本次实验选择编程类工具Pyecharts。

在Pycharm中新建python项目，并引入pyecharts，openpyxl，datetime等必要包。

创建main.py，写入以下内容。

import openpyxl

from pyecharts.charts import HeatMap, Parallel

import pyecharts.globals as globals

from pyecharts import options as opts

from pyecharts.charts import Pie

globals.\_WarningControl.ShowWarning = False

def getClassifyData():

filename = "classifyday1.xlsx"

ws = openpyxl.load\_workbook(filename)['classifyday1']

classifyData = []

for i in range(2, ws.max\_row + 1):

row = str(i)

classifyData.append([ws['A' + row].value, ws['B' + row].value])

return classifyData

def getRoomData():

filename = "time\_allocate\_day1.xlsx"

ws = openpyxl.load\_workbook(filename)['time\_allocate\_day1']

roomData, roomName = [], []

for room in range(24):

roomName.append(

opts.ParallelAxisOpts(dim=room, name=ws[chr(room + 65) + '1'].value, min\_='dataMin', max\_='dataMax'))

for i in range(2, ws.max\_row + 1):

row, person = str(i), []

for room in range(24):

person.append(ws[chr(room + 65) + row].value)

roomData.append(person)

return roomData, roomName

def drawRoomDataParallel():

roomData, roomName = getRoomData()

c = (

Parallel(init\_opts=opts.InitOpts(width="2000px", height="1000px")).add\_schema(

roomName).add(series\_name="人员分类图",

data=roomData,

linestyle\_opts=opts.LineStyleOpts(color={'type': 'linear', 'x': 0,

'y': 0,

'x2': 0,

'y2': 1,

'colorStops': [{

'offset': 0, 'color': 'red'

}, {

'offset': 1,

'color': 'blue'

}],

'global': False

}, opacity=0.5)

).set\_global\_opts(title\_opts=opts.TitleOpts(title="数据可视化实验五"))

)

c.render("不同人员平行坐标系图.html")

def genHeatBaseData():

x, y = [], []

for i in range(100):

fmt = str("{:0>2d}".format(i))

x.append(fmt)

y.append('1' + fmt)

return x, y

def classifyHeatMap(xAxisData: list, yAxisData: list, data: list):

HeatMap(init\_opts=opts.InitOpts(width="2000px", height="2000px")).add\_xaxis(xaxis\_data=xAxisData).add\_yaxis(

series\_name="classifyHeatMap",

yaxis\_data=yAxisData,

value=data,

label\_opts=opts.LabelOpts(

is\_show=False, color="#fff", position="bottom", horizontal\_align="50%"

),

).set\_series\_opts().set\_global\_opts(

legend\_opts=opts.LegendOpts(is\_show=True),

xaxis\_opts=opts.AxisOpts(

type\_="category",

splitarea\_opts=opts.SplitAreaOpts(

is\_show=True, areastyle\_opts=opts.AreaStyleOpts(opacity=1)

),

axislabel\_opts=opts.LabelOpts(

interval=0

)

),

yaxis\_opts=opts.AxisOpts(

type\_="category",

splitarea\_opts=opts.SplitAreaOpts(

is\_show=True, areastyle\_opts=opts.AreaStyleOpts(opacity=1)

),

),

visualmap\_opts=opts.VisualMapOpts(

orient="horizontal", pos\_left="center", is\_piecewise=True,

pieces=[

{"value": 2, "label": 'waiter'},

{"value": 4, "label": 'vip'},

{"value": 6, "label": 'participant'},

{"value": 8, "label": 'meeting'},

{"value": 10, "label": 'reporter'},

]

),

).render("classifyHeatMap.html")

def serializeClassifyData(data: list):

job = {'waiter': 2, 'vip': 4, 'participant': 6, 'meeting': 8, 'reporter': 10}

sData = {}

for person in data:

if not sData.\_\_contains\_\_(str(person[0])[0:3]):

sData[str(person[0])[0:3]] = {}

sData[str(person[0])[0:3]][str(person[0])[3:5]] = job[person[1]]

return sData

def genClassifyHeatMapData(data: dict):

rdata = [] # data中的每个列表内容依次是横坐标 纵坐标 值

for k, v in data.items():

for sk, sv in v.items():

rdata.append([int(sk), int(k) - 100, sv])

return rdata

def drawClassifyHeatMap():

classifyData = getClassifyData()

serializedClassifyData = serializeClassifyData(classifyData)

xAxisValue, yAxisValue = genHeatBaseData()

classifyHeatMapData = genClassifyHeatMapData(serializedClassifyData)

classifyHeatMap(xAxisValue, yAxisValue, classifyHeatMapData)

def getRoomData():

classifyData = getClassifyData()

serializedClassifyData = serializeClassifyData(classifyData)

filename = "time\_allocate\_day1.xlsx"

ws = openpyxl.load\_workbook(filename)['time\_allocate\_day1']

roomData = [[0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0]]

for i in range(2, ws.max\_row + 1):

row = str(i)

for room in range(6):

roomData[((serializedClassifyData[str(ws['A' + row].value)[0:3]][

str(ws['A' + row].value)[3:5]]) // 2) - 1][room] += ws[chr(103 + room) + row].value

# roomData=[[0, 0, 0, 0, 60831, 234033], [89477, 1235993, 30267, 0, 60746, 0], [78857, 0, 30304, 0, 3632303, 0],

# [633270, 0, 410705, 0, 0, 0], [3233, 1550, 2401, 119402, 0, 0]]

return roomData

def getRoomData2(roomData: list):

roomData2 = [[], [], [], [], [], []]

for i in roomData:

k = 0

for v in i:

roomData2[k].append(v)

k += 1

# roomData2=[[0, 89477, 78857, 633270, 3233], [0, 1235993, 0, 0, 1550], [0, 30267, 30304, 410705, 2401],[0, 0, 0,

# 0, 119402], [60831, 60746, 3632303, 0, 0], [234033, 0, 0, 0, 0]]

return roomData2

def drawRoomData2Pie():

roomData2 = getRoomData2(getRoomData())

jobs = ['waiter', 'vip', 'participant', 'meeting', 'reporter']

newData = [[], [], [], [], [], []]

k = 0

for room in roomData2:

i = 0

for job in room:

newData[k].append([jobs[i], job])

i += 1

k += 1

k = 0

pieCenter = [["20%", "30%"], ["55%", "30%"], ["85%", "30%"], ["20%", "70%"], ["55%", "70%"], ["85%", "70%"]]

pie = Pie()

for data in newData:

pie.add("Room" + str(k + 1), data, center=pieCenter[k], radius=[40, 70])

k += 1

pie.set\_global\_opts(

title\_opts=opts.TitleOpts(title="Room1-6各人员访问量"),

legend\_opts=opts.LegendOpts(is\_show=False),

).set\_series\_opts(label\_opts=opts.LabelOpts(formatter="{b}: {c}", is\_show=True))

pie.render("roomTime.html")

if \_\_name\_\_ == '\_\_main\_\_':

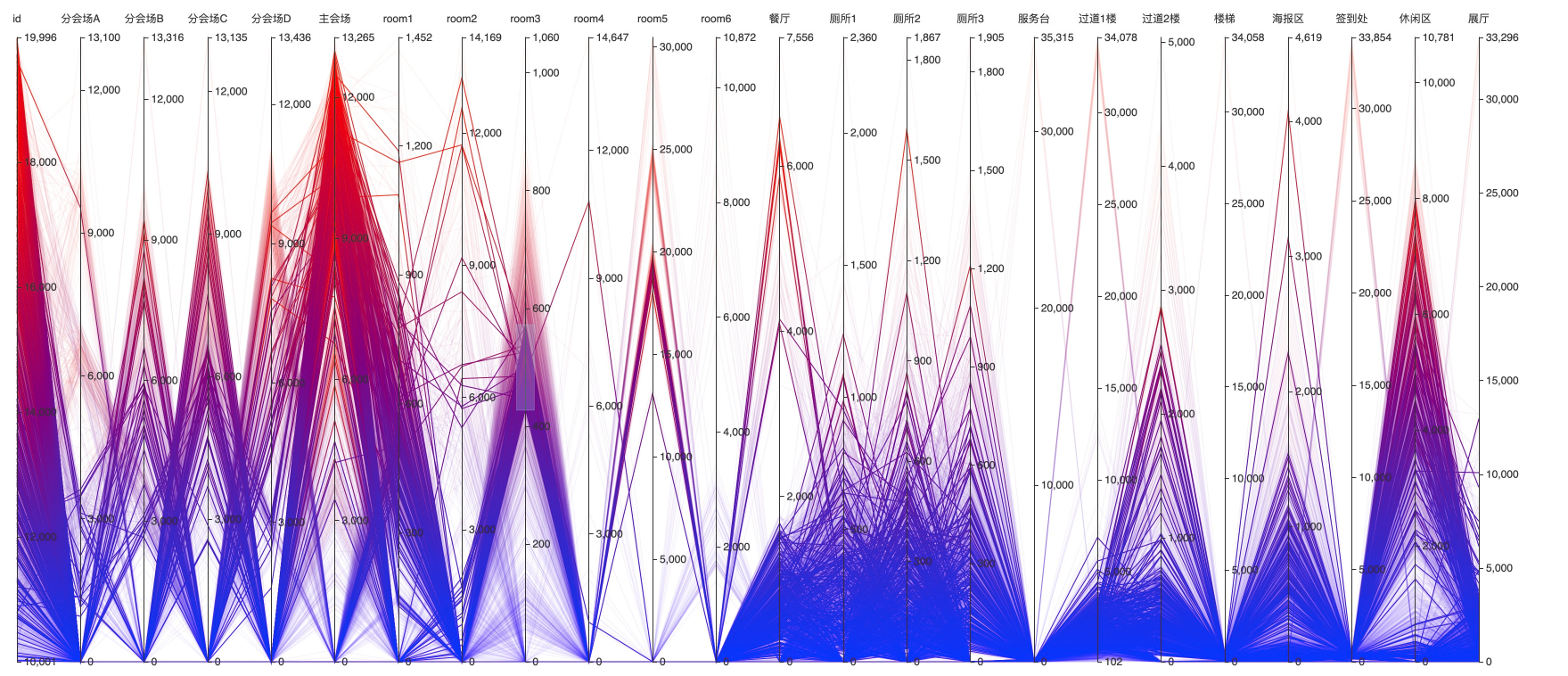
# drawClassifyHeatMap() # 热力图是在已知人员分类的情况下画出来的，但是第一问的前提是不知道人员分类

# 用平行坐标系画图

drawRoomDataParallel()

# 1:休息区 2:嘉宾休息区 3:休息区 4:记者区 5:黑客竞赛现场 6：工作人员休息区

drawRoomData2Pie()

所以结论为：

Room1:休息区

Room2:嘉宾休息区

Room3:休息区

Room4:记者区

Room5:黑客竞赛现场

Room6：工作人员休息区