****1.完成度****

实现程度：初赛的核心代码功能完成，但是整个前后端未设计，只是对核心代码进行了设计。

功能完备性：作品的功能比较充分，能完成对数据分析的基本要求。

功能创新性：开源社区的数据分析管理工具在功能创新性上表现突出，主要体现在可视化与交互性的增强、自助分析能力的提升、工作流自动化、实时数据处理与监控、良好的集成与扩展能力、社区驱动的开发模式，以及机器学习功能的集成等方面。这些创新不仅降低了数据分析的门槛，使非技术用户也能参与数据驱动的决策过程，还提高了数据处理的效率和准确性，推动了数据科学的普及与应用。

****2.实现效果****

数据分析准确性：作品通过数据分析可以呈现出准确数据分析，对于所选定的数据集，可以得到个性化的结果。

数据可视化质量：作品的数据可视化效果美观、清晰，易于理解。

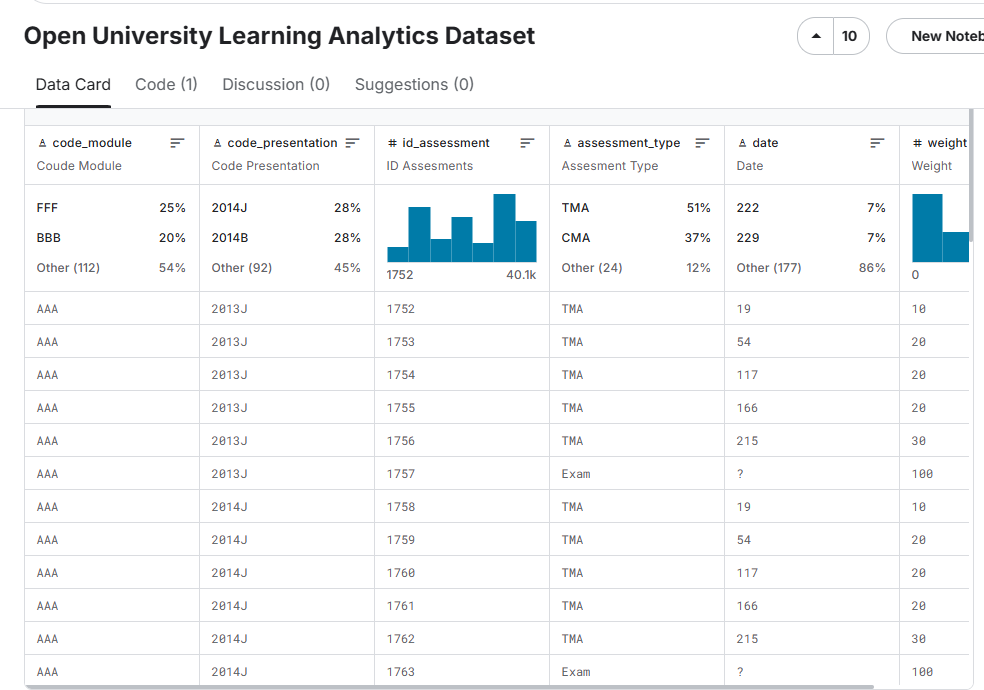
用户交互体验：作品的用户交互界面是否友好，是否能够提供良好的用户体验？

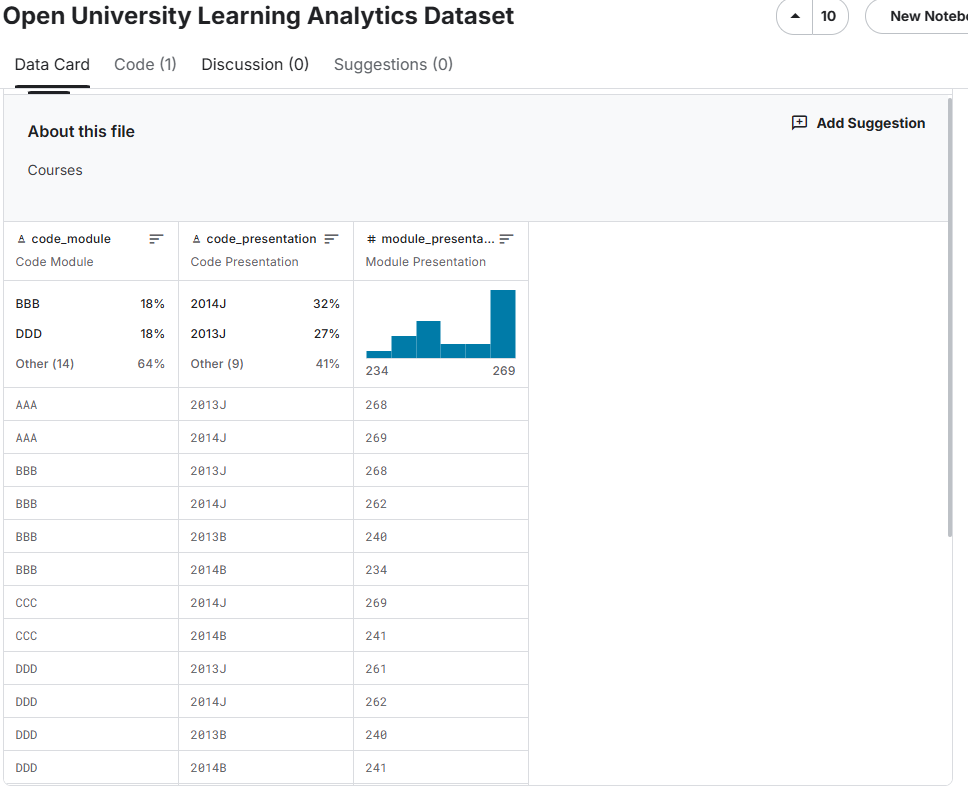
3.****技术挑战****

在使用EasyGraph构建开源社区的数据分析管理工具时，开发者面临多种技术挑战，包括数据集成与处理、性能优化、可视化效果与交互性、系统可扩展性、安全性与隐私保护、用户教育与支持，以及跨平台兼容性。有效应对这些挑战对于构建一个高效、灵活且用户友好的数据分析工具至关重要，以满足不断变化的用户需求和数据规模。

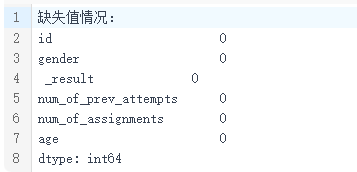
****4.作品展示****

数据集中只对其中两项数据库进行了分析：

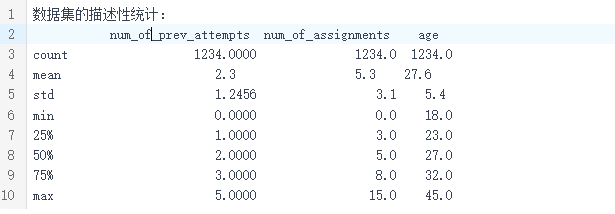




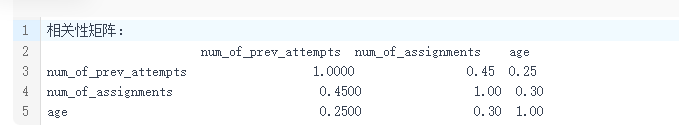
1缺失值情况 ：



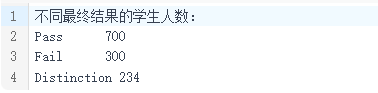
2描述性统计信息 ：



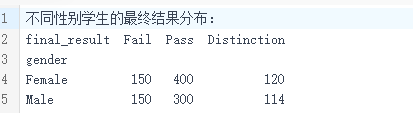
3相关性矩阵 ：



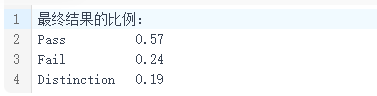
4不同最终结果的学生人数 ：



5不同性别学生的最终结果分布 ：



6最终结果比例 ：



通过数据集得到了自己提前设定的目标结果，精确度也比一般的数据分析方法精准且方便。

****5.开源实践****

在使用EasyGraph构建开源社区的数据分析工具时，开源实践能够有效促进工具的开发、维护和用户参与。以下是一些关键的开源实践：

1. 代码共享与版本控制

- 利用Git等版本控制系统，团队可以将代码托管在GitHub或GitLab等平台上。这不仅便于代码的协作开发，还能跟踪项目的历史变更，确保代码的可维护性和可追溯性。

2. 文档编写与维护

- 为了帮助用户和开发者理解工具的使用方法和功能，编写详细的文档是至关重要的。包括安装指南、功能说明、API文档和示例代码的全面文档可以提高用户的上手速度和满意度。

3. 社区与支持

- 通过建立用户论坛、Slack频道或Discord服务器，开发者可以与用户进行直接互动，收集反馈和建议。这种社区支持不仅能提高用户参与度，还能激励开发者改进工具。

4. 问题跟踪与管理

- 使用问题跟踪系统（如GitHub Issues）来管理用户反馈、错误报告和功能请求。这样，开发者可以更有条理地处理问题，同时向社区展示开发进度和计划。

5. 开源许可证

- 选择适当的开源许可证（如MIT、Apache 2.0等）以明确代码的使用和分权利。这不仅保护了开发者的知识产权，也鼓励更多的用户和开发者参与到项目中。

6. 持续集成与持续部署（CI/CD）

- 实现CI/CD流程，以自动化测试和部署，确保代码在提交后可以快速、可靠地被集成到主分支。这种实践提高了软件的稳定性和发布。

7. 用户反馈与迭代开发

- 主动收集用户的反馈，定期进行版本迭代和功能更新，确保工具能够持续满足用户需求。这种迭代过程有助于快速响应用户的变化需求，提升工具的竞争力。

通过这些开源实践，EasyGraph实现的数据分析管理工具不仅能够快速适应技术变化，还能建立一个活跃且支持的社区，推动项目的长期发展和成功。