**数据可视化技术实验**

**实验二（2023.12.12）**

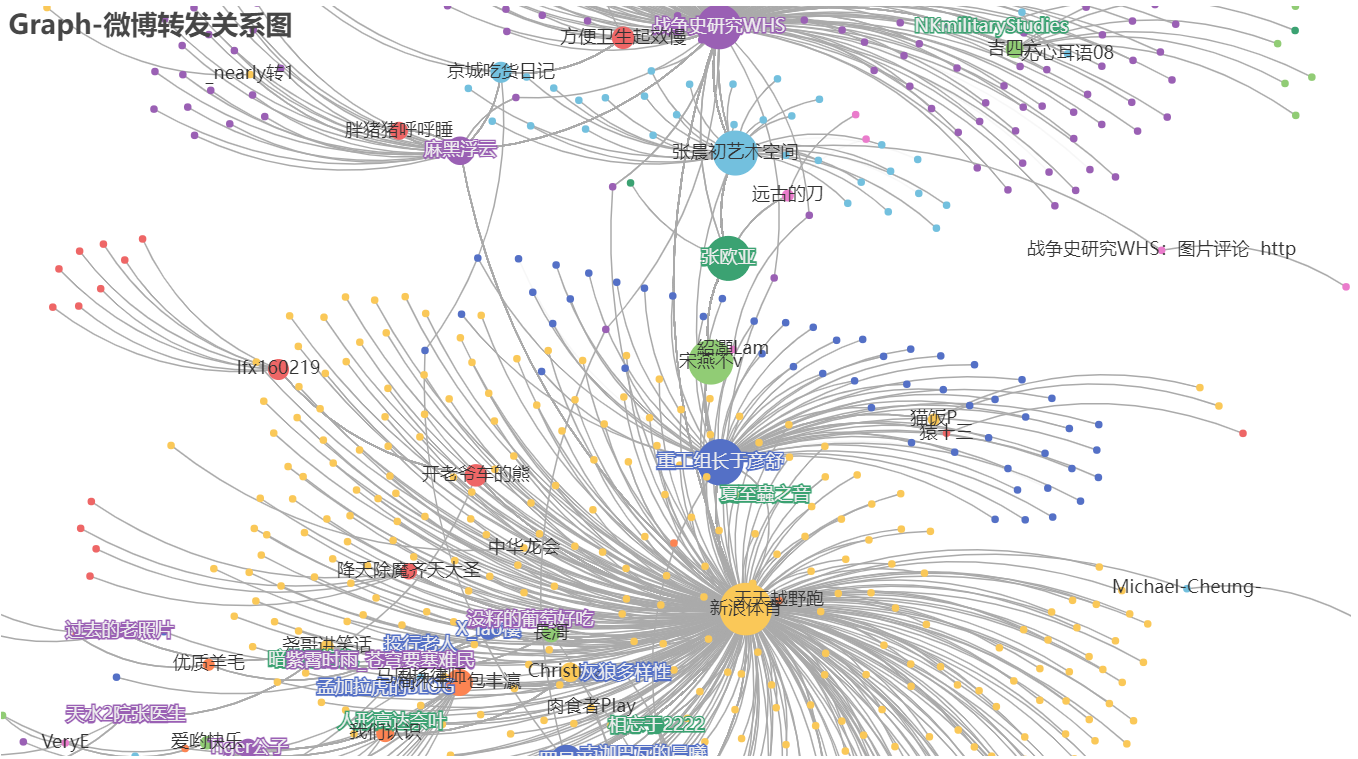
**大数据2101班 李嘉鹏 U202115652**

1. 使用weibo.json中的数据绘制关系图，观察关系图的交互操作。

代码：

**import** json  
**from** pyecharts **import** options **as** opts  
**from** pyecharts.charts **import** Graph  
  
**with** open(**r"weibo.json"**, **"r"**, encoding=**"utf-8"**) **as** f:  
 j = json.load(f)  
 nodes, links, categories, cont, mid, userl = j  
  
graph= (Graph()  
 .add(**""**, nodes, links, categories, repulsion=50,  
 linestyle\_opts=opts.LineStyleOpts(curve=0.2),  
 label\_opts=opts.LabelOpts(is\_show=**False**) )  
 .set\_global\_opts(legend\_opts=opts.LegendOpts(is\_show=**False**),  
 title\_opts=opts.TitleOpts(title=**"Graph-微博转发关系图"**)))  
graph.render(**'微博转发关系图.html'**)

结果：

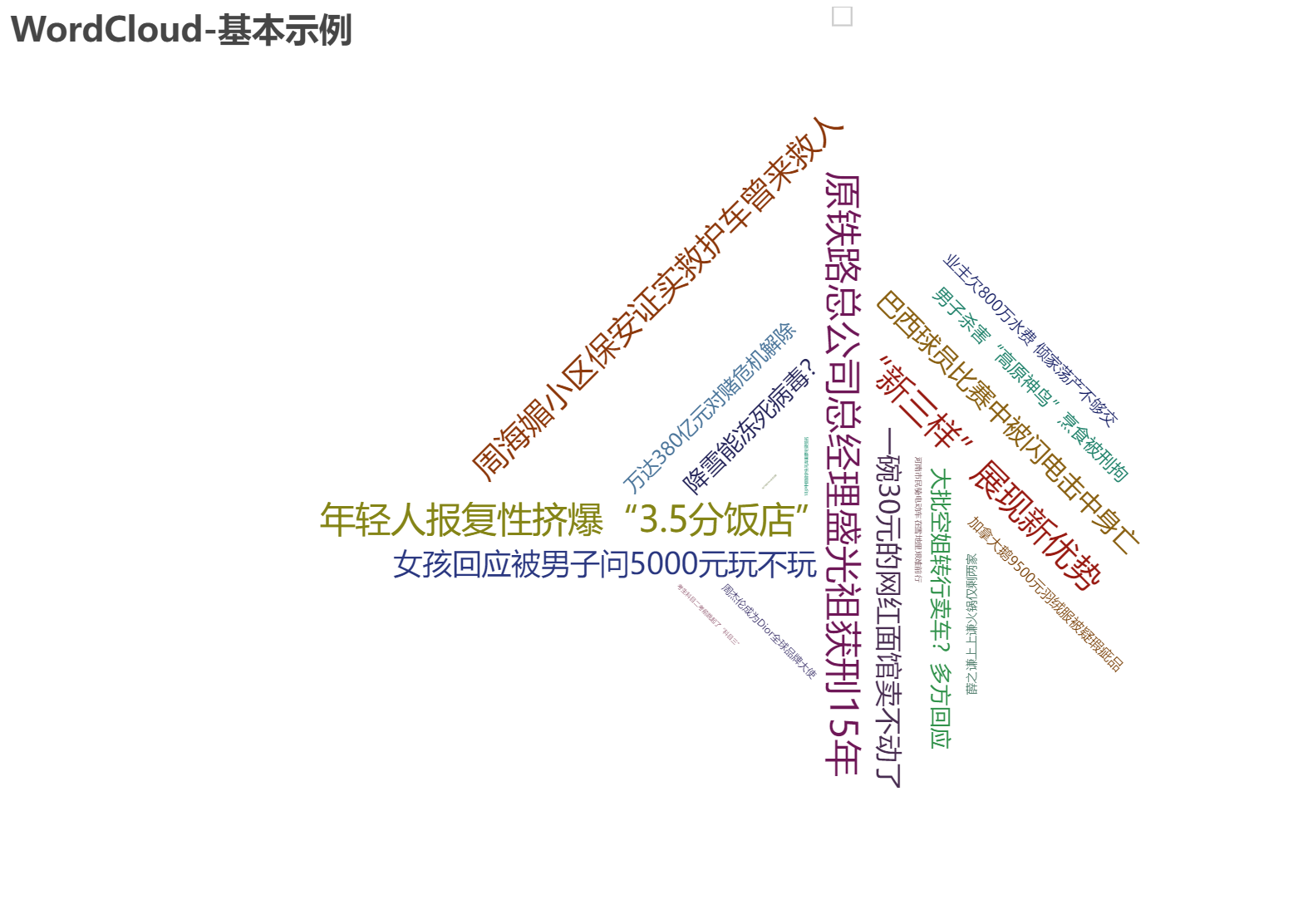


2. 搜索百度热搜榜（https://top.baidu.com/board?platform=wise）将前20个热搜词条以词云方式进行展示（热搜排名可以做为字体大小的权重，注意顺序）

代码：

**from** pyecharts **import** options **as** opts  
**from** pyecharts.charts **import** Page, WordCloud  
**from** pyecharts.globals **import** SymbolType  
  
words = [  
 (**"原铁路总公司总经理盛光祖获刑15年"**, 40),  
 (**"年轻人报复性挤爆“3.5分饭店”"**, 38),  
 (**"“新三样”展现新优势"**, 36),  
 (**"周海媚小区保安证实救护车曾来救人"**, 34),  
 (**"女孩回应被男子问5000元玩不玩"**, 32),  
 (**"一碗30元的网红面馆卖不动了"**, 30),  
 (**"巴西球员比赛中被闪电击中身亡"**, 28),  
 (**"降雪能冻死病毒？"**, 26),  
 (**"大批空姐转行卖车？多方回应"**, 24),  
 (**"万达380亿元对赌危机解除"**, 22),  
 (**"男子杀害“高原神鸟”烹食被刑拘"**, 20),  
 (**"业主欠800万水费 倾家荡产不够交"**, 18),  
 (**"加拿大鹅9500元羽绒服被疑瑕疵品"**, 16),  
 (**"薛之谦上上谦火锅仅剩两家"**, 14),  
 (**"周杰伦成为Dior全球品牌大使"**, 12),  
 (**"河南市民骑电动车在雪地里艰难前行"**, 10),  
 (**"考生科目二考前跳起了“科目三”"**, 8),  
 (**"18岁中国留学生闯库里豪宅要签名"**, 6),  
 (**"破产姐妹Max演员结婚"**, 4),  
 (**"小米：余承东发布言论与事实不符"**, 2),  
 ]  
wordcloud = (WordCloud()  
 .add(**""**, words, word\_size\_range=[0, 20])*# word\_size\_range为字体大小范围* .set\_global\_opts(title\_opts=opts.TitleOpts(title=**"WordCloud-基本示例"**)) )  
wordcloud.render(**'wordcloud.html'**)

结果：



3. 对文件student.xls中的数据进行可视化

（1）利用合适的条形图（堆叠或不堆叠）显示所有学生的总分；

（2）利用饼图展示总分前3名的分数构成；

（3）利用折线图显示英语，数分，高代，解几四门课程的成绩分布图（类似直方图，统计分数段内的人数，分数段按照每10分进行统计，然后用折线图展示）

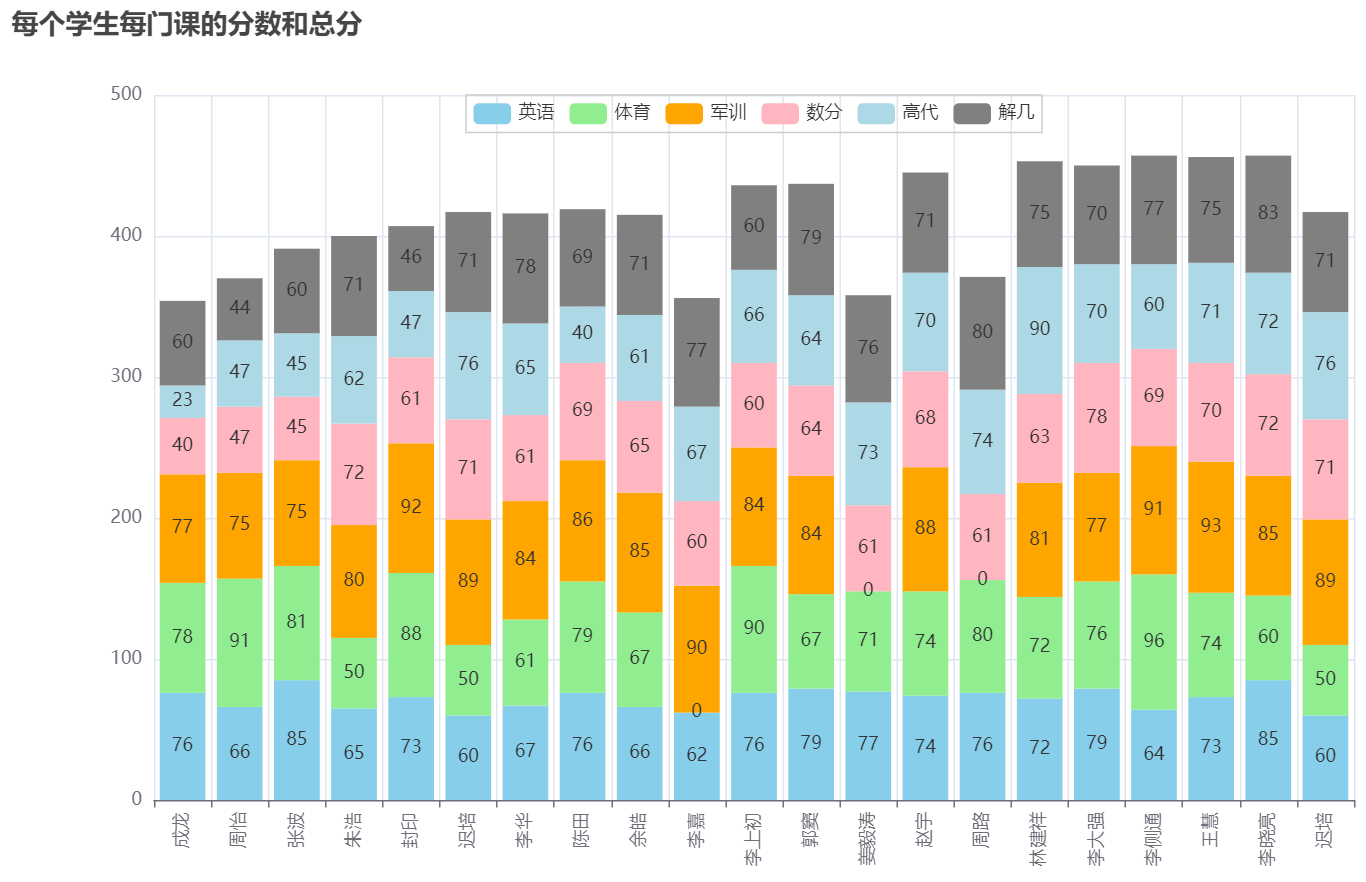
（4）利用合适的可视化图形展示男生和女生各科平均成绩的对比。

代码：

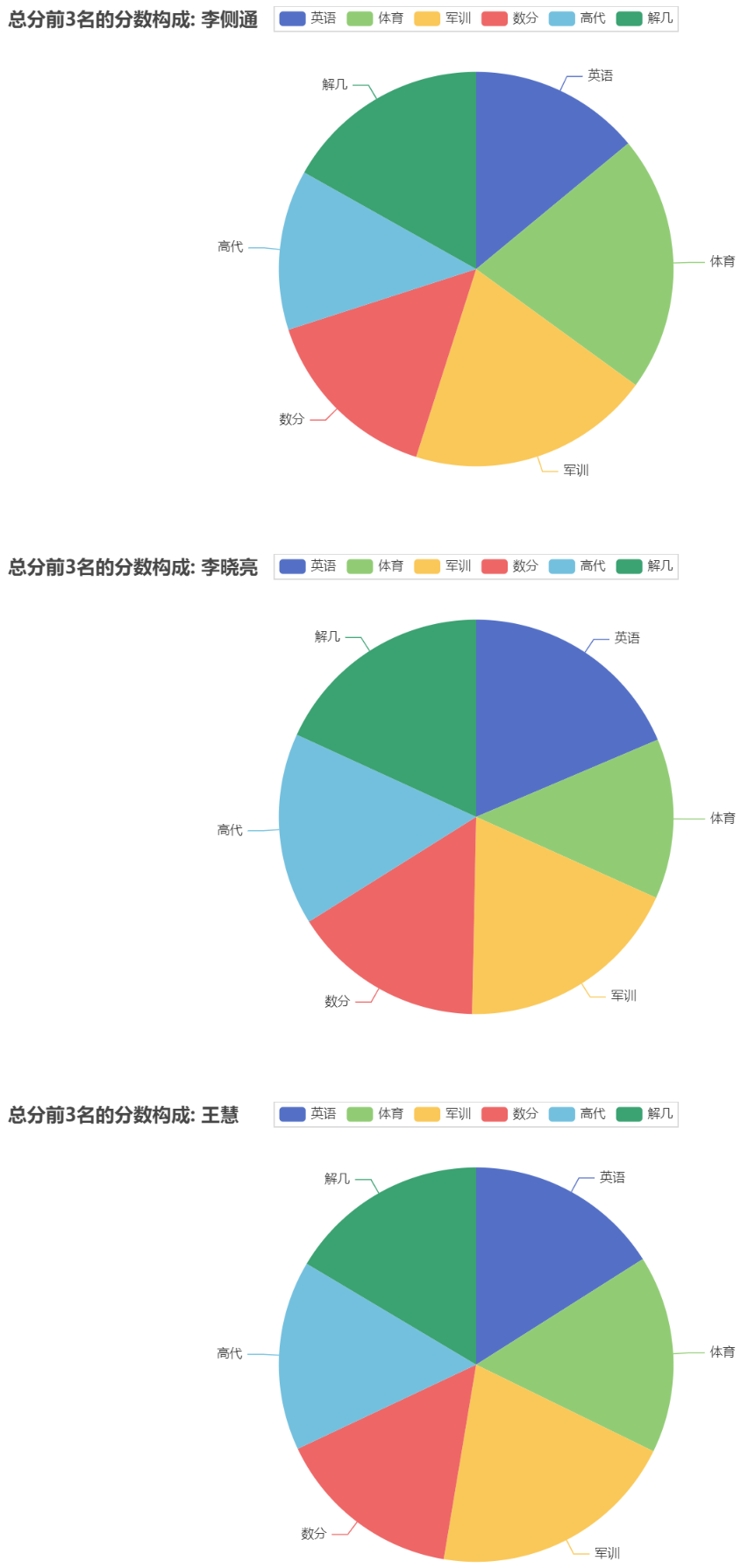
**import** pandas **as** pd  
**from** pyecharts **import** options **as** opts  
**from** pyecharts.charts **import** Bar, Pie, Line, Grid, Page  
  
data = pd.read\_excel(**'student.xls'**)  
data = data.replace([**"作弊"**, **"缺考"**], 0)  
data = data.fillna(0)  
  
data[**'总分'**] = data[[**'英语'**, **'体育'**, **'军训'**, **'数分'**, **'高代'**, **'解几'**]].sum(axis=1)  
subject\_scores = data[[**'英语'**, **'体育'**, **'军训'**, **'数分'**, **'高代'**, **'解几'**]]  
students = data[**'姓名'**]  
subjects = [**'英语'**, **'体育'**, **'军训'**, **'数分'**, **'高代'**, **'解几'**]  
colors = [**'skyblue'**, **'lightgreen'**, **'orange'**, **'lightpink'**, **'lightblue'**, **'gray'**]  
  
*# (1) 利用堆叠柱状图显示所有学生的总分*bar = (  
 Bar(init\_opts=opts.InitOpts(width=**'1000px'**, height=**'600px'**))  
 .set\_global\_opts(  
 title\_opts=opts.TitleOpts(title=**"每个学生每门课的分数和总分"**),  
 xaxis\_opts=opts.AxisOpts(axislabel\_opts=opts.LabelOpts(rotate=90)),  
 legend\_opts=opts.LegendOpts(type\_=**'scroll'**, pos\_top=**'10%'**, pos\_bottom=**'10%'**),  
 )  
)  
  
**for** i, subject **in** enumerate(subjects):  
 bar.add\_xaxis(students.tolist())  
 bar.add\_yaxis(subject, subject\_scores[subject].tolist(), stack=**'stack1'**, color=colors[i])  
  
bar.render(**'bar.html'**)  
  
*# (2) 利用饼图展示总分前3名同学的分数构成*top\_students = data.nlargest(3, **'总分'**)  
*# 使用page，把三个饼图画在一个page上*page = Page()  
**for** \_, student **in** top\_students.iterrows():  
 pie = Pie()  
 scores = [student[**'英语'**], student[**'体育'**], student[**'军训'**], student[**'数分'**], student[**'高代'**], student[**'解几'**]]  
 pie.add(student[**'姓名'**], list(zip([**'英语'**, **'体育'**, **'军训'**, **'数分'**, **'高代'**, **'解几'**], scores)))  
 pie.set\_global\_opts(title\_opts=opts.TitleOpts(title=**f"总分前3名的分数构成: {**student[**'姓名'**]**}"**))  
 page.add(pie)  
  
page.render(**'pie.html'**)  
  
*# (3) 利用折线图显示英语、数分、高代、解几四门课程的成绩分布图（统计分数段内的人数，分数段按照每10分进行统计，然后用折线图展示）  
# 这里引入了全部六门课程的成绩分布图，如果只需要保留其中一部分，则点击图例去掉无关项即可*score\_ranges = [**'0-59'**, **'60-69'**, **'70-79'**, **'80-89'**, **'90-100'**]  
course\_scores = {}  
**for** course **in** subjects:  
 score\_counts = pd.cut(data[course], bins=[0, 60, 70, 80, 90, 100], labels=score\_ranges, right=**False**).value\_counts().sort\_index()  
 course\_scores[course] = score\_counts.tolist()  
  
line = (  
 Line(init\_opts=opts.InitOpts(width=**'1000px'**, height=**'600px'**))  
 .set\_global\_opts(  
 title\_opts=opts.TitleOpts(title=**"六门课程的成绩分布"**),  
 xaxis\_opts=opts.AxisOpts(name=**"分数段范围"**),  
 yaxis\_opts=opts.AxisOpts(name=**"人数"**),  
 legend\_opts=opts.LegendOpts(type\_=**'scroll'**, pos\_top=**'10%'**, pos\_bottom=**'10%'**),  
 )  
)  
  
**for** course **in** subjects:  
 line.add\_xaxis(score\_ranges)  
 line.add\_yaxis(course, course\_scores[course], symbol=**'circle'**)  
  
line.render(**'score.html'**)  
  
*# (4) 利用分组柱状图展示男生和女生各科平均成绩的对比*male\_scores = data[data[**'性别'**] == **'男'**][[**'英语'**, **'体育'**, **'军训'**, **'数分'**, **'高代'**, **'解几'**]].mean().round(2)  
female\_scores = data[data[**'性别'**] == **'女'**][[**'英语'**, **'体育'**, **'军训'**, **'数分'**, **'高代'**, **'解几'**]].mean().round(2)  
  
bar = (  
 Bar(init\_opts=opts.InitOpts(width=**'1000px'**, height=**'600px'**))  
 .set\_global\_opts(  
 title\_opts=opts.TitleOpts(title=**"男生和女生各科平均成绩对比"**),  
 xaxis\_opts=opts.AxisOpts(name=**"科目"**),  
 yaxis\_opts=opts.AxisOpts(name=**"平均成绩"**),  
 legend\_opts=opts.LegendOpts(type\_=**'scroll'**, pos\_top=**'10%'**, pos\_bottom=**'10%'**),  
 )  
)  
  
bar.add\_xaxis(male\_scores.index.tolist())  
bar.add\_yaxis(**"男生"**, male\_scores.tolist())  
bar.add\_yaxis(**"女生"**, female\_scores.tolist())  
  
bar.render(**'average.html'**)

结果：

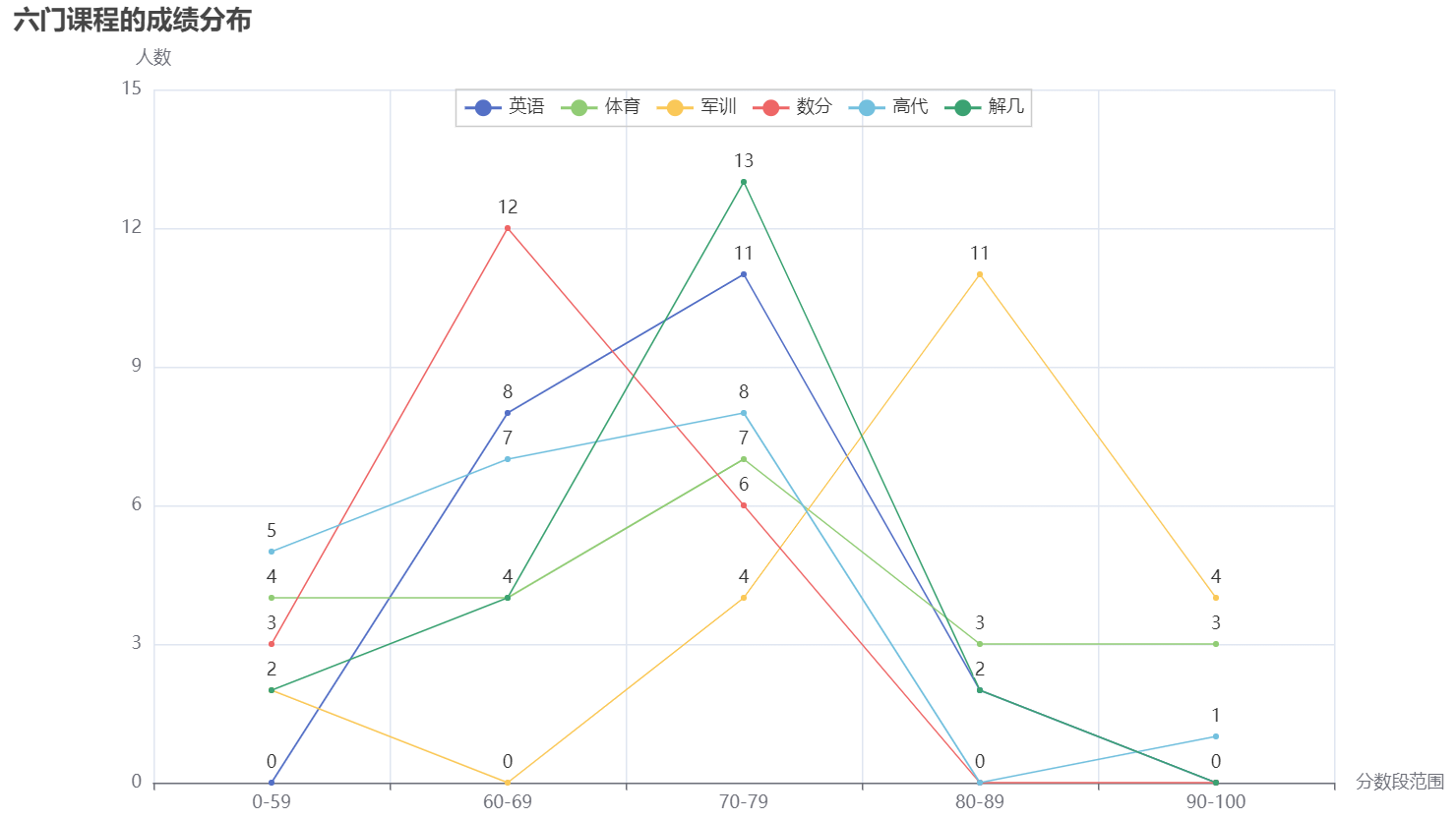
（1）



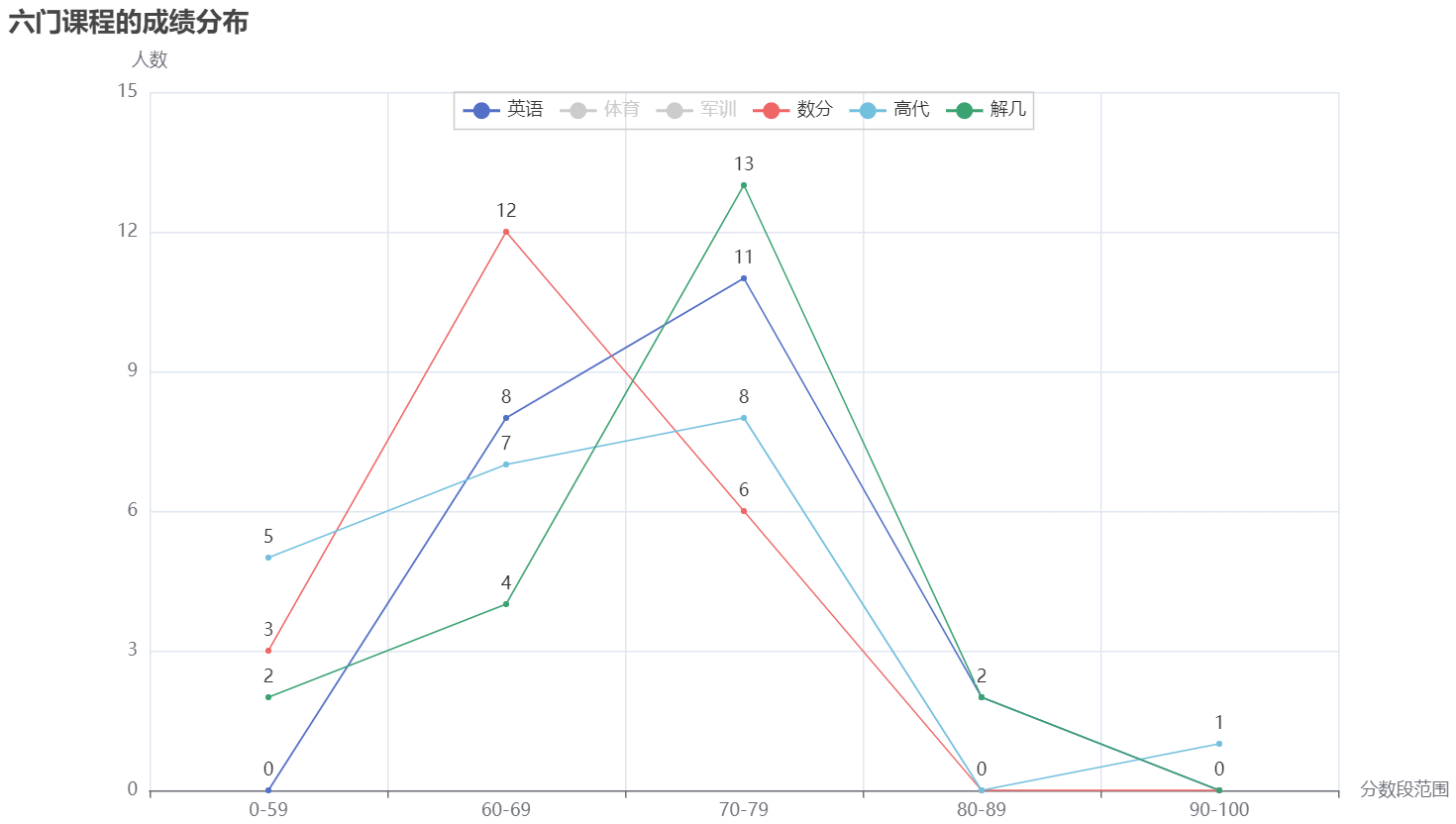
（2）



（3）



如果只保留题目中的“英语，数分，高代，解几四门课程”，则结果如下：



（4）

