Basic Analysis using numpy and pandas

Breast Cancer Prediction dataset

To import library

In [1]:

```
import numpy as np
import pandas as pd
```

To import dataset

In [2]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\8_BreastCancerPrediction.csv")
d
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	
...	...	...	...	...	...	...	
564	926424	M	21.56	22.39	142.00	1479.0	
565	926682	M	20.13	28.25	131.20	1261.0	
566	926954	M	16.60	28.08	108.30	858.1	
567	927241	M	20.60	29.33	140.10	1265.0	
568	92751	B	7.76	24.54	47.92	181.0	

569 rows × 33 columns

To get top 10 record

In [3]:

```
d.head(10)
```

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	
5	843786	M	12.45	15.70	82.57	477.1	
6	844359	M	18.25	19.98	119.60	1040.0	
7	84458202	M	13.71	20.83	90.20	577.9	
8	844981	M	13.00	21.82	87.50	519.8	
9	84501001	M	12.46	24.04	83.97	475.9	

10 rows × 33 columns

To get last 10

In [4]:

```
d.tail(10)
```

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness
559	925291	B	11.51	23.93	74.52	403.5	
560	925292	B	14.05	27.15	91.38	600.4	
561	925311	B	11.20	29.37	70.67	386.0	
562	925622	M	15.22	30.62	103.40	716.9	
563	926125	M	20.92	25.09	143.00	1347.0	
564	926424	M	21.56	22.39	142.00	1479.0	
565	926682	M	20.13	28.25	131.20	1261.0	
566	926954	M	16.60	28.08	108.30	858.1	
567	927241	M	20.60	29.33	140.10	1265.0	
568	92751	B	7.76	24.54	47.92	181.0	

10 rows × 33 columns

To describe statistics Analysis

In [5]:

```
d.describe()
```

Out[5]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.0(
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.09
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.0'
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.05
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.08
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.09
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.1(
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.16

8 rows × 32 columns

To get rows and columns

In [6]:

```
np.shape(d)
```

Out[6]:

```
(569, 33)
```

To get number of elements

In [7]:

```
np.size(d)
```

Out[7]:

```
18777
```

To get the missing value

In [8]:

```
d.isna()
```

Out[8]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
...	...	...	...	...	...	...	
564	False	False	False	False	False	False	
565	False	False	False	False	False	False	
566	False	False	False	False	False	False	
567	False	False	False	False	False	False	
568	False	False	False	False	False	False	

569 rows × 33 columns

To drop the missing elements

In [9]:

```
d.dropna(axis=1,how='any')
```

Out[9]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	
...	...	...	...	...	...	...	
564	926424	M	21.56	22.39	142.00	1479.0	
565	926682	M	20.13	28.25	131.20	1261.0	
566	926954	M	16.60	28.08	108.30	858.1	
567	927241	M	20.60	29.33	140.10	1265.0	
568	92751	B	7.76	24.54	47.92	181.0	

569 rows × 32 columns

In [10]:

```
d["id"]
```

Out[10]:

```
0        842302
1        842517
2      84300903
3      84348301
4      84358402
         ...
564      926424
565      926682
566      926954
567      927241
568       92751
Name: id, Length: 569, dtype: int64
```

In [11]:

```
data=pd.DataFrame(d[['radius_mean','texture_mean']][0:500])
data
```

Out[11]:

	radius_mean	texture_mean
0	17.99	10.38
1	20.57	17.77
2	19.69	21.25
3	11.42	20.38
4	20.29	14.34
...	...	...
495	14.87	20.21
496	12.65	18.17
497	12.47	17.31
498	18.49	17.52
499	20.59	21.24

500 rows × 2 columns

In [12]:

