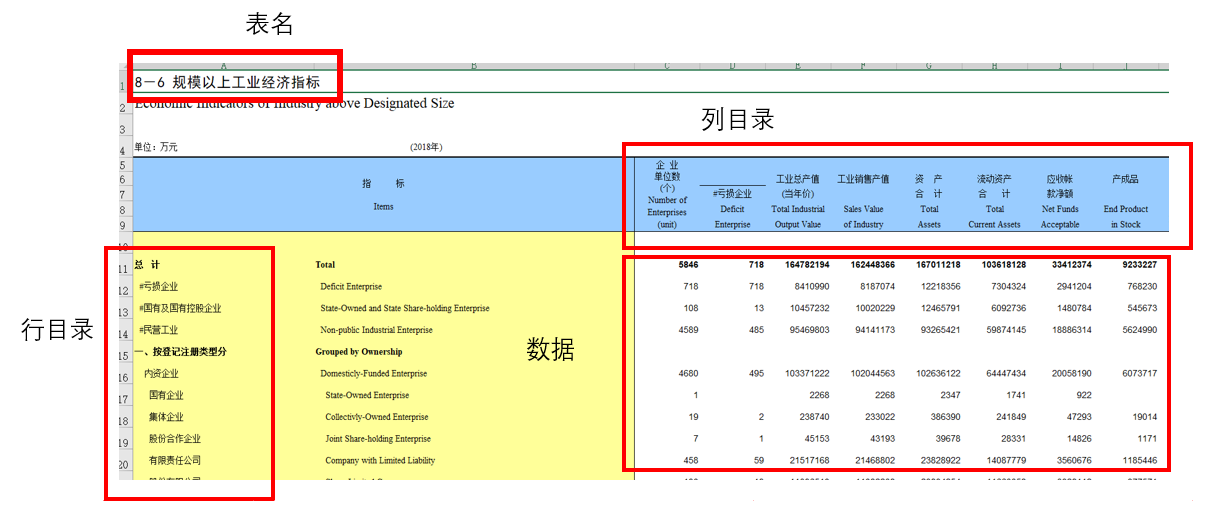
数据收集任务指引说明

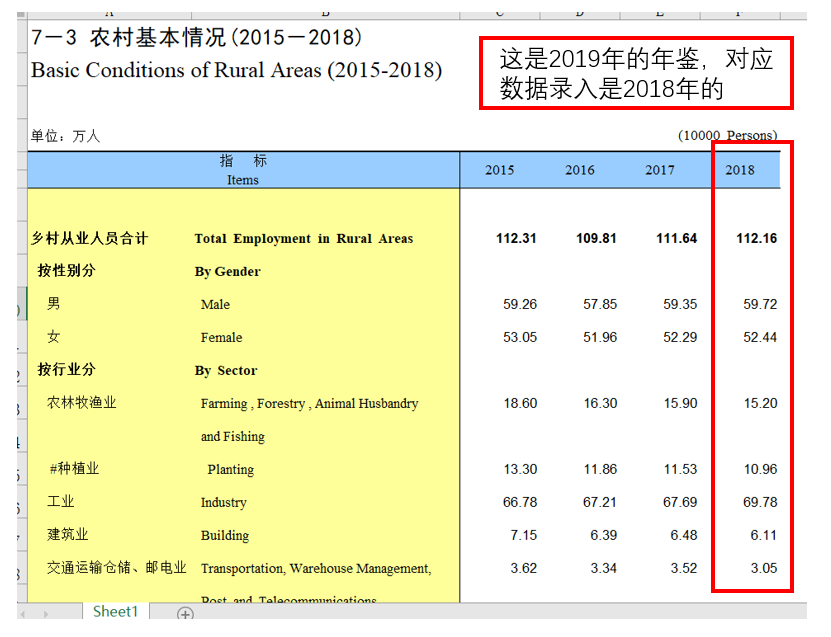
我们这个任务的主要目的就是在各个城市各个年份的年鉴中找到我们想要的数据并把它录入到excel表格中。

给大家的资料中共有8个文件，除本指引外，还有统一目录.xlsx，collect.py，modify.py，getData.py，match.py，auto.py，auto2.py，下面会对各项文件进行主要介绍，方便大家明白具体的流程细节。

**年鉴**

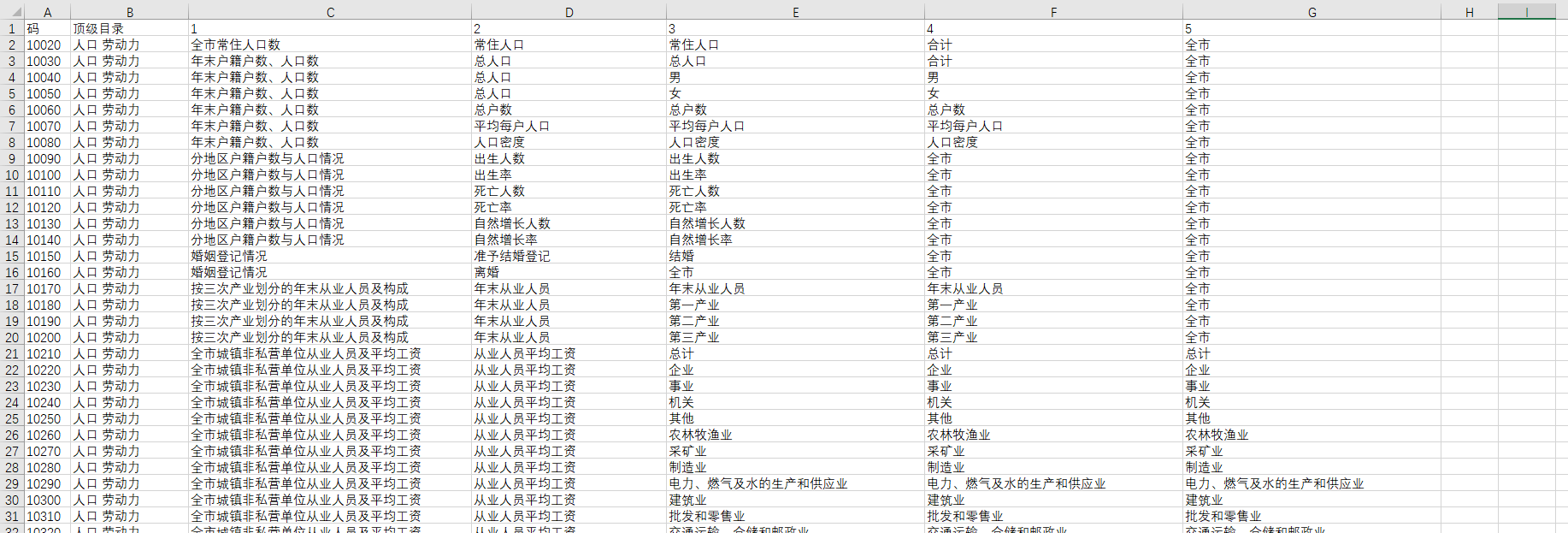
每个城市都会有统计年鉴，其主要是记录该城市上一年份的各项数据，一般包含200多个表格，下图为表格示例，我们可以看到，表格可分为：表名，行目录，列目录，数据。不是年鉴中的所有的表格数据都需要录入，我们会对照“统一目录.xlsx”中的每一行来查找年鉴中的数据并录入。需要注意的是，当年年鉴记录的是前一年的数据，如2019记录的就是2018年的数据，以此类推，故虽然名义上我们在录入当年的数据，但数据在实际录入中是去年的数据，表格的实际表达也是以上一年份的为标准的。





统一目录.xlsx

表格中的每一行代表了每一项数据的目录名称，A列为码，用于区别各项数据；B列为数据的大分类，如人口、工业等；C为年鉴中的表格的表名；D、E、F、G列对应表格中的行目录和列目录，**部分行目录和列目录下会继续细分**（这一部分的数据比较特殊，需要特别注意有没有准确录入）。以此统一目录为基准，我们需要在H列往后，录入2010-2019年的年鉴数据，H列为2010年，J列为2011年，以此类推，当所有年份的数据都录入完成后即任务完成，不一定统一目录所有的目录项都能在年鉴中找到对应的数据，如不存在可留空白。



collect.py

本任务的所有程序都是采用python语言进行编写，大家在运行过程中只需要在对应位置填入文件路径即可，如出现bug可随时找我解决。

本代码文件会针对统一目录的C列表名，在年鉴中进行查找，收集到对应的表格并把它们存放在collect文件夹中。



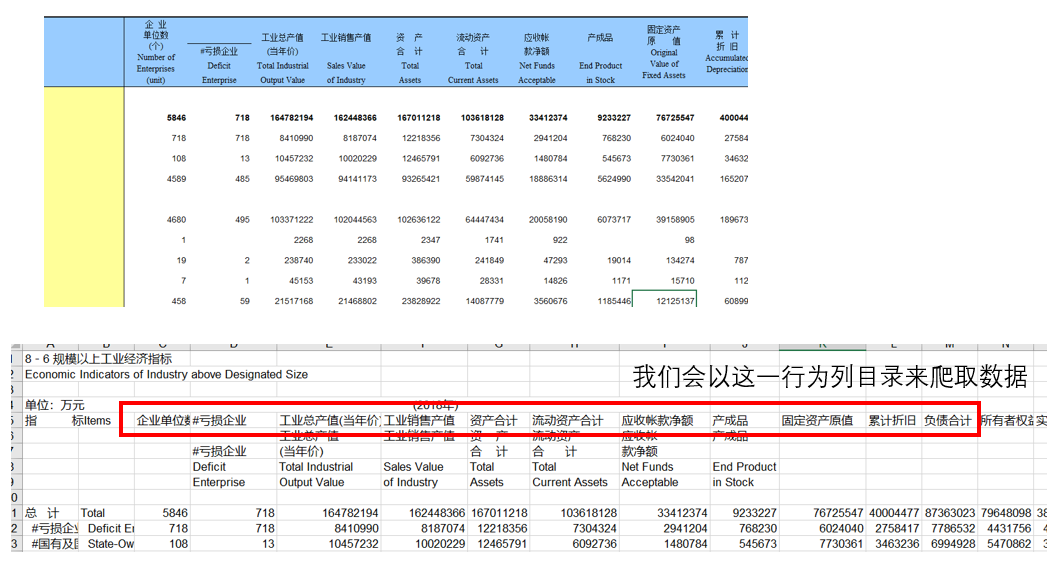
运行此代码后，会生成excel文件夹和collect文件夹。

excel文件夹：存放此城市此年份的所有的表格。

collect文件夹：存放与统一目录中表名相对应的表格，其中里面有个name.txt里面写有我们所收集到的表格的表名。

modify.py

本代码文件会针对collect文件夹中的所有表格进行格式调整，将列目录的所有目录名称都移至同一行并去除英文名称，方便下一步的数据爬取，因各个表格的格式往往不同，所以该程序并不能够对所有的表格格式都进行正确处理，如发现后续的data.xls中的**数据目录表达不完整**（主要集中在批发零售以及服务业的数据），可手动对此代码生成的modify文件夹中的表格进行调整。



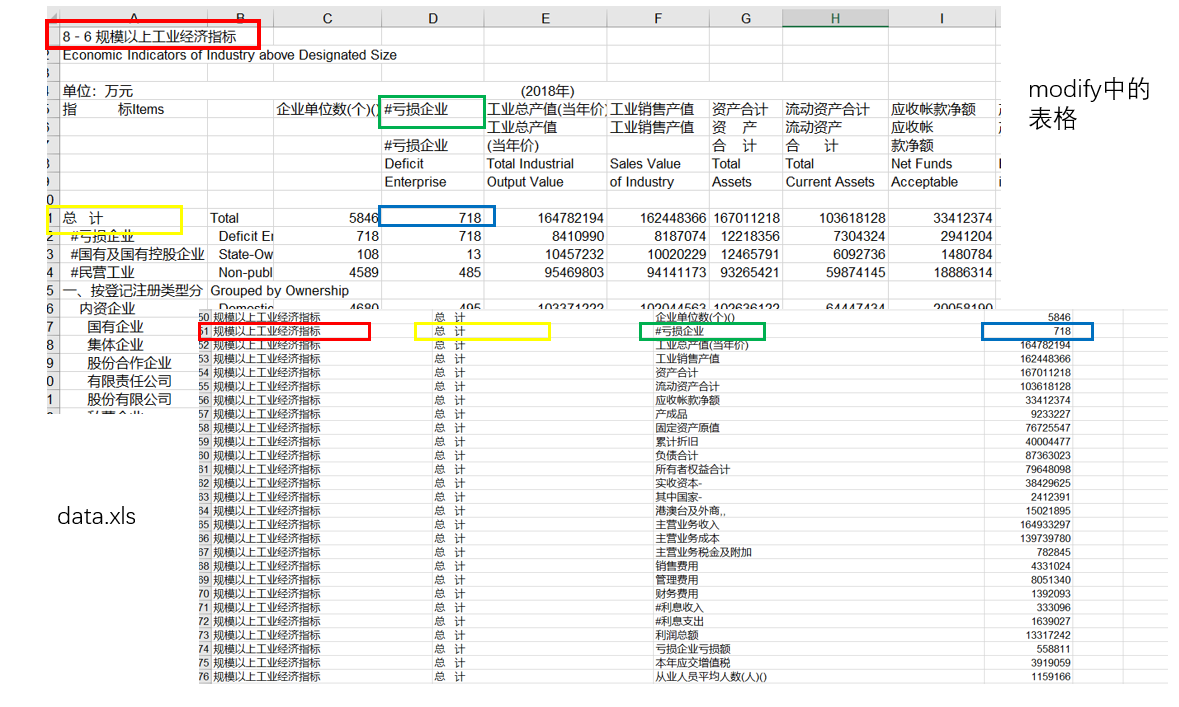


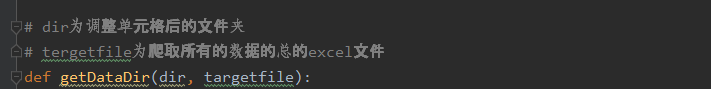
运行此代码后会生成modify文件夹。

modify文件夹：存放collect中表格的调整格式后的版本。

getData.py

本代码文件会把modify文件夹中的所有表格的表名、行目录、列目录、数据爬取下来，以每一行为一个数据项记录在data.xls中。





运行此代码文件后会生成data.xls

data.xls：存放modify文件夹中所有表格的表名、行目录、列目录、数据。

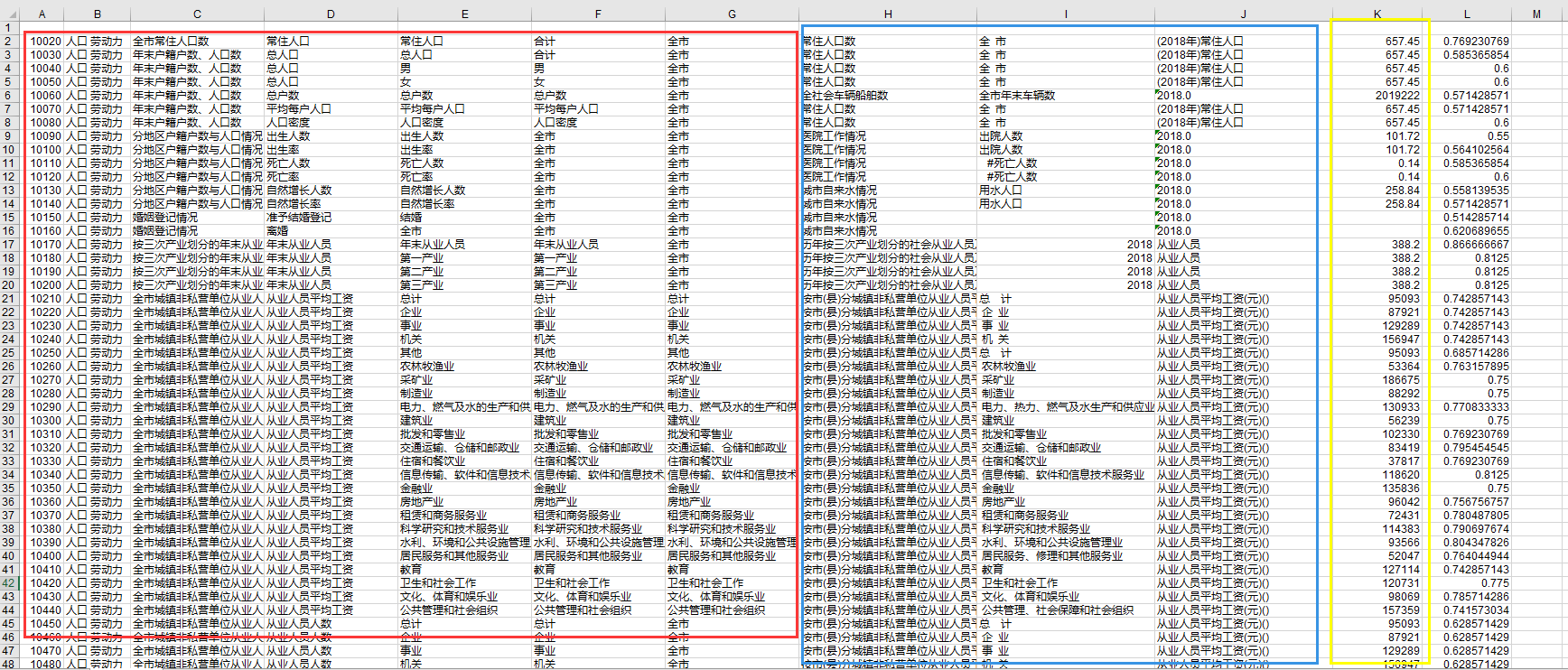
match.py

本代码文件会将“统一目录.xlsx”和“data.xls”各项数据的名称进行相似度匹配，找到相似度最高的数据并记录下来。



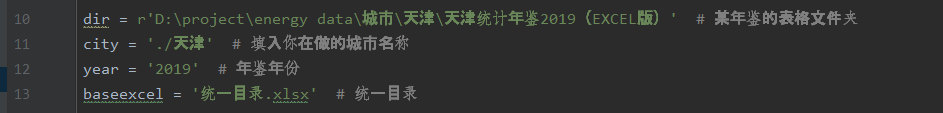
运行此代码后会生成match.xls文件。

match.xls：记录匹配结果的表格。其中A-G列为统一目录的各列名称，H-J为匹配到的data.xls（A-C列）中的各项数据的名称，K列为数据，L列为匹配的相似度∈[0, 1]，值越大相似度越高。此匹配结果不一定准确，所以需要进行人工审查并修正，主要针对H-K列。



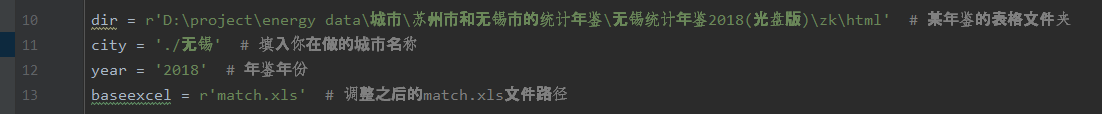
auto.py

将前面的所有代码文件整合起来，故在实际操作中我们只需要运行这份代码。在该代码第10-13行填入对应的文件路径或信息即可。运行之后它会在当前目录下创建该城市的文件，再在该城市文件夹中创建该年份文件夹，上述的所有生成文件都会在该城市该年份文件夹中。



auto2.py

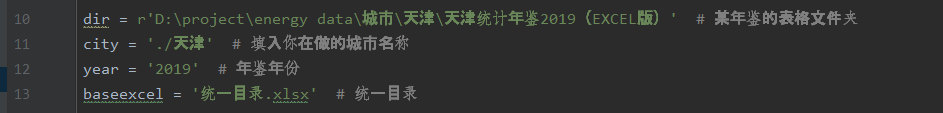
同auto.py，但auto.py是处理2019年的年鉴，auto2.py处理2010-2018年的年鉴。其中baseexcel为调整后的2019年的match.xls。

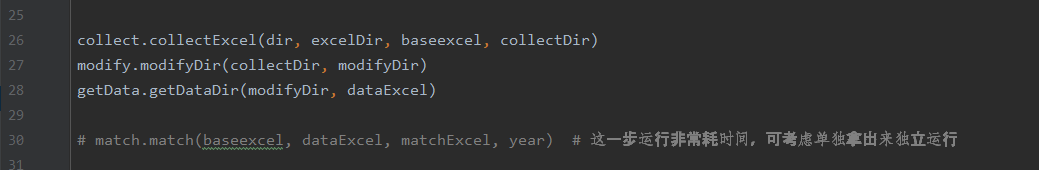


因为匹配准确率不能达到100%，且不同城市之间的表格格式差别较大，所以我们需要首先先对目标城市**2019年的年鉴**先进行数据收集，得到match.xls之后，进行人工审查调整，随后用调整后的2019年的match.xls来提高其余年份的准确率。

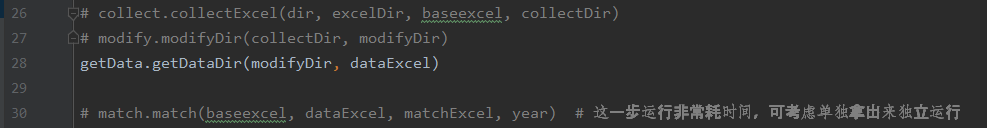
具体操作指引：

① 将collect.py、modify.py、getData.py、match.py、auto.py、auto2.py、统一目录.xlsx 放到同一个文件夹下，在auto.py第10-13行中填入文件路径等信息，运行auto.py。注意，此时运行26-28行代码，即生成data.xls，30行代码注释掉不运行。

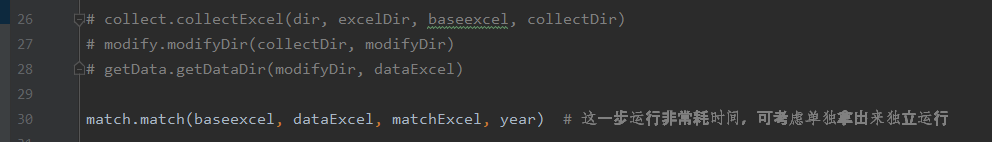




② 简单对data.xls进行检查，主要检查表名、行目录、列目录、数据是否有对应上，以及是否出现歧义。如出现问题，需对modify文件夹中的对应表格进行调整，之后再运行一次auto.py。注意，此时运行28行代码，26，27，30行代码注释掉。



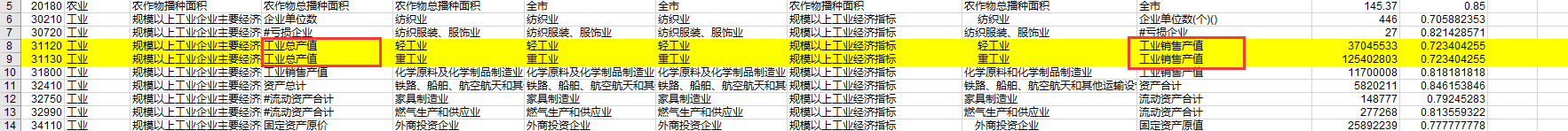
③ 进行匹配，这一部分会比较耗时，大致需要30分钟，等终端数字从1增加到4174即可。注意，此时运行auto.py时，需要注释掉26-28行代码。



④ 此时会得到2019年鉴的match.xls，备份好该文件（最好重命名成另一个文件）并打开，对其进行人工审查，如果有出错的地方，需要对H-K列进行修改（在审查2010-2018年的match.xls时，仅需修改K列的数据即可）。一般错误可分为3种：

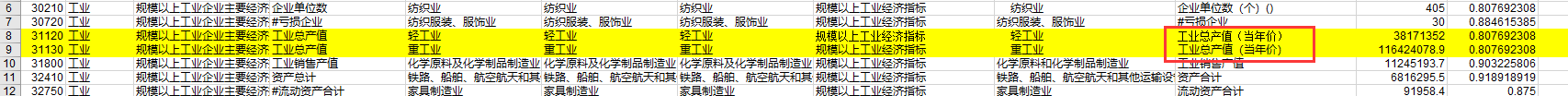
a. 年鉴中不存在此项数据，直接将该项数据的H-L列全部清空。

b. 年鉴中存在此项数据，但因为存在干扰项导致匹配出错。需要在data.xls中找到准确的数据项，并把它们的目录名称和数据录入到2019年的match.xls中，



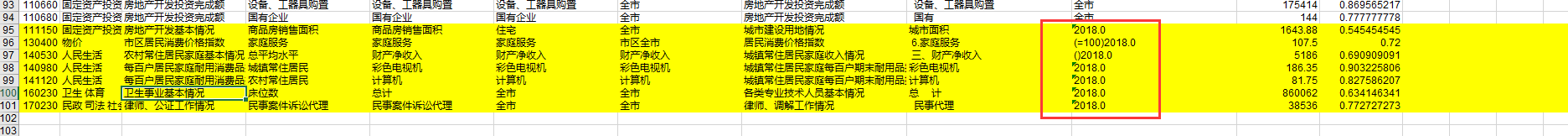


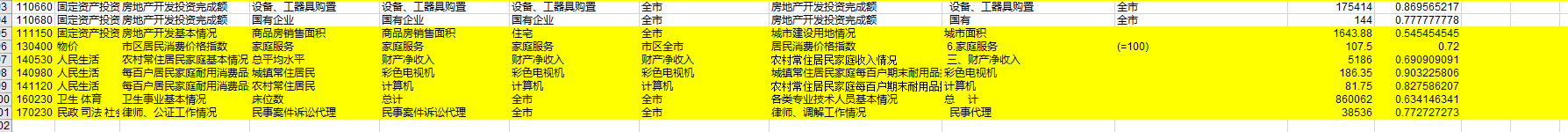
如这里因为年鉴中的工业总产值加了某些限定词，导致匹配出错，故对其进行修改。



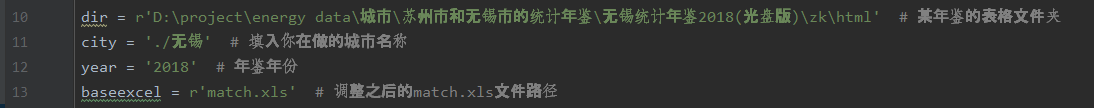
c. 年鉴中存在此项数据，但因为modify的过程中未能准确调整表格格式导致data.xls的数据项出现歧义等错误，需返回第②步进行操作，但因为第③步耗时实在太长，可以考虑将match.xls中出错的那些数据行都复制出来，新建成一个缩小版的统一目录进行填入第③步中进行匹配。

d. 如果在第H-J列的目录项出现年份信息，需将其删掉。





⑤ 此时我们得到一个准确率较高的2019年的match.xls文件，我们将这个文件作为统一目录来匹配其余年份的年鉴数据，对其余年份的年鉴重复①到④步，此时运行的代码为auto2.py（不是auto.py）。



⑥ 将所有年份的第K列（数据列）都合并到最开始的统一目录.xlsx中，合并完成后的统一目录和2019年的match.xls都需要提交给我。（做完2019年的match.xls之后也可将其发给我检查是否准确）

注意事项：

I. 进行任务的过程中记得及时**备份**好文件，最好是备份放到其他位置，多次运行代码过程中会对一些特定文件进行覆盖。

II. 如出现下图报错，打开白色框里的.py文件将第15行的 error = ‘strict’ 修改为 error = ‘ignore’

