**个人作品报告**

|  |  |
| --- | --- |
| **课程名称：** | 数据可视化 |
| **项目名称：** | 携程旅行酒店数据可视化 |
| **学 号：** | 2021220440 |
| **姓 名：** | 阳健 |
| **所属学院：** | 理工农学院 |
| **专业年级：** | 21级软件工程四班 |

目 录

[**一、项目需求分析和设计** 2](#_Toc164879523)

[1.1项目背景 2](#_Toc164879524)

[1.2数据来源 2](#_Toc164879525)

[1.3可视化目标 2](#_Toc164879526)

[1.4可视化工具 3](#_Toc164879527)

[**二、项目功能和实现** 3](#_Toc164879528)

[2.1数据预处理 3](#_Toc164879529)

[2.2可视化图表设计 3](#_Toc164879530)

[2.3 代码实现 3](#_Toc164879531)

[**三、使用技术** 9](#_Toc164879532)

[**四、项目发布或运行情况** 10](#_Toc164879533)

[4.1运行状态 10](#_Toc164879534)

[4.2问题和改进 10](#_Toc164879535)

[**五、项目总结** 10](#_Toc164879536)

[5.1 项目成果 10](#_Toc164879537)

[5.2 经验教训 10](#_Toc164879538)

[5.3 未来展望 10](#_Toc164879539)

# **一、项目需求分析和设计**

## 1.1项目背景

基于已完成的携程旅行酒店数据采集项目，本项目旨在对采集到的酒店数据进行可视化分析，以更加直观的方式呈现酒店数据特征，帮助用户了解酒店市场分布、价格趋势以及用户评价等信息。

## 1.2数据来源

本项目数据来源于已采集的携程旅行酒店数据，包括酒店名称、价格、评分、区域、地理位置、设施服务等信息。

## 1.3可视化目标

1. 酒店名称
2. 酒店价格
3. 酒店评分
4. 酒店区域
5. 酒店地理位置
6. 酒店设施服务
7. 酒店标签

## 1.4可视化工具

本项目采用 Python 的数据可视化库插件进行图表制作,如pyecharts等。

# **二、项目功能和实现**

## 2.1数据预处理

数据转换：将文本数据转换为数值型数据，例如将星级转换为数字。

数据聚合：根据可视化需求对数据进行聚合，例如统计每个城市的酒店数量。

数据清洗：去除重复数据和异常数据。

## 2.2可视化图表设计

酒店地理位置分布：使用地图图表展示酒店在不同城市的分布情况。

酒店价格分布：使用直方图或箱线图展示酒店价格的分布情况。

酒店星级分布：使用饼图或条形图展示不同星级酒店的数量占比。

酒店评分分布：使用直方图或分布图展示酒店评分的分布情况。

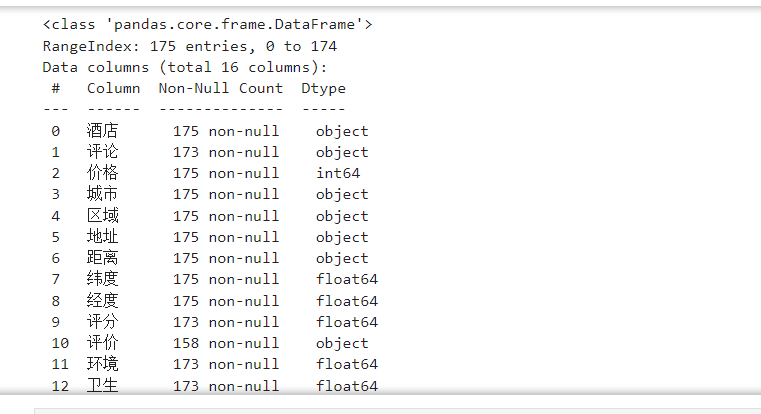
酒店设施服务分析：使用词云图展示酒店提供的设施服务。

## 2.3 代码实现

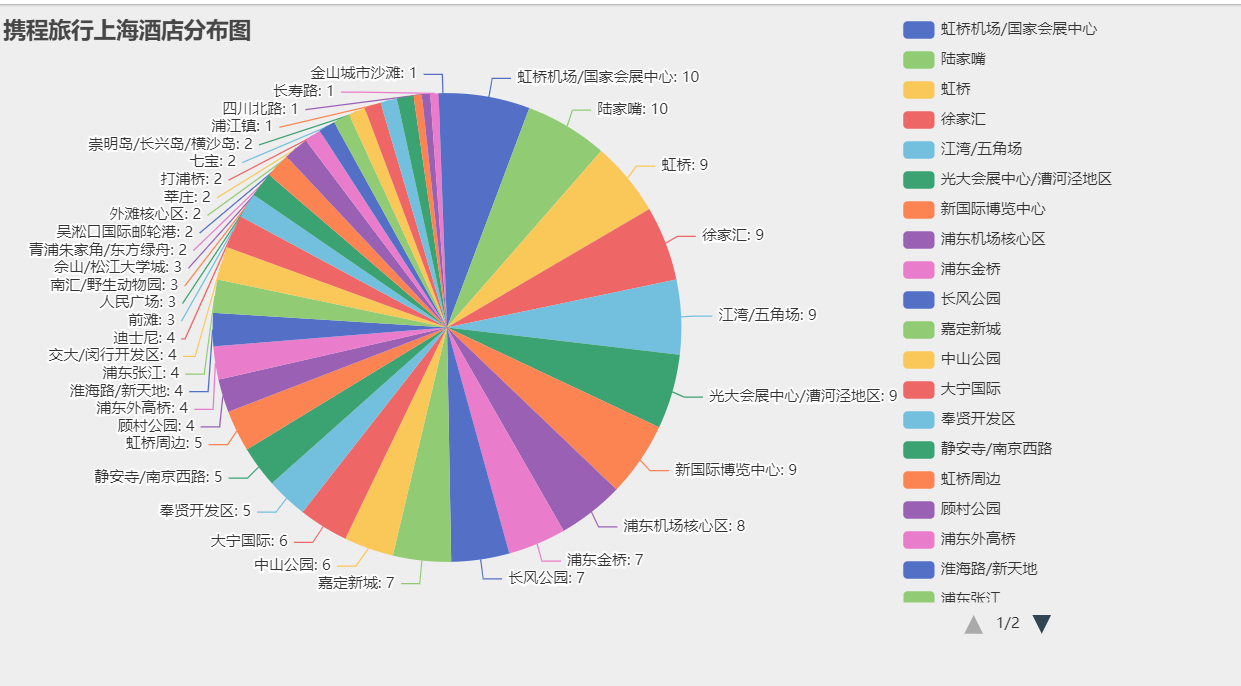
# pandas 数据处理模块  
import pandas as pd  
from pyecharts.globals import CurrentConfig, NotebookType  
CurrentConfig.NOTEBOOK\_TYPE = NotebookType.JUPYTER\_LAB  
  
df = pd.read\_csv('data.csv')  
df.head(10)



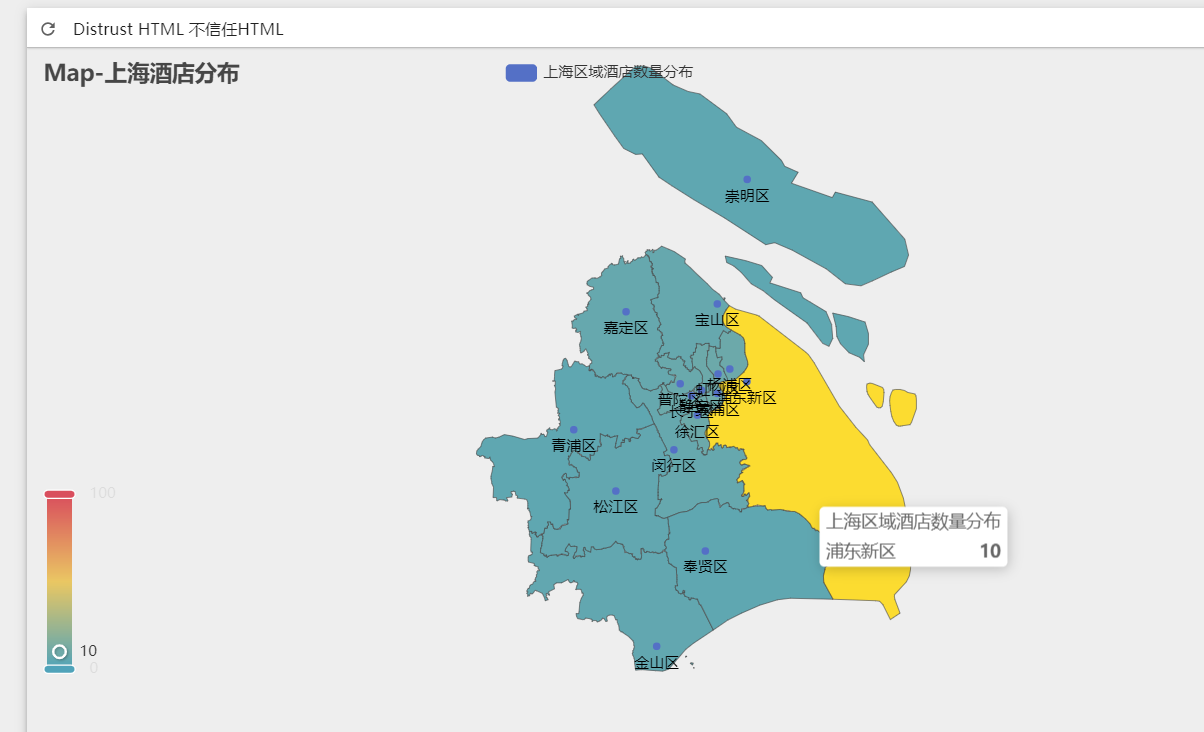
# 查看数据  
df.info()



from pyecharts import options as opts  
from pyecharts.charts import Pie  
from pyecharts.faker import Faker  
# from pyecharts.globals import CurrentConfig, NotebookType   
# CurrentConfig.NOTEBOOK\_TYPE = NotebookType.JUPYTER\_LAB   
y = df['区域'].value\_counts().to\_list()  
x = df['区域'].value\_counts().index.to\_list()  
c = (  
 Pie()  
 .add(  
 "",  
 [  
 list(z)  
 for z in zip(  
 x,  
 y,  
 )  
 ],  
 center=["40%", "50%"],  
 )  
 .set\_global\_opts(  
 title\_opts=opts.TitleOpts(title="携程旅行上海酒店分布图"),  
 legend\_opts=opts.LegendOpts(type\_="scroll", pos\_left="80%", orient="vertical"),  
 )  
 .set\_series\_opts(label\_opts=opts.LabelOpts(formatter="{b}: {c}"))  
 # .render("pie\_scroll\_legend.html")  
)  
c.render\_notebook()



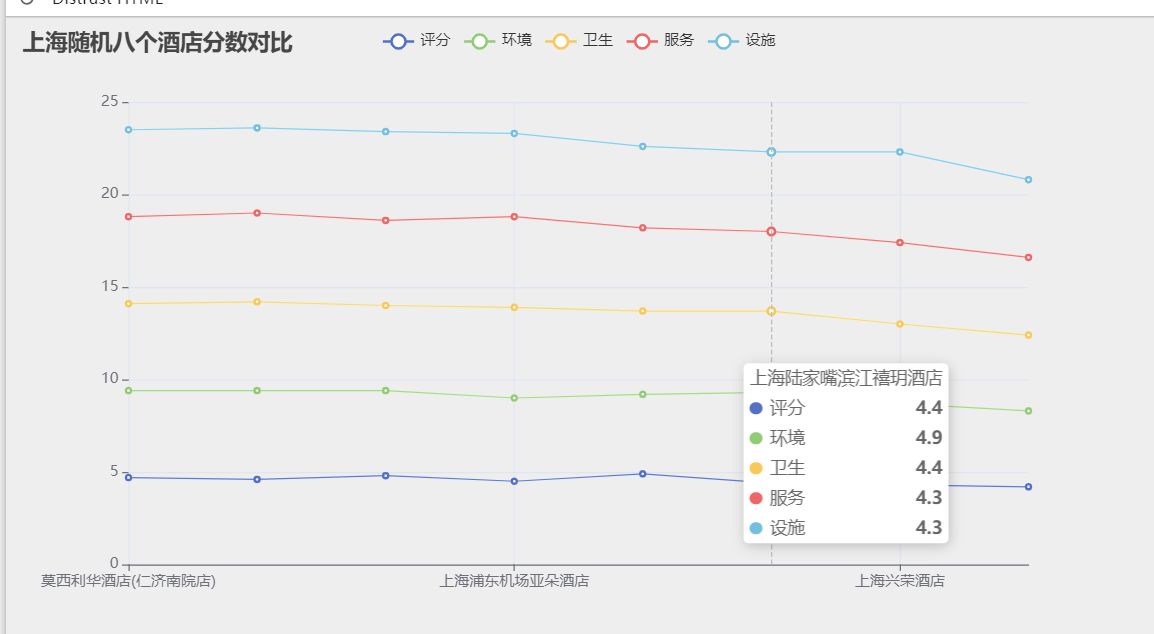
rom pyecharts import options as opts  
from pyecharts.charts import Map  
from pyecharts.faker import Faker  
from pyecharts.globals import CurrentConfig, NotebookType  
CurrentConfig.NOTEBOOK\_TYPE = NotebookType.JUPYTER\_LAB  
  
y = df['区域'].value\_counts().to\_list()  
c = (  
 Map()  
 .add("上海区域酒店数量分布", [list(z) for z in zip(['浦东新区','黄浦区','静安区','徐汇区','长宁区','虹口区','杨浦区','普陀区','闵行区','宝山区','嘉定区','金山区','松江区','青浦区','奉贤区','崇明区'], y)], "上海")  
 .set\_global\_opts(  
 title\_opts=opts.TitleOpts(title="Map-上海酒店分布"), visualmap\_opts=opts.VisualMapOpts()  
 )  
)  
c.render\_notebook()



import pyecharts.options as opts  
from pyecharts.charts import Bar, Line  
  
x\_data = df['酒店'].value\_counts().index.to\_list()[:9]  
def process\_price(price):  
 return price / 100  
bar = (  
 Bar()  
 .add\_xaxis(xaxis\_data=x\_data)  
 .add\_yaxis(  
 series\_name="每晚价格",  
 y\_axis=df['价格'].apply(process\_price).value\_counts().index.to\_list(),  
 label\_opts=opts.LabelOpts(is\_show=False),  
 )  
 .add\_yaxis(  
 series\_name="评分",  
 y\_axis=df['评分'].value\_counts().index.to\_list(),  
 label\_opts=opts.LabelOpts(is\_show=False),  
 )  
 .extend\_axis(  
 yaxis=opts.AxisOpts(  
 name="评分",  
 type\_="value",  
 min\_=0,  
 max\_=5,  
 interval=0.2,  
 axislabel\_opts=opts.LabelOpts(formatter="{value}"),  
 )  
 )  
 .set\_global\_opts(  
 tooltip\_opts=opts.TooltipOpts(  
 is\_show=True, trigger="axis", axis\_pointer\_type="cross"  
 ),  
 xaxis\_opts=opts.AxisOpts(  
 type\_="category",  
 axispointer\_opts=opts.AxisPointerOpts(is\_show=True, type\_="shadow"),  
 ),  
 yaxis\_opts=opts.AxisOpts(  
 name="价格",  
 type\_="value",  
 min\_=0,  
 max\_=8,  
 interval=0.5,  
 axislabel\_opts=opts.LabelOpts(formatter="{value}\*100"),  
 axistick\_opts=opts.AxisTickOpts(is\_show=True),  
 splitline\_opts=opts.SplitLineOpts(is\_show=True),  
 ),  
 )  
)  
  
line = (  
 Line()  
 .add\_xaxis(xaxis\_data=x\_data)  
 .add\_yaxis(  
 series\_name="平均评分",  
 yaxis\_index=1,  
 y\_axis=df['评分'].value\_counts().index.to\_list(),   
 label\_opts=opts.LabelOpts(is\_show=False),  
 )  
)  
  
bar.overlap(line).render\_notebook()



import pyecharts.options as opts  
from pyecharts.charts import Line  
  
x\_data = df['酒店'].value\_counts().index.to\_list()[:8]  
  
c = (  
 Line()  
 .add\_xaxis(xaxis\_data=x\_data)  
 .add\_yaxis(  
 series\_name="评分",  
 stack="总量",  
 y\_axis=df['评分'].value\_counts().index.to\_list(),  
 label\_opts=opts.LabelOpts(is\_show=False),  
 )  
 .add\_yaxis(  
 series\_name="环境",  
 stack="总量",  
 y\_axis=df['环境'].value\_counts().index.to\_list(),  
 label\_opts=opts.LabelOpts(is\_show=False),  
 )  
 .add\_yaxis(  
 series\_name="卫生",  
 stack="总量",  
 y\_axis=df['卫生'].value\_counts().index.to\_list(),  
 label\_opts=opts.LabelOpts(is\_show=False),  
 )  
 .add\_yaxis(  
 series\_name="服务",  
 stack="总量",  
 y\_axis=df['服务'].value\_counts().index.to\_list(),   
 label\_opts=opts.LabelOpts(is\_show=False),  
 )  
 .add\_yaxis(  
 series\_name="设施",  
 stack="总量",  
 y\_axis=df['设施'].value\_counts().index.to\_list(),  
 label\_opts=opts.LabelOpts(is\_show=False),  
 )  
 .set\_global\_opts(  
 title\_opts=opts.TitleOpts(title="上海随机八个酒店分数对比"),  
 tooltip\_opts=opts.TooltipOpts(trigger="axis"),  
 yaxis\_opts=opts.AxisOpts(  
 type\_="value",  
 axistick\_opts=opts.AxisTickOpts(is\_show=True),  
 splitline\_opts=opts.SplitLineOpts(is\_show=True),  
 ),  
 xaxis\_opts=opts.AxisOpts(type\_="category", boundary\_gap=False),  
 )  
)  
c.render\_notebook()



# **三、使用技术**

1. Jupyter：提供了一种环境来编写和测试代码，同时记录和分享数据分析过程。
2. PyrCharts：将数据转换为易于理解的图表。
3. Pandas：使用Pandas库进行数据的读取、处理和分析。

# **四、项目发布或运行情况**

## 4.1运行状态

项目运行稳定，可视化图表清晰美观，交互功能流畅。

代码已经上传至GitHub：https://github.com/oyj135/Python

## 4.2问题和改进

可视化图表类型需要进一步丰富，满足不同用户的分析需求。

# **五、项目总结**

## 5.1 项目成果

通过本项目的实践，我掌握了数据可视化的基本流程和技术，包括数据获取、清洗和处理、可视化设计和交互功能的实现。在项目中，我遇到了一些挑战，如数据清洗和处理的复杂性、可视化设计的多样性等，但通过不断学习和实践，我成功解决了这些问题，并取得了良好的效果。通过本项目，我对数据可视化的重要性和应用有了更深入的理解，为今后的数据分析和可视化工作打下了坚实的基础。

## 5.2 经验教训

数据可视化需要结合数据特征和用户需求选择合适的图表类型。

## 5.3 未来展望

扩展数据可视化功能，支持更多类型的图表和分析方法。

开发数据分析模型，提供更深入的数据洞察。

集成机器学习算法，实现个性化推荐和预测功能。