In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_csv("teleCust1000t.csv")
df.head()
```

Out[2]:

	region	tenure	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	13	44	1	9	64.0	4	5	0.0	0	2	1
1	3	11	33	1	7	136.0	5	5	0.0	0	6	4
2	3	68	52	1	24	116.0	1	29	0.0	1	2	3
3	2	33	33	0	12	33.0	2	0	0.0	1	1	1
4	2	23	30	1	9	30.0	1	2	0.0	0	4	3

In [4]:

```
df["custcat"].value_counts()
```

Out[4]:

```
3    281
1    266
4    236
2    217
Name: custcat, dtype: int64
```

In [5]:

```
df.shape
```

Out[5]:

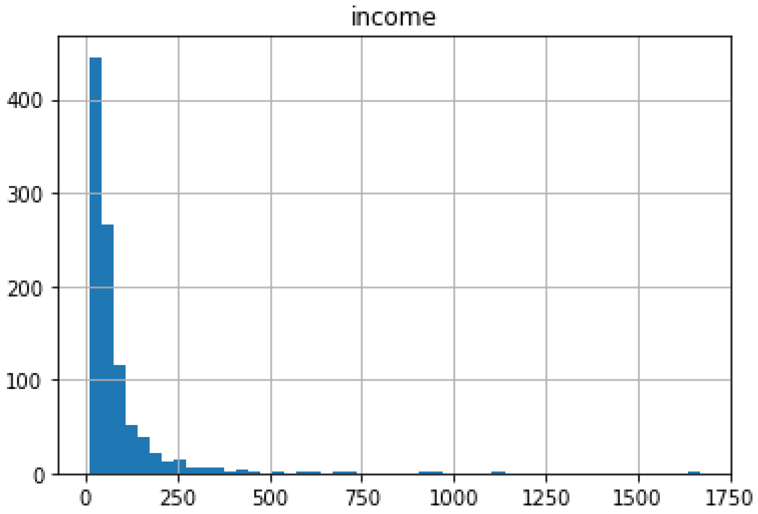
```
(1000, 12)
```

In [13]:

```
df.hist("income",bins=50)
```

Out[13]:

```
array([[<AxesSubplot:title={'center':'income'}>]], dtype=object)
```



In [19]:

```
(df["income"].value_counts())
```

Out[19]:

```
25.0     24
26.0     22
41.0     22
33.0     20
46.0     19
         ..
214.0     1
508.0     1
130.0     1
928.0     1
591.0     1
Name: income, Length: 218, dtype: int64
```

In [21]:

```
x = df.drop("custcat",axis=1)
x.head()
```

Out[21]:

	region	tenure	age	marital	address	income	ed	employ	retire	gender	reside
0	2	13	44	1	9	64.0	4	5	0.0	0	2
1	3	11	33	1	7	136.0	5	5	0.0	0	6
2	3	68	52	1	24	116.0	1	29	0.0	1	2

	region	tenure	age	marital	address	income	ed	employ	retire	gender	reside
3	2	33	33	0	12	33.0	2	0	0.0	1	1
4	2	23	30	1	9	30.0	1	2	0.0	0	4

In [22]:

```
y = df["custcat"]
y.head()
```

Out[22]:

```
0    1
1    4
2    3
3    1
4    3
Name: custcat, dtype: int64
```

In [25]:

```
from sklearn.preprocessing import StandardScaler
```

In [29]:

```
x = StandardScaler().fit(x).transform(x)
```

In [31]:

```
x
```

Out[31]:

```
array([[-0.02696767, -1.055125  ,  0.18450456, ..., -0.22207644,
        -1.03459817, -0.23065004],
       [ 1.19883553, -1.14880563, -0.69181243, ..., -0.22207644,
        -1.03459817,  2.55666158],
       [ 1.19883553,  1.52109247,  0.82182601, ..., -0.22207644,
         0.96655883, -0.23065004],
       ...,
       [ 1.19883553,  1.47425216,  1.37948227, ..., -0.22207644,
         0.96655883, -0.92747794],
       [ 1.19883553,  1.61477311,  0.58283046, ..., -0.22207644,
         0.96655883, -0.92747794],
       [ 1.19883553,  0.67796676, -0.45281689, ..., -0.22207644,
         0.96655883,  0.46617787]])
```

In [33]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
```

In [34]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [56]:

```
rissh = KNeighborsClassifier(n_neighbors=6)
iit = rissh.fit(x_train,y_train)
```

In [57]:

```
y_pred = iit.predict(x_test)
y_pred[0:5]
```

Out[57]:

```
array([4, 3, 3, 1, 1], dtype=int64)
```

In [58]:

```
y_pred.shape
```

Out[58]:

```
(200,)
```

In [59]:

```
from sklearn.metrics import accuracy_score
```

In [60]:

```
y_train_pred = iit.predict(x_train)
y_train_pred[0:5]
```

Out[60]:

```
array([2, 1, 2, 4, 3], dtype=int64)
```

In [61]:

```
print("Accuracy score of test data",accuracy_score(y_test,y_pred)*100)
print("Accuracy score of train data",accuracy_score(y_train,y_train_pred)*100)
```

```
Accuracy score of test data 38.0
Accuracy score of train data 51.87500000000001
```

In [ ]: