In [5]:

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

In [6]:

```
def check_empty_bins(dtm,bins):
    # check empty bins
    bin_list = np.unique(dtm[bins].astype(str)).tolist()
    if 'nan' in bin_list:
        bin_list.remove('nan')
    binleft = set([re.match(r'\((.+),(.+)\]', i).group(1) for i in bin_list]).di
    binright = set([re.match(r'\((.+),(.+)\]', i).group(2) for i in bin_list]).d
    if binleft != binright:
        bstbrks = sorted(list(map(float, ['-inf'] + list(binright) + ['inf'])))
        bstbrks.pop(-2)
        labels = ['[{},{})'.format(bstbrks[i], bstbrks[i + 1]) for i in range(le
        # print("The break points are modified into '[{}]'. There are empty bins
        # binning
        # dtm['bin'] = dtm['bin'].astype(str)
    # return
    return bstbrks
#字符型或者唯一值较少的变量
def psi1(data,psi_data,var):
    a = data[var].value_counts().reset_index(drop=False)#.astype(str)
    a.rename(columns={var:'name','count':'开发'+var},inplace=True)
    b=psi_data[var].value_counts().reset_index(drop=False)#.astype(str)
    b.rename(columns={var:'name','count':'验证'+var},inplace=True)
    m=pd.merge(a,b,on='name',how='inner')
    m[var+'开发频率']=m['开发'+var]/sum(m['开发'+var])
    m[var+'验证频率']=m['验证'+var]/sum(m['验证'+var])
    m['psi']=(m[var+'开发频率']-m[var+'验证频率'])*np.log(m[var+'开发频率']/m[va
    psi_sum=sum(m['psi'])
    return psi_sum
def psi2(data,psi_data,var,brk):
    a = pd.cut(data[var], brk, right=False).value_counts().reset_index(drop=Fals
    a.rename(columns={var:'name','count':'开发'+var},inplace=True)
    b=pd.cut(psi_data[var], brk,right=False).value_counts().reset_index(drop=Fal
    b.rename(columns={var:'name','count':'验证'+var},inplace=True)
    m=pd.merge(a,b,on='name',how='inner')
    m[var+'开发频率']=m['开发'+var]/sum(m['开发'+var])
    m[var+'验证频率']=m['验证'+var]/sum(m['验证'+var])
    m['psi']=(m[var+'开发频率']-m[var+'验证频率'])*np.log(m[var+'开发频率']/m[va
    psi_sum=sum(m['psi'])
    return psi_sum
def psi_hui(df,psi_data,target,n=5):
    chat=list(df.columns[df.dtypes == 'object'])
    name=df.columns.drop(target)
    psis=[]
    for i in name:
        X=df[i]
        Y=df[target]
        nuniq=X.nunique()
        if nuniq<=n:
            chat.append(i)
        if i in chat:
            psi=psi1(df,psi_data,i)
            psis.append(psi)
        else:
            d1=pd.DataFrame({"X":X,"Y":Y,"bin":pd.qcut(X,n,duplicates='drop')})
```

```
#            d1['bin']=d1['bin'].astype(str)
            brk=check_empty_bins(d1, 'bin')
            psi=psi2(df,psi_data,i,brk)
            psis.append(psi)
        print (i)
    d=pd.DataFrame({"name":name,"psi":psis})
    return d
```

In [7]:

```
data = pd.read_csv('test_data.csv')
print('Shape:',data.shape)
data.head(10)
```

```
Shape: (500, 7)
```

Out[7]:

	A	B	C	D	E	F	target
0	0.417022	87	0.154276	0.383389	NaN	84.100880	0
1	0.720324	57	0.758797	0.769808	NaN	74.668182	1
2	0.000114	24	0.197145	-0.105166	NaN	7.504475	0
3	0.302333	55	0.442048	0.300465	NaN	46.824730	1
4	0.146756	49	0.399363	0.096637	NaN	45.993059	1
5	0.092339	52	0.045981	-0.022774	NaN	50.068966	1
6	0.186260	24	0.322268	-0.039216	NaN	37.597628	0
7	0.345561	15	0.311033	0.412043	NaN	26.698194	1
8	0.396767	67	0.393938	0.435124	NaN	56.782580	0
9	0.538817	73	0.407933	0.509187	NaN	71.328520	0

In [8]:

```
psi_hui(data.iloc[0:250],data.iloc[250:500],'target')
```

```
A
B
C
D
E
F
```

Out[8]:

	name	psi
0	A	0.025587
1	B	0.038886
2	C	0.004393
3	D	0.037093
4	E	0.277259
5	F	0.015742

In [ ]: