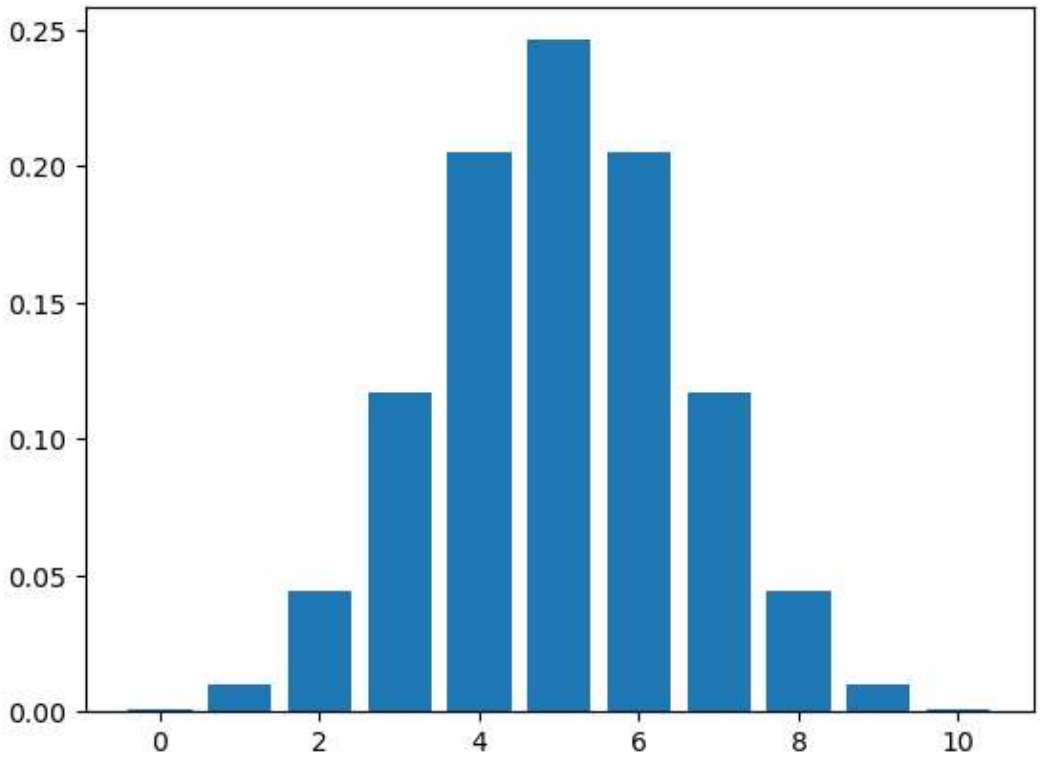
PROBLEM STATEMENT:

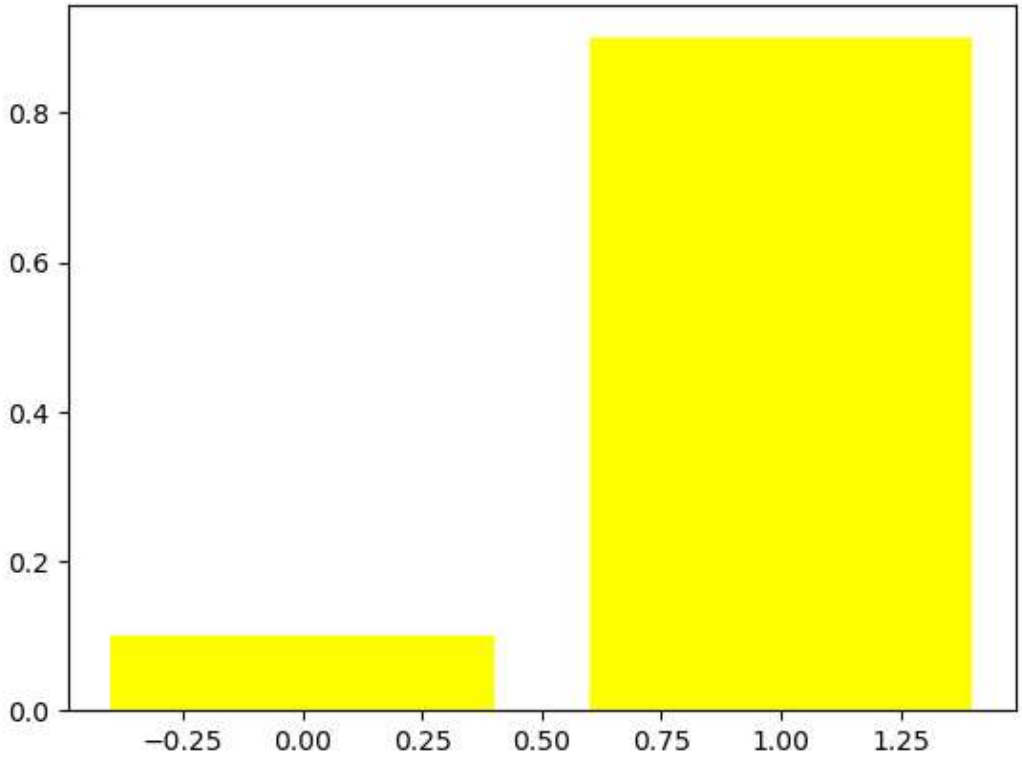
1. Perform Binomial, Bernoulli distributions

In [41]:
```
import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt
```

In [19]:
```
a=10
b=0.5
r=list(range(a+1))
dt=[binom.pmf(r,a,b) for r in r]
plt.bar(r,dt)
plt.show()
```

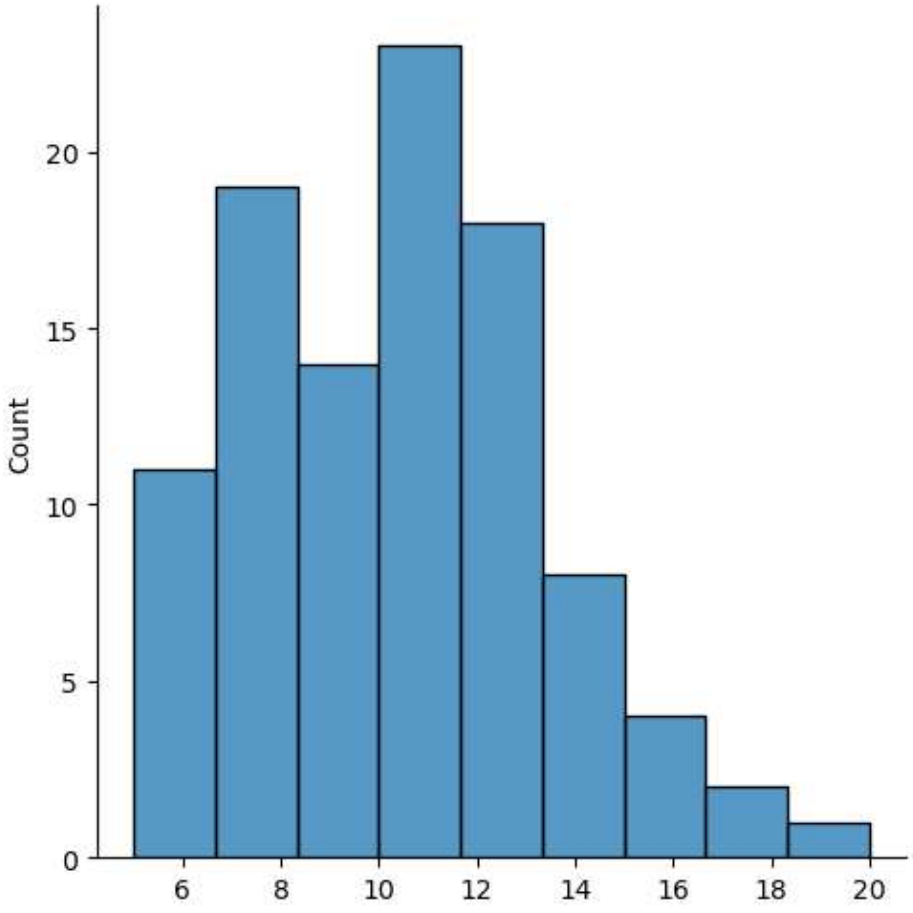


In [177]:
```
from scipy.stats import bernoulli
bd=bernoulli(0.9)
y=[0,1]
plt.bar(y,bd.pmf(y),color='yellow')
plt.show()
```



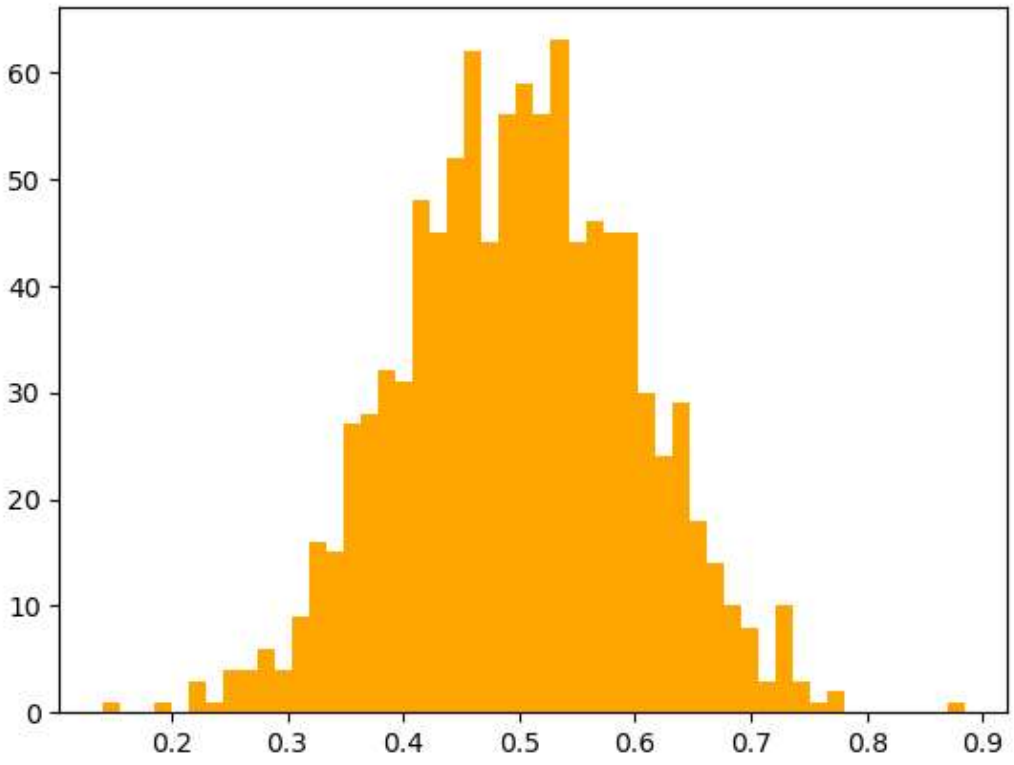
2. Perform Poisson distribution

In [38]:
```
from numpy import random
import seaborn as sea
sea.displot(random.poisson(lam=10,size=100))
plt.show()
```

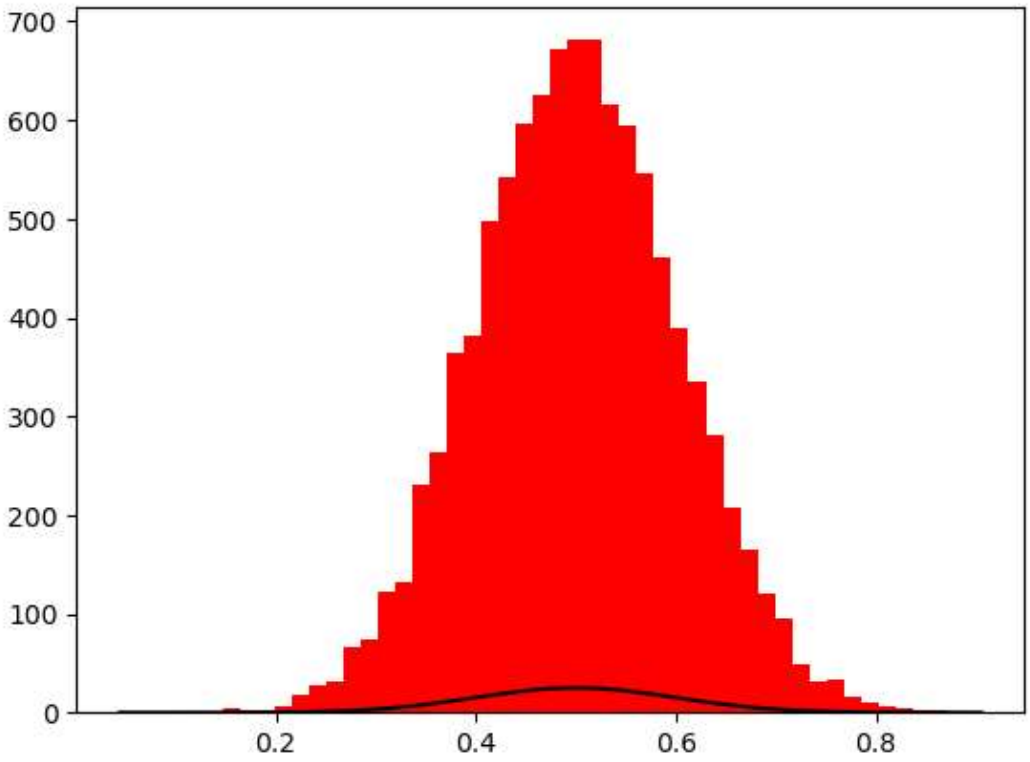


3. Perform Normal, Exponential distributions

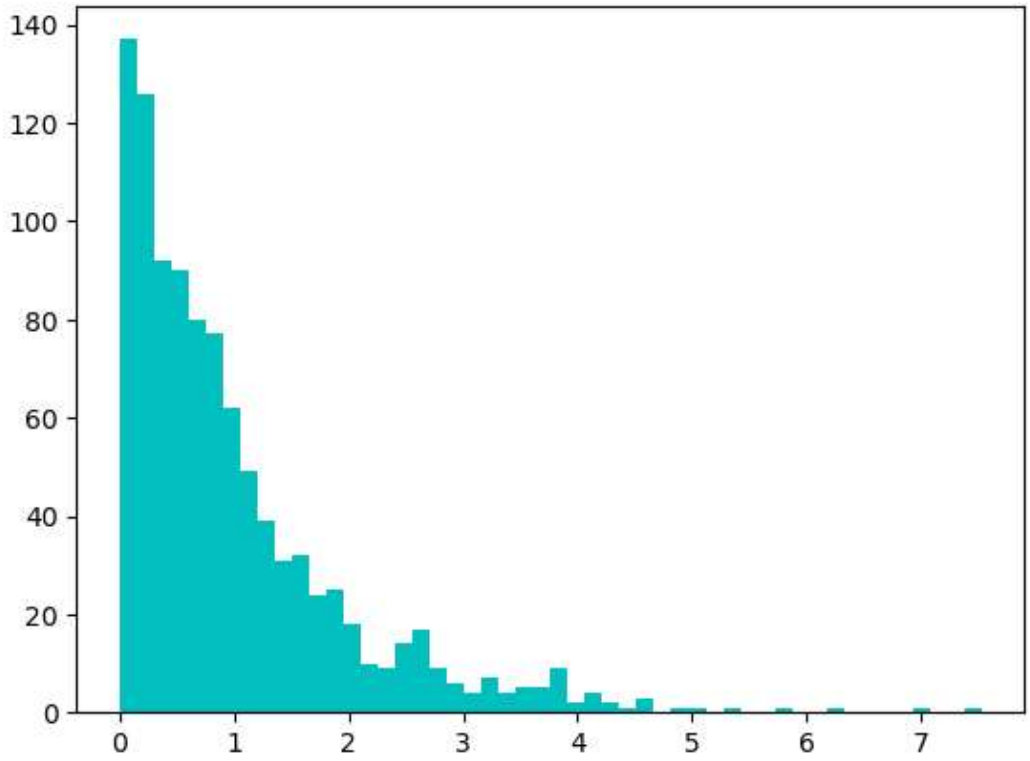
In [178]:
```
p,q=0.5,0.1
x=np.random.normal(p,q,1000)
count,bins,ignored=plt.hist(x,50,color='orange')
plt.show()
```



In [176]:
```
p,q=0.5,0.1
x=np.random.normal(p,q,10000)
count,bins,ignored=plt.hist(x,50,color='red')
plt.plot(bins,1/sigma*np.sqrt(2*np.pi)*np.exp(-(bins-mu)**2/(2*sigma**2)),color='blac
plt.show()
```



In [173]:
```
exp=np.random.exponential(1,1000)
c,bs,ig=plt.hist(exp,50,color='c')
plt.show()
```



EDA:

Take any three datasets and perform EDA with Data Collection, Data Cleaning and Pre-processing (such as finding missing values, replacing or dropping missing values) and data visualization (Apply all the possible functions what we have learnt so far)"

D1

In [90]:
```
import pandas as pd
```

In [91]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\8_BreastCancerPrediction.csv")
df
```

Out[91]:

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

569 rows × 33 columns

In [97]:
```
df.isnull()
```

Out[97]:

_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave points_worst
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
...	...	...	...	...	...	...
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False
False	False	False	False	False	False	False

In [98]:
```
df.isnull().sum()
```

Out[98]:
```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
concave points_worst       0
symmetry_worst             0
fractal_dimension_worst    0
Unnamed: 32              569
dtype: int64
```

In [100]:
```
df1=df.drop("Unnamed: 32",axis=1)
df1
```

Out[100]:

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

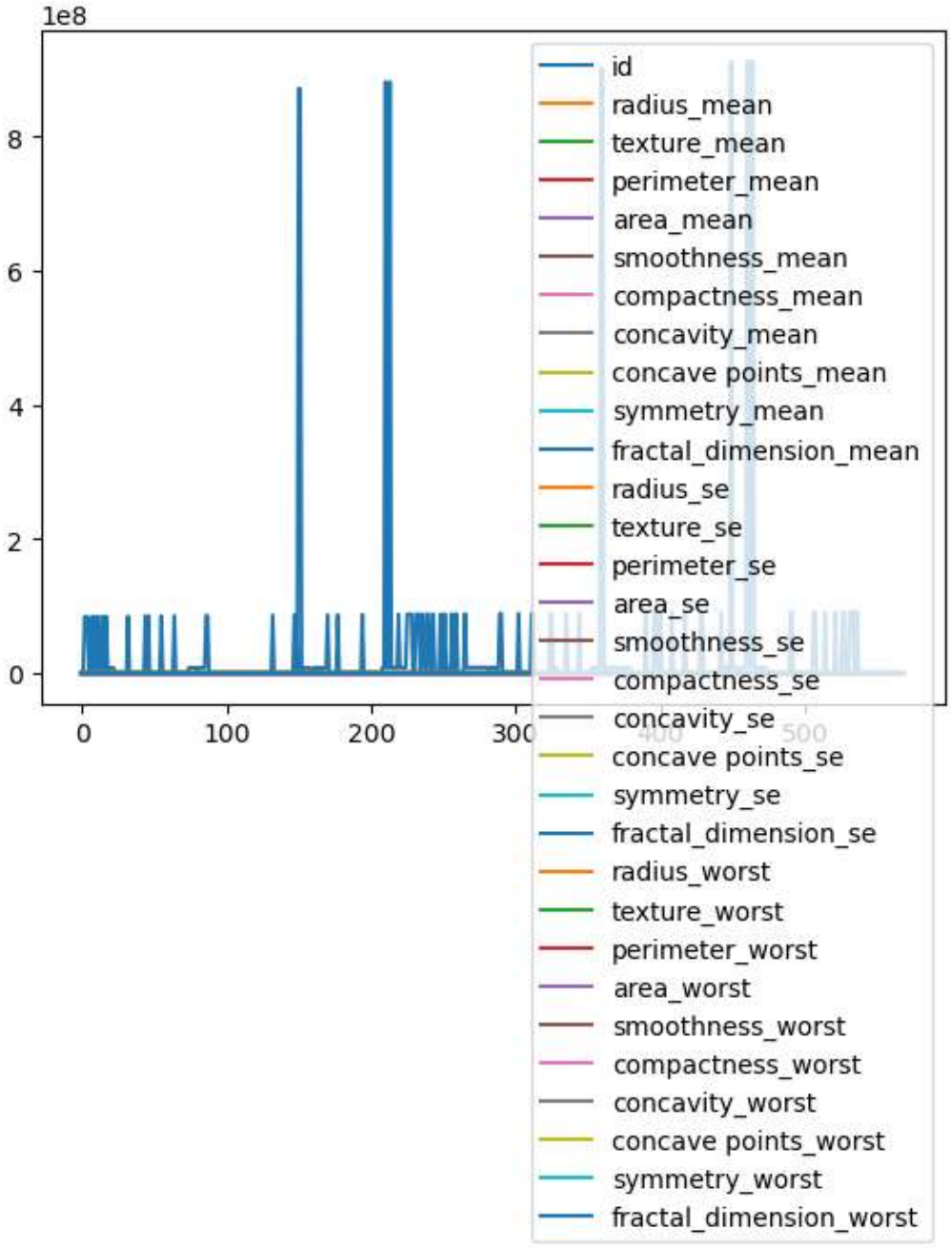
569 rows × 32 columns

In [102]:
```
df1.isnull().sum()
```

Out[102]:
```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
concave points_worst       0
symmetry_worst             0
fractal_dimension_worst    0
dtype: int64
```

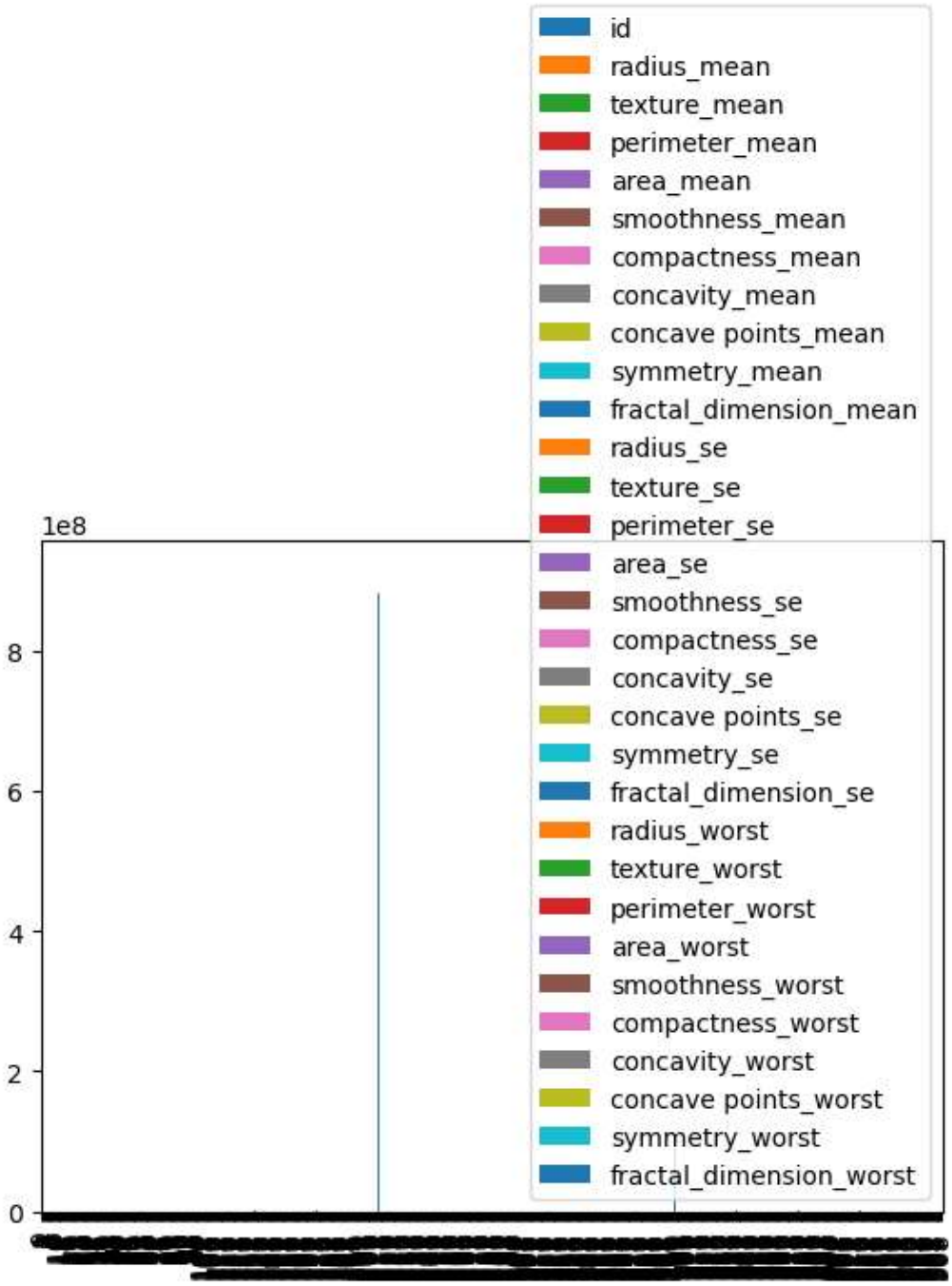
In [103]:
```
df1.plot.line()
```

Out[103]: <Axes: >



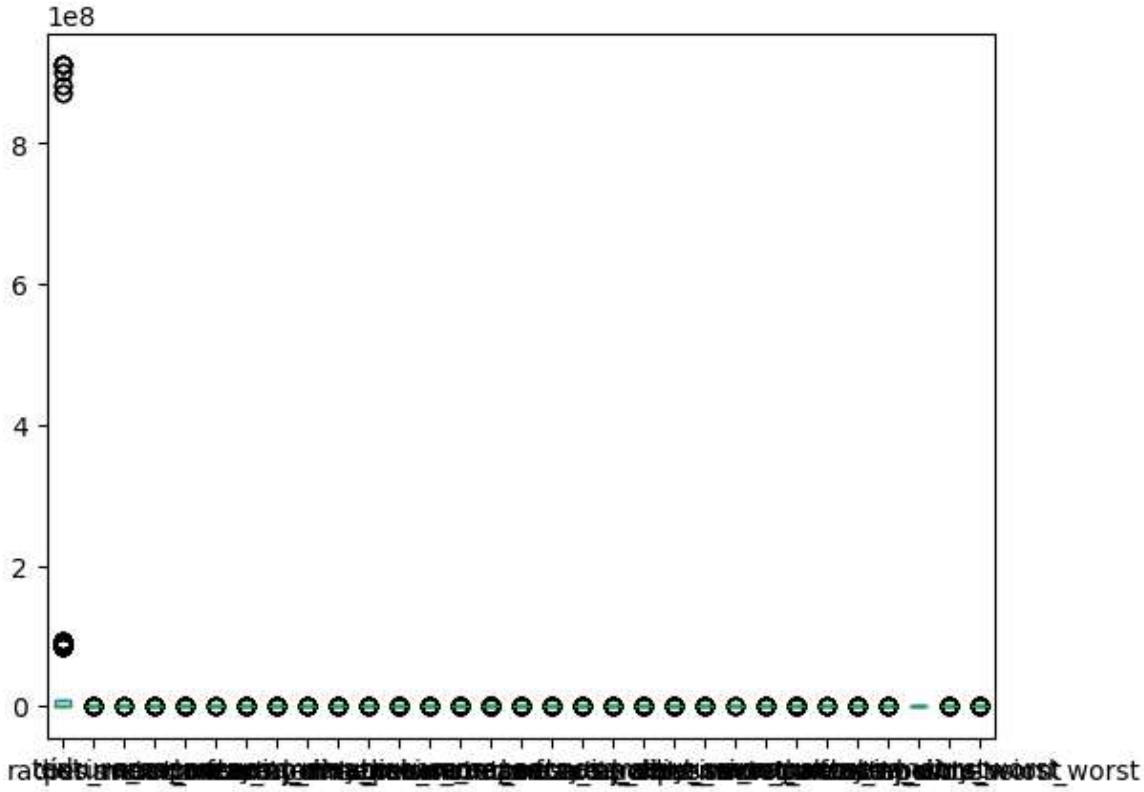
In [104]:
```
df1.plot.bar()
```

Out[104]: <Axes: >



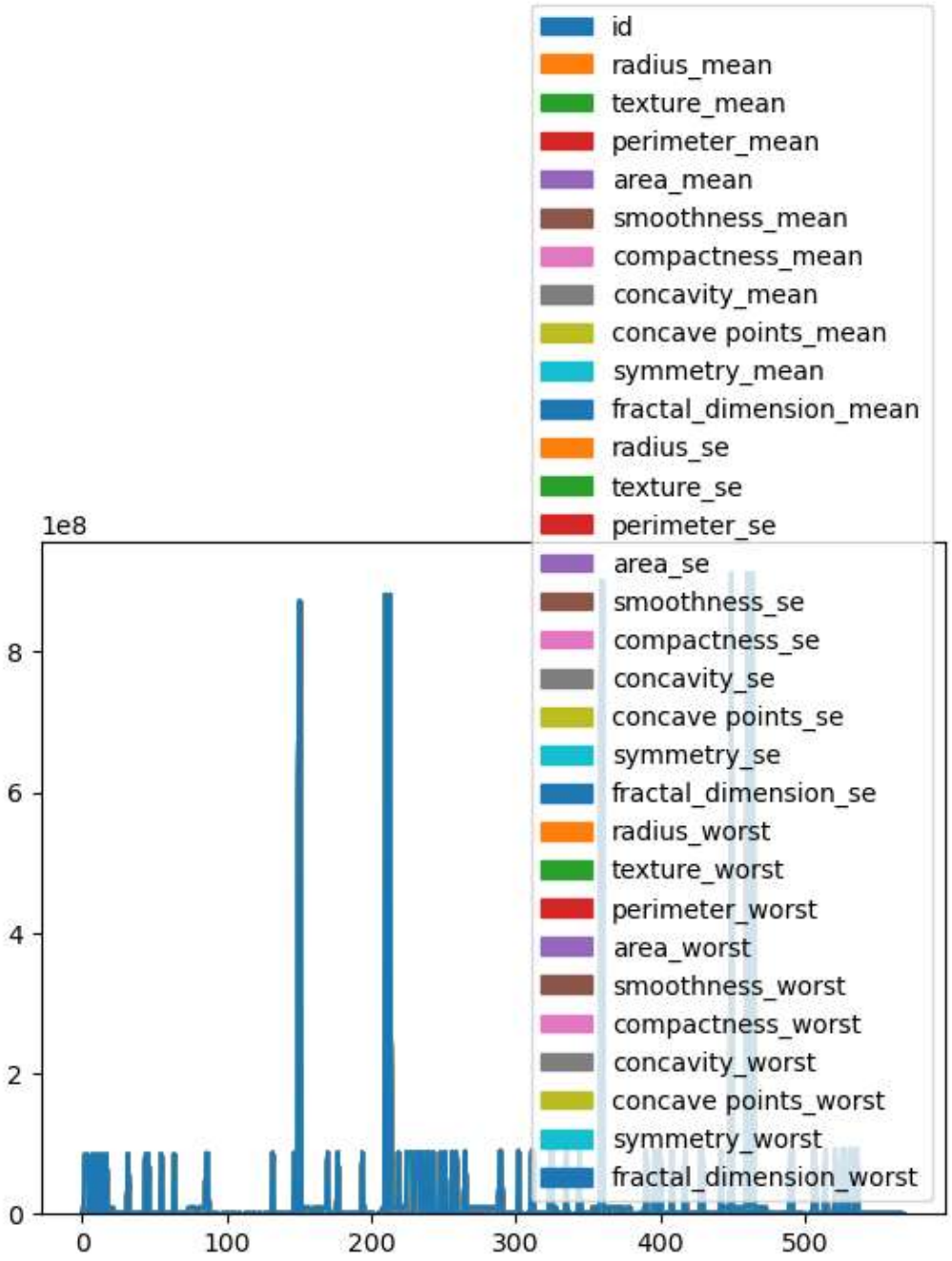
In [125]:
```
df1.plot.box()
```

Out[125]: <Axes: >



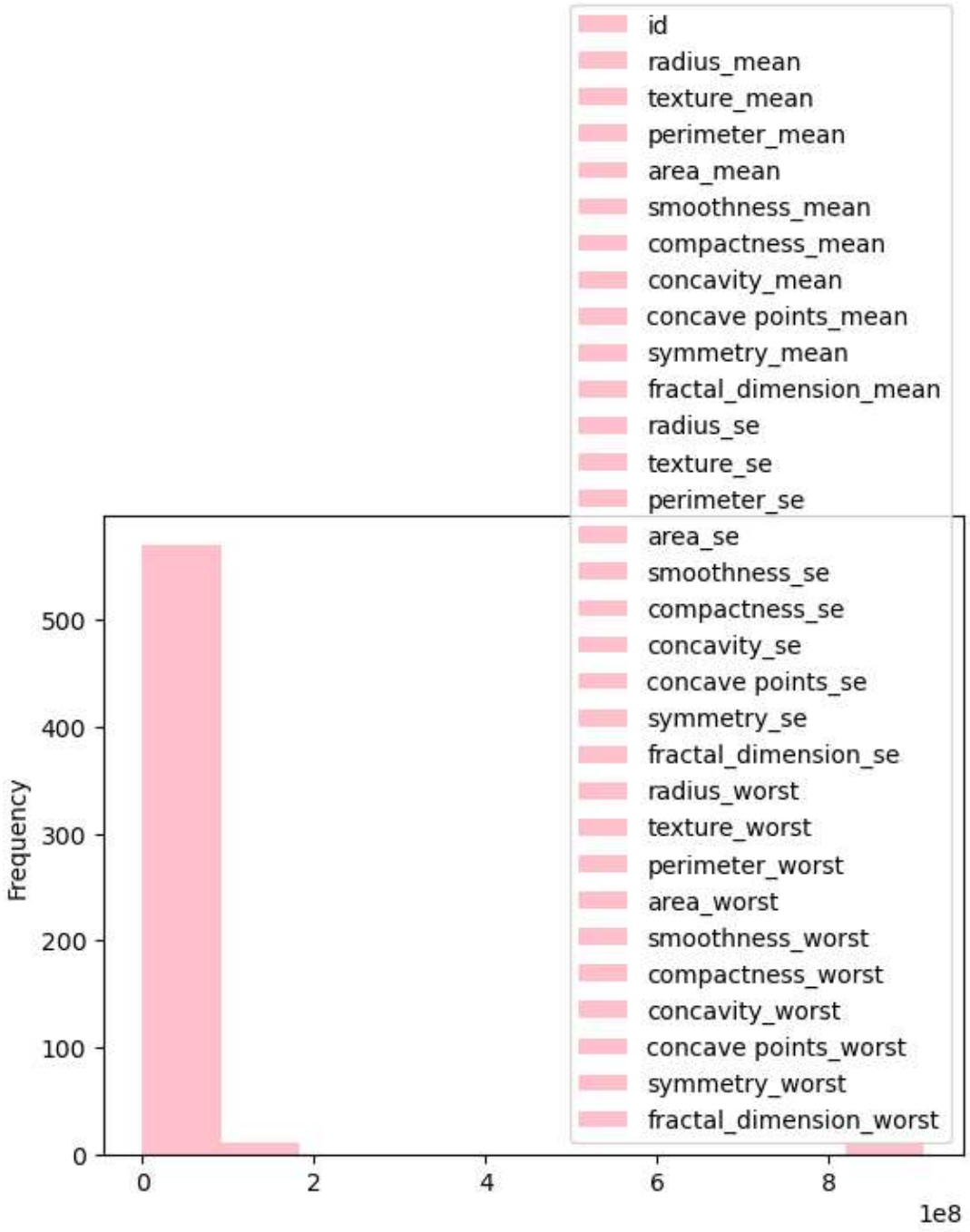
In [105]:
```
df1.plot.area()
```

Out[105]: <Axes: >



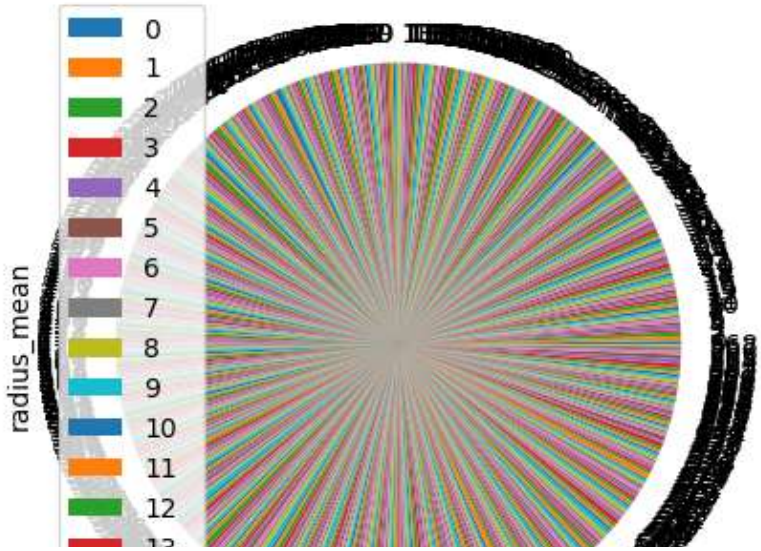
In [172]:
```
df1.plot.hist(color='pink')
```

Out[172]: <Axes: ylabel='Frequency'>



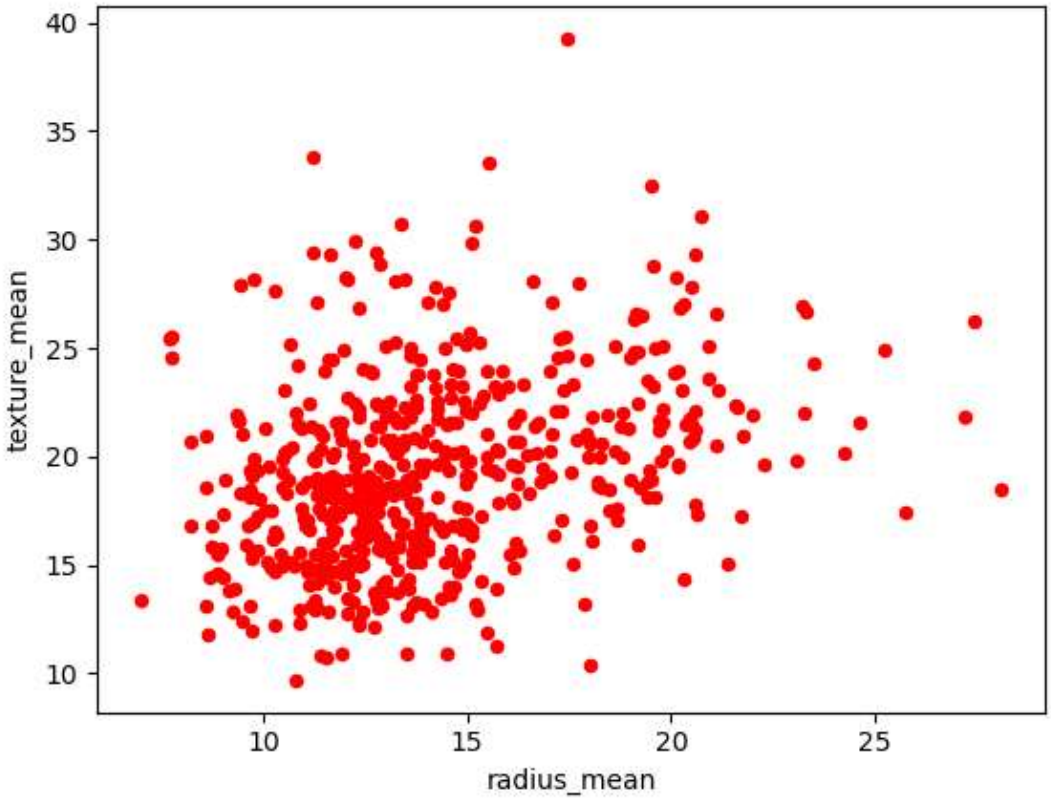
In [110]:
```
df1.plot.pie(y='radius_mean')
```

Out[110]:
```
<Axes: ylabel='radius_mean'>
```



In [171]:
```
df1.plot.scatter(x='radius_mean',y='texture_mean',color='r')
```

Out[171]:
```
<Axes: xlabel='radius_mean', ylabel='texture_mean'>
```



D2

In [112]:
```
df2=pd.read_csv(r"C:\Users\user\Downloads\2_2015.csv")
df2
```

Out[112]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(G
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	
...	...	...	...	...	...	...	...	...	...	
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	

158 rows × 12 columns

In [113]:
```
df2.isnull()
```

Out[113]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(Gover Corru
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
153	False	False	False	False	False	False	False	False	False	
154	False	False	False	False	False	False	False	False	False	
155	False	False	False	False	False	False	False	False	False	
156	False	False	False	False	False	False	False	False	False	
157	False	False	False	False	False	False	False	False	False	

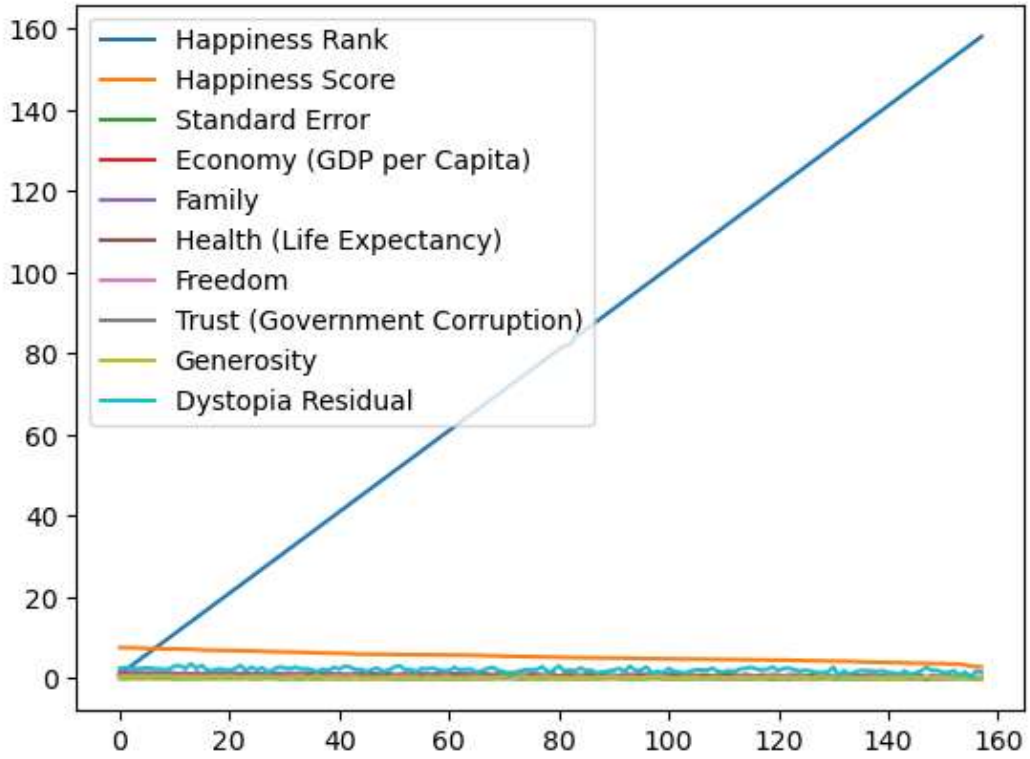
158 rows × 12 columns

In [115]:
```
df2.isnull().sum()
```

Out[115]:
```
Country                          0
Region                           0
Happiness Rank                   0
Happiness Score                  0
Standard Error                   0
Economy (GDP per Capita)         0
Family                           0
Health (Life Expectancy)         0
Freedom                          0
Trust (Government Corruption)    0
Generosity                       0
Dystopia Residual                0
dtype: int64
```

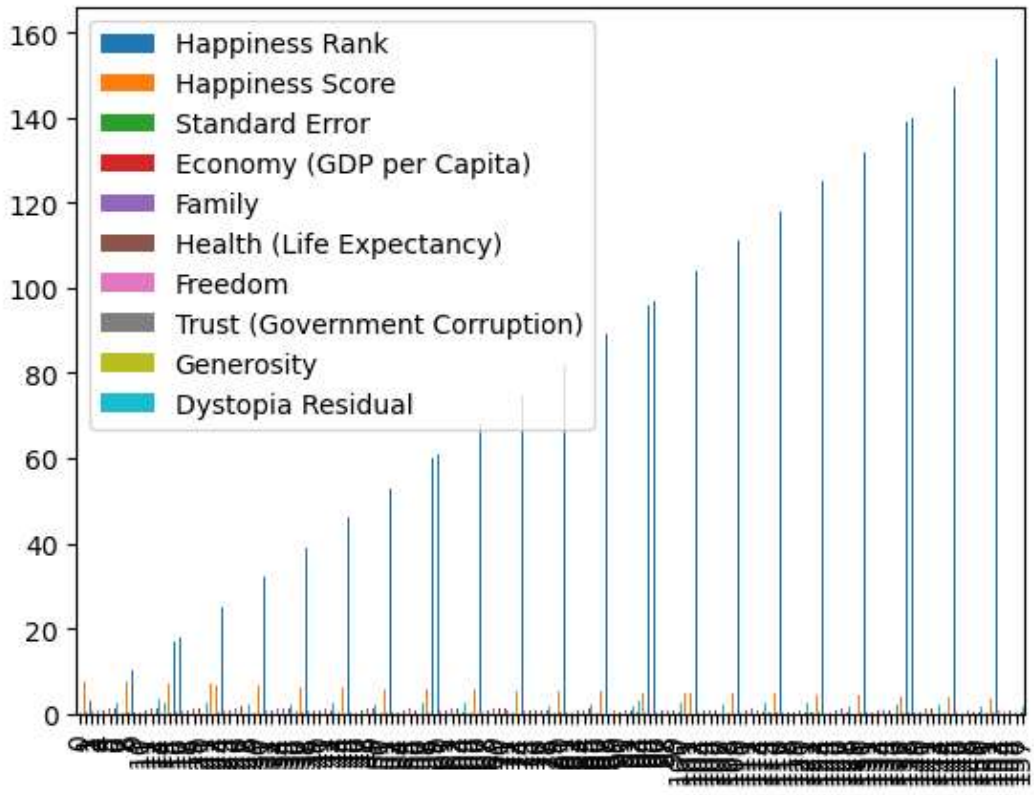
In [116]:
```
df2.plot.line()
```

Out[116]: <Axes: >



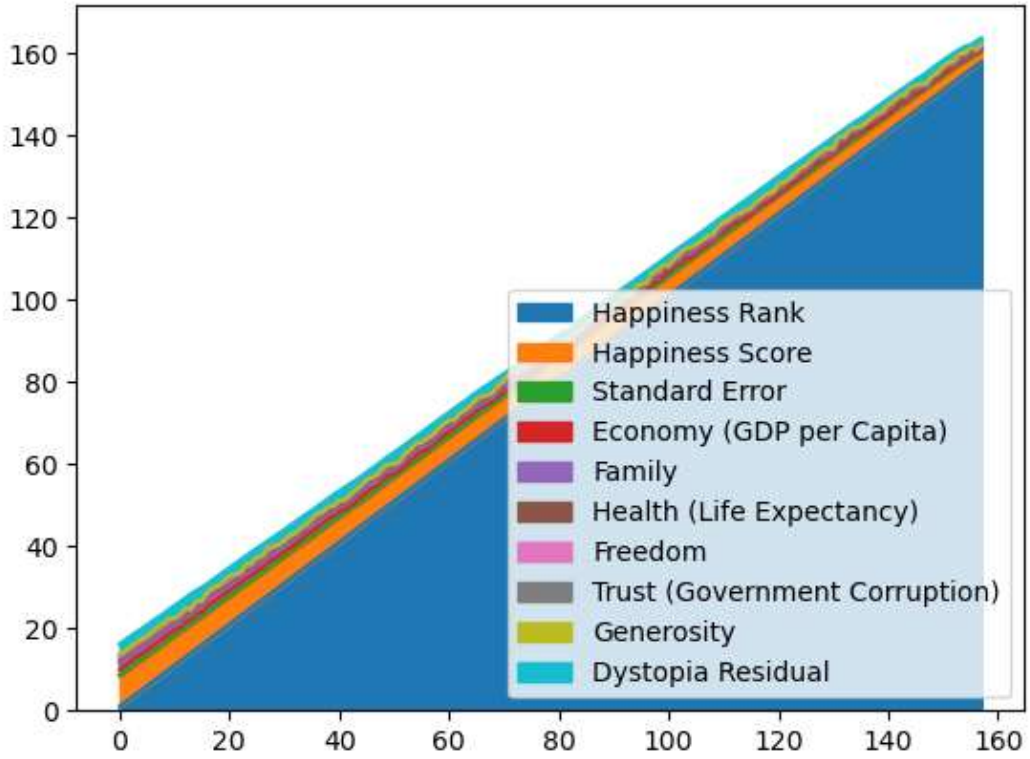
In [117]:
```
df2.plot.bar()
```

Out[117]: <Axes: >



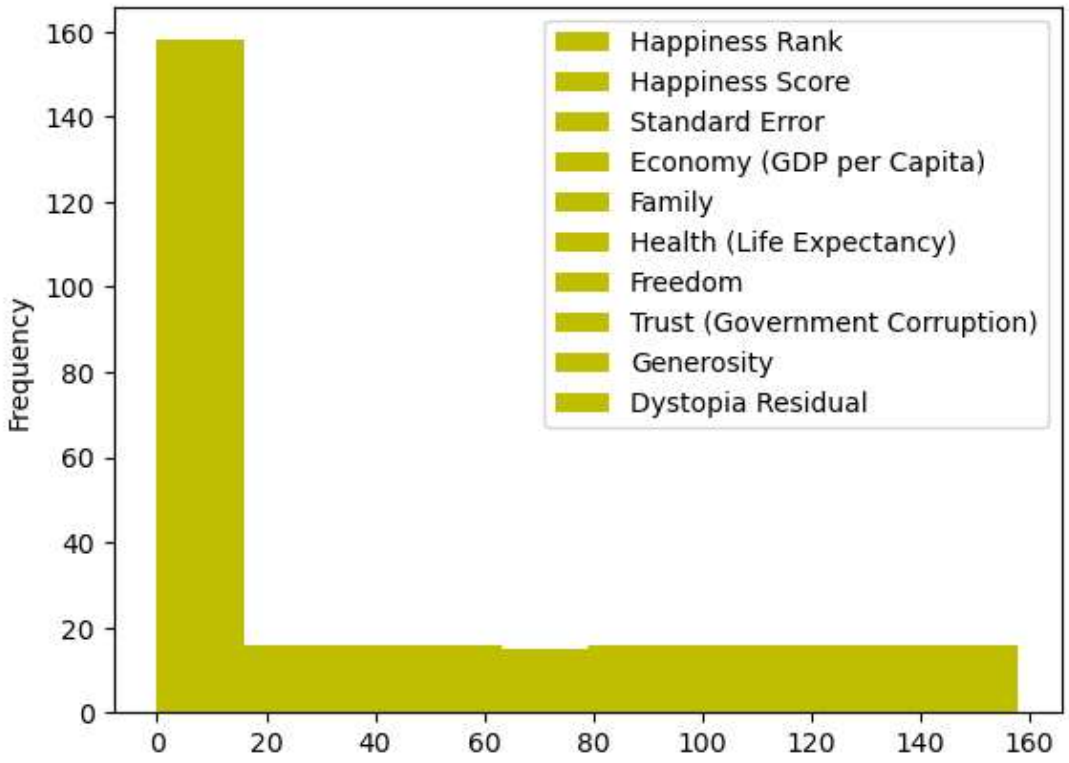
In [118]:
```
df2.plot.area()
```

Out[118]: <Axes: >



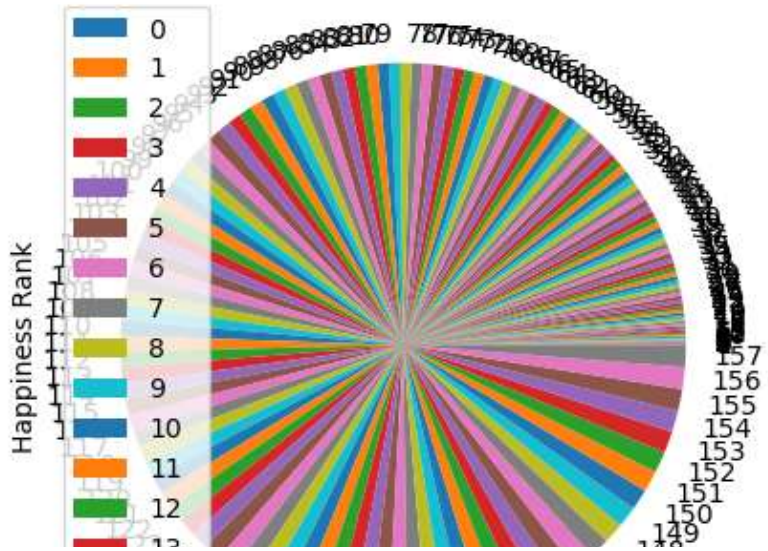
In [170]:
```
df2.plot.hist(color='y')
```

Out[170]: <Axes: ylabel='Frequency'>



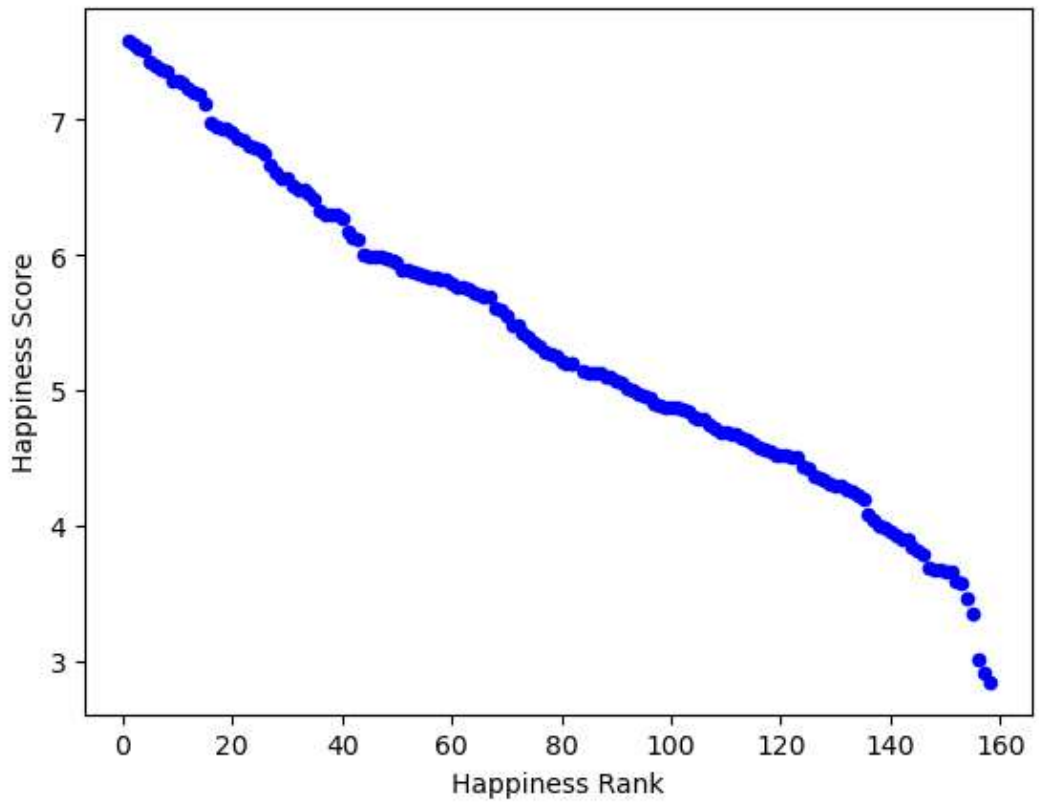
In [122]:
```
df2.plot.pie(y="Happiness Rank")
```

Out[122]:
```
<Axes: ylabel='Happiness Rank'>
```



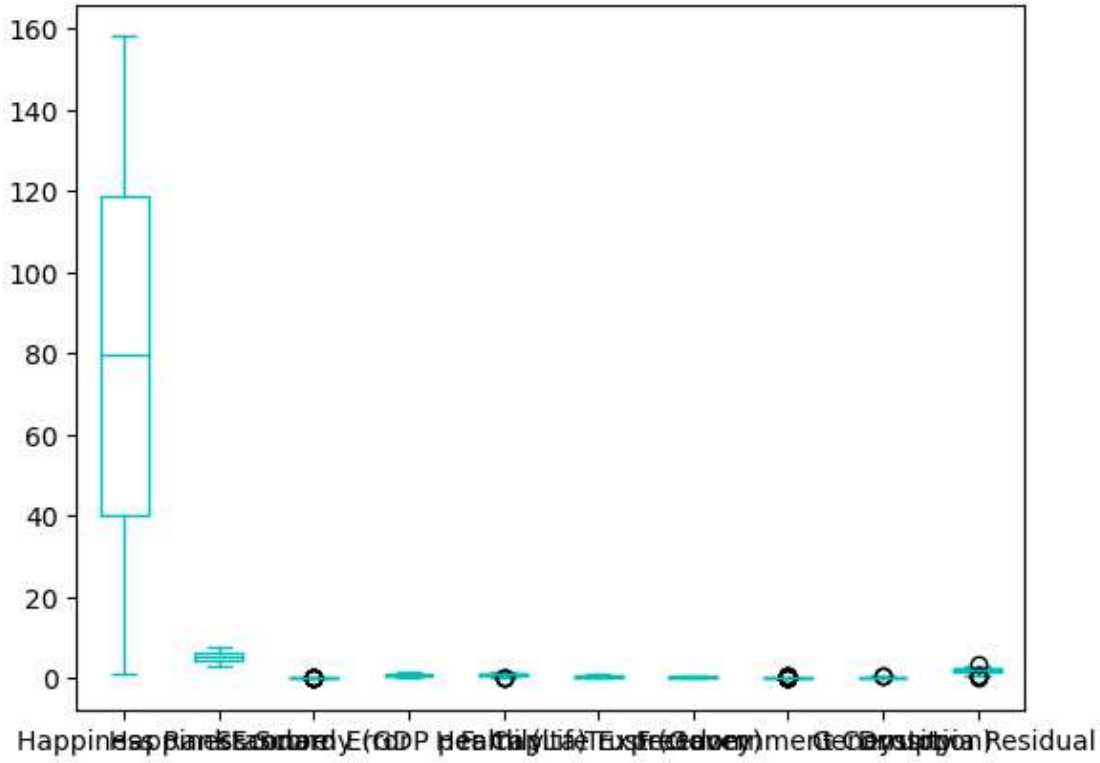
In [169]:
```
df2.plot.scatter(x="Happiness Rank",y="Happiness Score",color='b')
```

Out[169]:
