

# 《Python程序设计基础》程序设计作品说明书

题目：数据可视化

学院：21计科04

姓名：左莉

学号：B20210202307

指导教师：周景

起止日期：2023.11.10-2023.12.10

## 摘要

数据可视化指的是通过可视化表示来探索和呈现数据集内的规律。它与数据分析紧密相关，而数据分析指的是使用代码来探索数据集内的规律和关联。数据集既可以用一行代码就能装下的小型数值列表，也可以是数万亿字节、包含多种信息的数据。有效的数据可视化不仅仅是以漂亮的方式呈现数据。重要的是，通过以简单而引人注目的方式呈现数据，让观看者能够明白其含义：发现数据集中原本未知的规律和意义。

主要内容：使用Plotly生成交互式图形，处理常见类型和格式的数据集（比如：json,csv）绘制散点图和折线图，可视化获取的网站信息数据。关键词：Plotly 可视化

## 第1章 需求分析

需求分析：

1. 系统目标：开发一个可视化系统，通过使用Plotly库来模拟扔骰子、绘制天气数据折线图、制作全球地震散点图，并使用GitHub的Web API获取仓库数据，并将这些数据进行可视化展示。
2. 用户群体：该系统可以被广泛应用于数据分析师、研究人员、学生等用户群体，以及对数据可视化感兴趣的任何人。
3. 功能需求：

- 模拟扔骰子：实现通过Plotly库模拟扔骰子的功能，可以设置投掷次数和骰子面数，并显示投掷结果的直方图或饼图。
- 绘制天气数据折线图：读取CSV文件中的天气数据，并使用Plotly库绘制折线图，展示不同时间段内的天气变化情况。
- 制作全球地震散点图：通过获取地震数据源，并使用Plotly库绘制散点图，展示全球地震的位置和强度。
- 使用GitHub的Web API获取仓库数据：通过调用GitHub的Web API接口，获取指定仓库的相关信息，如提交次数、贡献者数量等。
- 使用Plotly可视化仓库：将获取到的GitHub仓库数据使用Plotly库进行可视化展示，例如绘制柱状图表示提交次数或绘制饼图显示贡献者数量。 解决的问题：

#### 4. 解决的问题：

- 提供了一个交互式的界面，让用户能够通过模拟扔骰子来进行随机事件的模拟和分析。
- 通过绘制天气数据折线图，用户可以直观地了解不同时间段内的天气变化趋势，帮助他们进行气象分析和预测。
- 展示全球地震散点图可以帮助用户了解地震的分布情况和强度，并有助于地震研究和风险评估。
- 通过获取GitHub仓库数据，并使用Plotly可视化展示，用户可以更好地了解仓库的活动情况和贡献者信息，有助于项目管理和开发过程的监控。

## 第2章 分析与设计

### 1. 系统架构:

本系统旨在提供数据可视化功能，主要包括投掷骰子模拟、天气数据折线图和全球地震散点图的绘制。以下是系统架构的简要描述：

系统由三个主要模块组成：数据获取模块、数据处理模块和数据可视化模块。

数据获取模块负责通过Web API获取所需数据。在第三单元中，该模块使用Github的Web API进行数据获取，获取Github仓库的相关信息。

数据处理模块接收从数据获取模块获得的原始数据，并对其进行清洗、转换和处理。在第一单元中，该模块将模拟投掷骰子的结果进行计算和统计；在第二单元中，该模块将解析和处理CSV格式的天气数据，提取并整理需要的信息；在第三单元中，该模块可能需要对从Github API获取的数据进行筛选和整理，以满足后续可视化的需求。

数据可视化模块利用Plotly库，根据处理后的数据生成相应的可视化图表。在第一单元中，该模块将生成投掷骰子结果的直方图或频率分布图；在第二单元中，该模块将生成天气数据的折线图；在第三单元

中，该模块将根据Github仓库的相关信息生成相应的可视化图表，如仓库星星数量的散点图、仓库语言使用情况的饼图等。

系统架构还需要考虑数据传递的接口和通信方式。数据获取模块通过HTTP协议与Github API进行通信；数据处理模块通过接口接收原始数据，并提供处理后的数据给数据可视化模块；数据可视化模块通过接口获取处理后的数据，并利用Plotly库生成图表。

## 2. 系统模块：

- 模拟扔骰子模块：使用Plotly库模拟扔骰子的过程，并绘制直方图或饼图。
- 天气数据可视化模块：读取CSV文件中的天气数据，并使用Plotly库绘制折线图。
- 地震数据可视化模块：利用Plotly库制作全球地震散点图，并展示地震的位置和强度。
- GitHub仓库可视化模块：使用GitHub的Web API获取仓库数据，并使用Plotly库进行可视化展示。

3. 数据库的设计： 本项目涉及到的数据主要来自CSV文件和GitHub的Web API接口，因此不需要使用数据库进行存储。在后端处理数据时，可以将数据存储在内存中，以便进行数据处理和可视化展示。

## 4. 关键实现：

- 模拟扔骰子：使用Python中的random模块生成随机数来模拟扔骰子的过程，并使用Plotly库绘制直方图或饼图。
- 天气数据可视化：利用Python中的pandas库读取CSV文件中的天气数据，并使用Plotly库绘制折线图。
- 地震数据可视化：通过获取地震数据源，使用Plotly库绘制全球地震散点图，并展示地震的位置和强度。
- GitHub仓库可视化：调用GitHub的Web API接口获取仓库数据，并使用Plotly库进行可视化展示，例如仓库名称，仓库项目名称，星数。

# 第3章 软件测试

## 第一单元测试：

投掷骰子模拟测试： 验证投掷次数的合理性对于程序的影响

投掷次数不合理时（投掷次数为-100），程序的运行情况：

Traceback (most recent call last):

```
File "d:\software\Git\git_python_practice\project\fifteen\die_test.py", line 52, in <module>
    test_roll_die()
File "d:\software\Git\git_python_practice\project\fifteen\die_test.py", line 38, in test_roll_
    results = roll_die(-1000)
                ^^^^^^^^^^^^^^^^^^^
```

```
File "d:\software\Git\git_python_practice\project\fifteen\die_test.py", line 7, in roll_die
    raise ValueError("Number of rolls must be a positive integer.")
```

ValueError: Number of rolls must be a positive integer.

投掷次数合理时（投掷次数为1000），程序的运行情况：

每个面出现的次数：

面1出现了182次

面2出现了160次

面3出现了142次

面4出现了179次

面5出现了180次

面6出现了157次

第二单元的测试：

测试数据文件存在与不存在的程序运行结果是否和预期一样

数据文件存在时：

```
MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6.
plt.style.use('seaborn')
```

数据文件不存在时：

Traceback (most recent call last):

```
File "d:\software\Git\git_python_practice\project\sixteen\test.py", line 24, in <module>
    lines=path.read_text().splitlines()
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
File "D:\software\Python\Lib\pathlib.py", line 1058, in read_text
    with self.open(mode='r', encoding=encoding, errors=errors) as f:
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
File "D:\software\Python\Lib\pathlib.py", line 1044, in open
    return io.open(self, mode, buffering, encoding, errors, newline)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

FileNotFoundError: [Errno 2] No such file or directory: 'death\_valley\_2021\_full.csv'

第三单元测试:

测试URL的有效与否的运行结果是否与期待一致;

url有效:

```
Total repositories: 5
Repository: 1128zl
URL: https://github.com/1128zl/1128zl
Repository: base-c
URL: https://github.com/1128zl/base-c
Repository: c-code-practice
URL: https://github.com/1128zl/c-code-practice
Repository: git_python_practice
URL: https://github.com/1128zl/git_python_practice
Repository: gorm-MySQL
URL: https://github.com/1128zl/gorm-MySQL
```

URL无效:

Traceback (most recent call last):

```
File "d:\software\Git\git_python_practice\project\seventeen\test.py", line 35, in <module>
    test_github_repository()
File "d:\software\Git\git_python_practice\project\seventeen\test.py", line 32, in test_github_
    assert response.status_code == 200
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

AssertionError

# 结论

本项目实现了以下功能和目标：

模拟扔骰子：通过使用Plotly库，实现了对扔骰子过程的可视化模拟，并绘制了直方图，为用户提供了直观的随机分布展示。天气数据折线图：成功读取了CSV文件中的天气数据，并使用Plotly库绘制了折线图，使用户可以直观地了解天气变化趋势。全球地震散点图：利用Plotly库绘制了全球地震的散点图，展示了地震的发生位置和强度分布，为用户提供了直观的地震信息呈现。GitHub数据可视化：通过调用GitHub的Web API接口，成功获取了仓库数据，并使用Plotly库进行可视化展示，帮助用户更好地理解GitHub仓库的相关信息。

项目的不足之处及改进方向：

用户交互体验：当前项目在用户交互方面还有待改进，可以考虑增加更多的交互式组件和反馈机制，提升用户体验。数据源的多样性：当前项目主要围绕扔骰子模拟、天气数据和地震数据进行可视化，未涉及其他多样化的数据源，可以考虑扩大数据源范围，丰富可视化内容。性能优化：针对大规模数据的处理和可视化，需要进一步优化算法和代码，提升系统性能和稳定性。文档和注释：应该完善项目文档和代码注释，方便他人阅读和维护项目。

# 参考文献

1. 《Python编程从入门到实践第3版》
2. 《流畅的Python 第2版》
3. 《Python数据分析第3版》