

# Basic Analysis using Numpy and Pandas

## import libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib as pp
```

## import dataset

In [2]:

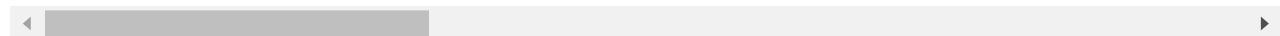
```
data=pd.read_csv(r"E:\154\8_BreastCancerPrediction - 8_BreastCancerPrediction.csv")
```

In [3]:

```
display(data)
```

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>compactness_mean</b>	<b>concavity_mean</b>	<b>concave points_mean</b>	<b>symmetry_mean</b>	<b>fractal_dimension_mean</b>	<b>radius_se</b>	<b>texture_se</b>	<b>perimeter_se</b>	<b>area_se</b>	<b>smoothness_se</b>	<b>compactness_se</b>	<b>concavity_se</b>	<b>concave points_se</b>	<b>symmetry_se</b>	<b>fractal_dimension_se</b>	<b>radius_worst</b>	<b>texture_worst</b>	<b>perimeter_worst</b>	<b>area_worst</b>	<b>smoothness_worst</b>	<b>compactness_worst</b>	<b>concavity_worst</b>	<b>concave points_worst</b>	<b>symmetry_worst</b>	<b>fractal_dimension_worst</b>	
<b>0</b>	842302	M	17.99	10.38	122.80	1001.0	0.11840																										
<b>1</b>	842517	M	20.57	17.77	132.90	1326.0	0.08474																										
<b>2</b>	84300903	M	19.69	21.25	130.00	1203.0	0.10960																										
<b>3</b>	84348301	M	11.42	20.38	77.58	386.1	0.14250																										
<b>4</b>	84358402	M	20.29	14.34	135.10	1297.0	0.10030																										
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...		
<b>564</b>	926424	M	21.56	22.39	142.00	1479.0	0.11100																										
<b>565</b>	926682	M	20.13	28.25	131.20	1261.0	0.09780																										
<b>566</b>	926954	M	16.60	28.08	108.30	858.1	0.08455																										
<b>567</b>	927241	M	20.60	29.33	140.10	1265.0	0.11780																										
<b>568</b>	92751	B	7.76	24.54	47.92	181.0	0.05263																										

569 rows × 32 columns



## To display top 10 rows

In [4]:

```
data.head()
```

Out[4]:

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>compactness_mean</b>	<b>concavity_mean</b>	<b>concave points_mean</b>	<b>symmetry_mean</b>	<b>fractal_dimension_mean</b>	<b>radius_se</b>	<b>texture_se</b>	<b>perimeter_se</b>	<b>area_se</b>	<b>smoothness_se</b>	<b>compactness_se</b>	<b>concavity_se</b>	<b>concave points_se</b>	<b>symmetry_se</b>	<b>fractal_dimension_se</b>	<b>radius_worst</b>	<b>texture_worst</b>	<b>perimeter_worst</b>	<b>area_worst</b>	<b>smoothness_worst</b>	<b>compactness_worst</b>	<b>concavity_worst</b>	<b>concave points_worst</b>	<b>symmetry_worst</b>	<b>fractal_dimension_worst</b>
<b>0</b>	842302	M	17.99	10.38	122.80	1001.0	0.11840																									

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>cor</b>
<b>1</b>	842517	M	20.57	17.77	132.90	1326.0	0.08474	
<b>2</b>	84300903	M	19.69	21.25	130.00	1203.0	0.10960	
<b>3</b>	84348301	M	11.42	20.38	77.58	386.1	0.14250	
<b>4</b>	84358402	M	20.29	14.34	135.10	1297.0	0.10030	

5 rows × 32 columns

## To display last 5 rows

In [5]:

`data.tail()`

Out[5]:

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>cor</b>
<b>564</b>	926424	M	21.56	22.39	142.00	1479.0	0.11100	
<b>565</b>	926682	M	20.13	28.25	131.20	1261.0	0.09780	
<b>566</b>	926954	M	16.60	28.08	108.30	858.1	0.08455	
<b>567</b>	927241	M	20.60	29.33	140.10	1265.0	0.11780	
<b>568</b>	92751	B	7.76	24.54	47.92	181.0	0.05263	

5 rows × 32 columns

In [6]:

`data.dtypes`

Out[6]:

<b>id</b>	<b>int64</b>
<b>diagnosis</b>	<b>object</b>
<b>radius_mean</b>	<b>float64</b>
<b>texture_mean</b>	<b>float64</b>
<b>perimeter_mean</b>	<b>float64</b>
<b>area_mean</b>	<b>float64</b>
<b>smoothness_mean</b>	<b>float64</b>
<b>compactness_mean</b>	<b>float64</b>
<b>concavity_mean</b>	<b>float64</b>
<b>concave points_mean</b>	<b>float64</b>
<b>symmetry_mean</b>	<b>float64</b>
<b>fractal_dimension_mean</b>	<b>float64</b>
<b>radius_se</b>	<b>float64</b>
<b>texture_se</b>	<b>float64</b>
<b>perimeter_se</b>	<b>float64</b>
<b>area_se</b>	<b>float64</b>
<b>smoothness_se</b>	<b>float64</b>
<b>compactness_se</b>	<b>float64</b>
<b>concavity_se</b>	<b>float64</b>
<b>concave points_se</b>	<b>float64</b>
<b>symmetry_se</b>	<b>float64</b>
<b>fractal_dimension_se</b>	<b>float64</b>

```
radius_worst          float64
texture_worst         float64
perimeter_worst       float64
area_worst            float64
smoothness_worst      float64
compactness_worst     float64
concavity_worst       float64
concave points_worst float64
symmetry_worst        float64
fractal_dimension_worst float64
dtype: object
```

## To view statistical summary

In [7]:

```
data.describe()
```

Out[7]:

	<b>id</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>com </b>
<b>count</b>	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	
<b>mean</b>	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	
<b>std</b>	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	
<b>min</b>	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	
<b>25%</b>	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	
<b>50%</b>	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	
<b>75%</b>	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	
<b>max</b>	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	

8 rows × 31 columns



## To Print no of elements

In [8]:

```
data.size
```

Out[8]: 18208

In [9]:

```
data.ndim
```

Out[9]: 2

## To print no of rows and columns

In [10]:

```
data.shape
```

Out[10]: (569, 32)

## To find missing values

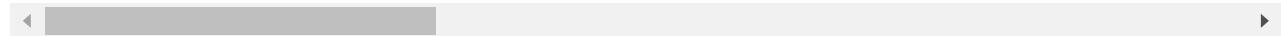
In [11]:

```
data.isna()
```

Out[11]:

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>comp</b>
<b>0</b>	False	False	False	False	False	False	False	False
<b>1</b>	False	False	False	False	False	False	False	False
<b>2</b>	False	False	False	False	False	False	False	False
<b>3</b>	False	False	False	False	False	False	False	False
<b>4</b>	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
<b>564</b>	False	False	False	False	False	False	False	False
<b>565</b>	False	False	False	False	False	False	False	False
<b>566</b>	False	False	False	False	False	False	False	False
<b>567</b>	False	False	False	False	False	False	False	False
<b>568</b>	False	False	False	False	False	False	False	False

569 rows × 32 columns



## To drop null values with constatns

In [12]:

```
data.fillna(5)
```

Out[12]:

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>comp</b>
<b>0</b>	842302	M	17.99	10.38	122.80	1001.0	0.11840	5
<b>1</b>	842517	M	20.57	17.77	132.90	1326.0	0.08474	5
<b>2</b>	84300903	M	19.69	21.25	130.00	1203.0	0.10960	5
<b>3</b>	84348301	M	11.42	20.38	77.58	386.1	0.14250	5
<b>4</b>	84358402	M	20.29	14.34	135.10	1297.0	0.10030	5
...	...	...	...	...	...	...	...	...
<b>564</b>	926424	M	21.56	22.39	142.00	1479.0	0.11100	5
<b>565</b>	926682	M	20.13	28.25	131.20	1261.0	0.09780	5
<b>566</b>	926954	M	16.60	28.08	108.30	858.1	0.08455	5
<b>567</b>	927241	M	20.60	29.33	140.10	1265.0	0.11780	5

	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>...</b>
<b>568</b>	92751	B	7.76	24.54	47.92	181.0	0.05263	

569 rows × 32 columns

In [13]: `data.dropna()`

Out[13]:

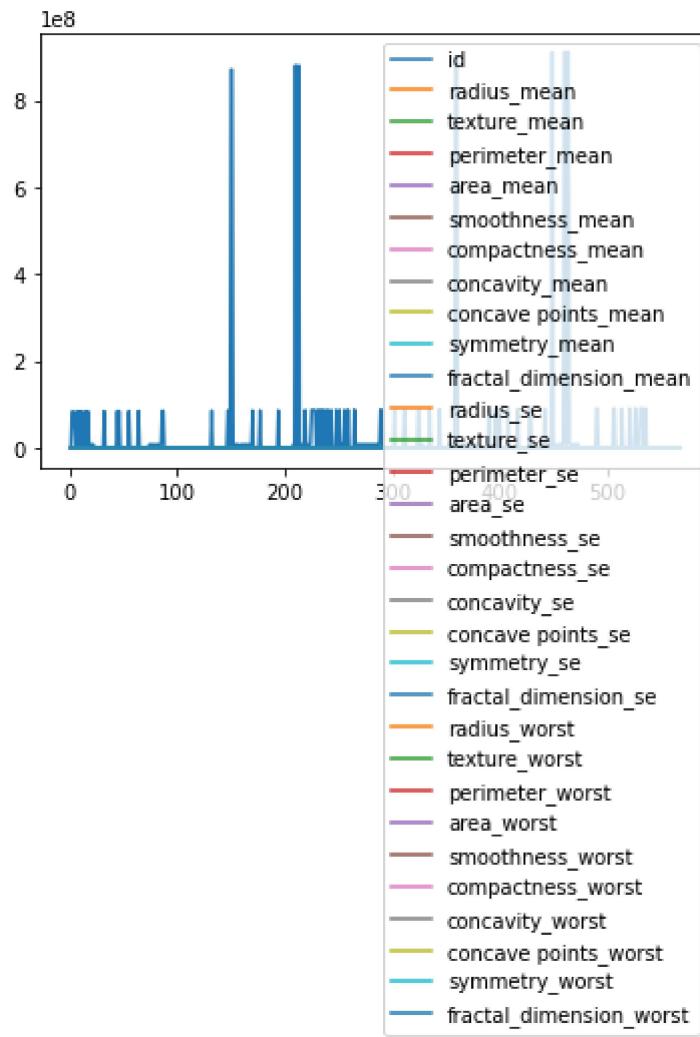
	<b>id</b>	<b>diagnosis</b>	<b>radius_mean</b>	<b>texture_mean</b>	<b>perimeter_mean</b>	<b>area_mean</b>	<b>smoothness_mean</b>	<b>...</b>
<b>0</b>	842302	M	17.99	10.38	122.80	1001.0	0.11840	
<b>1</b>	842517	M	20.57	17.77	132.90	1326.0	0.08474	
<b>2</b>	84300903	M	19.69	21.25	130.00	1203.0	0.10960	
<b>3</b>	84348301	M	11.42	20.38	77.58	386.1	0.14250	
<b>4</b>	84358402	M	20.29	14.34	135.10	1297.0	0.10030	
...	...	...	...	...	...	...	...	...
<b>564</b>	926424	M	21.56	22.39	142.00	1479.0	0.11100	
<b>565</b>	926682	M	20.13	28.25	131.20	1261.0	0.09780	
<b>566</b>	926954	M	16.60	28.08	108.30	858.1	0.08455	
<b>567</b>	927241	M	20.60	29.33	140.10	1265.0	0.11780	
<b>568</b>	92751	B	7.76	24.54	47.92	181.0	0.05263	

569 rows × 32 columns

## Line Plot

In [14]: `data.plot.line()`

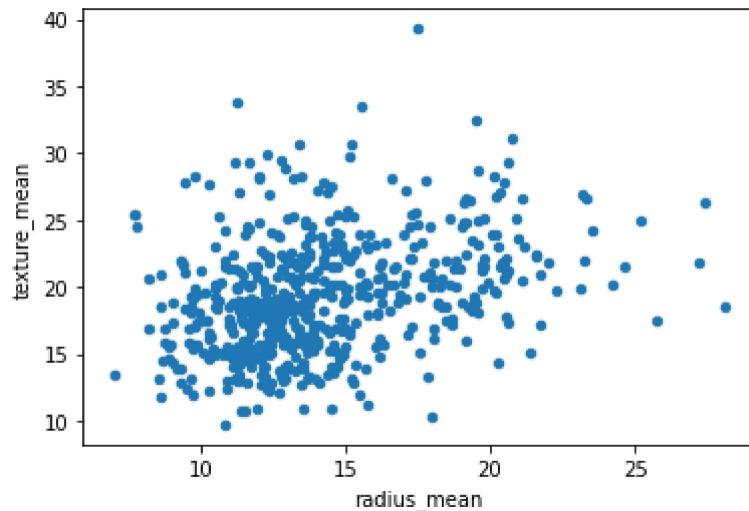
Out[14]: &lt;AxesSubplot:&gt;



## Scatter Plot

```
In [16]: data.plot.scatter(x='radius_mean',y='texture_mean')
```

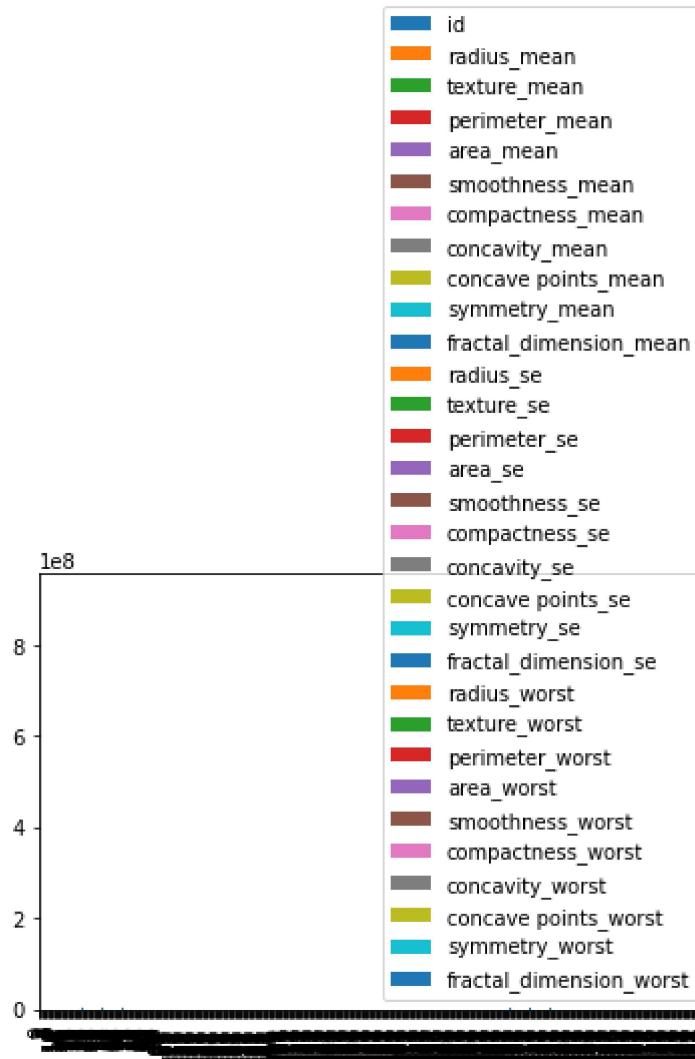
```
Out[16]: <AxesSubplot:xlabel='radius_mean', ylabel='texture_mean'>
```



## Bar Chart

```
In [17]: data.plot.bar()
```

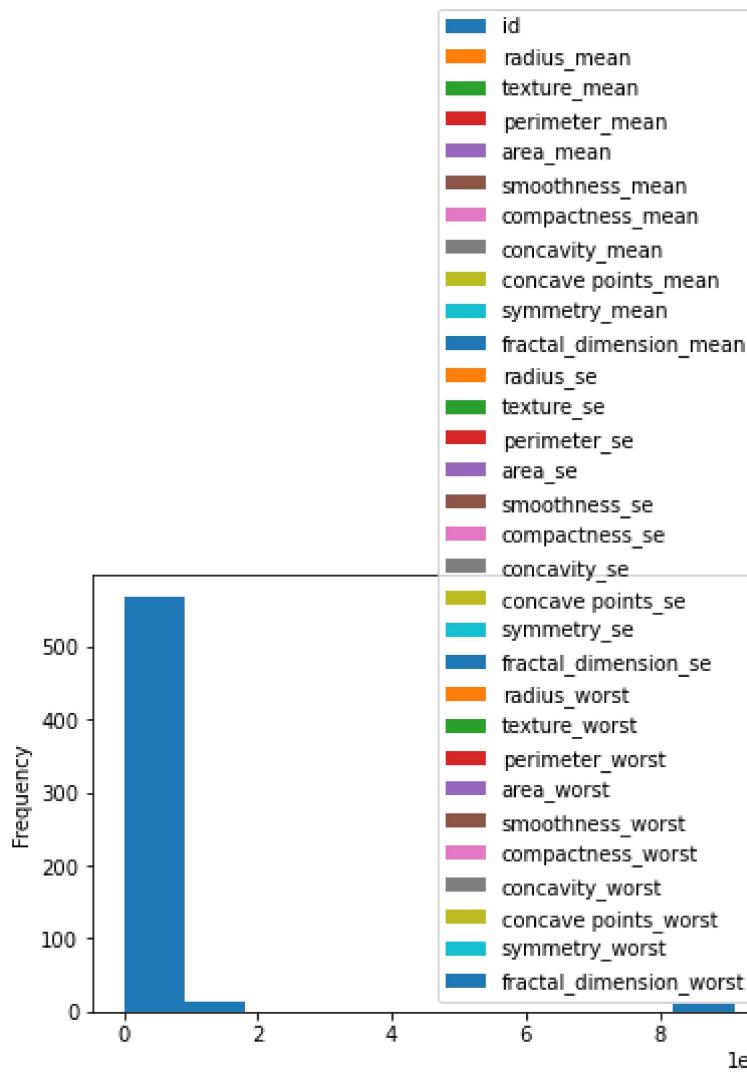
```
Out[17]: <AxesSubplot:>
```



## Histogram

```
In [18]: data.plot.hist()
```

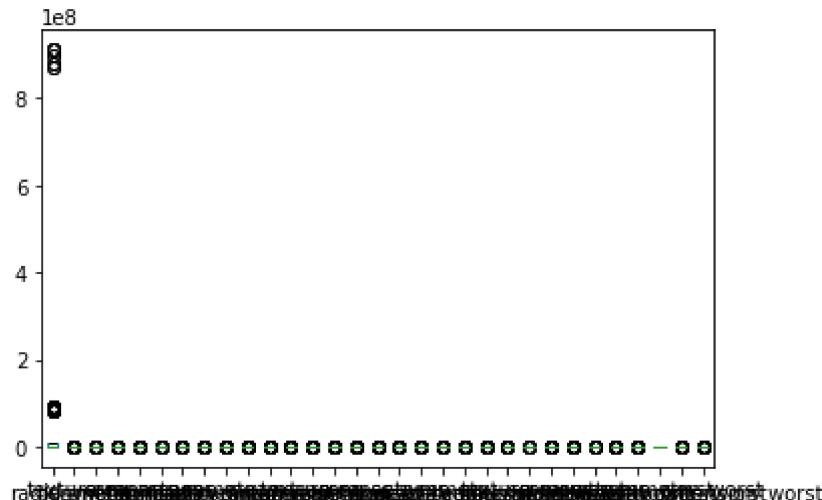
```
Out[18]: <AxesSubplot:ylabel='Frequency'>
```



## Box Plot

In [19]: `data.plot.box()`

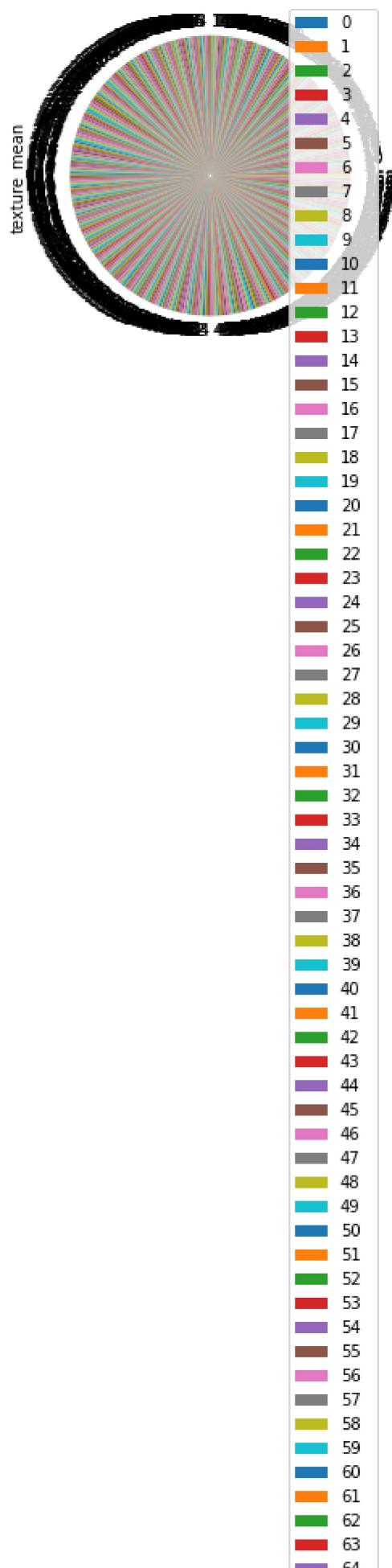
Out[19]: <AxesSubplot:>

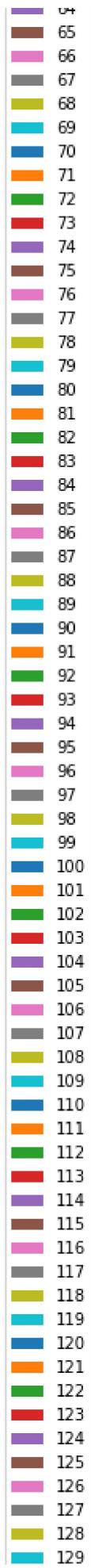


## Pie Chart

```
In [20]: data.plot.pie(y="texture_mean")
```

```
Out[20]: <AxesSubplot:ylabel='texture_mean'>
```

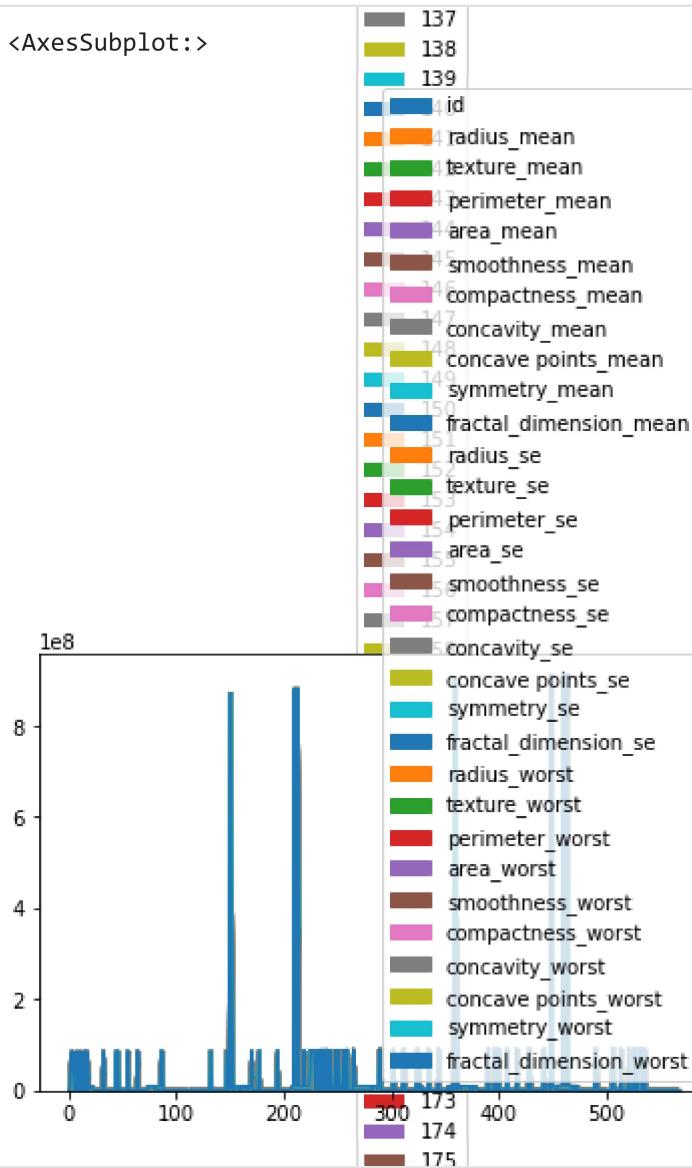




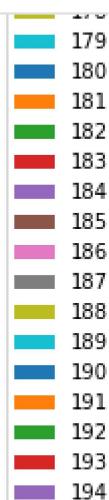
## Area

```
In [21]: data.plot.area()
```

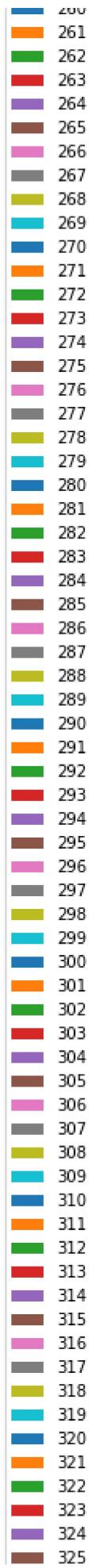
```
Out[21]: <AxesSubplot:>
```



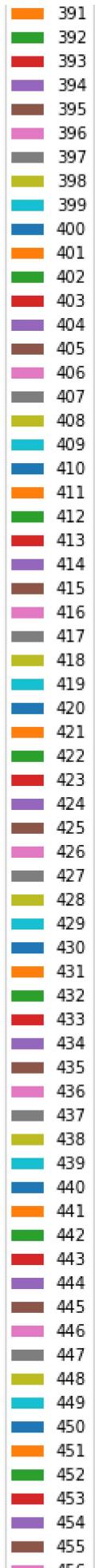
```
In [ ]:
```



■	195
■	196
■	197
■	198
■	199
■	200
■	201
■	202
■	203
■	204
■	205
■	206
■	207
■	208
■	209
■	210
■	211
■	212
■	213
■	214
■	215
■	216
■	217
■	218
■	219
■	220
■	221
■	222
■	223
■	224
■	225
■	226
■	227
■	228
■	229
■	230
■	231
■	232
■	233
■	234
■	235
■	236
■	237
■	238
■	239
■	240
■	241
■	242
■	243
■	244
■	245
■	246
■	247
■	248
■	249
■	250
■	251
■	252
■	253
■	254
■	255
■	256
■	257
■	258
■	259
■	260



■	326
■	327
■	328
■	329
■	330
■	331
■	332
■	333
■	334
■	335
■	336
■	337
■	338
■	339
■	340
■	341
■	342
■	343
■	344
■	345
■	346
■	347
■	348
■	349
■	350
■	351
■	352
■	353
■	354
■	355
■	356
■	357
■	358
■	359
■	360
■	361
■	362
■	363
■	364
■	365
■	366
■	367
■	368
■	369
■	370
■	371
■	372
■	373
■	374
■	375
■	376
■	377
■	378
■	379
■	380
■	381
■	382
■	383
■	384
■	385
■	386
■	387
■	388
■	389
■	390



■	450
■	457
■	458
■	459
■	460
■	461
■	462
■	463
■	464
■	465
■	466
■	467
■	468
■	469
■	470
■	471
■	472
■	473
■	474
■	475
■	476
■	477
■	478
■	479
■	480
■	481
■	482
■	483
■	484
■	485
■	486
■	487
■	488
■	489
■	490
■	491
■	492
■	493
■	494
■	495
■	496
■	497
■	498
■	499
■	500
■	501
■	502
■	503
■	504
■	505
■	506
■	507
■	508
■	509
■	510
■	511
■	512
■	513
■	514
■	515
■	516
■	517
■	518
■	519
■	520
■	521

522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568