湖北商贸学院
HUBEI BUSINESS COLLEGE

课程设计报告

课程名称：	数据分析（基于 Python）课程设计
设计题目：	高考录取分数及大学信息数据分析
学院名称：	人工智能学院
专业班级：	2021 级计算机科学与技术 2 班
学　　号：	21106080901212
学生姓名：	周余
指导教师：	张秋生

2023 年 6 月 10 日

目　录

一、需求分析

1、需求任务概述

通过对高考录取分数及大学信息数据的分析，可以了解各个省市高校录取分数线的变化趋势，掌握高校录取分数线的变化规律，为考生和家长提供科学参考，帮助他们更好地选择适合自己的院校。同时，这也可以为高校招生工作提供科学依据，为高校招生工作提供参考依据，为高校招生工作提供科学支持。除此之外，高考录取分数及大学信息数据分析还可以为教育部门提供科学依据，为教育部门提供参考依据，为教育部门提供科学支持。

二、数据源的获取、读取与数据预处理

1、数据源的获取、读取

```
#导入所需包
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#1. 获取高考录取分数数据
path =r'C:\Users\Zhou Yu\PycharmProjects\pythonProject\数据分析课程设计\data\
高考录取分数数据集.csv'
data=pd.read_csv(path)
print('输出高考录取分数前 5 行\n',data.head())
#2. 获取大学信息数据
path1 = r'C:\Users\Zhou Yu\PycharmProjects\pythonProject\数据分析课程设计
\data\quanguoputonggaodengxuexiaomingdan.xls'
data2=pd.read_excel(path)
print('输出大学信息前五行\n',data2.head())
```

2、数据预处理

```
#将’------’用 NULL 代替
df = df.replace('------',np.nan)
# 检测空值
null_data = df.isnull()
# 查看缺失值个数
print('缺失值情况：\n',null_data.sum())
# 对缺失值使用平均值填充
df['2020 录取分数'] = df['2020 录取分数'].fillna(df['近五年平均录取分数'])
df['2019 录取分数'] = df['2019 录取分数'].fillna(df['近五年平均录取分数'])
df['2018 录取分数'] = df['2018 录取分数'].fillna(df['近五年平均录取分数'])
df['2017 录取分数'] = df['2017 录取分数'].fillna(df['近五年平均录取分数'])
df['2016 录取分数'] = df['2016 录取分数'].fillna(df['近五年平均录取分数'])
# 查看重复值个数
print('重复值个数：\n',df.duplicated().sum())
# 查看录取分数数据描述
describe = df.describe()
print('描述性分析：\n',describe)
```

三、数据分析、处理

1、数据分析、处理

```
#1.对近五年平均录取分数进行分析
print('------------------------------')
bins = [0,200,300,400,500, 550, 600, 650, 700, 750]
labels = ["0-200","200-300","300-400","400-500", "500-550", "550-600", "600-650",
"650-700", "700-750"]
df["score_bin"] = pd.cut(df["近五年平均录取分数"], bins=bins, labels=labels)
dist = df["score_bin"].value_counts(normalize=True).sort_index()
print('平均录取成绩的分布表：\n',dist)
print('------------------------------')
```

```
------------------------------
平均录取成绩的分布表：
 score_bin
0-200      0.036968
200-300    0.226496
300-400    0.224662
400-500    0.321494
500-550    0.096875
550-600    0.055089
600-650    0.028781
650-700    0.008805
700-750    0.000831
Name: proportion, dtype: float64
------------------------------
```

如图 1 近五年平均录取分数分布

```
#2.在各省招生的大学个数
print('------------------------------')
struct1 = df["招生省份"].value_counts()
print('在各省招生的大学个数\n',struct1)
print('------------------------------')
```

```
------------------------------
在各省招生的大学个数
 招生省份
河南      2056
安徽      1977
甘肃      1921
山西      1915
四川      1854
山东      1783
河北      1753
```

```
贵州      1733
内蒙古     1720
云南      1698
湖北      1698
广西      1674
广东      1636
湖南      1630
重庆      1621
辽宁      1604
陕西      1595
江西      1579
新疆      1554
福建      1548
黑龙江     1495
吉林      1427
浙江      1381
宁夏      1324
江苏      1152
青海      1128
海南      1053
天津       956
西藏       903
上海       796
北京       763
Name: count, dtype: int64
--------------------------------
```

如图 2 在各省招生的大学个数

```
#3.近五年平均录取分数,并按照平均录取分数从高到低进行排序
print('--------------------------------')
grouped_df = df.groupby("招生省份")["近五年平均录取分数"].mean().reset_index()
grouped_df = grouped_df.sort_values("近五年平均录取分数", ascending=False)
print("近五年平均录取分数,并按照平均录取分数从高到低进行排\n",grouped_df)
print('--------------------------------')
```

```
--------------------------------
近五年平均录取分数,并按照平均录取分数从高到低进行排序
    招生省份     近五年平均录取分数
19    海南   557.080643
3     北京   473.563630
11    广东   436.379523
18    浙江   436.368899
9     山东   434.578108
24    西藏   422.267183
6     天津   421.780910
27    重庆   417.779416
```

```
21    湖南   415.266309
15    江西   413.511843
26    辽宁   408.707616
16    河北   407.263206
20    湖北   406.243316
23    福建   404.498137
5     四川   403.085428
8     安徽   396.617914
4     吉林   393.938788
28    陕西   390.543835
1     云南   388.371162
30   黑龙江   387.662687
17    河南   382.648833
12    广西   374.712366
7     宁夏   370.422407
0     上海   369.251759
22    甘肃   362.426844
10    山西   360.820853
13    新疆   350.198874
25    贵州   341.238988
29    青海   331.655984
2    内蒙古   326.635194
14    江苏   321.652286
--------------------------------
```

如图 3 近五年平均录取分数

```
#4.统计北京大学、清华大学、浙江大学、上海交通大学、南京大学、西安交通大学、中国科学技术大学、武汉大学、华中科技大学近五年在鄂招生分数变化
print('--------------------------------')
universityPartZh = ['北京大学', '清华大学', '浙江大学', '上海交通大学', '南京大学', '西安交通大学', '中国科学技术大学',
                    '武汉大学', '华中科技大学']
year = [2020, 2019, 2018, 2017, 2016]
scores = np.zeros([len(universityPartZh), 5], dtype=int, order='C')
for i in range(len(universityPartZh)):
    for j in range(46927):
        if (df['学校'][j] == universityPartZh[i]) & (df['招生省份'][j] == '湖北'):
            for k in range(5, 10):
                if df.iloc[j, k] == '------':   # 处理异常值
                    scores[i, k - 5] = df.iloc[j, 10]
                else:
                    scores[i, k - 5] = df.iloc[j, k]
print("湖北省历年录取分数线")
print("-" * 30)
for i in range(len(universityPartZh)):
```

```
        print(f"{universityPartZh[i]}: {list(scores[i])}")
print('------------------------------')
```

```
------------------------------
湖北省历年录取分数线
------------------------------
北京大学：[687，688，695，693，690]
清华大学：[687，683，663，690，679]
浙江大学：[673，674，651，659，664]
上海交通大学：[687，687，681，671，681]
南京大学：[670，657，651，671，653]
西安交通大学：[636，624，609，632，612]
中国科学技术大学：[657，674，663，658，663]
武汉大学：[641，637，626，637，631]
华中科技大学：[644，632，638，612，633]
------------------------------
```

如图 4 湖北省历年录取分数线

```
#5.统计武汉大学在每个省份招生的录取平均分数
print('------------------------------')
averageScores = []
for i in range(len(province)):
        for j in range(46927):
                if (df['学校'][j] == '武汉大学') & (df['招生省份'][j] == province[i]):
                        averageScores.append(df['近五年平均录取分数'][j])
print("武汉大学各省平均录取分数线")
print("-" * 30)
for i in range(len(province)):
        print(f"{province[i]}: {averageScores[i]}")
print('------------------------------')
```

```
------------------------------
武汉大学各省平均录取分数线
------------------------------
河南：650.4
安徽：649.8
甘肃：614.0
山西：623.2
四川：661.0
山东：657.0
河北：659.8
贵州：639.0
内蒙古：622.2
云南：660.4
湖北：634.4
广西：631.0
广东：618.6
```

```
湖南：638.4
重庆：641.0
辽宁：649.2
陕西：648.4
江西：637.8
新疆：599.2
福建：622.0
黑龙江：633.2
吉林：636.4
浙江：667.8
宁夏：566.6
江苏：390.4
青海：556.6
海南：756.8
天津：636.8
西藏：600.0
上海：461.6
北京：650.6
--------------------------------
```

如图 5 武汉大学在每个省份招生的录取平均分数

```
#6.分析各省的大学个数
print('--------------------------------')
groupby_province = data2.groupby('所属省份')['学校名称'].agg(数量=('count'))
print('各省大学个数：',groupby_province)
print('--------------------------------')
--------------------------------
各省大学个数：                数量
所属省份
上海市           64
云南省           77
内蒙古自治区       53
北京市           92
吉林省           62
四川省          109
天津市           57
宁夏回族自治区      19
安徽省          119
山东省          145
山西省           80
广东省          151
广西壮族自治区      74
新疆维吾尔自治区     47
江苏省          167
江西省          100
```

```
河北省          121
河南省          134
浙江省          107
海南省           19
湖北省          129
湖南省          124
甘肃省           49
福建省           89
西藏自治区         7
贵州省           70
辽宁省          115
重庆市           65
陕西省           93
青海省           12
黑龙江省          81
--------------------------------
```

如图 6 各省大学数

```
#7. 学校类型分析比较
print('--------------------------------')
school_property_bk =data2[data2['办学层次']=='本科']
school_property_zk =data2[data2['办学层次']=='专科']
#  本科
#  民办
school_property_bk_private = school_property_bk[school_property_bk['备注']=='民办']
school_property_bk_private_count = school_property_bk_private['学校名称'].count()
#合资
school_property_bk_jointly = school_property_bk[school_property_bk['备注']=='中外合作办学']
school_property_bk_jointly_count = school_property_bk_jointly['学校名称'].count()
#  专科
#民办
school_property_zk_private = school_property_zk[school_property_zk['备注']=='民办']
school_property_zk_private_count = school_property_zk_private['学校名称'].count()
#  合资
school_property_zk_jointly = school_property_zk[school_property_zk['备注']=='中外合作办学']
school_property_zk_jointly_count = school_property_zk_jointly['学校名称'].count()
school_property_bk_count = school_property_bk['办学层次'].count()
school_property_zk_count = school_property_zk['办学层次'].count()
#公办本科
school_property_bk_public_count =
school_property_bk_count-school_property_bk_private_count-school_property_bk_jo
```

```
intly_count
#公办专科
school_property_zk_public_count =
school_property_zk_count-school_property_zk_private_count-school_property_zk_joi
ntly_count
print('本科院校数量：',school_property_bk_count)
print('本科院校中的民办数量：',school_property_bk_private_count)
print('本科院校中的合资数量：',school_property_bk_jointly_count)
print('本科院校中的公办数量：',school_property_bk_public_count)
print('\n')
print('专科院校数量：',school_property_zk_count)
print('专科科院校中的民办数量：',school_property_zk_private_count)
print('专科院校中的合资数量：',school_property_zk_jointly_count)
print('专科院校中的公办数量：',school_property_zk_public_count)
print('------------------------------')
```

```
--------------------------------
本科院校数量： 1243
本科院校中的民办数量： 417
本科院校中的合资数量： 7
本科院校中的公办数量： 819

专科院校数量： 1388
专科科院校中的民办数量： 318
专科院校中的合资数量： 2
专科院校中的公办数量： 1068
--------------------------------
```

如图 7 学校类型数量

```
#8.各城市的大学数量
print('------------------------------')
groupby_province1 = data2.groupby('所在地')['学校名称'].agg(数量
=('count')).sort_values(by='数量',ascending=False)
print(groupby_province1)
print('------------------------------')
```

```
--------------------------------
              数量
所在地
北京市          92
武汉市          84
广州市          83
重庆市          65
上海市          64
...           ..
广安市           1
西双版纳傣族自治州    1
```

```
张家港市          1
张掖市            1
七台河市          1

[331 rows x 1 columns]
------------------------------
```

如图 8 各城市的大学数量

```
#9.主管部门和办学层级分析
print('------------------------------')
groupby_department = data2.groupby('主管部门')['学校名称'].agg(数量=('count'))
print(groupby_department)
print(groupby_department.describe())
print('数量多于平均值：\n',groupby_department[groupby_department['数量']>=40])
print('------------------------------')
------------------------------
            数量
主管部门
上海市           33
上海市\n 中国科学院    1
上海市教委         20
中华全国总工会        1
中华妇女联合会        1
...           ..
陕西省教育厅        30
青海省           11
青海省教育厅         1
黑龙江省          61
黑龙江省教育厅       17

[85 rows x 1 columns]
              数量
count   85.000000
mean    30.952941
std     29.074578
min      1.000000
25%      5.000000
50%     26.000000
75%     50.000000
max    105.000000
数量多于平均值：
            数量
主管部门
云南省         56
内蒙古自治区     43
```

```
吉林省          42
四川省          69
天津市          42
安徽省          86
山东省         102
山东省教育厅      40
山西省          65
广东省          92
广东省教育厅      54
广西壮族自治区    50
教育部          76
江苏省         105
江苏省教育厅      52
江西省          68
河北省          81
河南省          96
浙江省          69
湖北省          79
湖北省教育厅      42
湖南省          90
甘肃省          40
福建省          50
贵州省          55
辽宁省          77
陕西省          57
黑龙江省        61
--------------------------------
```

如图 9 主管部门和办学层级的关系

```
#10.主管部门和办学层级的联合分析（交叉分布）
print('--------------------------------')
df1 = data2[data2['办学层次']=='本科'].groupby('主管部门')['学校名称'].agg(本科
=('count'))
df1['专科'] = data2[data2['办学层次']=='专科'].groupby('主管部门')['学校名称
'].agg(专科=('count'))
data3 = groupby_department.sort_values(by='数量',ascending=False).head(15)
count_bin =[0,10,20,30,40,50,60,70,80]
count_labels = ['0-9 个','10-19','20-29','30-39','40-49','50-59','60-69','70-79']
df1['数量分层'] = pd.cut(df1['本科'], count_bin, right=False, labels=count_labels)
df1['专科数量分层'] = pd.cut(df1['专科'], count_bin, right=False, labels=count_labels)
ptResult1=df1.pivot_table(
    values=['本科'],
    index=['数量分层'],
    columns=[],
    aggfunc=[np.size]
```

```
)
ptResult2=df1.pivot_table(
  values=['专科'],
  index=['专科数量分层'],
  columns=[],
  aggfunc=[np.size]
)
print(ptResult1.join(ptResult2))
print('------------------------------')
```

```
------------------------------
       size
         本科   专科
数量分层
0-9 个     36   18
10-19     15   16
20-29     22   11
30-39      8    5
40-49      1    5
50-59      0    4
60-69      0    3
70-79      1    0
------------------------------
```

如图 10 主管部门和办学层级的联合分布

四、数据可视化

1、数据可视化代码

```
# 数据可视化
# 设置 Matplotlib 正常显示中文和负号
plt.rc("font", family='SimHei')
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用黑体显示中文
plt.rcParams['axes.unicode_minus'] = False  # 正常显示负号
# 11.对近五年平均录取分数分布进行可视化（如图 11）
dist.plot(kind="bar",figsize=(8,6)) # 用柱状图显示各区间的频数和频率
plt.xticks(rotation=30)
plt.title("录取分数分布")
plt.xlabel("分数范围")
plt.ylabel("频率")
plt.show()
# 12.用饼图显示招生省份在总体中的比例（如图 12）
struct1.plot.pie(autopct="%.1f%%") # 用饼图显示招生省份在总体中的比例
plt.title('招生省份在总体中的比例')
plt.tight_layout()
plt.show()
# 13.绘制箱线图：近五年平均录取分数（如图 13）
plt.boxplot(df['近五年平均录取分数'], vert=False)
plt.title('录取分数箱形图')
plt.xlabel('录取分数')
plt.show()
# 14.用四个子图，分别绘制最近五年内招生省份的录取平均分数、各省份招生分布情况、各省份文科和理科录取情况、不同省份的录取分数标准差（如图 14）
data_counts = data["招生省份"].value_counts()
data_counts = data_counts[data_counts>0] # 去掉数量为 0 的省份
data_counts_sorted = data_counts.sort_values(ascending=False)
fig, axs = plt.subplots(2, 2, figsize=(16, 12))
# 柱状图 1
axs[0, 0].bar(grouped_df["招生省份"], grouped_df["近五年平均录取分数"],
width=0.5)
axs[0, 0].set_xticks(range(len(grouped_df["招生省份"])))
axs[0, 0].set_xticklabels(grouped_df["招生省份"], rotation=45, ha="right")
axs[0, 0].set_xlabel("招生省份")
axs[0, 0].set_ylabel("近五年平均录取分数")
axs[0, 0].set_title("最近五年内招生省份的录取平均分数")
# 柱状图 2
axs[0, 1].bar(data_counts_sorted.index, data_counts_sorted.values, width=0.5)
axs[0, 1].set_xticks(range(len(data_counts_sorted.index)))
axs[0, 1].set_xticklabels(data_counts_sorted.index, rotation=45, ha="right")
```

```
axs[0, 1].set_xlabel("招生省份")
axs[0, 1].set_ylabel("数量/所")
axs[0, 1].set_title("各省份招生分布情况")
# 柱状图 3
cross_df = pd.crosstab(df["招生省份"], df["文/理"])
cross_df.plot(kind="bar", stacked=True, ax=axs[1, 0])
axs[1, 0].set_xticks(range(len(cross_df.index)))
axs[1, 0].set_xticklabels(cross_df.index, rotation=45, ha="right")
axs[1, 0].set_xlabel("招生省份")
axs[1, 0].set_ylabel("数量")
axs[1, 0].set_title("各省份文科和理科录取情况")
# 柱状图 4
std_df = df.groupby("招生省份")["2020 录取分数"].std().reset_index()
axs[1, 1].bar(std_df["招生省份"], std_df["2020 录取分数"], width=0.5)
axs[1, 1].set_xticks(range(len(std_df["招生省份"])))
axs[1, 1].set_xticklabels(std_df["招生省份"], rotation=45, ha="right")
axs[1, 1].set_xlabel("招生省份")
axs[1, 1].set_ylabel("录取分数标准差")
axs[1, 1].set_title("不同省份的录取分数标准差")
plt.subplots_adjust(hspace=0.4) # 调整子图之间的间距
plt.show()
# 15.绘制折线图，统计在每个省份招生的学校数目（如图 15）
plt.plot(range(1, len(province) + 1), numCollegeEnrollPerProvince, 'ro-')
plt.title('每个省份招生的学校数目')
plt.xticks(range(1, len(province) + 1), province, rotation=90)
plt.ylabel('Counts')
plt.show()
# 16.用两个子图，绘制北京大学、清华大学、浙江大学、上海交通大学、南京大学、西安交通大学、中国科学技术大学、武汉大学、华中科技大学近五年在鄂招生分数变化折线图（如图 16）
fig, axs = plt.subplots(1, 2, figsize=(16, 5))
# 北京大学和清华大学的变化
axs[0].plot(year, scores[0].tolist(), 'o--', label='北京大学')
axs[0].plot(year, scores[1].tolist(), 'x--', label='清华大学')
for a, b in zip(year, scores[0].tolist()):
    axs[0].text(a, b, b, ha='center', va='bottom', fontsize=8)
for a, b in zip(year, scores[1].tolist()):
    axs[0].text(a, b, b, ha='center', va='bottom', fontsize=8)
axs[0].legend()
axs[0].set_title('北京大学和清华大学近五年的在鄂招生分数变化')
axs[0].set_xlabel('年份')
axs[0].set_ylabel('录取分数')
# 武汉大学和华中科技大学的变化
axs[1].plot(year, scores[7].tolist(), 'o--', label='武汉大学')
```

```
axs[1].plot(year, scores[8].tolist(), 'x--', label='华中科技大学')
for a, b in zip(year, scores[7].tolist()):
    axs[1].text(a, b, b, ha='center', va='bottom', fontsize=8)
for a, b in zip(year, scores[8].tolist()):
    axs[1].text(a, b, b, ha='center', va='bottom', fontsize=8)
axs[1].legend()
axs[1].set_title('武汉大学和华中科技大学近五年的在鄂招生分数变化')
axs[1].set_xlabel('年份')
axs[1].set_ylabel('录取分数')
# 设置刻度间隔
x_major_locator = MultipleLocator(1)
y_major_locator = MultipleLocator(10)
for ax in axs:
    ax.xaxis.set_major_locator(x_major_locator)
    ax.yaxis.set_major_locator(y_major_locator)
plt.show()
# 17.绘制武汉大学在每个省份招生的录取平均分数折线图（如图 17）
plt.plot(list(range(1, len(province) + 1)), averageScores, 'o-')
plt.title('武汉大学各省平均录取分数线')
plt.xticks(range(1, len(province) + 1), province, rotation=90)
plt.ylabel('score')
plt.show()
# 18.绘制各省大学个数饼图（如图 18）
arr = list(groupby_province['数量'].values)
label=np.array([])
label=np.append(label,data2['所属省份'].drop_duplicates().dropna())
print('省份个数：',label.size)
plt.title('各省大学占比')
plt.pie(arr,labels=label,autopct='%1.1f%%')
plt.legend(loc='upper right',fontsize=7,bbox_to_anchor=(1.1, 1.05))
plt.show()
# 19 绘制柱状图和饼图分析学校类型的数量占比（如图 19）
x1=['本科总数量','公办本科','合资本科','民办本科','专科总数量','公办专科','合资专科','民办专科']
y
=[school_property_bk_count,(school_property_bk_count-school_property_bk_private_count-school_property_bk_jointly_count),school_property_bk_jointly_count,
school_property_bk_private_count,school_property_zk_count,(school_property_zk_count-school_property_zk_private_count-school_property_zk_jointly_count),school_property_zk_jointly_count,
    school_property_zk_private_count]
fig, axs = plt.subplots(1, 2, figsize=(12, 5))
axs[0].pie(y, labels=x1, autopct='%1.2f%%')
axs[0].legend(x1, loc='upper right', fontsize=7, bbox_to_anchor=(1.1, 1.05))
```

```
axs[0].set_title('学校类型占比')
#  学校类型数量
axs[1].bar(x1, y, width=0.8, align="center")
axs[1].plot(x1, y, 'r-')
axs[1].set_xlabel('办学层')
axs[1].set_ylabel('学校数量/个')
axs[1].set_title('学校类型数量')
axs[1].set_xticks(range(len(x1)))   #  设置刻度位置
axs[1].set_xticklabels(x1, rotation=30)
axs[1].legend(['数量'], loc='upper left', fontsize=7)
plt.show()
# 20.各城市大学数量频率直方图 && 各城市大学数量的众数柱状图（如图 20）
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(14,8))
ax1.set_title('各城市大学数量频率直方图')
ax1.hist(arr1,bins=65,color="green",edgecolor="black")
ax1.set_xlabel('区间(学校个数)')
ax1.set_ylabel('频数/频率')
ax2.set_title('各城市大学数量的众数柱状图')
ax2.bar(groupby_province1['数量
'].drop_duplicates().index,arr2,color="red",edgecolor="black")
ax2.plot(groupby_province1['数量'].drop_duplicates().index,arr2,"b-.")
ax2.set_xlabel('学校')
ax2.set_ylabel('单位/个')
plt.xticks(rotation=90,fontsize=6)
plt.show()
# 21.绘制主管部门和办学层级之间的数量关系条形图（如图 21）
plt.figure(figsize=(10,6))
x_t = range(len(arr5))
plt.yticks(x_t,arr5)
plt.barh(arr5,arr6,height=0.5)
plt.title('主管部门学校数量（仅显示大于平均值的）')
plt.ylabel("主管部门名称")
plt.xlabel('学校数量/个')
plt.xticks(rotation=90)
plt.show()
# 22.省份间办学层次的对比分析散点图（如图 22）
#  按数量从大到小排列
groupby_sf_all =data2.groupby('所属省份')['学校名称'].agg(数量
=('count')).sort_values(by='数量',ascending=False)
print(groupby_sf_all)
groupby_sf_bk =data2[data2['办学层次']=='本科'].groupby('所属省份')['学校名称
'].agg(数量=('count'))
print(groupby_sf_bk)
groupby_sf_zk =data2[data2['办学层次']=='专科'].groupby('所属省份')['学校名称
```

```
'].agg(数量=('count'))
print(groupby_sf_zk)
plt.figure(figsize=(10,6))
#  绘制全部学校
plt.scatter(list(groupby_sf_all.index),list(groupby_sf_all['数量']),label='全部学校
',c='r',marker='o')
#绘制本科学校
plt.scatter(list(groupby_sf_bk.index),list(groupby_sf_bk['数量']),label='本科学校
',c='g',marker='x')
#绘制专科学校
plt.scatter(list(groupby_sf_zk.index),list(groupby_sf_zk['数量']),label='专科学校
',c='b',marker='v')
plt.xticks(rotation=90)
# plt.tight_layout()
plt.title('省份间办学层次的对比分析')
plt.legend()
plt.xlabel('省份')
plt.ylabel('数量/个')
plt.show()
# 23.绘制各部门本科，专科数量分层热力图和气泡图（如图 23）
fig, axs = plt.subplots(1, 2, figsize=(12, 5))
#  学校数量分层和办学层级分布热力图
data = np.array([[36, 18], [15, 16], [22, 11], [8, 5], [1, 5],[0,4],[0,3],[1,0]])
axs[0].imshow(data)
axs[0].set_xticks(np.arange(len(["本科", "专科"])))
axs[0].set_yticks(np.arange(len(count_labels)))
axs[0].set_xticklabels(["本科", "专科"])
axs[0].set_yticklabels(count_labels)
plt.setp(axs[0].get_xticklabels(), rotation=45, ha="right", rotation_mode="anchor")
for i in range(len(count_labels)):
    for j in range(len(["本科", "专科"])):
        text = axs[0].text(j, i, data[i, j], ha="center", va="center", color="w")
axs[0].set_title("数量分层和办学层级分布热力图")
#  学校数量分层和办学层级分布气泡图
axs[1].scatter(data[:, 0], data[:, 1], s=data[:, 1] * 10)
axs[1].set_xlabel('专科')
axs[1].set_ylabel('本科')
axs[1].set_title("数量分层和办学层级分布气泡图")
plt.tight_layout()
plt.show()
```

2、数据可视化截图

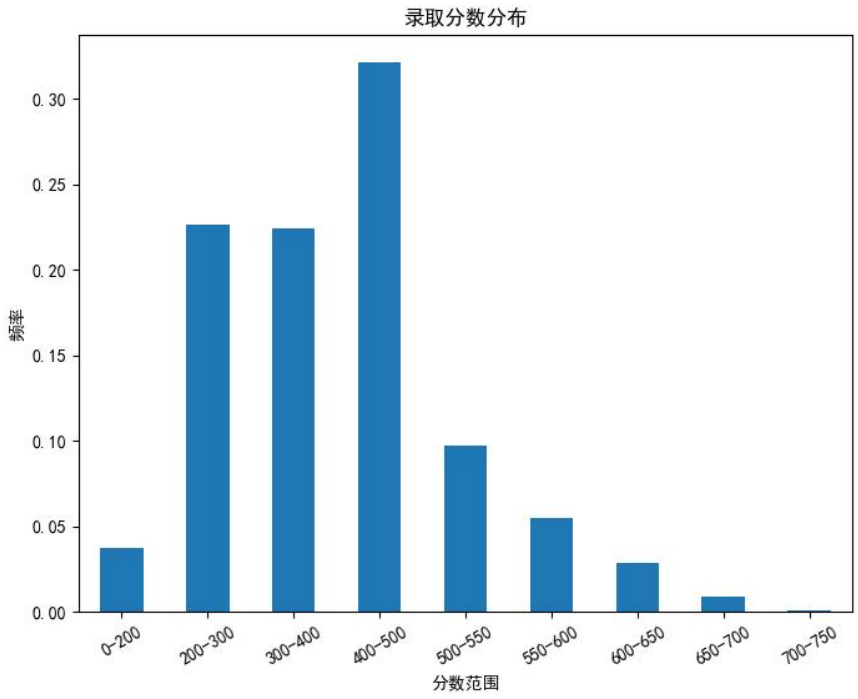

图 11 录取分数分布柱状图

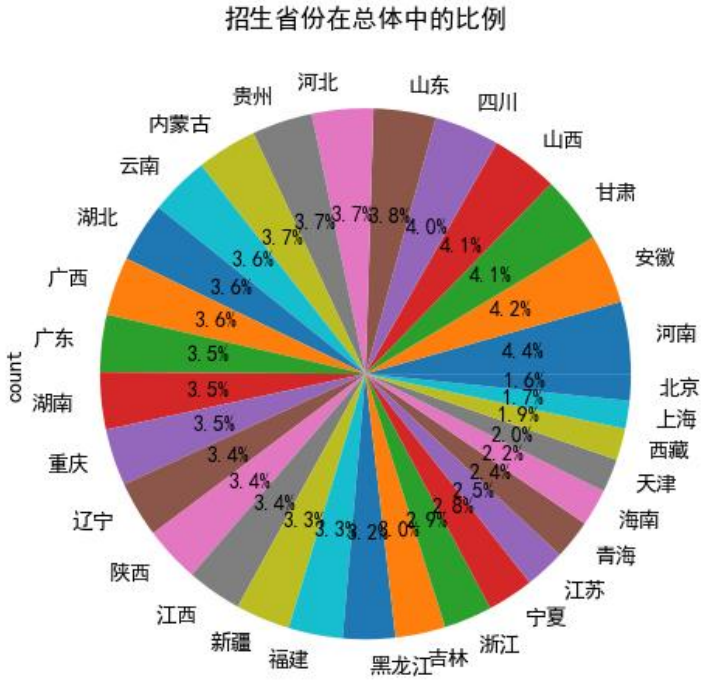

图 12 省份招生占比饼图

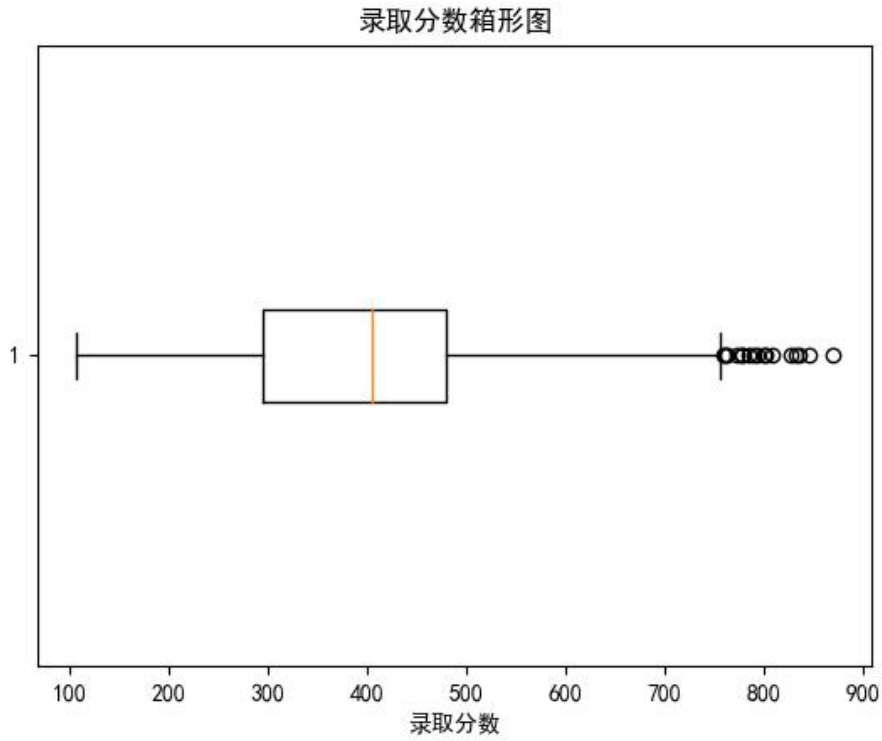

图 13 录取分数箱线图

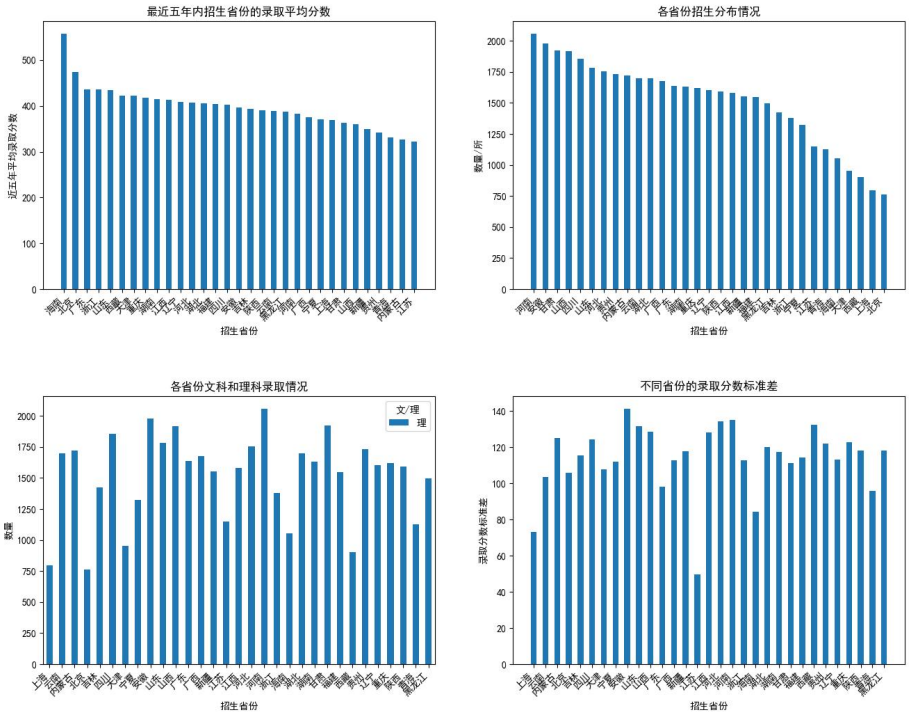

图 14 近五年内招生省份的录取平均分数、各省份招生分布情况、各省份文科和理科录取况、不同省份的录取分数标准差柱状图

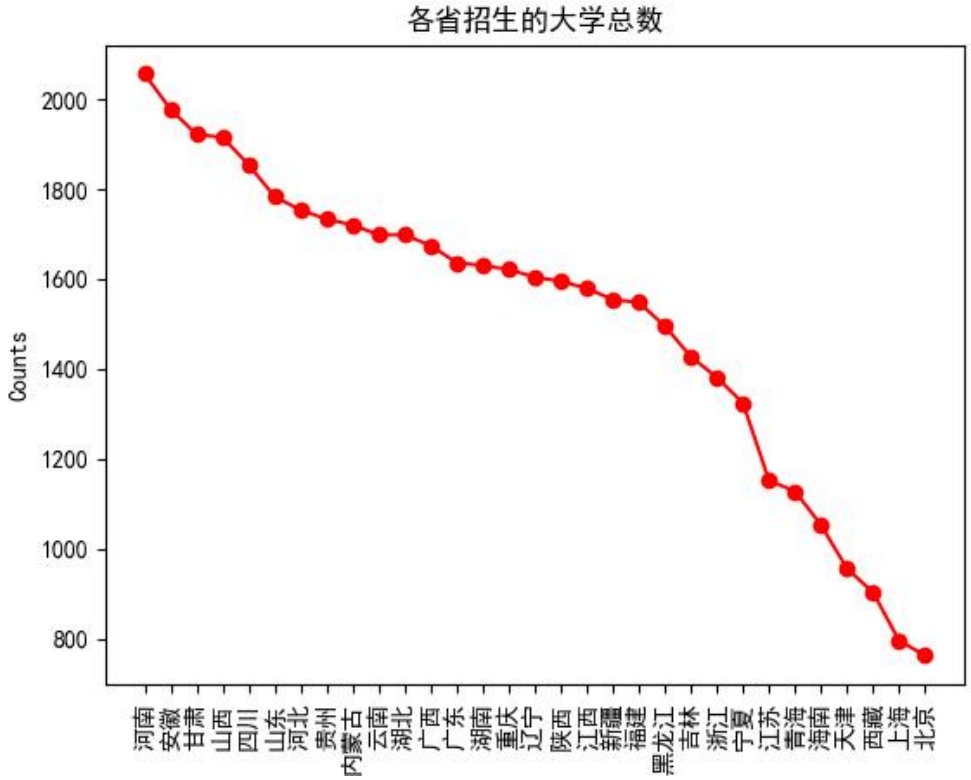

图 15 各省招生折线图

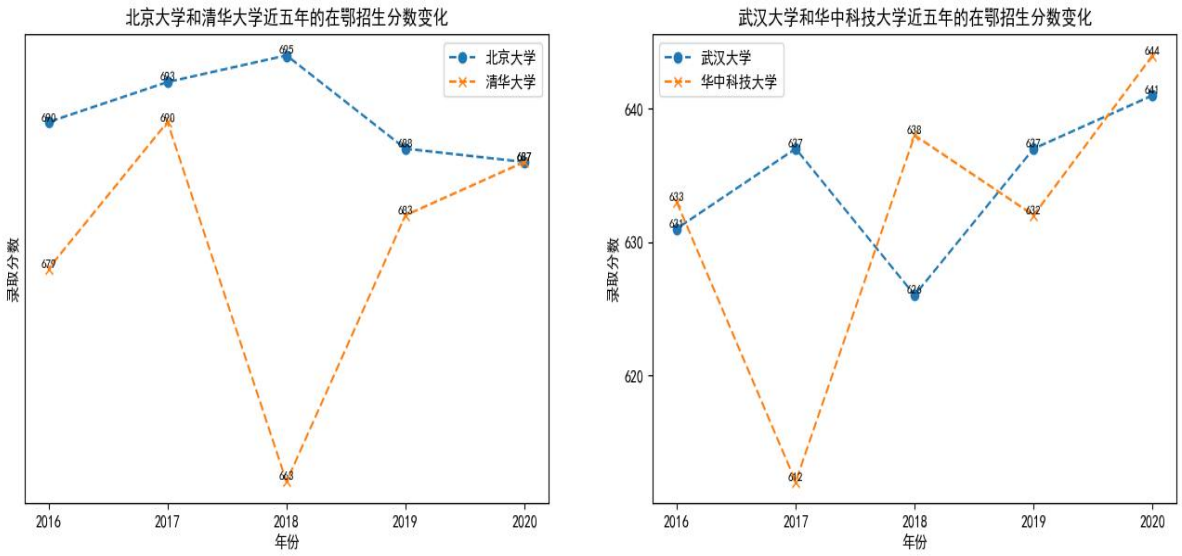

图 16 在鄂招生分数变化折线图

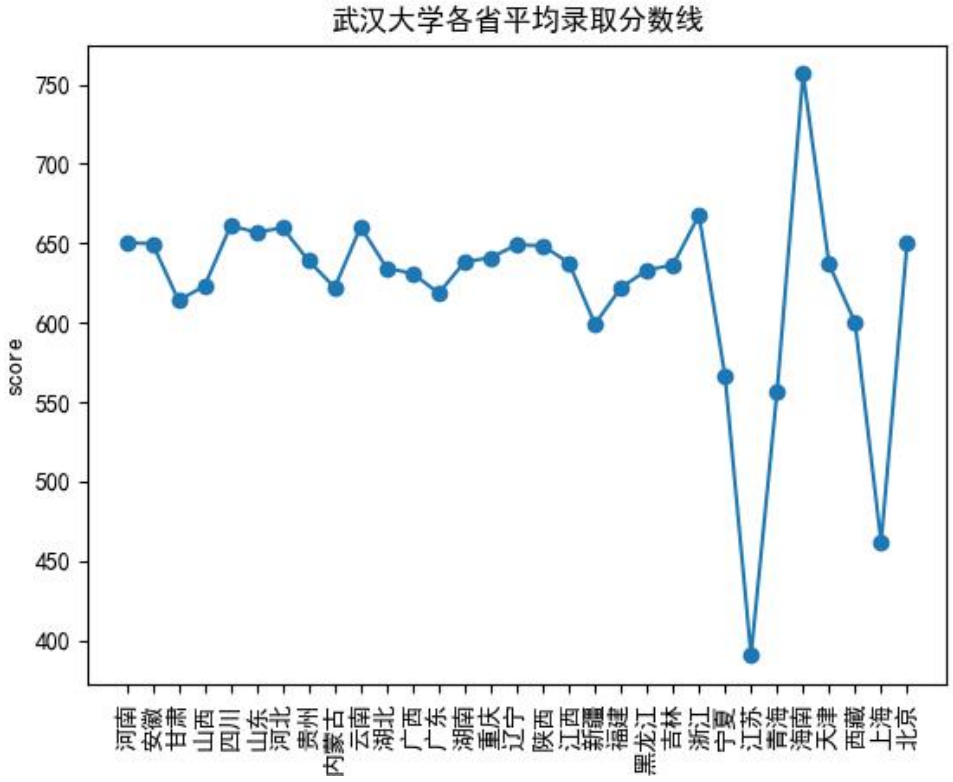

图 17　武汉大学在每个省份招生的录取平均分数折线图

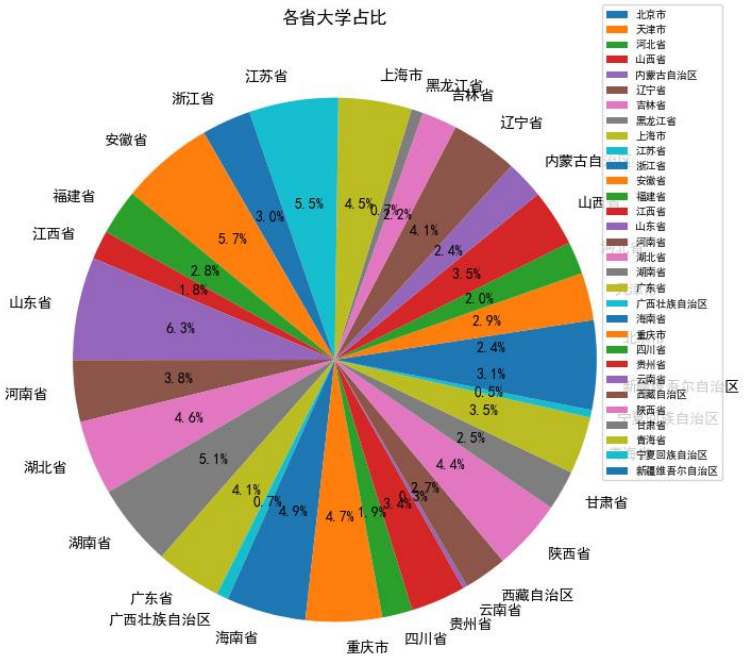

图 18　各省大学占比饼图

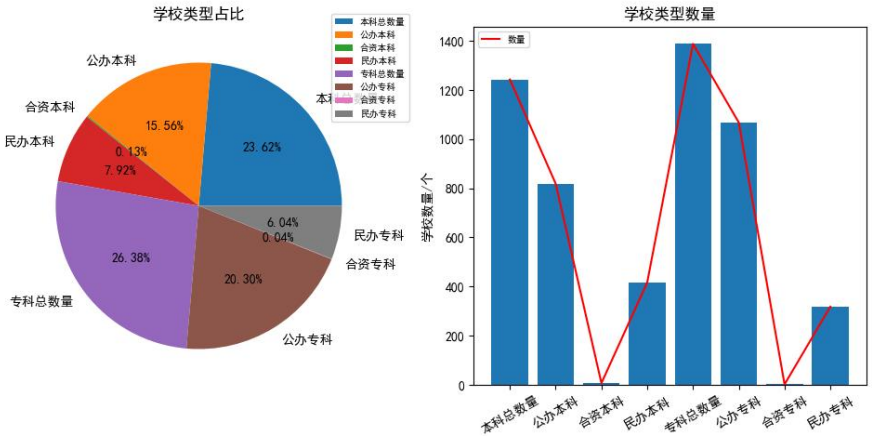

图 19　学校类型的数量占比饼图、柱状图和折线图

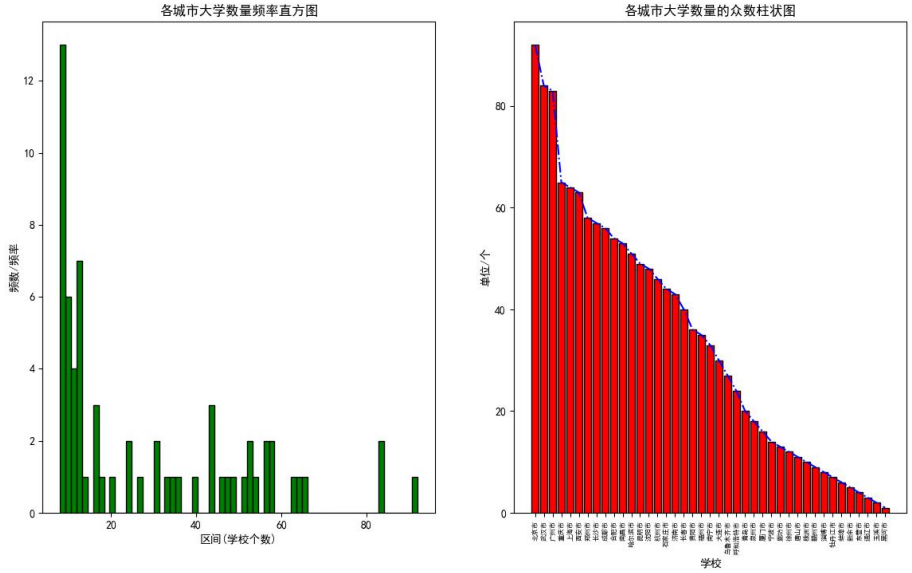

图 20 各城市大学数量频率直方图 && 各城市大学数量的众数柱状图

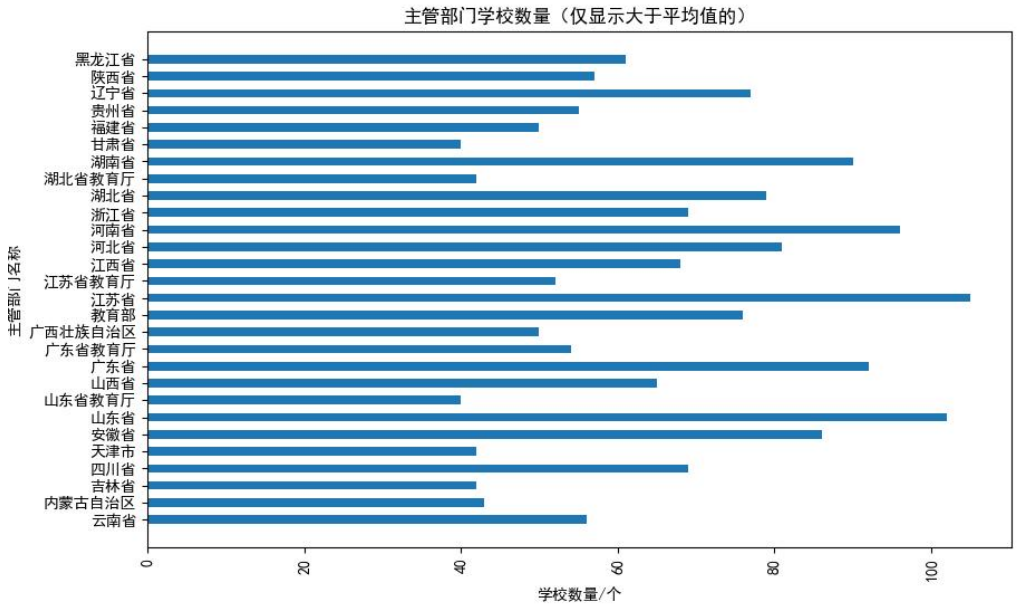

图 21 主管部门和办学层级之间的数量关系条形图

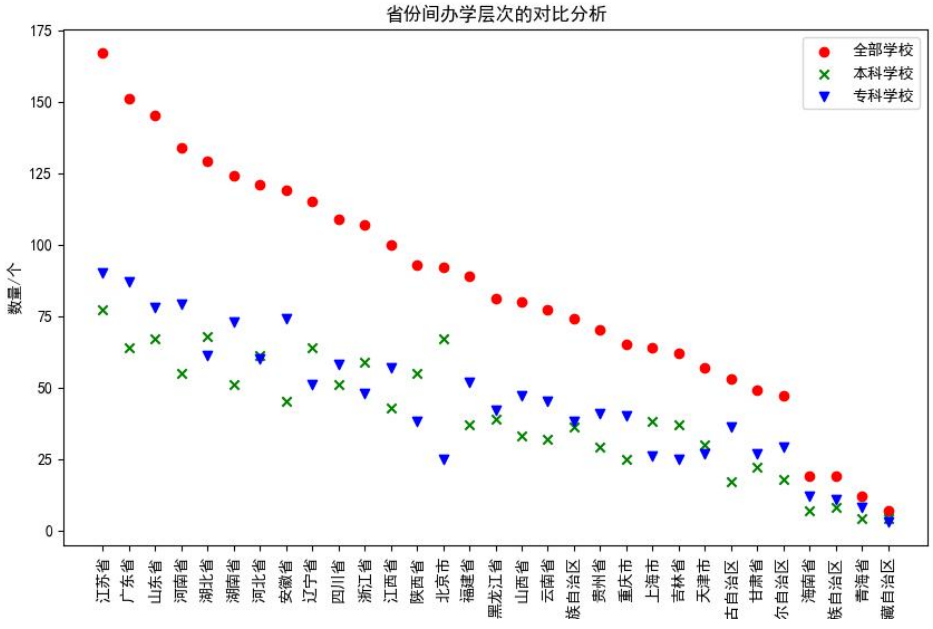

图 22 省份间办学层次的对比分析散点图

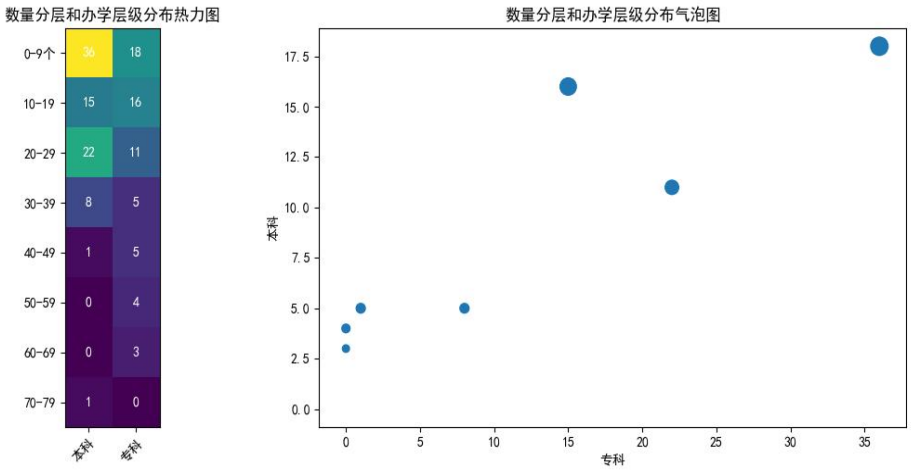

图 23 各部门本科，专科数量分层热力图和气泡图

四、结论

总结：

1.2018 年时，清华大学录取分数线较往年偏低；而北京大学录取分数线几乎每年持平，2019 年因为试卷难度较大，分数线较低。

2.2017 年和 2019 年因为试卷难度较大，故武大华科在鄂录取分数线较低。

3.由于上海江苏的招录模式和其它地区不同，故在图中录取分数偏低；海南因为地理位置偏远，招录模式需要与当地自然灾害发生频率相适应，故在图中显示为异常点。

4.中国大学数量的分布呈现出明显的地域差异，东部沿海省份和直辖市拥有更多的高等学校，而西部内陆省份则相对较少。

5.中国大学数量的分布也反映了经济发展水平和人口规模的影响，一般来说，经济发达、人口密集的地区有更多的需求和能力建设高等教育机构。

6.中国大学数量的分布还与历史文化传统有关，一些拥有悠久教育历史和文化底蕴的地区也有较多的高等学校。

预测分析：

首先，对于高考录取分数，目前的趋势是人才流向大城市和一线城市的现象越来越明显，导致这些地区的考生录取分数普遍较高。同时，由于高考科目和命题的变化，不同省份录取分数的变化也有所体现。因此，我预测未来的高考录取分数将会呈现分化趋势，高分段与低分段的差距将会进一步扩大。

其次，对于各省大学分布的预测，预计未来几年会有以下趋势：

1. 一流高校的优势将进一步加强，而二流高校将面临更大的竞争压力

2. 各省本土高校会越来越重视招收本地优秀学生，尤其是省级重点大学。

3. 一些省份可能会加大对人才引进的力度，吸引更多的高水平人才来到当地高校从事科研工作。

4. 由于教育资源的不均衡分布，一些地区的高校将面临较大困难。这些地区可能需要更多的政策支持和资金投入，以提高当地高校的教学水平和科研实力。

总之，未来几年的高考录取分数和各省大学分布都会受到多种因素的影响。但通过分析历史数据和当前趋势，我们可以预测到一些可能的情况，这将为高中学生和考生们提供有用的参考信息。

五、小结

我个人非常喜欢使用 Python 进行数据分析，因为 Python 有很多特点，如灵活的字符串操作、缩进的方式简单明了、语法简单等。同时，Python 也是一种高级语言，具有解释型语言的特点，具有可移植性、面向对象、强大的功能、开源、可扩展性、丰富的库和规范的代码等优点。这些特点使得 Python 在数据分析领域非常受欢迎。

在进行 Python 数据分析时，我通常会按照以下步骤进行：

1. 数据预处理：使用 Pandas 库对数据进行清洗、去重、格式转换和数据筛选等操作，使数据准备好进行后续的分析。

2. 数据可视化：使用 Matplotlib 可视化库进行数据可视化，绘制各种图表和图形，以更好地了解数据的分布、趋势、相关性和异常值等信息。

3. 数据分析：使用 Python 进行统计分析、探索性分析、假设检验和回归分析等操作，以回答各种问题和解决实际问题。

在这次课程设计中，我使用 Python 读取了一个数据文件，然后利用数据进行数图结合的分析。通过这次实践，我深入了解了 Python 数据分析的基础知识和操作技巧，也发现了自己的不足之处。我认为，要想学好 Python 数据分析，重在实践，需要不断上机操作，不断修正和改进自己的代码，才能更好地掌握相关技能。

六、参考文献

1.邱锡鹏，《Python 数据科学》，人民邮电出版社，2017 年

2.李航，《统计学习方法》，清华大学出版社， 2012 年

3.张艺勤,《Python 数据可视化实战》人民邮电出版社，2018 年

4.杨洋，《Python 编程快速上手》，人民邮电出版社，2019 年