



北京理工大学珠海学院

BEIJING INSTITUTE OF TECHNOLOGY, ZHUHAI

# 关于医疗数据的数据可视化

学    院： 数理与土木工程学院

课    程： 数据可视化

班    级： 数据科学与大数据技术 1 班

姓    名： 覃诗杰 学    号： 201205102261

中国·珠海

2022 年 12 月 18 日

# 医疗数据可视化

## 摘 要

此次数据集是某地区的医疗数据，目的是为了分析医疗费用的花费情况，在不同的医院花费的金额。分析当代人民生病的主要病理，为此地的医疗建设和群众的就医情况做一个数据分析。为此做关于医疗费用的数据可视化，增加可读性。

此次主要使用的是 python 软件，版本为 python3.9.11 和 jupyter，还有数据可视化工具 Tableau Public 2022.2。来进行数据分析。

数据可视化的意义是帮助人更好的分析数据，信息的质量很大程度上依赖于其表达方式。总结如下，在花费的费用方面，无论是一级医院还是社区医院，所花费的金额都是一样的。主动脉弓狭窄是最严重的问题，也是花费金额最多的疾病。社区医院就医占多数，其他的也都差不多。主要的医护人员处于中年。发病率最高的是糖尿病。

**关键词：** python   pyecharts   医疗   Tableau

目 录

<b>1</b>	<b>背景</b>	<b>1</b>
1.1	数据介绍 . . . . .	1
<b>2</b>	<b>数据处理</b>	<b>1</b>
2.1	数据清洗 . . . . .	1
<b>3</b>	<b>数据可视化</b>	<b>2</b>
3.1	条形图 . . . . .	2
3.2	相关系数热力图 . . . . .	3
3.3	看病平均花费 top10 . . . . .	4
3.4	双柱状图 . . . . .	5
3.5	饼图 . . . . .	5
3.6	玫瑰图 . . . . .	6
3.7	圆环图 . . . . .	7
3.8	象型柱图 . . . . .	7
3.9	环形饼图 . . . . .	8
3.10	以特征画图 . . . . .	8
<b>4</b>	<b>数据可视化大屏</b>	<b>9</b>
<b>5</b>	<b>总结与评价</b>	<b>10</b>
5.1	总结 . . . . .	10
5.2	评价 . . . . .	10
	<b>参考文献</b>	<b>11</b>
	<b>附录</b>	<b>12</b>

# 1 背景

对于某个具体的个体来说，医疗是指个体为了挽救生命、延长寿命、提高生存质量从而使个人效用最大化所最需要利用的、最优先利用的医疗服务或医疗措施；对于某个社会、某个群体（比如某个国家的公民）来说，医疗是指对改善全体社会公民健康、提高国民素质、推动社会发展贡献最大，最应该为全体公民所享受的医疗服务或医疗措施。所有我们对医疗进行探究

## 1.1 数据介绍

文件中都是以简称形式存在，介绍所有的信息，如下表：

ID	NL	XB	RYLB	YLLB	RYQH	JZQH
序号	年龄	性别	人员	医疗类型	人员区	就诊区
JGDJ	RYRQ	CYRQ	ZDMC	ZFY	TCFY	JSRQ
医疗机构等级	入院日期	出院日期	诊断名称	支付费用	统筹费用	结算日期

# 2 数据处理

## 2.1 数据清洗

先查询我们需要的数据的缺失值，异常值，做出缺失值可视化图：

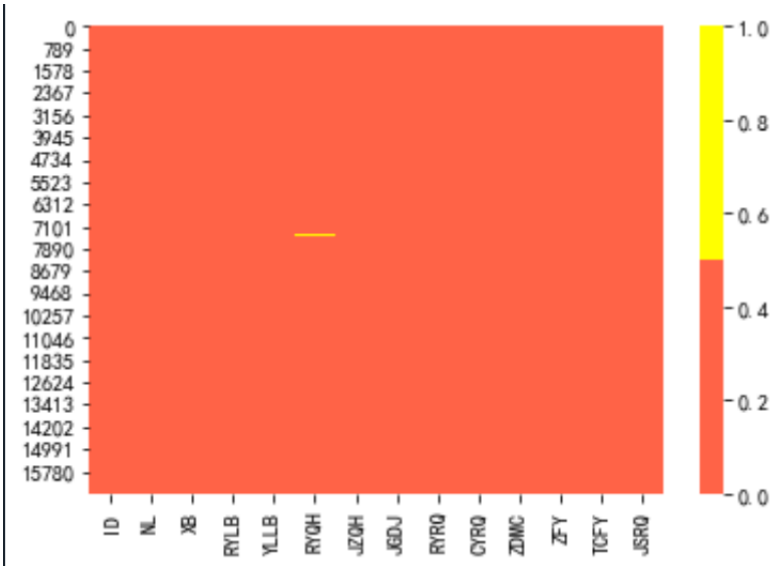


图 1: 数据可视化

用热力图展现一下相关数据结构：  
也可以从数据热力图看出缺失了数据

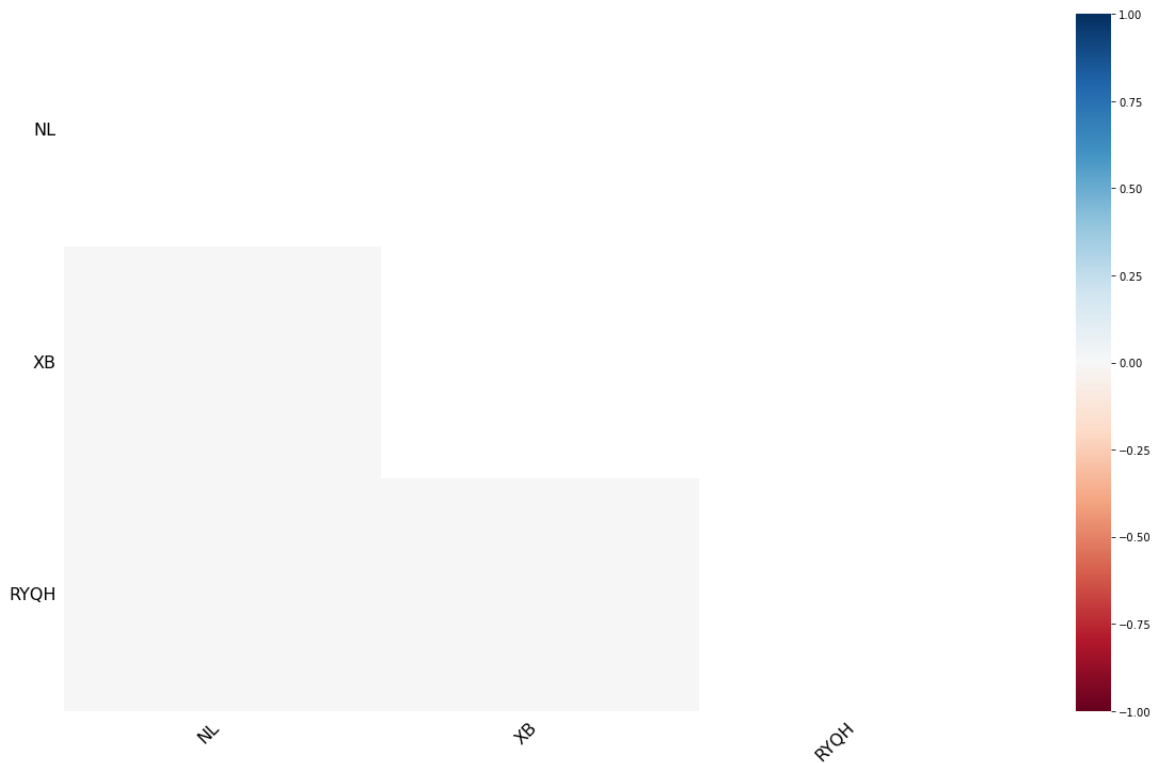


图 2: 数据可视化

然后做数据处理，因为有年龄的缺失，所有用中位数填充，然后还一缺失的是性别，这就用众数填补：

```
1 data = data.dropna(axis=0)
2 data.isna().sum()    # 统计缺失值
3 data.RYQH[data.RYQH.isnull()] = data.JZQH[data.RYQH.isnull()]
```

## 3 数据可视化

### 3.1 条形图

想用条形图方式，画出以地域的患病人数和以人们患病的地域。  
以人员患病所在区域，画病例人数统计条形图

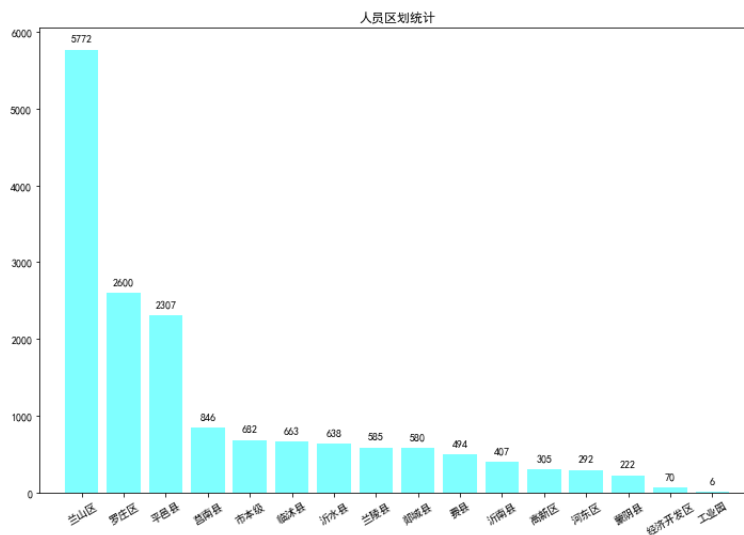


图 3: 人员患病统计分配图

以就诊区域为主，画病例人数统计图

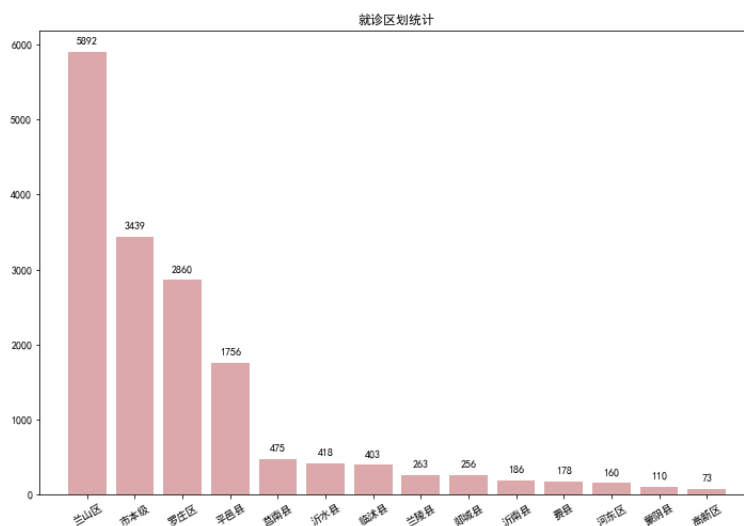


图 4: 按城市划分的收入分配图

我们首先可以得出：病患的人主要集中在兰山区和罗庄区、平邑区，然后还可以得出，大家都是就近就医，没有说单单往一级医院去进行就医。所有可以得出，政府应该要平均分配，考虑这些医院的设备等问题，定时做检查，资金投入也要合理。

### 3.2 相关系数热力图

该相关系数只能度量出变量之间的线性相关关系；也就是说，相关系数越高，则变量间的线性相关程度越高。对于相关系数小的两个变量，只能说明变量间的线性相关程度弱，但不能说明变量之间不存在其它的相关关系。通过热力图，我们可以直观地看到所给数值大小的差异状况。

如下图：

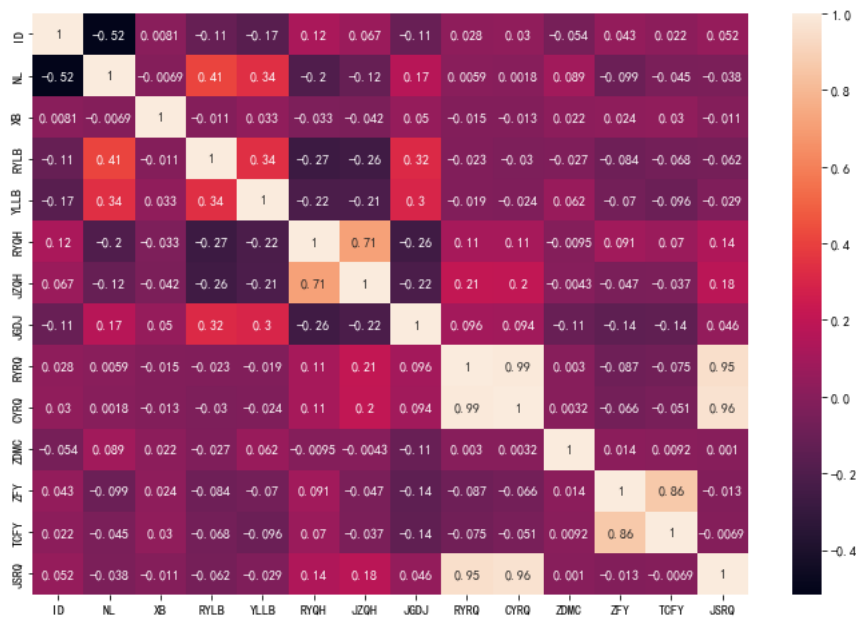


图 5: 数据相关系数可视化

热力图右侧的刻度展示了不同相关系数对应的颜色深浅。从图中可以看出，权益乘数和流动负债权益比率之间的相关性较高。

### 3.3 看病平均花费 top10

探究在所有病例中，为治疗此病例要花费的金额，可视化出 top10, 为了使政府知道那种病需要去了解用药情况，去把贵的药物价格降低，使城市居民不在害怕没钱不敢治疗。

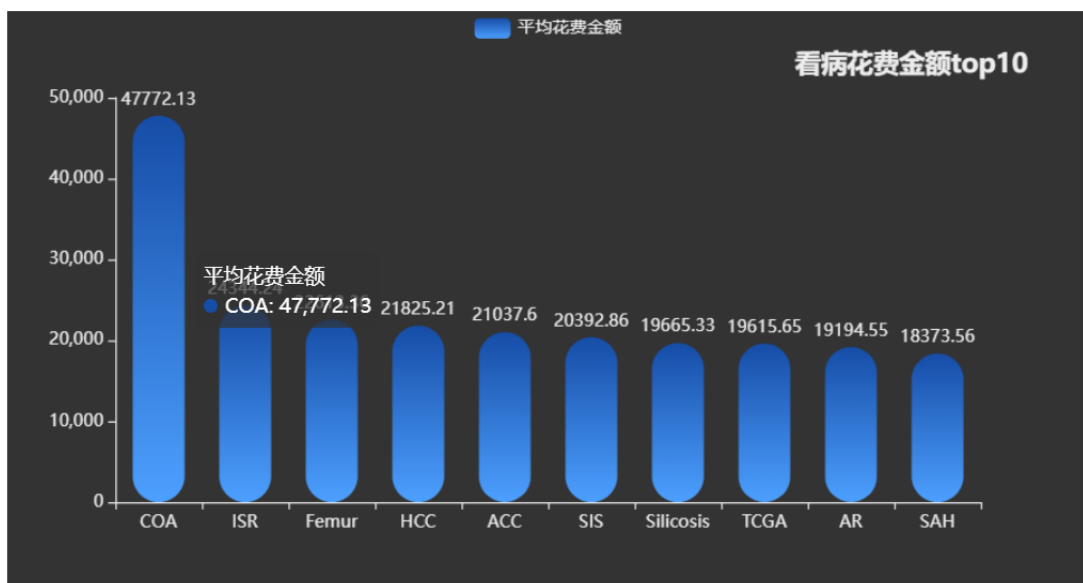


图 6: 看病平均花费 top10

可以看出 coa 也就是主动脉弓狭窄，是特别耗费钱的，比其他病都要花费多的多。

处理完数据如下表：

主动脉弓狭窄	动脉狭窄	股骨假体周围骨折	肝肿瘤	慢性丹囊炎	矽肺
47772.13	24344.24	22613.36	21825.21	21037.6	20392.86

### 3.4 双柱状图

因为社会舆论中存在不同医院收费不同，越是大医院收费越高问题，此次我们可视化一级医院和二级医院相同病例的平均收费情况，在理论情况下，有一定偏差，但是不存在很大偏差则可以认为医院没有越是大医院收费越高问题。证明所有医院的都是以正常标准收费。不乱收钱，可视化出结果，让居民方向。

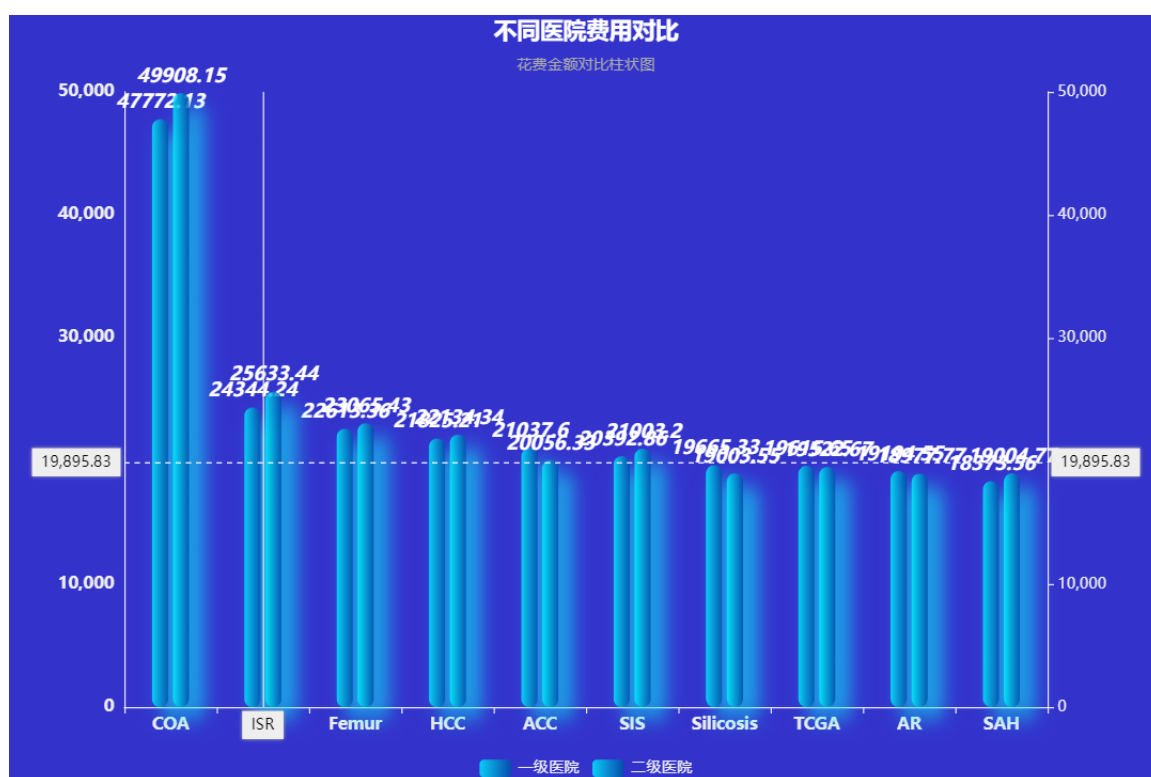


图 7: 医院费用对比可视化

可以很明显看出，其实差别都不大，如果样本量足够多的情况下，误差会很小很小，所有居民不用担心医院有乱收费情况。

### 3.5 饼图

做一次调查，以所有生病人员为主题，以去过的医院为样本，画出饼图，看看是否大家都喜欢去更好的医院治病。

可以看出，社区医院才是最多的。说明社区医院才是居民最主要的去的医院。也可以说明居民以就近原则去治疗，并没有更喜欢去大医院治疗。





### 3.7 圆环图

为了了解受病群体的工作情况，退休情况，还是离休的。画出此图表：

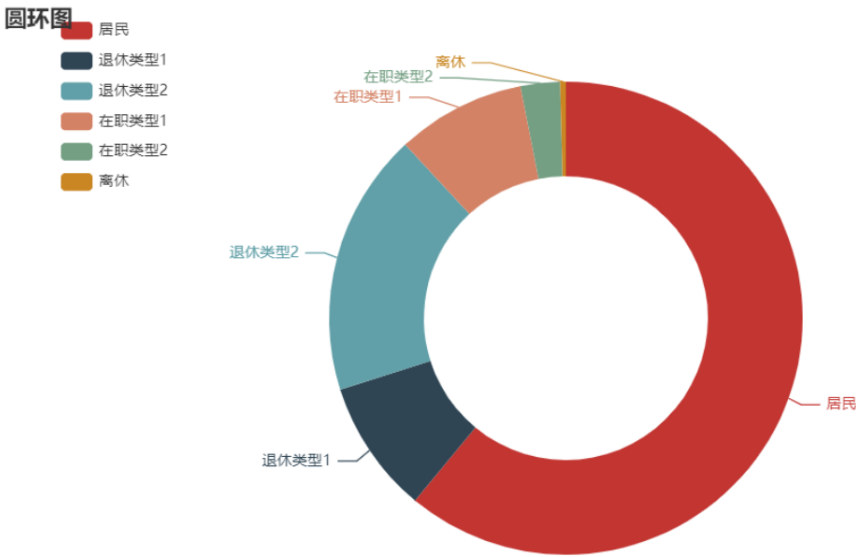


图 10: 各个医院病人的工作情况

由图可得，还是居民较多，离休较少，可能也是因为生病才离休的。

### 3.8 象型柱图

查看居民，去的医院类别：

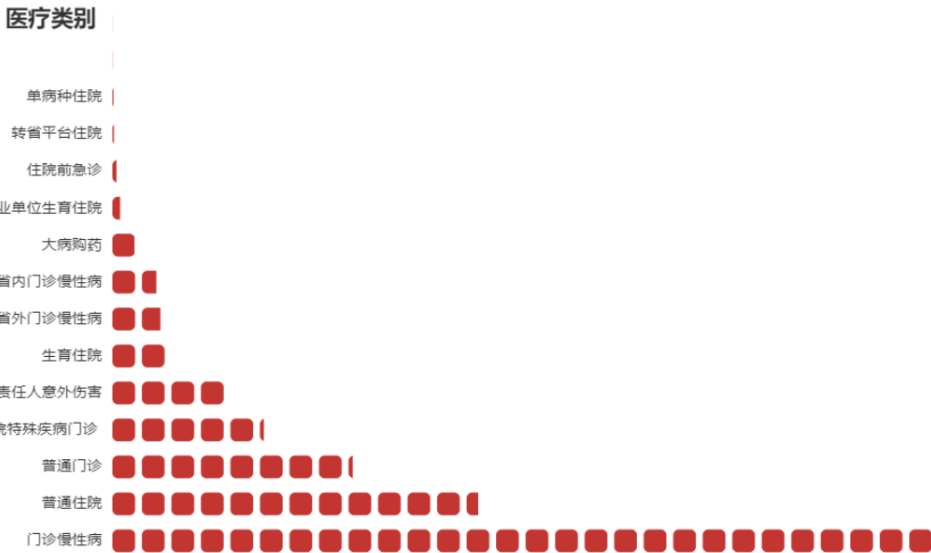


图 11: 去各个医院的占比

### 3.9 环形饼图

以外环为医疗等级，内圈为性别比例做环形图。可视化如下：

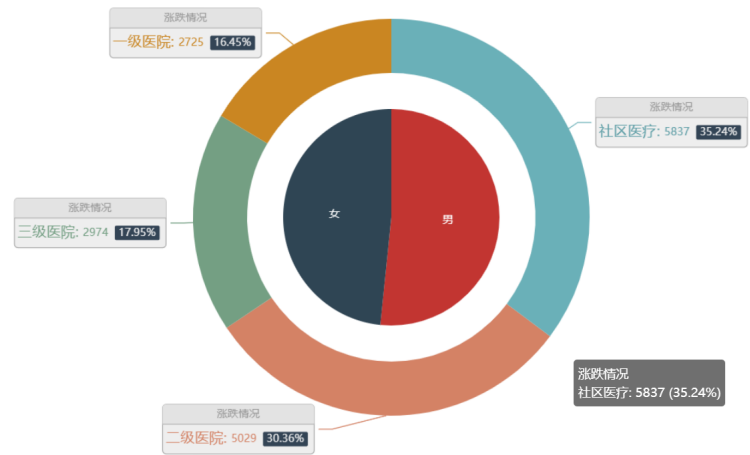


图 12: 医疗情况

可以看出男女比例都很正常。而且医院容纳人的数量也是很均衡。

### 3.10 以特征画图

在所有病例中，我们以小于七岁的年龄为特征，统计出小于七岁患病的病例，画出可视化图。

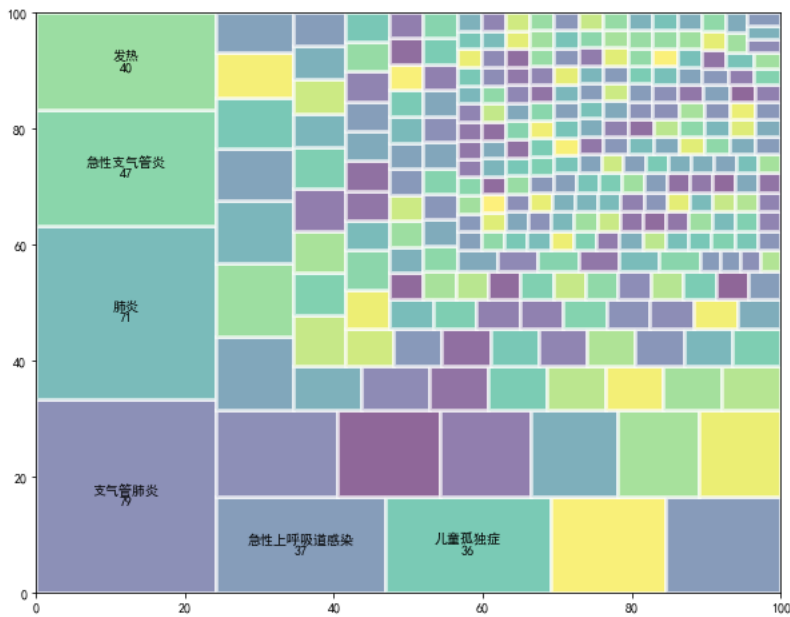


图 13: 儿童患病情况情况

由此可见：支气管肺炎和肺炎比较多，其次比较少，政府主要关注儿童的肺炎疾病。并且我们可以查看，以 70 岁为老年所患病病例：

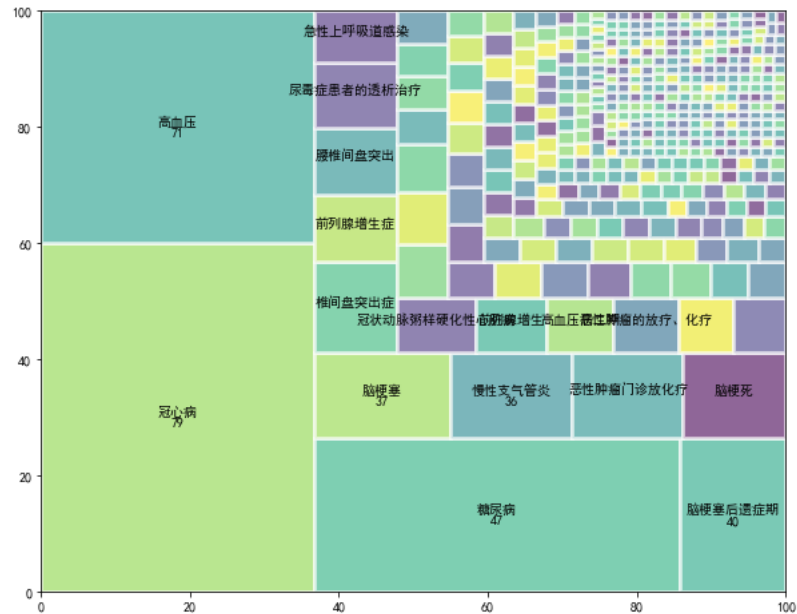


图 14: 老年人患病情况

可以看出来，老年人主要是以患冠心病，高血压，糖尿病为主。

## 4 数据可视化大屏

通过汇总，我们认为做一个数据可视化大屏，可以有效地反应所有的居民就医问题与就医意愿，如图：

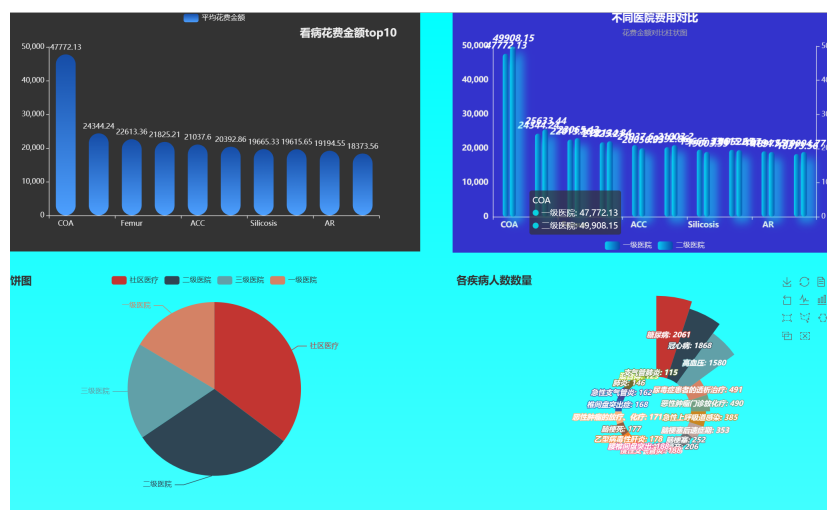


图 15: 可视化大屏

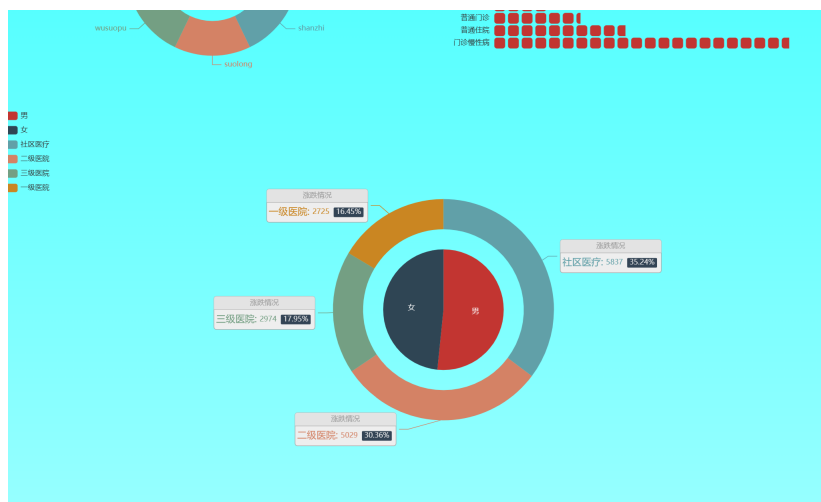


图 16: 可视化大屏

## 5 总结与评价

### 5.1 总结

医疗是为了挽救生命、延长寿命、提高生存质量从而使个人效用最大化所最需要利用的、最优先利用的医疗服务或医疗措施；对于某个社会、某个群体（比如某个国家的公民）来说，医疗是指对改善全体社会公民健康、提高国民素质、推动社会发展贡献最大，最应该为全体公民所享受的医疗服务或医疗措施。

由上述分析可得，患病者花费最多的是主动脉狭窄，患病最多的是糖尿病，大部分人都是就近就医。

### 5.2 评价

优点：对数据进行数据可视化分析，沟通效率更高，与数据直接交互，能轻松理解数据。分析结果可以为当地提供一些帮助。

缺点：数据量不够大，区域不够大，数据的更新不够快，容易影响数据可视化和预测的真实性。只能针对某区域，不能应用与其他地区。

## 参考文献

- [1] 期刊论文.《数据可视化的基本原理与方法(豆瓣)》.
- [2] 期刊论文.《重症医疗数据的可视化》.
- [3] 学习 pyecharts 网址.<https://github.com/pyecharts/pyecharts-gallery>.
- [4] 高等教育出版社. [https://blog.csdn.net/weixin\\_38100489/article/details/78175928](https://blog.csdn.net/weixin_38100489/article/details/78175928)

## 附录

### 附录 1：数据可视化

```

1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4
5 data = pd.read_csv(r'E:\2020paper\Data visualization 医疗数据\csv', encoding="gbk")
6 data.head()
7
8 #检查数据
9 type(data)
10 shape = data.shape
11 print(data.shape)
12 print('数据集的行列数分别为:', shape[0], '行', shape[1], '列\n') #行列数
13 data.isna().sum() # 统计缺失值
14
15 #可视化缺失图
16 colours = ['#FF6347', '#ffff00']
17 sns.heatmap(data.isnull(), cmap=sns.color_palette(colours))
18
19 import missingno as msno
20 # 矩阵图：矩阵的矩阵是一个数据密集的显示，它可以让您快速地可视化地挑选出模式
    的数据完成。nan
21 msno.matrix(data.sample(250)) # 表示需要显示的条数250
22 # 柱形图：通过列来表示的简单方法nan:
23 msno.bar(data.sample(1000))
24 # 热力图：缺失无相关热图测量零相关一个变量的存在或不存在对另一个变量存在的影
    响有多大：
25 msno.heatmap(data)
26 # 树状图：可以让你更全面地将变量完成情况关联起来，比相关热图中的两两对比更深
    入地揭示趋势：
27 msno.dendrogram(data)
28
29 ## 缺失率
30 data.isnull().sum(axis=0)/data.shape[0]
31 # 删除全空的行
32 data.dropna(how='all', inplace=True)
33 data = data.dropna(axis=0)
34 data.isna().sum() # 统计缺失值
35

```

```

36 #其中很多数值符合线性回归所以用前后的均值填补
37 #填补数据
38 data['NL'] = data['NL'].fillna(method='bfill')
39 data['XB'] = data['XB'].fillna(method='bfill')
40 data['RYQH'] = data['RYQH'].fillna(method='bfill')
41
42 data = data.dropna(axis=0)
43 data.isna().sum() # 统计缺失值
44
45
46 import pandas as pd
47 import numpy as np
48 import matplotlib.pyplot as plt
49
50 # 接下里做一些可视化分析
51 # 区域分布这里的划分比较乱，需要把他们统一划分到镇或者街道，这个应该是临沂市的
    区划。
52 shuju = {'费县': '费县', '兰山区': '兰山区', '双堠镇': '沂南县', '长城镇': '兰陵
    县', '流峪镇': '平邑县', '兰陵县': '兰陵县', '夏蔚镇': '沂水县', '四十里镇': '沂水
    县', '沂水县': '沂水县', '市本级': '市本级', '郑城镇': '平邑县', '山东省临沂
    市': '市本级', '新庄镇': '费县', '大兴镇': '临沭县', '依汶镇': '沂南县', '郯城县港
    上镇': '郯城县', '温水镇': '平邑县', '朱田镇': '费县', '高新区': '高新区', '上
    冶镇': '费县', '姚店子镇': '沂水县', '汤河': '河东区', '临港产业区': '莒南县', '孙
    祖镇': '沂南县', '沂南县': '沂南县', '圈里乡': '沂水县', '平邑镇': '平邑县', '郯城
    县马头镇': '郯城县', '临沭县': '临沭县', '莒南县': '莒南县', '南桥镇': '兰陵
    县', '高都街道办事处': '罗庄区', '砖埠镇': '沂南县', '青云镇': '临沭县', '河东
    区': '河东区', '湖头镇': '沂南县', '车辋镇': '兰陵县', '郯城县郯城镇': '郯城
    县', '探沂镇': '费县', '平邑县': '平邑县', '费城镇': '费县', '马庄镇': '费县', '郯城
    县胜利乡': '郯城县', '郯城县泉源乡': '郯城县', '大仲村镇': '兰陵县', '店头
    镇': '临沭县', '郯城县马头镇高册': '郯城县', '郯城县红花乡': '郯城县', '下村
    乡': '兰陵县', '郯城县': '郯城县', '郯城县庙山镇': '郯城县', '郯城县李庄镇': '郯
    城县', '郯城县杨集镇': '郯城县', '郯城县归昌乡': '郯城县', '铜石镇': '平邑
    县', '马站镇': '沂水县', '杨庄镇': '沂水县', '白彦镇': '平邑县', '仲村镇': '平邑
    县', '院东头镇': '沂水县', '蒙山旅游区': '蒙阴县', '大庄镇': '沂南县', '玉山
    镇': '临沭县', '许家湖镇': '沂水县', '罗庄区': '罗庄区', '诸葛镇': '沂水县', '经济
    开发区': '经济开发区', '郯城县重坊镇': '郯城县', '泉庄镇': '沂水县', '褚墩
    镇': '罗庄区', '卞庄街道': '兰陵县', '罗庄街道办事处': '罗庄区', '马厂湖镇': '兰
    山区', '临沭街道办': '临沭县', '蒲汪镇': '沂南县', '马牧池乡': '沂南县', '地方
    镇': '平邑县', '保太镇': '平邑县', '富官庄镇': '沂水县', '沂城街道办事处': '沂水
    县', '高桥镇': '沂水县', '沂堂镇': '罗庄区', '临洄镇': '平邑县', '张庄镇': '沂南
    县', '卞桥镇': '平邑县', '薛庄镇': '费县', '曹庄镇': '临沭县', '付庄街道办
    处': '罗庄区', '苏村镇': '沂南县', '梁邱镇': '费县', '磨山镇': '兰陵县', '石门
    镇': '临沭县', '兰陵镇': '兰陵县', '黄山镇': '罗庄区', '沙沟镇': '沂水县', '丰阳
    镇': '平邑县', '郑山街道办': '临沭县', '武台镇': '平邑县', '凤凰岭': '河东区', '龙
    家圈镇': '沂水县', '尚岩镇': '兰陵县', '盛庄街道办事处': '罗庄区', '新兴镇': '兰
    陵县', '庄坞镇': '兰陵县', '郯城县新村乡': '郯城县', '向城镇': '兰陵县', '岸堤
    镇': '沂南县', '南张庄乡': '费县', '胡阳镇': '费县', '铜井镇': '沂南县', '郯城县高
    峰头镇': '郯城县', '大田庄乡': '费县', '芦柞镇': '兰陵县', '鲁城镇': '兰陵县', '石
    井镇': '费县', '汤头': '河东区', '八湖': '河东区', '矿坑镇': '兰陵县', '神山镇': '兰
    陵县', '联城': '蒙阴县', '工业园': '工业园', '辛集镇': '沂南县', '相公': '河东

```



```

    区','郯城县花园乡':'郯城县','蛟龙镇':'临沭县','册山街道办事处':'罗庄
    区','郑旺':'河东区','蒙阴县':'蒙阴县','罗西办事处':'高新区','郯城县沙墩
    镇':'郯城县','太平':'河东区','青驼镇':'沂南县','界牌':'蒙阴县','高庄镇':'沂
    水县','金岭镇':'兰陵县','岱崮':'蒙阴县','崔家峪镇':'沂水县','蒙阴街道':'蒙
    阴县','高新区区直社区':'高新区','九曲':'河东区','桃墟':'蒙阴县','旧寨':'蒙
    阴县','垛庄':'蒙阴县','高都':'罗庄区','野店':'蒙阴县','道托镇':'沂水县'}
53 data['QH'] = data['RYQH'].map(shuju)
54
55 plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
56 plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
57
58 fig = plt.figure(figsize=(12,8))
59 plt.bar(range(16),data.QH.value_counts(),color='cyan',alpha=0.5)
60 plt.xlim(-1,16)
61 plt.title('人员区划统计')
62 for x,y in enumerate(data.QH.value_counts()):
63     plt.text(x,y+100,y,ha='center')
64 plt.xticks(range(16),['兰山区','罗庄区','平邑县','莒南县','市本级','临沭县','沂
    水县','兰陵县','郯城县','费县','沂南县','高新区','河东区','蒙阴县','经济开
    发区','工业园'],rotation=30);
65
66 data['JZ'] = data['JZQH'].map(shuju)
67 fig = plt.figure(figsize=(12,8))
68 plt.bar(range(14),data.JZ.value_counts(),color='brown',alpha=0.4)
69 plt.xlim(-1,14)
70 plt.title('就诊区划统计')
71 for x,y in enumerate(data.JZ.value_counts()):
72     plt.text(x,y+100,y,ha='center')
73 plt.xticks(range(14),['兰山区','市本级','罗庄区','平邑县','莒南县','沂水县','临
    沭县','兰陵县','郯城县','沂南县','费县','河东区','蒙阴县','高新
    区'],rotation=30);
74
75 fig = plt.figure()
76 plt.bar(range(2),data.XB.value_counts(),color='maroon',alpha=0.4)
77 plt.xlim(-1,2)
78 plt.title('分性别病患统计')
79 for x,y in enumerate(data.XB.value_counts()):
80     plt.text(x,y+100,y,ha='center',fontsize=15)
81 plt.xticks(range(2),['男','女']);
82
83 dd = data[data.NL<7].ZDMC.value_counts().reset_index()
84 import squarify

```

```

85 fig = plt . figure ( figsize =(10,8))
86 plot = squarify . plot ( sizes = dd.ZDMC, # 指定绘图数据
87                         label = dd[dd.ZDMC>30]['index'], # 指定标签
88                         alpha = 0.6, # 指定透明度
89                         value = dd[dd.ZDMC>30]['ZDMC'], # 添加数值标签
90                         edgecolor = 'white', # 设置边界框为白色
91                         linewidth =3 # 设置边框宽度为3
92                     )
93
94 ddd = data[ data . NL>70].ZDMC.value_counts().reset_index()
95 import squarify
96 fig = plt . figure ( figsize =(10,8))
97 plot = squarify . plot ( sizes = ddd.ZDMC, # 指定绘图数据
98                         label = ddd[ddd.ZDMC>30]['index'], # 指定标签
99                         alpha = 0.6, # 指定透明度
100                        value = dd[dd.ZDMC>30]['ZDMC'], # 添加数值标签
101                        edgecolor = 'white', # 设置边界框为白色
102                        linewidth =3 # 设置边框宽度为3
103                    )
104
105 # 为了做一个相关性分析，这里将一些字符型的字段，用转化成离散数字 sklearn
106 import seaborn as sns
107 from sklearn import preprocessing
108 enc = preprocessing .OrdinalEncoder()
109 fig = plt . figure ( figsize =(12,8))
110 data [[ 'XB','RYLB','YLLB','RYQH','JZQH','JGDJ','ZDMC']]=enc.fit_transform( data . loc
111                                    [:,[ 'XB','RYLB','YLLB','RYQH','JZQH','JGDJ','ZDMC']])
112 corr = data . corr ()
113 sns . heatmap(corr ,annot=True)
114
115 Cname=data['JGDJ']
116
117 num=np.array(range(0, len(Cname))) #用于记录每个地名
118     重复出现次数
119 print (num)
120 print (Cname)
121 space=range(0, len(Cname))
122 print (space, type(space))

```

```

122 space=pd.DataFrame(space,columns=['name'])
123 space['name'][0]=(Cname[2])
124 print(space)
125 print(data['name']) #取某一行
126
127
128 for i in range(len(Cname)):
129     k = 0 #记录次数
130     for j in range(len(Cname)):
131         if (Cname[i]==Cname[j]):
132             space['name'][i]=Cname[j]
133             k=k+1
134         else:
135             k=k
136     num[i] = k
137
138 # print(space)
139 # print(num)
140 num=pd.DataFrame(data=num,columns=['num'])
141 # print(num)
142 frame=[space,num] #进行两个合
    并concatdataframe
143 result=pd.concat(frame,axis=1) #axis为向右连接,
    =1 =0 为向下连接
144 # print(result)
145 result = result . drop_duplicates ('name',keep=' first ')

```

## 附录 2：数据可视化

```

1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4
5 data = pd.read_csv(r'E:\2020paper\Data visualization 医疗数据\ .csv ',encoding="gbk")
6 data.head()
7 data1 = {
8     'ZMDC主动脉弓狭窄动脉狭窄 ':'["", 股骨假体周围骨折肝肿瘤
9     ",", 慢性丹囊炎矽肺膝关节滑膜囊肿
10     ",", 骨恶性肿瘤主动脉瓣关闭不全
11     ",", 蛛网膜下腔出血
12     "],

```

```

13     'ZFY':[47772.13,24344.24,22613.36,
14         21825.21,21037.6,20392.86,19665.33,
15         19615.65,19194.55,18373.56]
16 }
17 df = pd.DataFrame(data1,index      =['1','2','3','4','5','6','7','8','9','10'])
18 df.head()
19 df['ZMDC']
20 import pandas as pd
21 from pyecharts import options as opts
22 from pyecharts . charts import Bar
23 from pyecharts . commons . utils import JsCode
24 from pyecharts . faker import Faker
25 from pyecharts . globals import ThemeType
26 cc = list ( df["ZMDC"].values )
27 cc
28 ccc = ['COA','ISR ','Femur ','HCC','ACC','SIS ','Silicosis ','TCGA','AR','SAH']画柱状
    图
29 #
30
31 values = []
32 for i in df.values:
33     dic = {}
34     dic["value"] = i[1]
35     values.append(dic)
36
37
38 c = (
39     Bar( init_opts =opts . InitOpts (width="750px", height="400px",theme = ThemeType.
        DARK ))
40     .add_xaxis( ccc )
41     .add_yaxis平均花费金额 ("", values , category_gap="40%")
42     . set_series_opts (
43         itemstyle_opts={
44             "normal": {
45                 "color": JsCode(
46                     ""new echarts . graphic . LinearGradient (0, 0, 0, 1, [{
47                     offset : 0,
48                     color: 'rgba(22, 77, 167, 1)'
49                 }, {

```

```

50         offset : 1,
51         color: 'rgba(77, 160, 321, 1)'
52     }], false)"""
53     ),
54     "barBorderRadius": [30, 30, 30, 30],
55     "shadowColor": "rgb(77, 160, 321)",
56     }
57 }
58 )
59 .set_global_opts ( title_opts =opts.TitleOpts ( title 看病花费金额="top10",
60 pos_bottom = "87%", pos_right = "5%"))
61 #.render(" bar_border_radius .html")
62 )
63 c.render_notebook()
64 # 绘制动态榜单
65 month_lis = 年月年月年月 ['201912','20201','20202']
66 month_data_lis = [month12_data,month1_data,month2_data]
67 color_function = """
68     function (params) {
69         if (params.value < 20000)
70             return ' #FF7256';
71         else if (params.value >= 20000 && params.value < 40000)
72             return '#19ed95';
73         else return '#3333cc';
74     }
75     """
76 # 新建一个对象 timeline
77 t2 = Timeline(
78     init_opts=opts.InitOpts (
79         bg_color='#FFE4E1', # 设置背景颜色
80         theme='macarons',      # 设置主题
81         width='900px',         # 设置图的宽度
82         height='700px'         # 设置图的高度
83     )
84 )
85 t2.add_schema(
86     is_auto_play = True,      # 是否自动播放
87     play_interval = 1500,     # 播放速度
88     is_loop_play = True,      # 是否循环播放

```

```

89 )
90
91 for i,data1 in zip(month_lis,month_data_lis):
92     day = i
93     bar = Bar(
94         init_opts=opts.InitOpts (
95             bg_color='#FFE4E1', # 设置背景颜色
96             theme='essos ',      # 设置主题
97             width='1200px',      # 设置图的宽度
98             height='600px'      # 设置图的高度
99         )
100     )
101     bar.add_xaxis(data1产品类别 [' id ']. tolist ())
102     bar.add_yaxis(销售额
103         ",
104         data1销售额 ["]. round(2). tolist (),
105         category_gap="40%"
106     )
107     bar.reversal_axis ()
108     bar.set_series_opts ( # 自定义图表样式
109         label_opts=opts.LabelOpts(is_show=True,position = " right "), # 是否显示数据
            标签
110         itemstyle_opts={
111             "normal": {
112                 "color": JsCode( color_function ),      # 调整柱子颜色渐变
113                 'shadowBlur': 8,    # 光影大小
114                 "barBorderRadius": [100, 100, 100, 100], # 调整柱子圆角弧度
115                 "shadowColor": "#E9B7D3", # 调整阴影颜色
116                 'shadowOffsetY': 6,
117                 'shadowOffsetX': 6, # 偏移量
118             }
119         }
120     )
121     bar.set_global_opts (
122         # 标题设置
123         title_opts=opts.TitleOpts (
124             title 每月各产品类别销售额='榜单top', # 主标题
125             subtitle =", # 副标题
126             pos_left =' center ', # 标题展示位置

```

```

127         title_textstyle_opts = dict(color='#5A3147'), # 设置标题字体颜色
128         subtitle_textstyle_opts = dict(color='#5A3147')
129     ),
130     legend_opts=opts.LegendOpts(
131         is_show=True, # 是否显示图例
132         pos_left='right', # 图例显示位置
133         pos_top='3%', 图例距离顶部的距离 #
134         orient='vertical', # 图例水平布局
135         textstyle_opts=opts.TextStyleOpts(
136             color='#5A3147', # 颜色
137             font_size='13', # 字体大小
138             font_weight='bolder', # 加粗
139         ),
140     ),
141     tooltip_opts=opts.TooltipOpts(
142         is_show=True, # 是否使用提示框
143         trigger='axis', # 触发类型
144         is_show_content=True,
145         trigger_on='mousemove|click', # 触发条件, 点击或者悬停均可出发
146         axis_pointer_type='cross', # 指示器类型, 鼠标移动到图表区可以查看效果
147         # formatter = '{a}<br>{b}:{c人}' # 文本内容
148     ),
149     yaxis_opts=opts.AxisOpts(
150         is_show=True,
151         splitline_opts=opts.SplitLineOpts(is_show=False), # 分割线
152         axistick_opts=opts.AxisTickOpts(is_show=False), # 刻度不显示
153         axislabel_opts=opts.LabelOpts( # 坐标轴标签配置
154             font_size=13, # 字体大小
155             font_weight='bolder' # 字重
156         ),
157     ), # 关闭轴显示Y
158     xaxis_opts=opts.AxisOpts(
159         boundary_gap=True, # 两边不显示间隔
160         axistick_opts=opts.AxisTickOpts(is_show=True), # 刻度不显示
161         splitline_opts=opts.SplitLineOpts(is_show=False), # 分割线不显示
162         axisline_opts=opts.AxisLineOpts(is_show=True), # 轴不显示
163         axislabel_opts=opts.LabelOpts( # 坐标轴标签配置
164             font_size=13, # 字体大小
165             font_weight='bolder' # 字重

```

```

166         ),
167     ),
168 )
169
170 t2.add(bar, day)
171
172 t2.render_notebook()
173 pro_category = {
174     'ZMDC':['COA','ISR','Femur','HCC','ACC','SIS ',' Silicosis ',' TCGA','AR','SAH'],
175     '#ZMDC主动脉弓狭窄动脉狭窄股骨假体周围骨折肝肿瘤慢性丹囊
        炎 ':'["","","", "矽肺膝关节滑膜囊肿骨恶性肿瘤主动脉瓣关闭不全蛛网膜下
        腔出血
        ","",""], 一级医院
176     ":[47772.13,24344.24,22613.36,21825.21,
177     21037.6,20392.86,19665.33,19615.65,19194.55,18373.56], 二级医院
178     ":[49908.15,25633.44,23065.43,22134.34,
179     20056.33,21003.2,19003.55,19522.67,18977.77,19004.77],
180 }
181
182 pro_category = pd.DataFrame(pro_category,index =['1','2',
183     '3','4','5','6','7','8','9','10'])
184 pro_category
185
186 def echarts_bar(x,y,y2, title = 主标题 "
187 , subtitle = 副标题 ", label = 图例 ", label2 = 图例 '2', color='color') :
188     """
189     x: 函数传入轴标签数据x: 函数传入轴数据
190     yy: 主标题
191     title : 副标题
192     subtitle : 图例
193     label
194     """
195     bar = Bar(
196         init_opts=opts.InitOpts (
197             bg_color='#3333cc', # 设置背景颜色
198             theme='dark ', # 设置主题
199             width='900px ', # 设置图的宽度
200             height='600px' # 设置图的高度
201         )
202     )

```



```

203 bar.add_xaxis(x)
204 bar.add_yaxis( label ,y,
205     label_opts=opts.LabelOpts(is_show=True) # 是否显示数据
206     ,category_gap="60%" # 柱子宽度设置
207     ,yaxis_index=0
208 )
209 bar.add_yaxis( label2 ,y2,
210     label_opts=opts.LabelOpts(is_show=True) # 是否显示数据
211     ,category_gap="60%" # 柱子宽度设置
212     ,yaxis_index=1
213 )
214 bar.set_series_opts ( # 自定义图表样式
215     label_opts=opts.LabelOpts(
216         is_show=True,
217         position='top', # position 标签的位置可选 'top', " left , '
218         ' right , " bottom, " inside , " insideLeft , " insideRight '
219         font_size=15,
220         color='white',
221         font_weight = 'bolder', # font_weight 文字字体的粗细
222         'normal, " bold, " bolder, " lighter '
223         font_style = 'oblique', # font_style 文字字体的风格, 可选
224         'normal, " italic , " oblique'
225     ),
226     itemstyle_opts={
227         "normal": {
228             "color": color, # 调整柱子颜色渐变
229             'shadowBlur': 15, # 光影大小
230             "barBorderRadius": [100, 100, 100, 100], # 调整柱子圆角弧度
231             "shadowColor": "#0EEEF9", # 调整阴影颜色
232             'shadowOffsetY': 10,
233             'shadowOffsetX': 10, # 偏移量
234         }
235     }
236 )
237 bar.set_global_opts (
238     # 标题设置
239     title_opts=opts.TitleOpts (
240         title = title , # 主标题
241         subtitle = subtitle , # 副标题

```

```

242         pos_left='center', # 标题展示位置
243         title_textstyle_opts=dict(color='#fff') # 设置标题字体颜色
244     ),
245     # 图例设置
246     legend_opts=opts.LegendOpts(
247         is_show=True, # 是否显示图例
248         pos_bottom='bottom',
249         orient='horizontal' # 图例水平布局
250     ),
251     tooltip_opts=opts.TooltipOpts(
252         is_show=True, # 是否使用提示框
253         trigger='axis', # 触发类型
254         is_show_content=True,
255         trigger_on='mousemove|click', # 触发条件, 点击或者悬停均可出发
256         axis_pointer_type='cross', # 指示器类型, 鼠标移动到图表区可以查看效果
257     ),
258     yaxis_opts=opts.AxisOpts(
259         is_show=True,
260         splitline_opts=opts.SplitLineOpts(is_show=False), # 分割线
261         axistick_opts=opts.AxisTickOpts(is_show=False), # 刻度不显示
262         axislabel_opts=opts.LabelOpts( # 坐标轴标签配置
263             font_size=13, # 字体大小
264             font_weight='bolder' # 字重
265         ),
266     ), # 关闭轴显示Y
267     xaxis_opts=opts.AxisOpts(
268         boundary_gap=True, # 两边不显示间隔
269         axistick_opts=opts.AxisTickOpts(is_show=True), # 刻度不显示
270         splitline_opts=opts.SplitLineOpts(is_show=False), # 分割线不显示
271         axisline_opts=opts.AxisLineOpts(is_show=True), # 轴不显示
272         axislabel_opts=opts.LabelOpts( # 坐标轴标签配置
273             font_size=13, # 字体大小
274             font_weight='bolder' # 字重
275         ),
276     ),
277 )
278 bar.extend_axis(yaxis=opts.AxisOpts())
279 return bar.render_notebook()

```

```

280
281 color = {
282     'type': 'linear',
283     'x': 0,
284     'y': 0,
285     'y2': 0,
286     'x2': 1,
287     'colorStops': [
288         {'offset': 0, 'color': 'rgba(0, 244, 255, 0.8)' },
289         {'offset': 1, 'color': 'rgba(0, 77, 167, 0.8)'}],
290     'global': False
291     }
292 echarts_bar(pro_category['ZMDC'].tolist(), pro_category一级医院[''].tolist(),
293             pro_category二级医院[''].tolist(), title 不同医院费用对
                比=", subtitle 花费金额对比柱状图=",
294             label一级医院=", label2二级医院=", color=color)
295
296 shuju = data['JGDJ'].value_counts()
297 shuju
298
299 from pyecharts.charts import *
300 from pyecharts import options as opts 饼图
301 """
302 L1 = 社区医疗二级医院三级医院一级医院 ['', '', '', '']
303 num = [5837, 5029, 2974, 2725]
304 C = Pie()
305 C.add饼图('',[ list(z) for z in zip(L1,num)])
306 C.set_global_opts(title_opts=opts.TitleOpts(title=饼图=''))
307 #C.set_series_opts(label_opts=opts.LabelOpts(formatter='{b}:{c}'))
308 C.render_notebook()
309
310 from pyecharts import options as opts
311 from pyecharts.charts import Pie
312 from pyecharts.commons.utils import JsCode
313 from pyecharts.globals import ThemeType
314
315 df1 = data['ZDMC'].value_counts()
316 df0 = df1.head(20)
317

```

```

318 # color={
319 #     'type': 'linear ',
320 #     'x': 0,
321 #     'y': 1,
322 #     'x2': 0,
323 #     'y2': 0,
324 #     'colorStops': [
325 #         {'offset': 0, 'color': 'black' },
326 #         {'offset': 1, 'color': 'orange'}],
327 #     'global': False
328 # }
329 c1 = (
330     Pie()
331     .add(' test ', [ list (z) for z in zip(df0.index.values . tolist (), df0.values . tolist
332         ())],
333         radius=['30%', '100%'],
334         center=['50%', '60%'],
335         rosetype='area ',
336         )
337     . set_global_opts ( title_opts =opts. TitleOpts ( title 地区景点数量 ="),
338                       legend_opts=opts. LegendOpts(is_show=False),
339                       toolbox_opts=opts. ToolboxOpts()
340                       )
341     . set_series_opts ( label_opts=opts. LabelOpts(is_show=True, position =' inside ',
342         font_size=12,
343         formatter='{b}: {c }', font_style ='
344         italic ',
345         font_weight='bold ', font_family ='
346         Microsoft YaHei'
347         ),
348         # itemstyle_opts =opts. ItemStyleOpts( color=color)
349         )
350     )
351 c1.render_notebook()
352
353 dff = data [' RYLB'].value_counts()
354 dff
355
356 from pyecharts . charts import *

```

```

353 from pyecharts import options as opts 圆环图
354     """
355 L1 = 居民退休类型退休类型在职类型在职类型离休 ["", "1", "2", "1", "2", ""]
356 num = [10104, 1510, 2984, 1458, 445, 64]
357 C = Pie()
358     """ 调节大小 radius """
359 C.add圆环图("", [list(z) for z in zip(L1, num)], radius=["45%", "75%"])
360 C.set_global_opts(title_opts=opts.TitleOpts(title="圆环图"),
361 legend_opts=opts.LegendOpts(orient="vertical", pos_top="2%", pos_left="5%"))
362 C.render_notebook()
363
364 dfff = data['XB'].value_counts()
365 dfff
366
367 import pyecharts.options as opts
368 from pyecharts.charts import Pie
369
370     """
371
372     """
373
374 inner_x_data = 男["", 女[""]]
375 inner_y_data = [8551, 8012]
376 inner_data_pair = [list(z) for z in zip(inner_x_data, inner_y_data)]
377
378 outer_x_data = 社区医疗二级医院三级医院一级医院 ["", "", "", ""]
379 outer_y_data = [5837, 5029, 2974, 2725]
380 outer_data_pair = [list(z) for z in zip(outer_x_data, outer_y_data)]
381
382 c = (
383     Pie(init_opts=opts.InitOpts(width="1600px", height="800px"))
384     .add(
385         series_name=涨跌情况="",
386         data_pair=inner_data_pair,
387         radius=[0, "30%"],
388         label_opts=opts.LabelOpts(position="inner"),
389     )
390     .add(
391         series_name=涨跌情况="",

```

```

392 radius=["40%", "55%"],
393 data_pair=outer_data_pair ,
394 label_opts=opts.LabelOpts(
395     position="outside ",
396     formatter="{a|{a}}{abg|}\n{hr|}\n {b|{b}: }{c} {per|{d}%} ",
397     background_color="#eee",
398     border_color="#aaa",
399     border_width=1,
400     border_radius=4,
401     rich={
402         "a": {"color": "#999", "lineHeight": 22, "align": "center"},
403         "abg": {
404             "backgroundColor": "#e3e3e3",
405             "width": "100%",
406             "align": "right",
407             "height": 22,
408             "borderRadius": [4, 4, 0, 0],
409         },
410         "hr": {
411             "borderColor": "#aaa",
412             "width": "100%",
413             "borderWidth": 0.5,
414             "height": 0,
415         },
416         "b": {"fontSize": 16, "lineHeight": 33},
417         "per": {
418             "color": "#eee",
419             "backgroundColor": "#334455",
420             "padding": [2, 4],
421             "borderRadius": 2,
422         },
423     },
424 ),
425 )
426 . set_global_opts (legend_opts=opts.LegendOpts(pos_left=" left ", orient=" vertical ")
427 )
428 . set_series_opts (
429     tooltip_opts=opts.TooltipOpts(
430         trigger="item", formatter="{a} <br/>{b}: {c} ({d}%)"

```

```

430         )
431     )
432     .render_notebook()
433 )
434
435 c
436
437 dffff = data ['YLLB'].value_counts()
438 dffff
439
440 from pyecharts import options as opts
441 from pyecharts.charts import PictorialBar
442 from pyecharts.globals import SymbolType
443
444 location = 门诊慢性病 [" 普通住院 ," , 普通门诊 " ; 转省外住院特殊疾病门
445             诊 ' , 转省内住院无责任人意外伤害 " 生育住院 ," , 转省外门诊慢性病
446             " 转省内门诊慢性病 ," 大病购药 ," 机关事业单位生育住院 ," 住院前
447             急诊 ," 转省平台住院单病种住院 ," ; ]
448 values = [7245, 3222 ,2117, 1334 ,1023, 490 , 423 ,387 ,194, 68 ,36, 13 , 10 , 2,1]
449
450 c = (
451     PictorialBar ()
452     .add_xaxis( location )
453     .add_yaxis(
454         "",
455         values,
456         label_opts=opts.LabelOpts(is_show=False),
457         symbol_size=18,
458         symbol_repeat="fixed",
459         symbol_offset=[0, 0],
460         is_symbol_clip=True,
461         symbol=SymbolType.ROUND_RECT,
462     )
463     .reversal_axis ()
464     .set_global_opts (
465         title_opts =opts. TitleOpts ( title 医疗类别=""),
466         xaxis_opts=opts.AxisOpts(is_show=False),
467         yaxis_opts=opts.AxisOpts(
468             axistick_opts =opts.AxisTickOpts(is_show=False),

```

```
467         axisline_opts =opts.AxisLineOpts(  
468             linestyle_opts =opts.LineStyleOpts( opacity=0)  
469         ),  
470     ),  
471 )  
472 .render_notebook()  
473 #.render(" pictorialbar_base .html")  
474 )  
475 c
```