In [ ]:
```
'''
【课程1.2】 分布分析

分布分析 → 研究数据的分布特征和分布类型，分定量数据、定性数据区分基本统计量

极差/频率分布情况/分组组距及组数

'''
```

In [7]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

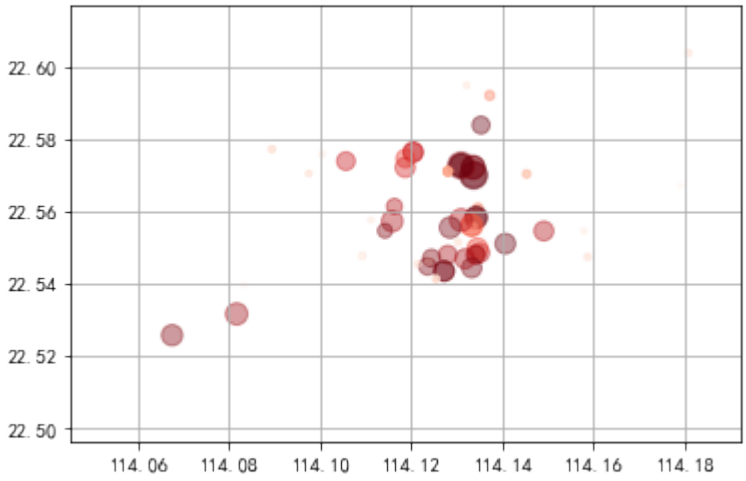
In [143]:
```
#数据读取

path = r"C:\Users\Administrator\Desktop\数据分析网易\00-0 QQ群资料\【非常重要】课程资料\CLASSDATA_ch04进阶算法学习：统计分析能力强化【瑞客论坛 www.ruike1.com】\CLASSDATA_ch04进阶算法学习：统计分析能力强化\CH01数据特征分析\深圳罗湖二手房信息.csv"
path = path.replace("\\","/")

data = pd.read_csv(path,engine="python")
plt.scatter(data["经度"],data["纬度"],#按照经纬度显示
            s = data["房屋单价"]/500,#按照单价显示大小
            c = data["参考总价"],#按照总价显示颜色
            alpha = 0.4, cmap = 'Reds')

plt.grid()
print('-------\n数据长度为%i条' % len(data))
data
```

```
-------
数据长度为75条
```

Out[143]:

	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
...	...	...	...	...	...	...	...	...
70	598258845	三九花园	南	5833	12.6	42.0	114.089539	22.577080
71	594221866	三九花园	南	5681	15.0	50.0	114.089539	22.577080
72	606700179	城市春天	南北	3571	7.5	25.0	114.083405	22.539505
73	603950517	皇御苑	东北	59701	54.0	180.0	114.081795	22.531393
74	605232094	晨晖家园	南	54285	57.0	190.0	114.067625	22.525508

75 rows × 8 columns

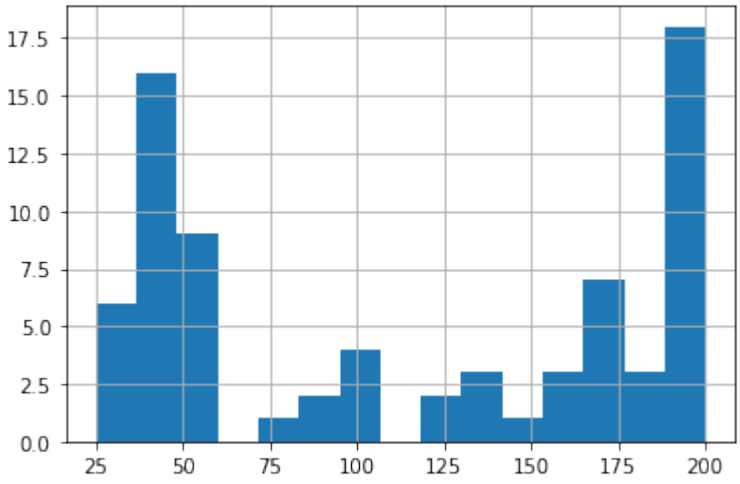


In [37]:
```
#极差max-min
#只针对定量字段

def d_range(df,*cols):
    krange=[]
    for col in cols:
        crange = df[col].max() - df[col].min()
        krange.append(crange)
    return(krange)
#创建函数求极差

key1,key2="参考首付","参考总价"
dr = d_range(data,key1,key2)
print(dr[0],dr[1])
#求出数据对应列的极差
```

