温州大学

**WENZHOU UNIVERSITY**



期 末 考 核

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学 期 | 2023-2024学年第二学期 | | | |
| 课 程 | **数据可视化** | | | |
| 专　 业 | 人工智能 | 班　级 | | 22人工智能1 |
| 学生姓名 | 李凯荣 | | 学　号 | 22211360121 |
| 指导教师 | 张楠 | | | |

**目录**

[摘要 3](#_Toc170650558)

[第一章 背景介绍 4](#_Toc170650559)

[第二章 数据信息 5](#_Toc170650560)

[**2.1 数据来源 5**](#_Toc170650561)

[**2.2 数据说明 5**](#_Toc170650562)

[第三章 数据分析与数据可视化 6](#_Toc170650563)

[**3.1 数据与语言 6**](#_Toc170650564)

[**3.2 数据预处理 6**](#_Toc170650565)

[**3.3 数据可视化（点击可以跳转html文件） 7**](#_Toc170650566)

[**3.3.1 按年的各市医疗人员总数条形图 7**](#_Toc170650567)

[**3.3.2各市医疗人员总数变化折线图 10**](#_Toc170650568)

[**3.3.3各市卫生资源雷达图 12**](#_Toc170650569)

[**3.3.4各市医疗人员总数3D条形图 12**](#_Toc170650570)

[**3.3.5各市千人均医疗人员总数3D条形图 13**](#_Toc170650571)

[**3.3.6各市医疗人员总数变化热力图 13**](#_Toc170650572)

[**3.3.7三市卫生资源分布箱型图 14**](#_Toc170650573)

[**3.3.8温州市在全省占比水滴图 15**](#_Toc170650574)

[**3.3.9温州市于全省千人均对比象形条形图 16**](#_Toc170650575)

[第四章 总结 17](#_Toc170650576)

[附录 18](#_Toc170650577)

[**1.浙江各省医疗人员总数 18**](#_Toc170650578)

[**2.浙江各省医疗资源属性雷达图 19**](#_Toc170650579)

[**3.浙江各市各职业数量按年变化折线图 21**](#_Toc170650580)

[**4.浙江全省医疗人员热力图 22**](#_Toc170650581)

[**5.浙江全省医疗资源总数相关条形图 23**](#_Toc170650582)

[**6.浙江三市（杭州、宁波、温州）卫生资源分布箱型图 24**](#_Toc170650583)

[**7.温州市每千人均象形柱状图 25**](#_Toc170650584)

[**8.温州市卫生资源在全省占比水滴图 27**](#_Toc170650585)

[**9.2021年最新千人均3D地图柱状图 29**](#_Toc170650586)

摘要

本文研究了浙江省各市医疗卫生资源分布数据详情。并且对浙江省各市的医疗卫生资源进行可视化数据分析，制成图片。本文的目的在于，通过数据可视化，实现对浙江省各市现有医疗资源的状况有清晰了解，以便于优化未来的医疗配置参考和预测发展趋势。本文采用了python编程语言，通过绘制分布图、散点图、热力图等，观察浙江省内各市在2018到2019年内，执业医师、卫生技术人员、注册护士、机构数量、床位的变化和分布规律。以及通过现有趋势，判断未来的发展趋势。依据数据可视化形成的图片，可以清晰呈现浙江省2018年到2021年的医疗资源配置数据分布详情。

本文的研究背景在于进入新时代，医疗作为提升民生福祉的重要一环，是不可忽视的重要内容。合理的医疗资源配置对于提高医疗服务的可及性和公平性至关重要。通过优化医疗资源分配结构，能够提高医疗服务质量。各市间医疗结构和资源分配的公平性得到保障，还能提高公共卫生的应对能力，像是在突发公共卫生事件能有较高的应急响应能力，确保迅速弥补医疗资源短缺的作用，特别是经历过疫情等突发状况，这样的研究十分有意义。优化资源的利用效率，能够最大化实现效益和资源的平衡。通过对于各市的医疗资源的研究，也为政府和医疗管理机构提供科学依据，有利于支持政策制定和实现。因此，本研究具有重要的显示意义。

本文采用python编程语言进行编程实现数据可视化。Python是一种高级编程语言，由Guido van Rossum于1991年首次发布。它以简洁、易读和强大的标准库著称，广泛应用于数据科学、人工智能、Web开发、自动化脚本等领域。Python支持多种编程范式，包括面向对象编程、函数式编程和过程式编程，使其成为开发者的热门选择。Python的数据可视化主要依靠第三方可视化绘图库实现。本文采用百度的pyecharts数据可视化库来实现。pyecharts是一个用于生成ECharts图表的Python库。ECharts是一个由百度团队开发的开源数据可视化库，以其强大的功能和高度的定制化能力广受欢迎。pyecharts使得Python开发者可以方便地使用ECharts的强大功能来创建丰富的交互式图表。并且pyechart库可以十分容易地生成html文件，实现可交互的可视化页面的效果

在具体数据可视化的分析过程中，搜集来自于浙江公开的医疗资源分配数据，输入到文件中，通过编程实现具体的数据可视化，将结果直观地展现。

**关键词**：浙江省医疗资源 数据可视化 python pyecharts

1. 背景介绍

浙江各省市医疗人员资源配置可视化是将浙江省的医疗相关资源数量数据以可视化的方式呈现。医疗资源内容包含床位、执业医师、卫生技术人员、注册护士、医疗机构等。卫生资源分配一直是提高民生福祉的重要内容。通过数据可视化呈现浙江省各市具体医疗资源的拥有量，帮助人们更好地优化卫生资源配置结构。

通过数据可视化展示浙江省的医疗卫生资源数据，可以直观地看到医疗资源配置的地区分布和数量变化趋势。这种可视化包括图表、图形和图像等形式，以展示各市医疗资源的数量和配置情况。例如，总数条形图可以清晰展示各市不同类型医疗资源的绝对数量；雷达图能够综合展示某一市的多种资源分配情况，便于比较各项资源的均衡度；折线图则用于展示某一医疗资源在各市之间的变化趋势。

可视化的目的是为了让人们了解当前浙江省区域内的医疗资源配置情况，便于对比各市间的医疗资源条件，评估医疗水平以及突出全省医疗协调能力和医疗水平的公平性。通过数据可视化，人们可以更好地了解当地的医疗资源配给情况，这为决策者提供了实际且具体的现状数据。

此外，通过这些数据可视化，还可以发现一些潜在的问题和趋势。例如，某些市可能存在医疗资源短缺，需要引起注意并采取相应措施；某些地区医疗资源过剩，可能需要重新分配和优化。这些信息对未来平均医疗资源、优化卫生资源分配提供了重要参考，有助于提高整体医疗服务水平，确保所有居民都能获得公平和高质量的医疗服务。

总的来说，浙江省医疗人员资源配置的可视化研究，不仅为政府和相关部门提供了科学决策依据，也为公众提供了透明、直观的医疗资源分布信息。通过这种方式，可以促进社会各界对医疗资源分配的关注和讨论，共同推动医疗资源的合理配置和优化，最终提高全省的公共卫生水平和居民的健康福祉。

第二章 数据信息

2.1 数据来源

医疗资源配置数据是研究和优化医疗服务的重要基础。了解这些数据的主要来源对于制定科学的医疗资源分配策略至关重要。以下是几种主要的医疗资源配置数据来源：

1. 政府和公共卫生机构

政府和公共卫生机构是医疗资源配置数据的主要来源之一。这些机构通常会收集并发布全国或地区范围内的医疗资源数据，包括医院数量、床位数、医护人员数量、医疗设备数量等。

2. 医院和医疗机构

各级医院和医疗机构是医疗资源配置数据的重要提供者。这些机构内部的数据系统记录了详细的资源信息，包括医院的科室设置、医护人员配置、医疗设备使用等。这些数据通常用于内部管理和对外报告。

3. 保险公司

保险公司通过健康保险理赔数据积累了大量的医疗资源使用信息。通过分析保险公司提供的数据，可以了解不同地区和人群的医疗服务需求和资源使用情况。这对于优化资源配置和制定政策有着重要的参考价值。

4. 科研机构和高校

从事医疗卫生研究的科研机构和高校也会收集和分析大量的医疗资源数据。这些数据通常通过科学研究、调查问卷和实地考察等方式获得，并用于研究报告和学术论文中，为医疗资源配置提供理论支持和数据基础。

5. 国际组织和数据库

一些国际组织和数据库也提供全球或区域范围内的医疗资源配置数据。例如：

世界卫生组织（WHO）：提供全球范围内的健康和医疗资源数据，包括全球卫生观察站。

世界银行：提供关于各国健康系统和资源分配的数据，帮助各国进行资源评估和规划。

6. 数据采集与调研公司

一些专业的数据采集与调研公司通过市场调研和数据分析，提供详细的医疗资源配置报告。这些公司通常使用先进的数据采集技术和分析方法，确保数据的准确性和时效性。

**2.2 数据说明**

1. 年份：数据获取的具体时间。
2. 地区：数据的适用区域。
3. 床位：该地区的医疗用床的具体统计数量
4. 执业医师：执业医师是指经过专业医学教育和培训，并通过国家执业医师资格考试，获得执业医师资格证书的医生。执业医师可以在医疗机构中从事医疗、预防、保健等工作，为患者提供医疗服务。该项于此指的是执业医师的数量。
5. 卫生技术人员：卫生技术人员是指在医疗卫生系统中从事各种技术支持和服务的专业人员。与医生、护士等直接提供临床服务的医务人员不同，卫生技术人员主要负责医疗技术操作、医学检验、影像诊断、设备维护等方面的工作，确保医疗服务的顺利进行和高质量。
6. 注册护士：注册护士是指经过专业护理教育和培训，取得护士资格证书并在卫生行政部门注册登记，具备合法执业资格的护理人员。注册护士在医疗系统中扮演着至关重要的角色，负责提供直接的护理服务、患者教育和支持等工作。

第三章 数据分析与数据可视化

**3.1 数据与语言**

所用数据的主要来源自官方的浙江数据开放网站（[浙江·数据开放 (zjzwfw.gov.cn)](https://data.zjzwfw.gov.cn/jdop_front/index.do)）。具体为浙江各省市医疗人员数量文件。

所用编程语言为python，数据可视化基于pyecharts可视化库。

**3.2 数据预处理**

数据预处理具体步骤：

1、数据获取：从浙江数据开放官网获取数据，包括浙江省各市各职业医疗人员数量于2019年至2021年的统计。

2、数据清洗：检查数据中的缺失值、重复值或错误值。根据数据的质量和完整性，可以选择删除、填充或修正这些值。

3、属性选择：根据需要选择和提取感兴趣的属性。可以根据分析目标，选择包含执业医师、卫生技术人员、注册护士等数据的属性。

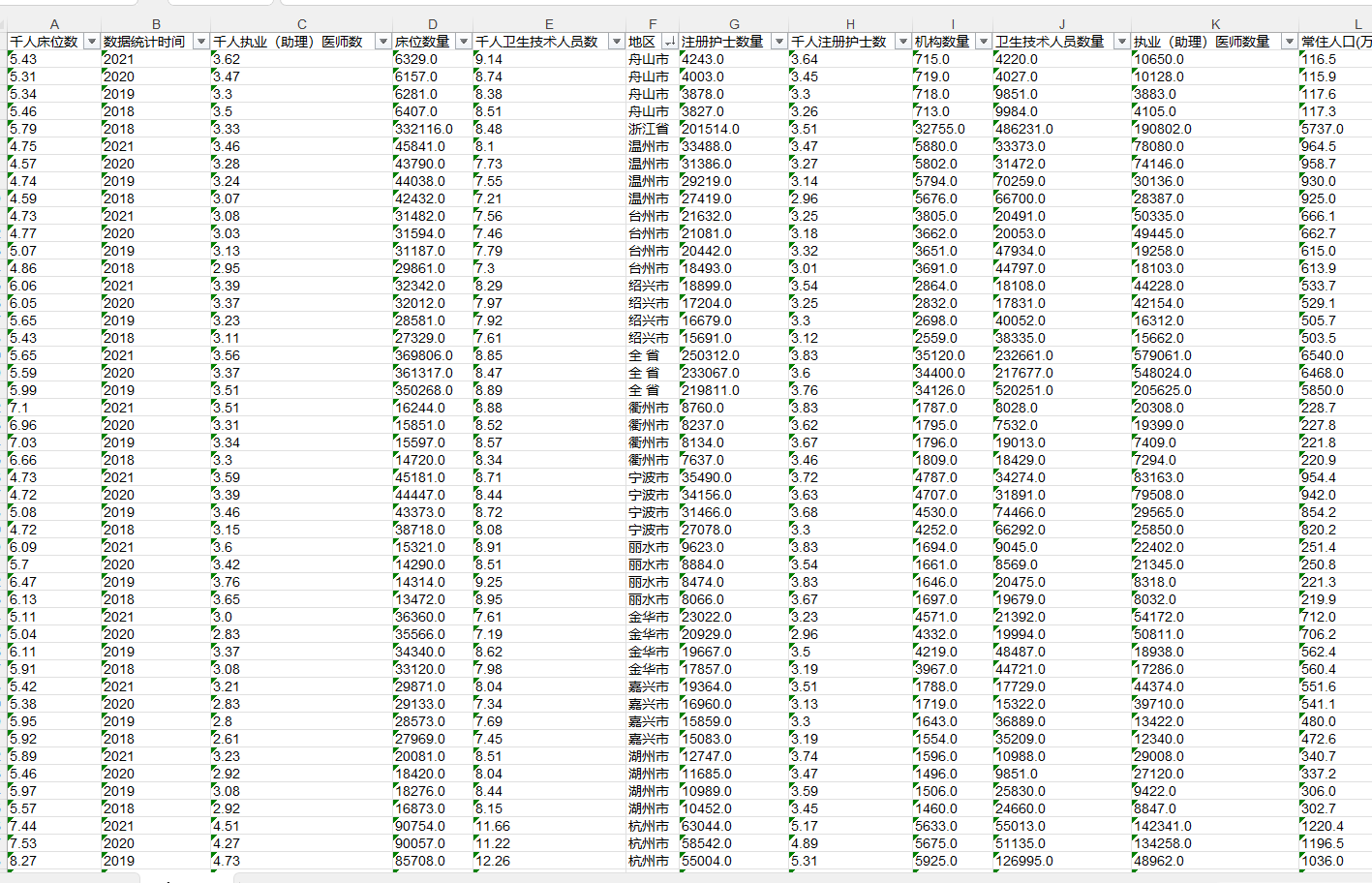
4、数据转换：对一些属性进行适当的转换。例如，将统一小数单位等。

5、数据归一化：如果需要对不同范围的数据进行比较，可以进行数据归一化。例如，使用标准化或归一化方法将不同范围的积分值缩放到相似的范围内。

6、数据集成：如果需要将来自不同数据源的数据进行整合，可以进行数据集成。确保数据源之间的属性和值的一致性和匹配性。

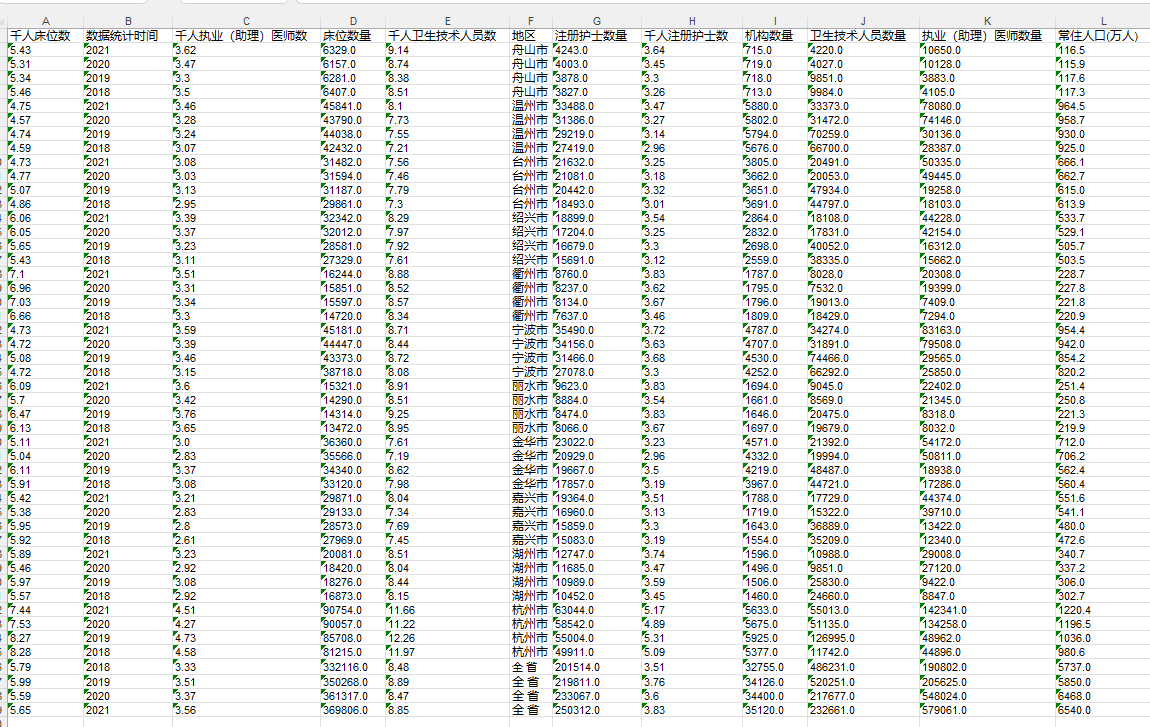
7、数据可视化：使用适当的图表、图形或可视化工具对预处理后的数据进行可视化。这有助于理解数据分布、趋势和关系。

1. 错误修改具体内容：



原生数据文件中，发现浙江省项和全省项。两者的意义是一样的，因此是一种命名错误，这里将其统一为全省。

2. 数据选择和整理具体内容：



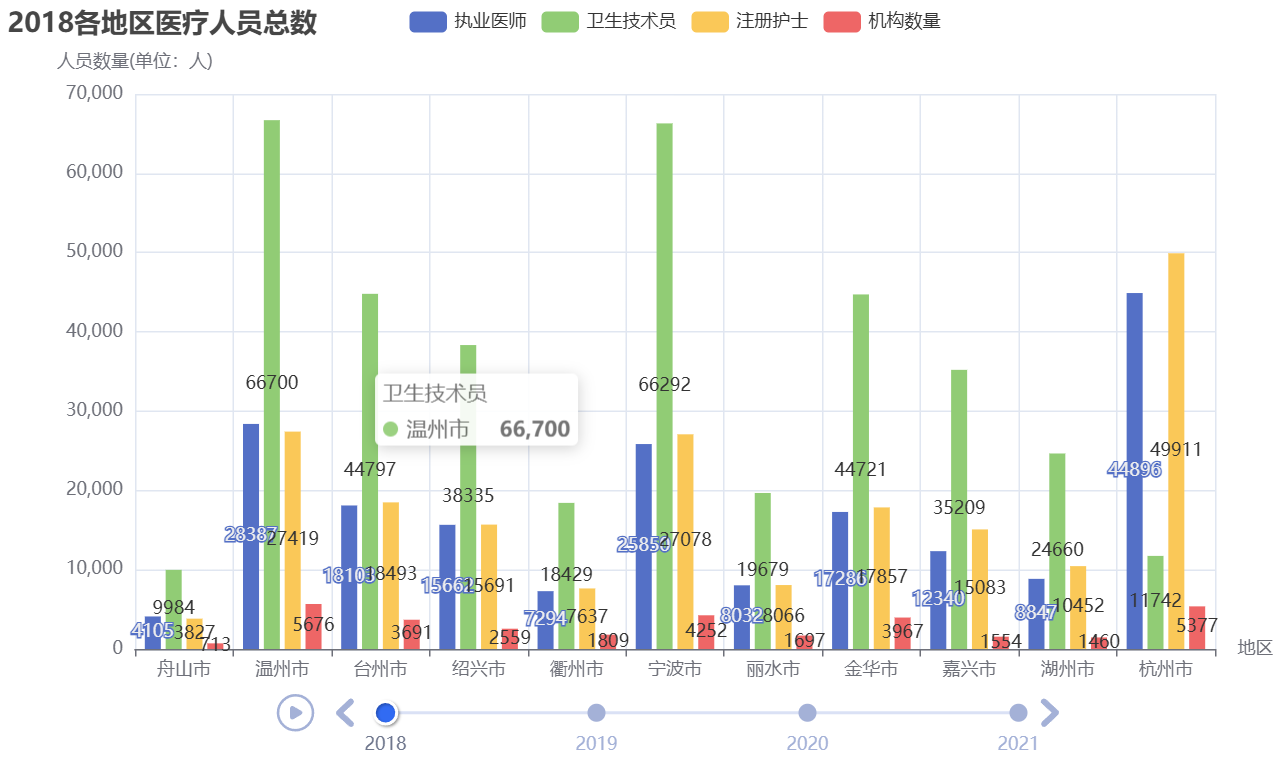


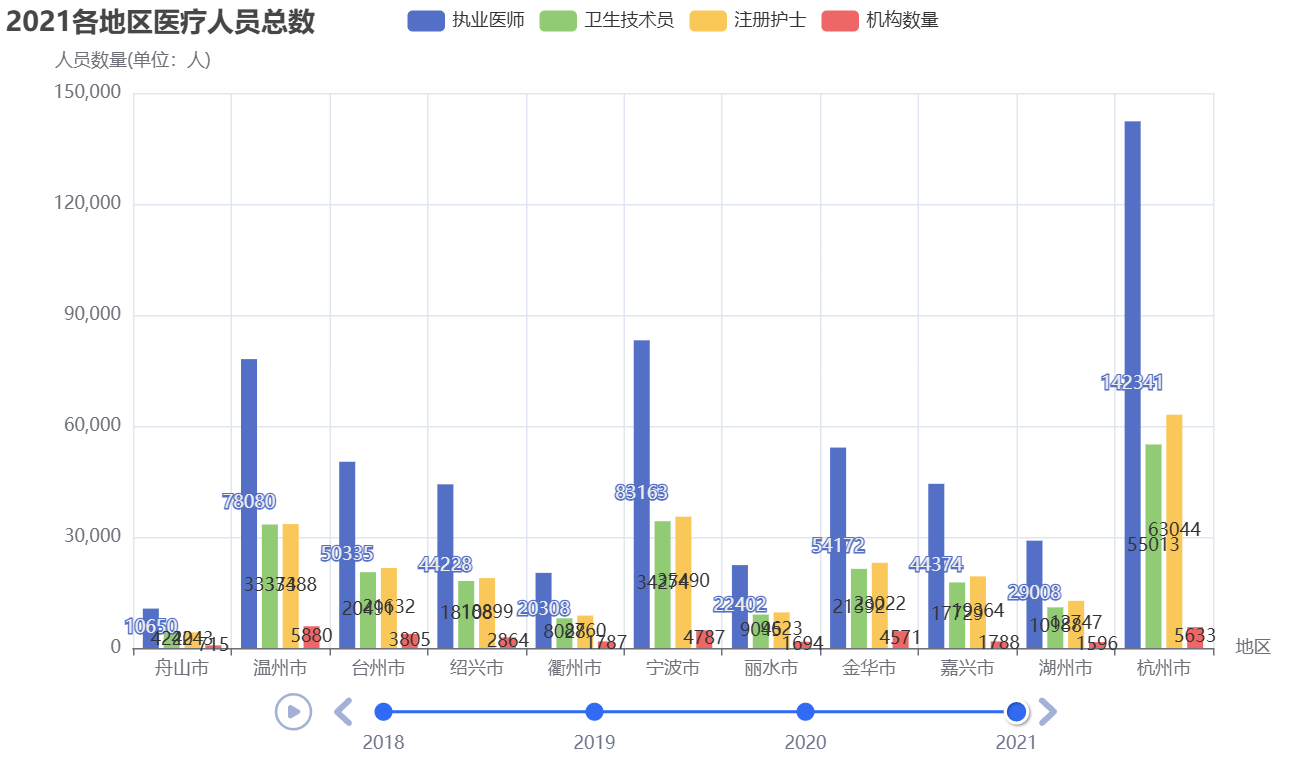
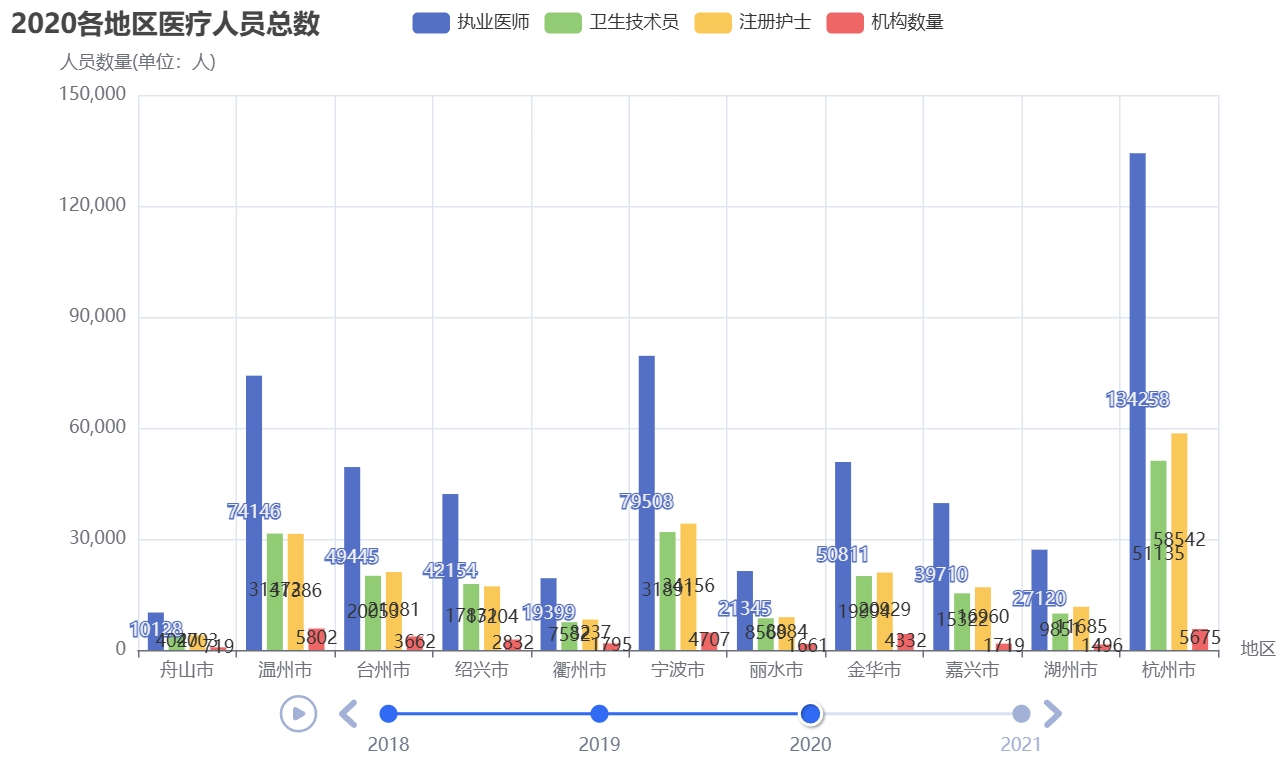
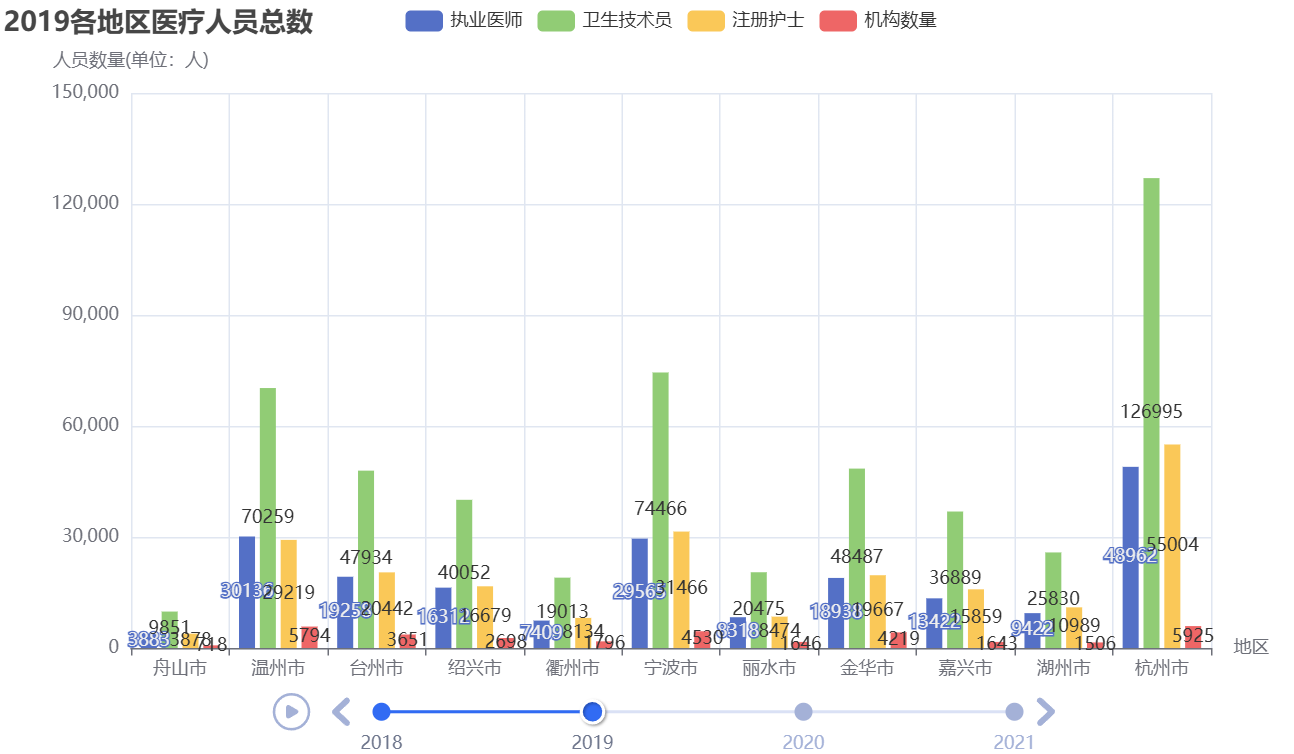
将数据集首先按地区排列，接着按时间倒序排列，初步将数据整理为更清晰的结构。

接着便可以开始数据可视化的具体内容实现。

**3.3 数据可视化（点击可以跳转html文件）**

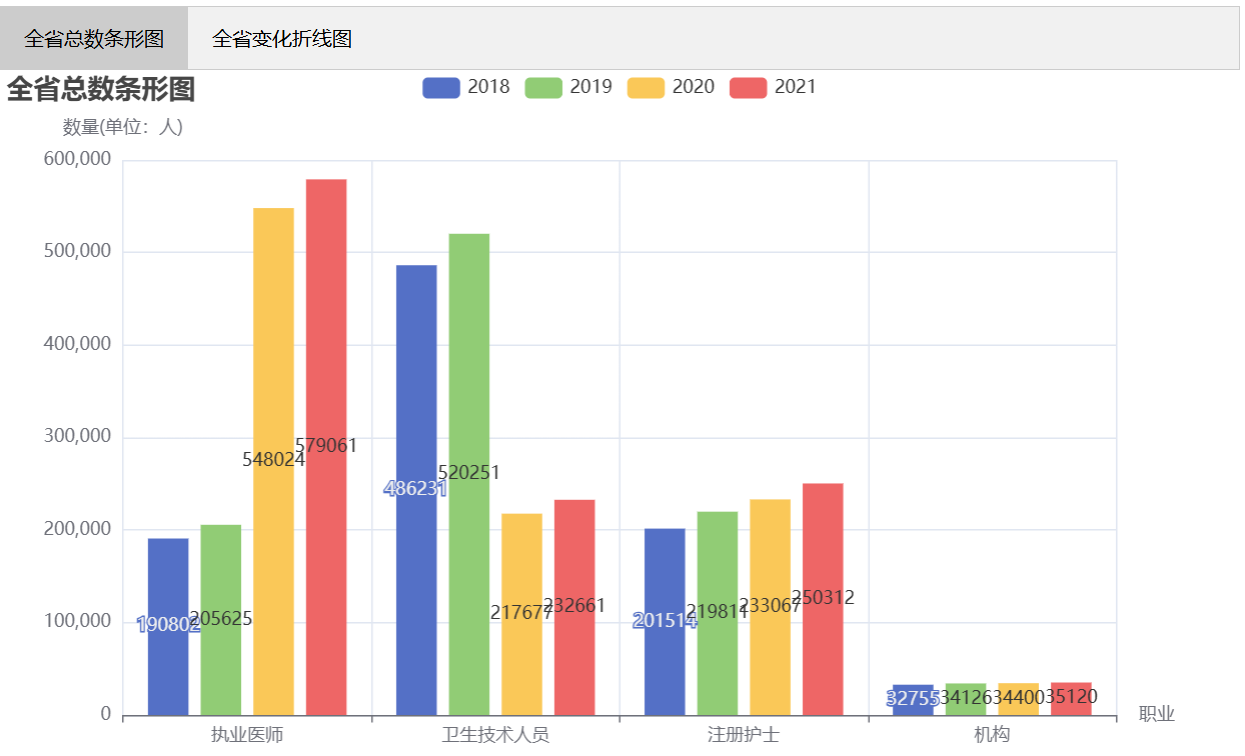
3.3.1 [按年的各市医疗人员总数条形图](Charts_Code/Graphics/各市区医疗人员总数.html)

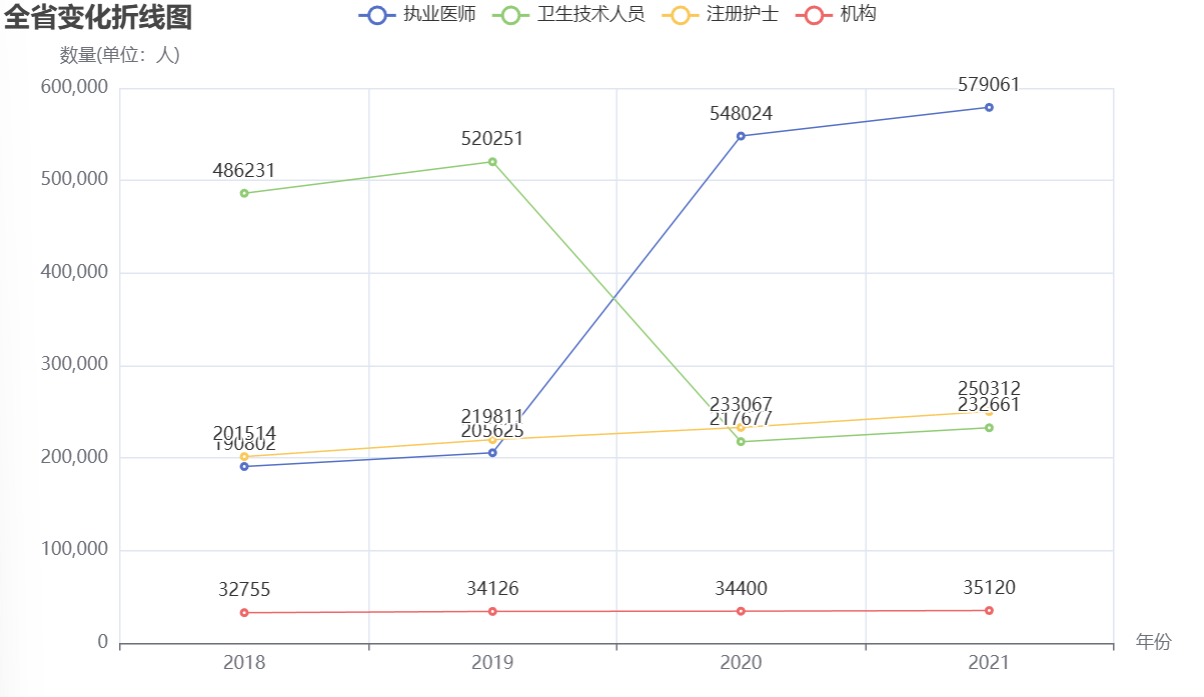




通过上图可以清晰看到浙江省11个市各自拥有的卫生资源总数量。我们可以看见在四个年份中，杭州市的医疗资源在所有类别中均占据显著的优势地位。特别是在执业医师数量和卫生技术人员。而其他两个拥有较为优势的医疗资源配置的地区为宁波市和温州市。结合浙江省内的经济状态，我们知道，经济水平高的区域一般拥有较多的医疗资源。明显地发现，即使是人口数量不少的丽水市，因为经济水平较差，导致其医疗资源的配备量也是少于其他更发达地区。

[全省总数相关图](Charts_Code/Graphics/全省总数相关.html)





以上两图是浙江全省医疗总数图。观察图片可以发现以下结论。

1. **执业医师数量的显著增长**

两张图都显示执业医师的数量从2018年到2021年呈现快速增长的趋势，特别是在2020年和2021年间增速显著。这表明在这几年中，对执业医师的需求大幅增加，可能是由于以下几个因素：