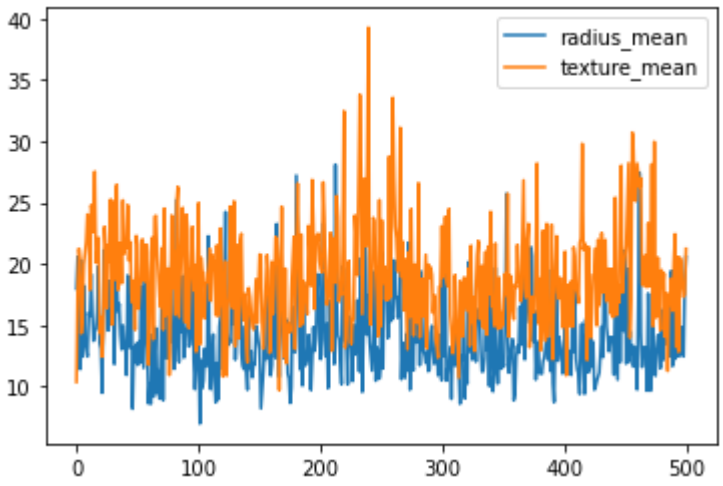
```
import matplotlib.pyplot as pp
```

In [13]:

```
data.plot.line()
```

Out[13]:

```
<AxesSubplot:>
```

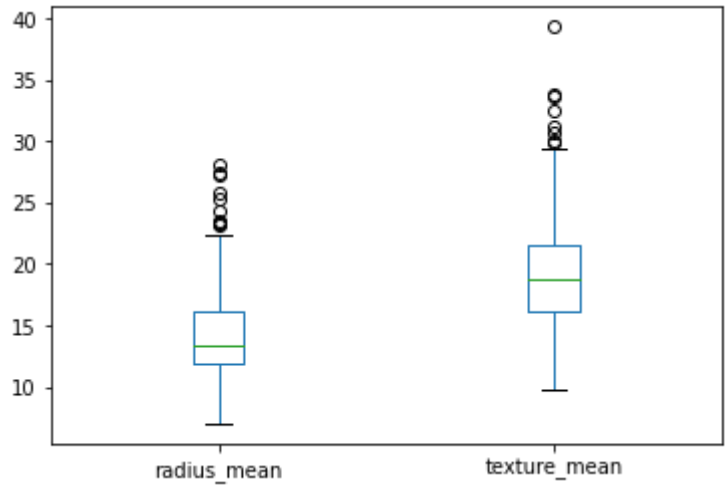


In [14]:

```
data.plot.box()
```

Out[14]:

```
<AxesSubplot:>
```

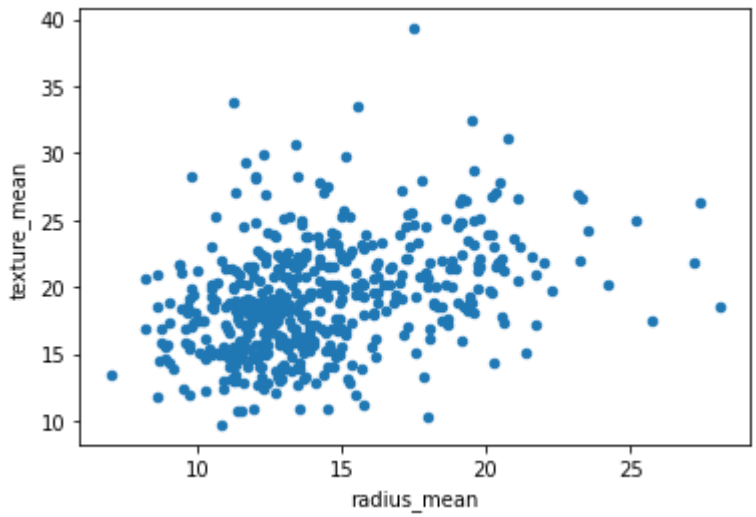


In [15]:

```
data.plot.scatter(x="radius_mean",y="texture_mean")
```

Out[15]:

```
<AxesSubplot:xlabel='radius_mean', ylabel='texture_mean'>
```

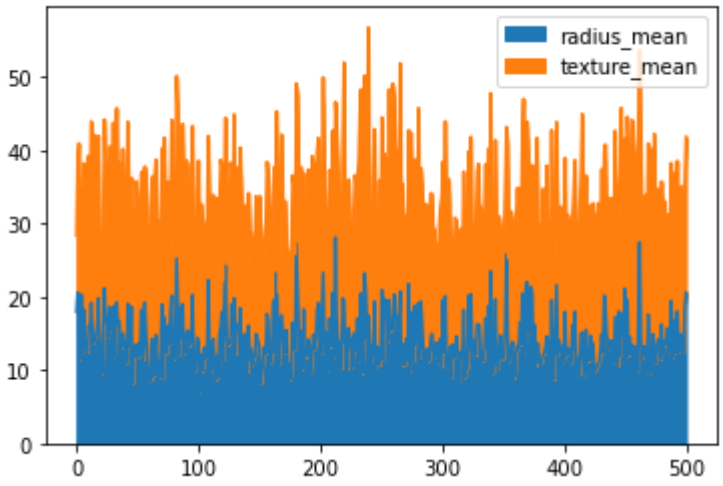


In [16]:

```
data.plot.area()
```

Out[16]:

```
<AxesSubplot:>
```

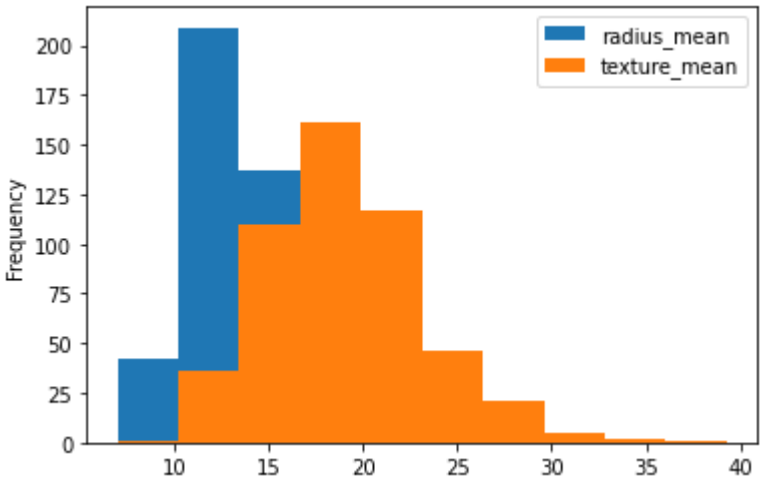


In [17]:

```
data.plot.hist()
```

Out[17]:

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

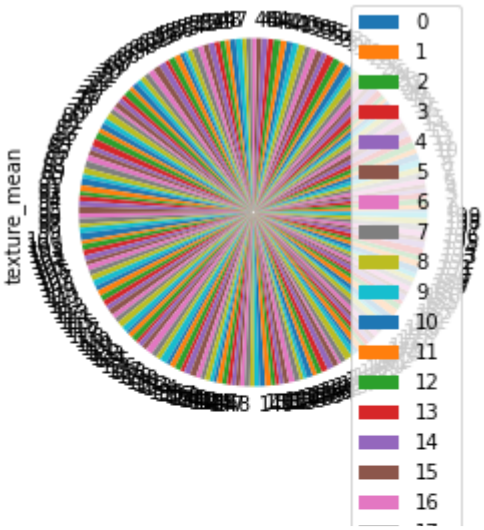


In [18]:

```
data=pd.DataFrame(d[['radius_mean','texture_mean']][0:200])
data.plot.pie(y="texture_mean")
```

Out[18]:

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



In [ ]: