单因子数据分析

In [21]:

```
import pandas as pd
import scipy.stats as ss
df = pd.read_csv('./data/HR.csv')
```

数据集探索

In [4]:

```
df.head()
```

Out[4]:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_co
0	0.38	0.53	2	157	
1	0.80	0.86	5	262	
2	0.11	0.88	7	272	
3	0.72	0.87	5	223	
4	0.37	0.52	2	159	

In [5]:

```
type(df)
```

Out[5]:

```
pandas.core.frame.DataFrame
```

In [6]:

```
type(df['satisfaction_level'])
```

Out[6]:

```
pandas.core.series.Series
```

In [7]:

```
# 均值
df.mean()
```

Out[7]:

```
satisfaction_level          0.612839
last_evaluation            67.373732
number_project              3.802693
average_monthly_hours     201.041728
time_spend_company          3.498067
Work_accident               0.144581
left                        0.238235
promotion_last_5years       0.021264
dtype: float64
```

In [8]:

```
# 中位数
df.median()
```

Out[8]:

```
satisfaction_level          0.64
last_evaluation             0.72
number_project              4.00
average_monthly_hours     200.00
time_spend_company          3.00
Work_accident               0.00
left                        0.00
promotion_last_5years       0.00
dtype: float64
```

In [12]:

```
# 中分位数-即中位数
df.quantile()
```

Out[12]:

```
satisfaction_level          0.64
last_evaluation             0.72
number_project              4.00
average_monthly_hours     200.00
time_spend_company          3.00
Work_accident               0.00
left                        0.00
promotion_last_5years       0.00
Name: 0.5, dtype: float64
```

In [13]:

```
# 下四分位数
df.quantile(q=0.25)
```

Out[13]:

```
satisfaction_level          0.44
last_evaluation             0.56
number_project              3.00
average_monthly_hours     156.00
time_spend_company          3.00
Work_accident               0.00
left                        0.00
promotion_last_5years       0.00
Name: 0.25, dtype: float64
```

In [14]:

```
# 众数
df.mode()
```

Out[14]:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_co
0	0.1	0.55	4.0	135	
1	NaN	NaN	NaN	156	

In [15]:

```
# 标准差
df.std()
```

Out[15]:

```
satisfaction_level          0.248623
last_evaluation          8164.407524
number_project              1.232733
average_monthly_hours      49.941815
time_spend_company          1.460053
Work_accident               0.351689
left                        0.426018
promotion_last_5years       0.144267
dtype: float64
```

In [16]:

```
# 方差
df.var()
```

Out[16]:

```
satisfaction_level       6.181359e-02
last_evaluation          6.665755e+07
number_project           1.519630e+00
average_monthly_hours    2.494185e+03
time_spend_company       2.131754e+00
Work_accident            1.236854e-01
left                     1.814911e-01
promotion_last_5years    2.081307e-02
dtype: float64
```

In [17]:

```
# 求和
df.sum()
```

Out[17]:

```
satisfaction_level                                        9192.59
last_evaluation                                       1.01074e+06
number_project                                              57048
average_monthly_hours                                     3016028
time_spend_company                                          52478
Work_accident                                                2169
left                                                         3574
promotion_last_5years                                         319
department               salessalessalessalessalessalessalessalessaless...
salary                   lowmediummediumlowlowlowlowlowlowlowlowlowl...
dtype: object
```

In [18]:

```
# 偏态系数
df.skew()
```

Out[18]:

```
satisfaction_level        -0.476438
last_evaluation          122.482652
number_project             0.337774
average_monthly_hours      0.053225
time_spend_company         1.853530
Work_accident              2.021481
left                       1.229057
promotion_last_5years      6.637677
dtype: float64
```

In [20]:

```
# 峰态系数
df.kurt()
```

Out[20]:

```
satisfaction_level          -0.670696
last_evaluation          15001.999987
number_project              -0.495810
average_monthly_hours       -1.135016
time_spend_company           4.774353
Work_accident                2.086664
left                        -0.489485
promotion_last_5years       42.064357
dtype: float64
```

正态分布

In [22]:

```
# 生成正态分布
zt = ss.norm
```

In [23]:

```
zt
```

Out[23]:

```
<scipy.stats._continuous_distns.norm_gen at 0x16967cbce48>
```

In [24]:

```
# 正态分布的指标：m-均值，v-方差，s-偏态系数，k-峰态系数
zt.stats(moments='mvsk')
```

Out[24]:

```
(array(0.), array(1.), array(0.), array(0.))
```

In [25]:

```
# 指定横坐标，返回纵坐标值
zt.pdf(0.0)
```

Out[25]:

```
0.3989422804014327
```

In [26]:

```
# 指定0-1之间的数，表示负无穷到这个值的累积值，就是上侧分位数
zt.ppf(0.9)
```

Out[26]:

```
1.2815515655446004
```

In [27]:

```
# 从负无穷到2的累积概率
zt.cdf(2)
```

Out[27]:

```
0.9772498680518208
```

In [28]:

```
# 正两倍的标准差-负两倍的标准差=所得的中间的累计概率
zt.cdf(2)-zt.cdf(-2)
```

Out[28]:

```
0.9544997361036416
```

In [29]:

```
# 得到size大小符合正态分布的数字
zt.rvs(size=10)
```

Out[29]:

```
array([ 0.58132524, -0.85269351, -0.4265185 ,  0.74607319, -1.75447628,
       -0.6633306 ,  1.85555626,  0.38276054,  0.05529418,  0.40815465])
```

卡方分布 | t分布 | f分布

In [30]:

```
chi = ss.chi2
```

In [31]:

```
chi
```

Out[31]:

```
<scipy.stats._continuous_distns.chi2_gen at 0x16967ca3a58>
```

In [32]:

```
t = ss.t
```

In [33]:

```
t
```

Out[33]:

```
<scipy.stats._continuous_distns.t_gen at 0x169678cd8d0>
```

In [34]:

```
f = ss.f
```

In [35]:

```
f
```

Out[35]:

```
<scipy.stats._continuous_distns.f_gen at 0x16967c58390>
```

抽样

In [37]:

```
# 抽10个样本
df.sample(n=10)
```

Out[37]:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spenc
7400	0.73	0.36	4	253	
6890	0.52	0.82	2	242	
6663	0.81	0.96	3	226	
4992	0.55	0.91	4	187	
10296	0.74	0.74	2	254	
2558	0.53	0.73	4	248	
11272	0.66	0.57	4	161	
488	0.10	0.83	7	276	
12098	0.11	0.97	6	284	
5958	0.97	0.61	3	208	

In [38]:

```
# 抽百分比为0.001的样本
df.sample(frac=0.001)
```

Out[38]:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spenc
6670	0.89	0.46	4	248	
7121	0.93	1.00	3	148	
9934	0.58	0.50	5	184	
8249	0.82	0.71	5	208	
11370	0.21	0.43	5	175	
2275	0.94	0.74	5	171	
8288	0.86	0.51	3	185	
12383	0.38	0.51	2	159	
10622	0.60	0.49	3	239	
7712	0.60	0.86	6	272	
7209	0.87	0.98	3	174	
3775	0.70	0.71	5	269	
5705	0.90	0.58	5	260	
8495	0.95	0.62	4	255	
14222	0.11	0.81	6	305	

In [ ]:

```
```