```
<Axes: xlabel='Happiness Rank', ylabel='Happiness Score'>
```



In [167]:
```
df2.plot.box(color='c')
```

Out[167]: <Axes: >



D3

In [126]:
```
df3=pd.read_csv(r"C:\Users\user\Downloads\3_Fitness-1.csv")
df3
```

Out[126]:

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
0	A	5.62%	7.73%	6.16%	75
1	B	4.21%	17.27%	19.21%	160
2	C	9.83%	11.60%	5.17%	101
3	D	2.81%	21.91%	7.88%	127
4	E	25.28%	10.57%	11.82%	179
5	F	8.15%	16.24%	18.47%	167
6	G	18.54%	8.76%	17.49%	171
7	H	25.56%	5.93%	13.79%	170
8	Grand Total	100.00%	100.00%	100.00%	1150

In [128]:
```
df3.isnull()
```

Out[128]:

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False

In [129]:
```
df3.isnull().sum()
```

Out[129]:
```
Row Labels            0
Sum of Jan            0
Sum of Feb            0
Sum of Mar            0
Sum of Total Sales    0
dtype: int64
```

In [137]:
```
df4=df3.drop(8)
```

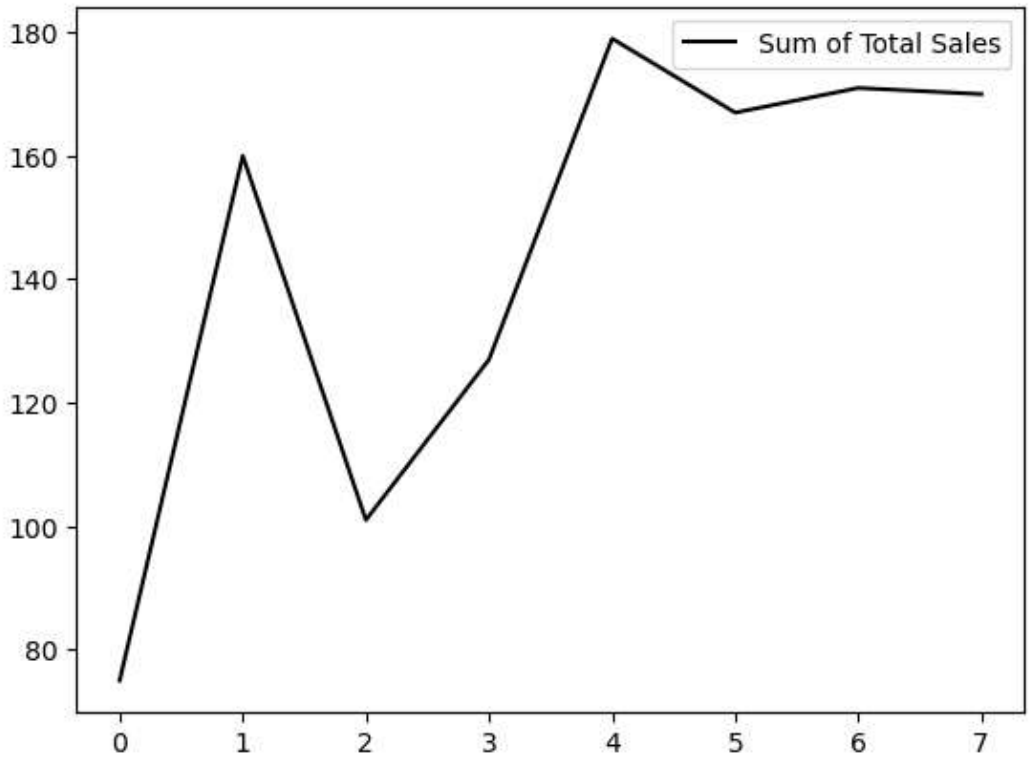
In [138]:
```
df4
```

Out[138]:

	Row Labels	Sum of Jan	Sum of Feb	Sum of Mar	Sum of Total Sales
0	A	5.62%	7.73%	6.16%	75
1	B	4.21%	17.27%	19.21%	160
2	C	9.83%	11.60%	5.17%	101
3	D	2.81%	21.91%	7.88%	127
4	E	25.28%	10.57%	11.82%	179
5	F	8.15%	16.24%	18.47%	167
6	G	18.54%	8.76%	17.49%	171
7	H	25.56%	5.93%	13.79%	170

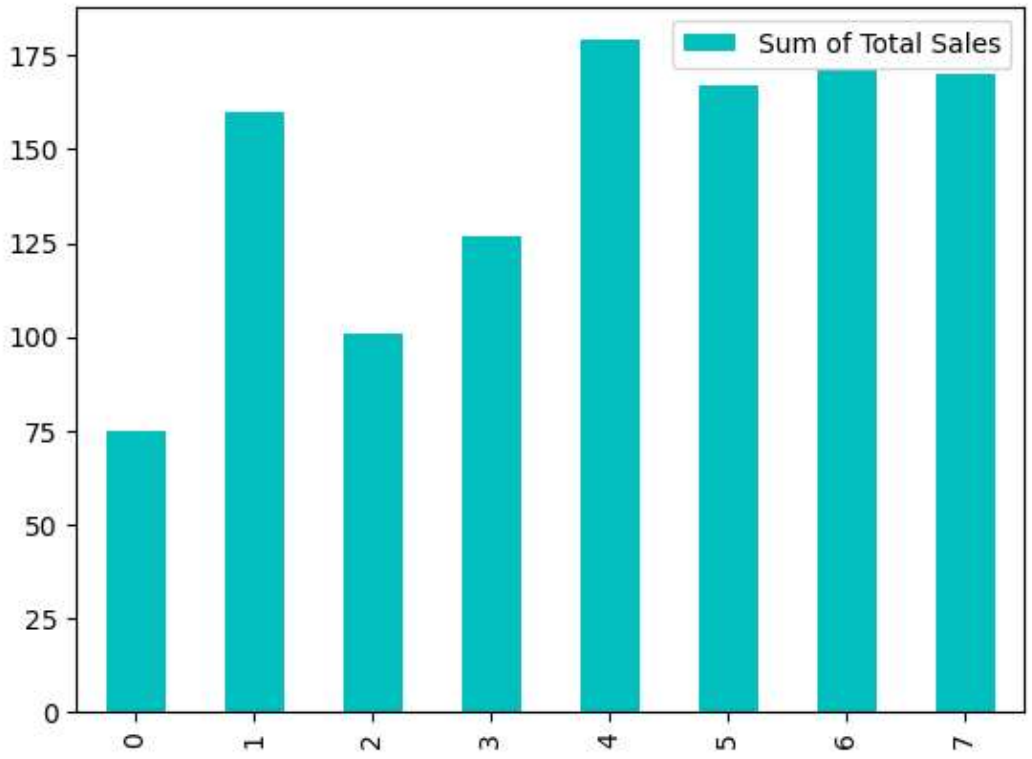
In [166]:
```
df4.plot.line(color='black')
```

Out[166]: <Axes: >



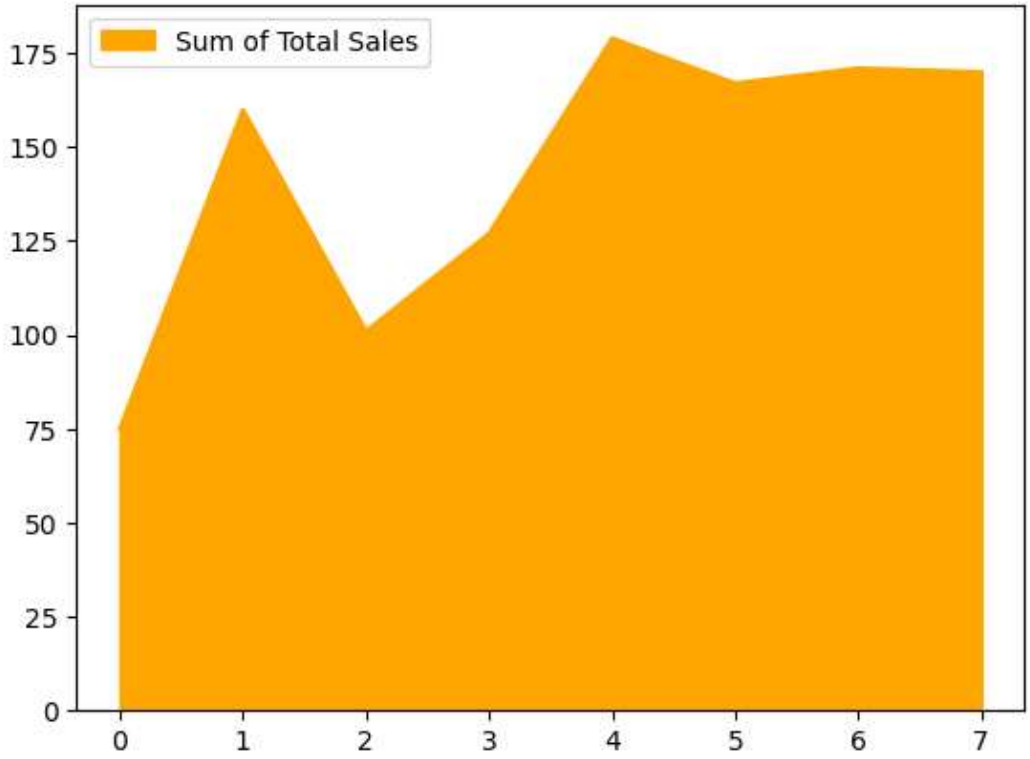
In [165]:
```
df4.plot.bar(color='c')
```

Out[165]: <Axes: >



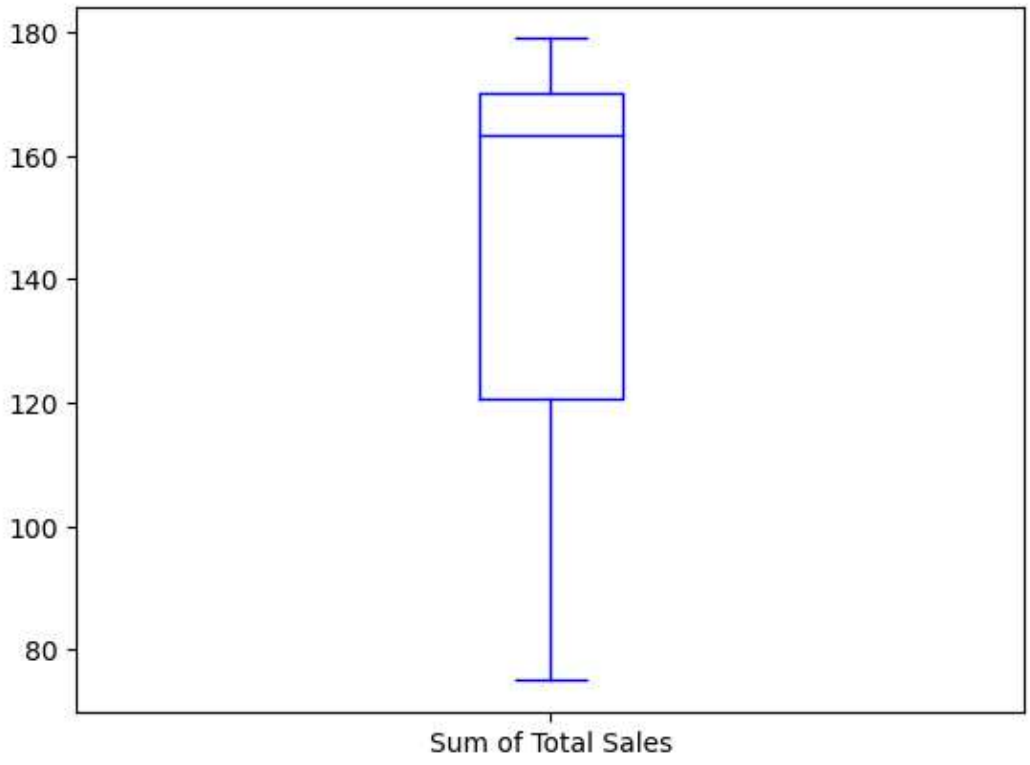
In [164]:
```
df4.plot.area(color='orange')
```

Out[164]: <Axes: >



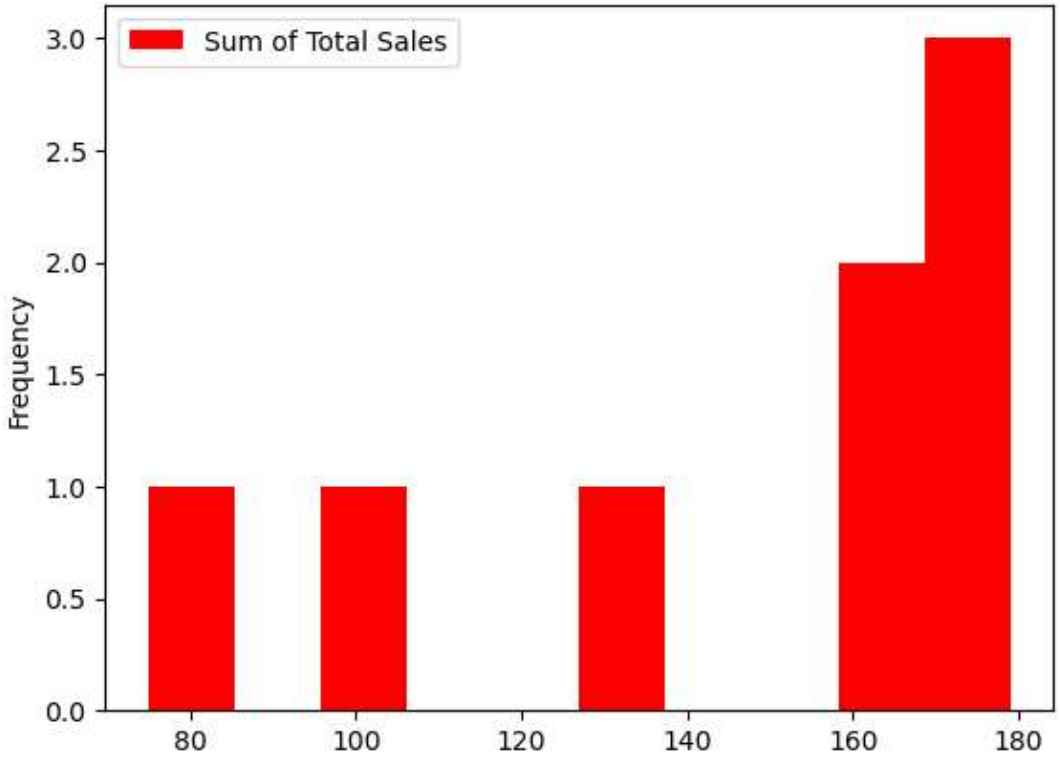
In [162]:
```
df4.plot.box(color='b')
```

Out[162]: <Axes: >



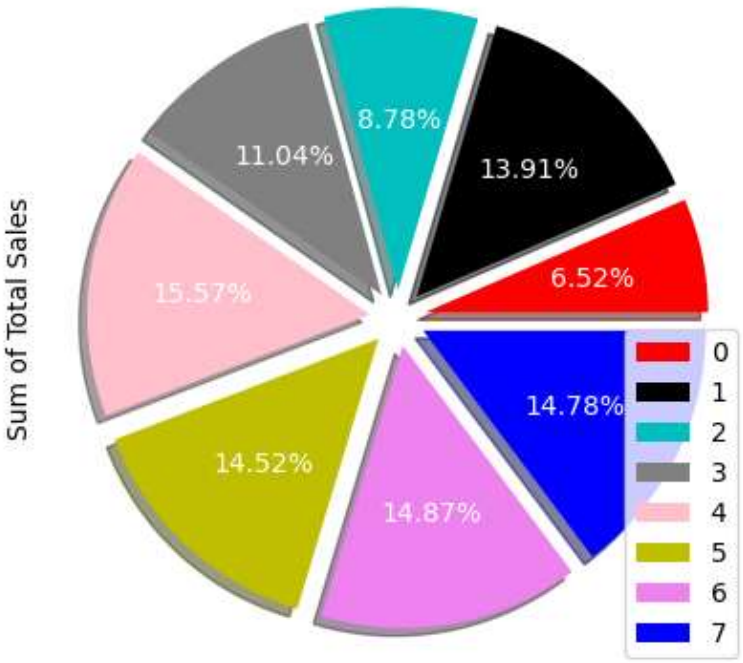
In [161]:
```
df4.plot.hist(color='r')
```

Out[161]: <Axes: ylabel='Frequency'>



In [160]:
```
ck','c','grey','pink','y','violet','blue']
="Sum of Total Sales",autopct='%.2f%%',startangle=1800,shadow=True,explode=(0.1,0.1,0.
```

Out[160]: <Axes: ylabel='Sum of Total Sales'>



In [ ]: