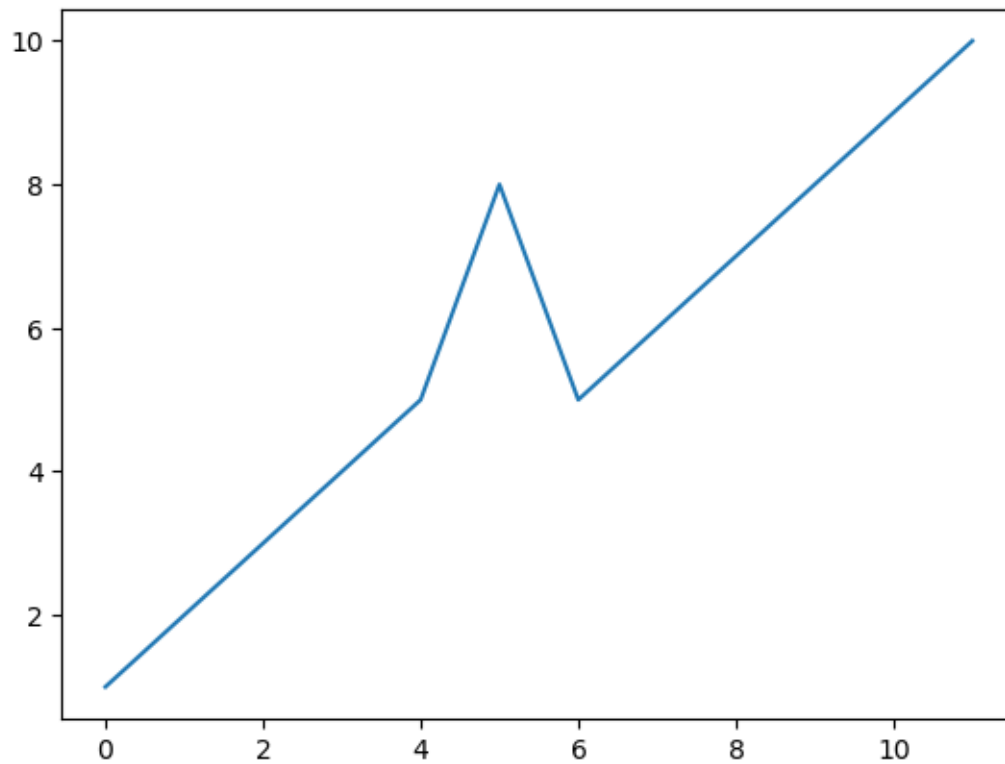
```
[76]: d = pd.Series([1,2,3,4,5,8,5,6,7,8,9,10])
```

```
[77]: d
```

```
[77]: 0      1  
1      2  
2      3  
3      4  
4      5  
5      8  
6      5  
7      6  
8      7  
9      8  
10     9  
11    10  
dtype: int64
```

```
[75]: d.plot()
```

```
[75]: <AxesSubplot: >
```



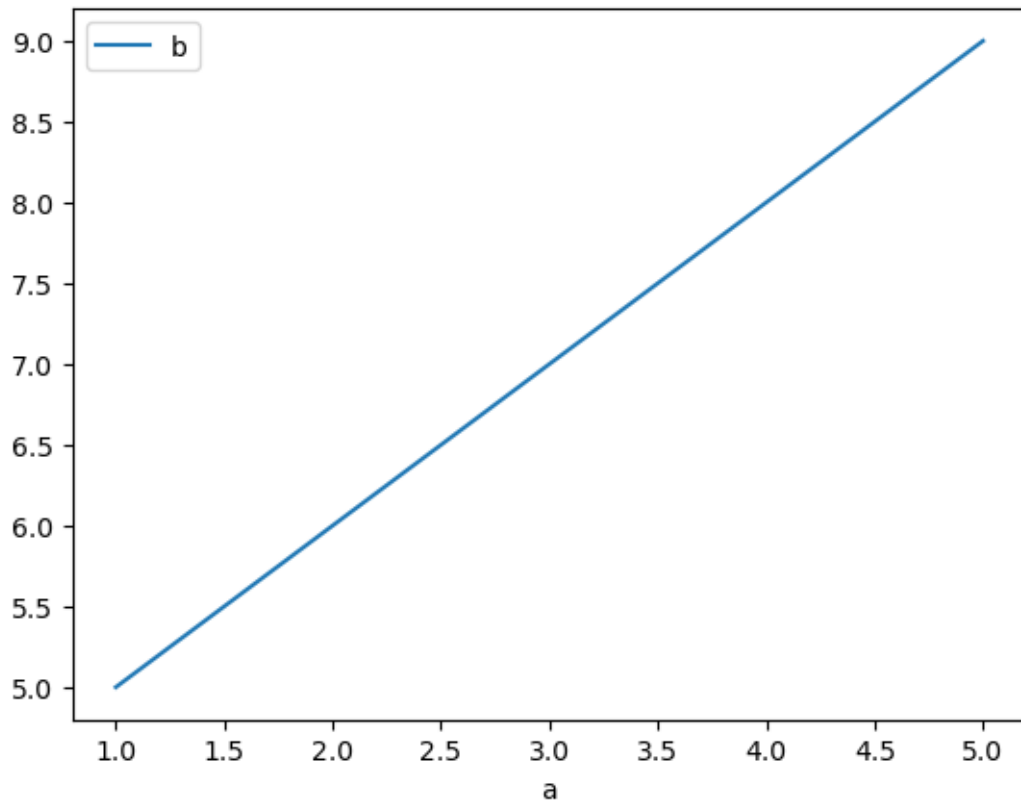
```
[78]: df = pd.DataFrame({'a' : [1,2,3,4,5],  
                        'b': [5,6,7,8,9]})
```

```
[79]: df
```

```
[79]:    a  b  
0   1  5  
1   2  6  
2   3  7  
3   4  8  
4   5  9
```

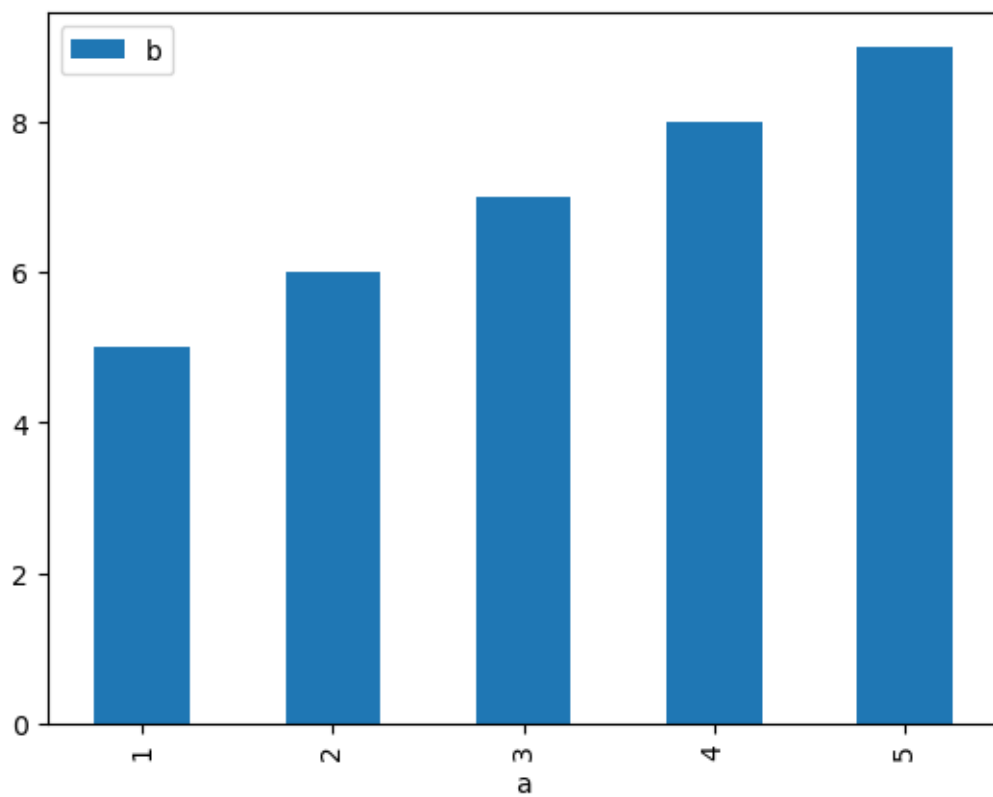
```
[80]: df.plot(x= 'a',y = 'b')
```

```
[80]: <AxesSubplot: xlabel='a'>
```



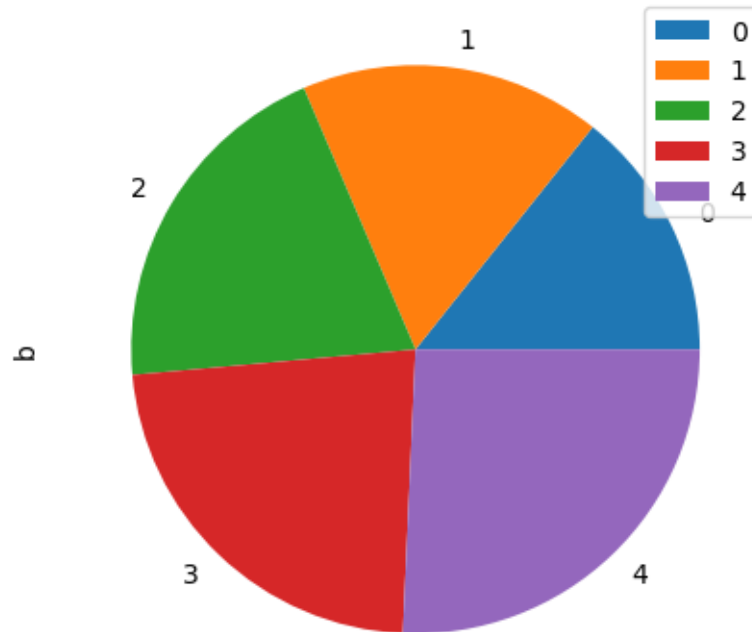
```
[81]: df.plot.bar(x= 'a',y = 'b')
```

```
[81]: <AxesSubplot: xlabel='a'>
```



```
[82]: df.plot.pie(x= 'a',y = 'b')
```

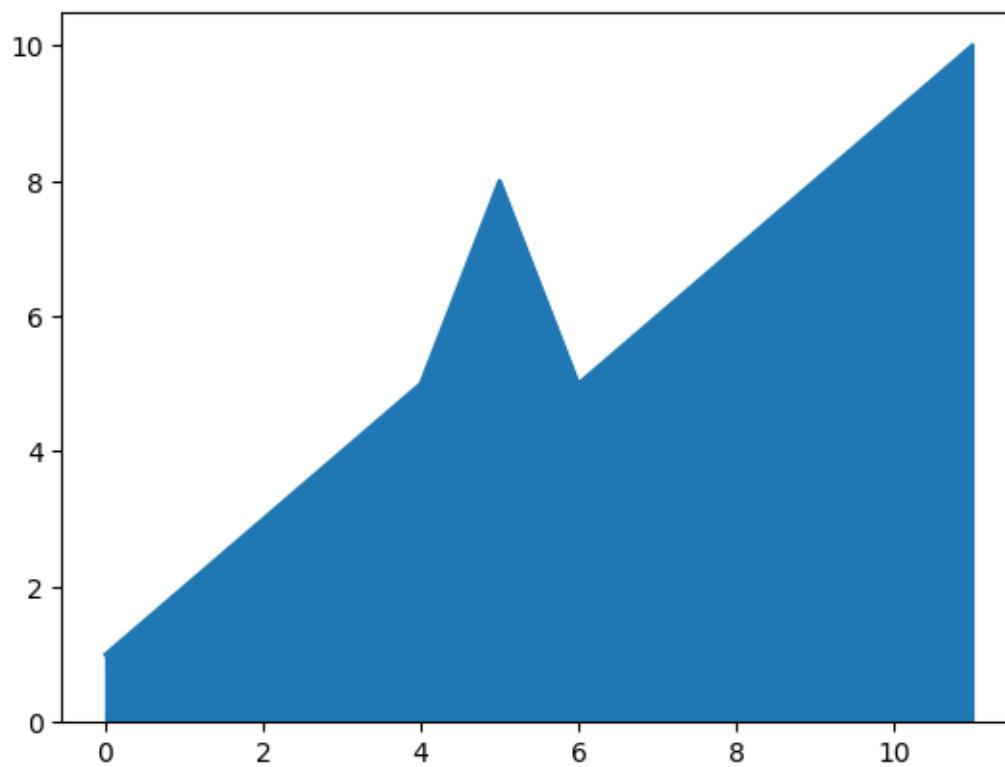
```
[82]: <AxesSubplot: ylabel='b'>
```

```
[83]: d = pd.Series([1,2,3,4,5,8,5,6,7,8,9,10])
```

```
[84]: d.plot.area()
```

```
[84]: <AxesSubplot: >
```



[]: