|  |  |
| --- | --- |
| 成绩 |  |

数据可视化实验四

实验项目

|  |
| --- |
| 1、pyecharts |

学生姓名： 学号：

2007040830

张恒瑞

学 院：

计算机科学与技术学院

专 业：

数据科学与大数据技术专业

指导教师：

耿艳兵

2023年 6 月

一实验项目：pyecharts

|  |
| --- |
| 一、实验内容  请每位同学任选csv文件，从中选取统计变量（即csv文件里的列组）：   1. 分析数据并用折线图可视化表示 2. 分析数据并用散点图可视化 3. 分析数据并绘制四个子图，要求这四个子图均为不同图形的叠加表示，以上四个图作为子图分栏显示； 4. 分析数据并分别用普通饼图、环形饼图表示，以上两个图作为子图分栏显示 5. 分析新冠疫情数据并绘制疫情地图 6. 自己下载航班数据绘制航线地图   7、分析现有数据或自己下载数据绘制漏斗图  统计图形要求：  1、要求每个图有标题，有x、y轴的刻度标签或刻度值，有图例   1. 要求图的美观 2. 除了上述数据，也可以自己收集数据进行分析，并进行可视化展示   二、数据简介  （1）crimeRatesByState2005.csv：这个数据集展示的为美国各州各种犯罪行为的发生率（每10万人口）。  （2）hot-dog-contest-winners.csv：这个数据集展示的为从1980年开始，每年吃热狗大赛的冠军，冠军吃掉热狗的数量，冠军的国籍，以及是否创新纪录（0表示没有破纪录，1表示创造了新纪录）。  hot-dog-places.csv：这个数据集展示的为各年份冠亚季军所吃热狗的数量。  （3）unemployment-rate-1948-2010.csv：这个数据集展示的是美国1948年到2010年各月份的失业率。  （4）us-postage.csv：这个数据集展示的是从1991年-2010年美国邮费的变化。  （5）world-population.csv：这个数据集展示的是从1960年-2009年的世界人口数量。  （6）subscribers.csv：这个数据集展示的为某微信公众号从2010年1月份的订阅、访问、点击数量  （7）疫情数据.csv：这个数据集展示的为2020-3-18号不同省市疫情确认、疑似、死亡和治愈人数的数据。 |
| 三、实验目的  熟练运用pyecharts的绘图类包Bar、Map、EffectScatter、Overlap、Geo、GeoLines等进行图形绘制 |
| 四、实验代码（代码用小五字号）  1.import pandas as pd  from pyecharts.charts import Line  from pyecharts import options as opts  data = pd.read\_csv("D:/data/crimeRatesByState2005.csv")  sorted\_data = data.sort\_values(by='murder', ascending=False)  top\_15\_states = sorted\_data.head(15)  states = top\_15\_states['state'].tolist()  crime\_rates = top\_15\_states[['murder', 'forcible\_rape', 'robbery', 'aggravated\_assault', 'burglary', 'larceny\_theft', 'motor\_vehicle\_theft']].values.tolist()  line = Line(init\_opts=opts.InitOpts(width='1000px', height='500px'))  color\_list = ['blue', 'green', 'purple', 'orange', 'gray', 'pink', 'rad',]  crime\_list = ['murder','forcible\_rape','robbery','aggravated\_assault','burglary','larceny\_theft','motor\_vehicle\_theft'  ]  for i, crime in enumerate(crime\_list):  line.add\_xaxis(states)  line.add\_yaxis(crime, [rate[i] for rate in crime\_rates],  label\_opts=opts.LabelOpts(color=color\_list[i]))  line.set\_global\_opts(title\_opts=opts.TitleOpts(title="State with the top 15 crime rate"),  legend\_opts=opts.LegendOpts(pos\_top='22%',orient='vertical',pos\_right='1%'),  xaxis\_opts=opts.AxisOpts(name="state",axislabel\_opts=opts.LabelOpts(rotate=-90)),  yaxis\_opts=opts.AxisOpts(name="Crime rate per 100,000 population"))  line.render("crime.html")  2.  import pandas as pd  from pyecharts.charts import Scatter  from pyecharts import options as opts  data = pd.read\_csv("D:/data/crimeRatesByState2005.csv")  scatter = Scatter()  scatter.add\_xaxis(data['population'])  scatter.add\_yaxis('Crime Data', data['murder'])  scatter.set\_series\_opts(title\_opts=opts.TitleOpts(title="Crime Data Visualization"),  xaxis\_opts=opts.AxisOpts(name="Population"),  yaxis\_opts=opts.AxisOpts(name="Murder Occurrences"),  itemstyle\_opts=opts.ItemStyleOpts(color="#00FF00")  )  scatter.render('crime.html')  3.  import pandas as pd  from pyecharts.charts import Bar, Line, Scatter, Grid  from pyecharts import options as opts  data = pd.read\_csv("D:/data/crimeRatesByState2005.csv")  data = data.sort\_values('murder', ascending=False).head(15)  bar\_chart = (  Bar()  .add\_xaxis(data['state'].tolist())  .add\_yaxis('Murder', data['murder'].tolist())  .set\_global\_opts(title\_opts=opts.TitleOpts(title='Murder Rate by State'))  )  line\_chart = (  Line()  .add\_xaxis(data['state'].tolist())  .add\_yaxis('Forcible Rape', data['forcible\_rape'].tolist(), is\_smooth=True)  .set\_global\_opts(title\_opts=opts.TitleOpts(title='Forcible Rape Rate by State'))  )  scatter\_chart = (  Scatter()  .add\_xaxis(data['state'].tolist())  .add\_yaxis('Robbery', data['robbery'].tolist())  .set\_global\_opts(title\_opts=opts.TitleOpts(title='Robbery Rate by State'))  )  xy\_chart = (  Scatter()  .add\_xaxis(data['state'].tolist())  .add\_yaxis('Aggravated Assault', data['aggravated\_assault'].tolist())  )  grid\_chart = (  Grid(init\_opts=opts.InitOpts(width='800px', height='600px'))  .add(bar\_chart, grid\_opts=opts.GridOpts(pos\_left='5%', pos\_top='10%', width='40%', height='40%'))  .add(line\_chart, grid\_opts=opts.GridOpts(pos\_right='5%', pos\_top='10%', width='40%', height='40%'))  .add(scatter\_chart, grid\_opts=opts.GridOpts(pos\_left='5%', pos\_bottom='10%', width='40%', height='40%'))  .add(xy\_chart, grid\_opts=opts.GridOpts(pos\_right='5%', pos\_bottom='10%', width='40%', height='40%'))  )  grid\_chart.render('crime.html')  4.  import pandas as pd  from pyecharts.charts import Pie, Grid  from pyecharts import options as opts  data = pd.read\_csv("D:/data/crimeRatesByState2005.csv")  data = data.sort\_values('murder', ascending=False).iloc[:15]  pie1 = (  Pie()  .add('', list(zip(data['state'], data['murder'])))  .set\_global\_opts(title\_opts=opts.TitleOpts(title='Top 15 States with Highest Murder Rate'))  )  pie1.set\_colors(['#FF4500', '#FF6347', '#FF7F50', '#FF8C00', '#FFA500', '#FFD700', '#FFFF00', '#ADFF2F', '#7CFC00', '#00FF00', '#32CD32', '#008000', '#006400'])  pie2 = (  Pie()  .add(  '',  list(zip(data['state'], data['murder'])),  radius=['40%', '55%']  )  .set\_global\_opts(  title\_opts=opts.TitleOpts(title='Top 15 States with Highest Murder Rate'),  legend\_opts=opts.LegendOpts(orient='vertical', pos\_top='15%', pos\_left='80%')  )  .set\_series\_opts(label\_opts=opts.LabelOpts(formatter='{b}: {d}%'))  )  grid = (  Grid(init\_opts=opts.InitOpts(width='800px', height='600px'))  .add(pie1, grid\_opts=opts.GridOpts(pos\_right='55%'))  .add(pie2, grid\_opts=opts.GridOpts(pos\_left='55%'))  )  grid.render('crime.html')  5.import pandas as pd  from pyecharts.charts import Geo  from pyecharts import options as opts  data = pd.read\_csv('D:/data/疫情数据.csv')  grouped\_data = data.groupby('province').sum().reset\_index()  province\_list = grouped\_data['province'].tolist()  confirm\_list = grouped\_data['confirm'].tolist()  geo = (  Geo()  .add\_schema(maptype="china")  .add(  series\_name="确诊人数",  data\_pair=[list(z) for z in zip(province\_list, confirm\_list)],  label\_opts=opts.LabelOpts(is\_show=False),  )  .set\_global\_opts(  title\_opts=opts.TitleOpts(title="COVID-19 Epidemic Map"),  visualmap\_opts=opts.VisualMapOpts(max\_=max(confirm\_list)),  )  )  geo.render('map.html')  6.  import pandas as pd  from pyecharts import options as opts  from pyecharts.charts import Geo  from pyecharts.globals import ChartType, SymbolType  df = pd.read\_table('./data/sentence.txt', sep=',')  lines = []  for i in range(7):  lines.append([(df.iloc[i, 0], df.iloc[i, 1]), (df.iloc[i, 2], df.iloc[i, 3])])  geo = (  Geo()  .add\_schema(maptype="china")  .add(  "航线",  lines,  type\_=ChartType.LINES,  effect\_opts=opts.EffectOpts(  symbol=SymbolType.ARROW, symbol\_size=6, color="yellow"  ),  linestyle\_opts=opts.LineStyleOpts(curve=0.2),  )  .set\_series\_opts(label\_opts=opts.LabelOpts(is\_show=False))  .set\_global\_opts(title\_opts=opts.TitleOpts(title="航线图"))  )  geo.render("line.html")  7.  from pyecharts.charts import Funnel  from pyecharts import options as opts  import pandas as pd  df = pd.read\_csv('D:/data/unemployment-rate-1948-2010.csv')  y = df.Value.tolist()  x = ['M01', 'M02', 'M03', 'M04', 'M05', 'M06', 'M07', 'M08', 'M09', 'M10', 'M11', 'M12']  y4 = y[36:48]  funnel = (  Funnel()  .add("1950 Unemployment Rate", [list(z) for z in zip(x, y4)])  .set\_global\_opts(title\_opts=opts.TitleOpts(title="Funnel Chart"))  )  funnel.render("unemployment.html") |
| 1. 实验结果   1.    2.    3.    4.    5.    6.    7. |