**处理数据可视化**

**数据导入**

对于所有的数据，将其导入到数据库之中，以便于对数据进行展示与使用。对于数据库的选择，采用了PostgreSQL。使用psycopg2与PostgreSQL进行连接之后，通过pandas将数据从已经处理好的csv中读出，最后利用cursor将数据插入到数据库之中。其中database、user、password填写为用户的数据库名称、用户名及密码即可，具体代码展示如下：

|  |
| --- |
| Python conn = psycopg2.connect(host="127.0.0.1",  database="bigdata\_demo",  user="\*\*\*\*",  password="\*\*\*\*") cur = conn.cursor()  for i in range(10):  filename = "data\_support\_4e-6\_confidence\_0.4\part-0000" + \  str(i)+"-7d62d74b-257c-4c5e-a9be-025595d3baa2-c000.csv"  print(i,' now is file: ',filename)  df = pd.read\_csv(filename)  arr = np.array(df)  for i in tqdm(arr):  cur.execute("insert into modesearch(key, value, confidence,lift,support) values(%(key)s, %(value)s, %(confidence)s,%(lift)s,%(support)s)",  {'key': i[0], 'value': i[1], 'confidence': i[2], 'lift': i[3], 'support': i[4]})  # 向数据库中插入数据即可  conn.commit()  for i in range(10, 30):  filename = "data\_support\_4e-6\_confidence\_0.4\part-000" + \  str(i)+"-7d62d74b-257c-4c5e-a9be-025595d3baa2-c000.csv"  print(i,'now is file: ',filename)  df = pd.read\_csv(filename)  arr = np.array(df)  for i in tqdm(arr):  cur.execute("insert into modesearch(key, value, confidence,lift,support) values(%(key)s, %(value)s, %(confidence)s,%(lift)s,%(support)s)",  {'key': i[0], 'value': i[1], 'confidence': i[2], 'lift': i[3], 'support': i[4]})  # 向数据库中插入数据即可  conn.commit() |

**数据展示**

对于数据的最终展示与接口，采用了python的TK进行前端展示，最终效果如下：



在查询栏中输入需要查询的关键词即可，词与词之间需要使用空格进行隔开，输入完成后点击查询即可找到频率较高的模式匹配。



点击随机查询，即可随机查询一个匹配用于展示：



注意将程序中的数据库同上正确的连接到本地的数据库即可,之后运行demo.py即可出现页面的展示。

|  |
| --- |
| Python conn = psycopg2.connect(host="127.0.0.1",  database="bigdata\_demo",  user="\*\*\*\*",  password="\*\*\*\*") cur = conn.cursor() |