1、数据分析项目实战分析指标

- 各城市对数据分析岗位的需求情况
- 不同细分领域对数据分析岗的需求情况
- 数据分析岗位的薪资状况
- 工作经验与薪水的关系
- 技能要求
- 公司都要求什么掌握什么技能
- 岗位的学历要求高吗

- 不同规模的企业对工资经验的要求以及提供的薪资水平

2、数据加载

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
job = pd.read_csv('./job.csv')
display(job.shape,job['city'].unique())
# 取出我们进行后续分析所需的字段
columns = ["positionName",
"companyShortName", "city", "companySize",
"education", "financeStage",
           "industryField", "salary",
"workYear","companyLabelList",
"job_detail"]
job = job[columns].drop_duplicates() # 去重
display(job.shape,job.head())
```

3、数据清洗

3.1、提取数据分析师数据

```
# 数据分析相应的岗位数量
cond = job["positionName"].str.contains("数据分析")  # 职位名中含有数据分析字眼的
# 筛选出我们想要的字段，并剔除positionName
job = job[cond]
job.reset_index(inplace=True) # 行索引 重置
display(job.shape,job.head())
```

3.2、薪资转换

招聘网站爬取下来的薪水是一个区间，这里用薪水区间的均值作为相应职位的薪水

```
# 处理过程
#1、将salary中的字符串均小写化（因为存在8k-16k和8K-16K）
#2、运用正则表达式提取出薪资区间
#3、将提取出来的数字转化为int型
#4、取区间的平均值
job["salary"] = job["salary"].str.lower()\
                .str.extract(r'(\d+)[k]-(\d+)k')\
                .applymap(lambda x:int(x))\
                .mean(axis=1)
display(job.head())
```

3.3、岗位技能

从job_detail中提取出技能要求 将技能分为以下几类：

- Python
- SQL
- Tableau
- Excel
- SPSS/SAS

处理方式： 如果job_detail中含有上述五类，则赋值为1，不含有则为0

```
job["job_detail"] =
job["job_detail"].str.lower().fillna("")  #
将字符串小写化，并将缺失值赋值为空字符串

job["Python"] =
job["job_detail"].map(lambda x:1 if
('python' in x) else 0)
job["SQL"] = job["job_detail"].map(lambda
x:1 if ('sql' in x) or ('hive' in x)  else
0)
job["Tableau"] =
job["job_detail"].map(lambda x:1 if
'tableau' in x  else 0)
job["Excel"] = job["job_detail"].map(lambda
x:1 if 'excel' in x  else 0)
job['SPSS/SAS'] =
job['job_detail'].map(lambda x:1 if ('spss'
in x) or ('sas' in x) else 0)
job.head()
```

3.4、处理行业信息

在行业信息中有多个标签，对其进行处理，筛选最显著的行业标签。

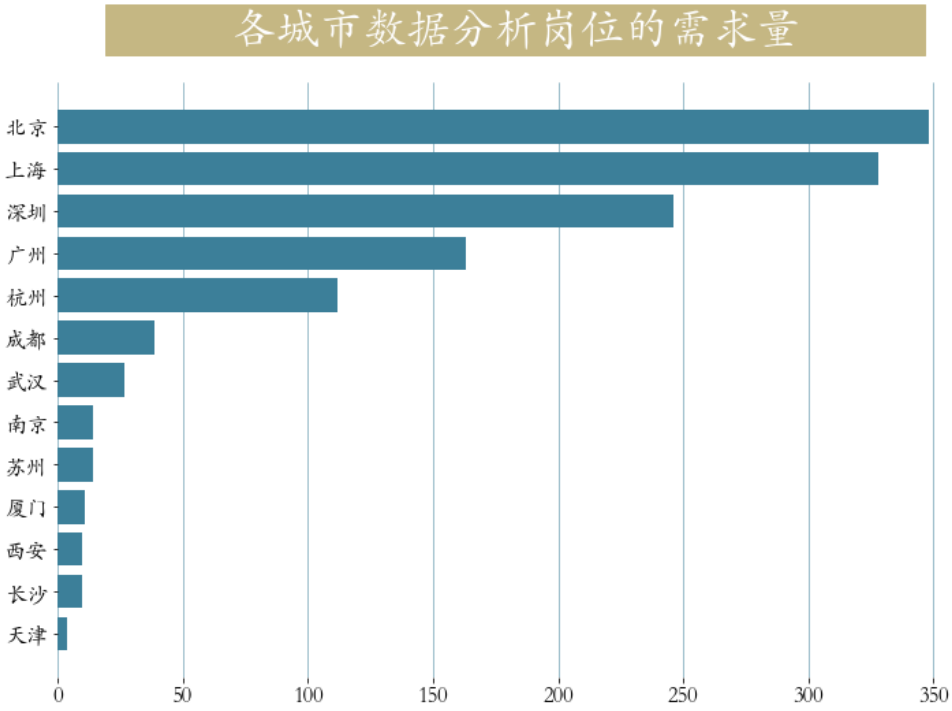
```
def clean_industry(industry):
    industry = industry.split(",")
    if industry[0]=="移动互联网" and
len(industry)>1:
        return industry[1]
    else:
        return industry[0]
job["industryField"] =
job.industryField.map(clean_industry)
job.head()
```

招聘网站数据分析师职位的数据预处理基本完成，后续使用Matplotlib进行数据可视化分析。

4、综合指标分析

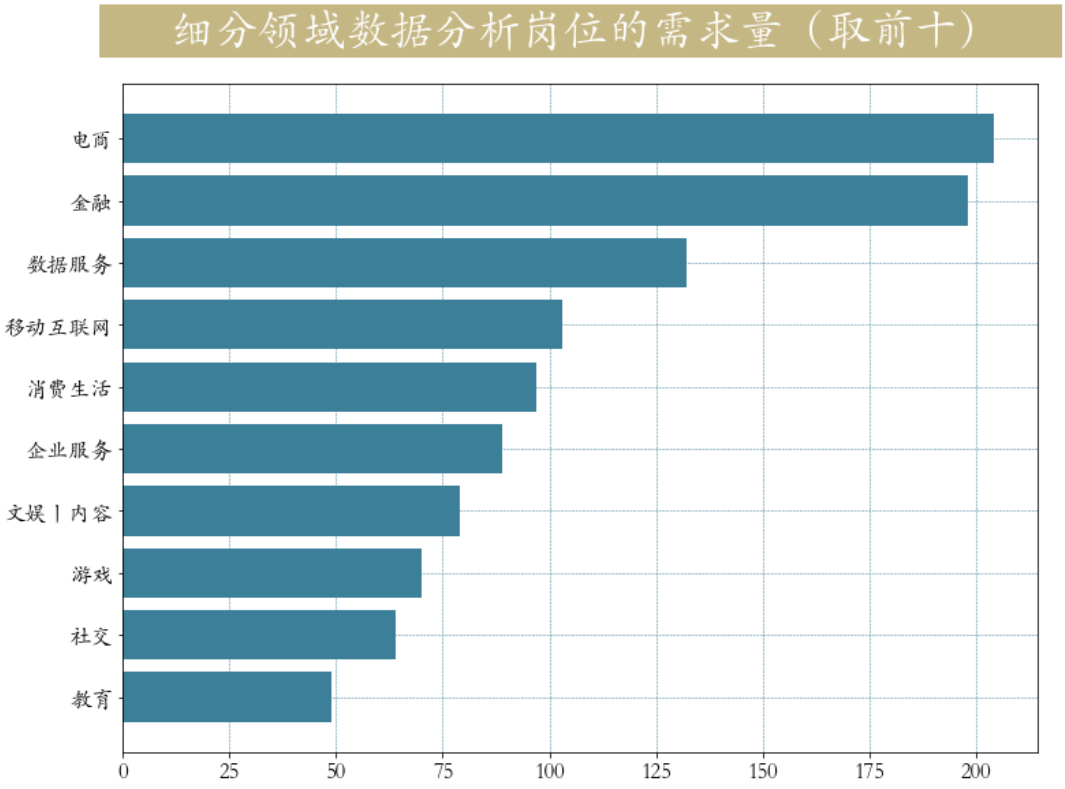
4.1、各城市对数据分析岗位的需求量

两种常用颜色：浅蓝色： #3c7f99 ， 淡黄色：#c5b783



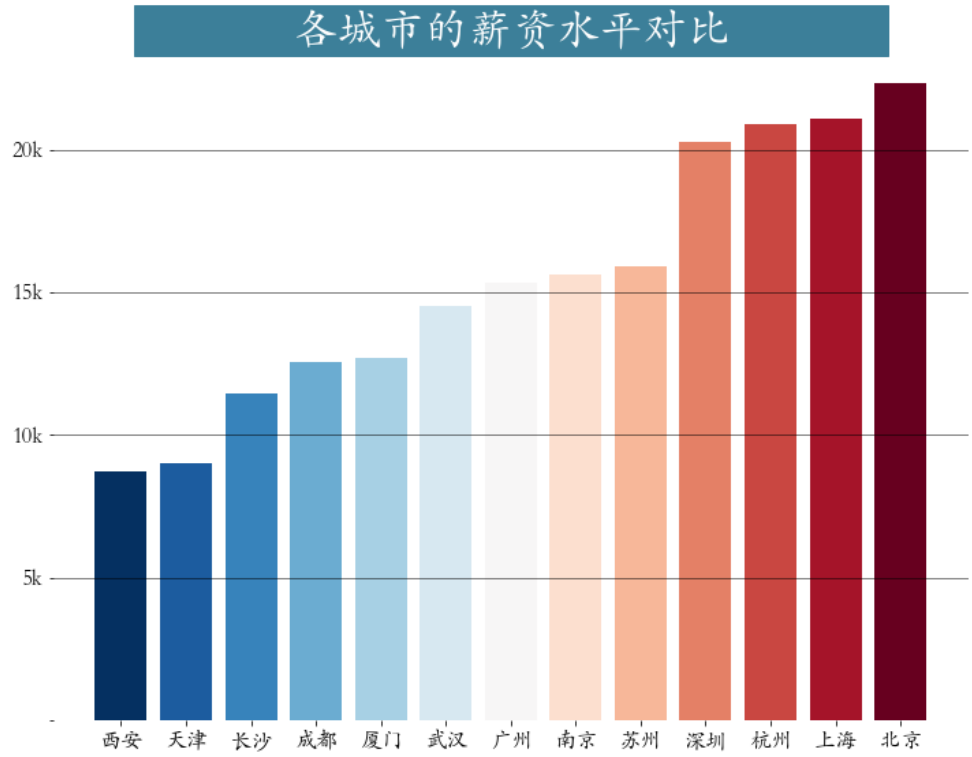

```
plt.figure(figsize=(12,9))
cities = job['city'].value_counts() # 统计城市工作数量
plt.barh(y = cities.index[::-1],
         width = cities.values[::-1],
         color = '#3c7f99')
plt.box(False) # 不显示边框
plt.title(label='                各城市数据分析岗位的需求量                ',
          fontsize=32, weight='bold', color='white',
          backgroundcolor='#c5b783',pad = 30 )
plt.tick_params(labelsize = 16)
plt.grid(axis = 'x',linewidth = 0.5,color = '#3c7f99')
```

4.2、不同领域对数据分析岗的需求量



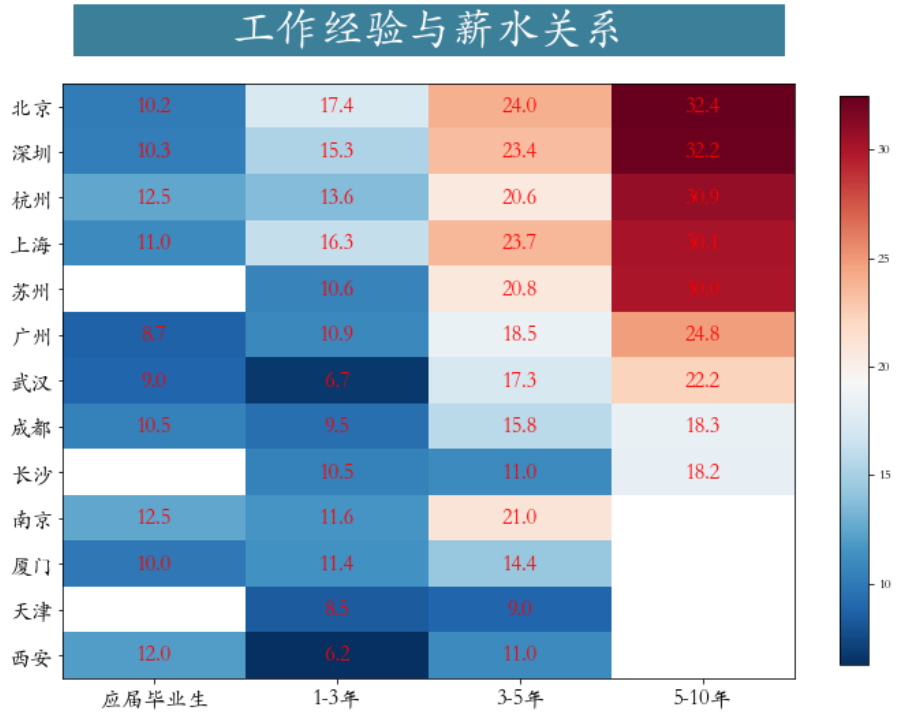
```
# 获取需求量前10多的领域
industry_index =
job["industryField"].value_counts()
[:10].index
industry
=job.loc[job["industryField"].isin(industry
_index),"industryField"]
plt.figure(figsize=(12,9))
plt.barh(y = industry_index[::-1],

width=pd.Series.value_counts(industry.value
s).values[::-1],
         color = '#3c7f99')
plt.title(label='      细分领域数据分析岗位的需
求量（取前十）      ',
          fontsize=32, weight='bold',
color='white',
          backgroundcolor='#c5b783',ha =
'center',pad = 30)
plt.tick_params(labelsize=16)
plt.grid(lw = 0.5,color = '#3c7f99',ls = '-
-')
```

4.3、各城市薪资状况


各城市的薪资水平对比
20k
15k
10k
5k
-
西安
天津
长沙
成都
厦门
武汉
广州
南京
苏州
深圳
杭州
上海
北京

```
plt.figure(figsize=(12,9))
city_salary = job.groupby("city")
["salary"].mean().sort_values() # 分组聚合运
算
plt.bar(x = city_salary.index,height =
city_salary.values,
        color =
plt.cm.RdBu_r(np.linspace(0,1,len(city_sala
ry))))
plt.title(label='                    各城市的薪资水
平对比                ',
          fontsize=32, weight='bold',
color='white', backgroundcolor='#3c7f99')
plt.tick_params(labelsize=16)
plt.grid(axis = 'y',linewidth = 0.5,color =
'black')
plt.yticks(ticks = np.arange(0,25,step =
5,),labels = ['','5k','10k','15k','20k'])
plt.box(False) # 去掉边框
```

4.4、工作经验与薪水关系

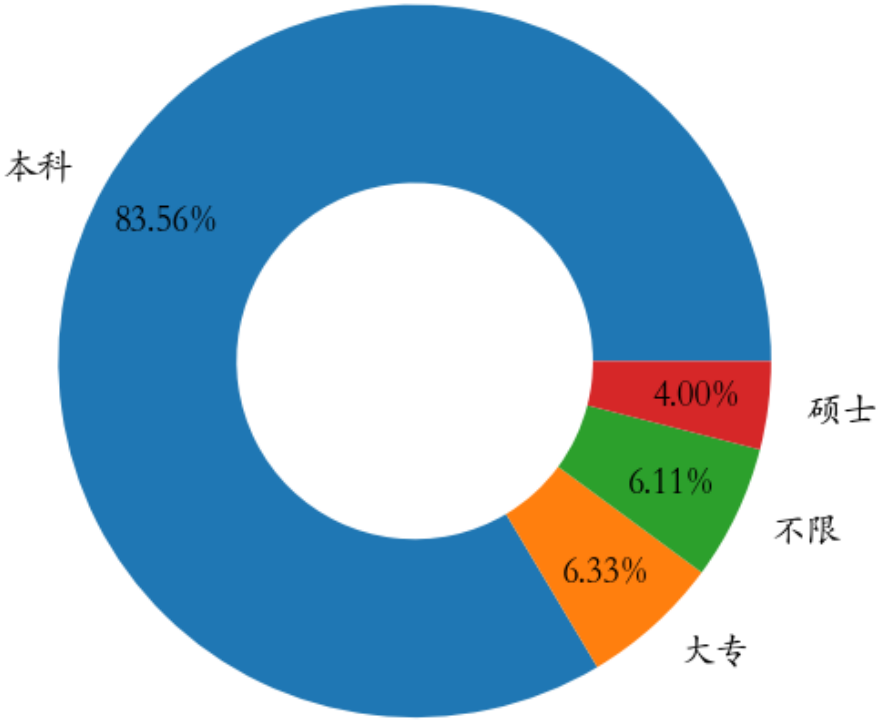



```
work_salary =
job.pivot_table(index="city",columns="workY
ear",values="salary") # 透视表
work_salary = work_salary[["应届毕业生","1-3
年","3-5年","5-10年"]]\
                .sort_values(by = '5-10
年',ascending = False) # 筛选一部分工作经验
data = work_salary.values
data = np.repeat(data,4,axis = 1) # 重复4
次，目的画图，美观，图片宽度拉大
plt.figure(figsize=(12,9))
plt.imshow(data,cmap='RdBu_r')
plt.yticks(np.arange(13),work_salary.index)
plt.xticks(np.array([1.5,5.5,9.5,13.5]),wor
k_salary.columns)
# 绘制文本
```

```
h,w = data.shape
for x in range(w):
    for y in range(h):
        if (x%4 == 0) and
(~np.isnan(data[y,x])):
            text = plt.text(x + 1.5, y,
round(data[y,x],1),
                            ha="center",
va="center", color='r',fontsize = 16)
plt.colorbar(shrink = 0.85)
plt.tick_params(labelsize = 16)
```

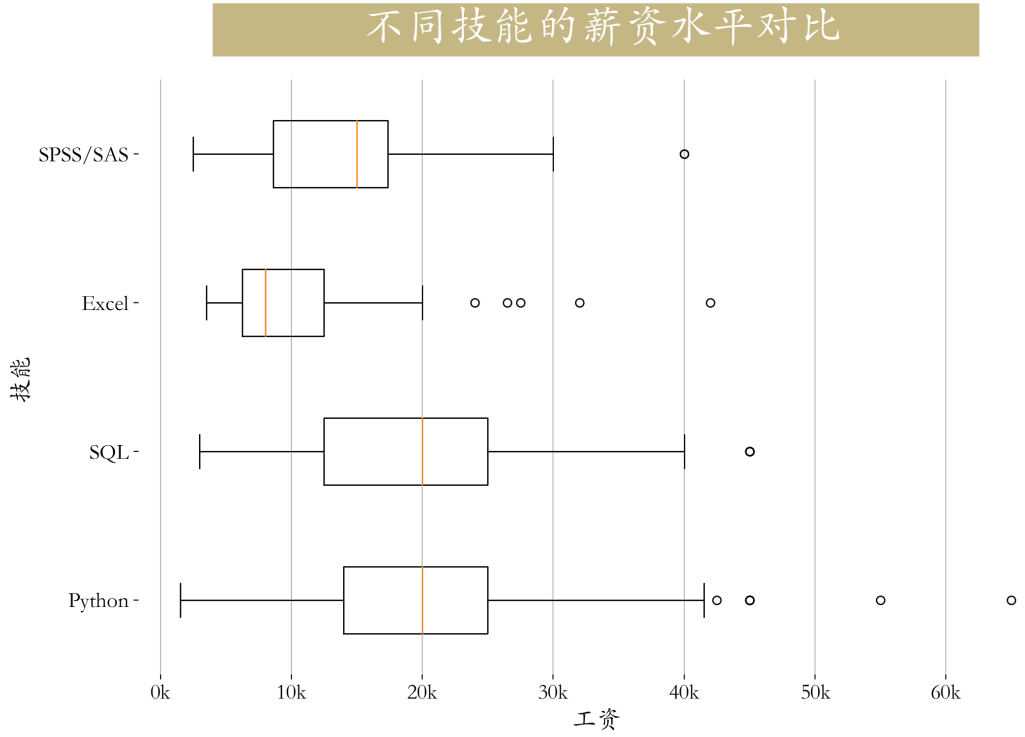
4.5、学历要求

学历要求



```
education =
job["education"].value_counts(normalize=Tru
e)
plt.figure(figsize=(9,9))
_ =
plt.pie(education,labels=education.index,au
topct='%0.2f%%',

 wedgeprops=dict(linewidth=3,width =
0.5),pctdistance=0.8,
            textprops = dict(fontsize =
20))
_ = plt.title(label='                学历要求
          ',
          fontsize=32, weight='bold',
          color='white',
backgroundcolor='#c5b783')
```

4.6、技能要求

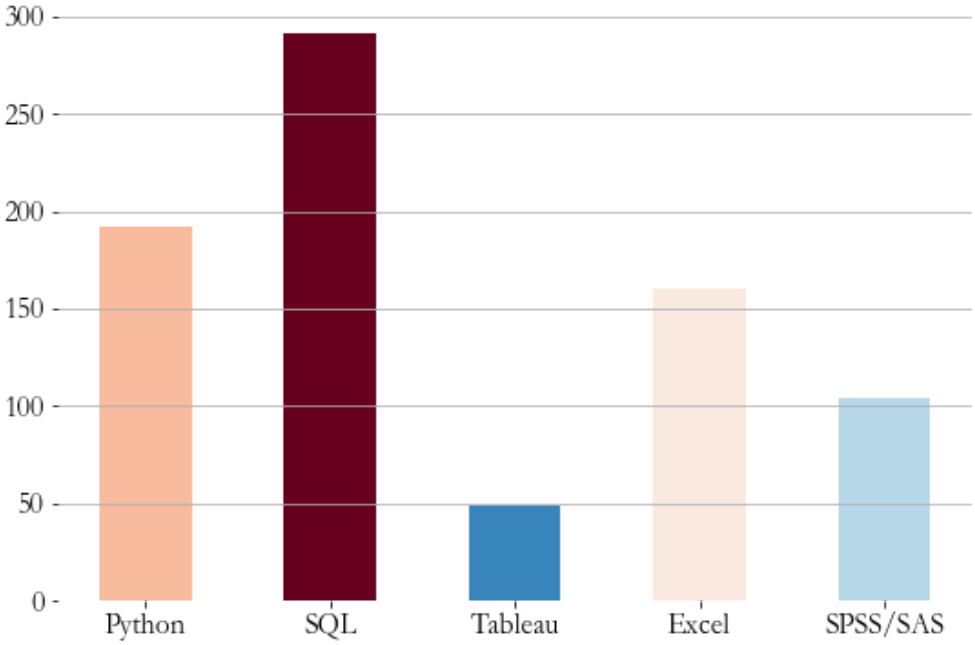



```
def get_level(x):
    if x["Python"] == 1:
        x["skill"] = "Python"
    elif x["SQL"] == 1:
        x["skill"] = "SQL"
    elif x["Excel"] == 1:
        x["skill"] = "Excel"
    elif x['SPSS/SAS'] == 1:
        x['skill'] = 'SPSS/SAS'
    else:
        x["skill"] = "其他"
    return x
job = job.apply(get_level,axis=1) # 数据转换

# 获取主要技能
x = job.loc[job.skill!='其他']
[['salary','skill']]
cond1 = x['skill'] == 'Python'
```

```
cond2 = x['skill'] =='SQL'
cond3 = x['skill'] == 'Excel'
cond4 = x['skill'] == 'SPSS/SAS'

plt.figure(figsize=(12,8))
plt.title(label='                不同技能的薪资水平对比                ',
          fontsize=32, weight='bold', color='white',
          backgroundcolor='#c5b783',pad = 30)
plt.boxplot(x = [job.loc[job.skill!='其他']['salary'][cond1],
                 job.loc[job.skill!='其他']['salary'][cond2],
                 job.loc[job.skill!='其他']['salary'][cond3],
                 job.loc[job.skill!='其他']['salary'][cond4]],
            vert = False,labels = ["Python","SQL","Excel",'SPSS/SAS'])
plt.tick_params(axis="both",labelsize=16)
plt.grid(axis = 'x',linewidth = 0.75)
plt.xticks(np.arange(0,61,10), [str(i)+"k" for i in range(0,61,10)])
plt.box(False)
plt.xlabel('工资', fontsize=18)
plt.ylabel('技能', fontsize=18)
```

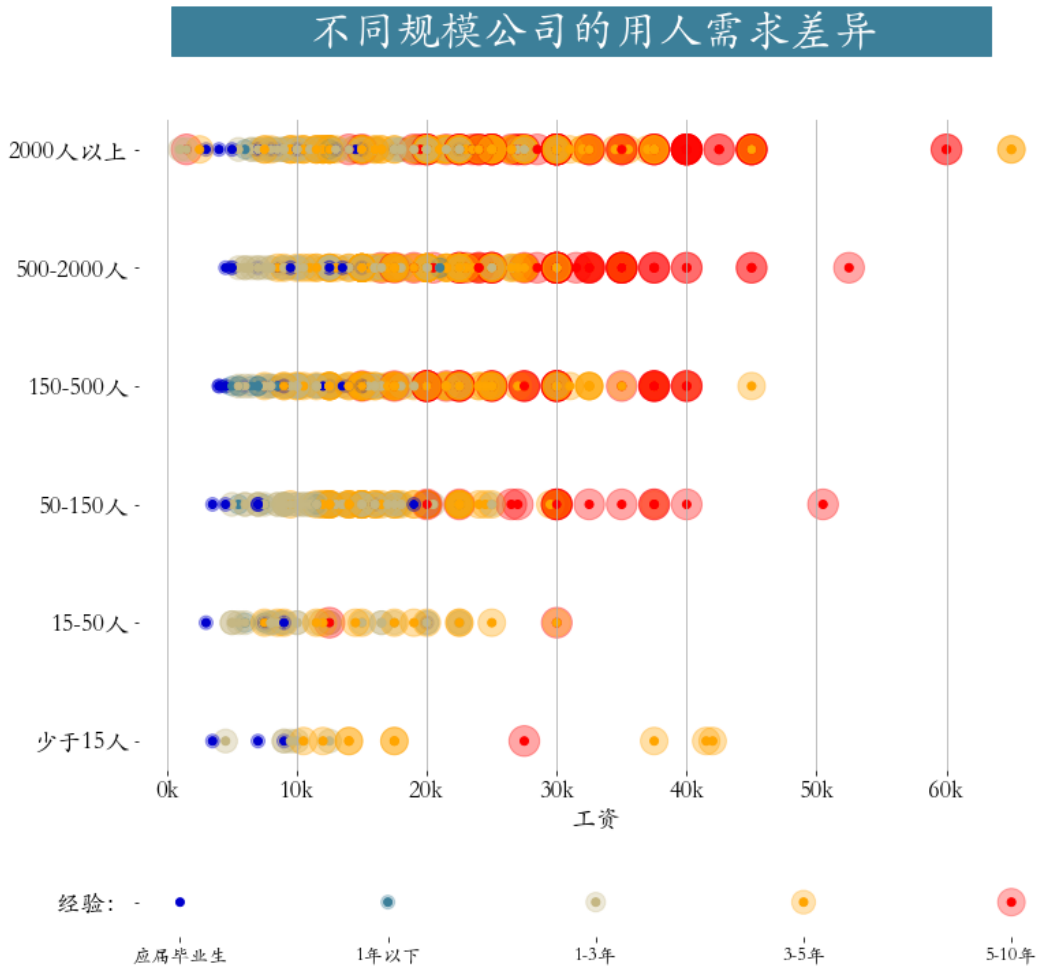
4.7、大公司对技能要求

大公司对技能的要求



```
skill_count = job[job['companySize'] ==
'2000人以上']
[['Python','SQL','Tableau','Excel','SPSS/SA
S']].sum()
plt.figure(figsize=(9,6))
plt.bar(np.arange(5),skill_count,
        tick_label =
['Python','SQL','Tableau','Excel','SPSS/SAS
'],
        width = 0.5,
        color =
plt.cm.RdBu_r(skill_count/skill_count.max()
))
_ = plt.title(label='                大公司对技
能的要求          ',
          fontsize=32, weight='bold',
color='white',
          backgroundcolor='#c5b783',pad =
30)
plt.tick_params(labelsize=16,)
plt.grid(axis = 'y')
plt.box(False)
```

4.8、不同规模的公司在招人要求上的差异




```
from matplotlib import gridspec
workYear_map = {
    "5-10年": 5,
    "3-5年": 4,
    "1-3年": 3,
    "1年以下": 2,
    "应届毕业生": 1}
color_map = {
    5:"#ff0000",
    4:"#ffa500",
    3:"#c5b783",
    2:"#3c7f99",
    1:"#0000cd"}
cond = job.workYear.isin(workYear_map)
```

```
job = job[cond]
job['workYear'] =
job.workYear.map(workYear_map)
# 根据companySize进行排序，人数从多到少
job['companySize'] =
job['companySize'].astype('category')
list_custom = ['2000人以上', '500-2000
人','150-500人','50-150人','15-50人','少于15
人']
job['companySize'] =
job['companySize'].cat.reorder_categories(l
ist_custom)
job.sort_values(by = 'companySize',inplace
= True,ascending = False)

plt.figure(figsize=(12,11))
gs = gridspec.GridSpec(10,1)
plt.subplot(gs[:8])
plt.suptitle(t='                不同规模公司的用人
需求差异              ',
         fontsize=32,
         weight='bold', color='white',
backgroundcolor='#3c7f99')
plt.scatter(job.salary,job.companySize,
            c =
job.workYear.map(color_map),
            s = (job.workYear*100),alpha =
0.35)
plt.scatter(job.salary,job.companySize,
```

```
            c =
job.workYear.map(color_map))
plt.grid(axis = 'x')
plt.xticks(np.arange(0,61,10), [str(i)+"k"
for i in range(0,61,10)])
plt.xlabel('工资', fontsize=18)
plt.box(False)
plt.tick_params(labelsize = 18)

# 绘制底部标记
plt.subplot(gs[9:])
x = np.arange(5)[::-1]
y = np.zeros(len(x))
s = x*100
plt.scatter(x,y,s=s,c=color_map.values(),al
pha=0.3)
plt.scatter(x,y,c=color_map.values())
plt.box(False)
plt.xticks(ticks=x,labels=list(workYear_map
.keys()),fontsize=14)
plt.yticks(np.arange(1),labels=['  经
验: '],fontsize=18)
```