52.5 175.0

In [43]:
```
#频率分布情况 - 定量字段
#① 通过直方图直接判断分组组数

data[key2].hist(bins=15)
#简单查看数据分布，确定分布组数 → 一般8-16即可
#这里以15组为参考
```

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x223b5588088>



In [ ]:
```
#cut 用法：https://www.cnblogs.com/sench/p/10128216.html
```

In [63]:
```
#频率分布情况 - 定量字段
#② 求出分组区间

gcut = pd.cut(data[key2],10,right=False) #right是否包含左边的区间
gcut_count = gcut.value_counts(sort=False) #不排序
gcut.values
data[f"{key2}分组区间"] = gcut.values
print(gcut.head())
print(gcut_count)
data.head()
#pd.cut(x, bins, right)：按照组数对x分组，且返回一个和x同样长度的分组dataframe，right → 是否右边包含，默认True
#通过groupby查看不同组的数据频率分布
#给源数据data添加“分组区间”列
```

```
0     [42.5, 60.0)
1     [25.0, 42.5)
2     [42.5, 60.0)
3     [25.0, 42.5)
4    [165.0, 182.5)
Name: 参考总价, dtype: category
Categories (10, interval[float64]): [[25.0, 42.5) < [42.5, 60.0) < [60.0, 77.5) < [77.5, 95.0) ... [130.0, 147.5) <
 [147.5, 165.0) < [165.0, 182.5) < [182.5, 200.175)]
[25.0, 42.5)       14
[42.5, 60.0)       17
[60.0, 77.5)        1
[77.5, 95.0)        2
[95.0, 112.5)       4
[112.5, 130.0)      2
[130.0, 147.5)      3
[147.5, 165.0)      4
[165.0, 182.5)      8
[182.5, 200.175)   20
Name: 参考总价, dtype: int64
```

Out[63]:

	房屋编码	小区	朝向	房屋单价	参考首付	参考总价	经度	纬度	参考总价分组区间
0	605093949	大望新平村	南北	5434	15.0	50.0	114.180964	22.603698	[42.5, 60.0)
1	605768856	通宝楼	南北	3472	7.5	25.0	114.179298	22.566910	[25.0, 42.5)
2	606815561	罗湖区罗芳村	南北	5842	15.6	52.0	114.158869	22.547223	[42.5, 60.0)
3	605147285	兴华苑	南北	3829	10.8	36.0	114.158040	22.554343	[25.0, 42.5)
4	606030866	京基东方都会	西南	47222	51.0	170.0	114.149243	22.554370	[165.0, 182.5)

In [80]:
```
#频率分布情况 - 定量字段
#③ 求出目标字段下频率分布的其他统计量 → 频数，频率，累计频率
#pandas.core.series.Series 可以.name把名字取出来

r_zj = pd.DataFrame(gcut_count)
r_zj.rename(columns = {gcut_count.name:"频数"},inplace=True)
r_zj['频率'] = r_zj["频数"] / r_zj["频数"].sum()
r_zj['累计频率'] = r_zj['频率'].cumsum()
```

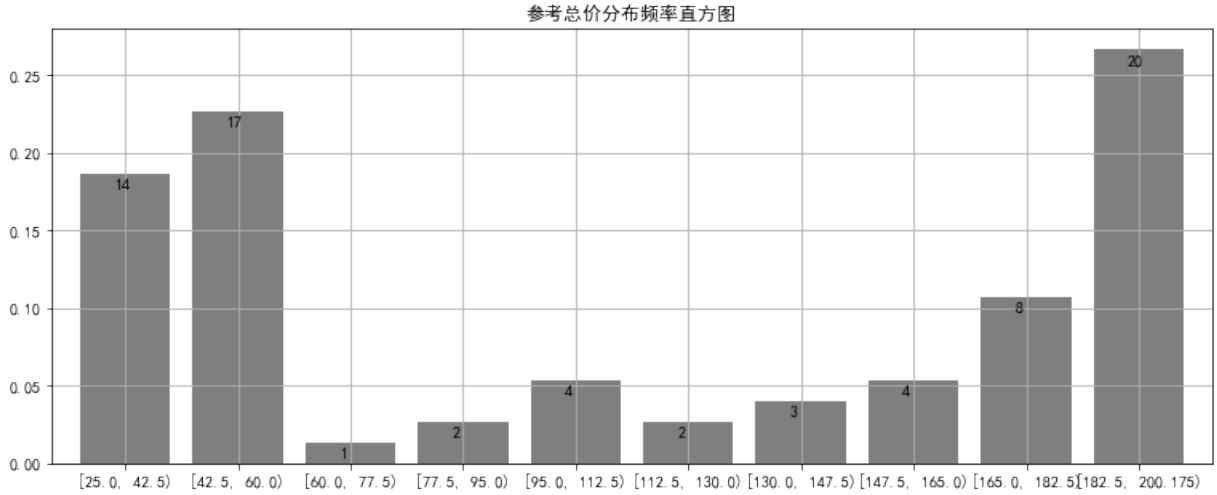
```
r_zj['频率%'] = r_zj['频率'].apply(lambda x:"%.2f" %(x*100))
r_zj['累计频率%'] = r_zj['累计频率'].apply(lambda x: "%.2f%%" % (x*100))
r_zj.style.bar(subset=['频率','累计频率'], color='green',width=100)
# 图的讲解https://www.jianshu.com/p/5c1491d708e0++++++++++++
```

Out[80]:

	频数	频率	累计频率	频率%	累计频率%
[25.0, 42.5)	14	0.186667	0.186667	18.67	18.67%
[42.5, 60.0)	17	0.226667	0.413333	22.67	41.33%
[60.0, 77.5)	1	0.013333	0.426667	1.33	42.67%
[77.5, 95.0)	2	0.026667	0.453333	2.67	45.33%
[95.0, 112.5)	4	0.053333	0.506667	5.33	50.67%
[112.5, 130.0)	2	0.026667	0.533333	2.67	53.33%
[130.0, 147.5)	3	0.040000	0.573333	4.00	57.33%
[147.5, 165.0)	4	0.053333	0.626667	5.33	62.67%
[165.0, 182.5)	8	0.106667	0.733333	10.67	73.33%
[182.5, 200.175)	20	0.266667	1.000000	26.67	100.00%

In [115]:

```
# 频率分布情况 - 定性字段
# ④ 绘制频率直方图

r_zj['频率'].plot(kind = 'bar',
                  width = 0.8,
                  figsize = (13,5),
                  rot = 0,
                  color = 'k',
                  grid = True,
                  alpha = 0.5)

plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False
plt.title('参考总价分布频率直方图')
# 绘制直方图

x = len(r_zj)
y = r_zj['频率']
m = r_zj['频数']
for i,j,k in zip(range(x),y,m):
    plt.text(i-0.1,j-0.01,'%i' % k, color = 'k')
# 添加频数标签
# https://blog.csdn.net/TeFuirnever/article/details/88947248
```



In [137]:
```
#频率分布情况 - 定性字段
#① 通过计数统计判断不同类别的频率

cx_g = data['朝向'].value_counts(sort=True)


r_cx = pd.DataFrame(cx_g)
r_cx.rename(columns = {cx_g.name:"频数"},inplace=True)
r_cx['频率'] = r_cx / r_cx['频数'].sum()
r_cx["累计频率"] = r_cx['频数'] / r_cx['频数'].sum()
r_cx["频率%"] = r_cx['累计频率'].apply(lambda x:"%.2f%%" % (x*100) )
r_cx['累计频率%'] = r_cx['累计频率'].apply(lambda x: "%.2f%%" % (x*100))  #以百分比显示累计频率
#r_cx
r_cx.style.bar(subset=['频率','累计频率'], color='#d65f5f',width=100)
```

Out[137]:

	频数	频率	累计频率	频率%	累计频率%
南北	29	0.386667	0.386667	38.67%	38.67%
南	20	0.266667	0.266667	26.67%	26.67%
东	8	0.106667	0.106667	10.67%	10.67%
东南	5	0.066667	0.066667	6.67%	6.67%
北	4	0.053333	0.053333	5.33%	5.33%
西南	4	0.053333	0.053333	5.33%	5.33%
西北	3	0.040000	0.040000	4.00%	4.00%
东北	1	0.013333	0.013333	1.33%	1.33%
东西	1	0.013333	0.013333	1.33%	1.33%

In [142]:
```
#频率分布情况 - 定量字段
#② 绘制频率直方图、饼图

plt.figure(num=1,figsize=(12,2))
r_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')
#绘制直方图

plt.figure(num=2)
plt.pie(r_cx['频数'],
        labels=r_cx.index,
        autopct="%.2f%%",
        shadow = True)

plt.axis('equal')
#https://www.cnblogs.com/biyoulin/p/9565350.html
```

Out[142]: (-1.1101621526291232,
1.1004839130571389,
-1.1062755172910221,
1.1205348076125872)