**电子科技大学 经济管理 学院**

****

**期**

**末**

**作**

**业**

**课程名称： 商务统计与数据分析**

**学 号 ： 2020150176022**

**姓 名 ： 洪衎**

**日 期 ： 2021年3月4日**

2.1餐饮团购活动.xlsx这个数据集里记录了某团购网站上某段时间内餐饮团购活动内容的具体信息。餐饮团购店铺.xlsx这个数据集中记录了团购店铺的相关数据，可以从这个数据集了解到每家餐饮店铺的特征数据。基于这两个数据，进行相应的数据整理与清洗，试着回答以下问题：(合计 20 分)

（1） 有多少家店铺参加了团购活动？

（2） 最受消费者喜爱的Top5餐饮类型是哪些？

（3） 销售最多的店铺Top10是哪些？

（4） 团购优惠的额度是否和购买数量存在关系？什么关系？

（5） 不同行政区域，口碑良好（选择评分大于4分）的商家数量？

2.2 使用python对京东超市销售数据.csv数据集中的合适变量进行基本的描述性统计分析，主要包括数据的频数分析、数据的集中趋势分析、数据离散程度分析、数据的分布、变量间的相关性以及一些基本的统计图形比如直方图和盒状图，并对结果进行管理学意义上的讨论。（小组为单位完成；合计 20 分）；

2.3 成都市青羊区锦江区是公认的小学初中教育资源最好的行政区，因此二手房价相对较高。课程作业文件夹中包含青羊区二手房.csv格式数据文档文件一份。数据为青羊区挂牌二手房的相关数据。要求：整理与描述青羊区挂牌二手房的相关信息，建立二手房房价的预测模型，揭示影响二手房房价的主要因素并对结果进行管理学意义上的阐述（小组为单位完成；合计 20分）；

2.4 基于纽约地铁站人流量数据（Newyork.csv），使用FineBI或Python进行数据的可视化描述天气状况如何影响地铁客流？这种影响效应是否存在差异性（在不同的条件下影响效应具有差异）？输出图表结果并对结果进行较为详细的阐述（小组为单位完成；合计 20 分）

2.5 51job.csv存放了51job网站成都地区的关于数据分析相关的职位的信息，试使用python的numpy和pandas包对数据进行整理清洗与分析，回答如下问题：（个人完成；合计 20 分）

（1） 哪个区对数据分析相关的岗位需求最高？

（2） 对数据分析相关有需求的企业具有什么特征？

（3） 薪资最高的 10 家企业？

（4） 数据分析相关的职位有哪些并进行排序？

（5） 对岗位职位进行可视化词云分析，并对结果进行阐述；

## 2.1

(1)

统计餐饮团购活动.xlsx中出现的不重复的“店名”数量。

Python代码：

import pandas as pd

def question1():

'''多少家店铺参加了团购活动'''

shops = set()

data = pd.read\_excel('餐饮团购活动.xlsx', sheet\_name='Sheet1', usecols = ['店名'])

#print(data.店名)

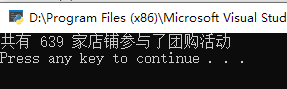
for name in data.店名:

shops.add(name)

print('共有 {} 家店铺参与了团购活动'.format(len(shops)))

question1()

运行结果：



共有639家店铺参加了活动。

(2)

Python代码：

import pandas as pd

import re

def question2():

'''最受消费者喜爱的Top5餐饮类型'''

data = pd.read\_excel('餐饮团购店铺.xlsx', sheet\_name='Sheet1', usecols = ['评价数','菜名'])

data = data.dropna(axis=0,how='any')

CanYinLeiXing = {}

for row in data.itertuples():

CaiMing\_str = row.菜名

CaiMing\_str = re.sub("】【", " ",CaiMing\_str)

CaiMing\_str = re.sub("[【】]", "",CaiMing\_str)

CaiMing\_list = CaiMing\_str.split(' ')

for CaiMing in CaiMing\_list:

if CaiMing in CanYinLeiXing:

CanYinLeiXing[CaiMing] += int(row.评价数)

else:

CanYinLeiXing[CaiMing] = int(row.评价数)

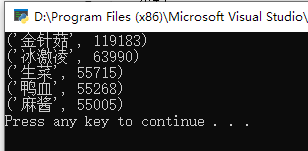
sorted\_CanYinLeiXing = sorted(CanYinLeiXing.items(),key=lambda d: d[1],reverse=True)

for i in range(5):

print(sorted\_CanYinLeiXing[i])

question2()

运行结果：



最受欢迎的Top5餐饮类型是：金针菇、冰激凌、生菜、鸭血、麻酱。

(3)

Python代码：

import pandas as pd

def question3():

'''销售最多的店铺Top10'''

data = pd.read\_excel('餐饮团购活动.xlsx', sheet\_name='Sheet1', usecols = ['店名', '购买人数'])

DianPuXiaoShou = {}

for row in data.itertuples():

if row.店名 in DianPuXiaoShou:

DianPuXiaoShou[row.店名] += row.购买人数

else:

DianPuXiaoShou[row.店名] = row.购买人数

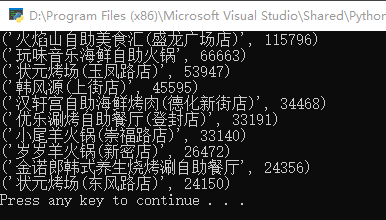
sorted\_DianPuXiaoShou = sorted(DianPuXiaoShou.items(),key=lambda d: d[1],reverse=True)

for i in range(10):

print(sorted\_DianPuXiaoShou[i])

question3()

运行结果：



销售最多的店铺Top10分别是：

火焰山自助美食汇(盛龙广场店)，

玩味音乐海鲜自助火锅，

状元烤场(玉凤路店)，

韩风源(上街店)，

汉轩宫自助海鲜烤肉(德化新街店)，

优乐涮烤自助餐厅(登封店)，

小尾羊火锅(崇福路店)，

岁岁羊火锅(新密店)、，

金诺郎韩式养生烧烤涮自助餐厅，

状元烤场(东风路店)。

(4)

Python代码：

import pandas as pd

import matplotlib.pyplot as plt

import scipy.stats as stats

def question4():

'''团购优惠的额度是否和购买数量关系'''

data = pd.read\_excel('餐饮团购活动.xlsx', sheet\_name='Sheet1', usecols = ['购买人数', '团购价','市场价'])

data = data.dropna(axis=0,how='any')

YouHuiEDu = []

for row in data.itertuples():

YouHuiEDu.append(row.市场价 - row.团购价)

data['优惠额度'] = YouHuiEDu

#print(data)

r,p = stats.pearsonr(data.购买人数,data.优惠额度)

print('相关系数r为 = {}，p值为 = {}' .format (r,p))

plt.figure(figsize = (6,6))

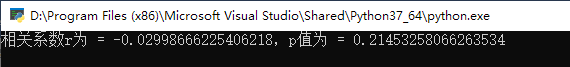
plt.scatter(data.购买人数, data.优惠额度)

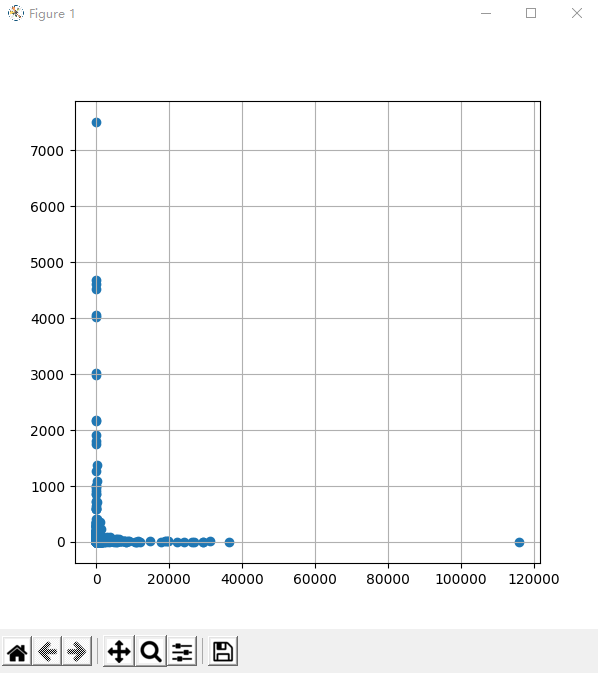
plt.grid()

plt.show()

question4()

运行结果：





相关系数约为-0.029，没有观察到显著的相关关系。

(5)

Python代码：

import pandas as pd

def question5():

'''不同行政区域，口碑良好（选择评分大于4分）的商家数量'''

data = pd.read\_excel('餐饮团购店铺.xlsx', sheet\_name='Sheet1', usecols = ['店名','评分','地址'])

data = data.dropna(axis=0,how='any')

select\_data = data[data['评分'] >= 4].copy() #筛选评分大于4的商家

ShangJiaShuLiang = {}

for row in data.itertuples():

DiZhi = row.地址

i = 0

j = 0

while i < len(DiZhi):

if DiZhi[i] == '区' or DiZhi[i] == '市' or DiZhi[i] == '县' or DiZhi[i] == '他':

j += 1

if j == 2:

XingZhengQu = DiZhi[0: (i+1)]

break

i += 1

if XingZhengQu in ShangJiaShuLiang:

ShangJiaShuLiang[XingZhengQu] += 1

else:

ShangJiaShuLiang[XingZhengQu] = 1

ShangJiaShuLiang\_se = pd.Series(ShangJiaShuLiang)

print(ShangJiaShuLiang\_se)

ShangJiaShuLiang\_se.to\_excel('不同行政区域商家数量.xlsx', header=False)

question5()

运行结果：

输出表格“不同行政区域商家数量”，结果如下：

|  |  |
| --- | --- |
| 西安市雁塔区 | 33 |
| 西安市高新区 | 13 |
| 西安市碑林区 | 29 |
| 西安市未央区 | 30 |
| 西安市莲湖区 | 26 |
| 西安市长安区 | 9 |
| 西安市新城区 | 12 |
| 西安市阎良区 | 12 |
| 西安市灞桥区 | 7 |
| 西安市户县 | 6 |
| 西安市高陵县 | 3 |
| 西安市临潼区 | 1 |
| 西安市周至县 | 4 |
| 西安市蓝田县 | 1 |
| 郑州市巩义市 | 59 |
| 郑州市新密市 | 42 |
| 郑州市登封市 | 26 |
| 郑州市二七区 | 38 |
| 郑州市金水区 | 106 |
| 郑州市新郑市 | 57 |
| 郑州市中原区 | 38 |
| 郑州市管城区 | 41 |
| 郑州市上街区 | 26 |
| 郑州市荥阳市 | 23 |
| 郑州市中牟县 | 33 |
| 郑州市其他 | 4 |
| 郑州市惠济区 | 16 |

## 2.2

import re

import numpy as np

import pandas as pd

import seaborn as sns

from matplotlib import pyplot as plt

sns.set(font="simsun",font\_scale=1.5)

get\_ipython().run\_line\_magic('matplotlib', 'inline')

pd.set\_option("max\_rows",None)

pd.set\_option("max\_columns",None)

# # 京东超市销售数据分析

jd = pd.read\_csv("data/京东超市销售数据.csv")

jd.head()

jd[["每笔订单利润", "销售额预测", "每名客户销售额", "利润","销售额"]] = jd[["每笔订单利润", "销售额预测",

"每名客户销售额", "利润","销售额"]].applymap(lambda x: x.replace(",", "")).astype("float")

jd["利润率"] = jd.利润/jd.每名客户销售额

jd.describe()

jd.corr()

pd.pivot\_table(jd,index=["地区"],values=["销售额","利润率"],aggfunc={"销售额":[np.max,np.sum,np.mean],"利润率":[np.sum,np.mean]})

plt.figure(figsize=(15, 20))

plt.subplot(211)

sns.stripplot(x="地区", y="销售额",

data=jd, jitter=True,

hue="装运状态", dodge=True,

palette="Blues")

plt.subplot(212)

sns.violinplot( x=jd["地区"], y=jd["销售额"],palette="Blues" )

plt.show()

sns.pairplot(jd)

plt.figure(figsize=(40, 40),dpi=150)

sns.catplot(x="类别",col="细分",

data=jd, kind="count",

height=8, aspect=.7,palette="Set2")



表2.2.1 数值变量的集中趋势和离散趋势



表2.2.2 变量间的相关系数



表2.2.3 利润率和销售额在不同地区的分布情况



图2.2.1 数值变量间的散点图

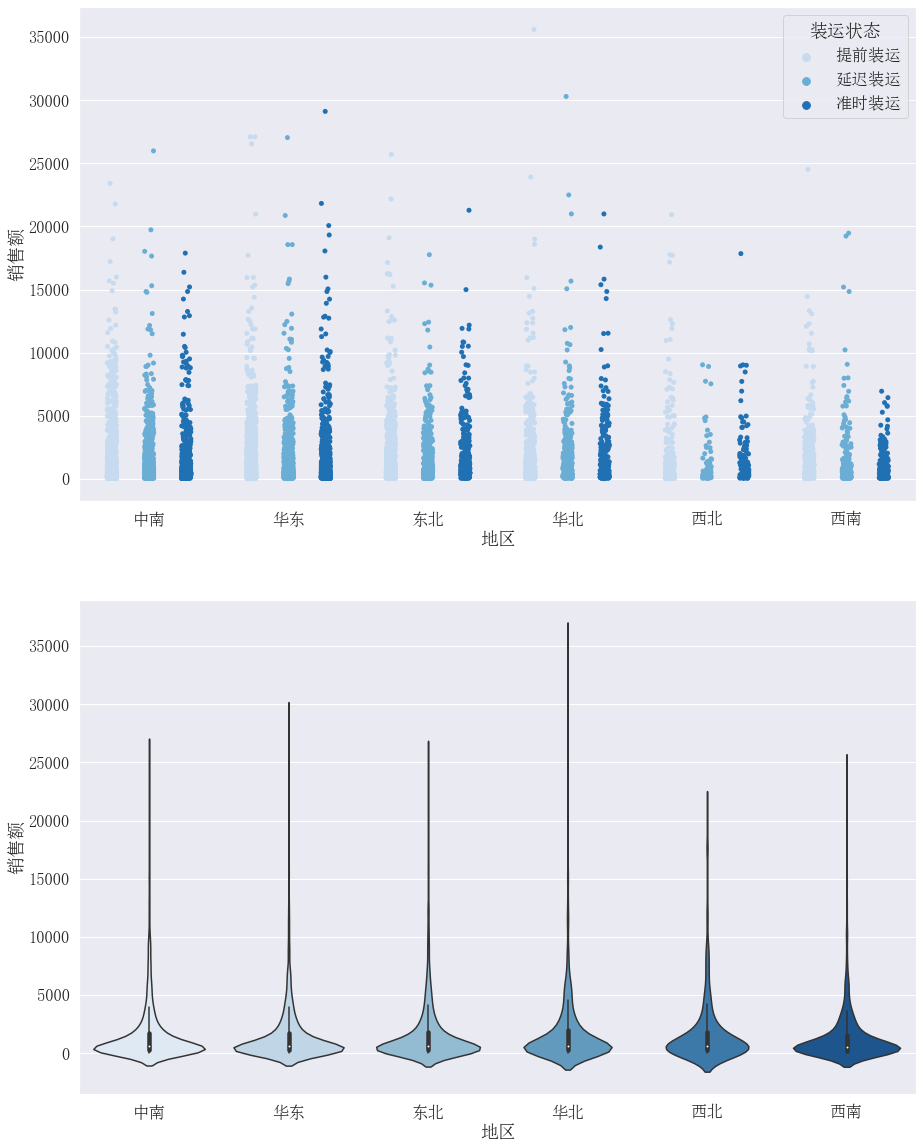


图 2.2.2 不同地区销售额的散点图（不同装运状态）和小提请图

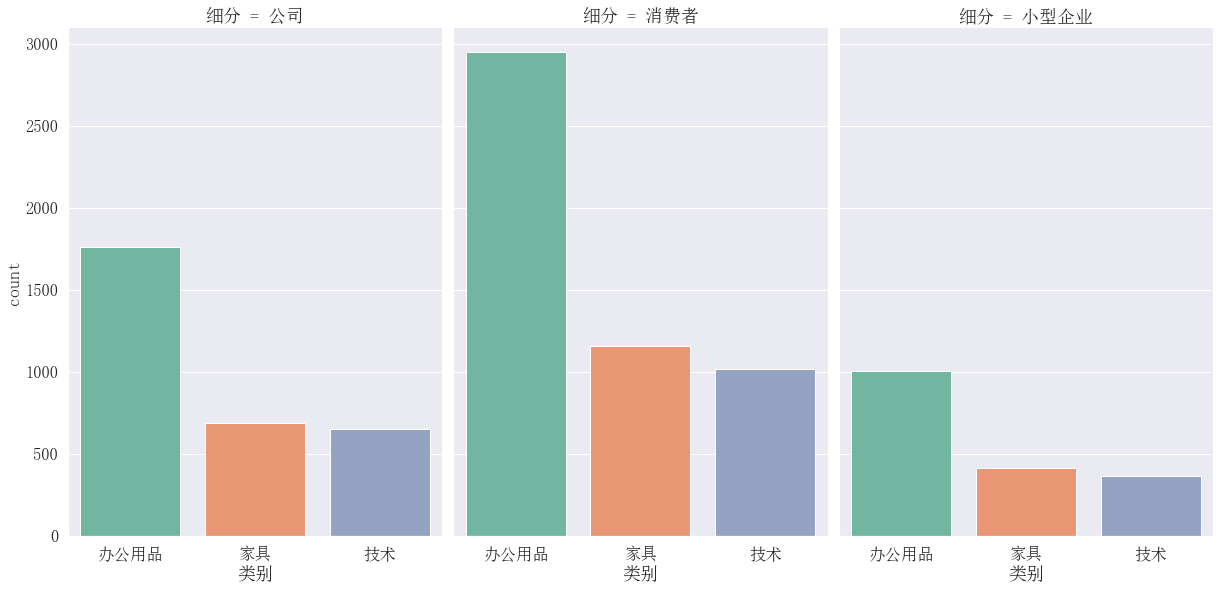


图 2.2.3 三类客户对不同类产品的订单量

## 2.3

# # 青羊区二手房价分析

# ## 数据读取

qyq = pd.read\_csv("data/青羊区二手房.csv",names=["title","room","area","height","dic","year","addr","cost","price"])

# 删去空行

qyq.dropna(axis=0, how='any', inplace=True)

# 去除换行符和制表符

qyq.title = qyq.title.apply(lambda x: x.replace("\t","").replace("\n",""))

qyq.head()

# qyq.room.apply(lambda x:'10室10厅'if x=="独栋" else x)

qyq.room = qyq.room.apply(lambda x:[x, "d室d厅"][x=="独栋"])

qyq.room = qyq.room.apply(lambda x:[x, "l室l厅"][x=="联排"])

qyq.room = qyq.room.apply(lambda x:[x, "s室s厅"][x=="双拼"])

qyq.room = qyq.room.apply(lambda x:[x, "j室j厅"][x=="叠加"])

qyq["shi"] = qyq.room.apply(lambda x: x[0])

qyq["ting"] = qyq.room.apply(lambda x: x[2])

# - 研究非别墅区

# - 提取数字

# - 处理定性变量

fbs = qyq.sort\_values(by="area", axis=0, ascending=True, inplace=False, kind='quicksort', na\_position='last').head(-103)

fbs["shi"] = fbs.room.apply(lambda x: int(x[0]))

fbs["ting"] = fbs.room.apply(lambda x: int(x[2]))

fbs["area"] = fbs.area.apply(lambda x: x.replace("㎡","")).astype("float")

fbs["layers"] = fbs.height.apply(lambda x: int(x.split("（共")[-1][:-2]))

fbs["height"] = fbs.height.apply(lambda x: x[0])

fbs["year"] = fbs.year.apply(lambda x: int(re.findall(r"\d+\.?\d\*",x)[0]))

fbs["addr\_1"] = fbs.addr.apply(lambda x: x.split("-")[0])

fbs["addr\_2"] = fbs.addr.apply(lambda x: x.split("-")[-1])

fbs["cost"] = fbs.cost.astype("float")

fbs["price"] = fbs.price.apply(lambda x: int(re.findall(r"\d+\.?\d\*",x)[0]))

fbs["price\_cor"] = fbs.cost/fbs.area\*10000

# 采用四分点和中点来映射层高

height\_state = {'高': 0.75,

'中': 0.5,

'低': 0.25

}

fbs["height\_state"] = fbs.height.map(height\_state)\*fbs.layers

# 独热编码

fbs = fbs.join([pd.get\_dummies(fbs.height),pd.get\_dummies(fbs.dic),pd.get\_dummies(fbs.addr\_1)])

fbs.head(1)

# ## 相关系数

fbs.corr().dropna(thresh=1).dropna(axis=1,thresh=1).sort\_values(by="price\_cor")

# ## 数据集划分与标准化

from sklearn.preprocessing import StandardScaler, MinMaxScaler

from sklearn.model\_selection import GridSearchCV, train\_test\_split

# from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score, recall\_score, f1\_score, classification\_report,roc\_auc\_score,roc\_curve

from sklearn.metrics import\*

scaler = StandardScaler()

mms = MinMaxScaler()

import random

fbs = fbs.sort\_values(by="price\_cor").reset\_index(drop=True)

# 注意不能保留总售价，会出现完全共线性

x = scaler.fit\_transform(fbs.loc[:,("area","year","shi","ting","layers","height\_state","中","低","高","东北向","东南向","东向","东西向","北向","南北向","南向","西北向","西南向","西向","万家湾","光华","八宝街","内光华","内金沙","外光华","外金沙","天府广场","府南新区","杜甫草堂","浣花小区","石人小区","苏坡","草市街","西南财大","贝森","长顺街","顺城街","骡马市")])

y = fbs.loc[:,"price\_cor"]

# 划分训练集与测试集；测试集选取全部数据的20%

train = []

test = []

for i in range(fbs.shape[0]):

if random.random()>0.2:

train.append(i)

else:

test.append(i)

x\_test = x[test]

x\_train = x[train]

y\_test = y[test]

y\_train = y[train]

# ## 多元线性回归

from sklearn.linear\_model import LinearRegression

linreg = LinearRegression()

lin\_fbs = linreg.fit(x\_train, y\_train)

y\_lin = lin\_fbs.predict(x\_test)

y\_hat\_2 = lin\_fbs.predict(x\_train)

fig=plt.figure(figsize=(16,16),dpi=150)

plt.subplots\_adjust(wspace =0.1, hspace =0.3)

x\_label = np.arange(1,len(y\_test))

plt.subplot(211)

plt.plot(x\_label,y\_test[:-1],linestyle='--',linewidth=2,label='y\_test 实际')

plt.plot(x\_label,y\_lin[:-1],linestyle='--',linewidth=1,label='y\_test 预测')

plt.title('Prediction\n $Price$')

plt.legend()

plt.subplot(212)

plt.plot(x\_label,y\_test[:-1]-y\_lin[:-1],linestyle='--',linewidth=2,label='y\_test 实际')

plt.title('Prediction\n $e$')

plt.legend()

# 回测

R\_square\_3 = r2\_score(y\_train,y\_hat\_2)

EVS\_3 = explained\_variance\_score(y\_train,y\_hat\_2)

MSE\_3 = mean\_squared\_error(y\_train,y\_hat\_2)

# 预测

R\_square\_4 = r2\_score(y\_test,y\_lin)

EVS\_4 = explained\_variance\_score(y\_test,y\_lin)

MSE\_4 = mean\_squared\_error(y\_test,y\_lin)

title = ["area","year","shi","ting","layers","height\_state","中","低","高","东北向","东南向","东向","东西向","北向","南北向","南向","西北向","西南向","西向","万家湾","光华","八宝街","内光华","内金沙","外光华","外金沙","天府广场","府南新区","杜甫草堂","浣花小区","石人小区","苏坡","草市街","西南财大","贝森","长顺街","顺城街","骡马市"]

coef = lin\_fbs.coef\_.tolist()

c = {

"title":title,

"coef":coef

}

pd.DataFrame(c).sort\_values(by="coef",ascending=False).reset\_index(drop=True)

# ## 支持向量机

from sklearn.svm import SVR

svr\_rbf = SVR(kernel='rbf', C=100)

svr\_rbf.fit(x\_train,y\_train)

y\_svr = svr\_rbf.predict(x\_test)

y\_hat\_1 = svr\_rbf.predict(x\_train)

fig=plt.figure(figsize=(16,16),dpi=150)

plt.subplots\_adjust(wspace =0.1, hspace =0.3)

x\_label = np.arange(1,len(y\_test))

plt.subplot(211)

plt.plot(x\_label,y\_test[:-1],linestyle='--',linewidth=2,label='y\_test 实际')

plt.plot(x\_label,y\_svr[:-1],linestyle='--',linewidth=1,label='y\_test 预测')

plt.title('Prediction\n $Price$')

plt.legend()

plt.subplot(212)

plt.plot(x\_label,y\_test[:-1]-y\_svr[:-1],linestyle='--',linewidth=2,label='y\_test 实际')

plt.title('Prediction\n $e$')

plt.legend()

# 回测

R\_square\_1 = r2\_score(y\_train,y\_hat\_1)

EVS\_1 = explained\_variance\_score(y\_train,y\_hat\_1)

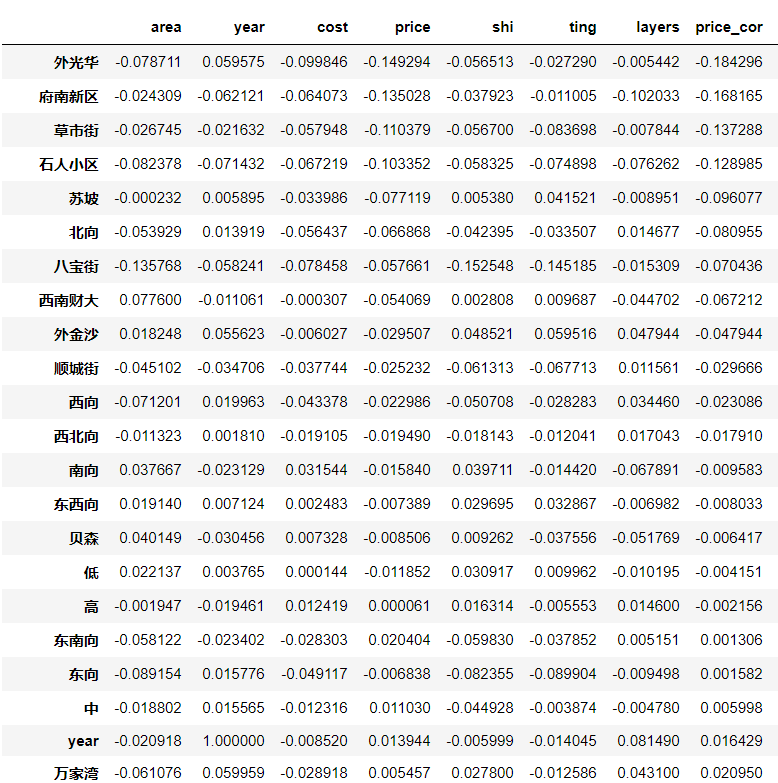
MSE\_1 = mean\_squared\_error(y\_train,y\_hat\_1)

# 预测

R\_square\_2 = r2\_score(y\_test,y\_svr)

EVS\_2 = explained\_variance\_score(y\_test,y\_svr)

MSE\_2 = mean\_squared\_error(y\_test,y\_svr)



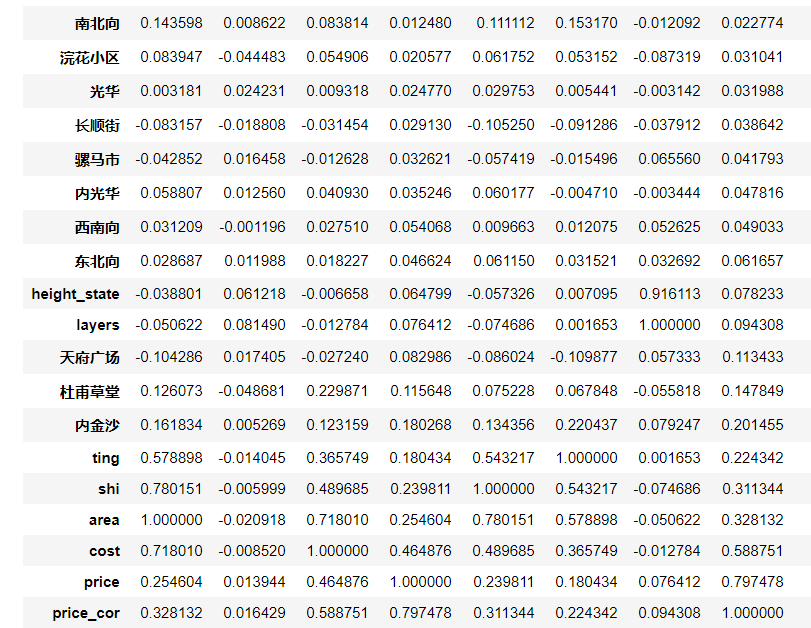


表2.3.1 变量间的相关系数矩阵



表2.3.2 支持向量机回测与预测效果

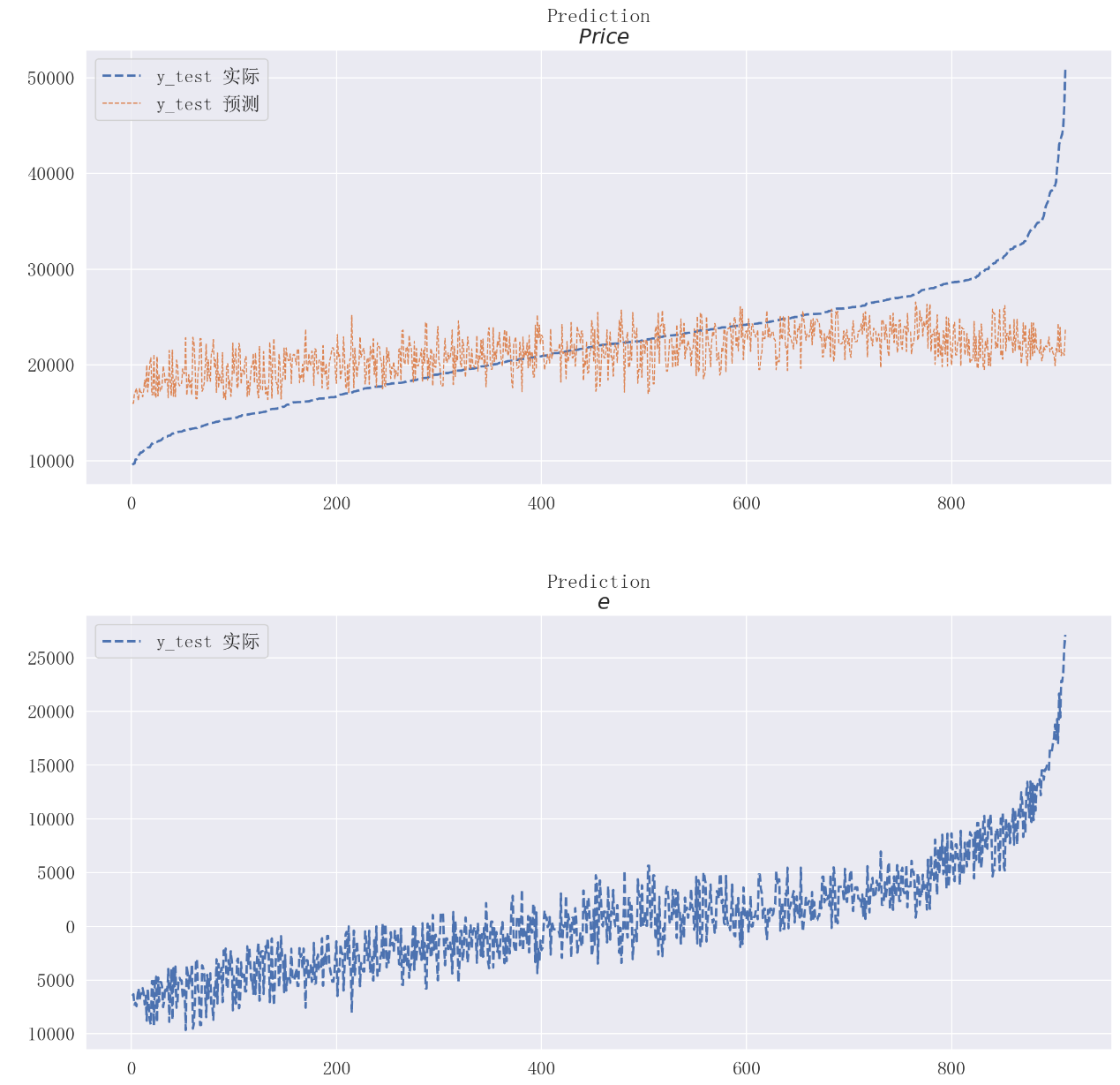


图 2,3.1 支持向量机回归结果（预测值与残差）



表2.3.3 多元线性回归模型回测与预测效果

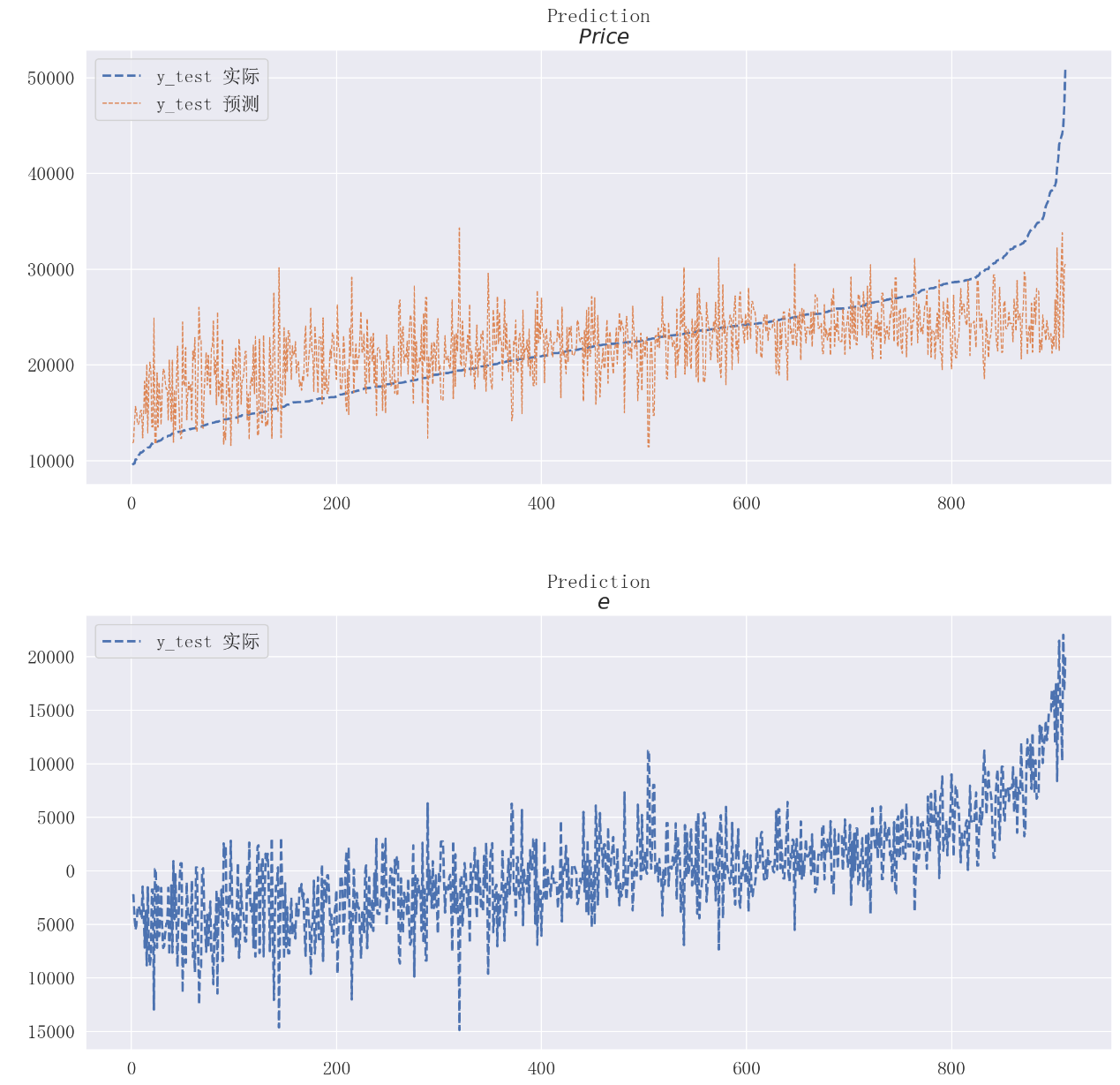


图 2,3.2 多元线性回归结果（预测值与残差）

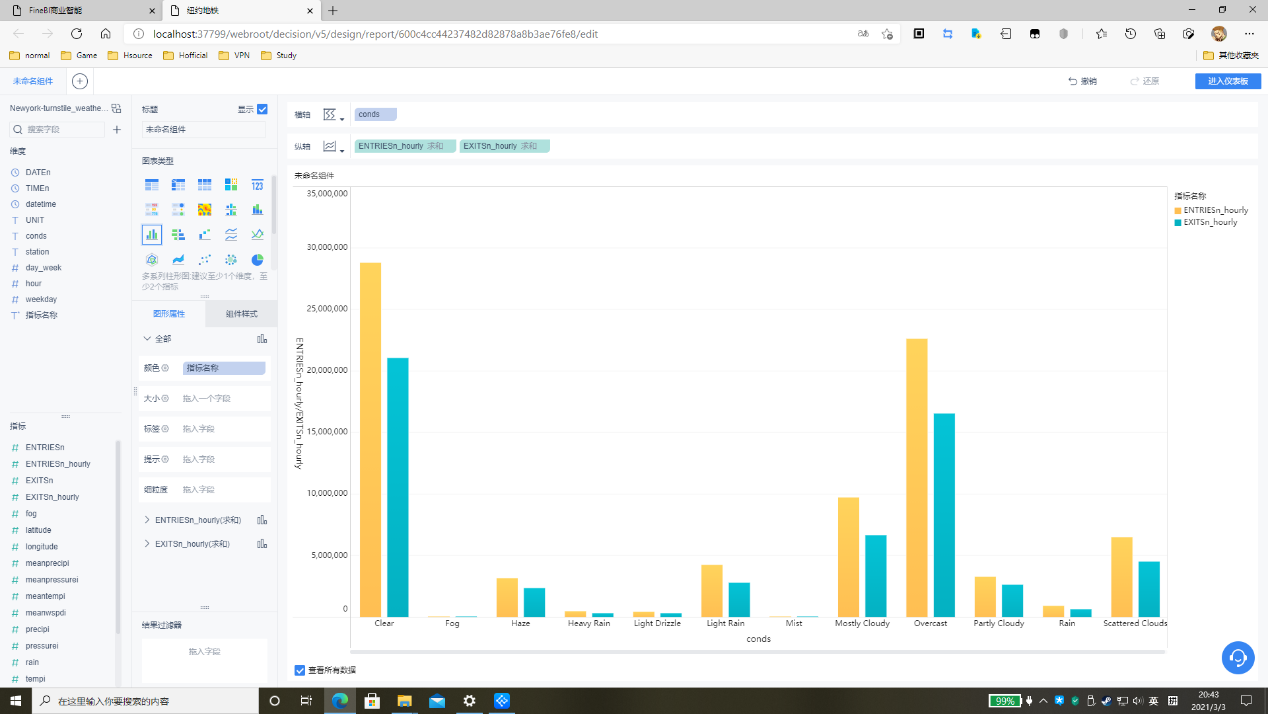




表2.3.4 多元线性回归系数列表

## 2.4

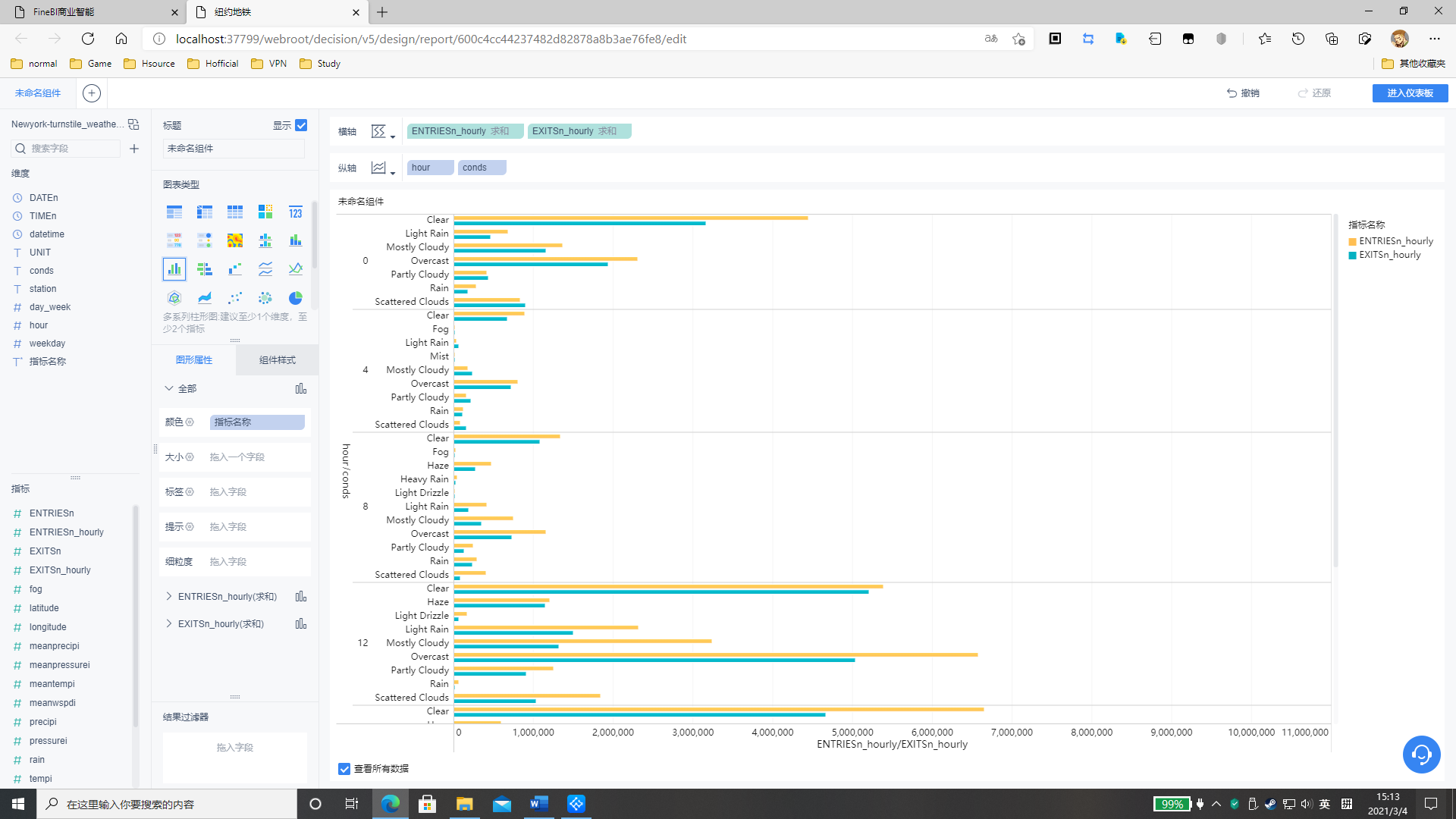
使用FineBi进行分析，总体而言，地铁的人流量在天气好时，如晴天（clear）、阴天（overcast）较多，而雾（mist）、雨（rain）、暴雨（heavy rain）等恶劣天气下的人流量较少。

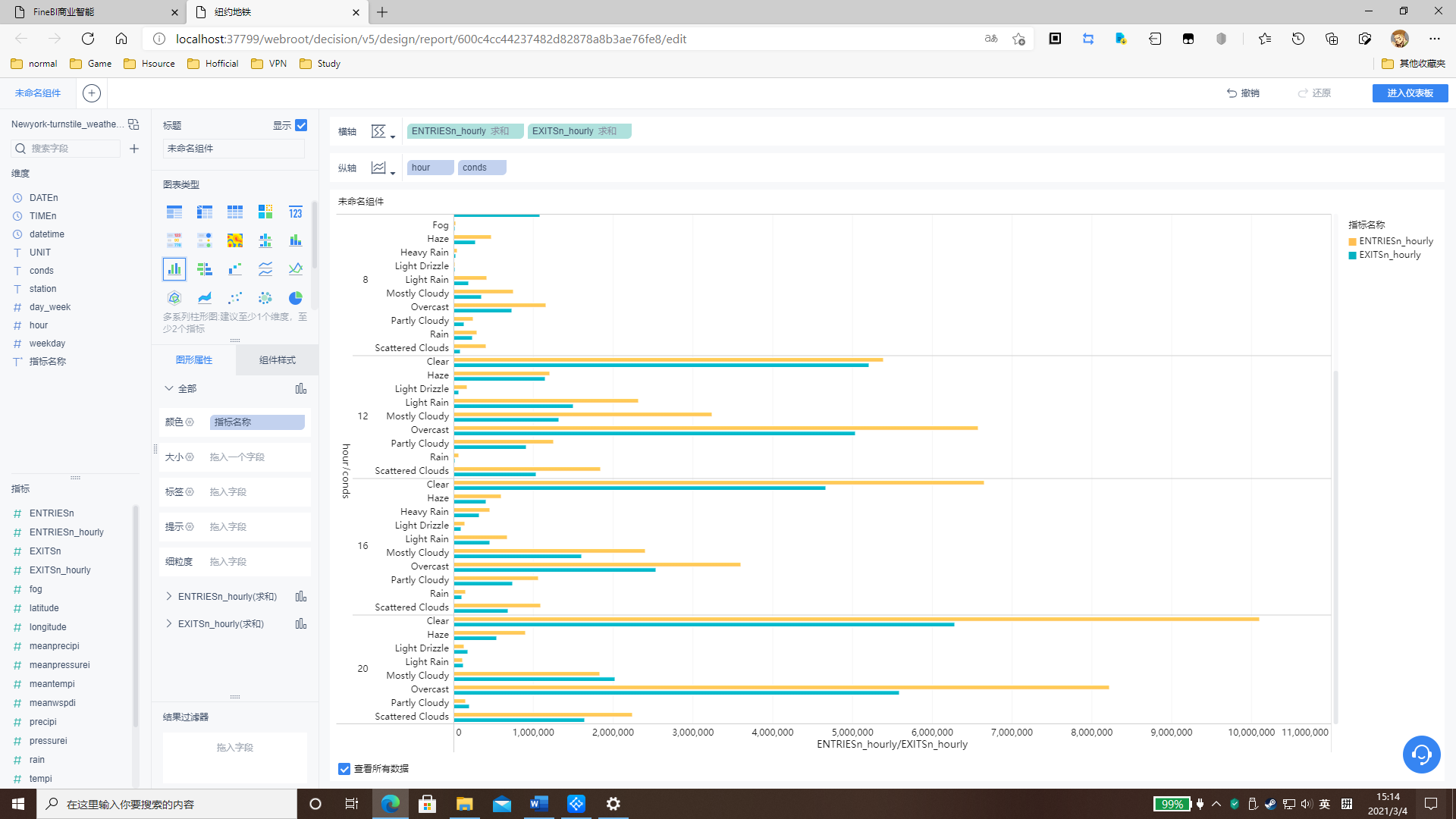


一天内不同时间点不同天气对客流量的影响如下两图所示

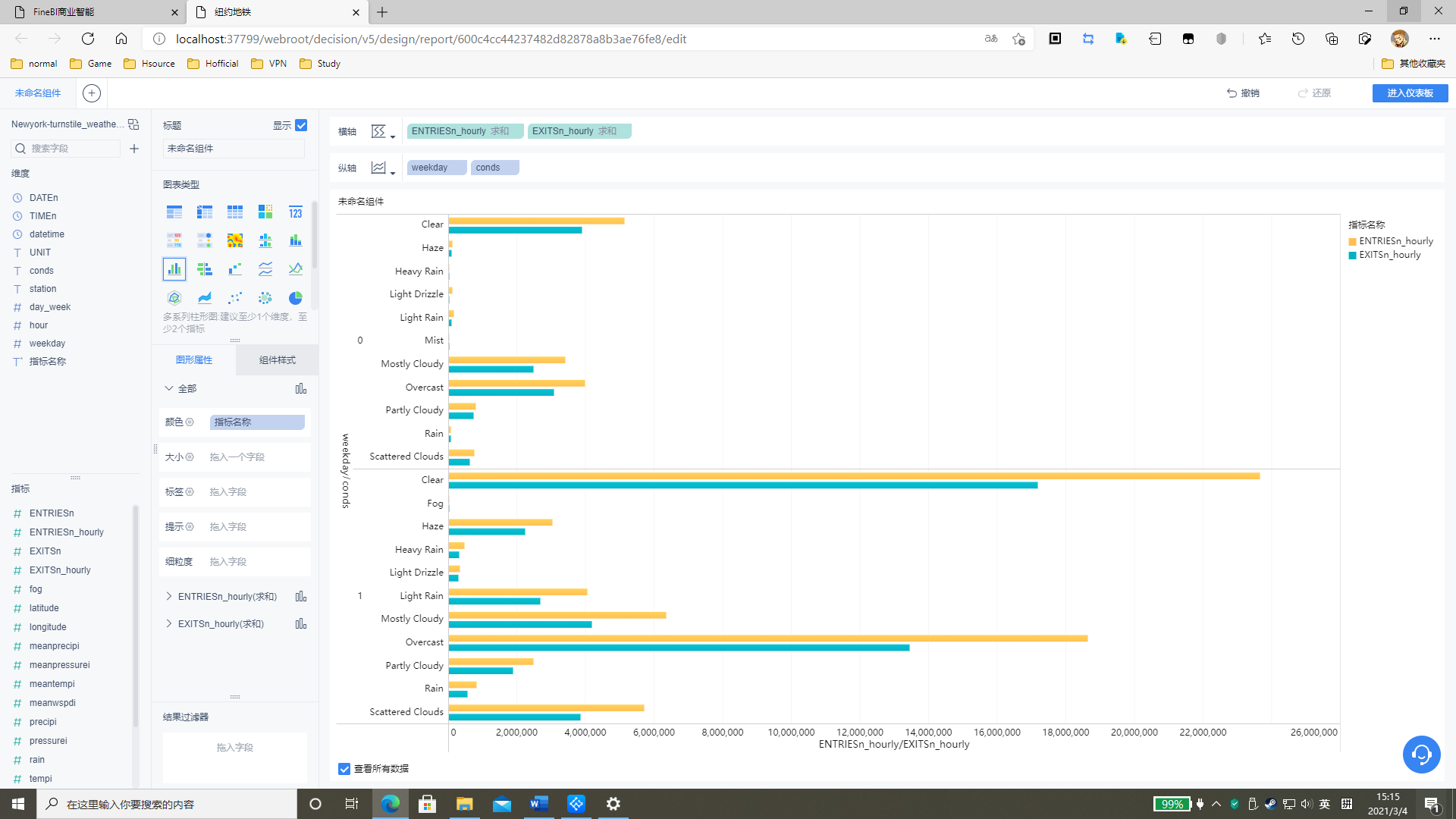
可以看出在，而无论什么时候clear、ovecast等好天气的人流量都较大。

在4点、8点、20点时，light rain，fog，mist天气会极大幅度的减少人流量，而0点、12点、16点时以上几个天气对人流量的减少相对其他时间点较少。





按工作日、非工作日分组，可以看出非工作日只要天气恶劣，几乎就不会有任何人进出地铁，而工作日虽然恶劣天气下人流量减少，但远多于非工作日。



## 2.5

(1)

Python代码：

import re

import pandas as pd

def question1():

'''哪个区对数据分析相关的岗位需求最高'''

data = pd.read\_excel('51job.xlsx', sheet\_name = '51job', header = None)

selected\_data = data[data[1].str.contains('数据分析')].copy() #筛选数据分析相关岗位

XuQiuRenShu\_list = [] #需求人数

for row in selected\_data.itertuples():

strlist = row[8].split('-')

if len(strlist) == 1:

strlist.append(strlist[0])

ZuiShaoRenShu = int(re.sub("[^0-9\.]", "", strlist[0]))

ZuiDuoRenShu = int(re.sub("[^0-9\.]", "", strlist[1]))

XuQiuRenShu = (ZuiShaoRenShu + ZuiDuoRenShu) // 2

XuQiuRenShu\_list.append(XuQiuRenShu)

selected\_data[10] = XuQiuRenShu\_list

QuYuXuQiu = {}

for row in selected\_data.itertuples():

if row[4] in QuYuXuQiu:

QuYuXuQiu[row[4]] += row[11]

else:

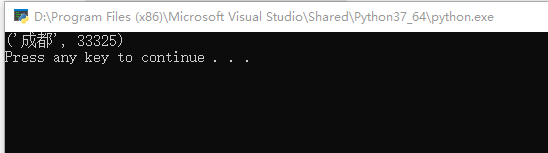
QuYuXuQiu[row[4]] = row[11]

sorted\_QuYuXuQiu = sorted(QuYuXuQiu.items(),key=lambda d: d[1],reverse=True)

print(sorted\_QuYuXuQiu[0])

question1()

运行结果截图：



对数据分析岗位需求最高的是成都

(2)

对数据分析相关企业进行分组统计

Python代码：

import pandas as pd

def question2():

'''对数据分析相关有需求的企业具有什么特征'''

data = pd.read\_excel('51job.xlsx', sheet\_name = '51job', header = None)

selected\_data = data[data[1].str.contains('数据分析')]

group\_index = [3,4,5,6,7,8]

groups=[]

for i in group\_index:

group=selected\_data.groupby(by=[i])

groups.append(group.size())

df = pd.concat(groups,axis = 1)

df.to\_excel('group.xlsx')

question2()

运行结果：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **北京-海淀区** | 1 |  |  |  |  |
| **成都** | 14 |  |  |  |  |
| **成都-双流区** | 2 |  |  |  |  |
| **成都-成华区** | 1 |  |  |  |  |
| **成都-新都区** | 1 |  |  |  |  |
| **成都-武侯区** | 12 |  |  |  |  |
| **成都-温江区** | 1 |  |  |  |  |
| **成都-金牛区** | 3 |  |  |  |  |
| **成都-锦江区** | 1 |  |  |  |  |
| **成都-青羊区** | 2 |  |  |  |  |
| **成都-高新区** | 16 |  |  |  |  |
| **深圳-南山区** | 1 |  |  |  |  |
| **1年经验** |  | 14 |  |  |  |
| **2年经验** |  | 12 |  |  |  |
| **3-4年经验** |  | 11 |  |  |  |
| **5-7年经验** |  | 4 |  |  |  |
| **8-9年经验** |  | 1 |  |  |  |
| **在校生/应届生** |  | 7 |  |  |  |
| **无需经验** |  | 6 |  |  |  |
| **大专** |  |  | 21 |  |  |
| **招10人** |  |  | 1 |  |  |
| **招1人** |  |  | 1 |  |  |
| **本科** |  |  | 31 |  |  |
| **硕士** |  |  | 1 |  |  |
| **上市公司** |  |  |  | 1 |  |
| **合资** |  |  |  | 3 |  |
| **国企** |  |  |  | 5 |  |
| **民营公司** |  |  |  | 46 |  |
| **1000-5000人** |  |  |  |  | 5 |
| **10000人以上** |  |  |  |  | 4 |
| **150-500人** |  |  |  |  | 8 |
| **50-150人** |  |  |  |  | 16 |
| **500-1000人** |  |  |  |  | 6 |
| **5000-10000人** |  |  |  |  | 4 |
| **少于50人** |  |  |  |  | 12 |

可以看出数据分析相关企业：

1. 大多在成都市区、武侯区、高新区
2. 通常要求有工作经验，但不要求很长的时间
3. 最低学历要求以本科、大专为主
4. 主要是民营公司招收该岗位
5. 招收人数普遍少于10000人

(3)

Python代码：

import re

import pandas as pd

def question3():

'''薪资最高的 10 家企业？'''

data = pd.read\_excel('51job.xlsx', sheet\_name = '51job', header = None)

ZuiDiGongZi\_list = [] #最低工资(按月计)

ZuiGaoGongZi\_list = [] #最高工资(按月计)

for row in data.itertuples():

strlist = row[3].split('-')

if len(strlist) == 1:

strlist.append(strlist[0])

if '千' in strlist[1]:

ZuiDiGongZi = float(re.sub("[^0-9\.]", "", strlist[0]))\*1000

ZuiGaoGongZi = float(re.sub("[^0-9\.]", "", strlist[1]))\*1000

elif '万' in strlist[1]:

ZuiDiGongZi = float(re.sub("[^0-9\.]", "", strlist[0]))\*10000

ZuiGaoGongZi = float(re.sub("[^0-9\.]", "", strlist[1]))\*10000

else:

ZuiDiGongZi = float(re.sub("[^0-9\.]", "", strlist[0]))

ZuiGaoGongZi = float(re.sub("[^0-9\.]", "", strlist[1]))

if '天' in strlist[1]:

ZuiDiGongZi \*= 30

ZuiGaoGongZi \*= 30

if '年' in strlist[1]:

ZuiDiGongZi /= 12

ZuiGaoGongZi /= 12

ZuiDiGongZi\_list.append(ZuiDiGongZi)

ZuiGaoGongZi\_list.append(ZuiGaoGongZi)

data[10] = ZuiDiGongZi\_list

data[11] = ZuiGaoGongZi\_list

sorted\_data\_1 = data.sort\_values(by = 10, ascending = False)

sorted\_data\_2 = data.sort\_values(by = 11, ascending = False)

print('以最低工资计，薪资最高的10家企业是：')

i = 0

for row in sorted\_data\_1.itertuples():

if i < 10:

print(row[1])

i += 1

else:

break

print('')

print('以最高工资计，薪资最高的10家企业是：')

i = 0

for row in sorted\_data\_2.itertuples():

if i < 10:

print(row[1])

i += 1

else:

break

question3()

运行结果：



以最低工资计，薪资最高的10家企业是：

四川睿领科技有限公司

四川富为科技有限公司

OPPO广东移动通信有限公司

中美创世成都物联科技有限公司

深圳市小葱资本管理有限公司

好未来教育（原学而思教育)

NIO蔚来

神话科技传媒（深圳）有限公司上海分公司

上海众旦信息科技有限公司

深圳市捷兴电子商务有限公司

以最高工资计，薪资最高的10家企业是：

四川睿领科技有限公司

四川富为科技有限公司

好未来教育（原学而思教育)

上海众旦信息科技有限公司

深圳市捷兴电子商务有限公司

奇安信集团

成都世纪光合作用科技有限公司

中美创世成都物联科技有限公司

四川优萃科技有限公司

四川三秋兮文化旅游发展有限公司

(4)

Python代码；

import re

import pandas as pd

def question4():

'''数据分析相关的职位有哪些并进行排序'''

data = pd.read\_excel('51job.xlsx', sheet\_name = '51job', header = None)

selected\_data = data[data[1].str.contains('数据分析')].copy() #筛选数据分析相关岗位

ZuiDiGongZi\_list = [] #最低工资(按月计)

ZuiGaoGongZi\_list = [] #最高工资(按月计)

for row in selected\_data.itertuples():

strlist = row[3].split('-')

if len(strlist) == 1:

strlist.append(strlist[0])

if '千' in strlist[1]:

ZuiDiGongZi = float(re.sub("[^0-9\.]", "", strlist[0]))\*1000

ZuiGaoGongZi = float(re.sub("[^0-9\.]", "", strlist[1]))\*1000

elif '万' in strlist[1]:

ZuiDiGongZi = float(re.sub("[^0-9\.]", "", strlist[0]))\*10000

ZuiGaoGongZi = float(re.sub("[^0-9\.]", "", strlist[1]))\*10000

else:

ZuiDiGongZi = float(re.sub("[^0-9\.]", "", strlist[0]))

ZuiGaoGongZi = float(re.sub("[^0-9\.]", "", strlist[1]))

if '天' in strlist[1]:

ZuiDiGongZi \*= 30

ZuiGaoGongZi \*= 30

if '年' in strlist[1]:

ZuiDiGongZi /= 12

ZuiGaoGongZi /= 12

ZuiDiGongZi\_list.append(ZuiDiGongZi)

ZuiGaoGongZi\_list.append(ZuiGaoGongZi)

selected\_data[10] = ZuiDiGongZi\_list

selected\_data[11] = ZuiGaoGongZi\_list

sorted\_data\_1 = selected\_data.sort\_values(by = 10, ascending = False)

sorted\_data\_2 = selected\_data.sort\_values(by = 11, ascending = False)#排序

sorted\_data\_1.to\_excel('最低工资降序.xlsx', index=False, header=False)

sorted\_data\_1.to\_excel('最高工资降序.xlsx', index=False, header=False)

question4()

运行结果：

按最低工资从高到低排序：

|  |
| --- |
| OPPO广东移动通信有限公司 |
| 成都极米科技股份有限公司 |
| 天池创新（北京）软件技术有限公司 |
| 中国电子科技网络信息安全有限公司 |
| 联联永欣科技（成都）有限公司 |
| 成都创人所爱科技股份有限公司 (tap4fun) |
| 成都积微物联集团股份有限公司 |
| 四川万网鑫成信息科技有限公司 |
| 成都创雅达科技有限公司 |
| 成都美尔贝科技股份有限公司 |
| 字节跳动 |
| 宜宾新青年电子商务有限公司 |
| 易视腾科技股份有限公司 |
| 成都霖翰丰恒信息技术咨询有限公司 |
| 成都返空汇网络技术有限公司 |
| 成都若溪科技有限公司 |
| 爱奇艺（www.iqiyi.com） |
| 成都霖翰丰恒信息技术咨询有限公司 |
| 京东物流西南分公司 |
| 四川福摩数字科技有限公司 |
| 卓越质胜（北京）科技发展有限公司 |
| 腾讯RDM |
| 北京陌陌信息技术有限公司成都分公司 |
| 成都米大科技有限公司 |
| 四川金熊猫新媒体有限公司 |
| 北京国研网信息有限公司 |
| 美团点评 |
| 纽仕达（泉州）供应链有限公司 |
| 成都南迪鞋业有限公司 |
| 北京国研网信息有限公司 |
| 成都霖翰丰恒信息技术咨询有限公司 |
| 成都与梦同行科技有限公司 |
| 成都艾森威科技有限公司 |
| 成都萌想科技有限责任公司 |
| 成都泰便利电子商务有限公司 |
| 成都新视角商贸有限公司 |
| 成都亿盟恒信科技有限公司 |
| 成都长晟鑫电子商务有限公司 |
| 四川爽呗智能设备有限公司 |
| 成都仆儿电子商务有限公司 |
| 成都匠领科技有限公司 |
| 成都仆儿电子商务有限公司 |
| 四川安云优聘企业管理咨询服务有限公司 |
| 成都匠领科技有限公司 |
| 成都仆儿电子商务有限公司 |
| 启晨科技 |
| 湖南潭州教育网络科技有限公司成都第一分公司 |
| 成都汇扬聚鑫信息技术服务有限公司 |
| 易店无忧 |
| 成都仆儿电子商务有限公司 |
| 明略科技集团 |
| 成都我行我数科技有限公司 |
| 四川奇迹众创科技有限公司 |
| 成都部落窝科技有限公司 |
| 成都匠领科技有限公司 |

按最高工资从高到低排序：

|  |
| --- |
| OPPO广东移动通信有限公司 |
| 成都极米科技股份有限公司 |
| 天池创新（北京）软件技术有限公司 |
| 中国电子科技网络信息安全有限公司 |
| 联联永欣科技（成都）有限公司 |
| 成都创人所爱科技股份有限公司 (tap4fun) |
| 成都积微物联集团股份有限公司 |
| 四川万网鑫成信息科技有限公司 |
| 成都创雅达科技有限公司 |
| 成都美尔贝科技股份有限公司 |
| 字节跳动 |
| 宜宾新青年电子商务有限公司 |
| 易视腾科技股份有限公司 |
| 成都霖翰丰恒信息技术咨询有限公司 |
| 成都返空汇网络技术有限公司 |
| 成都若溪科技有限公司 |
| 爱奇艺（www.iqiyi.com） |
| 成都霖翰丰恒信息技术咨询有限公司 |
| 京东物流西南分公司 |
| 四川福摩数字科技有限公司 |
| 卓越质胜（北京）科技发展有限公司 |
| 腾讯RDM |
| 北京陌陌信息技术有限公司成都分公司 |
| 成都米大科技有限公司 |
| 四川金熊猫新媒体有限公司 |
| 北京国研网信息有限公司 |
| 美团点评 |
| 纽仕达（泉州）供应链有限公司 |
| 成都南迪鞋业有限公司 |
| 北京国研网信息有限公司 |
| 成都霖翰丰恒信息技术咨询有限公司 |
| 成都与梦同行科技有限公司 |
| 成都艾森威科技有限公司 |
| 成都萌想科技有限责任公司 |
| 成都泰便利电子商务有限公司 |
| 成都新视角商贸有限公司 |
| 成都亿盟恒信科技有限公司 |
| 成都长晟鑫电子商务有限公司 |
| 四川爽呗智能设备有限公司 |
| 成都仆儿电子商务有限公司 |
| 成都匠领科技有限公司 |
| 成都仆儿电子商务有限公司 |
| 四川安云优聘企业管理咨询服务有限公司 |
| 成都匠领科技有限公司 |
| 成都仆儿电子商务有限公司 |
| 启晨科技 |
| 湖南潭州教育网络科技有限公司成都第一分公司 |
| 成都汇扬聚鑫信息技术服务有限公司 |
| 易店无忧 |
| 成都仆儿电子商务有限公司 |
| 明略科技集团 |
| 成都我行我数科技有限公司 |
| 四川奇迹众创科技有限公司 |
| 成都部落窝科技有限公司 |
| 成都匠领科技有限公司 |

(5)

Python代码：

import re

import pandas as pd

import matplotlib.pyplot as plt

from wordcloud import WordCloud

import jieba

def question5():

'''对岗位职位进行可视化词云分析，并对结果进行阐述'''

data = pd.read\_excel('51job.xlsx', sheet\_name = '51job', header = None)

words\_total = []

for row in data.itertuples():

words = row[1] + row[9] + row[10]

words = re.sub(r'\\r', '', words)

words = re.sub(r'\\n', '', words)

words = re.sub(r'\\xa0', '', words)

words = re.sub('[0-9]', '', words)

words = re.sub('\W', '', words)

words\_total += words

text\_cut = '/'.join(words\_total)

wordcloud = WordCloud( background\_color='white',font\_path = 'msyh.ttc', width=1000, height=860, margin=2).generate(text\_cut)

# 显示图片

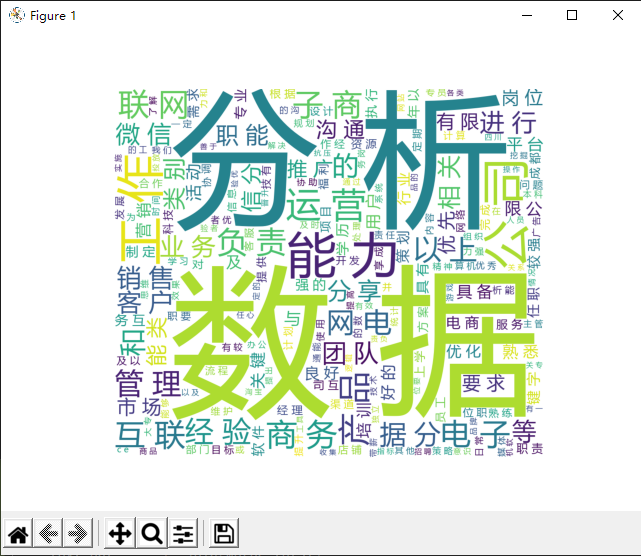
plt.imshow(wordcloud)

plt.axis('off')

plt.show()

question5()

运行结果：



可以看出，数据、分析两词出现的次数最多，这说明和数据分析相关的职位很多。除了数据分析以外，和销售、运营、管理有关的职位较多。

“能力”出现频次高，说明企业重视员工的能力，除了能力，企业还重视的员工员工素质包括：沟通、团队协作能力强，负责任、有经验等。