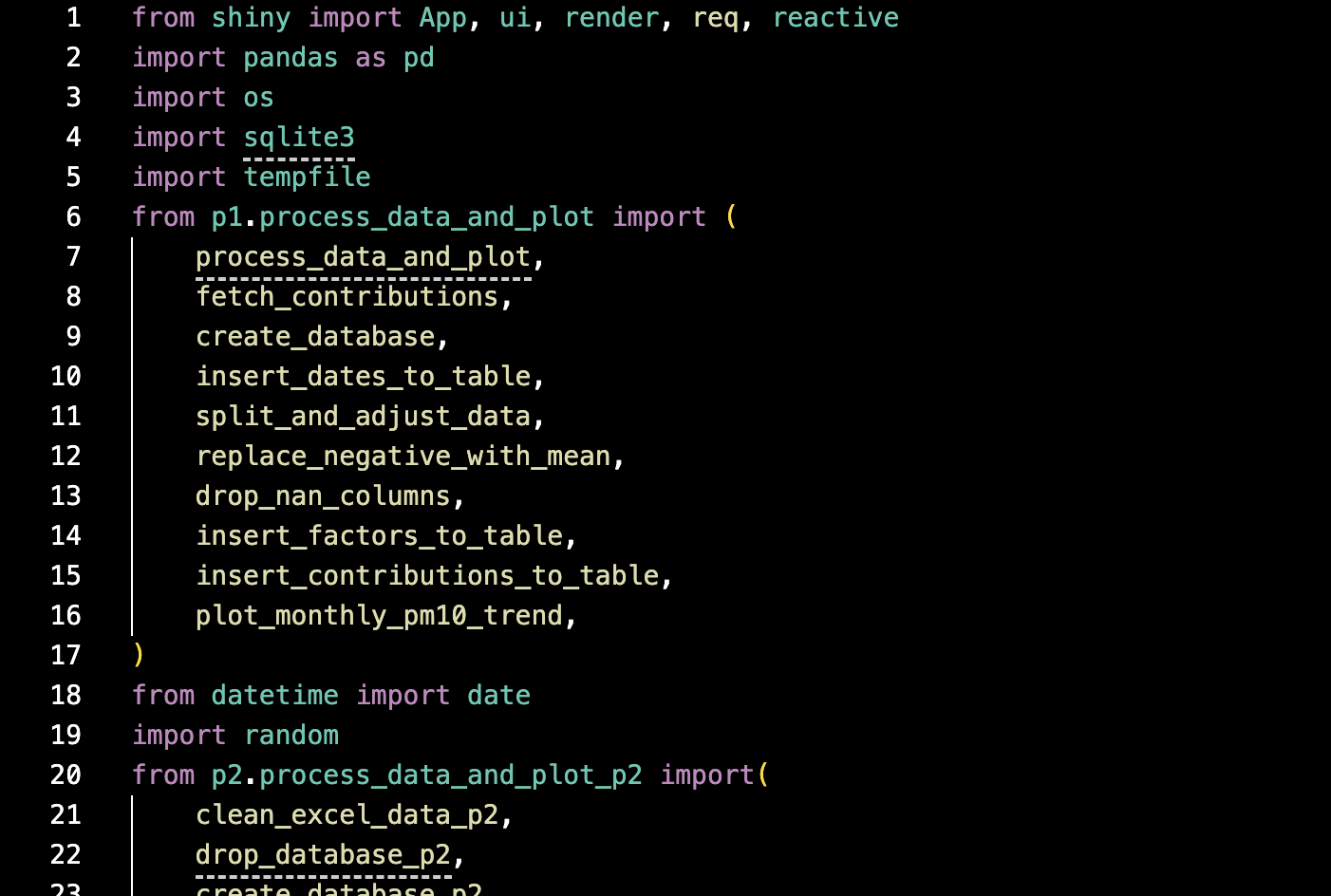
Shiny 应用开发步骤:

Shiny 主要分2部分: 第一部分是app的UI, 第二部分是server.在第一部分, 我们可以设置app的layout. 具体想要什么的样式, 就从shiny python的官网中找到 <https://shiny.posit.co/py/component>, 但是shiny python有2个不同的版本, gpt 或者kimi 给到的code 在很多时候忽略了不同版本这一点, 所以code 用不上. 这时候, 我们可以手动改一下code, 一般vs code你输入了前面几个字母, 它就会弹出你使用那个版本对应功能的code.

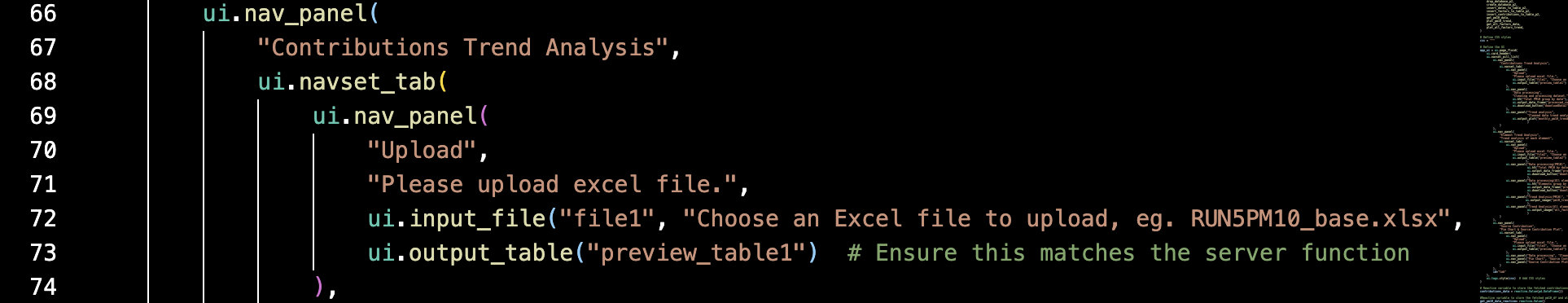
**如何导入P1/P2的外接的py code?** 在import 的section, 需要导入要用到的外接函数.  


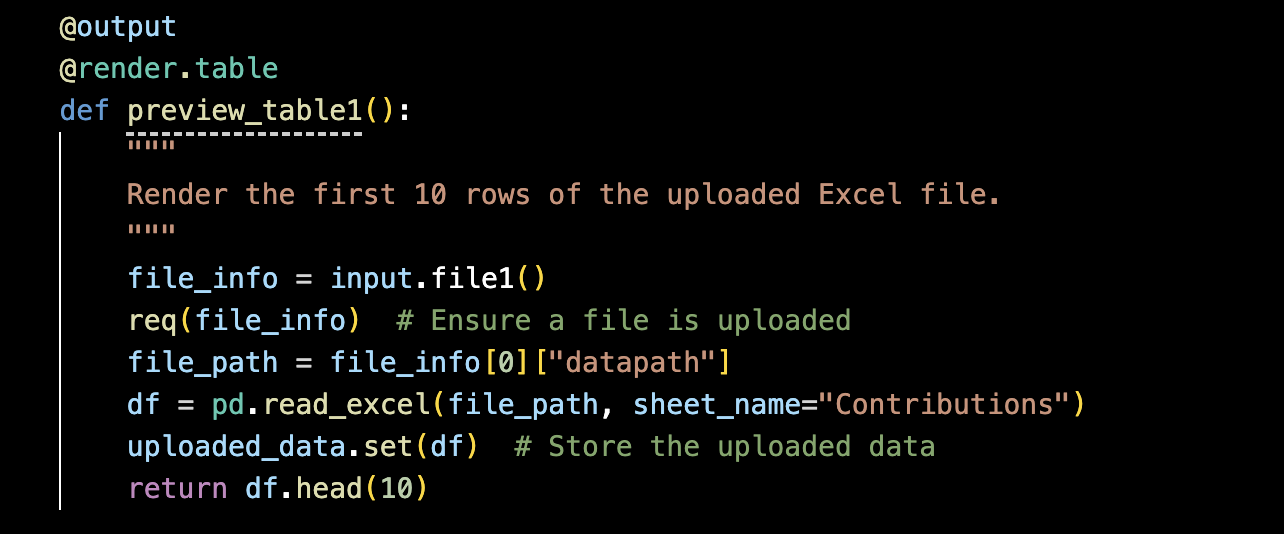
我已经做了tab 1 Contribution Trend Analysis 和 tab 2 Elements Trend Analysis.

Tab 1 处理的是 RUN5PM10\_base.xlsx 数据集.

包含panel 1) Upload, 2) Data Processing, 3) Trend Analysis

1. Upload

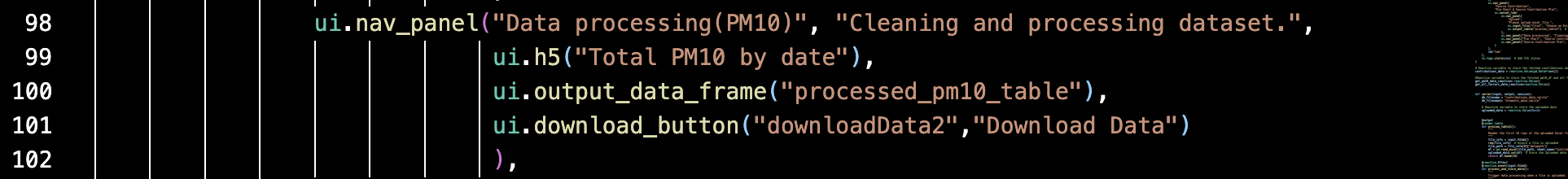
Upload对应的UI code:  


Upload 对应的server code:  


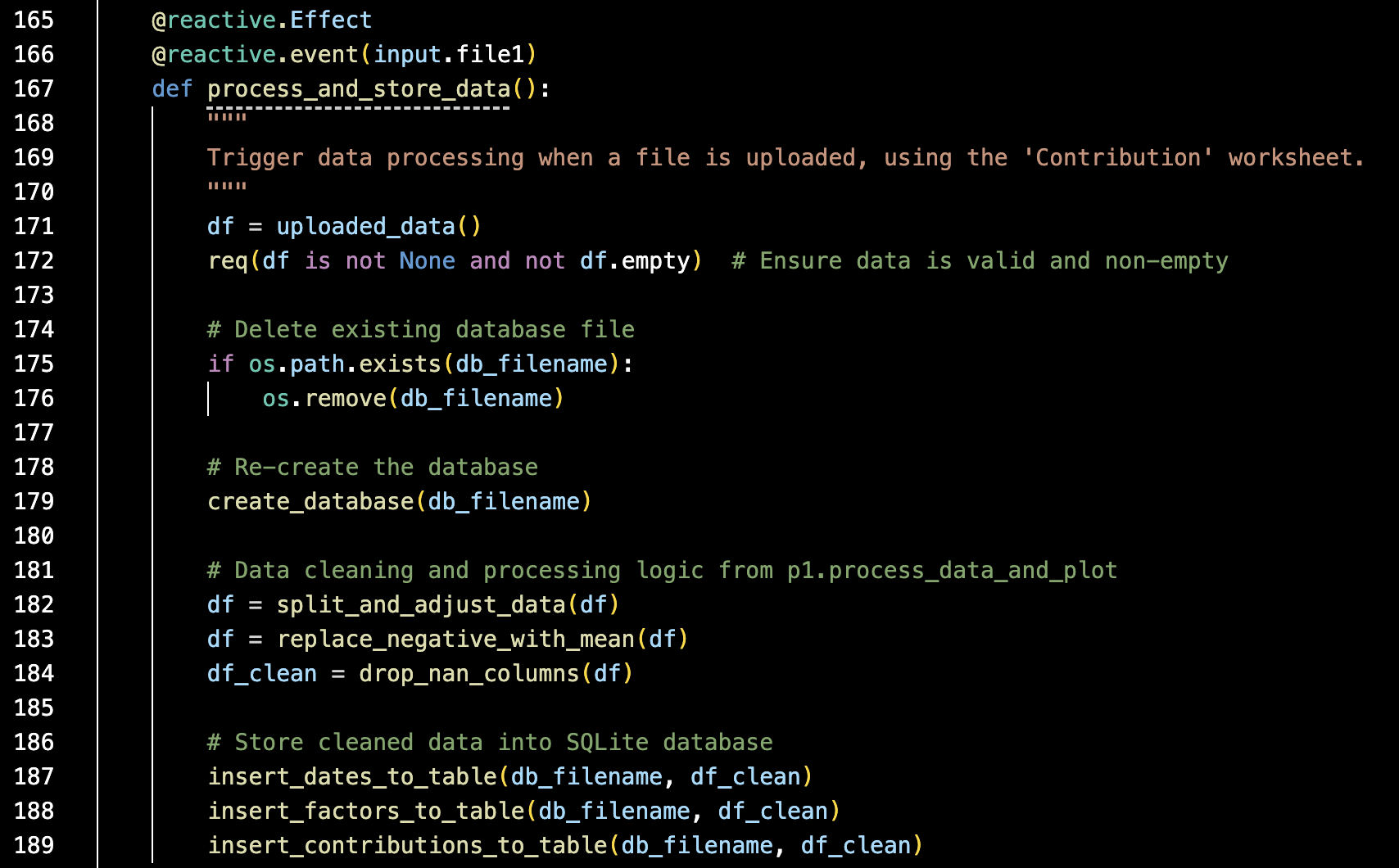
@output 和@render 是把逻辑渲染到app的指令, 如果在server写了执行的逻辑, 但是没有在app中没有反应, 就要检查是不是漏了这两个指令.

1. Data Processing

对应的UI code:



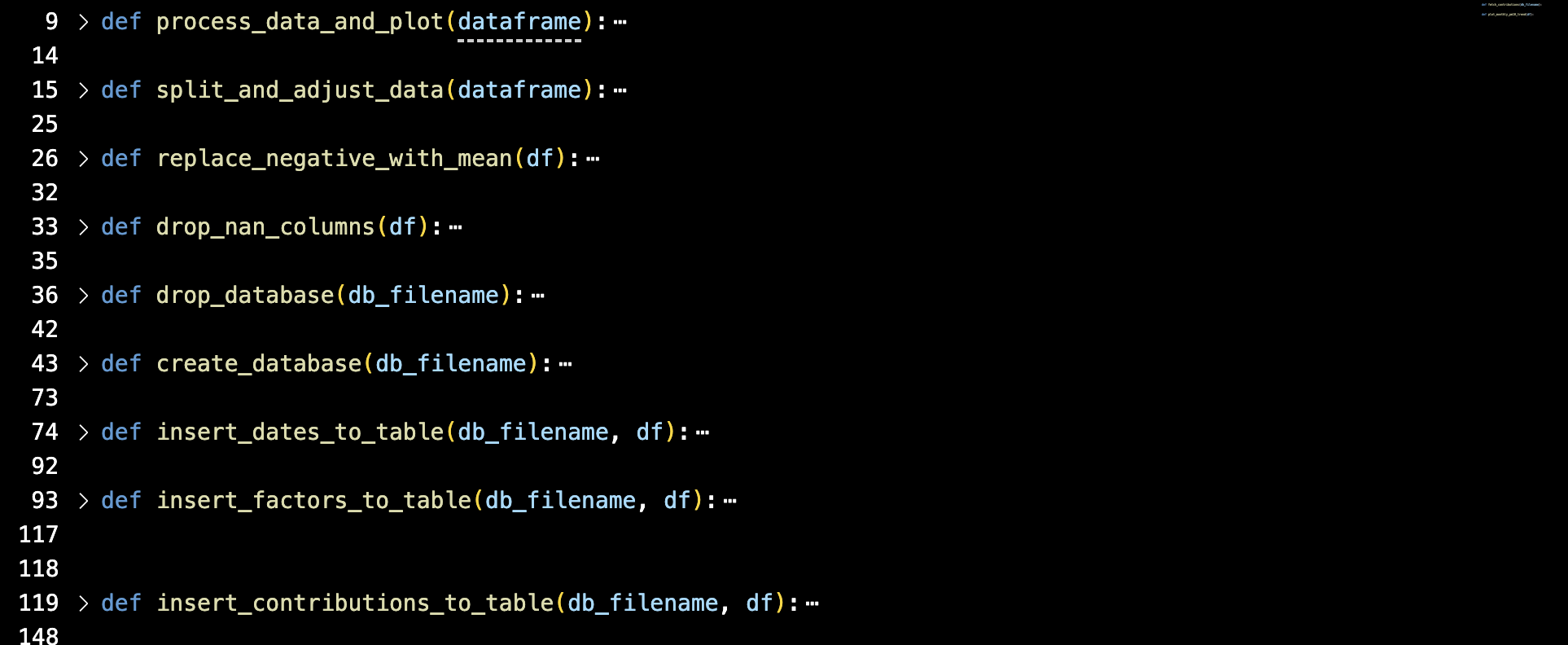
对应的server code:





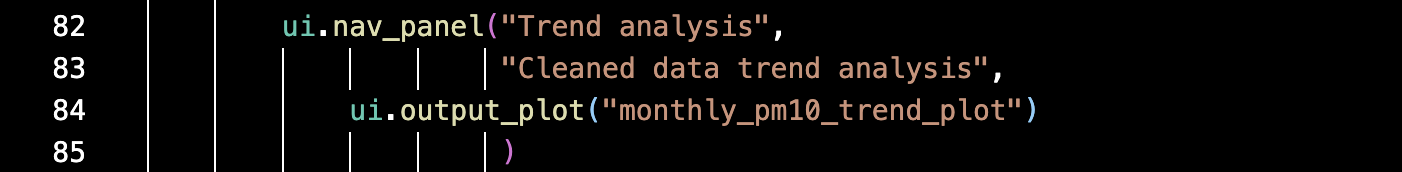
在这个server 部分, 有调用到P1的函数部件, 黄色部分的字都是调用了P1中提前预设好的函数.

P1中的函数

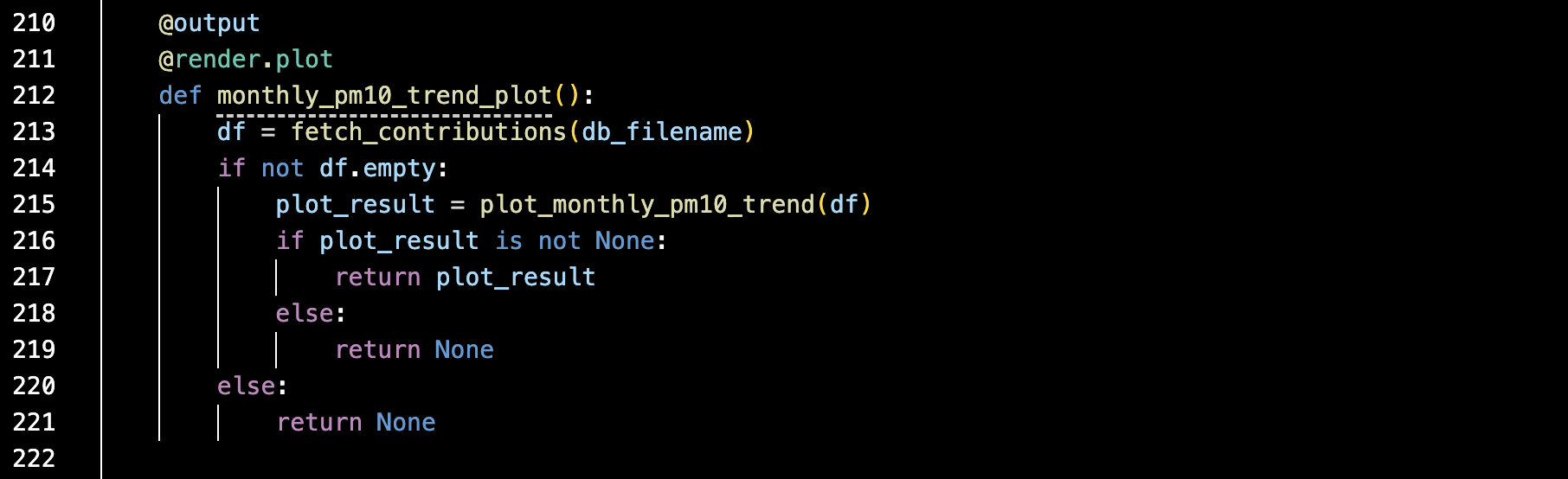


1. Trend Analysis

UI code:



Server code:



P1对应的函数  


Tab 2 处理的是 QueenStreetPM10PMFDataUncert\_May2023.xlsx 数据集.

包含panel Upload, Data processing(pm10), Data processing(All elements), Trend analysis(PM10), Trend analysis(all elements)

这部分的处理逻辑基本和Tab1 一样, 区别是视图出来的时候, tab 1出的是plot, tab2两个图出得都是image , 而且指定了缓存路径. **这个地方要再想优化的方法.**