数据特征分析技能—— 分布分析

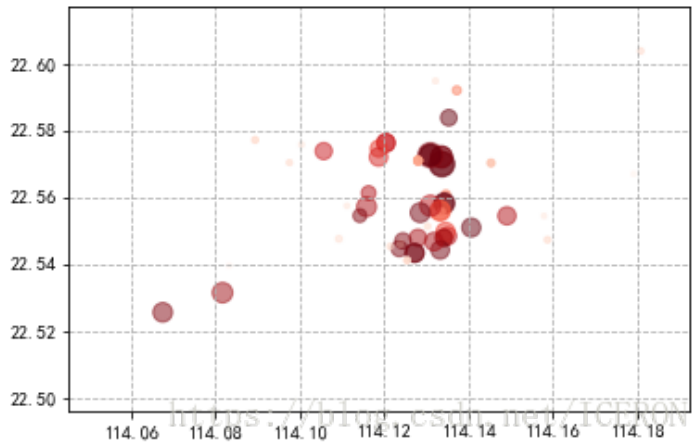
分布分析法又称**直方图法**。它是将搜集到的质量数据进行分组整理，绘制成频数分布直方图，用以***描述质量分布状态***的一种分析方法

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

```
data = pd.read_csv(r'E:\DataScience\Python\统计分析技能\深圳罗湖二手房息.csv',engine='python')
data.head()
```

	房屋编码	小区	朝向	房屋单价	参考首付	参考总价	经度	纬度
0	605093949	大望新平村	南北	5434	15.0	50.0	114.180964	22.603698
1	605768856	通宝楼	南北	3472	7.5	25.0	114.179298	22.566910
2	606815561	罗湖区罗芳村	南北	5842	15.6	52.0	114.158869	22.547223
3	605147285	兴华苑	南北	3829	10.8	36.0	114.158040	22.554343
4	606030866	京基东方都会	西南	47222	51.0	170.0	114.149243	22.554370

```
plt.scatter(data['经度'], data['纬度'],#  按照地理位置显示
                    s=data['房屋单价']/500,   #  按照单价显示大小
                    c=data['参考总价'],
                    cmap='Reds',alpha=0.5,)   #  按照总价显示颜色
plt.grid(linestyle='--')
```



极差

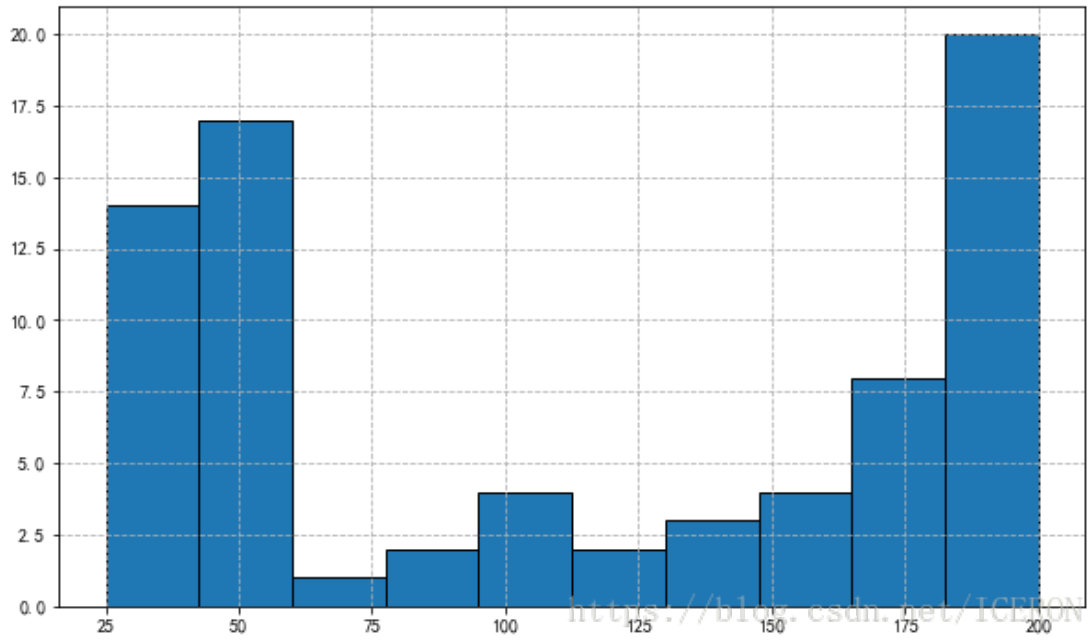
针对定量数据

```
# 极差
def d_range(df,*cols):
    krange=[]
    for col in cols:
        krange.append(df[col].max() - df[col].min())
    return krange

key1 = '参考首付'
key2 = '参考总价'
k = d_range(df,key1,key2)
print('%s 的极差为：%.2f \n%s 的极差为：%.2f'%(key1,k[0],key2,k[1]))
```

参考首付的极差为：52.50 参考总价的极差为：175.00

```
data['参考总价'].hist(bins=10, figsize=(10,6),edgecolor='black')
plt.grid(linestyle='--')
```



频率分布情况

```
# 频率的分布情况

gcut = pd.cut(data[key1],10,right=False) # right：是否包括末端值
gcut_count = gcut.value_counts(sort=False)
data['%s 区间'%key1] = gcut.values
data.head()
```

	房屋编码	小区	朝向	房屋单价	参考首付	参考总价	经度	纬度	参考首付区间
0	605093949	大望新平村	南北	5434	15.0	50.0	114.180964	22.603698	[12.75, 18.0)
1	605768856	通宝楼	南北	3472	7.5	25.0	114.179298	22.566910	[7.5, 12.75)
2	606815561	罗湖区罗芳村	南北	5842	15.6	52.0	114.158869	22.547223	[12.75, 18.0)
3	605147285	兴华苑	南北	3829	10.8	36.0	114.158040	22.554343	[7.5, 12.75)
4	606030866	京基东方都会	西南	47222	51.0	170.0	114.149243	22.554370	[49.5, 54.75)

```
# 区间出现的频率
df1 = pd.DataFrame(gcut_count)
df1.rename(columns={gcut_count.name:'频数'},inplace=True)
df1['频率'] = df1['频数'] / df1['频数'].sum()
df1['累计频率'] = df1['频率'].cumsum()
df1['频率%'] = df1['频率'].apply(lambda x: '%.2f%%'%(x*100))
df1['累计频率%'] = df1['累计频率'].apply(lambda x: '%.2f%%'%(x*100))
df1.style.bar(subset=['频率','累计频率'],width=100)
```

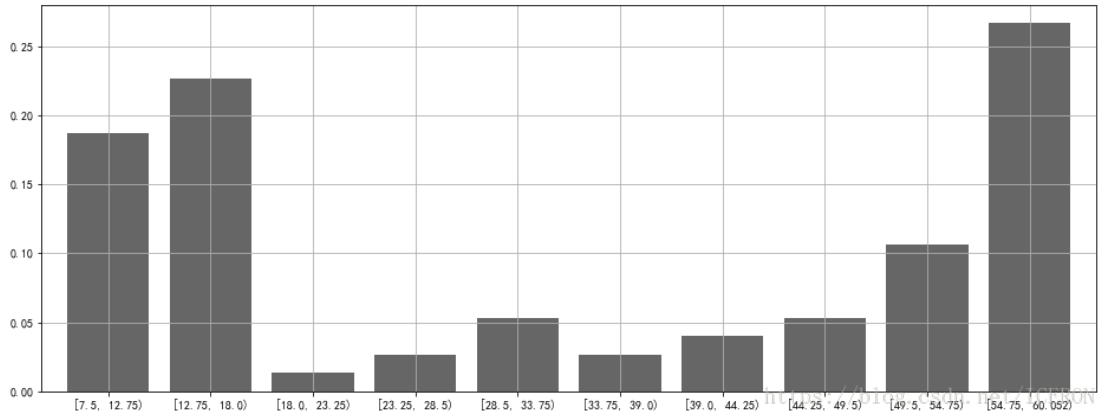
	频数	频率	累计频率	频率%	累计频率%
[7.5, 12.75)	14	0.186667	0.186667	18.67%	18.67%
[12.75, 18.0)	17	0.226667	0.413333	22.67%	41.33%
[18.0, 23.25)	1	0.0133333	0.426667	1.33%	42.67%
[23.25, 28.5)	2	0.0266667	0.453333	2.67%	45.33%
[28.5, 33.75)	4	0.0533333	0.506667	5.33%	50.67%
[33.75, 39.0)	2	0.0266667	0.533333	2.67%	53.33%
[39.0, 44.25)	3	0.04	0.573333	4.00%	57.33%
[44.25, 49.5)	4	0.0533333	0.626667	5.33%	62.67%
[49.5, 54.75)	8	0.106667	0.733333	10.67%	73.33%
[54.75, 60.052)	20	0.266667	1	26.67%	100.00%

绘制直方图

其实是平常并不必要那么麻烦的计算区间啊什么的，直接使用 plt.hist 就可以了

```
# 直方图

df1['频率'].plot(kind='bar',figsize=(16,6),width=0.8,
                 rot=0, color='k',grid=True,alpha=0.6)
```



定性字段的频率分布

方法相似

```
df2 = data['朝向'].value_counts()
print(df2)

# 统计票频率

df2_cx = pd.DataFrame(df2)
df2_cx.rename(columns={df2.name:'频数'}, inplace=True)
df2_cx['频率'] = df2_cx['频数'] / df2_cx['频数'].sum()
df2_cx['累计频率'] = df2_cx['频率'].cumsum()
print(df2_cx)

df2_cx.style.bar(subset=['频率','累计频率'], color='#d65f5f',width=100)
```

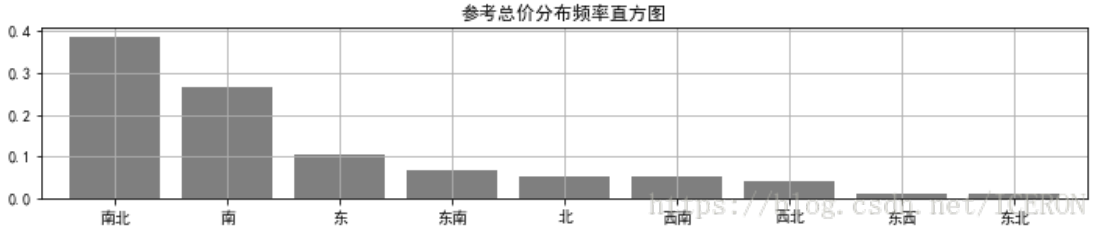
```
南北    29
南      20
东       8
东南     5
北       4
西南     4
西北     3
东西     1
东北     1
Name: 朝向, dtype: int64
    频数        频率       累计频率
南北  29  0.386667  0.386667
南   20  0.266667  0.653333
东    8  0.106667  0.760000
东南   5  0.066667  0.826667
北    4  0.053333  0.880000
西南   4  0.053333  0.933333
西北   3  0.040000  0.973333
东西   1  0.013333  0.986667
东北   1  0.013333  1.000000
```

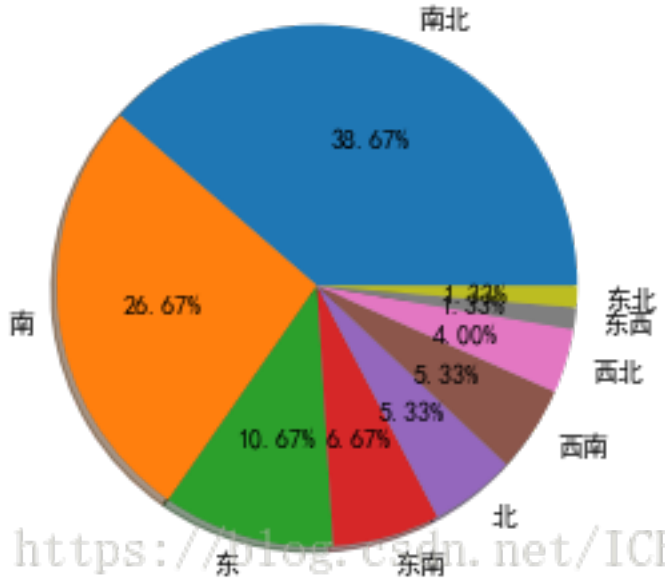
	频数	频率	累计频率
南北	29	0.386667	0.386667
南	20	0.266667	0.653333
东	8	0.106667	0.76
东南	5	0.0666667	0.826667
北	4	0.0533333	0.88
西南	4	0.0533333	0.933333
西北	3	0.04	0.973333
东西	1	0.0133333	0.986667
东北	1	0.0133333	1

```
# 绘制频率直方图和饼图

plt.figure(num = 1,figsize = (12,2))
df2_cx['频率'].plot(kind = 'bar',
                    width = 0.8,
                    rot = 0,
                    color = 'k',
                    grid = True,
                    alpha = 0.5)
plt.title('参考总价分布频率直方图')

plt.figure(num = 2)
plt.pie(df2_cx['频数'],
        labels = df2_cx.index,
        autopct='%.2f%%',
        shadow = True)
plt.axis('equal')
plt.show()
```





建议在分析数据分布的时候，在**尽量不选用饼图**，特别是在差别不明显的时候，饼图并不容易看出大小差异。**直方图或者柱状图**可以做替代。