



西安邮电大学

毕业设计（论文）

题 目： 骨科测量数据可视化及分析系统的设计实现

学 院： 计算机学院

专 业： 网络工程

班 级： 网络 2101

学 号： 04212027

学生姓名： 席欢

导师姓名： 王亚刚 职称： 副教授

起止时间： 2024.11 至 2025.6

年 月

毕业设计（论文）承诺书

本人所提交的毕业设计（论文）《基于 Django 的骨形态参数分析及可视化模块的设计与实现》是本人在指导教师指导下独立研究、写作的成果，毕业设计（论文）中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人深知本承诺书的法律责任，违规后果由本人承担。

论文作者签名：_____ 日 期：_____

关于毕业设计（论文）使用授权的声明

本人在导师指导下所完成的论文及相关的职务作品，知识产权归属西安邮电大学。本人完全了解西安邮电大学有关保存、使用毕业设计（论文）的规定，同意学校保存或向国家有关部门或机构送交论文的纸质版和电子版，允许论文被查阅和借阅；本人授权西安邮电大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业设计（论文）。本人离校后发表、使用毕业设计（论文）或与该毕业设计（论文）直接相关的学术论文或成果时，第一署名单位仍然为西安邮电大学。

本毕业设计（论文）研究内容：

☒可以公开

☐不宜公开，已办理保密申请，解密后适用本授权书。

（请在以上选项内选择其中一项打“√”）

论文作者签名：_____ 导师签名：_____

日 期：_____ 日 期：_____

摘要

骨科疾病是全球范围内的常见健康问题，影响着人们的生活质量和工作能力。随着人口老龄化的加剧，骨科疾病的发病率逐年上升，尤其是骨折、关节炎、脊柱病等。传统的骨科测量方法往往受限于静态图像、依赖于手动记录和分析，无法提供足够的交互性和实时性。这不仅效率低下，还容易导致数据丢失和错误，限制了医学从业者对骨骼结构的深入学习和操作。此外，随着医疗技术的进步，产生的数据量日益增加，如何有效管理、分析和可视化这些数据成为亟待解决的问题。

为应对这一现状，本研究设计并开发了一套基于 Web 技术的骨科测量数据可视化及分析系统，整合 Django 后端框架、Vue.js 前端框架及 ECharts 数据可视化工具，实现了骨科数据的智能化管理与交互式分析。

系统采用三层架构设计：后端基于 Django REST Framework 构建高效数据接口，完成骨科测量数据（包括骨密度、骨形态参数、关节间隙等）的标准化存储、清洗及统计分析；前端通过 Vue.js 实现响应式用户界面，利用 ECharts 动态生成多维度可视化图表（如散点图、折线图、时间序列分析图等），支持医生进行数据探索与交互诊断，辅助医生快速评估骨质疏松、关节炎等疾病风险。通过测试，本文设计和实现的骨科测量数据可视化及分析系统能达到预期效果。

关键词：数据可视化；django 框架；axios 技术；echarts 模块；mysql 数据库

ABSTRACT

Orthopedic diseases are common health problems worldwide, affecting people's quality of life and work ability. With the intensification of population aging, the incidence of orthopedic diseases has been increasing year by year, especially fractures, arthritis, spinal diseases, etc. Traditional orthopedic measureData methods are often limited by static images and rely on manual recording and analysis, failing to provide sufficient interactivity and real-time performance. This is not only inefficient but also prone to data loss and errors, restricting medical practitioners' in-depth study and manipulation of bone structures. In addition, with the advancement of medical technology, the amount of generated data is increasing day by day. How to effectively manage, analyze, and visualize these data has become an urgent problem to be solved.

To address this situation, this study designed and developed a web-based orthopedic measureData data visualization and analysis system. By integrating the Django backend framework, Vue.js frontend framework, and ECharts data visualization tool, it achieved intelligent management and interactive analysis of orthopedic data.

The system adopts a three-tier architecture design. The backend, based on the Django REST Framework, constructs efficient data interfaces to complete the standardized storage, cleaning, and statistical analysis of orthopedic measureData data (including bone mineral density, bone shape parameters, joint space, etc.). The frontend, through vue.js, realizes a responsive user interface and uses ECharts to dynamically generate multi-dimensional visual charts (such as scatter plots, line charts, time-series analysis charts, etc.), enabling doctors to explore data and conduct interactive diagnosis, and assisting doctors in quickly assessing the risks of osteoporosis, arthritis, and other diseases. Through testing, the orthopedic measureData data visualization and analysis system designed and implemented in this paper can achieve the expected results.

Keywords: Data Visualization; Django Framework; Axios Technology; ECharts Module; MySQL Database.

目 录

第 1 章 绪论	1
1.1 研究背景和意义	1
1.2 国内外研究现状	2
1.3 研究目标与方法	3
第 2 章 相关原理与技术	5
2.1 Vue 框架概述	5
2.2 Echarts 框架概述	5
2.3 Django 技术概述	6
2.4 开发工具概述	7
第 3 章 需求分析	8
3.1. 业务需求分析	8
3.2 用户需求分析	8
3.3 功能需求分析	9
第 4 章 系统架构说明	10
4.1 总体架构说明	10
4.2 前端架构与模块设计	10
4.2.1 前端逻辑结构	11
4.2.2 前端模块设计	12
4.3 后端架构与模块设计	13
4.3.1 后端模块设计	13
4.3.2 数据库模块设计	14
第 5 章 系统实现	16
5.1 开发环境与工程化配置	16
5.1.1 前端开发环境	16
5.1.2 后端开发环境	16
5.2 前端系统实现	16
5.2.1 数据获取	17
5.2.2 数据可视化	18
5.2.3 数据分析	20
5.2.4 数据导入	21

5.3 后端系统实现	22
5.3.1 Django 配置说明	22
5.3.2 模型层开发	23
5.3.3 路由层开发	25
5.3.4 视图层开发	26
第 6 章 系统测试	28
6.1 测试环境	28
6.2 系统模块测试	28
6.2.1 数据可视化模块测试	28
6.2.2 测量结果可视化展示模块	31
6.2.3 数据分析模块	36
6.2.4 数据导入模块	38
结 论	41
参考文献	42
致 谢	44

第 1 章 绪论

1.1 研究背景和意义

随着全球人口老龄化进程加速，骨科疾病已成为影响人类健康的重要问题。世界卫生组织统计数据显示，骨质疏松症在全球范围内影响着超过 2 亿人口^[14]，而骨关节炎在 65 岁以上人群中的发病率高达 50%。传统的骨科诊断主要依赖 X 光、CT 等影像学检查结合医师经验判断，普遍存在三个方面的局限性：首先，目前大多数医疗机构仍采用纸质化记录或分散的电子存储方式，导致骨科测量数据难以实现统一管理和深度挖掘。这种碎片化的数据管理模式不仅增加了医护人员的工作负担，也阻碍了临床研究的开展；其次，静态数据展示在面对大数据和实时数据流时，往往显得力不从心，无法提供灵活的交互操作和动态更新功能；第三，不同测量参数之间的关联性分析缺乏有效工具，限制了临床决策的深度。

与此同时，现代 Web 技术的快速发展为解决这些问题提供了新的可能性^[5]。HTML5、Web 等技术的成熟使得在浏览器端实现高质量的图表渲染成为现实，前后端分离的架构模式为医疗信息系统的开发提供了更优化的解决方案。特别是近年来数据可视化技术的进步，为医疗大数据的分析和展示开辟了新的途径。

本研究旨在开发一套基于 Web 技术的骨科测量数据可视化与分析系统，通过创新性的数据呈现方式解决上述问题。该系统首先实现了便捷的数据录入功能，支持包括多种影像学测量数据的标准化采集。在数据呈现方面，系统采用动态交互式图表技术，将传统的静态报告升级为可缩放、可筛选的智能可视化界面。例如，脊柱侧凸患者随访过程中的 Cobb 角变化可以自动生成趋势曲线，并与临床症状评分进行关联分析。

本研究基于 Vue.js+ECharts+Django 技术栈开发骨科测量数据可视化及分析系统，具有重要的理论价值和实践意义：

该系统的创新性主要体现在三个方面：其一，基于云计算架构的 Web 应用模式，使医疗人员能够通过任何终端设备实时访问患者数据，显著提升了诊疗效率；其二，采用 ECharts 数据可视化库实现多维度、多层次的数据展示。系统支持从宏观趋势分析到微观细节探索的全方位数据可视化，包括但不限于：柱状图展示测量数据分布，折线图分析图标展示骨密度等数据变化趋势、散点图模拟测量数据分布、雷达图呈现异常数据分布等；其三，系统内置的分析模块可自动计算不同参数间的相关性，为临床研究提供数据支持，统计分析测量数据的平均值、方差、最大值最小值等；其四，技术架构层面也进行了一定的创新，前端利用 Vue.js 框架的响应式特性和组件化开发模式，构建高性能、可维护的前端应用程序。通过虚拟 DOM 技术优化渲染性能，确

保系统在处理大规模骨科数据时仍能保持流畅的用户体验^[5]。后端基于 Django 框架开发 RESTful API，采用 MTV（Model-Template-View）设计模式实现业务逻辑的解耦。

在数据管理方面，系统设计了完善的数据存储和安全机制。采用分布式数据库架构，能够高效处理日益增长的医疗数据量；通过权限管理确保患者隐私信息的安全性。

本研究的临床应用价值主要体现在三个方面：第一，通过实时数据可视化缩短了诊断决策时间；第二，通过多参数关联分析提升了诊疗方案的个性化程度；第三，通过标准化数据管理为多中心研究奠定了基础。随着精准医疗理念的普及，这种智能化的骨科数据分析系统将成为提升医疗质量的重要工具，其应用前景不仅限于临床诊疗，还可扩展到医学教育和科研创新等多个领域^[1]。

本研究开发的骨科测量数据可视化及分析系统，不仅解决了当前骨科诊疗中的关键技术问题，也为医疗信息系统的开发提供了可借鉴的技术方案。系统的推广应用将对提升骨科诊疗水平、优化医疗资源配置、推动医学研究产生积极影响。

1.2 国内外研究现状

骨科测量数据可视化及分析系统的研究在国内外呈现出不同的发展轨迹与技术特征。国内研究近年来取得显著进展，上海交通大学医学院附属瑞金医院构建的骨科临床大数据平台已整合超过 10 万例患者的影像、病历及随访数据，基于 Python Django 框架开发的后端接口实现了基础数据统计与可视化功能，如手术量趋势分析与患者分布图谱展示。^[17]，支持不同治疗方案的疗效对比分析，但其三维交互功能仍依赖第三方库（如 Three.js）实现。中科院深圳先进研究院研发的 Med-X 平台提供了骨科影像分割与三维重建功能，支持自定义测量工具（如椎体旋转度测量），但其可视化模块依赖 VTK 库，在移动端的交互性能与跨平台兼容性方面存在不足。

在国际范围内，欧美发达国家凭借医疗信息化的先发优势与技术积累，在该领域展现出较强的竞争力。加拿大 OsteoVis 系统利用 ECharts 技术开发了关节活动度（ROM）动态监测模块，通过时序图表与 3D 动画相结合的方式，直观展示患者康复训练中的角度变化趋势，但其数据仅局限于运动学参数，未整合生物力学与影像学数据。英国 Sheffield 大学团队开发的 FRAX 模型（2024 年更新版）整合了骨密度、年龄、BMI 等多维度数据^[15]，最新版本纳入 DXA 影像纹理特征后，骨折风险预测 AUC 值提升至 0.89，并获得 FDA 认证，但其可视化界面仍以简单的风险评分图表为主，缺乏对预测模型的可解释性展示。

国内外研究在技术成熟度、数据整合能力与临床转化方面存在显著差异。国际上三维重建、动态可视化技术已趋成熟，商业软件（如德国 Brainlab Curve 系统）采用边缘计算架构，可在 20 秒内完成全脊柱模型重建与测量，但其系统封闭且价格高昂，限制了在发展中国家的普及应用。部分国际系统虽实现了多源数据整合，但临床数据

与科研数据未完全打通，导致数据价值未能充分挖掘。相比之下，国内研究开源框架与商业系统并存，但核心算法（如高精度分割）仍依赖国外技术，例如中科院 Med-X 平台依赖 VTK 库进行三维重建，关键分割算法（如 U-Net++）的 Dice 系数平均为 0.82，较国际先进水平低约 10%。医院级大数据平台建设起步较晚，数据标准化程度低，跨机构数据共享率不足 15%，主要受制于《数据安全法》和伦理审批流程。人工智能辅助诊断研究虽活跃，但多停留在算法验证阶段。此外，国内通用型平台较多，专科特色功能（如脊柱三维矢状面平衡分析）开发不足，北京协和医院开发的 ScolioScore 系统虽能基于深度学习测量 Cobb 角（与金标准的皮尔逊相关系数 $r=0.96$ ），但缺乏对矢状面平衡参数的三维可视化分析能力，难以满足复杂脊柱畸形诊疗需求。

前沿技术正在为骨科测量数据可视化及分析系统带来新的发展机遇。中国医学科学院正在构建骨科联邦学习网络，已完成 3 家中心的初步测试，模型性能损失控制在 8% 以内。这些技术的发展与应用，有望推动骨科测量数据可视化及分析系统向更加智能化、精准化和普惠化的方向发展。

1.3 研究目标与方法

该研究旨在开发一个基于 Vue+ECharts+Django 技术栈的骨科测量数据可视化及分析系统，通过整合多源医疗数据、实现多维可视化展示与智能分析，为骨科临床决策与医学科研提供高效工具。系统的研究目标可分解为数据录入、数据处理、可视化展示、结果分析四个维度。在数据处理方面解决单一的录入患者的相关临床数据，在针对临床数据录入效率问题，系统将开发批量数据导入功能，支持 CSV 等格式文件的快速解析，同时设计数据校验机制，确保数据质量。具体研究方向概括为以下几点：

可视化展示层面，系统将构建多层次的展示体系。基于 ECharts 开发参数可视化组件，实现了髌骨、胫骨、股骨、尺骨四种骨骼数据的高效展示，以胫骨为例，胫骨中需要测量的数据主要有 Tibial Angle（胫骨角）、Tibial Internal Inversion Angle（胫骨内翻角）、Superior Tibial Angle（胫骨上角）、Tibial Plateau Angle（胫骨平台角）、Posterior Tibial Slope（胫骨后倾角），因此我们的胫骨数据模块可以通过展示以上五种参数的各个图形展示，包括但不限于柱状图、折线图、散点图以及雷达图、箱线图^[12]。

智能分析功能是系统的核心优势之一。系统将计算数据关键指标的统计学参数（均值、标准差、极值等），生成标准化分析结果；开发患者个体数据与群体标准值的对比分析模块，通过雷达图等可视化方式直观展示差异。

在研究方法层面，系统采用多层架构设计实现松耦合、高内聚的系统特性。后端基于 Django REST Framework 构建的 API 服务遵循 RESTful 规范^[7]，采用视图集

（ViewSet）与序列化器（Serializer）分离的模式，实现数据访问与展示逻辑的解耦。数据库设计采用第三范式（3NF）减少数据冗余，并通过索引优化提升复杂查询性能。

前端基于 Vue.js 的组件化设计遵循单一职责原则，将 ECharts 封装为可复用的图表组件，通过 Vuex 实现全局状态管理，确保多图表联动时的数据一致性。交互层开发自定义指令（v-drag、v-filter）简化 DOM 操作，利用 Vue Router 实现路由懒加载，提升系统响应速度。系统还集成了权限控制模块，基于角色的访问控制（RBAC）确保不同用户（医生、研究员、管理员）只能访问授权数据。

系统集成层面上，前后端通信采用 JWT 认证机制，通过 axios 实现 API 请求的拦截与重试，设计基于事件总线的实时通知系统，定义清晰的 API 版本控制与向后兼容策略。

第 2 章 相关原理与技术

2.1 Vue 框架概述

Vue.js 是一款渐进式 JavaScript 前端框架，由尤雨溪于 2014 年首次发布。作为当前最流行的前端框架之一，Vue 以其轻量级、高性能和易用性著称，特别适合构建现代化的单页面应用（SPA）和复杂的前端交互界面。

最具特色的设计是其响应式系统。该框架采用 MVVM（Model-View-ViewModel）架构模式，通过 ES5 的 `Object.defineProperty`（Vue 2.x）^[3]或 ES6 的 `Proxy`（Vue 3.x）来实现数据劫持，自动追踪依赖关系。当数据发生变化时，系统能够智能地确定需要更新的组件，并以最优的方式更新 DOM。这种机制不仅大大简化了开发者的工作，还确保了应用的性能表现。Vue 3.0 引入的 Composition API 更是将响应式能力提升到了新的高度，开发者可以更灵活地组织和复用业务逻辑。

Vue 的核心优势在于其组件化开发模式。开发者将模板、脚本和样式封装在一个 `vue` 文件中，提高代码的可维护性和复用性。框架提供了 `props/events` 机制实现父子组件通信，支持插槽（slot）实现内容分发，以及动态组件等高级功能。这种组件化设计使得大型应用的开发和维护变得更加高效和系统化。

在性能优化方面，Vue 采用了虚拟 DOM 技术。当状态变化时，Vue 会先生成一个虚拟 DOM 树，通过高效的差异化（diff）算法计算出最小更新范围，然后批量更新真实 DOM。这种机制显著提升了渲染性能，特别是在处理复杂视图和频繁数据更新时表现尤为突出。此外，Vue 3.0 引入的 Composition API 进一步提升了代码组织和逻辑复用的灵活性，使开发者能够更好地管理复杂组件的状态和逻辑。

围绕 Vue.js 形成的生态系统丰富且成熟，官方提供了 Vue Router 用于单页面应用的路由管理、Vuex 解决中大型应用的状态管理难题、Vue CLI 帮助快速搭建项目框架。同时，还有 Element UI、Vuetify 等第三方组件库，以及 Nuxt.js 等服务端渲染框架，进一步拓展了 Vue.js 的应用场景。目前 Vue.js 主要存在 2.x 与 3.x 两个版本，Vue 2.x 兼容性良好，支持 IE9 及以上浏览器；Vue 3.x 则在性能上实现提升，增强了对 TypeScript 的支持，并引入 Composition API，优化了大型应用的代码组织方式。

2.2 Echarts 框架概述

ECharts 是由百度团队开发并捐赠给 Apache 基金会的一款开源 JavaScript 数据可

视化库，现已成为国内最主流的前端可视化解决方案之一。作为一款企业级图表工具，ECharts 提供了超过 30 种基础图表类型和 10 余种专业领域图表，涵盖从基础的折线图、柱状图到复杂的三维地球、关系网络图等多种可视化需求^[9]。其核心优势在于强大的数据处理能力和丰富的交互特性，能够流畅支持百万级数据量的渲染，并提供数据筛选、区域缩放、图表联动等专业交互功能。

在技术实现上，ECharts 采用分层架构设计，底层基于自主开发的 ZRender 图形渲染引擎，支持 Canvas 和 SVG 双渲染模式；中层实现图表核心逻辑和交互系统；上层提供友好的配置接口和扩展机制。这种架构使得 ECharts 既保证了渲染性能，又保持了足够的灵活性。特别值得一提的是其模块化设计，开发者可以按需引入所需图表类型，有效控制代码体积。

从应用场景来看，ECharts 在商业智能、科研分析、医疗健康等领域都有出色表现。在医疗领域，ECharts 特别适合用于展示骨科测量数据中的多维指标分析、时空变化趋势和三维结构可视化。其提供的热力图可以直观展示骨密度分布，3D 散点图能清晰呈现骨骼结构特征，而动态折线图则非常适合观察骨折愈合过程中的各项指标变化趋势。这些特性使其成为开发骨科数据可视化系统的理想选择。

ECharts 的生态系统也十分完善。官方提供了 React、Vue 等主流框架的封装组件，以及 Python 等后端语言的接口库。ECharts 采用了智能渲染、渐进式加载等策略，确保在大数据量场景下仍能保持流畅交互。总的来说，ECharts 凭借其全面的图表能力、专业的交互设计和稳定的性能表现，完全能够满足骨科测量数据可视化系统的开发需求，是医疗数据可视化领域的优质技术选型。

2.3 Django 技术概述

Django 是一个基于 Python 的高级 Web 框架，遵循 MVC（Model-View-Controller）架构模式，实际采用 MTV（Model-Template-View）模式实现，强调快速开发与 DRY（Don't Repeat Yourself）原则。它由经验丰富的开发者设计，通过提供全面的组件（如 ORM、路由系统、模板引擎、表单处理等），帮助开发者高效处理常见 Web 开发任务，如用户认证、内容管理、表单验证、缓存等。Django 的核心优势包括内置的安全机制（防范 SQL 注入、跨站脚本攻击、跨站请求伪造等）、强大的数据库抽象层（ORM 支持多种数据库后端）、自动生成的管理界面（Admin）以及灵活的模板系统。其生态系统丰富，拥有大量第三方插件（如 DRF 用于构建 RESTful API、Celery 处理异步任务）和社区支持，适用于从简单网站到大型企业级应用的各种项目。Django 的设计哲学使其特别适合需要快速迭代开发、重视安全性与可维护性的项目，是 Python Web 开发领域的主流框架之一。

在技术架构层面，Django 采用松耦合的模块化设计^[8]，主要包含模型层、视图层

和模板层三大核心组件。模型层通过面向对象的方式定义数据结构，其强大的 ORM 系统支持多种数据库后端，能自动生成 SQL 语句并处理数据迁移；视图层负责业务逻辑处理，支持函数视图和类视图两种范式；模板层则提供灵活的页面渲染机制，支持模板继承和标签扩展。特别值得注意的是 Django 的安全机制，框架默认提供 CSRF 防护、XSS 过滤、SQL 注入防御等多重安全保护，这在处理敏感的医疗数据时尤为重要。

2.4 开发工具概述

Visual Studio Code（简称 VSCode）是一款由微软开发的轻量级、高性能开源代码编辑器，支持跨平台运行（Windows、macOS、Linux）。VS Code 的设计初衷是既轻量又强大，启动速度快、内存占用低，可以开发各种现代需求。同时支持多种编程语言（如 JavaScript、Python、Java、C++ 等），通过安装扩展可进一步增强对特定语言和框架的支持。VSCode 的核心优势在于其强大的扩展性，拥有庞大的插件生态系统，开发者可根据需求安装版本控制工具（如 Git）、代码格式化工具（如 Prettier）、自动补全插件、主题皮肤等，极大提升开发效率。其内置的终端、集成调试器和 Git 支持使开发者能够在单一环境中完成编码、测试和版本管理等全流程操作。此外，VSCode 支持自定义快捷键、主题和工作区设置，满足个性化开发需求，是目前前端开发、后端开发、数据科学等领域广泛使用的主流开发工具之一。

第3章 需求分析

3.1. 业务需求分析

本系统旨在为骨科测量数据提供专业的数据可视化分析工具，重点支持髌骨、胫骨、股骨、尺骨四种骨骼的测量数据管理与分析。业务需求分析可以从以下几个方面展开：

（1）髌骨模型：首先是髌骨模型的相关测量数据的可视化展示，该骨模型有五个基本参数分别为 tt_tg （胫骨结节到股骨滑车沟的距离）、 $tkLength$ （通髌线长度）、 tt_te （胫骨结节到股的距离）、 tt_tg / tt_te （胫骨结节到股骨滑车沟的距离与胫骨结节到股的距离两者相除）、 $tt_tg / tkLength$ （胫骨结节到股骨滑车沟的距离、通髌线长度两者相除）。需要对五个基本参数的数据以多个图表的方式对其进行可视化展示，包括但不限于柱状图、折线图、雷达图以及散点图，同时在绘制图表后，我们可以对相应的数据进行结果分析

（2）胫骨模型：胫骨中需要测量的数据主要有胫骨角、胫骨内翻角、胫骨上角、胫骨平台角。

（3）股骨模型：股骨模型中需要测量的参数主要有 $FemoralNeckShaftAngle$ 股骨颈干角（股骨颈与股骨干的夹角）、 $FemoralVersionAngle$ 股骨前倾角（股骨颈相对于股骨髁的旋转角度）、 $FemoralTorsionAngle$ 股骨扭转角（股骨近端与远端的扭转角度）、 $FemoralHeadOffset$ 股骨头偏移量（股骨头中心与股骨干轴线的距离）

（4）尺骨模型：尺骨模型需要测量的数据主要有 $UlnarLength$ ：尺骨长度（鹰嘴到尺骨茎突的距离）、 $UlnarTiltAngle$ ：尺骨倾斜角（尺骨远端关节面与水平面的夹角）两个基本参数，同时对该数据生成的可视化展示模块进行分析

3.2 用户需求分析

骨科测量数据的可视化及分析系统的设计与实现，用户需求可以分为以下几个方面阐述

用户希望能够通过项目实现对各个骨骼模型相应形态参数绘制多个图表，包括折线图、柱状图等图形绘制。

用户希望能够对测量到的骨骼模型的数据进行参数统计分析以及统计详情体现

（3）用户对项目的易用性和交互体验有明确需求，期望系统界面简洁直观、操作流程便捷顺畅，确保用户能快速上手并高效完成操作，同时通过优化交互逻辑与视觉反馈，提升使用过程中的舒适性与流畅度。

（4）用户对项目的性能有一定要求，希望可以快速加载可视化图表，在展示时流畅

自然，且操作响应实时高效，以此保障用户体验的流畅性与使用效率。

3.3 功能需求分析

针对骨科测量数据可视化及分析系统的设计与实现，功能需求可以分为以下几个方面阐述：

（1）数据可视化展示：用户希望能够进行对骨模型相关数据做可视化展示，包括但不限于骨骼长度及密度。系统支持从宏观趋势分析到微观细节探索的全方位数据可视化，包括但不限于：对各个骨模型测量的数据以多种方式进行呈现，如柱状图、折线图、盒需图、散点图以及雷达图等。通过柱状图可以反映不同人的 `tkLength` 通髌线长度的分布，折线图可以反应变化趋势，而雷达图极好的进行了多维度的分析对比。

（2）数据分析：测量数据的统计分析系统通过三个关键环节实现全面的数据评估。在基础统计量计算环节，系统自动对 `tt_tg` 等五个核心参数进行多维度量化分析，包括计算算术平均值、方差、标准差、最大值、最小值及极差等六大核心指标，所有结果保留两位小数以确保临床精度。统计结果通过三种形式实现可视化集成：在图表中使用 `markLine` 标注平均值参考线，`markArea` 显示均值±标准差的范围区间；独立统计面板以表格形式展示完整统计量；交互提示功能则在悬停时显示当前值相对于均值的偏离程度。分析结果最终通过双重方式呈现：在可视化方面，图表通过颜色区分正常与异常结果，并标注均值±1 标准差范围（如 11.88-15.38mm）及临床参考线；在数据统计方面，系统输出包括超过/低于平均值的组数、连续上升/下降最大次数，以及按临床标准分类的正常、临界、异常病例数等关键指标，为临床决策提供量化依据。

（3）数据导入：支持以 Excel 等常见文件格式进行多量级数据批量导入。用户上传文件后，系统自动解析文件内容，将数据映射至对应的数据字段，存储在数据库中。

（4）采用医疗行业通用的简洁界面布局，功能模块分区清晰（如测量方式选择区、图表展示区、统计详情区、统计分析区），关键操作按钮（如“数据提交”“生成图表”）突出显示。在数据加载、图表绘制、复杂计算等耗时操作中，显示进度条或 Loading 动画，避免用户误操作；操作完成后通过 Toast 提示或音效反馈结果。

（5）页面 UI：系统整体 UI 设计美观，交互简单明了，且对于各种初次使用系统的人来说不存在一定的上手成本。

（6）性能优化：需要确保该系统可以流畅的加载各种可视化图表，同时在文件导入过程中也可以顺利导入，不会造成页面的卡顿崩溃等，在系统运行时页面性能良好，用户体验舒适^[2]。

第 4 章 系统架构说明

4.1 总体架构说明

该数据分析展示系统采用分层架构设计，涵盖前端展示、后端服务与数据库存储三大核心模块。经调研论证，后端系统选用 Python 的 Django 框架实现快速开发，前端基于 Vue 构建用户交互界面，数据库则采用 MySQL 完成对髌骨等多类型骨模型数据的存储与管理，形成前后端分离、数据持久化支撑的完整技术架构。

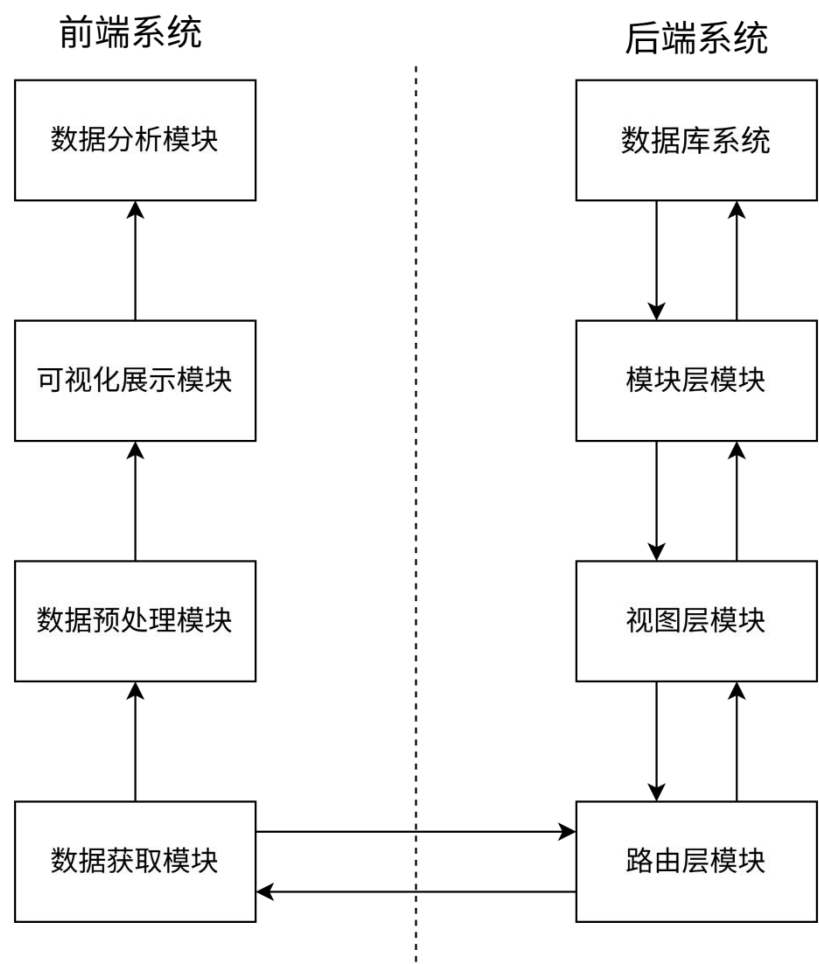


图 4.1 总体架构说明

4.2 前端架构与模块设计

前端系统采用 Vue2+TypeScript+Echarts+ElementUI 技术栈

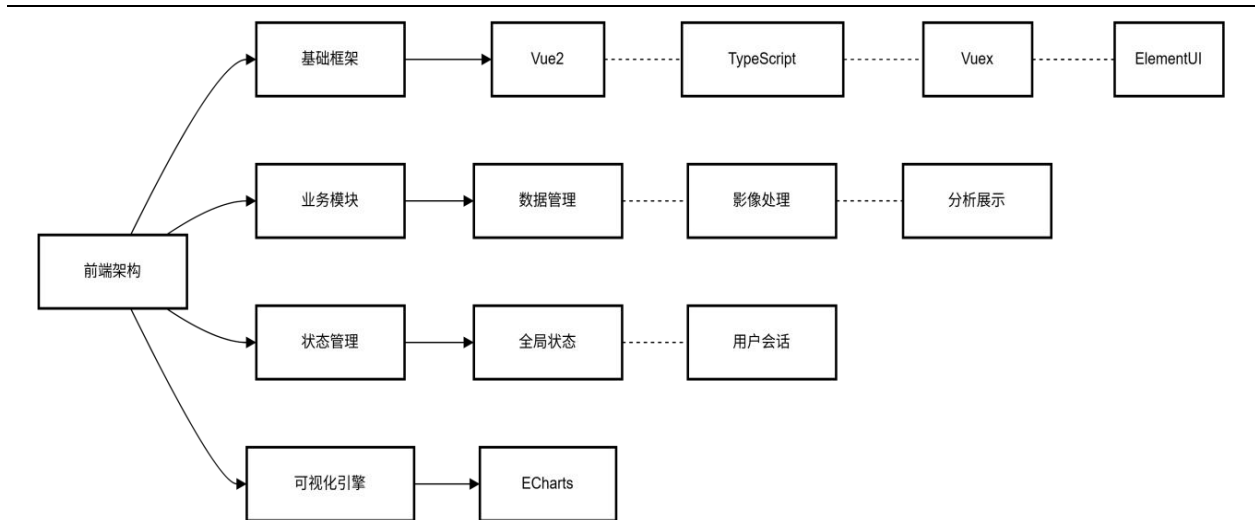


图 4.2 前端架构

4.2.1 前端逻辑结构

骨科测量数据可视化及分析系统的前端模块采用现代化 Web 技术架构，基于 Vue2+TypeScript 构建，通过模块化设计实现了数据可视化展示、数据分析、数据上传等核心功能。系统前端架构分为数据交互层、业务逻辑层、可视化渲染层和用户界面层四个层级^[16]。

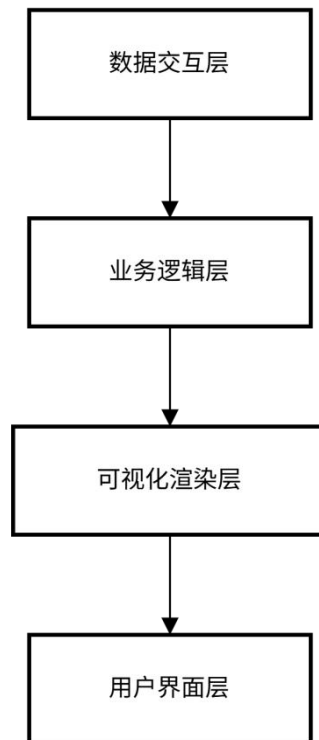


图 4.3 前端逻辑架构

在数据交互层，系统通过 Axios 封装 RESTful API 请求，实现与后端服务的通信。该层特别针对骨科数据特点设计了专用的请求拦截器和响应处理器，能够将数据转换为前端可处理的 JSON 结构，本地数据管理采用 Vuex 进行状态持久化。

业务逻辑层作为系统的核心处理引擎，包含两大功能模块：数据清洗模块将不符合规范的数据进行过滤，提取需要的数据，同时对未直接给出的数据，使用现有的值进行计算得到新的数据；业务规则模块实现了现有的数据分析，通过计算得出相应的均值等。该层采用 TypeScript 严格类型校验，确保计算过程的准确性。

可视化渲染层整合了 ECharts 处理图表。针对骨科需求特别开发了多种图表类型，包括多种模型的多种参数的多种图表展示，用来分析参数的分布以及趋势等

用户界面层基于 Element UI 构建符合用户操作习惯的数据测量的可视化展示交互图。采用响应式布局适配从网页到手机的多种设备，核心操作路径优化至 3 步以内完成。系统界面通过模块化设计，可根据测量方法动态加载不同的骨模型数据。

4.2.2 前端模块设计

（1）入口文件（main.ts）：初始化 Vue 应用，注册全局组件、插件（如 Element Plus、Pinia、Router），挂载到根节点。

（2）路由管理（router/index.ts）：所有页面通过 Vue Router 进行路由跳转，采用懒加载和命名路由。路由守卫可实现登录鉴权、权限控制等逻辑。

（3）状态管理（Vuex）：用户信息、全局消息、权限等通过 Vuex 集中管理，组件间共享状态。

（4）布局与导航：统一的布局组件（如 Header、Sidebar），通过插槽或嵌套路由渲染各业务页面。响应式设计，适配不同终端。

（5）页面与业务逻辑：每个页面（如 createData、MediaImage、Feature 等）为独立组件，负责各自的数据获取、表单处理、数据可视化等逻辑。组件内部通过 API 请求获取/提交数据，处理用户交互，调用 ECharts 渲染图表。

（6）通用组件：表单、对话框、卡片、按钮等通用组件，提升复用性和一致性。

（7）样式与主题：模块化 CSS，支持主题切换和响应式布局。

通过这样的前端架构设计，系统实现了良好的用户界面和交互体验，同时保证了代码的可维护性和可扩展性。Vue 2 的组合式 API 和 TypeScript 的结合，也为开发团队提供了更好的开发体验和代码质量保证。

4.3 后端架构与模块设计

后端架构主要分为 django 后端系统和 mysql 数据库系统^[4]

4.3.1 后端模块设计

后端系统主要基于 Django 框架实现后端开发，采用 MVT 架构，将功能划分为模型层、路由层、视图层三大核心模块。

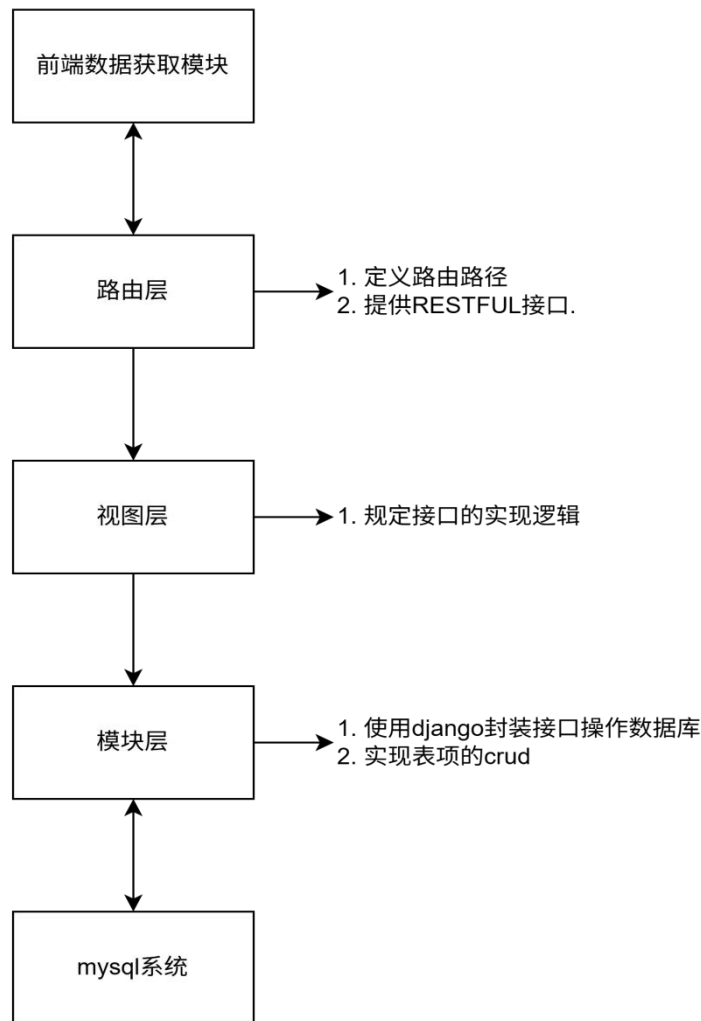


图 4.4 后端模块设计

路由层通过模块化设计提升系统可维护性与扩展性。借助 `include()` 函数，系统可将不同功能模块（如患者数据管理、数据上传、测量分析）的路由配置拆分至独立的应用级 `urls.py` 文件，实现代码解耦；通过命名空间机制（`app_name`），有效避免不同应用间同名路由的冲突，增强反向 URL 解析的准确性与稳定性。此外，路由层与中间件体系深度集成，在请求匹配前，可通过中间件对请求进行身份认证、权限检查、限流等预处理操作，并支持全局异常处理器配置，统一处理路由匹配失败（404）或服务器内部错误（500）等异常情况，如表 1 所示为主要 url 接口地址及用途。

表 1 主要 url 接口地址及用途

url 地址	接口用途
/measureData/PatellaOldData	髌骨稳定性老方法
/measureData/PatellaNewData	髌骨稳定性新方法
/measureData/TibiaBoneData	胫骨测量方法
/measureData/FemurBoneData	股骨角度测量方法
/measureData/UlnarBoneData	尺骨测量方法
/measureData/Show	全局默认展示

模型层作为数据持久化的抽象表示，通过定义 Python 类映射数据库表结构（如 PatellaOldModel、FemurModal、TibiaModel 等模型），封装数据存取逻辑，提供 ORM 操作接口，确保数据的完整性与一致性。在骨科系统中，模型层负责存储各个骨模型各个参数等核心数据，并通过外键关联实现实体间的关系建模，为上层业务提供标准化的数据访问方式^[13]。

视图层作为请求处理的逻辑中枢，接收客户端 HTTP 请求，调用模型层数据并执行业务逻辑（如测量算法计算、数据可视化处理），最终返回格式化响应。在骨科系统中，视图层实现了测量分析计算、结果可视化等核心功能。例如，通过 CBV（类视图）或 ViewSet 快速实现 RESTful 接口，利用 Django REST Framework 的序列化器完成数据校验与转换，将复杂的骨科测量算法（如 骨密度分析）封装为可复用的服务组件，并通过异步任务处理耗时的影像分析操作，确保系统响应性能。模型层与视图层的协同工作，使系统既能高效管理医疗数据，又能灵活响应临床业务需求，成为骨科数据测量与分析的坚实技术基础。

4.3.2 数据库模块设计

本系统采用分层模块化设计，基于 Django ORM 与 MySQL8.构建，实现医疗数据从采集到分析的全生命周期管理。整体分为三大核心模块，数据存储层采用 JSON 字段融合 DICOM 元数据与结构化测量结果，支持多模态数据查询；业务处理层利用 Celery 异步任务实现测量计算，通过星型模型优化分析查询性能^[4]；系统接口层通过标准化 API 对接数据与前端应用。

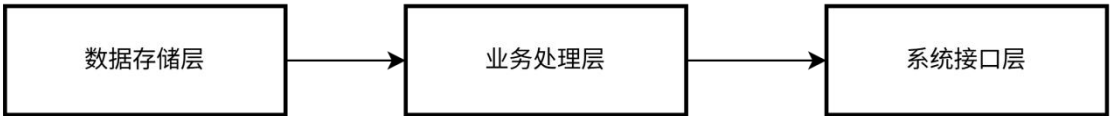


图 4.5 核心模块

系统采用 MySQL 8.0 作为数据库管理系统，主要包含以下核心表，以患者的病号 id 作为主键将五个表进行连接：

(1) 髌骨数据记录表(PatellaOldTable)

存储五种基本形态参数，包含 tt_tg（胫骨结节到股骨滑车沟的距离）、tkLength（通髌线长度）、tt_te（胫骨结节到股的距离）和 tt_tg / tkLength、tt_tg / tt_te 作为进行存储的数据类别。

(2) 胫骨测量数据表(TibiaBoneTable)

存储五种基本形态参数，包含 Tibial Shaft Angle（TSA 胫骨角）、Tibial Varus Angle（TVA 胫骨内翻角）、Proximal Tibial Angle（PTA 胫骨上角）、Posterior Tibial Slope（PTS 胫骨后倾角）、Tibial Plateau Angle（TPA 胫骨平台角）。

(3) 股骨测量数据表(TibiaBoneTable)

主要包含五个核心测量指标。首先是股骨长度(Femoral Length, FL)，其次是颈干角(Neck-Shaft Angle, NSA)，第三个参数为股骨前倾角(Femoral Anteversion Angle, FAA)，第四个存储参数是内侧偏移量(MO)，最后是股骨远端外侧角(LDFA)，所有参数均保留两位小数精度。

(4) 尺骨测量数据表(TibiaBoneTable)

主要为尺骨内翻角点至鹰嘴尖的距离。

(5) 首页数据展示(MainTable)

首页图表数据，存储多种数据结构，包含 showBoneDiease 骨质疾病与年龄分布数据，showBoneAboutData 骨头相关数据展示（肱骨长度等）

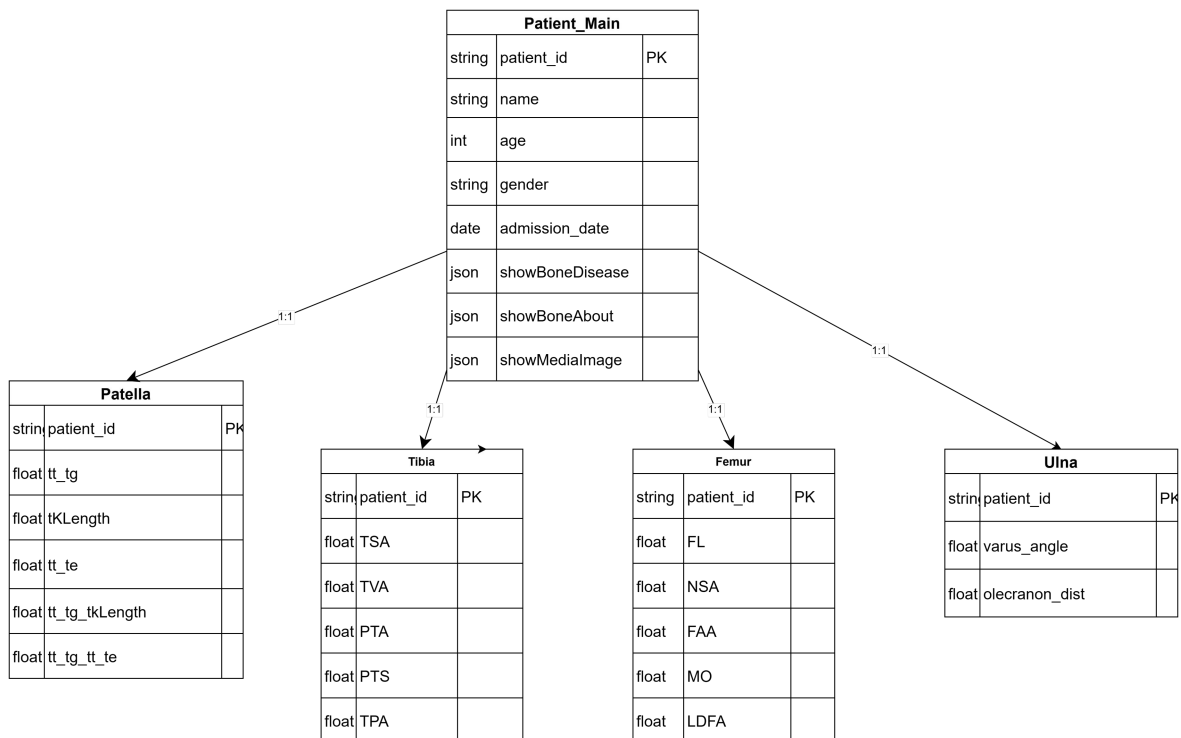


图 4.6 数据库 ER 图

第 5 章 系统实现

5.1 开发环境与工程化配置

5.1.1 前端开发环境

（1）基础环境搭建：安装 Node.js ($\geq 16.x$) 和 npm ($\geq 8.x$)，确保支持 ES6+ 特性。使用 Vue CLI 创建项目，选择 Vue 3 + TypeScript + Vite 模板，提升构建速度。集成 ECharts 用于医学数据可视化，支持骨骼结构三维重建、测量指标趋势分析等功能。

（2）开发工具链：代码格式化 Prettier 统一代码风格，配合 ESLint 检查语法错误。类型检查通过 TypeScript 增强代码健壮性，尤其在处理复杂医学数据结构时减少运行时错误。

（3）状态管理：Vuex 实现组件间数据共享，管理用户会话、测量结果等全局状态。路由管理：Vue Router 实现单页面应用导航，支持懒加载优化性能。

（4）工程化配置：模块化开发使用按功能拆分组件（如测量工具、影像查看器、数据分析面板），提高代码复用性。API 封装：axios 统一处理 HTTP 请求，包含拦截器实现 JWT 认证和错误处理。

（5）环境变量：区分开发、测试、生产环境配置（如 API 地址、ECharts 主题）。

5.1.2 后端开发环境

（1）基础环境搭建：创建虚拟环境（venv 或 conda），安装 Django (≥ 4.0) 和必要依赖。数据库：MySQL 存储结构化数据（患者信息、测量记录），配置主从复制保障数据可靠性。REST API：Django REST Framework 构建接口，支持认证（JWT）、权限控制 and 数据序列化。

（2）核心模块配置：模型层：设计规范化数据表（患者、影像、测量指标），使用外键关联确保数据完整性。视图层：基于类的视图（Class-Based Views）实现高效的请求处理逻辑。中间件：集成 CORS 处理跨域请求，添加自定义中间件记录请求日志和性能监控。

（3）扩展功能：异步任务：Celery + Redis 处理耗时操作（如影像三维重建、复杂算法计算）。文件存储：配置 MEDIA_ROOT 存储医学影像，支持 S3 等云存储扩展。定时任务：Celery Beat 执行数据备份、统计报表生成等周期性操作。

5.2 前端系统实现

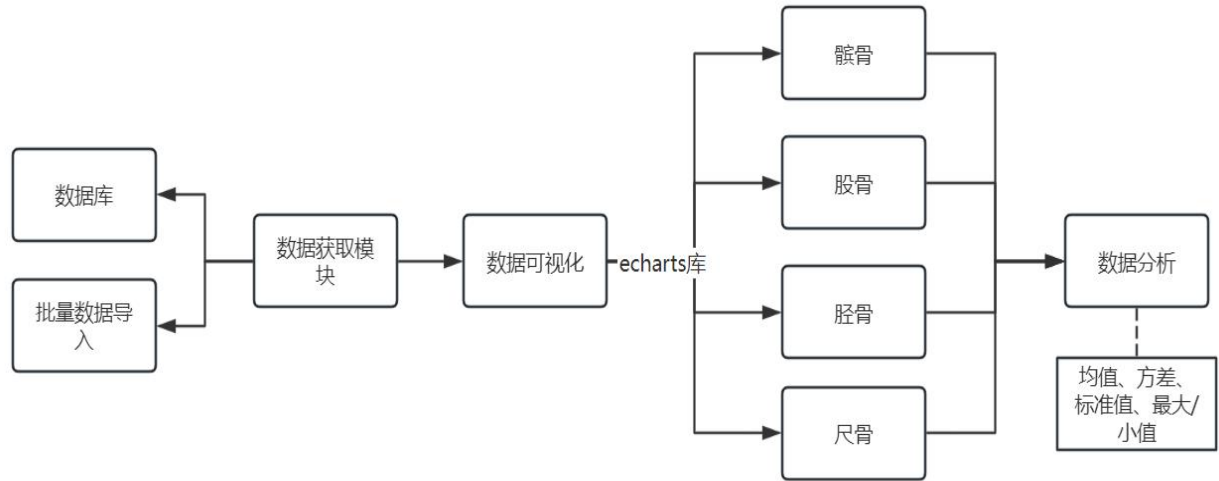


图 5.1 系统功能实现

5.2.1 数据获取

在骨科测量数据可视化及分析系统的前端开发中，数据获取是连接后端数据库与前端展示的关键环节，前端通过 Axios 库实现与后端 API 的 HTTP 通信。数据获取流程包括请求发起、参数处理、接口响应三个核心环节，结合组件状态管理 Vuex 实现数据的跨组件共享与响应式更新。架构设计遵循分层原则，将 API 请求逻辑与组件视图分离，提高代码复用性。

（1）请求发起：在 Vue 组件中，主要通过生命周期钩子触发数据请求。请求发起是数据获取的起点，即创建 HTTP 请求并发送到服务端。以数据测量及展示组件为例，在 mounted 钩子函数通过 fetchData 方法调用相应接口获取数据,主要是通过 axios 向后端发起相应的 get/post

（2）参数处理：确保请求数据正确传递的关键环节。首先是参数类型,主要分为两大类,一种是 Query 参数,用于 GET 请求,附加在 url 后方.以我们的髌骨相关测量数据获取接口为例,我们在发起 get 请求时可以携带 method 作为 query 参数,进行数据请求.另一种是请求体 (Request Body),用于 POST/PUT 请求,通常为 JSON 或表单数据.此外,一些特殊情况下我们还需要进行参数转换,将前端数据结构转换为后端需要的数据结构

（3）接口响应：接口响应作为前后端数据交互的关键环节，指后端接收到前端请求后返回数据及状态信息的过程，前端需对其进行解析与处理。响应数据通常包含状态码 (如 200 表示成功、404 表示资源未找到)、提示信息和具体业务数据,以系统对测量肱骨数据发起的请求为例，前端通过响应拦截器统一捕获响应结果，如果返回 code 为 200 则表示请求成功,此时 messgae 为数据请求成功,同时返回肱骨相关测量数据.并采用 JSON 等格式传输.接着对成功响应进行数据解析和结构适配得到最终的用于页面渲染的数据，对错误响应（如网络异常、权限不足）进行分类处理，包括提示用户、自动重试或跳转页面；同时，为提升性能和安全性，还需结合缓存机制减少重复请求，对敏感数据进行脱敏，并通过 CSRF 防护等手段防止攻击，确保数据交互稳定可靠。

至此,我们数据的数据获取已完成,即对于下一步的页面的渲染做好了准备.

5.2.2 数据可视化

骨模型数据可视化通过整合数据库存储、API 服务和前端 ECharts 图表技术,实现了髌骨、胫骨、股骨、尺骨四种骨骼数据的高效展示。系统将数据存入数据库,通过 REST API 提供标准化数据接口,前端调用接口获取骨骼长度、密度等参数后,使用 ECharts 渲染柱状图对比不同骨骼参数差异^[1],或用雷达图综合展示多维度指标。前端实现流程主要分为以下几个步骤:

用户选择阶段: 骨模型选择器: 通过下拉菜单 (Select) 或卡片式组件让用户选择髌骨、胫骨、股骨或尺骨。图表类型选择器: 提供柱状图、雷达图、折线图等选项。数据请求阶段: 用户选择后, 前端通过 axios 调用后端 API, 传递所选骨模型参数 (如 boneType=股骨), 接收后端返回的 JSON 数据 (如长度、密度等参数)。ECharts 渲染阶段: 根据用户选择的图表类型, 动态配置 ECharts 选项 (option)。调用 echarts.init() 初始化图表实例, 再通过 setOption() 渲染图表。交互更新阶段: 支持用户切换骨模型或图表类型, 重新请求数据并更新图表。提供图表内交互 (如悬停显示详情、点击查看数据点)。

(1) 髌骨数据可视化模块

首先是髌骨模型的相关测量数据的可视化展示, 该骨模型有五个基本参数分别为 tt_tg (胫骨结节到股骨滑车沟的距离)、tkLength (通髌线长度)、tt_tg / tkLength (参数 1 和参数 2 相除)、tt_te (胫骨结节到股的距离)、tt_tg / tt_te (参数 1 和参数 4 两者相除), 因此我们的髌骨数据模块每种图 (柱状图、折线图、雷达图等) 都需要展示五张图。

我们需要定义获取数据的函数, 即 fetchData 函数, 通过 axios 进行访问 http://47.122.26.11:8080/measureData/ show 接口, 获取到相关数据, 整体数据以对象数据构成, 每一个对象中主要包含有 tt_tg、tt_te、tkLength 三种数据。每一个对象待变一个患者的相关数据。

以 tt_tg 胫骨结节到股骨滑车沟的距离的柱状图表展示为例, 我们需要提取出来 tt_tg 的相关数据, 就需要进行对其进行遍历然后将 tt_tg 的值存储在一个新的数组中。使用这部分的数据进行图形的绘制。

获取到相关数据后我们就需要进行图表的绘制了, ECharts 作为专业的数据可视化库, 其图表配置对象包含 title, 我们将测量的数据命名为图表名称即胫骨结节到股骨滑车沟的距离, xAxis/yAxis: 直角坐标系的横纵轴配置, 包含 type: 坐标轴类型 ('category'用于患者 ID, 'value'用于测量值), name: 轴名称 (必须包含单位, 如"距离(mm)") min/max: 显式指定坐标范围, 确保不同图表量纲一致。series: 核心数据系列配置数组, 每个元素代表一组数据展示形式, type: 图表类型 (bar 柱状图), name: 系列名称 (显示在图例

中），**data**：绑定数据源（支持 Array|Object 格式），**coordinateSystem**：指定坐标系类型（'cartesian2d'|'polar'），**itemStyle**：数据项样式，医学图表建议使用 **fill:'none'**保持透明度。

系统实现过程中特别注重交互体验的优化。通过 Vue 的 v-model 指令实现图表类型的双向绑定，用户可通过下拉菜单自由切换视图模式。初始数据获取可以在 mounted 中进行，后续我们需要对获取到数据存储起来进行监听，一旦数据发生变化，我们需要再次调用 **fetchData** 函数，保证视图数据同步更新。所有图表均配置了自适应调整机制，当浏览器窗口尺寸变化时自动触发 **resize** 方法重绘图表。为提高渲染性能，对大数据集采用了渐进式渲染策略，并利用虚拟 DOM 技术减少不必要的重渲染。图表提示框（**tooltip**）经过定制化开发，可显示患者 ID、参数值和临床参考范围等关键信息。

髌骨数据可视化模块中各个参数的图形展示同 **tt_tg** 胫骨结节到股骨滑车沟的距离的图表展示，值得注意的是 **tt_tg / tkLength**（胫骨结节到股骨滑车沟的距离、通髌线长度两者相除）、**tt_tg / tt_te**（前两者相除）这两个参数，我们获取到的参数中不存在直接的值，需要我们进行间接计算，所以在对这两个类型的数据进行存储时，同样进行遍历，计算 **tt_tg / tkLength**、**tt_tg / tt_te** 的值将其依次到数组中，在进行可视化时使用新的数据。

（2）胫骨数据可视化模块

胫骨测量数据与髌骨数据可视化展示类似，测量值存在差异，胫骨中需要测量的数据主要有胫骨角、胫骨内翻角、胫骨上角、胫骨平台角。这四个值，因此我们不同类型的图表展示至少需要四张图。

通过调用胫骨数据获取的接口 **'http://47.122.26.11:8080/measureData/Show_TibiaBone'**，后端从数据库中拿到相应的数据，将其返回，前端进行接收数据，此次四种数据都存在，因此不需要进行额外计算，只需对其进行遍历提取出相应的数据即可。

在 vue 组件中，使用四个 **div** 进行这四种测量数据的 echarts 图的展示，接下来进行 echarts 的相关配置，以胫骨角为例，**title** 设置为胫骨角，**xAxis** 对象的 **type** 设置为患者 ID，**data** 为数组中每一个测量对象的 **Id**，**yAxis** 设置为 **value** 值，**series** 中的 **data** 即拿到的数据转换后的数据，**type** 为 **line**，表示我们绘制的为折线图。

（3）尺骨数据可视化模块

尺骨测量数据为主要有两个基本参数：**UlnarLength** 尺骨长度（鹰嘴到尺骨茎突的距离）、**UlnarTiltAngle** 尺骨倾斜角（尺骨远端关节面与水平面的夹角）。我们需要调用后端接口 **http://47.122.26.11:8080/measureData/Show_UlnarBone** 获取到这两组数据，对参数进行解析转换，然后创建图表。

以绘制股骨测量数据的雷达图为例，雷达图需要完全不同的配置结构：**radar** 对象：定义 **indicator** 数组，每个指标包含名称和最大值。**series**：**type: 'radar'**指定为雷达图，**data** 数组包含一个对象，其 **value** 为尺骨测量值的数组，自动生成从中心点向外辐射的坐标轴，直观展示各项指标的相对关系

（4）股骨数据可视化模块

股骨角度测量数据主要为四个基本参数，因此对该模块下的数据可视化展示我们至少需要四张图表，即我们需要设置四个 `div`，我们通过 `http` 去请求后端接口 `http://47.122.26.11:8080/measureData/Show_FemurBone` 获取数据，将获取到数据分别进行提取解析，生成四个数组使用数据中的数据去渲染图表，然后将图表挂载在设置的四个 `div` 的 `dom` 节点上。

5.2.3 数据分析

在完成髌骨测量参数的可视化展示后，需要对数据进行系统的统计分析，以提取具有临床价值的量化指标。统计分析主要包括以下三个关键环节：

（1）基础统计量计算

针对每个测量参数（`tt_tg`、`tkLength` 等），系统自动计算六个核心统计指标：

算术平均值：反映数据的集中趋势，计算公式为所有观测值之和除以样本量。在髌骨测量中，平均值可帮助判断患者群体的整体特征。

方差：衡量数据离散程度，计算各数据点与均值差值的平方平均数。较大的方差提示患者个体间存在显著差异。

标准差：方差的平方根，与原始数据同量纲，更直观地显示数据波动范围。

最大值与最小值：识别参数的极端值，辅助发现异常病例。

极差：最大值与最小值之差，快速评估数据分布范围。

系统采用 `JavaScript` 的 `reduce()` 等数组方法实现高效计算，所有结果保留两位小数以确保临床报告的精确性。计算结果以结构化对象形式存储，便于后续展示和导出

（2）统计结果可视化集成

统计分析结果通过三种形式与图表展示有机结合：

图表标注：在 `ECharts` 配置中，使用 `markLine` 组件标注平均值参考线，`markArea` 组件显示均值±标准差的范围区间。

独立统计面板：在图表下方以表格形式展示完整统计量，采用响应式设计适配不同屏幕尺寸。

交互提示：当鼠标悬停在数据点上时，`tooltip` 不仅显示当前值，同时提示相对于均值的偏离程度（如“高于均值 1.2 个标准差”）。

（3）分析结果

分析结果以文字阐述和 `UI` 体现两种方式进行

图表说明：通过图表显示每位患者的骨科测量数据的测量值，正常结果和异常结果颜色进行区分显示，正常区域显示均值±1 标准差范围（11.88-15.38mm），存在标准值虚线进行辅助查看说明

数值统计：本次抽样数组内，统计超过平均值组数、低于平均值组数、连续上升最大数、连续下降最大数以及用来医学分析的正常数、临界数和异常数。

如下图即 tt tg 胫骨结节到股骨滑车沟的距离相关数据分析：

我们在选择髌骨稳定性方法后，选择柱状图进行展示，髌骨主要有五类测量数据，下图即为胫骨结节到股骨滑车沟的距离的 21 组数据展示，我们通过计算得出平均值为 80.95，方差为 217.00，最大值 92，最小值 20，标准差为 14.73，而在计算出这些值的基础之上，我们进行了相应的分析，比如超过平均值的连续段数为 3，而低于平均值的连续段数为 4，此外还有连续上升、连续下降等数量，以及从医学角度对正常、临界、异常数也进行了分析，异常数我们在图表中会以红色来进行警示，同时我们在柱状图中会将平均值以蓝色虚线的形式进行展示，便于查看图中数据分布，当点击每一个单独的数据时，会出现当前的值的一些分析，例如，此数据由平均值 \pm n 个标准差构成。



图 5.2 数据分析展示

5.2.4 数据导入

系统除可以直接从数据库中提取数据，我们也设计了标准化的数据导入流程，实现了髌骨、胫骨、股骨和尺骨等测量数据的高效录入。

前端导入模块主要包含以下功能实现：数据上传功能支持 Excel 文件（.xlsx/.csv）的批量导入^[10]，用户可通过拖拽或文件选择对话框上传符合模板规范的数据文件。

数据转换阶段，前端将文件上传成功后，后端将原始 Excel 数据转换为标准 JSON 格式，并进行三重验证：首先检查必填字段（如患者 ID、测量日期等），其次验证数值范围（如 tt_tg 值需在 5-25mm 之间），最后校验 ID 有效性。

数据提交采用分块上传机制，大文件自动拆分为 5MB 的块数据，通过 Web Worker 实现多线程传输^[18]，避免界面因文件过大造成卡顿。上传进度可以通过进度条进行可视化展示，帮助用户在提交前确认数据质量。系统支持断点续传和错误自动重试（最多 3 次），网络中断后可从最后一个成功块继续上传^[19]，符合现阶段使用。

数据存储采用差异化的数据库策略：结构化测量数据（如 tt_tg 值）存入 MySQL 关系型数据库，确保事务完整性；而原始文件及其解析日志则存入 MongoDB 文档数据库，保留完整的导入溯源记录。数据库操作通过 RESTful API 封装，前端调用统一的/upload 端点，由后端根据 Content-Type 自动路由到对应处理模块（如 multipart/form-data 走文件流程，application/json 走直插流程）。

成功导入的数据会触发反馈：前端立即显示本次导入的统计摘要（如“成功导入 32 条骸骨记录”）。系统保留最近 10 次的导入历史记录，支持按时间范围、操作人员和骨骼类型进行联合查询。

数据导入的实现方案通过严谨的前端验证、高效的数据传输和可靠的存储机制，在保证数据质量的前提下，将典型 Excel 文件的导入处理时间控制在极好的性能水平，同时提供完善的错误处理和溯源能力，满足临床研究对数据准确性和完整性的严格要求。

5.3 后端系统实现

5.3.1 Django 配置说明

（1）基础环境配置

在 Python 项目开发实践中，模块版本兼容性问题不容忽视，特别是 Python 运行环境与第三方框架版本的适配关系，直接影响系统稳定性。本项目采用 Python 3.11 作为开发环境，经过多轮测试验证，最终选定 Django 3.2.23 版本作为后端开发框架，确保各组件间的协同工作。

搭建 Django 项目架构时，遵循标准开发流程。首先执行 `django-admin.py startproject BoneDataproject` 命令创建项目根目录，使用 BoneDataproject 作为项目主应用名称。针对数据可视化模块开发需求，通过 `python3 manage.py makemigrations` 指令创建名为 `measureData` 的子应用注意的是，在 CentOS 操作系统环境下开发时，系统默认存在 Python 2 和 Python 3 双版本共存的情况，因此所有指令均需显式指定 `python3` 版本，以避免版本混淆问题。

（2）项目核心配置

完成基础环境搭建后，进入项目核心配置阶段。在路由配置方面，通过在 `BoneDataproject/urls.py` 文件的 `urlpatterns` 列表中添加 `path('measureData/', include(('measureData.urls', 'measureData')))` 语句，建立起主路由与 `measureData` 子应用的映射关系，为后续二级路由开发奠定基础。该配置实现了 URL 路径与功能模块的解耦，便于进行模块化开发和维

护。

项目配置文件 `settings.py` 中，多项参数需根据实际需求调整。DEBUG 参数在开发阶段可保持默认的 TRUE 状态，方便开发人员通过详细报错信息定位问题；项目上线前则需修改为 FALSE，避免敏感信息泄露。ALLOWED_HOSTS 配置项用于设置访问白名单，开发测试阶段可临时设置为通配符*，允许任意 IP 访问；在生产环境中，为保障系统安全，需结合服务器防火墙规则，采用精准 IP 限制策略，防止恶意访问。

INSTALLED_APPS 配置项负责注册项目中所有应用，只有在此处声明的应用才能被 Django 框架识别并启用。与之类似，MIDDLEWARE 配置列表用于注册系统中间件，开发者在引入第三方中间件时，除通过 pip 完成安装外，还需在此处进行注册配置，才能激活中间件功能。

通过上述步骤，完成了 Django 项目的基础配置工作，为后续后端业务逻辑开发、数据接口实现以及系统功能扩展提供稳定可靠的运行环境。

5.3.2 模型层开发

在骨科测量数据可视化系统的后端实现中，数据模型层作为连接业务逻辑与数据库的核心枢纽，承担着数据结构定义、关系映射及持久化的重要职责。

在数据库模型定义方面，基于 Django 框架的 ORM 机制，系统采用领域驱动设计理念，构建面向骨科医学场景的多层次数据模型体系。确定骨模型类型，涵盖髌骨、胫骨、股骨、尺骨四大类。针对不同骨骼类型，分析其测量数据维度如下：

髌骨：包含 `tt_tg`（胫骨结节至股骨滑车沟距离）、`tkLength`（通髌线长度）、`tt_tg/tkLength`（两者之比）、`tt_te`（胫骨结节至股骨外上髁距离）、`tt_tg/tt_te`（两者之比）。

胫骨：涉及胫骨角、胫骨内翻角、胫骨上角、胫骨后倾角、胫骨平台角等角度测量数据。

股骨：包括股骨髁间沟角、股骨髁间角、股骨下角、股骨髁间角、股骨髁间沟角、股骨外侧髁角等参数。

尺骨：主要为尺骨内翻角点至鹰嘴尖的距离。

对测量数据进行确定完成后，我们需要在 `models.py` 文件中进行数据库的模型定义。对于 Django 来说，`models.py` 文件的定义是通过 python 的类的数据结构来实现的。利用 Python 类结构映射数据库表，借助 Django 默认父类 `models.Model` 的 `id` 属性作为主键，将不同骨模型类关联至患者维度，其中 `id` 属性绑定患者病号，确保数据溯源性。经分析，定义以下核心模型类，如下表所示

表 2 模型层定义

模型	含义
PatellaOldModel	髌骨稳定性老型测量方法数据模型
PatellaNewModel	髌骨稳定性新型测量方法数据模型
TibiaModel	胫骨相关测量数据模型
FemurModel	股骨相关测量数据模型
UlnarModel	尺骨相关测量数据模型

在完成骨模型测量数据的定义后，要想与数据库一致。首先需要连接数据库。Django 对不同的数据库有不同的数据连接方式，即不同的连接驱动。配置在 `settings.py` 中，具体的字段 `DATABASES`，在 `django` 中默认的数据库连接是 `python` 自带的 `sqlite3`，但是该数据库属于键值数据库，对于我们的需求的满足能力不如 `mysql`，所以我们使用 `mysql` 的数据连接驱动。因为我们的数据库安装在阿里云的云服务器中，我们需要的字段分别是 ‘ENGINE’（数据连接引擎）、‘NAME’（数据库名）、‘USER’（用户名）、‘PASSWORD’（数据库密码）、‘HOST’（阿里云云服务的 ip 地址或者是域名）和 ‘PORT’（端口）。其中连接引擎对于 `mysql` 就是 `django.db.backends.mysql`，数据库名我们设置为最终创建的 `3DModel` 库，用户名理论上不能设置为 `root`，要设置为自定义的用户，以保证权限的最小化保证数据库安全，这里为了快速开发，设置为 `root` 即管理员，并且已经提前开启了远程登录的权限。端口也是同理，不能在生产环境中简单的设置为默认的 `3306`，这里也为了快速开发设置为默认。完成上述配置后，测试数据库的连通性，执行 `install_as_MySQLdb()` 函数。完成连通性测试后就可以进行 ORM 的交互。

在完成对骨模型测量数据的定义和数据库连接的配置后，接下来就需要对数据进行操作已完成最初定义的保证数据可视化后端数据提供的需求。在此之前，需要让我们定义的类和 `mysql` 数据库的表进行绑定。反应在终端的操作上就是完成迁移和同步，即 `python3 manage.py makemigrations` 和 `python3 manage.py migrate`。通过迁移和同步后，在数据操作层面，系统基于 ORM 机制实现完整的 CRUD 功能：

数据保存：通过接口接收外部数据后，创建模型对象并调用 `save()` 方法，例如将胫骨新型测量数据字段 `TSA`、`TVA`、`PTA`、`PTS`、`TPA` 我们创建一个 `obj` 字典，这个字典继承于我们定义的 `ShowModel` 类，将字段进行对应后，导入 `model` 模块，执行 `obj.save()`。在数据库的对应表中进行保存，通过终端连接到数据库，查看对应的表，具体执行 `select * from measureData_showmodel limit 1`；如果数据一致，则证明保存完成。

数据删除：前端通过 `Axios` 发送携带 `ID \Method` 或 `chartType` 参数的 `GET` 请求，后端在接受到请求后，通过 `orm` 的匹配操作例如 `obj = ShowSTLModel.objects.get(pk=object_id)` 匹配到该数据，执行 `obj.delete()` 就完成了删除，删

除出问题时需要定义错误类型，定义如果删除错误返回 500，成功执行返回 200。数据更新：基于 ID 查询对象后，通过 `update()` 方法修改字段值，成功返回 200，失败返回 501；

数据修改：首先通过前端携带的 `query` 参数，后端通过 `orm` 的匹配操作例如 `obj = ShowSTLModel.objects.get(pk=object_id)`，然后再重新执行 `update()` 函数进行数据更新。这里定义成功改正返回的状态码为 200，执行错误则为 501。

数据查询：提供全量数据查询接口，将数据返回前端用于图表生成等可视化操作。

5.3.3 路由层开发

我们知道浏览器发出一个请求，在经过 `wsgiref` 和中间件后，就到达了路由层。那路由层的作用是啥呢？如果我们把一个 `django` 项目当作一本书，那么路由层相当于就是书的目录，我们可以根据目录查找到相关的章节位置，同样在 `django` 中，也是根据路由层来找到对应的视图函数。路由，就是请求地址和视图函数的映射关系。

`Django` 中路由层的默认配置在 `urls.py` 文件中，由一条条映射关系组成的 `urlpatterns` 这个列表称之为路由表。`url` 本质就是一个函数，其中有四个参数：`regex` 正则表达式，用来匹配 `url` 地址的路径部分，例如地址为：`http://127.0.0.1:8001/index/`，正则表达式要匹配的部分是 `index/`，`view` 通常为一个视图函数，用来处理业务逻辑，`kwargs` 用于在路由中传递参数，`name`：用于路由反向解析。而第二个参数不再是单独的视图函数而是一个具体的 `python` 文件，由此来导入我们应用里面的路由层定制化配置。

以本系统来说，我们在项目下的路由层配置文件 `urls.py` 进行应用的配置，具体是在 `urlpatterns` 列表中，定义对应的函数值 `'path('measureData/', include(('measureData.urls', 'measureData'))),'` `'measureData'` 是我们的应用名，即自动测量及数据可视化模块，`path` 函数是 `django` 提供的函数，其作用是进行路由匹配，即当我们在浏览器中输入 `http://127.0.0.1:8000/measureData`，`django` 会拿着路径部分“`measureData`”去路由表中自上而下匹配，一旦匹配到了就不会继续往下匹配了。

完成项目级路由映射后，需在子应用 `measureData` 的 `urls.py` 文件中进行细粒度的接口路由注册。根据业务需求，需配置增删改查、全量展示及分类型展示等接口。对这些接口进行注册也是在 `urlpatterns` 列表下添加函数类型的变量即可，只是文件位置不同。

全部展示的接口函数在视图层下，即 `views.Show`，`'path('ShowALL', views.ShowALL, name='ShowALL'),'`，部分展示我们根据骨模型的不同，分别为 `'path('Show', views.Show, name='Show'),'`、`'path('Show_New', views.Show_New, name='Show_New'),'`、`'path('Show_TibiaBone', views.Show_TibiaBone, name='Show_TibiaBone'),'`、`'path('Show_FemurBone', views.Show_FemurBone, name='Show_FemurBone'),'` 和 `'path('Show_UlnarBone', views.Show_UlnarBone, name='Show_UlnarBone'),'`，分别代表髌骨稳定性老方法、髌骨稳定性新方法、胫骨测量方法、股骨角度测量方法和尺骨测量方法，根据不同骨模型的分析方法，拆分为多个子路由，实现精细化功能划分：

（1）全量展示接口，路径：`path('ShowALL', views.ShowALL, name='ShowALL')`，用于触发全量数据展示的请求，可能整合所有骨模型的相关信息或展示统一的结果页面。

按骨模型分类的展示接口

（2）髌骨稳定性老方法，路径：`path('Show', views.Show, name='Show')`，对应传统髌骨稳定性分析方法的展示逻辑，用于兼容旧业务逻辑或对比新旧方法差异。

（3）髌骨稳定性新方法，`path('Show_New', views.Show_New, name='Show_New')`，实现改进后的髌骨稳定性分析结果的展示，可能包含优化的算法或交互逻辑。

（4）胫骨角度测量方法 `path('Show_TibiaBone', views.Show_TibiaBone, name='Show_TibiaBone')` 专注于胫骨相关参数（如长度、角度等）的测量结果展示，用于下肢骨骼结构分析。

（5）股骨角度测量方法，`path('Show_FemurBone', views.Show_FemurBone, name='Show_FemurBone')`，处理股骨角度（如颈干角、扭转角等）的测量数据展示，用于髋关节或膝关节疾病的辅助诊断场景。

（6）尺骨测量，`path('Show_UlnarBone', views.Show_UlnarBone, name='Show_UlnarBone')`，聚焦尺骨（前臂骨骼）的测量结果展示

5.3.4 视图层开发

在完成路由层与模型层的配置后，视图层的开发成为连接前后端逻辑的关节环节。视图层通过在 `views.py` 中编写具体接口逻辑，主要实现根据 URL 路由配置，接收前端发送的 HTTP 请求、通过调用模型层获取数据，处理业务逻辑；然后将处理结果转换为 HTTP 响应返回给前端。采用 Django REST Framework 实现视图层，通过序列化器和视图集简化数据处理流程。

首先为我们测量的五种骨模型创建对应的序列化器，用于将模型数据转换为 JSON 格式，分别如下所示：**PatellaOldSerializer**：处理髌骨稳定性老型测量方法数据，定义字段映射规则。**PatellaNewSerializer**：处理髌骨稳定性新型测量方法数据，支持数据格式化。**TibiaSerializer**：处理胫骨相关测量数据，如角度、长度等参数的序列化。**FemurSerializer**：处理股骨相关测量数据，支持复杂数据的转换。**UlnarSerializer**：处理尺骨相关测量数据，定义特殊字段的转换规则。

其次完成通用视图集的实现，基于 Django REST Framework 的 `ModelViewSet` 实现各模型的通用 CRUD 操作。包括 `PatellaOldView` 用于髌骨稳定性老方法的测量数据创建、查询、更新以及删除。`PatellaNewViewSet` 用于髌骨稳定性新型测量方法的数据处理。`TibiaViewSet` 用于胫骨测量数据的增加更新等。`FemurViewSet` 用于处理股骨相关测量数据包括一些数据的计算展示等。`UlnarViewSet` 支持尺骨相关测量数据的一些特殊业务逻辑。

对于各个模型的视图层实现逻辑就是从模型层导入对应的模块类（例如 `PatellaOldModel` 类），我们要继承其模块父类，然后遵循资源导向的 URL 设计，其次将

其测量数据转换为 JSON 格式的数据类型，将其返回，使得结构清晰，同时约定状态码（200 成功，400 为参数错误，401 未认证等）。

第 6 章 系统测试

6.1 测试环境

处理器：AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz
机带 RAM：16.0 GB (15.9 GB 可用)
系统类型：64 位操作系统, 基于 x64 的处理器
浏览器：Chorme
开发工具：Visual Studio Code、Google Chrome
数据库系统：Python 3.10 + Django
接口测试工具：Postman、浏览器调试工具

6.2 系统模块测试

6.2.1 数据可视化模块测试

表 3 数据可视化模块测试结果

测试用例	期望结果	实际结果	图例编号
首页	正常展示	通过测试	图 6.1
数据分类表格	正常展示	通过测试	图 6.2
雷达图	正常展示	通过测试	图 6.3
四种图集聚合	正常展示	通过测试	图 6.4
骨骼年龄分布图	正常展示	通过测试	图 6.5



图 6.1 首页展示

在系统中对骨骼数据的分类维度以及数据类型还有相关具体示例以及图表做了简单说明，使得用户对网站存在一定的功能认知，其次我们在后面的可视化展示中需要对各种数据以不同的图标及进行展现，所以我们在起初的数据分类表格中将不同维度适用的不同图表进行体现，使得用户后期在选择合适的图表进行展示。

骨骼数据分类表格及可视化体现			
分类维度	数据类型	具体示例	适用图表
解剖部位	分类变量（计数/百分比）	- 脊柱（颈椎/颈椎） - 四肢（股骨/桡骨/胫骨） - 关节（腕/膝/肩）	柱状图/饼图
骨折类型	分类变量	- 闭合性 vs. 开放性（Gustilo分级） - 粉碎性/横行/螺旋形骨折 - 病理性骨折	堆叠柱状图
手术方式	分类变量	- 内固定（钢板/髓内钉） - 关节置换（全髋/半髋） - 微创 vs. 传统手术	分组柱状图
治疗效果	连续变量/等级变量	- 骨折愈合时间（周） - Harris髋关节评分（0-100分） - 愈合等级（优/良/差）	折线图/箱线图
并发症	计数/百分比	- 感染率 - 内固定失效 - 深静脉血栓（DVT） - 神经损伤	柱状图（带误差线）
骨密度（BMD）	连续变量	- 腰椎T值（-2.5~+1.0） - 股骨颈骨密度（g/cm²） - 治疗后变化差值	折线图/散点图

图 6.2 骨骼数据分类表格

左图为骨科几种的基本的形态参数，如股骨长度、髌骨长度、骨中段直径等参数，右图主要为骨骼评估、抗压/抗弯强度等参数，以下两幅图均以雷达图的形式进行展示。雷达图以其直观的形态和多维数据整合能力展示特性在进行数据展示时有着特别的适用性。雷达图可以将多个关键骨科测量参数同时呈现在一个闭合的坐标系中，形成清晰可辨的几何轮廓。虽然雷达图每个轴线都表示不同维度，但使用上为了容易理解和统一比较，经常会人为地将多个坐标轴都统一成一个度量，比如：统一成分数、百分比等，这种雷达图在日常生活中也更常见、更常用。

雷达图也称为网络图、蜘蛛图、星图、蜘蛛网图，它被认为是一种表现多维数据的图表。它将多个维度的数据量映射到坐标轴上，每一个维度的数据都分别对应一个坐标轴。

另外，雷达图还可以展示一组数据中各个变量的权重高低情况。

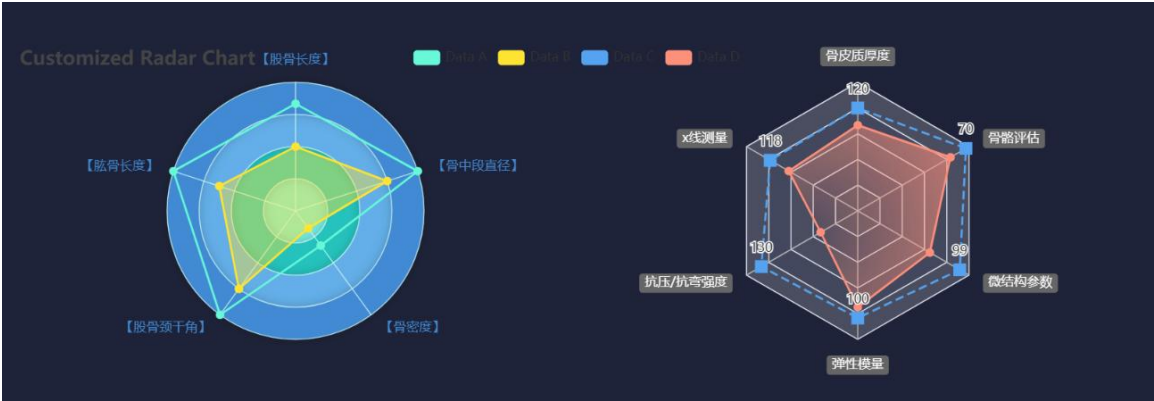


图 6.3 雷达图展示

我们将儿童骨科生长曲线、骨骼模型分类分布，以及骨密度分布，骨科疾病或并发症的流行病系统都以不同的图表进行展示，以骨骼模型分类为例，因为它主要代表了骨骼的各个模型分类都有哪些同时占比多少，因此这种采用饼图的方式进行展示是非常适宜的。而以骨密度分布来看，散点图是指在回归分析中，数据点在直角坐标系平面上的分布图，散点图表示因变量随自变量而变化的大致趋势，据此可以选择合适的函数对数据点进行拟合。因为我们通过适用散点图可以直观的看出骨密度的分布特点。

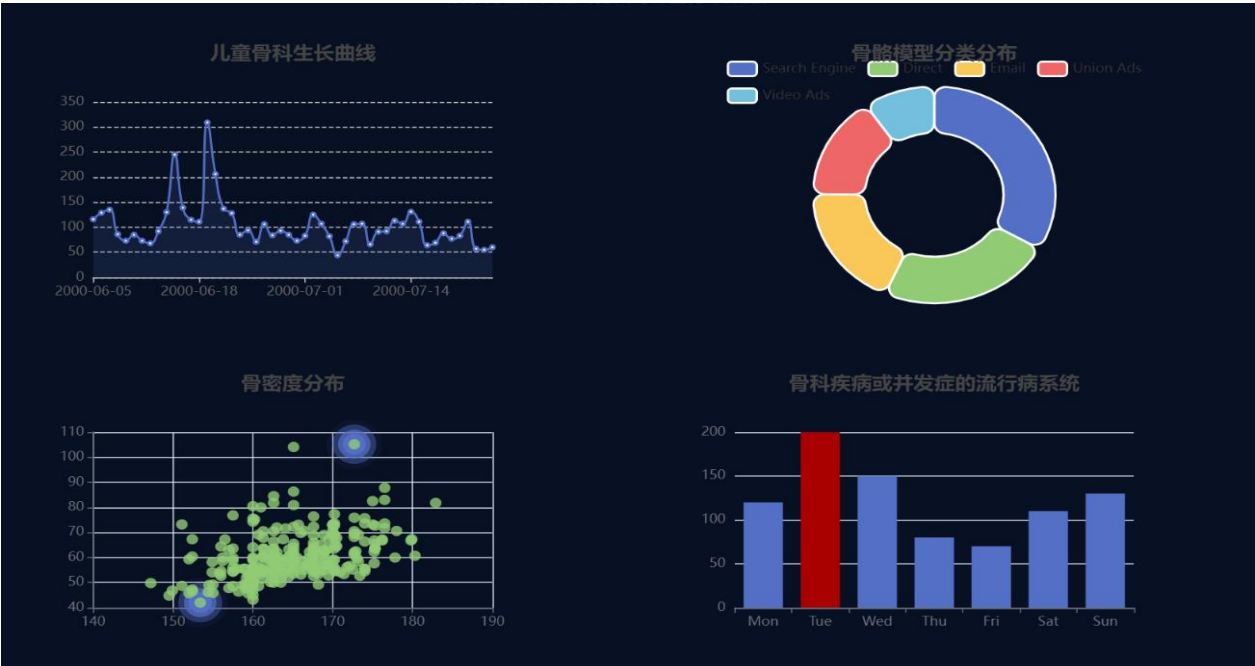


图 6.4 多类型图表展示

下图主要体现男性和女性的身高体重与骨骼年龄的分布图，我们从图中可以看出不同年龄阶段男生和女生的骨骼情况



图 6.5 柱状图展示

6.2.2 测量结果可视化展示模块

表 4 数据测量及展示

测试用例	期望结果	实际结果	图例编号
点击“方法选择”按钮	出现五种方法	通过测试	图 6.6
点击“图表类型”	出现三种方法，包括柱状图、折线图和雷达图	通过测试	图 6.7
再次点击“髌骨稳定性老方法”按钮	展示区域展示对应柱状图 该测量方法下的所有参数的柱状图	通过测试	图 6.8 图 6.9 6.10
再次点击“髌骨稳定性老方法”按钮	展示区域展示对应折线图 该测量方法下的所有参数的折线图	通过测试	图 6.11 图 6.12 6.13
再次点击“髌骨稳定性老方法”按钮	展示区域展示对应雷达图 该测量方法下的所有参数的雷达图	通过测试	图 6.14
再次点击“髌骨稳定性老方法”按钮	展示区域展示对应散点图 该测量方法下的所有参数的散点图	通过测试	图 6.15
再次点击“髌骨角度测量”按钮	展示区域展示对应雷达图 该测量方法下的所有参数的雷达图	通过测试	图 6.16

测量方法的选择，点击即可出现五种对应的方法：髌骨稳定住者方法、髌骨稳定性者方法 2、髌骨测量、髌骨角度测量、尺骨测量。



图 6.6 测量方法选择

图表类型的选择，点击即可出现三种对应的图表类型：柱形图、折线图、雷达图、散点图。



图 6.7 图表类型选择

选择髌骨稳定性方法和柱形图后，即可显示出对于该方法测量下的胫骨结节到股骨滑车沟的距离的柱状图。使用该柱状图可以通过高度不同的柱形来对比不同人员数据的数值大小，从而直观展示出胫骨结节到股滑车沟的距离数据的分布、差异以及趋势。



图 6.8 tt tg 胫骨结节到股骨滑车沟的距离

选择髌骨稳定性方法和柱形图后，即可显示出对于该方法测量下 tkLength 通髌线长度的柱状图。对比不同患者的髌骨 tkLength 通髌线长度值，可发现异常值（过高/过低可能提示病理变化）包括其值是否在正常范围内，如果偏离正常范围，具体偏离正常值为多少。

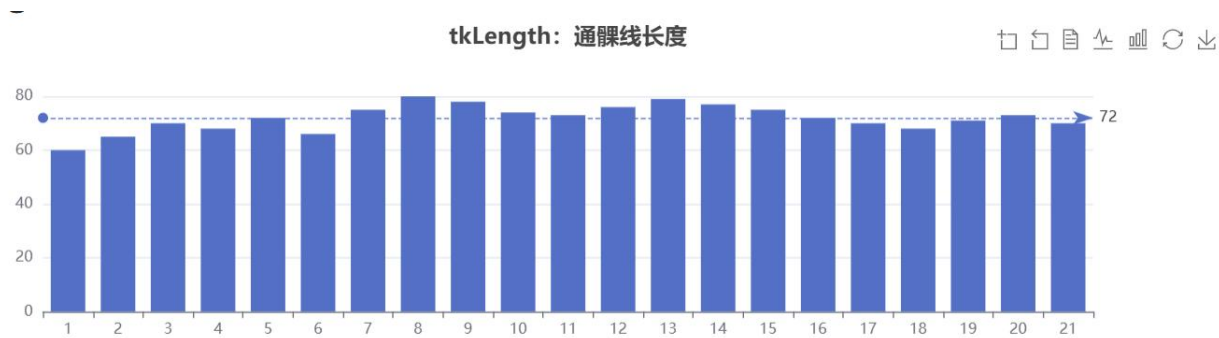


图 6.9 tkLength 通髌线长度

选择髌骨稳定性方法和柱形图后，即可显示出对于该方法测量下胫骨结节到股的距离的柱状图。不同稳定性方法（如方法 A、方法 B、保守治疗）下，胫骨结节-股骨距离的数值差异可通过柱形高度清晰呈现。髌骨稳定性方法是分类变量（非连续型），柱状图是展示分类数据对比的标准选择。通过参考线（如正常解剖距离范围）快速定位异常治疗结果。

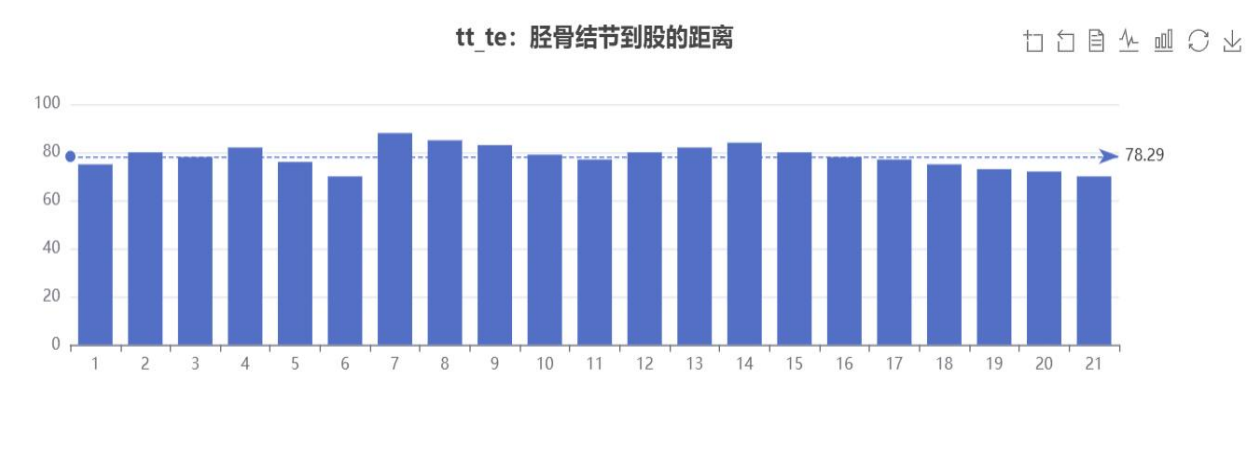


图 6.10 tt_te 胫骨结节到股的距离

选择髌骨稳定性方法和折线图后，即可显示出对于该方法测量下胫骨结节到股的距离的折线图。

折线图能够清晰不同患者的胫骨结节到股滑车沟的距离的动态变化趋势，这对于评估该方法术后恢复效果或康复训练进展至关重要，我们通过连续的线条可以直观看出距离参数是否逐渐趋近正常解剖范围。

其次，折线图特别适合对比多组人员的数据差异，通过比较线条的斜率、波动情况和最终收敛位置，可以直接判断不同方法的疗效优劣和稳定性。再者，折线图因为其表现特点可以快速发现数据点的异常波动，如术后距离值高于上限可能提示该患者髌骨高位，而突然下降则可能暗示该患者过度矫正风险。

最后，这种图表类型还支持多维数据分析，如叠加误差线显示数据离散程度，或整合其他相关参数曲线进行协同分析，为制定个性化治疗方案提供更全面的可视化依据。

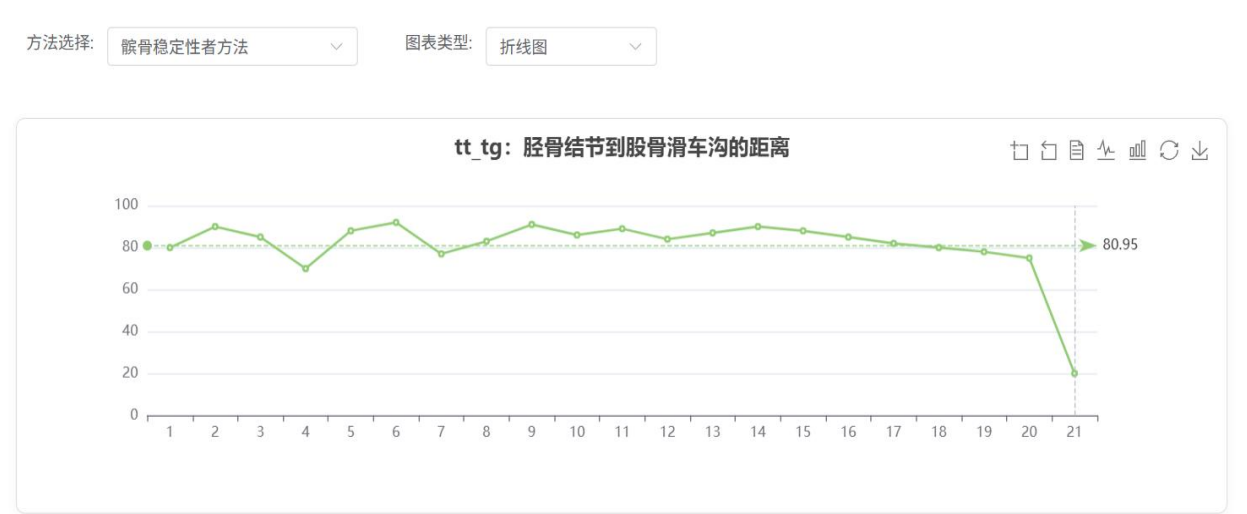


图 6.11 tt_tg 胫骨结节到股骨滑车沟的距离

选择髌骨稳定性方法和折线图后，即可显示出 $tt_tg/tkLength$ 两者之比的折线图。折线图对连续性数据的呈现比柱状图更具连贯性，避免了离散时间点的割裂感，使用户能够整体把握、胫骨结节到股的距离以及胫骨结节到股滑车沟的距离与通髌线长度之比的演变规律。



图 6.12 $tt_tg/tkLength$ 两者之比

选择髌骨稳定性方法和折线图后，即可显示出不同患者的 tt_te 胫骨结节到股的距离的折线图。折线图可以显示随时间（根据常用比例设置）或者采样人员而变化的连续数据，因此非常适用于显示在相等时间间隔下数据的趋势。

在折线图中，类别数据沿水平轴均匀分布，所有值数据沿垂直轴均匀分布。下图中横轴即为 21 组患者，数轴即为胫骨结节到股的距离的数值。通过折线图的方式反映不同患者的 tt_te 数值图，同时引入平均值使得极为容易的判断出患者偏离均值的严重程度。

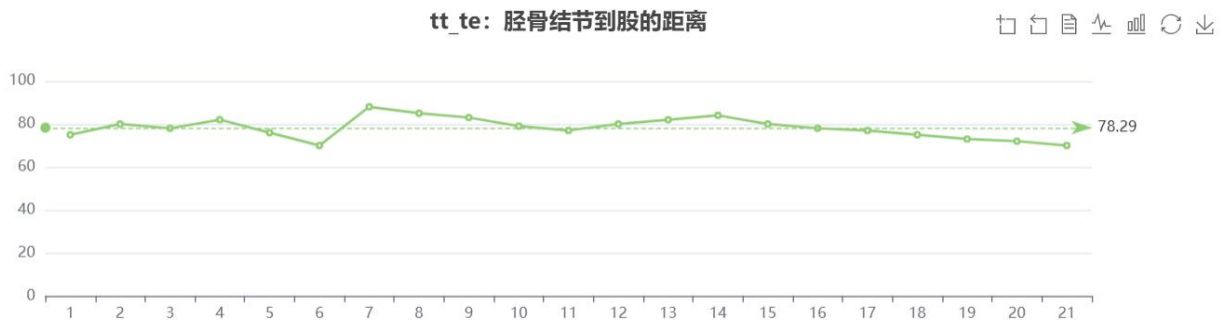


图 6.13 tt_te 胫骨结节到股的距离

选择髌骨稳定性测量方法，图表类型选择雷达图后即可展示对应方法下的几种测量参数的雷达图。包括了胫骨结节到股滑车沟的距离、通髌线长度、胫骨结节到股的距离以及胫骨结节到股滑车沟的距离与通髌线长度之比，胫骨结节到股的距离等参数。可以看出不同参数在在统一维度下的展示程度。

雷达图主要用于比较多个变量在不同维度上的表现，展示各个变量之间的相对关系。用来分析这些多个变量数据的关联和趋势

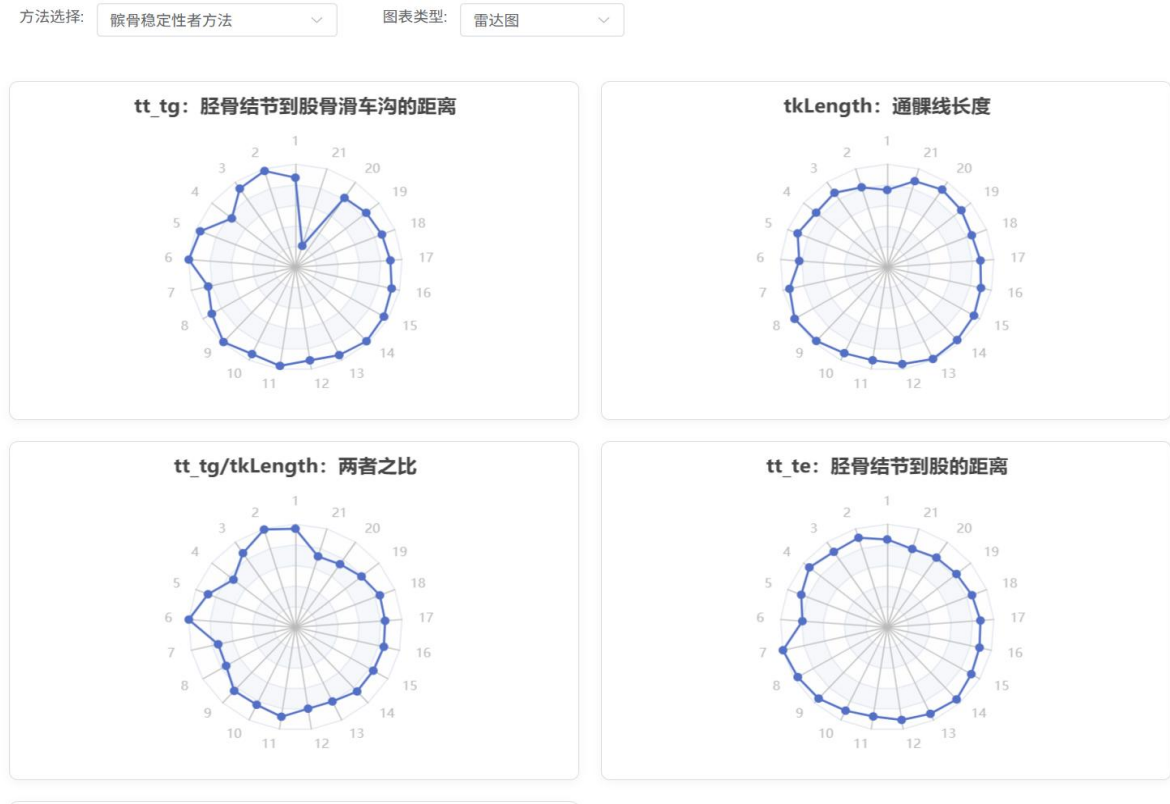


图 6.14 骸骨稳定性方法 雷达图展示

选择骸骨稳定性方法和雷达图后即可展示对应方法下的几种测量参数的散点图

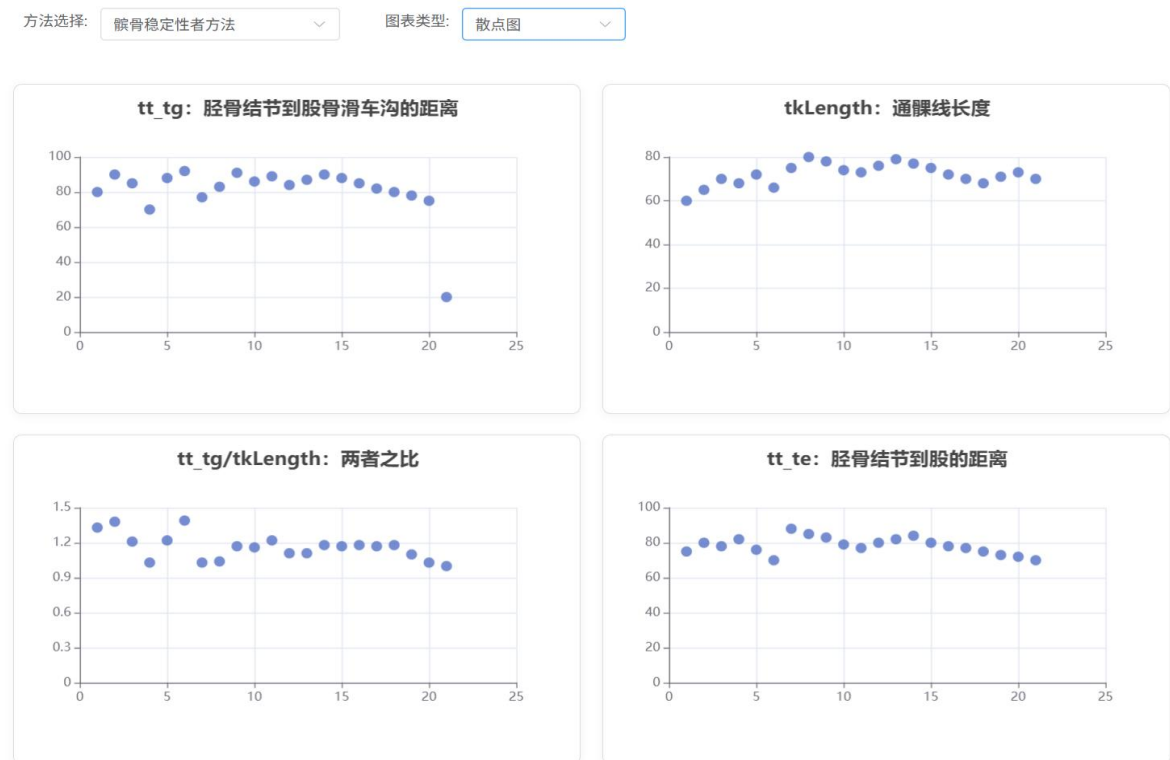


图 6.15 骸骨稳定性方法 散点图展示

选择髌骨角度测量及雷达图后，即可出现髌骨角度测量的雷达图。雷达图在髌骨角度测量分析中展现出非常的优势，主要体现在其直观的形态和多维数据整合能力展示特性。与传统的柱状图或折线图的进行比较，雷达图可以将前倾角（FAA）、外展角（LDFA）、颈干角（NSA）等多个关键骨科测量参数同时呈现在一个闭合的坐标系中，形成清晰可辨的几何轮廓。这种可视化方式使用户能够很直观的判断各角度参数的协调性，快速识别异常模式。例如，当某位患者的髌骨角度测量数据在雷达图上呈现明显不对称或超出正常参考范围时，可以直观反映出髌内翻、外翻或旋转畸形等解剖结构异常，这种整体性的形态判断是其他图表类型难以实现的。

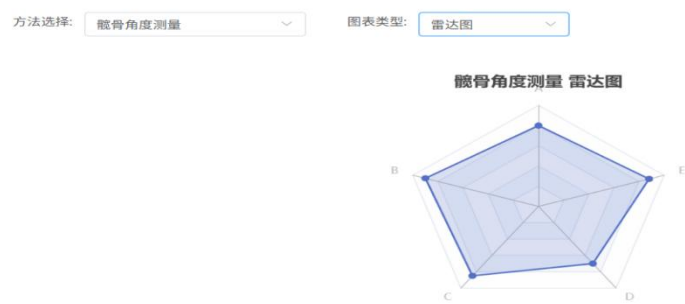


图 6.16 髌骨角度测量雷达图

6.2.3 数据分析模块

以当我们选择髌骨稳定性者方法，同时选择柱状图为基础，对 tt tg 胫骨结节到股骨滑车沟的距离做数据分析

表 5 图表说明

测试用例	期望结果	实际结果	图例编号
平均值虚线	出现对应数据	通过测试	图 6.17 6.18
正常区域	出现对应数据	通过测试	图 6.17 6.18
异常提示	出现对应数据	通过测试	图 6.17 6.18

表 6 统计详情数据

测试用例	期望结果	实际结果	图例编号
均值	出现对应数据	通过测试	图 6.17 6.18
方差	出现对应数据	通过测试	图 6.17 6.18
最大值	出现对应数据	通过测试	图 6.17 6.18
最小值	出现对应数据	通过测试	图 6.17 6.18
标准差	出现对应数据	通过测试	图 6.17 6.18
标准差	出现对应数据	通过测试	图 6.17 6.18

表 7 统计分析数据

测试用例	期望结果	实际结果	图例编号
超过平均值连续段数	出现对应数据	通过测试	图 6.17 6.18
低于平均值连续段数	出现对应数据	通过测试	图 6.17 6.18
连续上升最大数	出现对应数据	通过测试	图 6.17 6.18
连续下降最大数	出现对应数据	通过测试	图 6.17 6.18
正常数	出现对应数据	通过测试	图 6.17 6.18
临界数	出现对应数据	通过测试	图 6.17 6.18
异常数	出现对应数据	通过测试	图 6.17 6.18

当我们对下图中柱状图进行分析就可以看出 tt - tg 胫骨结节到股骨滑车沟距离的平均值为 80.95 ，说明在本次研究样本中，该距离的一般水平处于此数值 。方差为 217.00 ，标准差为 14.73，表明数据离散程度相对较大，即样本中各测量值之间的差异较为明显。

最大值为 92 ，最小值为 20 ，二者差距较大，进一步体现了样本数据的离散特征 ，提示在研究对象中，胫骨结节到股骨滑车沟的距离存在显著个体差异。

统计分析中显示异常数为 1 ，说明在所有测量数据中，有 1 个数据点偏离正常范围较多，可能是由于测量误差、个体特殊解剖结构等原因导致，需进一步排查确认，以避免对整体研究结果产生较大干扰 。



图 6.17 tt tg 胫骨结节到股骨滑车沟的距离数据分析

我们对髌骨长度的统计分析结果通过三种形式与图表展示有机结合，首先是图表标注：在 ECharts 配置中，使用 markLine 组件标注平均值参考线，markArea 组件显示均值±标准差的范围区间。其次是独立统计面板：在图表下方以表格形式展示完整统计量，采用响应式设计适配不同屏幕尺寸。最后是交互提示：当鼠标悬停在数据点上时，tooltip 不仅显示当前值，同时提示相对于均值的偏离程度（如"高于均值 1.2 个标准差"）。

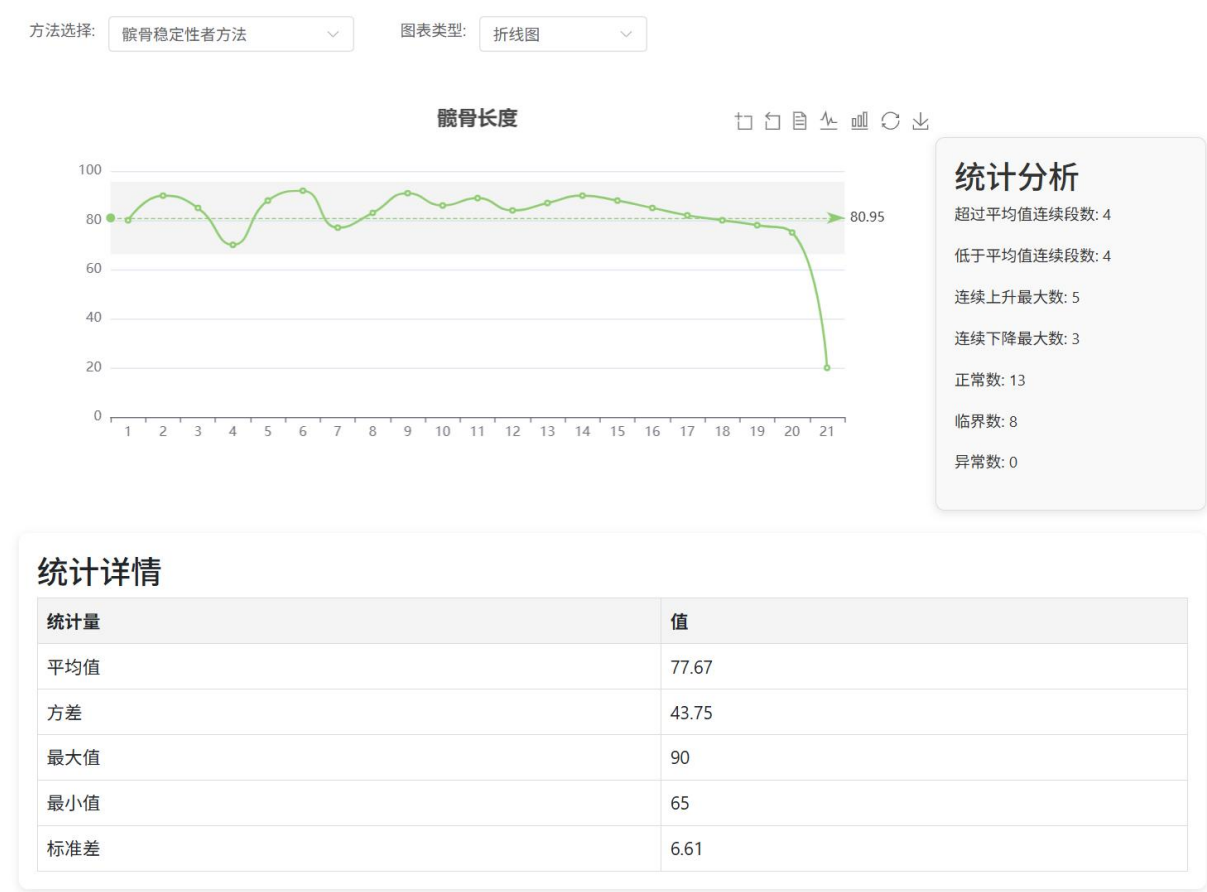


图 6.18 髌骨长度数据分析

6.2.4 数据导入模块

表 8 批量数据导入模块测试

测试用例	期望结果	实际结果	图例编号
上传文件	正常上传文件，且文件内容与上传的保持一致	通过测试	图 6.19 6.20
图表展示	绘制相关柱状图，数据与文件一致	通过测试	图 6.21

我们除过从数据库中直接读取数据以外，为了系统的适用性，我们也可以进行批量的数据读入，首先在数据导入页面中，点击上传文件，然后选择我们需要上传的骨科测量数据的文件进行上传。

数据导入

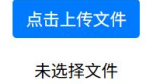


图 6.19 点击上传按钮进行文件上传

如下图为我们需要进行上传的源文件，该文件主要内容为胫骨的五种基本形态参数：有胫骨角(TSA)、胫骨内翻角(TVA)、胫骨上角(PTA)、胫骨平台角(TPA)和胫骨后倾角(PTS)，主要包含 20 组数据。

	A	B	C	D	E	F
1	编号	TSA(°)	TVA(°)	PTA(°)	PTS(°)	TPA(°)
2	1	87.2	3.1	88.5	8.2	89.8
3	2	88.1	2.8	89.2	7.9	90.1
4	3	86.9	3.3	88	8.5	89.5
5	4	87.8	3	88.8	8.1	89.9
6	5	88.5	2.7	89.6	7.8	90.4
7	6	86.5	3.5	87.7	8.7	89.2
8	7	87.3	3.2	88.4	8.3	89.7
9	8	88.2	2.9	89.3	8	90.2
10	9	87	3.4	88.1	8.6	89.6
11	10	87.9	3.1	88.9	8.2	90
12	11	88.6	2.6	89.7	7.7	90.5
13	12	86.8	3.6	87.9	8.8	89.3
14	13	87.4	3.3	88.5	8.4	89.8
15	14	88.3	2.8	89.4	7.9	90.3
16	15	87.1	3.5	88.2	8.7	89.7
17	16	88	3	89	8.3	90.1
18	17	88.7	2.5	89.8	7.6	90.6
19	18	86.7	3.7	87.8	8.9	89.4
20	19	87.5	3.4	88.6	8.5	89.9
21	20	88.4	2.9	89.5	8	90.4

图 6.20 源文件数据

当我们将 20 组数据以五种基形态参数的方式导入成功后，即可显示出这五组数据的合并柱状图，不同参数以不同颜色进行表示区分，判断不同参数的数值差异。比如蓝色表示 TSA 胫骨角度，红色表示胫骨内翻角(TVA)、绿色表示胫骨上角(PTA)、紫色表示胫骨平台角(TPA)和黄色表示胫骨后倾角(PTS)

数据导入

点击上传文件

已选择文件：胫骨基本形态参数.xlsx

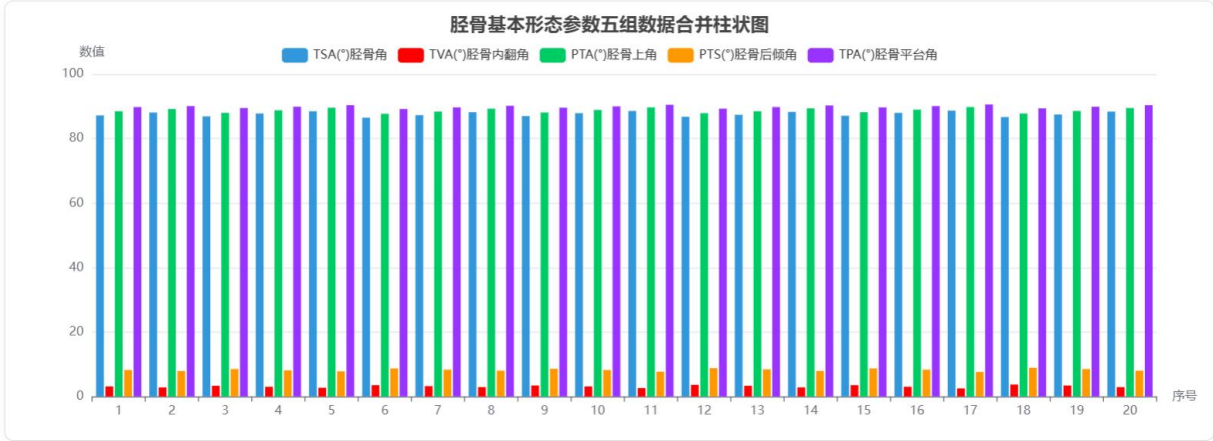


图 6.21 数据导入成功图表显示

结 论

本研究设计并实现的骨科测量数据可视化及分析系统，成功突破传统二维模型展示的技术瓶颈，构建了一个高度交互的三维骨骼模型可视化平台。系统以四个核心模块为架构基础：模型渲染模块采用前沿图形渲染技术，实现骨骼模型的高精度三维展示；模型参数设计模块支持多维度数据自定义，满足个性化研究需求；模型做点做线做面模块赋予用户直接操作骨骼结构的能力；模型信息展示模块实时反馈模型状态与坐标数据，强化人机交互体验。各模块协同运作，为医学从业者提供了一个全方位、多层次的骨骼结构研究平台，显著提升了数据解读效率与分析准确性。

在技术创新层面，本系统具备显著的领先性与实用价值。其一，系统采用行业主流技术框架，确保服务的高效性、稳定性与安全性，能够适配各类复杂的骨骼模型可视化场景；其二，通过整合分散功能模块，打造一体化操作平台，用户得以在同一界面完成从模型浏览、参数调整到信息分析的全流程操作，极大简化 workflow；其三，基于对用户行为的深度分析，系统实现智能交互反馈，显著优化用户体验，提升用户使用粘性与满意度。这些创新成果为骨科医学研究与临床实践提供了新的技术路径，对推动骨骼三维模型设计领域的发展具有积极意义。

尽管系统已取得阶段性成果，但仍存在优化空间。未来研究将聚焦以下方向：一是拓展骨骼模型功能特性，优化用户界面设计，降低操作复杂度；二是探索前沿图形技术（如光线追踪、实时渲染算法）的应用，提升模型视觉表现力；三是深化模型编辑功能，增强交互维度；四是优化后端架构，提升大数据处理与复杂模型计算的效率。该系统在生物医学研究、医学教育、手术模拟规划等领域潜力巨大，随着技术的持续迭代，有望为多学科交叉应用提供更强大的技术支撑，推动骨科医学数字化、智能化发展进程。

参考文献

- [1]郝晓慧,王宏志,韩博超,等.儿童青少年骨龄发育提前现状及其与体重指数及骨密度关系[J].公共卫生与预防医学,2025
- [2]滕翠,李容.基于Web的客户端脚本攻击的实践研究[J].网络安全技术与应用,2025,(05)
- [3]丁子木,刘美彤,韩梦杰,等.Vue框架中的MVVM思想的实践与优化[J].电脑编程技巧与维护,2025,(04)
- [4]赵新平.MySQL数据库在高并发Web系统中的优化技术[J].软件,2025,46(03)宋文凯,王丽萍,荆巍巍.基于Vue框架的多源WEB前端页面数据可视化方法[J].电子设计工程,2025,33(06)
- [5]陈思楠,陈吉平,黄红倩,等.基于ECharts的医疗数据交互式可视化研究[J].现代信息科技,2024,8(14)
- [6]宋佳倩,黄泽阳.基于ECharts的设备点位分析设计与实现[J].现代计算机,2024,30(13)
- [7]吴文俊,刘莉,李俊,等.基于Django的医学3D打印数据库系统的设计与实现[J].软件,2024,45(06).
- [8]郭艺辉.医学图像信息可视化关键技术研究探讨[J].电子元器件与信息技术,2021,5(05)
- [9]李长红,贺永进,王飞,等.螺旋CT结合图像后处理对上颈椎椎弓根的解剖学测量数据及临床意义探究[J].山西医药杂志,2016,45(09)
- [10]马晟皓.基于Python语言的Excel数据导入程序设计与实现[J].电脑编程技巧与维护,2025,(05)
- [11]郑琦,严梓桢,王飞,等.基于CiteSpace的全髋关节置换术患者出院准备度的可视化分析[J].护理与康复,2025,24(05):58-61.
- [12]吴瑶瑶,盛松.基于二、三维GIS的城市动态可视化[J].北京测绘,2025,39(04)
- [13]陈华.Python在大数据分析中的可视化应用[J].软件,2025,46(02)

- [14]王远智,王礼宁,郭杨,等.骨质疏松性骨折风险预测模型研究进展[J].中国骨质疏松杂志,2024,30(10)
- [15]张弛,吴海龙,张帅,等.后凸 Cobb's 角、BMD、PINP 对骨质疏松性椎体压缩性骨折患者 PKP 术后再发骨折风险的预测价值分析[J].疑难病杂志,2024,23(04)
- [16]曹惠茹,袁跃钊,周子健,等.基于 Vue 与 ECharts 的粮食价格可视化系统设计与实现[J].电脑编程技巧与维护,2024,(01)
- [17]陶广义,王琳梓,杨彬,等.人工智能在脊柱畸形领域研究热点的可视化分析[J].中国组织工程研究,2024,28(30)
- [18]阮晓龙,李朋楠.基于 Web 的大文件高效上传方法[J].计算机系统应用,2020,29(03)
- [19]王莉敏,梁正和,段全锋.基于 HTML5 大文件断点续传的实现方案[J].计算机与现代化,2016

致 谢

2025 年，我 21 岁，完成了我本科阶段的求学生涯。

行文至此，落笔为终。2021 年秋到 2025 年夏的故事就此落下帷幕，是结束也是开始。

感谢这四年来所有的相遇和经历，好的不好的都有它存在的意义。感谢我的论文指导老师在论文期间给予我的关注与指导，让我的毕业论文顺利完成。感谢求学路上的每一位老师，谢谢你们的倾囊相授，涓涓师恩，铭记于心。感谢我的父母家人朋友，给了我无限的爱与包容，托起我的理想与远方。最后，感谢走得很慢但一直在向前的自己。

“追风赶月莫停留,平芜尽处是春山”.....