项目简介：

爬虫

1.爬取近三年(2021-2019)对外经济贸易大学在各省份的录取人数，录取分数等信息；

文件处理

2.爬取数据存储至json文件，筛选出各个省份各个年份的数据，分别导出至不同csv文件，便于处理；

数据清洗

3.对不提供录取平均分（2021年部分省份的数据）的数据，将其平均分设置为“暂无数据”，删去不需要的条目，无数据条目，无效条目，保留官网对外展示的条目；

数据可视化

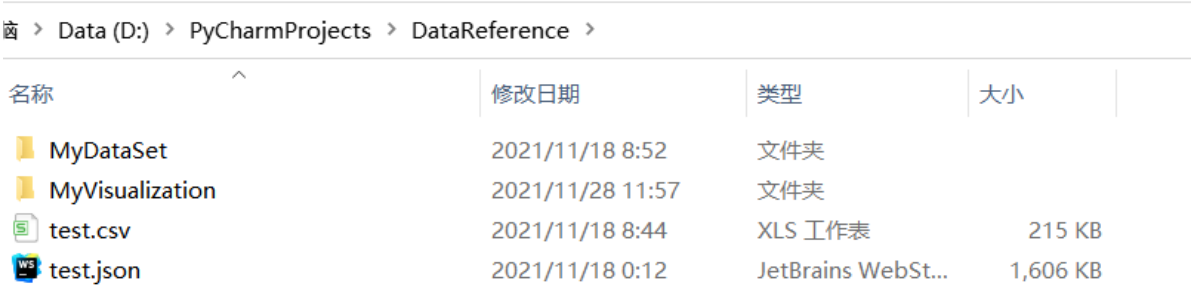
4.最后对录取人数，录取分数做可视化处理。

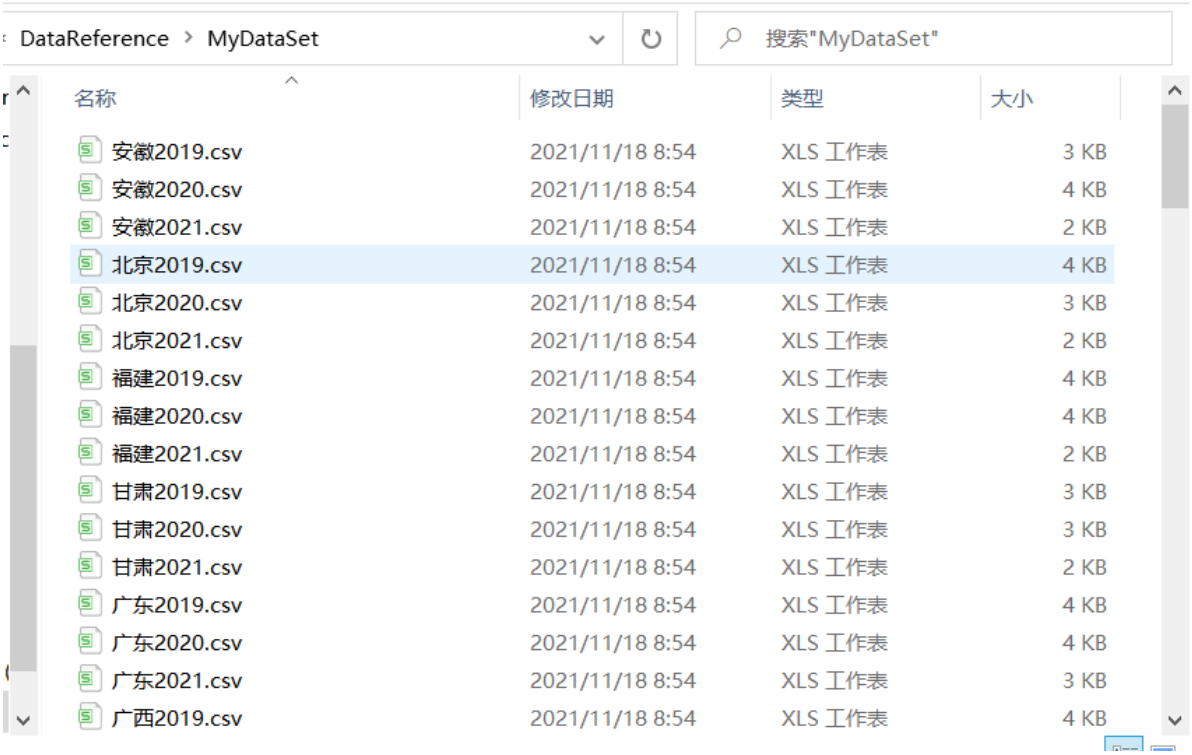
注：虽然没有使用函数进行封装，但是设计上是按模块设计的，程序可以一块一块单独运行。

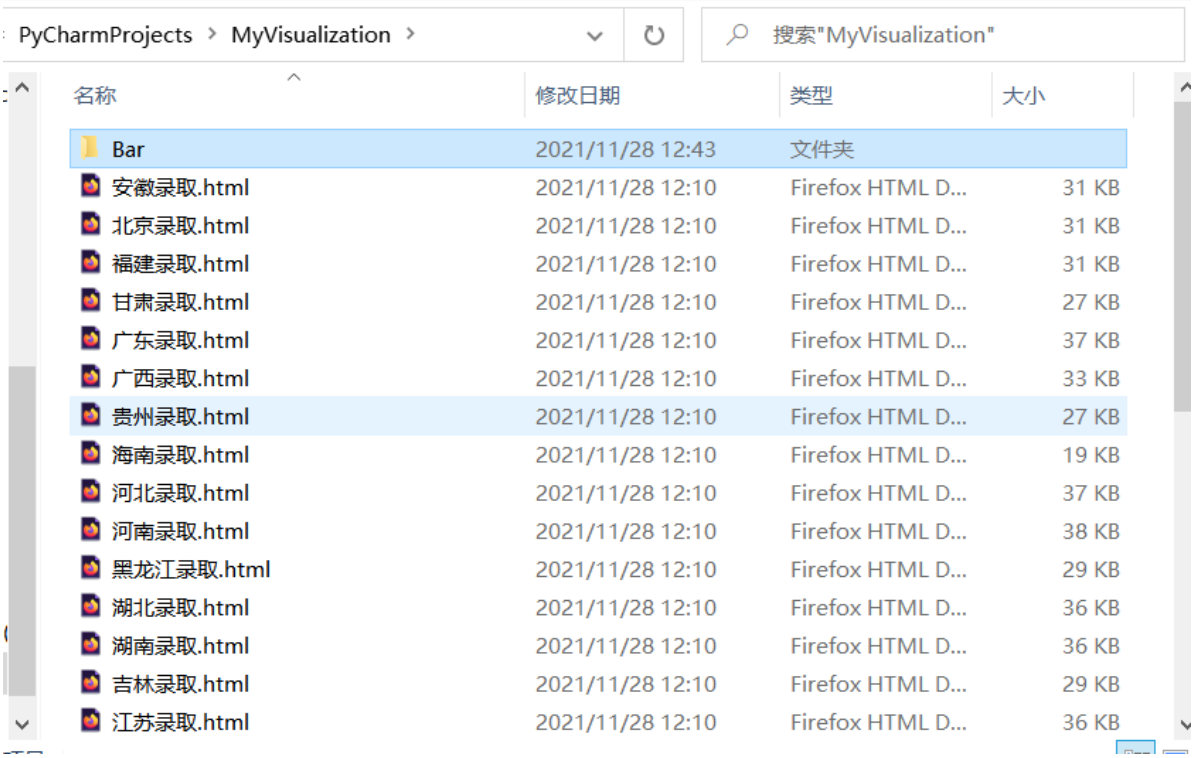
Priorities:

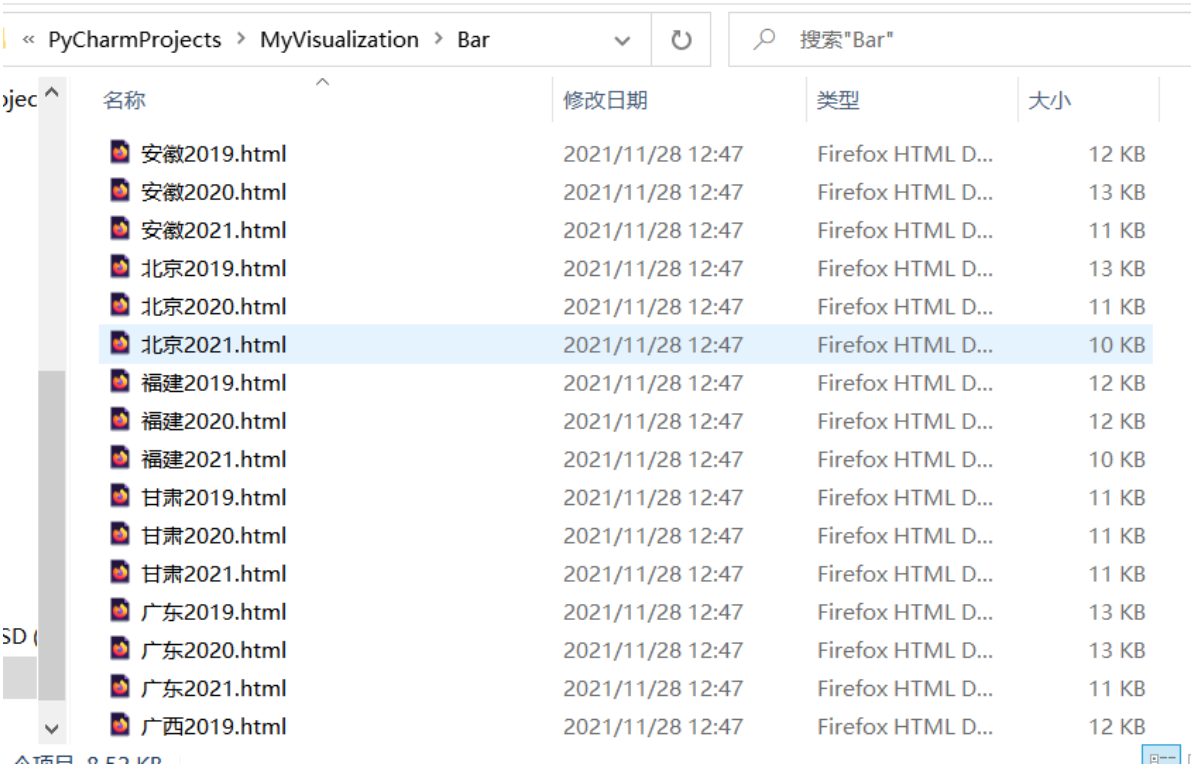
测试环境：PyCharm2021.2.2 python3.8.12(anaconda集成)

DataReference文件夹给出参考生成文件，按照程序正常进行，应该有以下文件存于根目录(py文件所在目录)生成，注意的是，不能单独将py文件直接放在某一文件夹下运行，生成DataSet和Visualization的时候，文件夹是不能自动创建的，应该将py文件与MyDataSet和MyVisualization（含一个bar文件夹）文件夹一起放在同级目录下运行。




DataReference › MyDataSet
搜索"MyDataSet"
名称 修改日期 类型 大小
安徽2019.csv 2021/11/18 8:54 XLS 工作表 3 KB
安徽2020.csv 2021/11/18 8:54 XLS 工作表 4 KB
安徽2021.csv 2021/11/18 8:54 XLS 工作表 2 KB
北京2019.csv 2021/11/18 8:54 XLS 工作表 4 KB
北京2020.csv 2021/11/18 8:54 XLS 工作表 3 KB
北京2021.csv 2021/11/18 8:54 XLS 工作表 2 KB
福建2019.csv 2021/11/18 8:54 XLS 工作表 4 KB
福建2020.csv 2021/11/18 8:54 XLS 工作表 4 KB
福建2021.csv 2021/11/18 8:54 XLS 工作表 2 KB
甘肃2019.csv 2021/11/18 8:54 XLS 工作表 3 KB
甘肃2020.csv 2021/11/18 8:54 XLS 工作表 3 KB
甘肃2021.csv 2021/11/18 8:54 XLS 工作表 2 KB
广东2019.csv 2021/11/18 8:54 XLS 工作表 4 KB
广东2020.csv 2021/11/18 8:54 XLS 工作表 4 KB
广东2021.csv 2021/11/18 8:54 XLS 工作表 3 KB
广西2019.csv 2021/11/18 8:54 XLS 工作表 4 KB


PyCharmProjects › MyVisualization ›
搜索"MyVisualization"
名称 修改日期 类型 大小
Bar 2021/11/28 12:43 文件夹
安徽录取.html 2021/11/28 12:10 Firefox HTML D... 31 KB
北京录取.html 2021/11/28 12:10 Firefox HTML D... 31 KB
福建录取.html 2021/11/28 12:10 Firefox HTML D... 31 KB
甘肃录取.html 2021/11/28 12:10 Firefox HTML D... 27 KB
广东录取.html 2021/11/28 12:10 Firefox HTML D... 37 KB
广西录取.html 2021/11/28 12:10 Firefox HTML D... 33 KB
贵州录取.html 2021/11/28 12:10 Firefox HTML D... 27 KB
海南录取.html 2021/11/28 12:10 Firefox HTML D... 19 KB
河北录取.html 2021/11/28 12:10 Firefox HTML D... 37 KB
河南录取.html 2021/11/28 12:10 Firefox HTML D... 38 KB
黑龙江录取.html 2021/11/28 12:10 Firefox HTML D... 29 KB
湖北录取.html 2021/11/28 12:10 Firefox HTML D... 36 KB
湖南录取.html 2021/11/28 12:10 Firefox HTML D... 36 KB
吉林录取.html 2021/11/28 12:10 Firefox HTML D... 29 KB
江苏录取.html 2021/11/28 12:10 Firefox HTML D... 36 KB



使用的库文件：

版本限制：

代码中使用了f字符串，如果python版本低于3.6，可能无法运行。

库需求：requests，pandas，pyecharts

```
import time # python内置 用来sleep设置爬虫间隔
import requests # requests爬虫核心，获取对应URL内容，非自带
import json # python内置 处理json文件
import pandas as pd # pandas数据清洗，筛选，导出，读取，处理
'''
pyecharts库，可视化核心，用于作可交互的可视化，想了解更多可以看看echarts
最后可视化文件是html文件（js交互）
'''
from pyecharts.charts import Timeline, Pie, Bar
from pyecharts import options as opts
from pyecharts.options import PieLabelLineOpts
```

数据分析结果与可视化建议直接打开html网页看，网页是可交互的

代码直接使用py文件，下方代码按模块分析解释。

查看文档：

设置了目录，最终pdf中应该有如左边的目录。

项目简介：

- 爬虫
- 文件处理
- **数据清洗**
- 数据可视化
- Priorities:
- 使用的库文件：
  - 版本限制：

Part1 爬虫准备

Part2 爬虫（对比了中国教育在线爬虫的数据）

Part3 数据清洗

Part4 筛选数据，导出为不同csv文件

Part5 数据集完整性验证

Part6 饼状图可视化（各省份（年份按时间轴）各专业录取人数）

Part7 柱状图可视化（录取最高分最低分）

可视化交互：

Q && A

Part1 爬虫准备

准备待爬取的数据目标，年份选取近三年（2021-2019），省份选取官网提供数据的所有省份，存于列表中，用于后续迭代遍历。

```
city_name_set = [
    "北京",
    "天津",
    "河北",
    "山西",
    "内蒙古",
    "辽宁",
    "吉林",
    "黑龙江",
    "上海",
    "江苏",
    "浙江",
    "安徽",
    "福建",
```

```
    "江西",
    "山东",
    "河南",
    "湖北",
    "湖南",
    "广东",
    "广西",
    "海南",
    "重庆",
    "四川",
    "贵州",
    "云南",
    "西藏",
    "陕西",
    "甘肃",
    "青海",
    "宁夏",
    "新疆"]
year_set = [2021, 2020, 2019]
```

Part2 爬虫（对比了中国教育在线爬虫的数据）

先确定目标URL，对每个省份每个城市发送不完全的URL，获取网页上json文件中的科类条目信息（不同省份不同年份不一样，自行枚举过于繁杂），获取后对条目内容进行筛选，获取完整URL，此时再爬取获取目标数据（分一次普通招生，一次国家专项），其余详见注释。

```
'''
爬虫，爬取近三年(2021-2019)上述所有(提供数据的)城市
录取分数与专业
注：官网中2021年多地并没有提供平均数数据，而中国教育在线
中甚至没有2021年的数据，在2020-2019年的数据比较中，数据
是一致的，但区别在于官方提供的平均分中有小数，中国教育在线
提供没有小数，如分数  官网：649.8  中国教育在线：649
因为从官网直接获取的数据，故以官网的为准，不进行清洗
'''
with open('test.json','w',encoding="utf-8") as fw:
    for year in year_set:
        for city in city_name_set:
            science_class = requests.get(

 f"https://admin.zhinengdayi.com/front/enroll/getMajorSelectChange?cityName=
{city}&sCode=NLGYFE&year={year}")
            '''
            爬取各城市各年度的科类条目
            (不同城市不同年度有：综合改革，理工/物理类 文史/历史类 理科 文科)
            '''
            science_class.encoding='utf-8'
            time.sleep(0.3) #设置爬取间隔
            for sc in science_class:

                sc_str = sc.decode(encoding='utf-8')
                '''
                提取科类条目内容，获得完整目标URL
                '''
```

```
                if "scienceList" in sc_str:
                    pattern = sc_str.split(",")[1].split('[')[1].split(']')[0]
                    if "理工/物理类" in sc_str.split(',')[2]:
                        pattern2 = "理工/物理类"
                    elif "理科" in sc_str.split(',')[2]:
                        pattern2 = "理科"
                    else:
                        pattern2 = ""
                else:
                    continue
                pattern = eval(pattern)
                '''
                爬取普通招生
                '''
                r = requests.get(

 f"https://admin.zhinengdayi.com/front/enroll/findMajorScoreCompareList?
sCode=NLGYFE&cityName={city}&year={year}&scienceClass={pattern}&type=普通招生
&batch=")
                r.encoding = "utf-8"
                '''
                打印测试 & 保存至json
                '''
                print(r.json())
                for i in r.json()['list']:
                    fw.write("\n")
                    json.dump(i, fw, ensure_ascii=False)
                '''
                爬取国家专项
                '''
                r = requests.get(

 f"https://admin.zhinengdayi.com/front/enroll/findMajorScoreCompareList?
sCode=NLGYFE&cityName={city}&year={year}&scienceClass={pattern}&type=国家专项
&batch=")
                r.encoding = "utf-8"
                print(r.json())
                if r.json()['list']:
                    for i in r.json()['list']:
                        fw.write("\n")
                        json.dump(i, fw, ensure_ascii=False)

                if pattern2 != "":
                    r = requests.get(

 f"https://admin.zhinengdayi.com/front/enroll/findMajorScoreCompareList?
sCode=NLGYFE&cityName={city}&year={year}&scienceClass={pattern2}&type=普通招生
&batch=")
                    r.encoding = "utf-8"
                    for j in r.json()['list']:
                        fw.write("\n")
                        json.dump(j, fw, ensure_ascii=False)

                    r = requests.get(

 f"https://admin.zhinengdayi.com/front/enroll/findMajorScoreCompareList?
sCode=NLGYFE&cityName={city}&year={year}&scienceClass={pattern2}&type=国家专项
&batch=")
```

```
                    r.encoding = "utf-8"
                    if r.json()['list']:
                        for j in r.json()['list']:
                            fw.write("\n")
                            json.dump(j, fw, ensure_ascii=False)

print("finished!!!")
```

Part3 数据清洗

将原本写入json文件的数据进行清洗，删去不需要的条目，导出为csv文件，核实数据。

```
'''
读取json数据
将NaN数据清洗为"暂无数据"
删去'majorCategory','sCode','cityId'等其他无效条目
即无数据条目或不需要的条目
'''
with open('test.json','r',encoding='utf-8') as f:
     df = pd.read_json(f,lines=True)
     df.avgScore = df.avgScore.fillna("暂无数据")
     df.drop('majorCategory',axis=1,inplace=True)
     df.drop('cityId',axis=1,inplace=True)
     df.drop('sCode',axis=1,inplace=True)
     # df.drop(index=[i for i in range(13,42)],axis=1,inplace=True)
     store_cols = df.columns.values[11:]
     df.drop(columns=[col for col in store_cols],axis=1,inplace=True)
     df.to_csv('test.csv')
print("finished!!!")
```

Part4 筛选数据，导出为不同csv文件

```
'''
按照城市名与年份筛选数据
输出相应城市 相应年份下 全部录取情况
'''
with open('test.json', 'r', encoding='utf-8') as f:
    df = pd.read_csv("test.csv")
    for year in year_set:
        for city in city_name_set:
            temp = df[(df.year == year) & (df.cityName == city)]
            temp.to_csv('MyDataSet/'+city + str(year) + '.csv')
```

Part5 数据集完整性验证

（其他条目打开看就行，不需要多加代码）：

```
'''
输出城市数目，
城市数31*年份3，
应为93份数据，
验证数据集完整性
'''
print(len(city_name_set))
```

Part6 饼状图可视化（各省份（年份按时间轴）各专业录取人数）

```
for city in city_name_set:
    data_test = pd.read_csv(f'MyDataSet/{city}2021.csv')
    data_test2 = pd.read_csv(f'MyDataSet/{city}2020.csv')
    data_test3 = pd.read_csv(f'MyDataSet/{city}2019.csv')
    data_test_set = [data_test,data_test2,data_test3]
    timeline = Timeline(init_opts=opts.InitOpts(width="1680px", height='720px'))
    for i in [2021,2020,2019]:
        pie = (Pie(init_opts=opts.InitOpts(width="1680px", height='1080px'))
         .add("", [list(z) for z in zip(data_test_set[i-2019]['majorName'],
data_test_set[i-2019]['enrollNum'])],
              center=['30%', '50%'],
              radius=['10%', '50%'],

              )
         .set_global_opts(
            title_opts=opts.TitleOpts(title=f"{city}{i}年度各专业录取人数饼状图"),
            legend_opts=opts.LegendOpts(type_='scroll', pos_left='70%',
orient='vertical'),

        )
         .set_series_opts(label_opts=opts.LabelOpts(formatter='{b}:{c}')))
        timeline.add(pie,f'{i}年')
    timeline.render(f"MyVisualization/{city}录取.html")
```

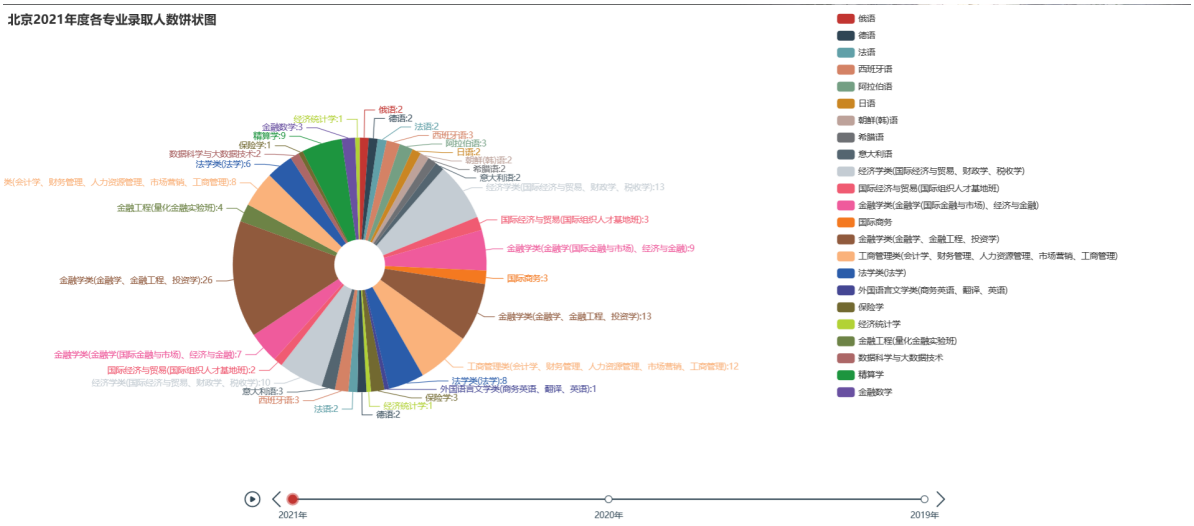
Part7 柱状图可视化（录取最高分最低分）

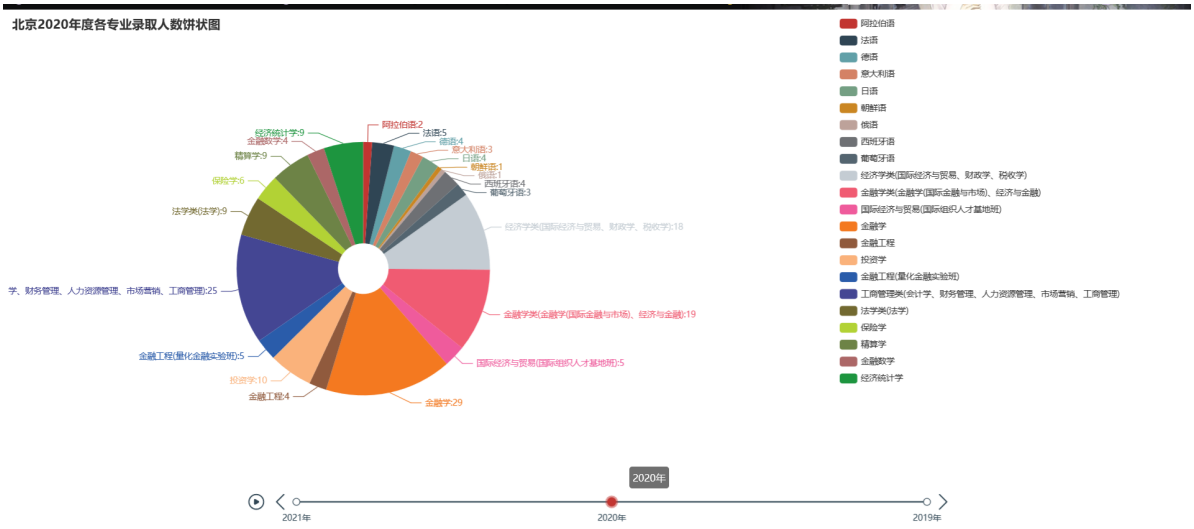
```
for city in city_name_set:

    data_test = pd.read_csv(f'MyDataSet/{city}2021.csv')
    data_test2 = pd.read_csv(f'MyDataSet/{city}2020.csv')
    data_test3 = pd.read_csv(f'MyDataSet/{city}2019.csv')
    data_test_set = [data_test,data_test2,data_test3]

    for i in [2019,2020,2021]:
        bar = (Bar(init_opts=opts.InitOpts(width='1080px',height='720px'))
        .add_xaxis(list(data_test_set[i-2019]['majorName']))
        .add_yaxis('最低分',list(data_test_set[i-2019]
['lowScore']),stack="stack1")
        .add_yaxis('最高分',list(data_test_set[i-2019]
['hightScore']),stack="stack1")
        .set_global_opts(
            xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=-15)),
```

```
            title_opts=opts.TitleOpts(title="Bar-旋转X轴标签", subtitle="解决标签名字过长的问题"),
            datazoom_opts=[opts.DataZoomOpts()],
            toolbox_opts=opts.ToolboxOpts(),
        )
        )
        bar.render(f"MyVisualization/Bar/{city}{year_set[i-2019]}.html")
```

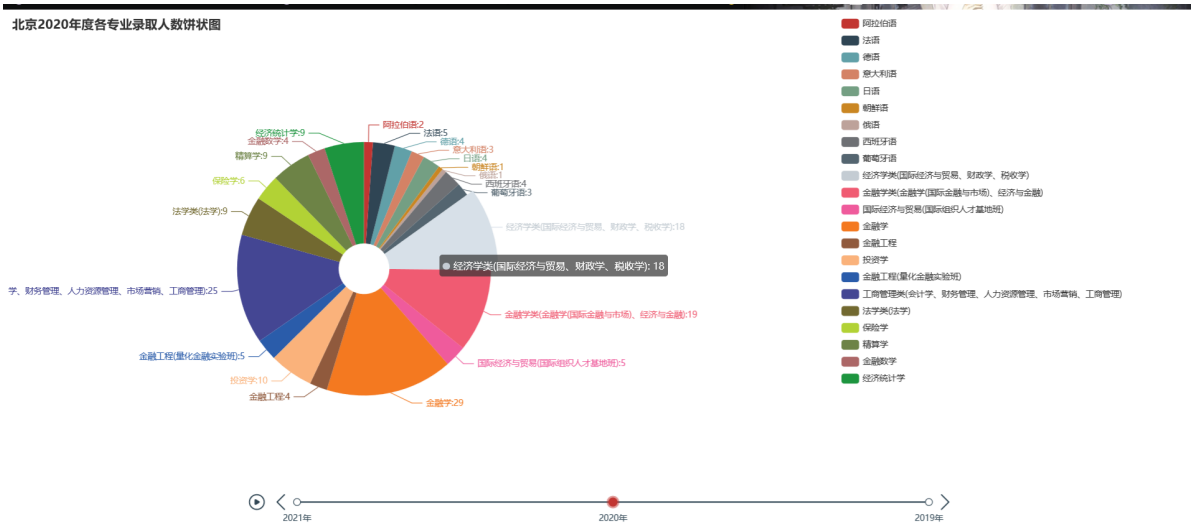
可视化交互：

切换时间轴：

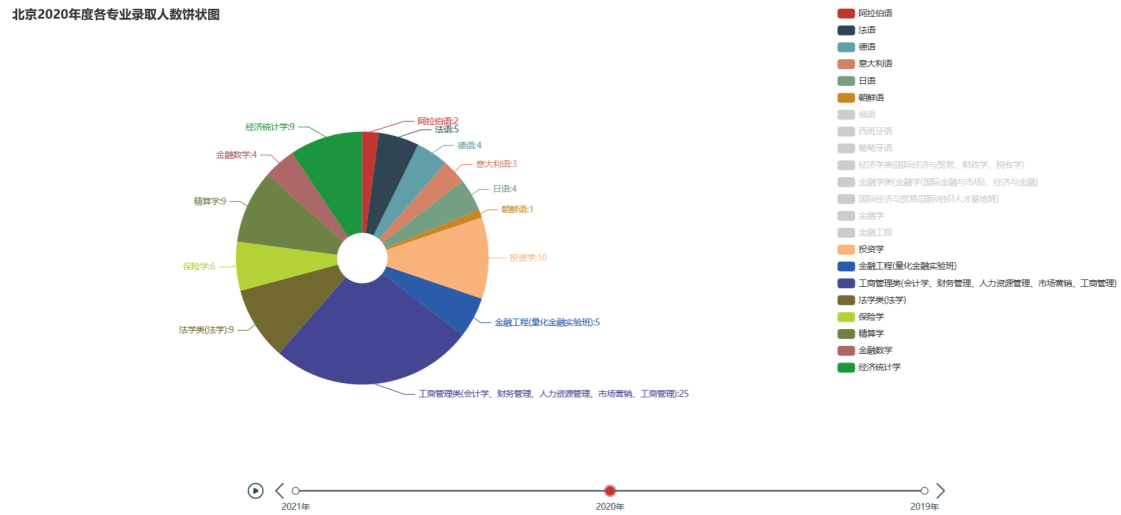




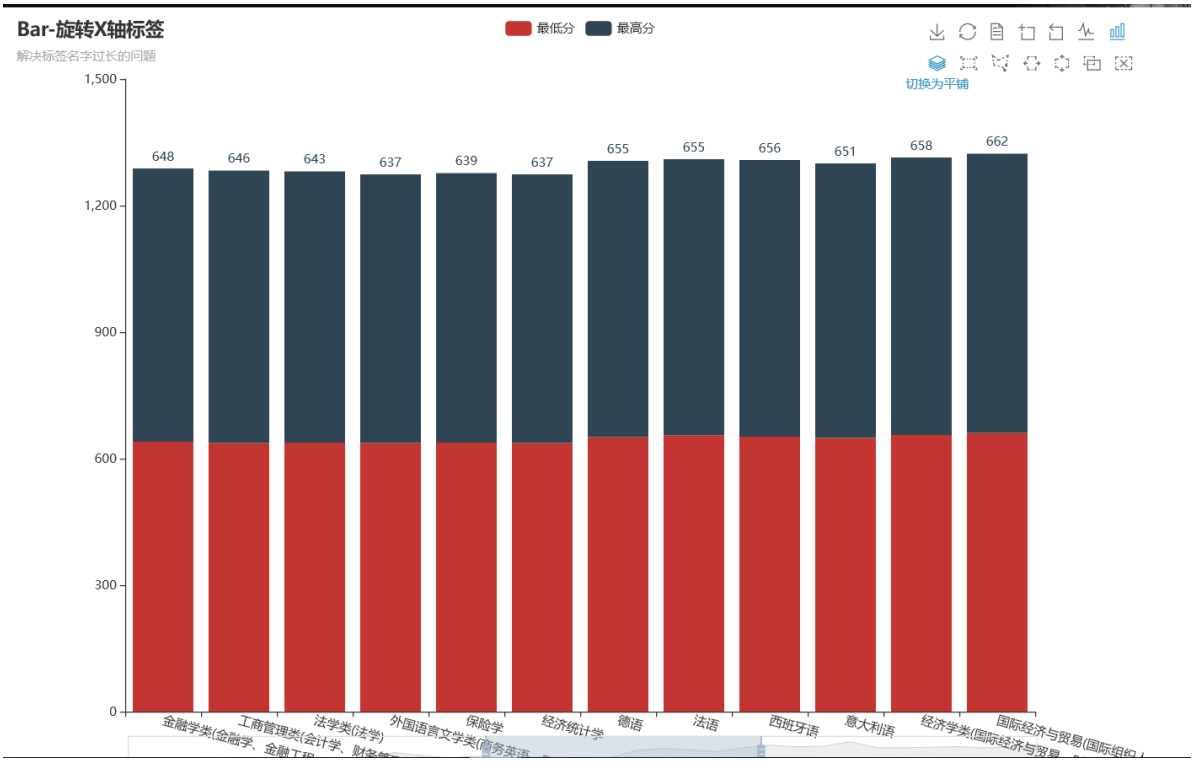
查看条目：



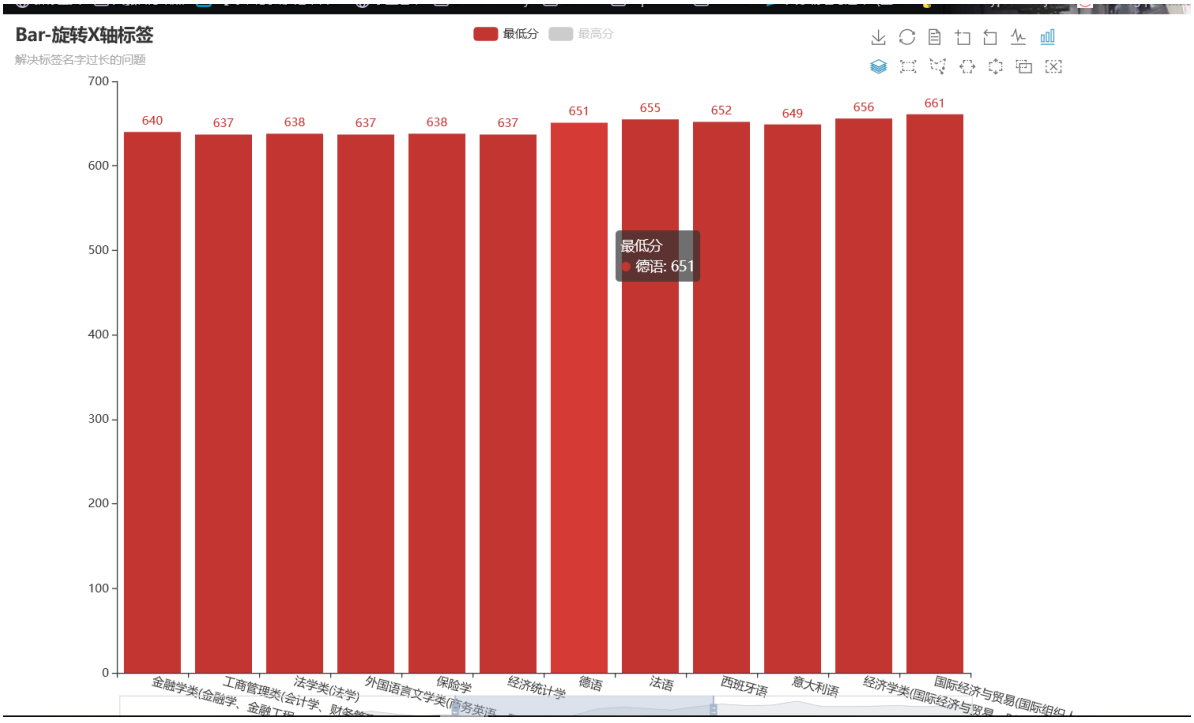
筛选条目：



右上角工具栏，鼠标悬浮查看功能，可以转为折线，改为平铺，查看数据表单，保存，截取等，点击最高分或最低分隐藏相应目标元素



下方托条可以移动（页面有限，显示不全，使用拖动条），可以使用滚轮确定ZOOM放缩



Q && A

Q:柱状图不采用时间轴?

A:柱状图采用时间轴会取消上方的工具栏，解决方法不知道，官方文档其实内容不多，很多都没有覆盖到。

Q:项目目标的选取?

A:爬虫本身是计算机院Python编程课的作业，之前写的时候顺带写了数据分析的内容。

Q:柱状图，饼状图的合并?

A:页面显示不下，且调整比较繁琐，需要像做网页一样一直运行，F5调整看结果。主要还是条目过多。

Q:所有省份可视化的合并?

A:可以利用百度地图的开发者工具做成各省份录取人数的分布，但是具体到专业做不了，意义不大。

Q:可视化使用matplotlib?

A:matplotlib不便于交互，也缺少动画效果。

Q:可视化平均分?

A:2021年部分省份并没有平均分，并且中国教育在线没有显示这部分数据，做出来效果也显得不统一，就不做了。