Chartfield Value Set Program

The purpose of this project was to standardize the financial data stored online in a database called FAME, which stands for Financial Accounting Made Easier. FAME is the NYU's core financial system. To clean the data, I was tasked to find any errors within the data that did not match the accepted standard. I wrote this program in Python with the skills I learned from my sophomore programming class. The next step would be to find duplicates which cause incorrect balances in financial reports. I hope to start the second part of this project soon.

Below are snippets of code:

```
def list_4_original(Original):

    data = Original.read()

    data2 = data.split('\n')

    hold = []

    skip = 0
    for line in data2:
        skip += 1
        if skip == 1:
            pass
        else:
            datasplit = line.split('\t')
            hold.append(datasplit)
    return hold

def list_compare(cleandoc):
    cleanread = cleandoc.read()

    cleansplit = cleanread.split('\n')

    cleanlist = []

    skip = 0
    for line in cleansplit:
        skip += 1
        if skip == 1:
            pass
        else:
            datasplit = line.split('\t')
            cleanlist.append(datasplit)
    return cleanlist
```

Here are functions I defined to streamline the process of my analysis. These functions helped in identifying which pieces of data were incorrect.

```
#-------------------------------------------------------------------------------#

#FUND ERRORS
##opening the file

Original = open(file)

hold = list_4_original(Original)

fund_doc = open('J:\FSM\Maria Restrepo\PROGRAMS\\ERRORS\\CVS with FUND errors'+date+'.csv.txt','w')
#fund_doc.write(''+'\t'+''+'\t'+''+'\t'+'FUND'+'\t'+'FUND'+'\n')


cleandoc = open('J:\FSM\Maria Restrepo\PROGRAMS\\CLEAN\\CVS clean w.o MONTHS'+date+'.csv.txt')

newclean = open('J:\FSM\Maria Restrepo\PROGRAMS\\CLEAN\\CVS clean w.o MONTHS.FUNDS ERR'+date+'.csv.txt','w')


for row in hold:
    FUNDFROM__ = row[3].startswith('-')
    FUND_LEN = part_len('fund',row)
    FUND_EMPTY = empty('fund',row)
    FUNDTO__ = row[4].startswith('-')

    fund_errors = FUNDFROM__ or FUND_LEN or FUND_EMPTY or FUNDTO__

    if fund_errors:
        fund_doc.write(row[0]+'\t'+row[1]+'\t'+row[2]+'\t'+row[3]+'\t'+row[4]+'\t'+row[5]+'\t'+row[6]+'\t'+row[7]+'\t'+row[8]+'\t'+row[9]+'\t'+row[10]+'\t'+row[11]+'\t'+row[12]+'\n')

print("All done with CVS with FUND errors" + date+'.csv.txt')
fund_doc.close()
Original.close()

cleanlist = list_compare(cleandoc)

try:
    for row in cleanlist:
        FUNDFROM__ = row[3].startswith('-')
        FUND_LEN = part_len('fund',row)
        FUND_EMPTY = empty('fund',row)
        FUNDTO__ = row[4].startswith('-')

        fund_errors = FUNDFROM__ or FUND_LEN or FUND_EMPTY or FUNDTO__

        if fund_errors:
            lol = 'lol'

        else:
            newclean.write(row[0]+'\t'+row[1]+'\t'+row[2]+'\t'+row[3]+'\t'+row[4]+'\t'+row[5]+'\t'+row[6]+'\t'+row[7]+'\t'+row[8]+'\t'+row[9]+'\t'+row[10]+'\t'+row[11]+'\t'+row[12]+'\n')
except:
    if row != ['']:
       print('An exception was raised and not because it is an empty line at the end of the text document.')

#YOU CANT USE AN ELSE STATEMENT BECAUSE AN EXCEPTION WILL ALWAYS BE RAISED BECAUSE NOTEPAD WILL ALWAYS ADD A BLANK LINE TO THE END
#OF THE FILE AND THEREFORE THE ERROR OF 'INDEX OUT OF RANGE' WILL ALWAYS OCCUR
#print('''All done with CVS with "w.o MONTHS.FUNDS ERR"''' + date+'.csv.txt')
cleandoc.close()
newclean.close()
```

This next piece of code shows one part of the analysis. Comparisons are done row by row and a new document was created where the clean data will be stored after its analyzed. There were also occasional placeholders where it was needed.

Thank you for taking the time to review my project and application. I acknowledge my code is a little disorganized, but I placed a lot of heart and patience with the resources I had. Thank you again.