政策推动：政府可能出台了鼓励更多医学生进入执业医师行业的政策。

医疗需求增加：人口老龄化和慢性病增加导致对医疗服务需求的上升。

疫情影响：新冠疫情可能导致对医护人员，尤其是执业医师的紧急需求增加。

2. **卫生技术人员数量的波动**

卫生技术人员的数量在2019年达到峰值后显著下降并趋于稳定。这一现象可能有以下原因：

职业定义和统计口径变化：可能是卫生技术人员的定义或统计方式发生了变化。

**3. 注册护士数量的稳定增长**

两张图显示注册护士的数量在这几年中稳定增长。这个趋势表明注册护士在医疗体系中的需求稳步上升，可能是由于以下因素：

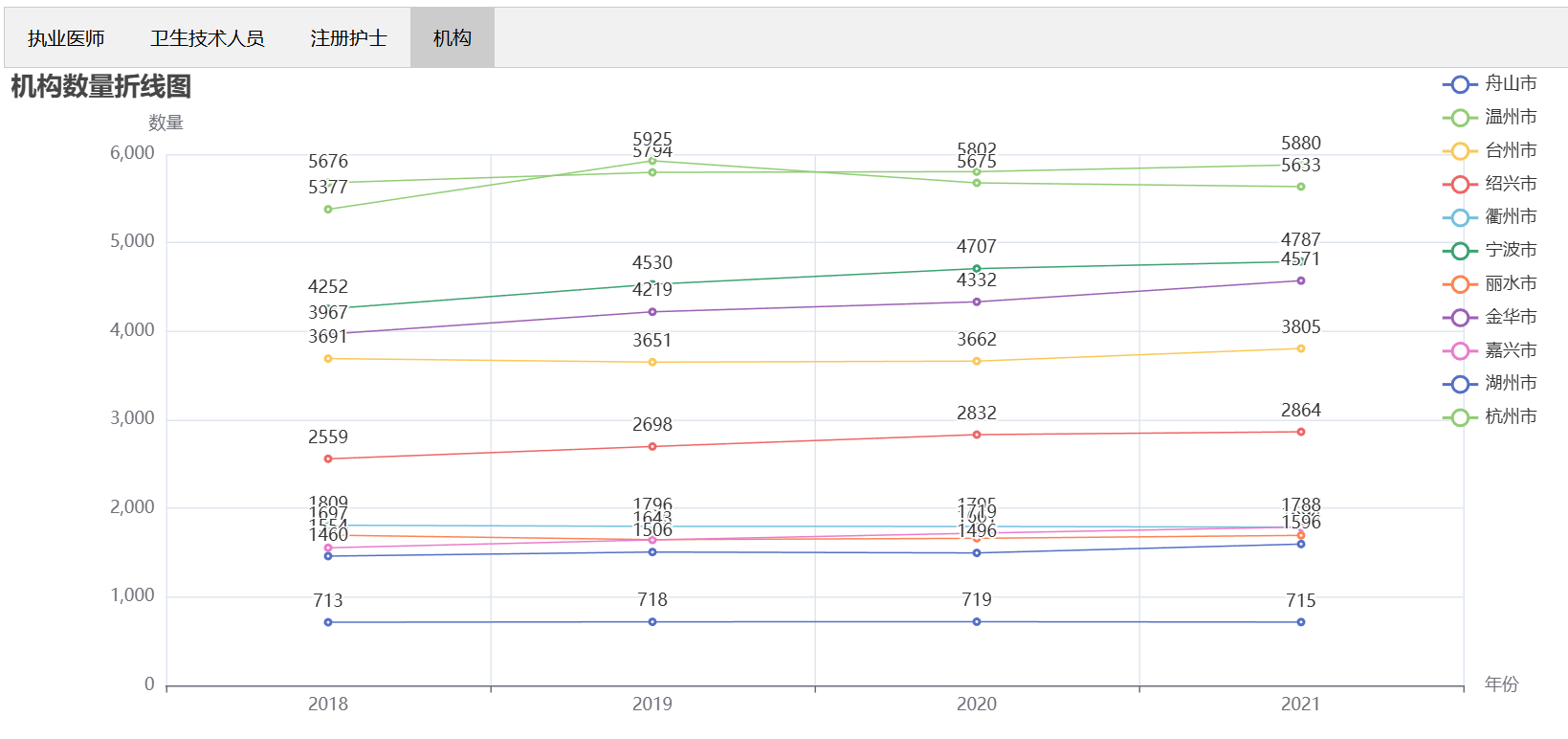
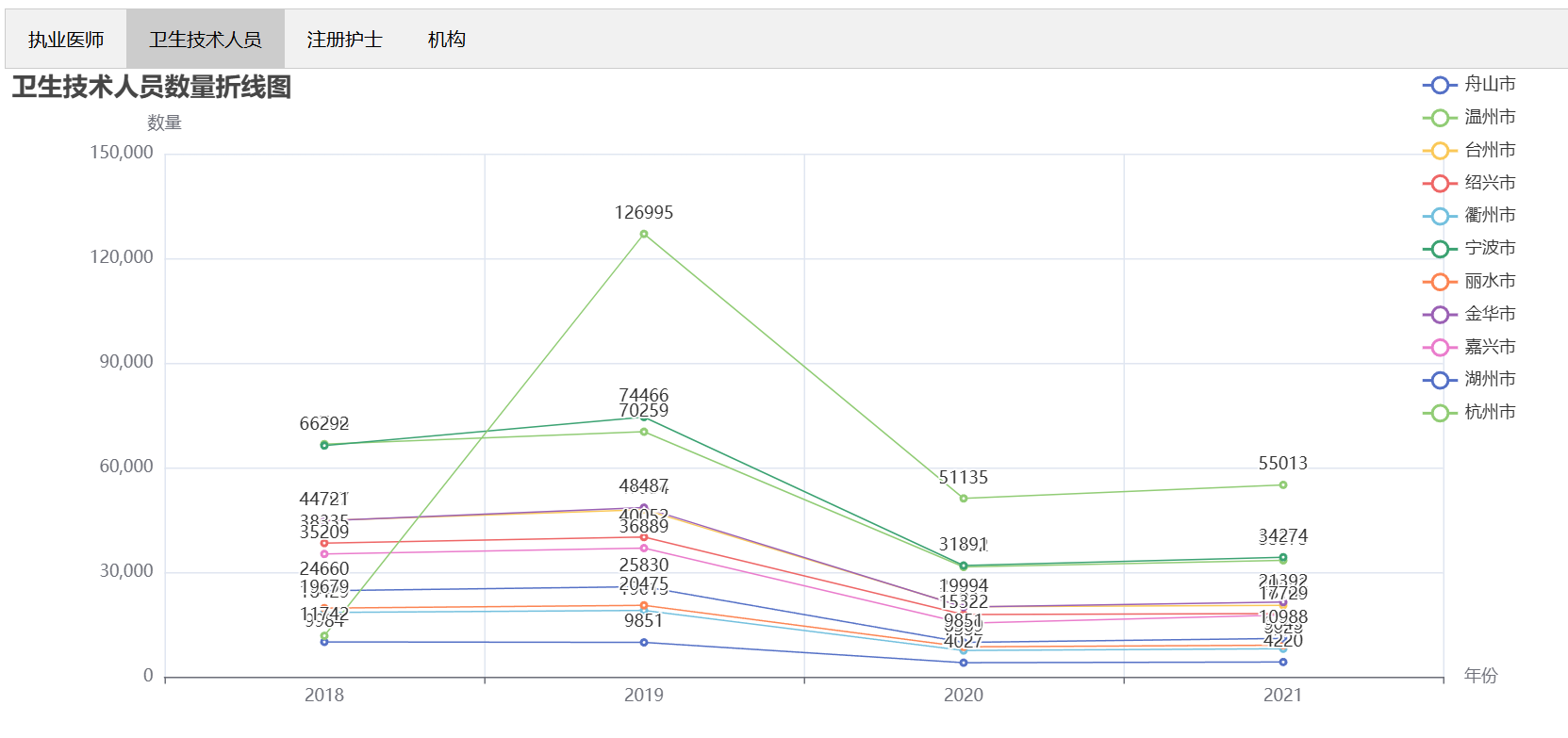
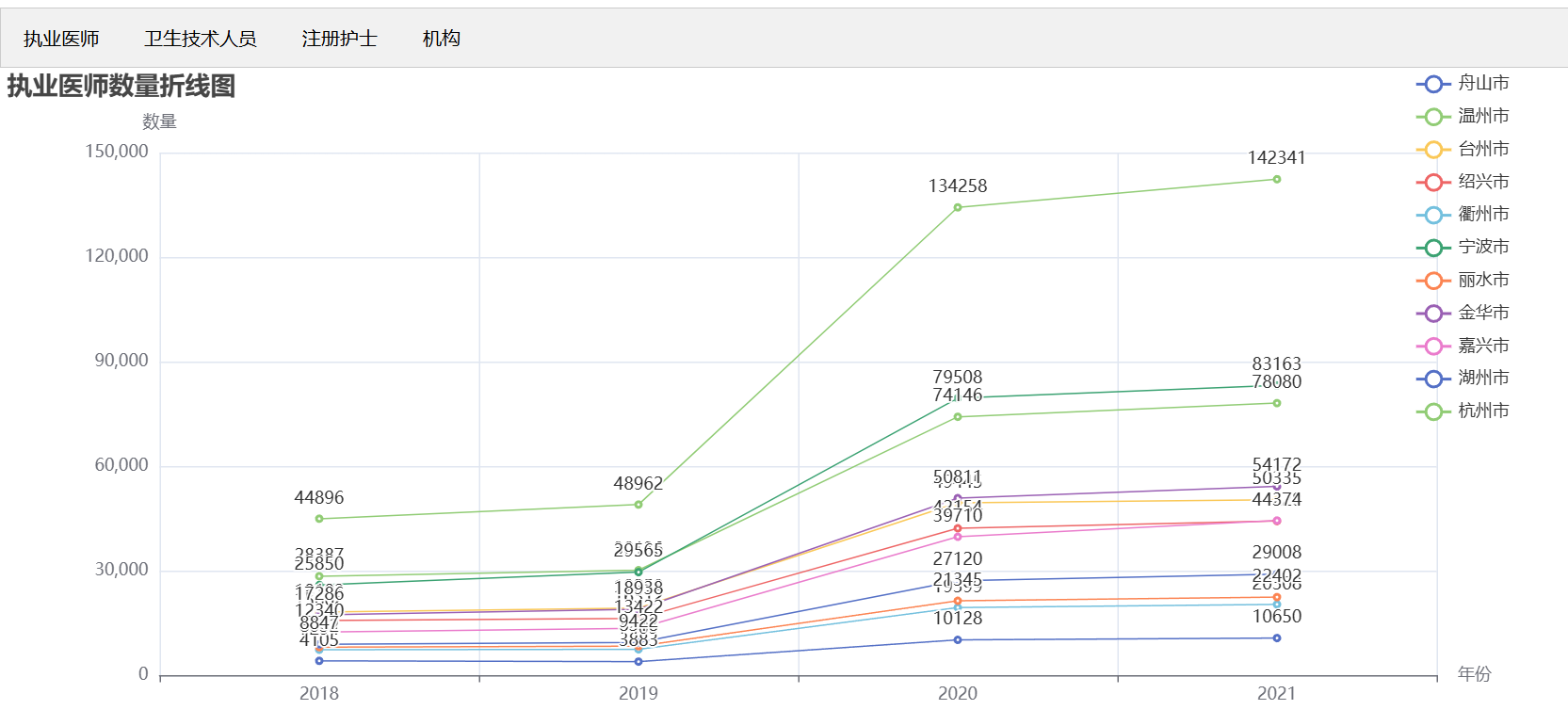
护理需求增加：随着人口老龄化，长期护理和慢性病管理的需求增加。

职业吸引力：护理职业的稳定性和就业机会可能使更多人选择进入这一领域。

**4. 医疗机构数量的相对稳定**

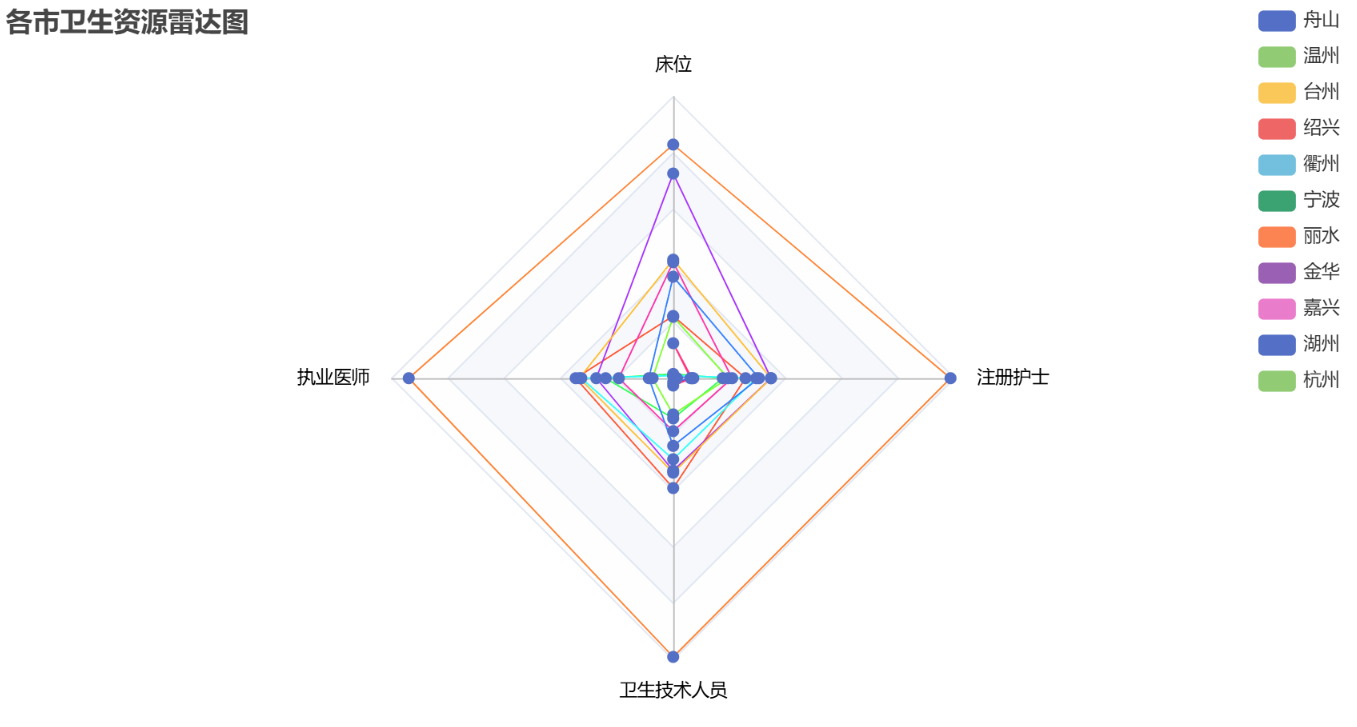
两张图都表明医疗机构的数量在2018年至2021年期间变化不大，维持在32,755到35,120之间。

3.3.2[各市医疗人员总数变化折线图](Charts_Code/Graphics/按职业标签的人数增长折线图.html)



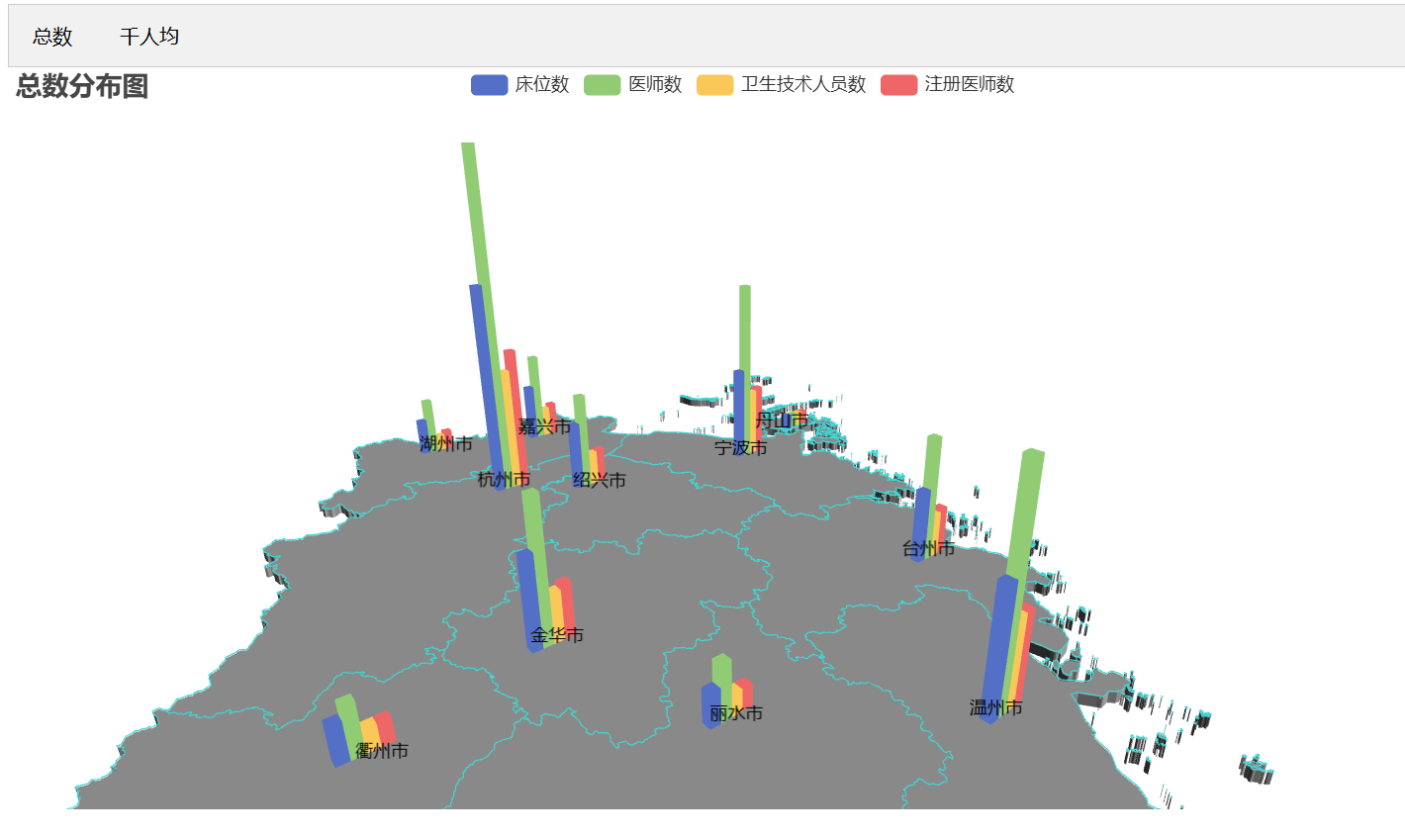
通过四个医疗资源在各市的年际变化折线图，我们可以发现，全省的整体趋势跟各市各资源的变化趋势整体上一致。都是执业医师、注册护士的数量在稳步增长，机构数量趋于稳定，而卫生技术人员数量在2019年到达顶峰后逐年下降。

3.3.3[各市卫生资源雷达图](Charts_Code/Graphics/各省卫生资源雷达图.html)

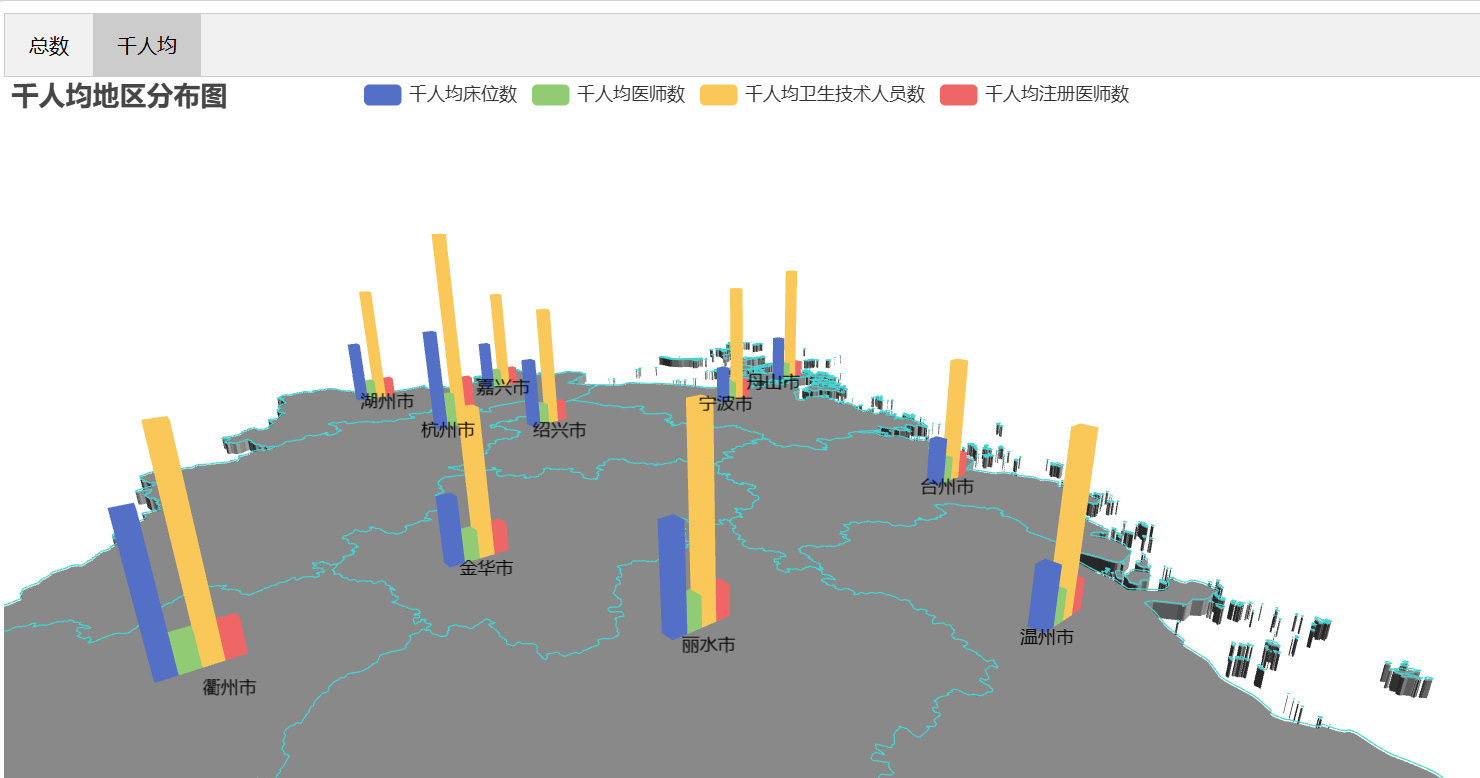


通过各市医疗资源雷达图，可见杭州市在各个方面都有较好的医疗资源配置。而作为对比，其他地区的卫生资源数量分配都出现了不平均的状况。

3.3.4[各市医疗人员总数3D条形图](Charts_Code/Graphics/3D分布柱状图.html)

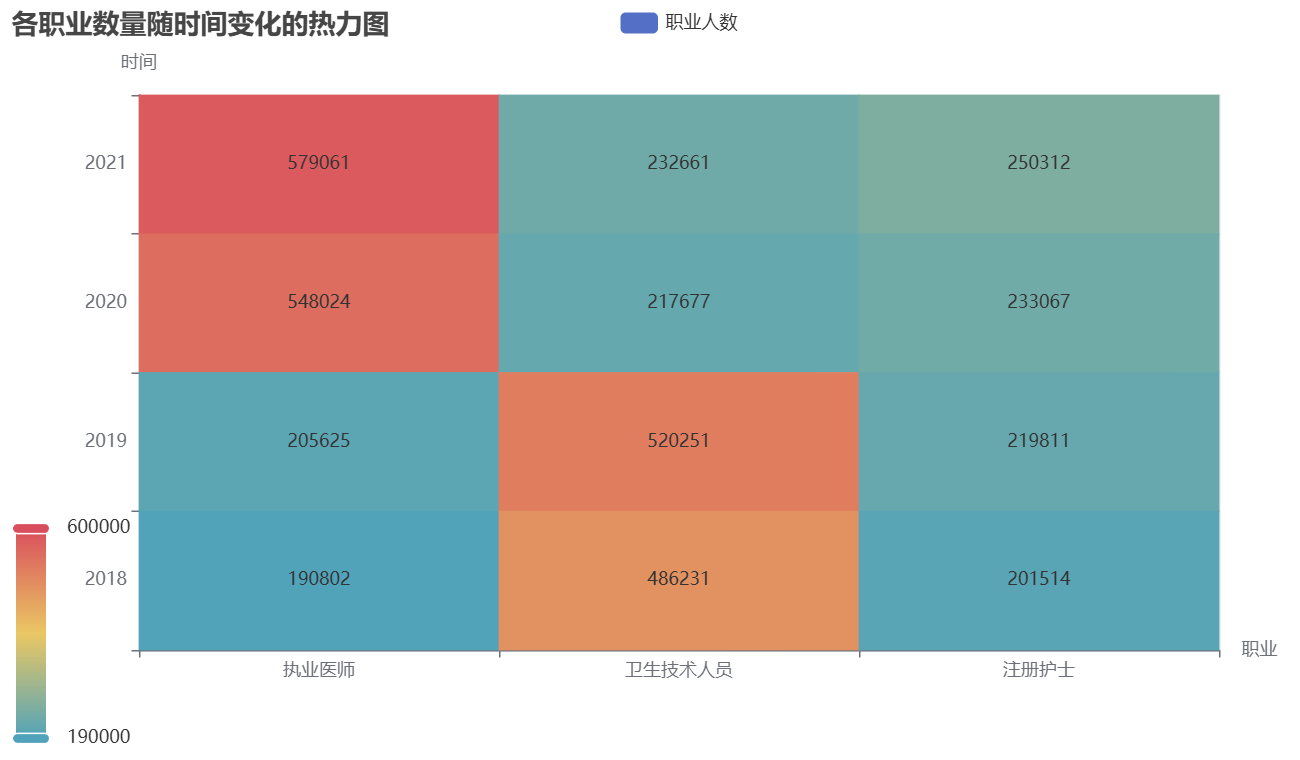


3.3.5[各市千人均医疗人员总数3D条形图](Charts_Code/Graphics/3D分布柱状图.html)



通过两张3D条形图，我们可以清晰地看到卫生资源在地理上的分布状态。

3.3.6[各市医疗人员总数变化热力图](Charts_Code/Graphics/全省各职业数量随时间变化热力图.html)

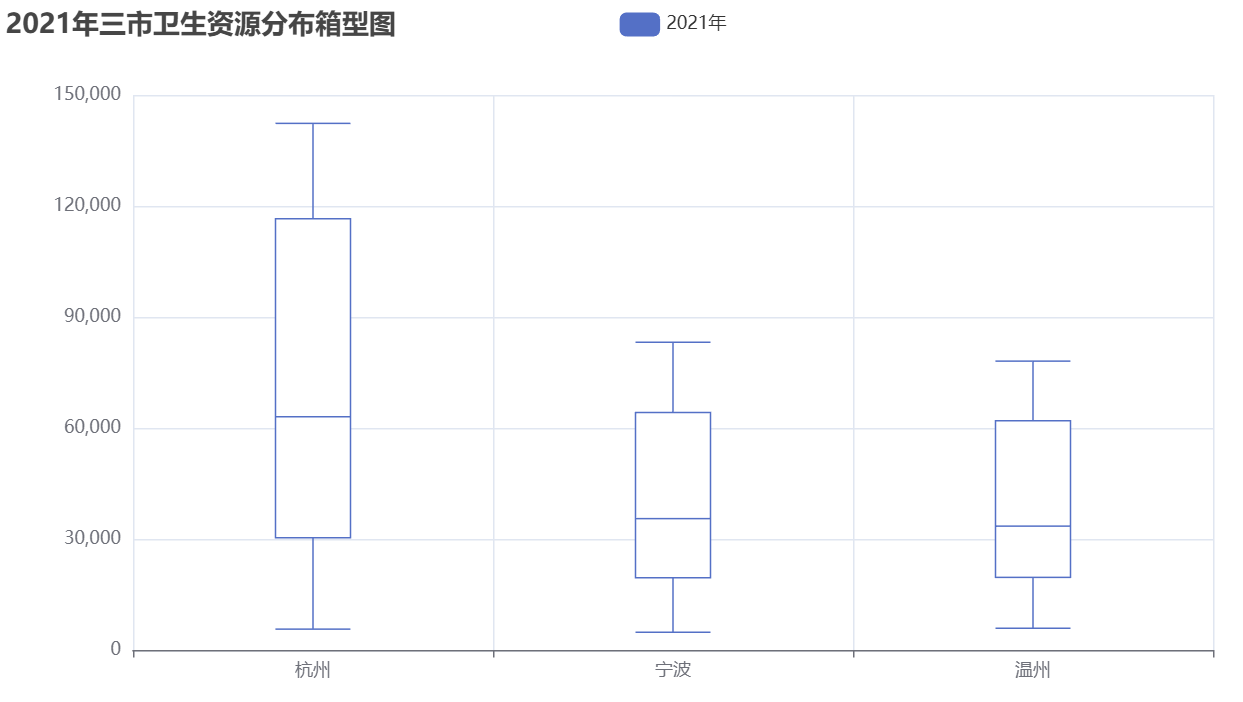


这张热力图展示了各职业数量随时间变化的情况。不同颜色和色块大小代表了不同职业在不同年份的数量变化。热力图的颜色越深表示数量越多，颜色越浅表示数量较少。

通过热力图可以得出以下主要结论：

1. 执业医师数量显著增长：从2018年至2021年，执业医师的数量快速增加，显示出对医师职业需求的显著上升。可能的驱动因素包括政府政策、人口老龄化和新冠疫情。
2. 卫生技术人员数量波动：虽然在2019年达到高峰，但之后数量明显下降。这一趋势需要进一步研究来确定原因，可能涉及政策调整、职业定义变化或市场需求变化。
3. 注册护士数量稳定增长：从2018年至2021年，注册护士的数量稳步增加，反映了护理需求的持续增长和职业吸引力的提升。
4. 医疗机构数量相对稳定：与医师和护士数量的变化相比，医疗机构的数量相对稳定，表明基础设施投资和资源配置较为平衡。

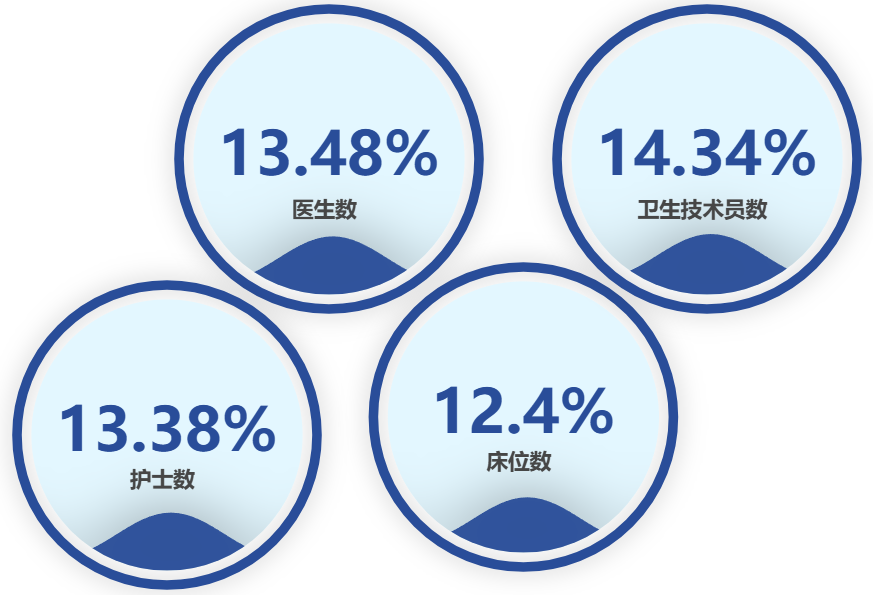
3.3.7[三市卫生资源分布箱型图](Charts_Code/Graphics/2021三市卫生资源分散箱型图.html)



箱型图内容：

1. 杭州市：卫生资源分布范围最大，从约25,000到近125,000。中位数大约在75,000左右，表明多数资源集中在这一水平。
2. 宁波市：卫生资源分布范围较广，从约15,000到约90,000。中位数大约在45,000左右，资源集中度较高。
3. 温州市：卫生资源分布范围最小，从约10,000到约65,000。中位数大约在35,000左右，资源分布相对均衡。
4. 从三市卫生资源分布的箱型图来看，杭州市拥有最多的卫生资源，资源分布范围也最大，而宁波市和温州市的资源相对较少但分布均衡，表明不同城市在资源配置上的差异和特点。

3.3.8[温州市在全省占比水滴图](Charts_Code/Graphics/温州各总数在全省占比.html)



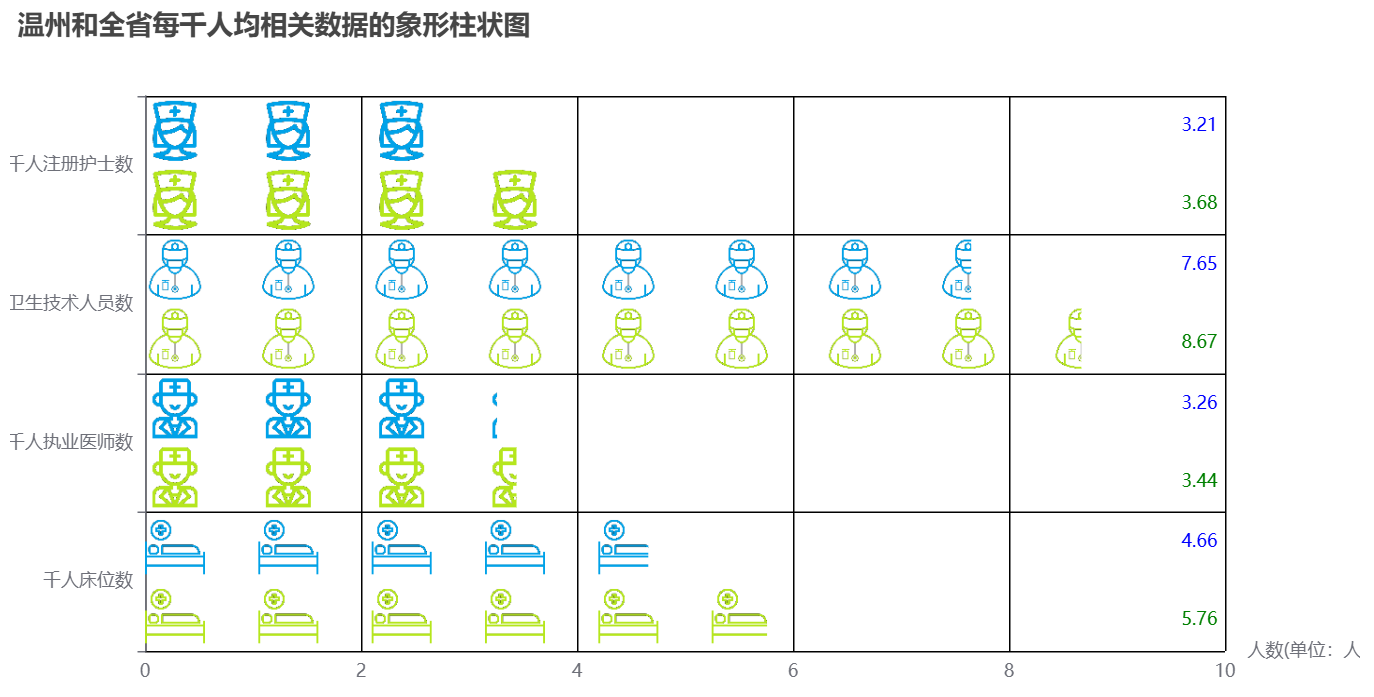
从水滴图中可以得出以下主要结论：

1. 医疗资源的相对丰富：温州市在全省的医生、卫生技术员和护士资源占比较高，分别为13.48%、14.34%和13.38%，表明该市在医疗人力资源方面的相对丰富。
2. 卫生技术员的优势：卫生技术员的占比最高，达到14.34%，这表明温州市在技术支持和辅助医疗服务方面具有优势。
3. 医疗基础设施的完善：虽然床位数占比相对较低，但仍有12.4%，显示出温州市在医疗基础设施方面的较好配置，能够支持其医疗服务的需求。

综合建议

1. 继续加强医生和护士的培养和引进：尽管目前占比较高，但随着医疗需求的不断增长，继续加强医生和护士的培养和引进是必要的。
2. 保持并提升卫生技术员的优势：利用现有的优势，进一步提升卫生技术员的技能和数量，可以提高整体医疗服务的质量。
3. 优化医疗基础设施：尽管床位数占比较低，但仍需持续优化和增加，以满足未来可能增加的住院和治疗需求。

3.3.9[温州市于全省千人均对比象形条形图](Charts_Code/Graphics/千人均.html)



通过象形柱状图的对比分析，可以得出以下主要结论：

1. 温州市在各项医疗资源上的千人占比均低于全省平均水平：无论是注册护士、卫生技术人员、执业医师还是床位数，温州市的千人占比均低于全省的平均水平。这提示了温州市在医疗资源配置方面存在提升空间。
2. 护理和卫生技术人员资源需重点关注：由于温州市在千人注册护士数和千人卫生技术人员数上均显著低于全省平均水平，建议在这些方面重点增加资源投入，提升相关人员的数量和质量。
3. 医疗基础设施建设需加强：千人床位数的明显不足表明温州市在医疗基础设施方面存在较大差距。建议增加医院床位数量，提升医疗服务的承载能力。

改进建议：

1. 加大医护人员的培养和引进：增加注册护士、卫生技术人员和执业医师的培养和引进力度，改善医护资源的短缺情况。
2. 优化医疗资源配置：通过合理的政策和规划，优化卫生技术人员和医疗设备的配置，提升整体医疗服务水平。
3. 加强基础设施建设：增加医疗床位数，特别是在大医院和综合医院，满足更多患者的住院需求，提高医疗服务的可及性和质量。

通过以上措施，温州市有望在医疗资源配置上取得显著提升，更好地满足市民的健康需求。

第四章 总结

在对浙江省各市医疗卫生资源分布数据进行深入研究和全面的可视化分析后，通过数据的展示与分析，我们不仅看到了当前各市医疗资源的具体情况，还发掘了未来资源优化和配置的潜在方向。

本研究的背景在于，随着社会的发展和人们生活水平的提高，医疗作为提升民生福祉的重要一环，变得愈发重要。合理的医疗资源配置对于提高医疗服务的可及性和公平性至关重要。通过优化医疗资源分配结构，可以显著提升医疗服务质量，确保各市之间医疗资源的公平分配，增强公共卫生的应对能力，尤其是在突发公共卫生事件中能迅速响应并弥补资源短缺。经历了新冠疫情等突发状况后，研究医疗资源的配置变得尤为重要。优化资源利用效率，最大化实现效益和资源的平衡，也能为政府和医疗管理机构提供科学依据，支持政策制定和实施。

本研究的数据主要来源于浙江省数据开放平台，包括2018至2021年间各市医疗人员数量、医疗机构数量、床位数等。我们对这些数据进行了清洗和预处理，处理了缺失值和异常值，确保数据的准确性和完整性。数据预处理的过程还包括数据归一化和数据集成，以便进行后续的分析和可视化。

通过使用Python编程语言和pyecharts库，我们制作了多种类型的图表，包括条形图、折线图、雷达图、热力图、箱型图和象形条形图等。这些图表直观地展示了浙江省各市的医疗资源分布和变化趋势。

通过对浙江省各市医疗资源的可视化分析，我们得出以下结论：

1. 医疗资源分布不均：尽管杭州市、宁波市和温州市等经济较发达地区拥有较多的医疗资源，但在全省范围内，资源分布仍不均衡，特别是一些偏远地区和经济欠发达地区，医疗资源相对匮乏。
2. 增长速度不一：不同市的医疗资源增长速度存在差异。杭州市等大城市的资源增长较快，而一些小城市和县区的增长相对缓慢，需要引起注意并采取相应措施。
3. 需加强资源调配：为了实现全省范围内医疗资源的均衡分布，需要进一步加强资源调配，特别是向资源匮乏地区倾斜，确保所有居民都能获得公平和高质量的医疗服务。
4. 政策支持与优化：本研究为政府和医疗管理机构提供了科学的决策依据，支持进一步优化医疗资源配置的政策制定。例如，可以根据数据分析结果，制定针对性的资源调配政策，提高资源利用效率，提升整体医疗服务水平。

基于本次研究的成果，未来可以进一步开展以下工作：

1. 深入数据挖掘：通过引入更多维度的数据，如人口密度、经济水平、疾病谱等，进行更深入的多变量分析，以获得更全面的医疗资源配置情况。
2. 动态监测与调整：建立动态监测系统，实时跟踪各市医疗资源的变化情况，根据需要进行资源调整，确保及时响应突发公共卫生事件。
3. 跨区域合作：促进区域间的医疗资源共享与合作，通过建立区域医疗联盟，优化资源配置，提高整体医疗服务水平。

浙江省各市医疗人员资源配置的可视化研究，不仅为政府和相关部门提供了科学决策依据，也为公众提供了透明、直观的医疗资源分布信息。通过这种方式，可以促进社会各界对医疗资源分配的关注和讨论，共同推动医疗资源的合理配置和优化，最终提高全省的公共卫生水平和居民的健康福祉。通过进一步的研究和优化措施，我们有望实现全省范围内更加均衡和高效的医疗资源配置，提升整体医疗服务质量和应对能力。

附录

数据可视化代码

1.浙江各省医疗人员总数

#按地区分的总数图

#x轴为床位数量、执业医师数量、卫生技术人员数量、注册护士数量、机构数量

#库

#options模块包含图表外观和行为的各种配置选项

from pyecharts import options as opts

#导入条形图和时间线图表

from pyecharts.charts import Bar,Timeline

#允许在python代码中插入JavaScript代码，对于自定义图表交互性功能（自定义提示框内容、设置图表动态行为）特别有用

from pyecharts.commons.utils import JsCode

#读取excel的库

import pandas as pd

#文件路径

data\_path='C:/Users/22273/Desktop/数据可视化/Data.xls'

#每年的行数

#1~45步长为4的序列,excel中索引从0开始，因此要减1

rows\_2021=list(range(1,42,4))

#前面插入一个0索引，保证把首行读进来

rows\_2021.insert(0,0)

rows\_2020=list(range(2,44,4))

rows\_2020.insert(0,0)

rows\_2019=list(range(3,45,4))

rows\_2019.insert(0,0)

rows\_2018=list(range(4,46,4))

rows\_2018.insert(0,0)

#lambda函数检查x是否在rows中，否则跳过

data\_2021=pd.read\_excel(data\_path,sheet\_name='Sheet2',skiprows=lambda x:x not in rows\_2021)

data\_2020=pd.read\_excel(data\_path,sheet\_name='Sheet2',skiprows=lambda x:x not in rows\_2020)

data\_2019=pd.read\_excel(data\_path,sheet\_name='Sheet2',skiprows=lambda x:x not in rows\_2019)

data\_2018=pd.read\_excel(data\_path,sheet\_name='Sheet2',skiprows=lambda x:x not in rows\_2018)

#创建时间线

timeline=Timeline(init\_opts=opts.InitOpts(page\_title="各市医疗人员总数图"))

#定义创建每年柱状图函数

def create\_bar(data,year):

    bar=Bar()

    bar.add\_xaxis(data.iloc[:,0].tolist())

    bar.add\_yaxis("执业医师",data.iloc[:,2].tolist())

    bar.add\_yaxis("卫生技术人员",data.iloc[:,3].tolist())

    bar.add\_yaxis("注册护士",data.iloc[:,4].tolist())

    bar.add\_yaxis("机构数量",data.iloc[:,5].tolist())

    bar.set\_global\_opts(

        title\_opts=opts.TitleOpts(title=f"{year}各地区医疗人员总数"),

        xaxis\_opts=opts.AxisOpts(name="地区"),

        yaxis\_opts=opts.AxisOpts(name="人员数量(单位：人)"),

    )

    return bar

#为每年柱状图添加时间线

timeline.add(create\_bar(data\_2018,2018),"2018")

timeline.add(create\_bar(data\_2019,2019),"2019")

timeline.add(create\_bar(data\_2020,2020),"2020")

timeline.add(create\_bar(data\_2021,2021),"2021")

#设置时间线全局选项

timeline.add\_schema(

    play\_interval=2000, #播放间隔

    is\_timeline\_show=True, #时间线显示

    is\_auto\_play=False, #自动播放

    is\_loop\_play=False, #循环播放

)

#渲染HTML文件

timeline.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/各市区医疗人员总数.html")

2.浙江各省医疗资源属性雷达图

import pandas as pd

from pyecharts import options as opts

from pyecharts.charts import Radar

from pyecharts.globals import ThemeType

# 文件路径

data\_path = 'C:/Users/22273/Desktop/数据可视化/Data.xls'

# 读取数据

data = pd.read\_excel(data\_path, sheet\_name="Sheet5", usecols='A,D,E,F,G,K')

# 数据分离

area\_list = data["地区"].tolist()

# 定义颜色列表

colors = [

"#FF5733", "#33FF57", "#3357FF", "#FF33A8", "#A833FF", "#33FFF7",

"#FFBF33", "#FF3380", "#80FF33", "#3380FF", "#FF8333", "#FF338A"

]

# 创建雷达图

radar = (

Radar()

#设置雷达图指标和样式

.add\_schema(

schema=[

opts.RadarIndicatorItem(name="床位", max\_=8,min\_=4.7),

opts.RadarIndicatorItem(name="执业医师", max\_=4.6,min\_=3.1),

opts.RadarIndicatorItem(name="卫生技术人员", max\_=11.7,min\_=7.5),

opts.RadarIndicatorItem(name="注册护士", max\_=5.2,min\_=3.1),

],

#设置雷达图分割区域

splitarea\_opt=opts.SplitAreaOpts(

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

),

textstyle\_opts=opts.TextStyleOpts(color="black"),

)

.set\_series\_opts(label\_opts=opts.LabelOpts(is\_show=False))

.set\_global\_opts(

title\_opts=opts.TitleOpts(title="各市卫生资源雷达图"),

legend\_opts=opts.LegendOpts(

orient="vertical",

pos\_right=True

)

)

)

# 循环插入数据并分配颜色

for i, area in enumerate(area\_list):

radar.add(

series\_name=area,

data=[data.iloc[i, 1:].tolist()],

linestyle\_opts=opts.LineStyleOpts(color=colors[i % len(colors)]),

label\_opts=opts.LabelOpts(is\_show=False),

)

# 渲染生成 HTML 文件

radar.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/各省卫生资源雷达图.html")

3.浙江各市各职业数量按年变化折线图

#各职业的增长折线图

#按分页拼接

#x轴为年份，y轴为数量，不同颜色为不同地区

#库

import pandas as pd

from pyecharts import options as opts

#导入线图、标签

from pyecharts.charts import Line,Tab

from pyecharts.globals import ThemeType

#文件路径

data\_path='C:/Users/22273/Desktop/数据可视化/Data.xls'

#创建标签

tab=Tab(page\_title="各职业年变化图")

#读取数据

data=pd.read\_excel(data\_path,sheet\_name="Sheet2",usecols="A:F",nrows=44)

#年份提取

years=sorted(data['数据统计时间'].unique())

#按职业循环创建

for career in data.columns[2:6].unique():

line=Line(init\_opts=opts.InitOpts(theme=ThemeType.LIGHT))

#str(year)将years中的每个元素转换为字符串类型

line.add\_xaxis([str(year) for year in years])

#以地区循环

for region in data['地区'].unique():

#布尔索引出data['地区']等于region的值

#.sort\_values('数据统计时间')按数据统计时间列升序排序

region\_data=data[data['地区']==region].sort\_values('数据统计时间')

#分离出职业为career的部分

y\_data=region\_data[career].tolist()

line.add\_yaxis(

series\_name=region,

y\_axis=y\_data,

)

#线图全局设置

line.set\_global\_opts(

#f-string语法：将career与字符串"数量折线图"连接

title\_opts=opts.TitleOpts(title=f"{career}数量折线图"),

xaxis\_opts=opts.AxisOpts(name="年份"),

yaxis\_opts=opts.AxisOpts(name="数量(单位：人)"),

#图例设置到右边

legend\_opts=opts.LegendOpts(

#图例位置右边5%

pos\_right="5%",

#图例排列方向垂直

orient="vertical",

),

)

#加入到标签

tab.add(line,career)

#渲染成html

tab.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/按职业标签的人数增长折线图.html")

4.浙江全省医疗人员热力图

import pandas as pd

from pyecharts.charts import HeatMap

from pyecharts import options as opts

professions = ['执业医师', '卫生技术人员', '注册护士']

time=["2018","2019","2020","2021"]

heatmap\_data = [

[0, 3, 579061],

[0, 2, 548024],

[0, 1, 205625],

[0, 0, 190802],

[1, 3, 232661],

[1, 2, 217677],

[1, 1, 520251],

[1, 0, 486231],

[2, 3, 250312],

[2, 2, 233067],

[2, 1, 219811],

[2, 0, 201514],

]

# 创建热力图

heatmap = (

HeatMap()

.add\_xaxis(professions)

.add\_yaxis("职业人数",time,heatmap\_data)

.set\_global\_opts(

title\_opts=opts.TitleOpts(title="各职业数量随时间变化的热力图"),

#颜色显示范围自定义

visualmap\_opts=opts.VisualMapOpts(

max\_=600000,

min\_=190000,

),

xaxis\_opts=opts.AxisOpts(name="职业"),

yaxis\_opts=opts.AxisOpts(name="时间")

)

)

heatmap.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/全省各职业数量随时间变化热力图.html")

5.浙江全省医疗资源总数相关条形图

#全省总数相关统计图

#各职业条形图，x轴为时间，y轴为数量，颜色为职业

#各职业数量变化折线，x轴为时间，y轴为数量,颜色为职业

#库

from pyecharts import options as opts

from pyecharts.charts import Bar,Line,Tab

import pandas as pd

#文件路径

data\_path='C:/Users/22273/Desktop/数据可视化/Data.xls'

#读取数据

data=pd.read\_excel(data\_path,sheet\_name="Sheet3",nrows=5)

#数据分离

#获取3到6列的列名，转换为列表

bar\_x\_data=list(data.columns[1:6])

years=sorted(data['数据统计时间'].tolist()) #注意升序排序

#创建一个year字典，每个键值为年份

#筛选出数据统计时间==year的行

#取出其中x\_data包含的列

#转换为numpy数组、展平为1维数组、转为列表

bar\_y\_data={year:data[data["数据统计时间"]==year][bar\_x\_data].values.flatten().tolist() for year in years}

#柱状图

bar=Bar()

bar.add\_xaxis(bar\_x\_data)

for year in years:

bar.add\_yaxis(str(year),bar\_y\_data[year])

#柱状图全局设置

bar.set\_global\_opts(

title\_opts=opts.TitleOpts(title="全省总数条形图"),

xaxis\_opts=opts.AxisOpts(name="职业"),

yaxis\_opts=opts.AxisOpts(name="数量(单位：人)")

)

#折线图绘制

# 创建数据

x\_data = ["2018","2019","2020","2021"]

y1 = [190802,205625, 548024,579061]

y2=[486231,520251,217677,232661]

y3=[201514,219811,233067,250312]

y4=[32755,34126,34400,35120]

# 创建折线图对象

line = Line()

#绘图

line.add\_xaxis(x\_data)

line.add\_yaxis("执业医师",y1)

line.add\_yaxis("卫生技术人员",y2)

line.add\_yaxis("注册护士",y3)

line.add\_yaxis("机构",y4)

# 设置全局配置项

line.set\_global\_opts(

title\_opts=opts.TitleOpts(title="全省变化折线图"),

yaxis\_opts=opts.AxisOpts(name="数量(单位：人)"),

xaxis\_opts=opts.AxisOpts(name="年份")

)

#标签

tab=Tab(page\_title="全省总数相关图")

tab.add(bar,"全省总数条形图")

tab.add(line,"全省变化折线图")

#输出

tab.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/全省总数相关.html")

6.浙江三市（杭州、宁波、温州）卫生资源分布箱型图

import pandas as pd;

from pyecharts.charts import Boxplot;

from pyecharts import options as opts;

data\_w=[45841,78080,33373,33488,5880]

data\_h=[90754,142341,55013,63044,5633]

data\_n=[45181,83163,34274,35490,4787]

data=[data\_h,data\_n,data\_w]

box=Boxplot()

box.add\_xaxis(["杭州","宁波","温州"])

#处理成箱型图可接受数据

box.add\_yaxis("2021年",box.prepare\_data(data))

box.set\_global\_opts(title\_opts=opts.TitleOpts(title="2021年三市卫生资源分布箱型图"))

box.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/2021三市卫生资源分散箱型图.html")

7.温州市每千人均象形柱状图

#温州市和全省千人均象形柱状图对比

#库

import pandas as pd

from pyecharts import options as opts

#象形柱状图

from pyecharts.charts import PictorialBar

# 数据

categories = ["千人床位数", "千人执业医师数", "千人卫生技术人员数", "千人注册护士数"]

#四年千人均平均数

data\_Wenzhou = [4.66, 3.26, 7.65, 3.21]

data\_Province = [5.76, 3.44, 8.67, 3.68]

# 象形条形图绘制

pictorialbar = PictorialBar(init\_opts=opts.InitOpts(page\_title="象形条形图"))

pictorialbar.add\_xaxis(categories)

pictorialbar.add\_yaxis(

"温州千人均",

[

{"value": 4.66, "symbol": "image://C:/Users/22273/Desktop/数据可视化/Icon/BedW.png"},

{"value": 3.26, "symbol": "image://C:/Users/22273/Desktop/数据可视化/Icon/DoctorW.png"},

{"value": 7.65, "symbol": "image://C:/Users/22273/Desktop/数据可视化/Icon/SurgeonW.png"},

{"value": 3.21, "symbol": "image://C:/Users/22273/Desktop/数据可视化/Icon/NurseW.png"},

],

#展示数值标签

label\_opts=opts.LabelOpts(is\_show=True,position="insideTopRight",color="blue"),

symbol\_size=40,

#图标偏移量：x移动0，y移动-23（向上）

symbol\_offset=[0,-23],

#设定为根据数值固定重复

symbol\_repeat="fixed",

#图标间距

symbol\_margin="45%",

#是否裁剪图标

is\_symbol\_clip=True,

)

pictorialbar.add\_yaxis(

"全省千人均",

[

{"value": 5.76, "symbol": "image://C:/Users/22273/Desktop/数据可视化/Icon/Bed.png"},

{"value": 3.44, "symbol": "image://C:/Users/22273/Desktop/数据可视化/Icon/Doctor.png"},

{"value": 8.67, "symbol": "image://C:/Users/22273/Desktop/数据可视化/Icon/Surgeon.png"},

{"value": 3.68, "symbol": "image://C:/Users/22273/Desktop/数据可视化/Icon/Nurse.png"},

],

label\_opts=opts.LabelOpts(is\_show=True,position="insideBottomRight",color="green"),

symbol\_size=40,

symbol\_offset=[0,23],

symbol\_margin="50%",

symbol\_repeat="fixed",

is\_symbol\_clip=True,

)

# 全局设置

pictorialbar.set\_global\_opts(

title\_opts=opts.TitleOpts(title="温州和全省每千人均相关数据的象形柱状图"),

xaxis\_opts=opts.AxisOpts(

name="人数(单位：人)",

is\_show=True,

#网格线设置

splitline\_opts=opts.SplitLineOpts(

is\_show=True,

#网格线条设置

linestyle\_opts=opts.LineStyleOpts(color="black"),

)

),

yaxis\_opts=opts.AxisOpts(

is\_show=True,

splitline\_opts=opts.SplitLineOpts(

is\_show=True,

linestyle\_opts=opts.LineStyleOpts(color="black")

),

),

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

)

# 反转轴使条形图横向显示

pictorialbar.reversal\_axis()

# 渲染图表到 Notebook

pictorialbar.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/千人均.html")

8.温州市卫生资源在全省占比水滴图

import pandas as pd

from pyecharts import options as opts

from pyecharts.charts import Liquid, Grid

from pyecharts.commons.utils import JsCode

# 数据

data = pd.read\_excel("C:/Users/22273/Desktop/数据可视化/Data.xls", sheet\_name="Sheet6")

# 数据分离

data\_bed = data.iloc[:, 1].tolist() # [:,1]选择所有行和索引为1列（第二列）

data\_doctor = data.iloc[:, 2].tolist()

data\_surgeon = data.iloc[:, 3].tolist()

data\_nurse = data.iloc[:, 4].tolist()

# 水滴图绘制

lbed = (

Liquid(init\_opts=opts.InitOpts(width="500px", height="500px"))

.add(

"床位数占比",

[round(data\_bed[0] / data\_bed[1], 4)],

center=["58%", "70%"],

label\_opts=opts.LabelOpts(

formatter=JsCode(

"""function(param){

return (Math.floor(param.value \* 10000) / 100) + '%';

}"""

),

position="inside",

),

)

.set\_global\_opts(title\_opts=opts.TitleOpts(

title="床位数",

pos\_top="75%",

pos\_left="54%",

))

)

ldoctor = (

Liquid(init\_opts=opts.InitOpts(width="500px", height="500px"))

.add(

"执业医生数占比",

[round(data\_doctor[0] / data\_doctor[1], 4)],

center=["40%", "27%"],

label\_opts=opts.LabelOpts(

formatter=JsCode(

"""function(param){

return (Math.floor(param.value \* 10000) / 100) + '%';

}"""

),

position="inside",

),

)

.set\_global\_opts(title\_opts=opts.TitleOpts(

title="医生数",

pos\_left="36%",

pos\_top="33%",

))

)

lsurgeon = (

Liquid(init\_opts=opts.InitOpts(width="500px", height="500px"))

.add(

"卫生技术员数占比",

[round(data\_surgeon[0] / data\_surgeon[1], 4)],

center=["75%", "27%"],

label\_opts=opts.LabelOpts(

formatter=JsCode(

"""function(param){

return (Math.floor(param.value \* 10000) / 100) + '%';

}"""

),

position="inside",

),

)

.set\_global\_opts(title\_opts=opts.TitleOpts(

title="卫生技术员数",

pos\_left="68%",

pos\_top="33%",

))

)

lnurse = (

Liquid(init\_opts=opts.InitOpts(width="300px", height="300px"))

.add(

"注册护士数占比",

[round(data\_nurse[0] / data\_nurse[1], 4)],

center=["25%", "73%"],

label\_opts=opts.LabelOpts(

formatter=JsCode(

"""function(param){

return (Math.floor(param.value \* 10000) / 100) + '%';

}"""

),

is\_show=True,

position="inside",

),

)

.set\_global\_opts(title\_opts=opts.TitleOpts(

title="护士数",

pos\_left="21%",

pos\_top="78%",

))

)

# 合并水滴图

grid = Grid(init\_opts=opts.InitOpts(page\_title="水滴图"))

grid.add(lbed, grid\_opts=opts.GridOpts(pos\_left="10%", pos\_top="10%"))

grid.add(ldoctor, grid\_opts=opts.GridOpts(pos\_left="55%", pos\_top="5%"))

grid.add(lsurgeon, grid\_opts=opts.GridOpts(pos\_left="5%", pos\_top="55%"))

grid.add(lnurse, grid\_opts=opts.GridOpts(pos\_left="55%", pos\_top="55%"))

# 渲染图表到 HTML 文件

grid.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/温州各总数在全省占比.html")

9.2021年最新千人均3D地图柱状图

#使用3D地图展示数值地理分布性

#堆叠柱状图显示医疗配备分布

import pandas as pd

from pyecharts import options as opts

from pyecharts.charts import Map3D,Tab

#该库是用于指定不同图表的枚举类，内含多种图表

from pyecharts.globals import ChartType

#数据框

df=pd.read\_excel('C:/Users/22273/Desktop/数据可视化/data.xls',sheet\_name='Sheet5')

#数据分离：提取row行中对应属性对应的元素，重组列表

data\_bed=[

(row['地区'],[row['经度'],row['维度'],row['千人床位数']])

#循环，定义两个变量，index接收行索引，row接收行数据

#使用pandas中函数iterrows遍历dataframe的每一行，返回行索引和包含含数据的series对象

for index, row in df.iterrows()

]

data\_doctor=[

(row['地区'],[row['经度'],row['维度'],row['千人执业医师数']])

for index, row in df.iterrows()

]

data\_surgeon=[

(row['地区'],[row['经度'],row['维度'],row['千人卫生技术人员数']])

for index, row in df.iterrows()

]

data\_nurse=[

(row['地区'],[row['经度'],row['维度'],row['千人注册护士数']])

for index, row in df.iterrows()

]

#3d地图绘制

map=Map3D()

#地图配置方案

map.add\_schema(

maptype="浙江",

#y轴的高度重定义，实现小数值显示较高的柱状条

box\_height=20,

#地图高度

region\_height=1,

#配置图元样式

itemstyle\_opts=opts.ItemStyleOpts(

color="white",

opacity=1,#图元透明度，1为完全不透明

border\_width=0.8,#图元边框粗细

border\_color="rgb(62,215,213)", #边框颜色

),

#3D光照

light\_opts=opts.Map3DLightOpts(

#主光源颜色：白

main\_color="#fff",

#光强

main\_intensity=1.2,

#阴影质量

main\_shadow\_quality="high",

#主光源阴影

is\_main\_shadow=False,

#主光源方位角

main\_beta=10,

#环境光强度

ambient\_intensity=0.3,

),

#标签：这里为地名

map3d\_label=opts.Map3DLabelOpts(

is\_show=True

)

)

#添加3d柱状图

#由于堆叠实现不理想，因此平铺实现，所以每个柱子间需要错开

#依次更新图标经度

t=0.05

data\_bed = [(city, [longitude-t\*2,latitude, value]) for city, (longitude, latitude, value) in data\_bed]

data\_doctor = [(city, [longitude-t, latitude, value]) for city, (longitude, latitude, value) in data\_doctor]

data\_nurse = [(city, [longitude+t, latitude, value]) for city, (longitude, latitude, value) in data\_nurse]

#添加到3d地图

map.add(

series\_name="千人均床位数",

data\_pair=data\_bed,

type\_=ChartType.BAR3D,

bar\_size=1,

)

map.add(

series\_name="千人均医师数",

data\_pair=data\_doctor,

type\_=ChartType.BAR3D,

bar\_size=1,

)

map.add(

series\_name="千人均卫生技术人员数",

data\_pair=data\_surgeon,

type\_=ChartType.BAR3D,

bar\_size=1,

)

map.add(

series\_name="千人均注册医师数",

data\_pair=data\_nurse,

type\_=ChartType.BAR3D,

bar\_size=1,

)

map.set\_global\_opts(

title\_opts=opts.TitleOpts(title="千人均地区分布图"),

)

#总数图

#数据

#数据分离

datas\_bed=[

(row['地区'],[row['经度'],row['维度'],row['床位']])

for index, row in df.iterrows()

]

datas\_doctor=[

(row['地区'],[row['经度'],row['维度'],row['执业医师']])

for index, row in df.iterrows()

]

datas\_surgeon=[

(row['地区'],[row['经度'],row['维度'],row['卫生技术人员']])

for index, row in df.iterrows()

]

datas\_nurse=[

(row['地区'],[row['经度'],row['维度'],row['注册护士']])

for index, row in df.iterrows()

]

#数据更新

t=0.05

datas\_bed = [(city, [longitude-t\*2,latitude, value]) for city, (longitude, latitude, value) in datas\_bed]

datas\_doctor = [(city, [longitude-t, latitude, value]) for city, (longitude, latitude, value) in datas\_doctor]

datas\_nurse = [(city, [longitude+t, latitude, value]) for city, (longitude, latitude, value) in datas\_nurse]

maps=Map3D()

#地图配置方案

maps.add\_schema(

maptype="浙江",

box\_height=40,

region\_height=1,

emphasis\_label\_opts=opts.EmphasisOpts(is\_disabled=1),

itemstyle\_opts=opts.ItemStyleOpts(

color="white",

opacity=1,

border\_width=0.8,

border\_color="rgb(62,215,213)",

),

light\_opts=opts.Map3DLightOpts(

main\_color="#fff",

main\_intensity=1.2,

main\_shadow\_quality="high",

is\_main\_shadow=False,

main\_beta=10,

ambient\_intensity=0.3,

),

map3d\_label=opts.Map3DLabelOpts(

is\_show=True,

)

)

#添加到3d地图

maps.add(

series\_name="床位数",

data\_pair=datas\_bed,

type\_=ChartType.BAR3D,

bar\_size=1,

)

maps.add(

series\_name="医师数",

data\_pair=datas\_doctor,

type\_=ChartType.BAR3D,

bar\_size=1,

)

maps.add(

series\_name="卫生技术人员数",

data\_pair=datas\_surgeon,

type\_=ChartType.BAR3D,

bar\_size=1,

)

maps.add(

series\_name="注册医师数",

data\_pair=datas\_nurse,

type\_=ChartType.BAR3D,

bar\_size=1,

)

maps.set\_global\_opts(

title\_opts=opts.TitleOpts(title="总数分布图"),

)

#添加入标签

tab=Tab(page\_title="3D分布图")

tab.add(maps,"总数")

tab.add(map,"千人均")

tab.render("C:/Users/22273/Desktop/数据可视化/Charts\_Code/Graphics/3D分布柱状图.html")