项目概述：

房价一直是很多人关注的热点，本项目爬取了链家杭州市的楼盘，通过对楼盘位置、楼盘价格等方面来进行几个方面的展示和分析，探究大多数杭州的用户在购房时是怎样的打算

In [1]:
```
#导入所需模块
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
import requests
import matplotlib.pyplot as plt
from matplotlib import cm
import seaborn as sn
from pyecharts import Bar
from pyecharts import Pie
from pyecharts import Boxplot
import statsmodels.api as sm
sn.set_style('darkgrid')
%matplotlib inline
#设置可视化可用中文显示
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus']=False
```

```
D:\program\lib\site-packages\statsmodels\compat\pandas.py:56: FutureWarning: The pandas.core.datetools module is deprecated and will be rem
oved in a future version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
```

In [2]:
```
#建立各类别的空列表
name = []
price = []
loc = []
style = []
state = []
area = []
```

In [3]:
```
#爬取链家上杭州楼盘信息
for i in range(1,83):
    url = 'https://hz.fang.lianjia.com/loupan/pg{}/'.format(i)
    response = requests.get(url)
    soup = BeautifulSoup(response.content,'lxml')
```

```
    for j in range(10):
        #添加房屋名称
        name.append(soup.find_all(attrs={'class':'name'})[j].contents[0])
        #添加房屋价格
        price.append(soup.find_all(attrs={'class':'main-price'})[j].contents[1].contents[0])
        #添加房屋位置
        loc.append(soup.find_all(attrs={'class':'resblock-location'})[j].contents[1].contents[0])
        #添加房屋类型
        style.append(soup.find_all(attrs={'class':'resblock-type'})[j].contents[0])
        #添加房屋状态
        state.append(soup.find_all(attrs={'class':'sale-status'})[j].contents[0])
        #添加房屋面积
        try:
            area.append(soup.find_all(attrs={'class':'resblock-area'})[j].contents[1].contents[0])
        except:
            area.append('null')
```

In [4]:
```
#创建字典
dic = {'name':name,
       'price':price,
       'location':loc,
       'style':style,
       'state':state,
       'area':area}
```

In [5]:
```
#通过字典创建DataFrame二维表
df = pd.DataFrame(dic)
```

In [6]:
```
df.head()
```

Out[6]:

	area	location	name	price	state	style
0	建面 89-134㎡	西湖	中国铁建西湖国际城	23233	未开盘	住宅
1	null	临安	官山邸	17003	在售	别墅
2	建面 88-92㎡	海宁市	欣隆府	14800	在售	住宅
3	建面 74㎡	海盐县	海盐阳光城翡丽湾	12000	在售	住宅
4	建面 40-64㎡	拱墅	赞成星谷	130	在售	商业类

In [7]:
```
# 查看数据类型和缺失值
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 820 entries, 0 to 819
Data columns (total 6 columns):
area        820 non-null object
location    820 non-null object
name        820 non-null object
price       820 non-null object
```

```
state       820 non-null object
style       820 non-null object
dtypes: object(6)
memory usage: 38.5+ KB
```

In [8]:
```
#由于 style 商业类和商业同属一类房屋类型，为便于数据整理，现替换所有商业类为商业
df['style'] = df['style'].str.replace("类","")
```

In [9]:
```
#看到area下数据不易之后的统计，现删除建面和m²
df['area'] = df['area'].str.replace("建面","").str.replace("m²","")
```

In [10]:
```
#查看了下现在的area，发现存在多个数值，我想把值拆分开，分别放到两个列中
df[['area_lower','upper']] = df.area.str.split("-",expand=True)
```

In [11]:
```
#查看重复行
df.duplicated().value_counts()
```

Out[11]:
```
False    815
True       5
dtype: int64
```

In [12]:
```
#删除重复行
df.drop_duplicates(inplace=True)
```

In [13]:
```
df.head()
```

Out[13]:

	area	location	name	price	state	style	area_lower	upper
0	89-134	西湖	中国铁建西湖国际城	23233	未开盘	住宅	89	134
1	null	临安	官山邸	17003	在售	别墅	null	None
2	88-92	海宁市	欣隆府	14800	在售	住宅	88	92
3	74	海盐县	海盐阳光城翡丽湾	12000	在售	住宅	74	None
4	40-64	拱墅	赞成星谷	130	在售	商业	40	64

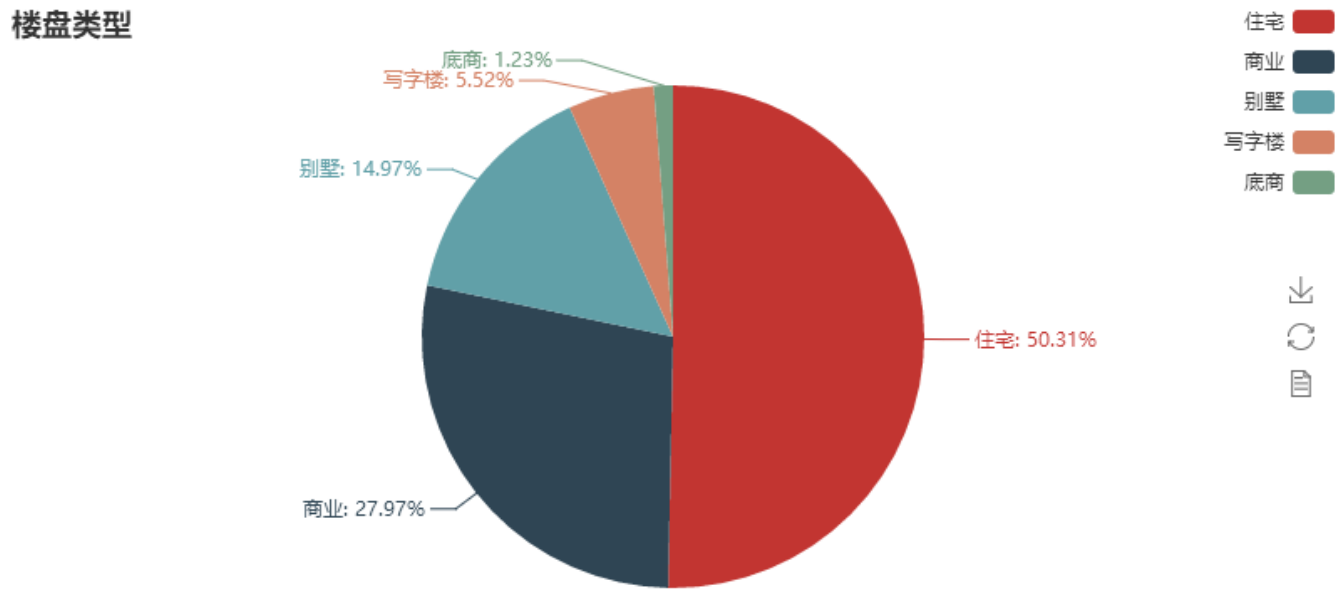
以上是对爬取的初始数据进行整理，包括查看数据类型和缺失值，删除重复行，更改商业类为商业，增加最小面积与最大面积两列，关于面积方面我没有做出探索，所以这次的可视化与数据分析与面积无关

In [17]:
```
#分别把索引和对应值存贮给变量
sty = df['style'].value_counts().index
value = df['style'].value_counts()
```

In [18]:
```
#引用上面变量绘制饼图
pie = Pie("楼盘类型")
pie.add("类型",sty,value,legend_orient='vertical',legend_pos='right',is_label_show=True)
```

```
pie
```

Out[18]:

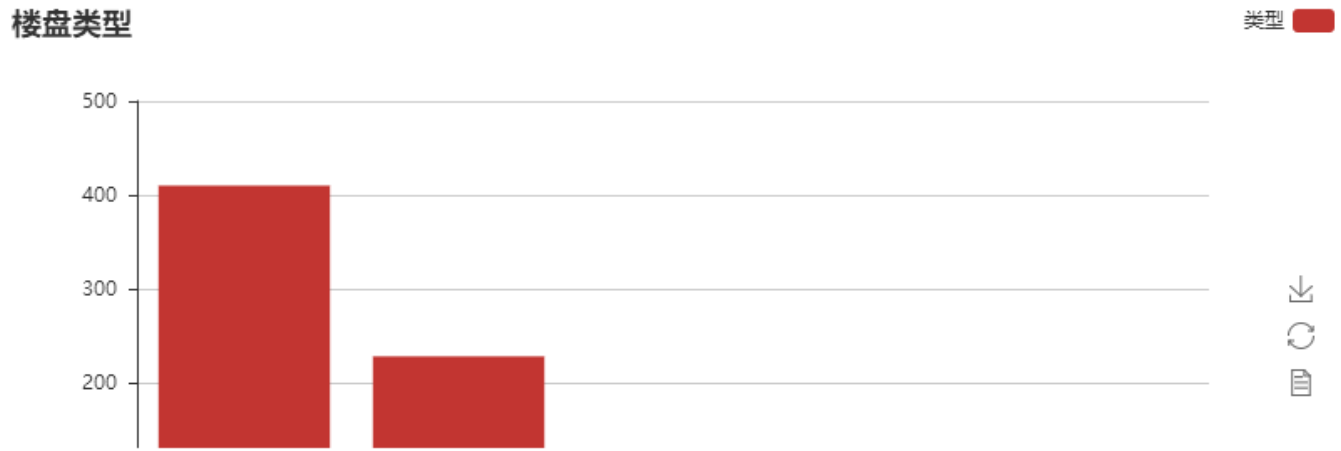


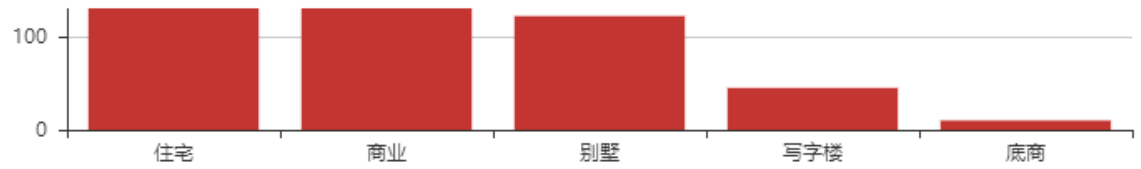
首先可视化我们搜集的该网站给出的所有杭州的楼盘，查看各个楼盘类型所占比重，看到住宅的比重占到一半，推断开发商的主要产品是住宅类型，其次是商业房

In [19]:

```
bar = Bar("楼盘类型")
bar.add("类型",sty,value,legend_orient='vertical',legend_pos='right')
bar
```

Out[19]:





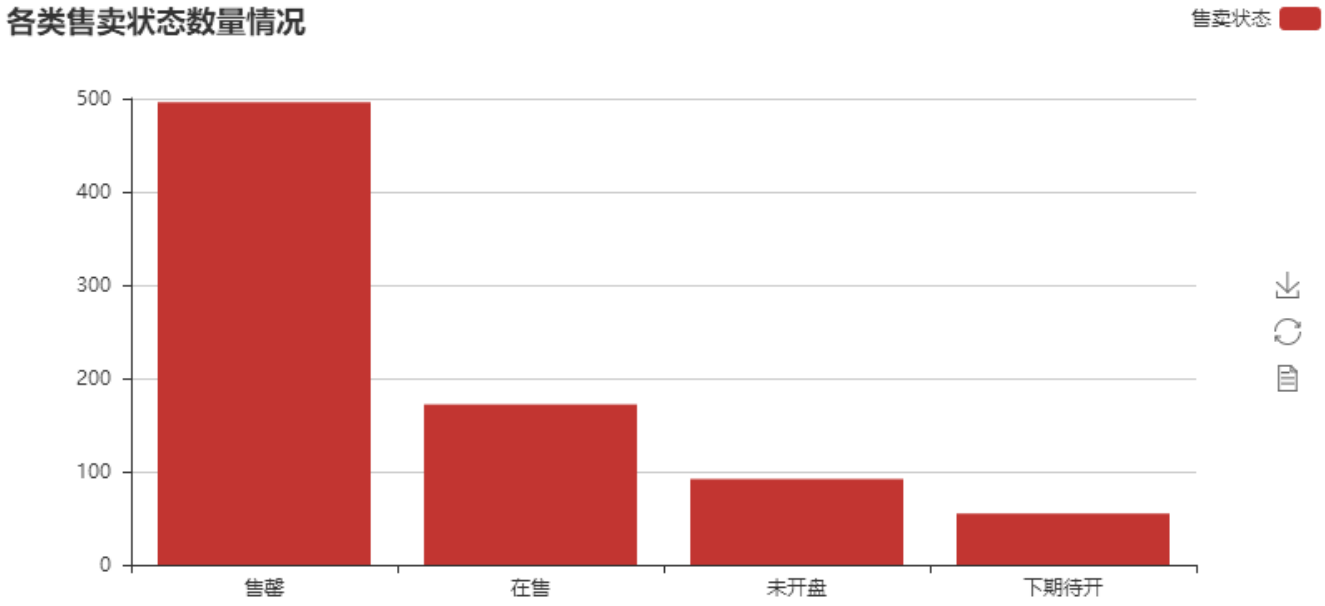
条形图查看各类型楼盘数量

之前我在表中看到部分楼盘售罄、部分在售、部分未开盘，如果我是购房者，我想在选房之前看下售出的楼盘中哪些地段卖的比较多，这可能帮助我选择一个不错的地段，但在这之前我想看下该网站给出的楼盘销量怎么样，靠不靠谱我不知道，但销量多总好过于销量少，起码我能接下来根据销量进一步探究。

In [20]:
```
#分别把索引和对应值存贮给变量
sta = df.state.value_counts().index
value = df.state.value_counts()
```

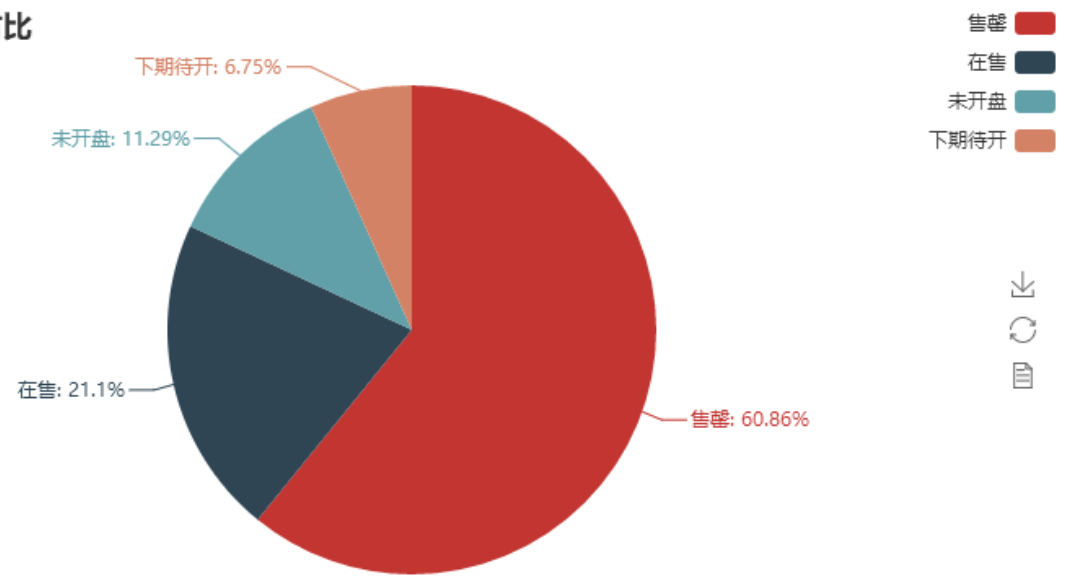
In [22]:
```
#看销量的话个人偏好先观察下条形图
bar = Bar("各类售卖状态数量情况")
bar.add("售卖状态",sta,value,legend_orient='vertical',legend_pos='right')
bar
```

Out[22]:



In [23]:
```
#比重的话我也给出来了
pie = Pie("各类售卖状态所占比")
pie.add("售卖状态",sta,value,legend_orient='vertical',legend_pos='right',is_label_show=True)
pie
```

Out[23]:



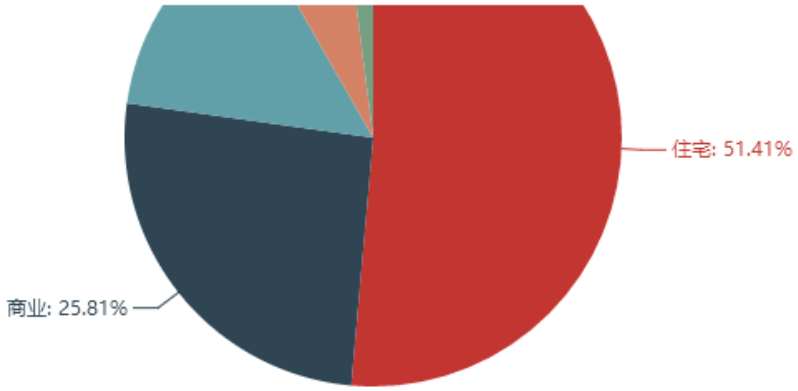
到目前为止售罄的比重占到60.86%，除去18%未开盘的，还剩21.1%在售楼盘。就数据猜测杭州新房很抢手，当然我担心数据不全，所以现在只能做个猜想

现在我看下已售罄的楼盘中用户对哪些类型的楼盘需求量大，还有哪个区域最受用户青睐，为此我需要制作两个图，分别是已售罄的楼盘和在售的楼盘加已售罄的楼盘

In [30]:
```
#只要售罄楼盘的数据
df_sell_out = df.query("state == '售罄'")
value = df_sell_out['style'].value_counts()
sty = df_sell_out['style'].value_counts().index
```

In [31]:
```
pie = Pie(title="只保留已售罄的楼盘")
pie.add("类型",sty,value,legend_orient='vertical',legend_pos='right',is_label_show=True)
```

Out[31]:

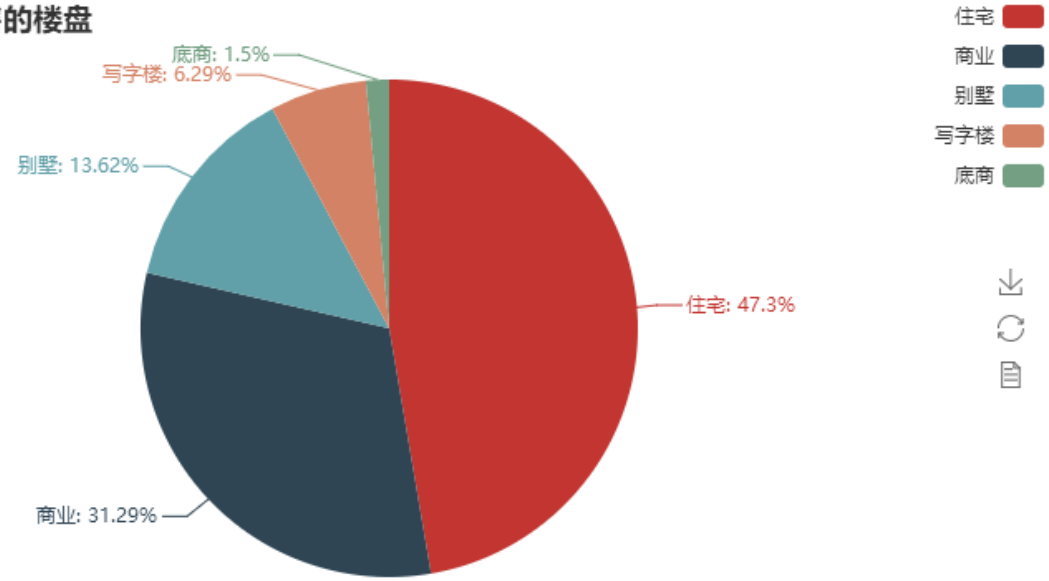


现在只保留了已售罄的楼盘，方便与下图做比较

In [28]:
```
#只要售罄楼盘和在售楼盘的数据
df_ing_ed = df.query('state == ["在售","售罄"]')
sty = df_ing_ed['style'].value_counts().index
value = df_ing_ed['style'].value_counts()
```

In [29]:
```
pie = Pie("只保留已售罄与在售的楼盘")
pie.add("类型",sty,value,legend_orient='vertical',legend_pos='right',is_label_show=True)
```

Out[29]:



对比上两图可以进一步对给出的数据进行猜测，用户对住宅的需求的比重，要大于市面上可出售住宅的比重，相反的，商业类住房销售程度要低于理想情况下的的销售量，这里其它类型的住房变动不是很明显，由于数据量的限制，这里不做猜测

误差来源：

1.由于数据集中没有时间项，可能部分楼盘售卖结果受日期的影响

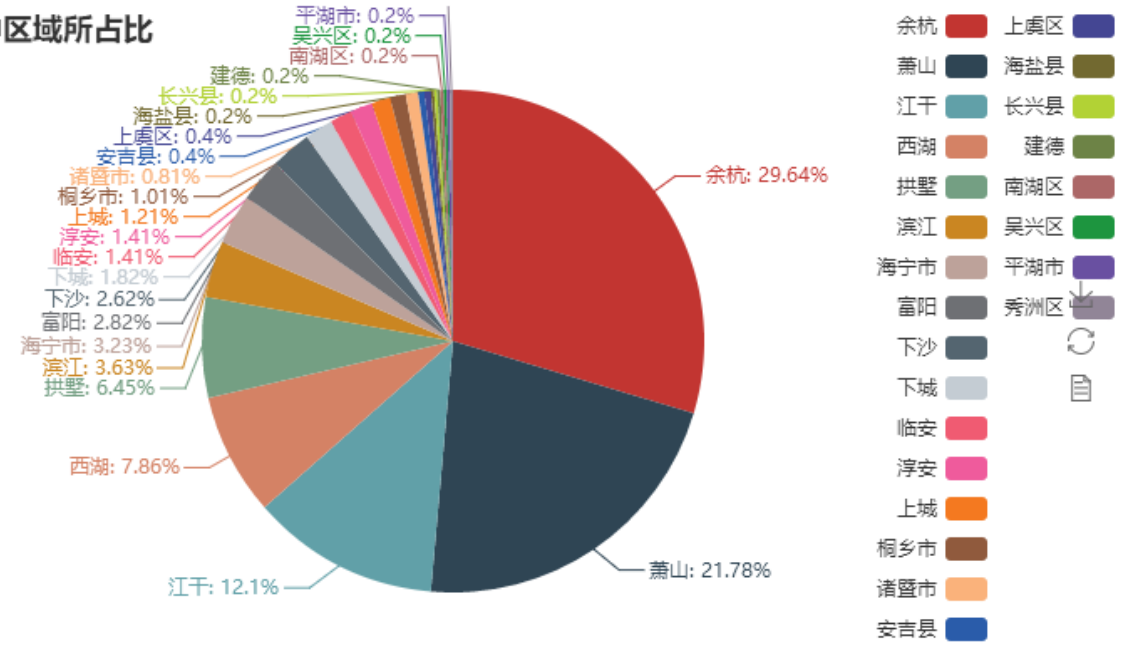
2.由于数据仅是爬取了一个网站，并不能保证数据集的完整性

3.可能受楼盘批次的影响，每批次能够售卖的楼盘类型数量略有不同

现在我们来看下售出的楼盘所属区域情况，进而能推测普遍用户更青睐哪一区域

In [32]:
```
# 这次是给出售出区域的比重
value = df_sell_out.location.value_counts()
lists = df_sell_out.location.value_counts().index
pie = Pie("已售罄的楼盘中区域所占比")
pie.add("区域",lists,value,legend_orient='vertical',legend_pos='right',is_label_show=True)
```

Out[32]:



对这一结果我的猜想有两点

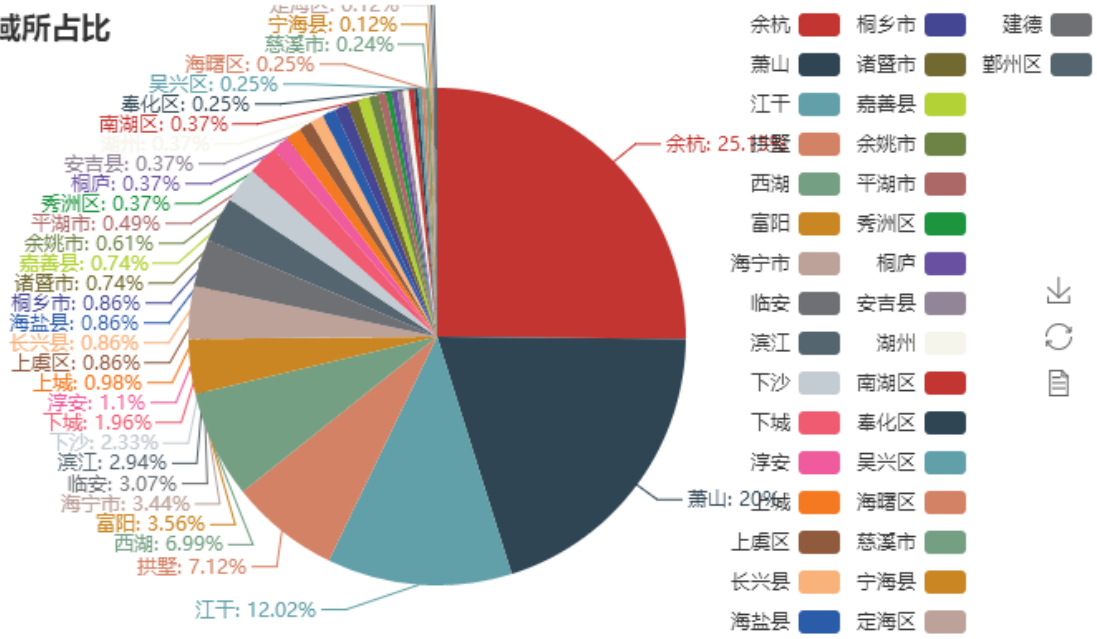
1.可能占比重最大的前四个区所给出的楼盘数量一开始就比其他区域多。

2.可能大部分用户出于经济和工作上的考量选择了交通相对便利，房价相对低些的区域

现在我想证明下上面的猜想 **1** ，只需要看下所有的楼盘区域比重就可以了

In [33]:
```
# 所有楼盘区域的比重
value = df.location.value_counts()
lists = df.location.value_counts().index
pie = Pie("所有的楼盘中区域所占比")
pie.add("区域",lists,value,legend_orient='vertical',legend_pos='right',is_label_show=True)
```

Out[33]:



结果很明显，前几个区域不仅楼盘多，而且卖的也不错，这也映射这些区域确实是许多开发商和用户的重点关注地，性价比的首选，遗憾现在缺少购房用户的薪资与择业信息，如此便能进一步探索用户所选择这些地方的更多原因了

接着我想证明猜想 **2**：“可能大部分用户出于经济和工作上的考量选择了交通相对便利，房价相对低些的区域”。为此我需要两个图，一个是所有区域的平均房价的条形图，一个是所有区域包含的所有房价的箱线图

In [34]:
```
# 删除没有给出价格的楼盘
h = df['price'] == '价格待定'
df_new = df[~h]
```

In [35]:
```
# 把价格从字符串类型转换为数值型
df_new['price'] = df_new['price'].str[:].astype(int)
```

```
D:\program\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
```

In [36]:
```
# 删除个别的 xxx/套的楼盘
df_new = df_new.query('price > 8000')
```

In [37]:
```
# 计算每个地段的面积均价
price_mean = df_new.groupby('location')['price'].mean()
price_mean = price_mean.sort_values(ascending=False)
```

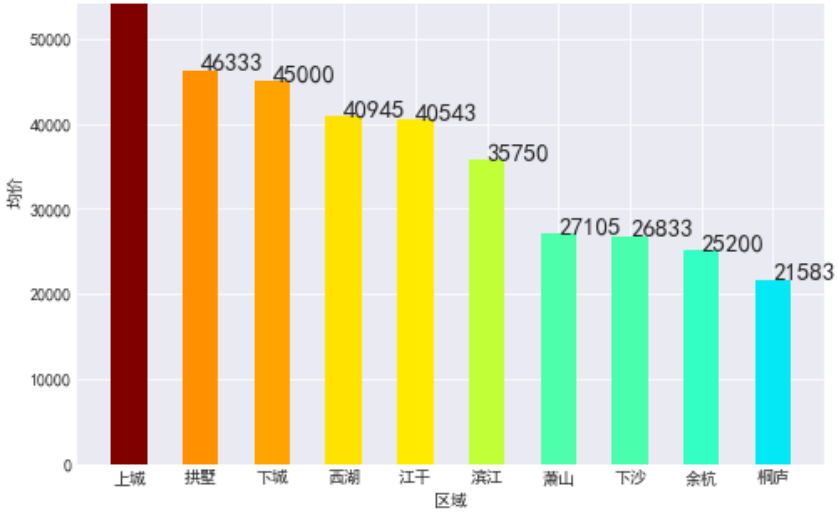
In [38]:
```
# 将得出的面积均值值与地段名分别添加到列表
mean = []
lists = []
for _ in range(10):
    mean.append(price_mean.values[_])
    lists.append(price_mean.index[_])
```

In [39]:
```
# 设置柱状图颜色
color = cm.jet(np.array(mean)/max(mean))
```

In [40]:
```
# 做出均价前十名地区的柱状图
plt.figure(figsize=(8,6))
plt.xlabel("区域")
plt.ylabel("均价")
plt.title("所有区域均价的 TOP10")
for _ in range(len(mean)):
    plt.text(lists[_],mean[_],int(mean[_]),fontsize=15)
plt.bar(left = lists,height = mean,width=0.5,color=color,yerr=1)
```

```
D:\program\lib\site-packages\matplotlib\__init__.py:1855: MatplotlibDeprecationWarning: The *left* kwarg to `bar` is deprecated use *x* ins
tead. Support for *left* will be removed in Matplotlib 3.0
  return func(ax, *args, **kwargs)
```

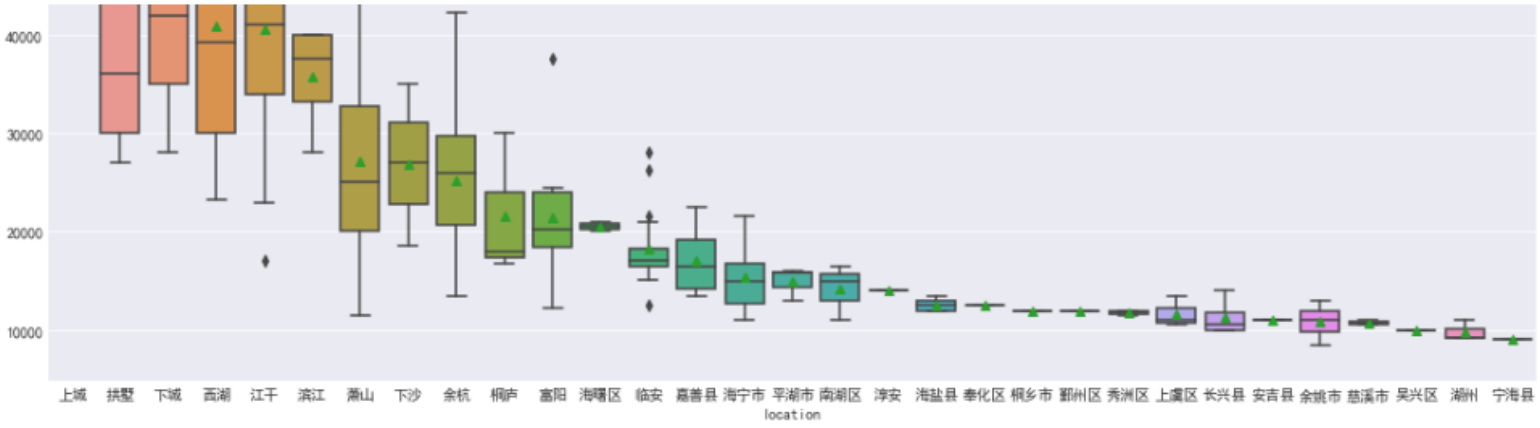
Out[40]: <BarContainer object of 10 artists>



出于前面得出的热销地区前五名，我主要关注余杭、萧山、江干、拱墅、西湖。我发现这五个区他们的均值呈阶梯形，其中西湖区与江干区，萧山区与余杭区的均价较为接近。所以大致上三个价位分别是46000，41000，26000。

In [41]:
```
# 计算所有地区房屋面积均值
rank = df_new.groupby('location')['price'].mean().sort_values(ascending=False).index
```

In [42]:
```
# 绘制箱线图
plt.figure(figsize=(18,10))
plt.title("区域与楼盘价格")
sn.boxplot(x="location",y="price",data=df_new[['location','price']],order = rank,showmeans=True);
```



箱线图中我按照各区域楼房面积平均值从大到小排序保留了所有的行政区，图中的绿三角代表各个行政区楼房面积的均价，均值与中位数不等会形成偏态分布，回到我们之前着重观察的五个热销地，其中拱墅、西湖、萧山呈右偏态，说明该地区房屋均价受价格偏高的楼房较为明显，继续对比这五个地区的箱线图，发现拱墅、西湖、萧山的四分位差值要比江干、余杭的四分位差要大，进而说明虽属同一地区，但高低价位的楼盘参差不齐，拱墅、西湖、萧山这三个地区要尤为明显。

下面我通过回归模型查看热销前五名地区售出的房屋比例与面积均价的相关性

In [43]:
```
# 通过迭代得出五个地区的房屋售出比例和面积均价
name_list = ['拱墅','西湖','江干','萧山','余杭']
proportion = []
average = []
for _ in name_list:
    out = df_sell_out[df_sell_out['location'] == _].shape[0]
    total = df_ing_ed[df_ing_ed['location'] == _].shape[0]
    avg = df_new[df_new['location'] == _]['price'].mean()
    proportion.append(out/total)
    average.append(avg)
```

In [44]:
```
# 建立DataFrame工作表
dic = {'average_price':average,
       'proportion':proportion}
avg_pro = pd.DataFrame(dic)
```

In [45]:
```
# 建立回归模型查看决定系数
avg_pro['intercept'] = 1
lm = sm.OLS(avg_pro['proportion'],avg_pro[['intercept','average_price']])
results = lm.fit()
results.summary()
```

```
D:\program\lib\site-packages\statsmodels\stats\stattools.py:72: ValueWarning: omni_normtest is not valid with less than 8 observations; 5 s
amples were given.
  "samples were given." % int(n), ValueWarning)
```

Out[45]:

OLS Regression Results

Dep. Variable:	proportion	**R-squared:**	0.664
Model:	OLS	**Adj. R-squared:**	0.553
Method:	Least Squares	**F-statistic:**	5.940
Date:	Mon, 10 Sep 2018	**Prob (F-statistic):**	0.0927
Time:	12:06:19	**Log-Likelihood:**	8.5020
No. Observations:	5	**AIC:**	-13.00
Df Residuals:	3	**BIC:**	-13.79
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
intercept	1.0581	0.113	9.353	0.003	0.698	1.418
average_price	-7.457e-06	3.06e-06	-2.437	0.093	-1.72e-05	2.28e-06

Omnibus:	nan	**Durbin-Watson:**	2.479
Prob(Omnibus):	nan	**Jarque-Bera (JB):**	0.312
Skew:	0.560	**Prob(JB):**	0.856
Kurtosis:	2.507	**Cond. No.**	1.64e+05

通过所得的回归模型，决定系数**R-squared**为**0.664**，相关性较强，可以说通过面积均价有**66.4%**可信度来说明房屋的售卖程度

猜想2的结论：

就该网站的楼盘信息，绝大多数已购房的用户衡量了房屋的价格与地理位置选择了就自身而言性价比较高的地段。

猜想2的结论主要依据：

1.就之前的饼图得出，已售罄的楼盘中余杭占**29.6%**，萧山占**21.8%**，江干占**12.1%**，西湖和拱墅分别占**7.9%**。数据表明选择余杭和萧山地区的用户要明显多于江干、西湖、拱墅的总和。且均价方面，余杭和萧山平均价位在**26000**，江干和西湖平均价位在**41000**，拱墅平均价位在**46000**。

2.刚刚得出的平均价位与售卖程度之间的决定系数约为**0.66**，表明两者相关性较强。在影响购房者的所有因素中，平均价位占到**66%**。

猜想二的误差来源：

1.地理位置，关于地理位置的重要性判断不够准确，很遗憾没能爬到各个楼盘的地铁线路与公交线路，文中所说的仅仅是依赖距离市中心的远近来说明地段。

2.相关性，把楼盘面积均价和售出的房屋比例分别作为自变量和因变量，虽得出的决定系数看似说明了模型效果，不过用这种方法来说明我们的猜想二不能完全保证它的合理性

猜想二误差的解决方案：

1.地理位置，通过寻找我所爬取的网站我发现还是可以找到每个楼盘的大致地铁线数量和公交线数量的，不过爬取的话较为复杂，必要时需要手动添加数据到原数据表。

2.相关性，据我目前所知道的验证模型的方法，可通过交叉验证来探究之前所建立的模型是否合理，如果有机会的话会单独写一篇通过交叉验证验证回归模型的文章。

项目总结：

选择杭州房价数据为分析目标出自于个人的兴趣，本文分析的较为浅显，下面我分别给我得出的结论和推断做些概况说明。

一：通过分析了已售的楼盘与所有楼盘做出比对，推断用户更加需求哪种类别的楼盘，所得结果是用户对住宅需求量非常明显，而且房产商给出的楼盘比例也更加偏重住宅。

二：通过分析了已售的楼盘与所有楼盘做出比对，推断用户更加看好哪个地段的楼盘，所得结果是绝大多数已购房的用户更加看好余杭、萧山、江干、西湖和拱墅。

三：通过热销TOP10的地区楼盘均价直方图，所有地区楼盘价格的箱线图以及相关性分析，寻找价格是否是导致用户所选择区域的主要原因，如果是，那么价格能在多大的程度上说明，所得结果说明面积均价是用户选择房屋位置的主要原因，其占所有原因的66%。之前的箱线图连带着展示了各个地区房屋价格分布的离散程度。

In [ ]: