一、多学历目前代码都处于分支：analisys-refactoring下。

多学历可以兼容单学历

二、主要文件：

1、data\_analyze.py：对于单个excel文件的产生

2、data\_calculate.py：excel里面单个sheet计算

3、formulas.py：公式类，eg：就业率、占比、五纬占比

4、result\_write.py：输出到excel文件，同时设置百分比、小数位数

5、analyze\_run.py：报表启动类

6、file\_loader.py： 读取文件类

目前 客观数据是新建了目录objective，里面只有data\_analyze.py，原本计划将data\_calculate.py也提取一份，但是与多学历有公用代码的可能性，暂时放一起。

2.1 data\_analyze.py：

以“毕业生规模和结构.xlsx报表”为例：df为原始报表数据，dict\_config 读取的配置文件（多学历根据题号解析sheetName）

1、提取相关计算要用到多列；

2、调用Calculator类产生sheet里面多内容；

class SizeAndStructureAnalyzer(DataAnalyzer):  
 *"""毕业生规模和结构"""* def \_\_init\_\_(self, df, dict\_config):  
 super().\_\_init\_\_(df, dict\_config)  
 self.\_question\_cols = '学历合并'  
  
 def analyse(self):  
 rel\_cols = CONFIG.OBJECTIVE\_BASE  
 de = DataExtractor(self.\_df, self.\_question\_cols)  
 df = de.extract\_objective\_cols(rel\_cols)  
  
 result = dict()  
 style = ObjectiveOrderStyler()  
 result['总体规模'] = ObjectiveSizeCalculator(df,  
 self.\_question\_cols,  
 styler=style).calculate()  
 result.update(common\_grp\_anaysis(df, self.\_question\_cols, ObjectiveGrpSizeCalculator, '结构'))  
 return result