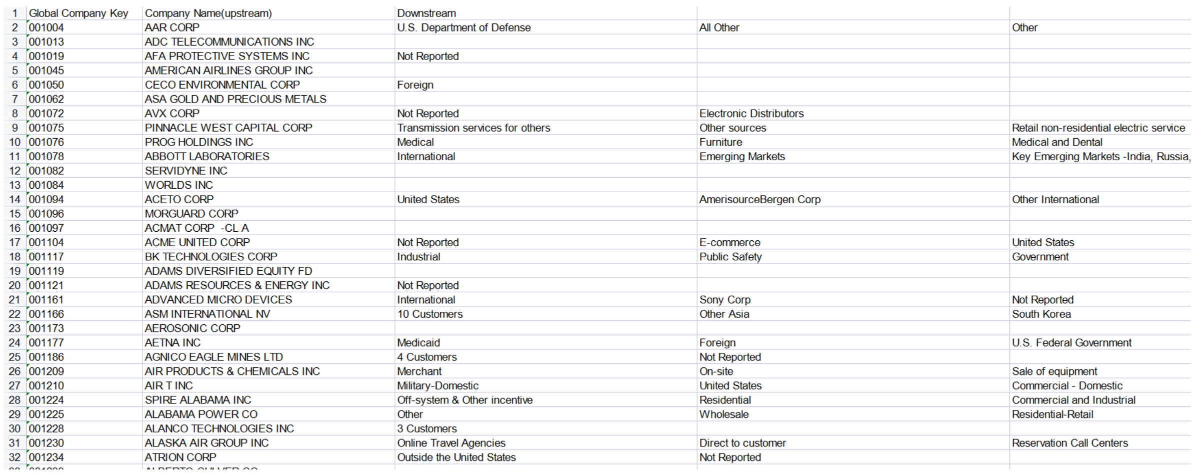
1. 仔细阅读了《基于供应链关系的股票收益预测研究》，读了一半但还没读完。

2. （代码真的写的很慢,需要练习）将数据去重后写入表中，如下：



1	Global Company Key	Company Name(upstream)	Downstream		
2	001004	AAR CORP	U.S. Department of Defense	All Other	Other
3	001013	ADC TELECOMMUNICATIONS INC			
4	001019	AFA PROTECTIVE SYSTEMS INC	Not Reported		
5	001045	AMERICAN AIRLINES GROUP INC			
6	001050	CECO ENVIRONMENTAL CORP	Foreign		
7	001062	ASA GOLD AND PRECIOUS METALS			
8	001072	AVX CORP	Not Reported	Electronic Distributors	
9	001075	PINNACLE WEST CAPITAL CORP	Transmission services for others	Other sources	Retail non-residential electric service
10	001076	PROG HOLDINGS INC	Medical	Furniture	Medical and Dental
11	001078	ABBOTT LABORATORIES	International	Emerging Markets	Key Emerging Markets -India, Russia,
12	001082	SERVIDYNE INC			
13	001084	WORLDS INC			
14	001094	ACETO CORP	United States	AmerisourceBergen Corp	Other International
15	001096	MORGUARD CORP			
16	001097	ACMAT CORP -CL A			
17	001104	ACME UNITED CORP	Not Reported	E-commerce	United States
18	001117	BK TECHNOLOGIES CORP	Industrial	Public Safety	Government
19	001119	ADAMS DIVERSIFIED EQUITY FD			
20	001121	ADAMS RESOURCES & ENERGY INC	Not Reported		
21	001161	ADVANCED MICRO DEVICES	International	Sony Corp	Not Reported
22	001166	ASM INTERNATIONAL NV	10 Customers	Other Asia	South Korea
23	001173	AEROSONIC CORP			
24	001177	AETNA INC	Medicaid	Foreign	U.S. Federal Government
25	001186	AGNICO EAGLE MINES LTD	4 Customers	Not Reported	
26	001209	AIR PRODUCTS & CHEMICALS INC	Merchant	On-site	Sale of equipment
27	001210	AIR T INC	Military-Domestic	United States	Commercial - Domestic
28	001224	SPIRE ALABAMA INC	Off-system & Other incentive	Residential	Commercial and Industrial
29	001225	ALABAMA POWER CO	Other	Wholesale	Residential-Retail
30	001228	ALANCO TECHNOLOGIES INC	3 Customers		
31	001230	ALASKA AIR GROUP INC	Online Travel Agencies	Direct to customer	Reservation Call Centers
32	001234	ATRION CORP	Outside the United States	Not Reported	

```
import xlrd
import xlwt
import difflib


# global firm names 和 us names 的 Global Company Key 和 Company Name 提取
global_workbk = xlrd.open_workbook(r'C:\Users\jc\Documents\Pydata'+
                                  '\Database Table\global firm names.xlsx')
global_worksh = global_workbk.sheet_by_name('0x77igavdumz8vul')
global_cpnames = global_worksh.col_values(colx = 7, start_rowx = 1)
global_cpkey = global_worksh.col_values(colx = 0, start_rowx = 1)
# 组成列表
global_namekey = list(zip(global_cpnames, global_cpkey))


us_workbk = xlrd.open_workbook(r'C:\Users\jc\Documents\Pydata'
                              +'\Database Table\\us names.xlsx')
us_worksh = us_workbk .sheet_by_name('76aqys7wh9axjpme')
us_cpnames = us_worksh.col_values(colx = 9, start_rowx = 1)
us_cpkey = us_worksh.col_values(colx = 0, start_rowx = 1)
# 组成列表
us_namekey = list(zip(us_cpnames,us_cpkey))


# 将customer表中的Customer Name列和Global Company Key 提取出
customer_workbk = xlrd.open_workbook(r'C:\Users\jc\Documents\Pydata'+
                                     '\Database Table\customer.xlsx')
customer_worksh = customer_workbk.sheet_by_name('vozkv0ioajsw5wov')
customer_downstream = customer_worksh.col_values(colx = 2, start_rowx = 1)
customer_upkey = customer_worksh.col_values(colx = 0, start_rowx = 1)
# 组成列表
```

```
customer_up_down = list(zip(customer_upkey,customer_downstream))


# global firm names 和 us names 的Global Company Key 和 Company Name列表合并后去重
Allnamekey_lst = list(set( global_namekey+us_namekey))
#对customer_up_down列表也去重
customer_up_down = list(set(customer_up_down))
#按照Global Company Key进行排序
Allnamekey_lst.sort(key=lambda x:x[1])

# 将Global Company Key 和 Company Name 写入Allcompany表中
Allcompany = xlwt.Workbook()
Allcompany_sheet = Allcompany.add_sheet('sheet1')
name_list = ['Global Company Key','Company Name(upstream)','Downstream']
for i in name_list:
    Allcompany_sheet.write(0, name_list.index(i), i)

for namekey in Allnamekey_lst:
    Allcompany_sheet.write(Allnamekey_lst.index(namekey)+1,0,namekey[1])
    Allcompany_sheet.write(Allnamekey_lst.index(namekey)+1,1,namekey[0])

cust_num = [[i[1],0] for i in Allnamekey_lst]
#比较 upstream key(customer_up_down[i][0])和global company key(Allnamekey_lst[j]
[1])
for upkey_down in customer_up_down:
    for name_key in Allnamekey_lst:
        if upkey_down[0] == name_key[1]:
            Allcompany_sheet.write(Allnamekey_lst.index(name_key)+1
                                   ,2+cust_num[Allnamekey_lst.index(name_key)]
[1]
                                   ,upkey_down[1])
            cust_num[Allnamekey_lst.index(name_key)][1] += 1

# 保存文件
Allcompany.save('Allcompany.xlsx')
```

3. 将BM算法以python语言实现,还在测试阶段

pattern模式串是小串，target目标串是大串。
以下代码可以实现查找小串在大串中的位置，如果没有，则返回-1，如果找到了，返回小串的位置：

```
import xlrd
import xlwt
# pattern模式串是小串，target目标串是大串。
#以下代码可以实现查找小串在大串中的位置，如果没有，则返回-1，如果找到了，返回小串的位置
def pattern(pattern,target):
    patter = pattern.lower()
    target = target.lower()
    tLen = len(target)
    pLen = len(pattern)
    # 当模式串比主串长时，没有可比性，直接返回-1
    if ( pLen > tLen ):
        return -1
    bad_table = build_bad_table(pattern)
    good_table = build_good_table(pattern)

    for i in range(pLen-1,tLen):
        j = pLen-1
        while(target[i] == pattern[j]):
            if (j ==0):
                return i
            i -= 1
            j -= 1
        i += max(good_table[pLen-j-1],bad_table[ord(target[i])])
    return -1

def build_bad_table(pattern):
    patter = pattern.lower()
    TABLE_SIZE = 256
    bad_table = ['0' for i in range(TABLE_SIZE)]
    pLen = len(pattern)

    for i in range(len(bad_table)):
        bad_table[i] = pLen

    for i in range((pLen)-1):
        k = ord(pattern[i])# 记录下当前字符的ASCII码值
        bad_table[k] = pLen-1-i

    return bad_table

def build_good_table(pattern):
    patter = pattern.lower()
    pLen = len(pattern)
    lastPrefixPosition = pLen
    good_table = [0 for i in range(pLen)]
    for i in range(pLen-1 , 0 , -1):
        if(isPrefix(pattern, i+1)):
            lastPrefixPosition = i+1
        good_table[pLen-1-i] = lastPrefixPosition-i+pLen-1

    for i in range(pLen-1):
        slen = suffixLength(pattern,i)
        good_table[slen] = pLen -1 -i +slen

    return good_table

def isPrefix(pattern, p):
    patter = pattern.lower()
    patternLength = len(pattern)
    j = 0
    for i in range(patternLength):
        if(pattern[i] != pattern[j]):
            return False
    return True

def suffixLength(pattern, p):
    pLen = len(pattern)
    length = 0
    i = p
    j = pLen -1
    while((i>=0)&(pattern[i]==pattern[j])):
        length += 1
        i -= 1
        j -= 1

    return length

x = pattern('other','FARMER BROTHERS CO')
print(x)
```

```
>>> pattern('other','FARMER BROTHERS CO')
 9
```

以上算法实现了**字符串位置的查找**，应该是可以用于本实验中**大部分公司名和缩写字符串的匹配的。**