In [5]:
```
!pip install faker
```

```
Requirement already satisfied: faker in c:\users\complab18\anaconda3\lib\si
te-packages (18.4.0)
Requirement already satisfied: python-dateutil>=2.4 in c:\users\complab18\a
naconda3\lib\site-packages (from faker) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\users\complab18\anaconda3\lib
\site-packages (from python-dateutil>=2.4->faker) (1.15.0)
```

In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from faker import Faker
```

In [7]:
```
fake=Faker()
Faker.seed()
names_list=[]
for i in range(100):
    names_list.append(fake.name())
```

In [8]:
```
names_list
```

Out[8]:
```
['Jeffrey Dunn',
 'Jordan Bird',
 'Tammy Rodriguez',
 'Melissa Berry',
 'Nicole Cardenas',
 'Martin Stewart',
 'Adriana Smith',
 'Matthew Cross',
 'Emily Johnson',
 'Juan Adams',
 'Matthew Joyce',
 'Ashlee Anderson',
 'Kevin Bowers',
 'Cody Walker',
 'Emma Mccullough',
 'Sarah Gaines',
 'Stacy Davis',
 'Garrett Martin',
 'Gregory Lee',
 'Valerie Williams',
 'Bobby Martinez',
 'Sarah Solomon',
 'Tricia Allen',
 'Terri Watts',
 'Joseph Weaver',
 'Dakota Garner',
 'Thomas King',
 'Andrea Reyes',
 'Mark Johnson',
 'Mark Elliott',
 'Thomas Neal',
 'James Martinez',
 'Stephanie Mitchell',
 'Brady Krueger',
 'Randy Fisher',
 'Joseph Martin',
 'Suzanne Holmes',
 'Kelsey Johnson',
 'Mark Martin',
 'Ashley Vang',
 'Charles Delgado',
 'Joseph Stewart',
 'Amy Lee',
 'Mark West',
 'William Hill',
 'Robert Wong',
 'Theodore Santos',
 'Marcus Norris',
 'Melvin Cox',
 'Adrian Fisher',
 'Courtney Barrett',
 'Charles Mitchell',
 'Amanda Ward',
 'Sean Dunn',
 'Larry Hernandez',
 'Melissa Wells',
 'Derrick Warner',
 'Jasmine Thomas',
 'Brian Johnson',
 'Nicole Walker',
 'Susan Thompson',
 'Anthony Williams',
 'Debra Allen',
 'Vicki Murphy',
 'Cindy Brown',
 'Donna Khan',
 'Roger Salinas',
 'Morgan Brown',
 'Christopher Peters',
 'Laura Williams',
 'Kevin Holt',
 'Douglas Mills',
 'Mary Scott',
 'Jack Taylor',
 'Jennifer Long',
 'Jenna Palmer',
 'Joshua Rodriguez',
 'Vanessa Ramirez',
 'Randall Brooks',
 'Charles Davis',
 'Ryan Franklin',
 'Amber Whitehead',
 'Jennifer Salazar',
 'Kimberly Bryan',
 'Amy Williams',
 'Joseph Harris',
 'Jessica Mcdaniel',
 'David Sutton',
 'Mario Wright',
 'Ryan Reynolds',
 'Kaylee Cox',
 'Jennifer Rodriguez',
 'Holly Scott',
 'Rodney Johnson',
 'Dr. William Gonzalez',
 'Nicole Orozco',
 'Brooke Reed',
 'Mark Ortiz',
 'Brendan Warren',
 'Andrea Diaz']
```

In [11]:
```
np.random.seed(7)
salaries=[]
for i in range(100):
    salary=np.random.randint(1000,2500)
    salaries.append(salary)
```

In [12]:
```
salaries
```

Out[12]:
```
[1175,
 2220,
 1537,
 1502,
 1211,
 1919,
 2372,
 2209,
 2422,
 1535,
 1345,
 1366,
 1554,
 1730,
 1904,
 2191,
 2092,
 2456,
 1391,
 1940,
 2099,
 1823,
 1250,
 2030,
 2468,
 2068,
 2349,
 2176,
 1183,
 1949,
 2136,
 1763,
 2213,
 1290,
 1312,
 1201,
 2486,
 1550,
 1772,
 1494,
 2161,
 2219,
 2096,
 1944,
 1257,
 1400,
 2398,
 2373,
 1940,
 1604,
 1764,
 1279,
 1745,
 1803,
 2472,
 2341,
 2415,
 2175,
 1092,
 1759,
 2248,
 1356,
 1931,
 1481,
 1579,
 2243,
 1481,
 2043,
 1923,
 1787,
 1033,
 2147,
 2310,
 1741,
 1989,
 2439,
 2154,
 2107,
 1402,
 1021,
 1203,
 1047,
 1784,
 1524,
 1349,
 1107,
 1393,
 1844,
 1622,
 1654,
 1636,
 1276,
 1309,
 1827,
 1035,
 1260,
 1456,
 2498,
 2007,
 2239]
```

In [14]:
```
df=pd.DataFrame({"person":names_list,"salary":salaries})
df.head(3)
```

Out[14]:

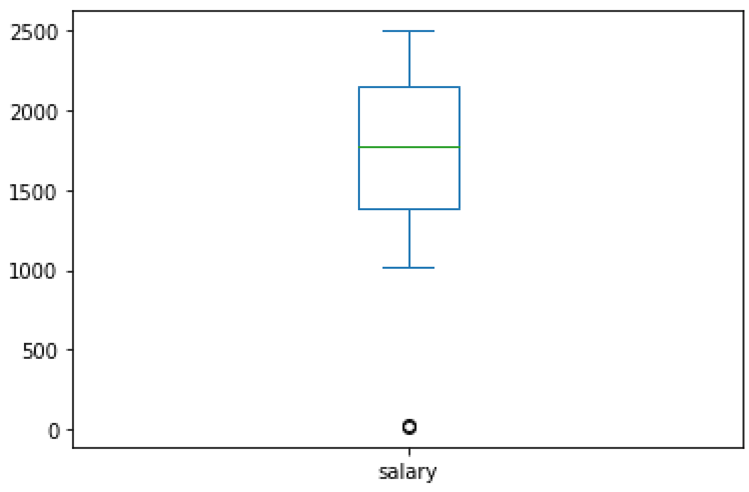
	person	salary
0	Jeffrey Dunn	1175
1	Jordan Bird	2220
2	Tammy Rodriguez	1537

In [16]:
```
df.at[16,"salary"]=23
df.at[65,"salary"]=17
print(df.loc[16])
print(df.loc[65])
```

```
person    Stacy Davis
salary             23
Name: 16, dtype: object
person    Donna Khan
salary            17
Name: 65, dtype: object
```

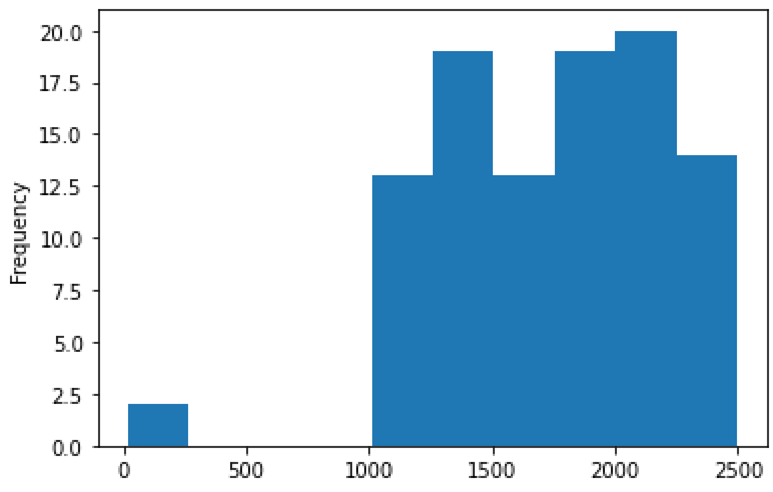
In [17]:
```
df["salary"].plot(kind="box")
```

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



In [18]:
```
df["salary"].plot(kind="hist")
```

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



In [19]:
```
raw_salary=df["salary"].values
raw_salary
```

Out[19]:
```
array([1175, 2220, 1537, 1502, 1211, 1919, 2372, 2209, 2422, 1535, 1345,
       1366, 1554, 1730, 1904, 2191,   23, 2456, 1391, 1940, 2099, 1823,
       1250, 2030, 2468, 2068, 2349, 2176, 1183, 1949, 2136, 1763, 2213,
       1290, 1312, 1201, 2486, 1550, 1772, 1494, 2161, 2219, 2096, 1944,
       1257, 1400, 2398, 2373, 1940, 1604, 1764, 1279, 1745, 1803, 2472,
       2341, 2415, 2175, 1092, 1759, 2248, 1356, 1931, 1481, 1579,   17,
       1481, 2043, 1923, 1787, 1033, 2147, 2310, 1741, 1989, 2439, 2154,
       2107, 1402, 1021, 1203, 1047, 1784, 1524, 1349, 1107, 1393, 1844,
       1622, 1654, 1636, 1276, 1309, 1827, 1035, 1260, 1456, 2498, 2007,
       2239], dtype=int64)
```

In [20]:
```
raw_salary=raw_salary.reshape(-1,1)
raw_salary
```

```
Out[20]: array([[1175],
               [2220],
               [1537],
               [1502],
               [1211],
               [1919],
               [2372],
               [2209],
               [2422],
               [1535],
               [1345],
               [1366],
               [1554],
               [1730],
               [1904],
               [2191],
               [  23],
               [2456],
               [1391],
               [1940],
               [2099],
               [1823],
               [1250],
               [2030],
               [2468],
               [2068],
               [2349],
               [2176],
               [1183],
               [1949],
               [2136],
               [1763],
               [2213],
               [1290],
               [1312],
               [1201],
               [2486],
               [1550],
               [1772],
               [1494],
               [2161],
               [2219],
               [2096],
               [1944],
               [1257],
               [1400],
               [2398],
               [2373],
               [1940],
               [1604],
               [1764],
               [1279],
               [1745],
               [1803],
               [2472],
               [2341],
               [2415],
               [2175],
               [1092],
               [1759],
               [2248],
               [1356],
               [1931],
               [1481],
               [1579],
               [  17],
               [1481],
               [2043],
               [1923],
               [1787],
               [1033],
               [2147],
               [2310],
               [1741],
               [1989],
               [2439],
               [2154],
               [2107],
               [1402],
               [1021],
               [1203],
               [1047],
               [1784],
               [1524],
               [1349],
               [1107],
               [1393],
               [1844],
               [1622],
               [1654],
               [1636],
               [1276],
               [1309],
               [1827],
               [1035],
               [1260],
               [1456],
               [2498],
               [2007],
               [2239]], dtype=int64)
```

In [21]:
```
raw_salary=raw_salary.astype("float64")
raw_salary
```

Out[21]:
```
array([[1175.],
       [2220.],
       [1537.],
       [1502.],
       [1211.],
       [1919.],
       [2372.],
       [2209.],
       [2422.],
       [1535.],
       [1345.],
       [1366.],
       [1554.],
       [1730.],
       [1904.],
       [2191.],
       [  23.],
       [2456.],
       [1391.],
       [1940.],
       [2099.],
       [1823.],
       [1250.],
       [2030.],
       [2468.],
       [2068.],
       [2349.],
       [2176.],
       [1183.],
       [1949.],
       [2136.],
       [1763.],
       [2213.],
       [1290.],
       [1312.],
       [1201.],
       [2486.],
       [1550.],
       [1772.],
       [1494.],
       [2161.],
       [2219.],
       [2096.],
       [1944.],
       [1257.],
       [1400.],
       [2398.],
       [2373.],
       [1940.],
       [1604.],
       [1764.],
       [1279.],
       [1745.],
       [1803.],
       [2472.],
       [2341.],
       [2415.],
       [2175.],
       [1092.],
       [1759.],
       [2248.],
       [1356.],
       [1931.],
       [1481.],
       [1579.],
       [  17.],
       [1481.],
       [2043.],
       [1923.],
       [1787.],
       [1033.],
       [2147.],
       [2310.],
       [1741.],
       [1989.],
       [2439.],
       [2154.],
       [2107.],
       [1402.],
       [1021.],
       [1203.],
       [1047.],
       [1784.],
       [1524.],
       [1349.],
       [1107.],
       [1393.],
       [1844.],
       [1622.],
       [1654.],
       [1636.],
       [1276.],
       [1309.],
       [1827.],
       [1035.],
       [1260.],
       [1456.],
       [2498.],
       [2007.],
       [2239.]])
```

In [22]:
```
from sklearn.cluster import KMeans
```

In [23]:
```
cm=KMeans(n_clusters=4)
cm
```

Out[23]:
```
KMeans(n_clusters=4)
```

In [24]:
```
cm.fit(raw_salary)
```

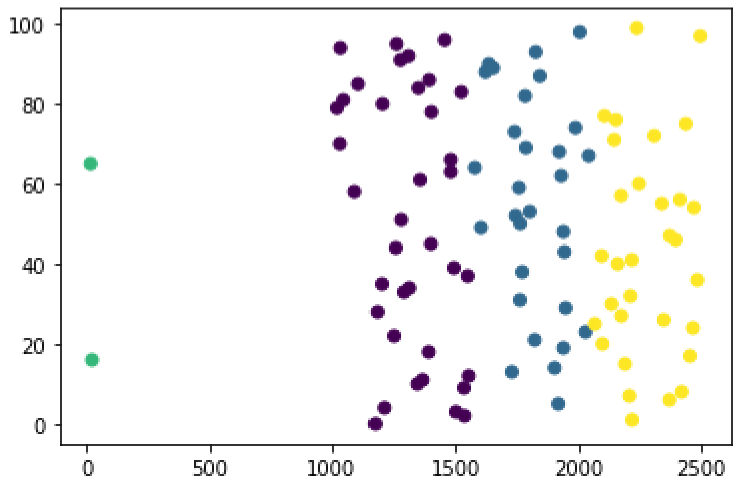
Out[24]:
```
KMeans(n_clusters=4)
```

In [25]:
```
cm_labels=cm.labels_
cm_labels
```

Out[25]:
```
array([0, 3, 0, 0, 0, 1, 3, 3, 3, 0, 0, 0, 0, 1, 1, 3, 2, 3, 0, 1, 3, 1,
       0, 1, 3, 3, 3, 3, 0, 1, 3, 1, 3, 0, 0, 0, 3, 0, 1, 0, 3, 3, 3, 1,
       0, 0, 3, 3, 1, 1, 1, 0, 1, 1, 3, 3, 3, 3, 0, 1, 3, 0, 1, 0, 1, 2,
       0, 1, 1, 1, 0, 3, 3, 1, 1, 3, 3, 3, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 0, 1, 0, 0, 0, 3, 1, 3])
```

In [26]:
```
plt.scatter(raw_salary,np.arange(0,100),c=cm_labels)
```

Out[26]:
```
<matplotlib.collections.PathCollection at 0x2dca7783d90>
```



In [27]:
```
df["class"]=0
df["class"]
```

Out[27]:
```
0     0
1     0
2     0
3     0
4     0
     ..
95    0
96    0
97    0
98    0
99    0
Name: class, Length: 100, dtype: int64
```

In [29]:
```
df.at[16,"class"]=1
df.at[65,"class"]=1
```

In [30]:
```
print(df.at[16,"class"])
```

```
1
```

In [31]:
```
df["class"].values
```

Out[31]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

In [32]:
```
!pip install pyod
```

```
Requirement already satisfied: pyod in c:\users\complab18\anaconda3\lib\sit
e-packages (1.0.9)
Requirement already satisfied: numba>=0.51 in c:\users\complab18\anaconda3
\lib\site-packages (from pyod) (0.53.1)
Requirement already satisfied: joblib in c:\users\complab18\anaconda3\lib\s
ite-packages (from pyod) (1.0.1)
Requirement already satisfied: scipy>=1.5.1 in c:\users\complab18\anaconda3
\lib\site-packages (from pyod) (1.6.2)
Requirement already satisfied: scikit-learn>=0.20.0 in c:\users\complab18\a
naconda3\lib\site-packages (from pyod) (0.24.1)
Requirement already satisfied: numpy>=1.19 in c:\users\complab18\anaconda3
\lib\site-packages (from pyod) (1.22.4)
Requirement already satisfied: matplotlib in c:\users\complab18\anaconda3\l
ib\site-packages (from pyod) (3.6.2)
Requirement already satisfied: six in c:\users\complab18\anaconda3\lib\site
-packages (from pyod) (1.15.0)
Requirement already satisfied: setuptools in c:\users\complab18\anaconda3\l
ib\site-packages (from numba>=0.51->pyod) (52.0.0.post20210125)
Requirement already satisfied: llvmlite<0.37,>=0.36.0rc1 in c:\users\compla
b18\anaconda3\lib\site-packages (from numba>=0.51->pyod) (0.36.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\complab18\a
naconda3\lib\site-packages (from scikit-learn>=0.20.0->pyod) (3.1.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\complab18\anac
onda3\lib\site-packages (from matplotlib->pyod) (4.38.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\complab18\anaconda
3\lib\site-packages (from matplotlib->pyod) (8.2.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\complab18\anaco
nda3\lib\site-packages (from matplotlib->pyod) (2.4.7)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\complab18\anaco
nda3\lib\site-packages (from matplotlib->pyod) (1.0.6)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\complab18\anac
onda3\lib\site-packages (from matplotlib->pyod) (1.3.1)
Requirement already satisfied: packaging>=20.0 in c:\users\complab18\anacon
da3\lib\site-packages (from matplotlib->pyod) (21.3)
Requirement already satisfied: cycler>=0.10 in c:\users\complab18\anaconda3
\lib\site-packages (from matplotlib->pyod) (0.10.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\complab18\a
naconda3\lib\site-packages (from matplotlib->pyod) (2.8.1)
```

In [33]:
```
from pyod.models.knn import KNN
```

In [34]:
```
X=df["salary"].values.reshape(-1,1)
X
```

```
Out[34]: array([[1175],
               [2220],
               [1537],
               [1502],
               [1211],
               [1919],
               [2372],
               [2209],
               [2422],
               [1535],
               [1345],
               [1366],
               [1554],
               [1730],
               [1904],
               [2191],
               [  23],
               [2456],
               [1391],
               [1940],
               [2099],
               [1823],
               [1250],
               [2030],
               [2468],
               [2068],
               [2349],
               [2176],
               [1183],
               [1949],
               [2136],
               [1763],
               [2213],
               [1290],
               [1312],
               [1201],
               [2486],
               [1550],
               [1772],
               [1494],
               [2161],
               [2219],
               [2096],
               [1944],
               [1257],
               [1400],
               [2398],
               [2373],
               [1940],
               [1604],
               [1764],
               [1279],
               [1745],
               [1803],
               [2472],
               [2341],
               [2415],
               [2175],
               [1092],
               [1759],
               [2248],
               [1356],
               [1931],
               [1481],
               [1579],
               [  17],
               [1481],
               [2043],
               [1923],
               [1787],
               [1033],
               [2147],
               [2310],
               [1741],
               [1989],
               [2439],
               [2154],
               [2107],
               [1402],
               [1021],
               [1203],
               [1047],
               [1784],
               [1524],
               [1349],
               [1107],
               [1393],
               [1844],
               [1622],
               [1654],
               [1636],
               [1276],
               [1309],
               [1827],
               [1035],
               [1260],
               [1456],
               [2498],
               [2007],
               [2239]], dtype=int64)
```

In [35]:
```
y=df["class"].values
y
```

Out[35]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

In [36]:
```
clf=KNN(contamination=0.02,n_neighbors=5)
clf.fit(X)
```

Out[36]:
```
KNN(algorithm='auto', contamination=0.02, leaf_size=30, method='largest',
  metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=5, p=2,
  radius=1.0)
```

```
In [37]: y_train_scores=clf.decision_scores_
         y_train_scores
```

```
Out[37]: array([  68.,   28.,   35.,   33.,   39.,   21.,   43.,   30.,   46.,
                  33.,   36.,   27.,   30.,   34.,   36.,   28., 1024.,   34.,
                  35.,   17.,   48.,   39.,   39.,   66.,   30.,   39.,   49.,
                  29.,   67.,   26.,   37.,   21.,   26.,   30.,   36.,   49.,
                  47.,   29.,   15.,   38.,   25.,   28.,   51.,   21.,   33.,
                  44.,   41.,   42.,   17.,   50.,   20.,   29.,   19.,   31.,
                  33.,   57.,   42.,   28.,   71.,   18.,   39.,   37.,   13.,
                  43.,   43., 1030.,   43.,   54.,   19.,   24.,   74.,   29.,
                  63.,   23.,   49.,   33.,   22.,   40.,   46.,   86.,   47.,
                  60.,   21.,   30.,   40.,   74.,   37.,   60.,   68.,   76.,
                  82.,   26.,   36.,   43.,   72.,   30.,   54.,   59.,   61.,
                  30.])
```

```
In [38]: X_test=np.array([[35.]])
         X_test
```

```
Out[38]: array([[35.]])
```

```
In [39]: clf.predict(X_test)
```

```
Out[39]: array([1])
```

```
In [40]: X_test2=np.array([[1005.]])
         clf.predict(X_test2)
```

```
Out[40]: array([0])
```

```
In [ ]:
```

```
In [ ]:
```