Nonprofit Tax Data Analysis

Jessica Cerda

2024-04-10

Nonprofit Tax Data Analysis

Introduction

An analysis on the yearly Total Contribution Amount for the following five nonprofits specific to education

- STEP UP FOR STUDENTS INC.
  - STEP UP FOR STUDENTS EMPOWERS FAMILIES TO PURSUE AND ENGAGE IN THE MOST APPROPRIATE LEARNING OPTIONS FOR THEIR CHILDREN, WITH AN EMPHASIS ON FAMILIES WHO LACK THE INFORMATION AND FINANCIAL RESOURCES TO ACCESS THESE OPTIONS. BY PURSUING THIS MISSION, WE HELP PUBLIC EDUCATION FULFILL THE .PROMISE OF EQUAL OPPORTUNITY.
- HENNEPIN HEALTHCARE SYSTEM INC
  - WE PARTNER WITH OUR COMMUNITY, OUR PATIENTS, AND THEIR FAMILIES TO ENSURE OUTSTANDING CARE FOR EVERYONE, WHILE IMPROVING HEALTH AND WELLNESS THROUGH TEACHING, PATIENT AND COMMUNITY EDUCATION, AND RESEARCH.
- Queen's University at Kingston
  - Providing secondary and post-graduate education and undertaking research activities.
- INDIANA UNIVERSITY HEALTH BALL MEMORIAL
  - Improve the health of our patients and community through innovation and excellence in care, education, research and service.
- EVERGLADES COLLEGE INC
  - KEISER UNIVERSITY AND EVERGLADES UNIVERSITY ARE REGIONALLY ACCREDITED PRIVATE CAREER UNIVERSITIES THAT PROVIDE EDUCATIONAL PROGRAMS AT THE UNDERGRADUATE AND GRADUATE LEVELS FOR A DIVERSE STUDENT BODY IN TRADITIONAL, NONTRADITIONAL AND ONLINE DELIVERY FORMATS. THE MAIN CAMPUS IS LOCATED IN FORT LAUDERDALE, WITH CAMPUSES LOCATED THROUGHOUT THE STATE OF FLORIDA AND INTERNATIONALLY. THROUGH QUALITY TEACHING, LEARNING AND RESEARCH, THE UNIVERSITY IS COMMITTED TO PROVIDE STUDENTS WITH OPPORTUNITIES TO DEVELOP THE KNOWLEDGE, UNDERSTANDING AND SKILLSNECESSARY FOR SUCCESSFUL EMPLOYMENT. COMMITTED TO A "STUDENTS FIRST" PHILOSOPHY, KEISER UNIVERSITY PREPARES GRADUATES FOR CAREERS IN BUSINESS, CRIMINAL JUSTICE, HEALTH CARE, TECHNOLOGY, HOSPITALITY, EDUCATION AND CAREER-FOCUSED STUDIES.

The data set was created by going through 2.3 million IRS990 files located on the IRS website.

Plot

The following are various plots that depict the findings from the `nonprofit` data set.

#Plot for the Total Revenue Count

ggplot(data = nonprofits,

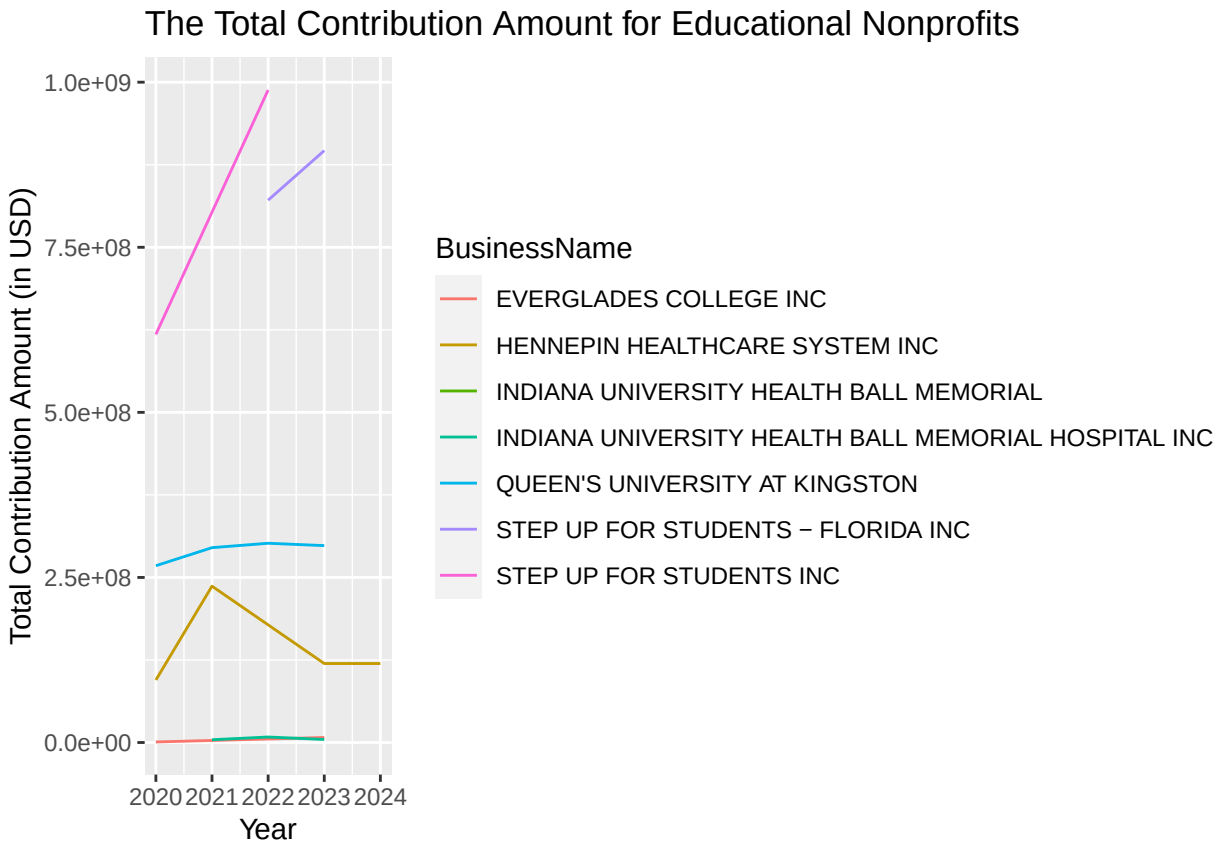
mapping = aes(x = Year, y = CYTotalRevenueAmt, col = BusinessName)) + geom_line() + labs(x = "Year", y = "Yearly Revenue (in USD)", title = "The Yearly Total Revenue Count for Educational Nonprofits", caption = "All of the nonprofits above saw growth within their Total Yearly Revenue throughout the years reported, with Hennepin Healthcare System INC having the highest yearly total revenue.")

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

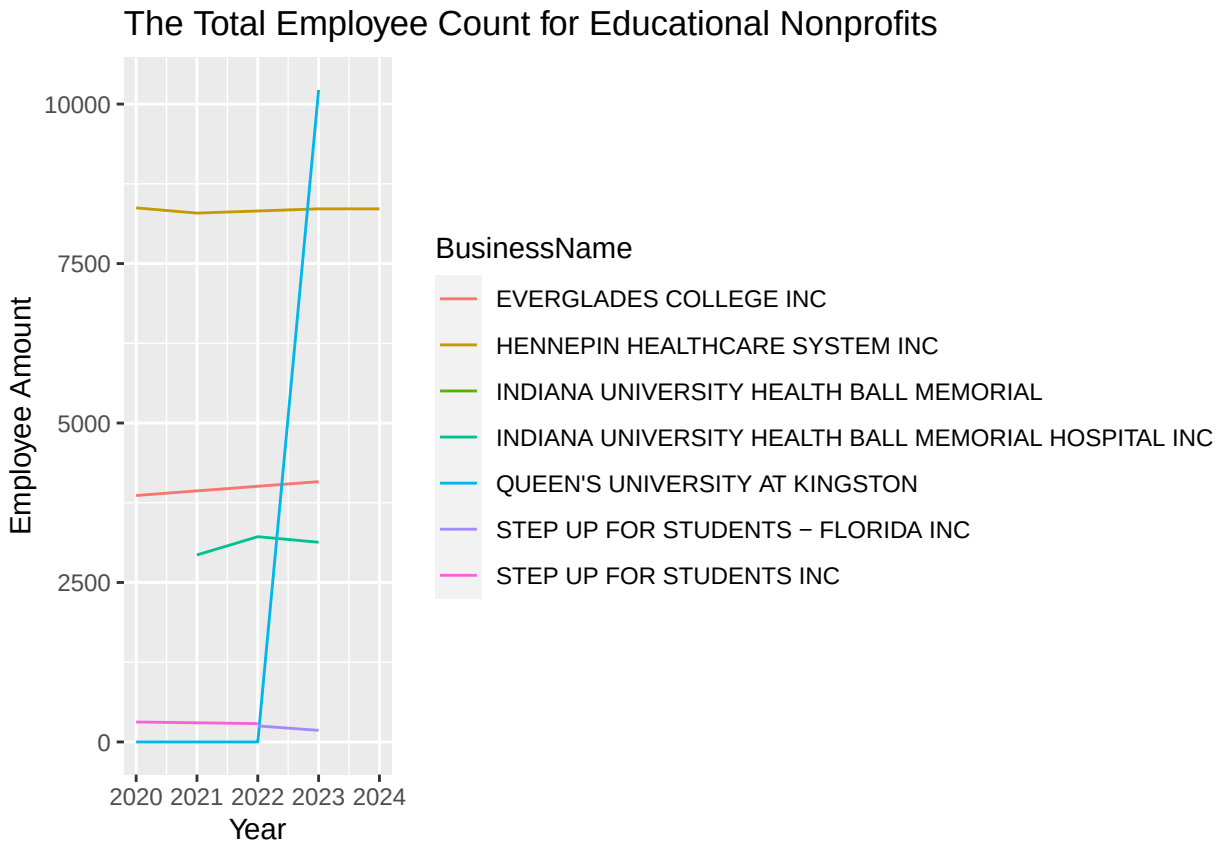
```
#reading in the data created for the Plot
nonprofits <- read.csv("/Users/jessicacerda/Desktop/STAT 129/nonprofits.csv")

#Converts the Business names to All Caps
nonprofits$BusinessName <- toupper(nonprofits$BusinessName)


#convert returnDateStamp to Date
nonprofits$returnDateStamp <- as.Date.character(nonprofits$returnDateStamp)

#extract the year from the date
nonprofits <- nonprofits |> mutate(Year = year(nonprofits$returnDateStamp))
```

```
#plot for Total Contribution Amount For Each Nonprofit

ggplot(
  data = nonprofits,
  mapping = aes(x = Year, y = TotalContributionsAmt, col = BusinessName)) + geom_line() + labs(x = "Yea
```



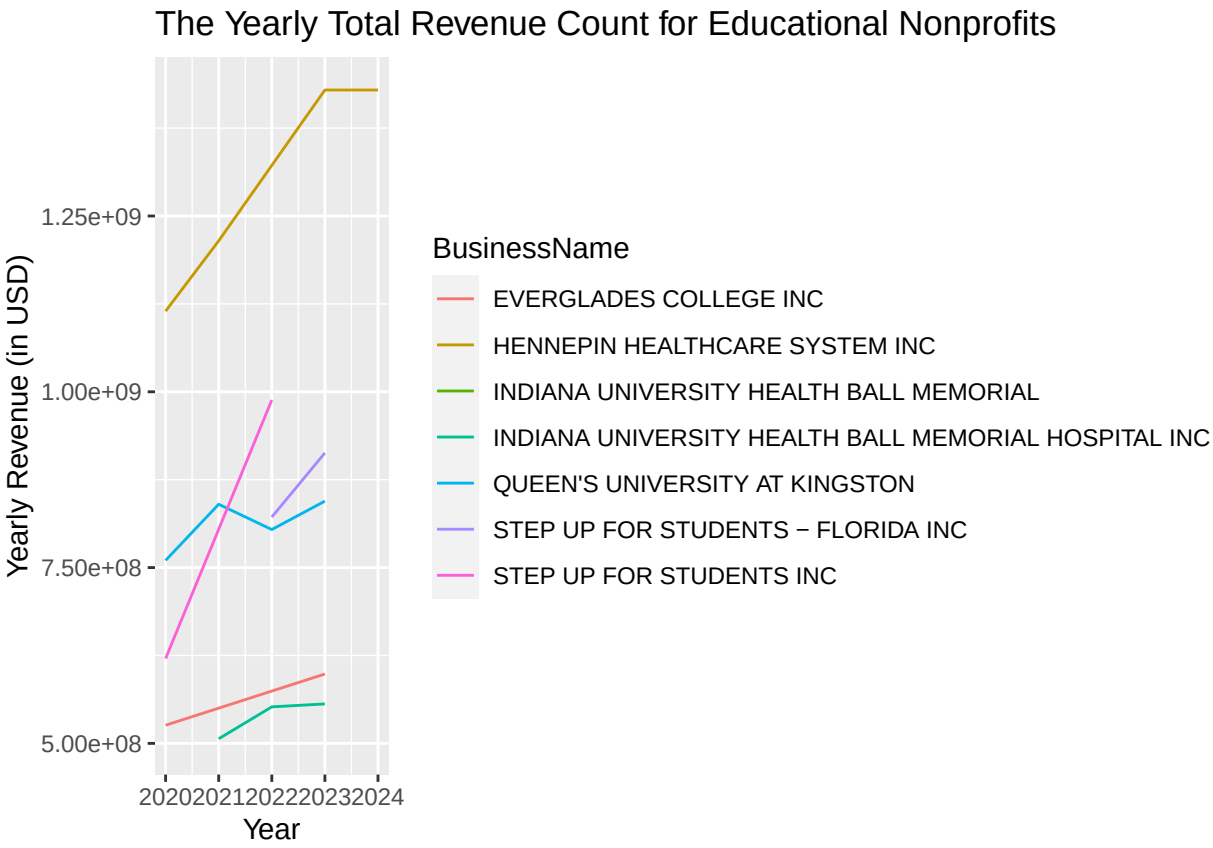
The graph shows several different findings. The graph shows that Everglades College Inc. and Indiana University Health Ball Memorial contribute little if any of their earnings per year. Hennepine Healthcare System INC saw a growth and then a decline within their yearly contribution amount while Queen's University at Kingston was mostly stagnant. Step up for Students was consistent within their growth in their yearly contribution amount.

```
#Plot for the Total Employee Count
ggplot(data = nonprofits,
     mapping = aes(x = Year, y = TotalEmployeeCnt, col = BusinessName)) + geom_line() + labs(x = "Year"
```



The graph depicts nearly stagnant Employee numbers for four of the five nonprofits. The nonprofit Queen's Univeristy at Kingston saw major growth in the number of emplyees from 2022 to the year 2023.

```
#Plot for the Total Revenue Count
ggplot(data = nonprofits,
       mapping = aes(x = Year, y = CYTotalRevenueAmt, col = BusinessName)) + geom_line() + labs(x = "Ye
```



All of the nonprofits above saw growth within their Total Yearly Revenue throughout the years reported, with Hennepin Healthcare System INC having the highest yearly total revenue.

XPATH

```
    #regular expression to filter out nonprofits for education
    pattern = r'\beducation\b'
    value = 'ActivityOrMissionDesc'

    if value in result and re.search(pattern, result[value], flags=re.IGNORECASE) is None:
            return None
```

The following node in which the regular expression was applied to was the 'ActivityOrMissionDesc' node. This allowed the if statement to evaluate whether there was a mention of education within nonprofit's Activity or Mission Description.

```
#creates a subset for just a specific year to determine 5 most important nonprofits
outputyear20 = [x for x in outputint if "2020" in x["returnDateStamp"]]

#sorting based of CYTotalRevenueAmt

outputyear20.sort(key = lambda x: x["CYTotalRevenueAmt"])
```

In order to determine which nonprofits were the most important, I created a lambda function that would sort all of the nonprofits based off of their Total Revenue Amount from the year 2020. I then took only the bottom five after the sort since the sort function organizes nonprofits from the lowest revenue amount to the highest revenue amount.

Parallel Programming

```
# 10 parallel workers
with Pool(10) as p:
    # Parallel map
    r = p.map(extract, all990)

    #filters out the None in Results
    output = list(filter(None,r))
```

The parallel programming was used in this program to create 10 parallel workers that would conduct the extract function across all the files in the all990 variable which in this case was over two million three hundred thousand files from the IRS. Using parallel programming allowed the run time for the extract function run time to be cut down to less than a minute with the exact run time being depicted in the screen capture below.

```
[In [1]: %time %run nonprofit.py
CPU times: user 2.26 s, sys: 644 ms, total: 2.9 s
Wall time: 50.1 s
```

Regular Expression

```
    #regular expression to filter out nonprofits for education
    pattern = r'\beducation\b'
    value = 'ActivityOrMissionDesc'

    if value in result and re.search(pattern, result[value], flags=re.IGNORECASE) is None:
            return None
```

The following code above depicts the regular expression used to find all the nonprofits pertaining to education in the 2.3 million XML IRS 990 Files. The regular expression was put in an "*if*" statement that checked the Activity and Mission Description of the IRS Files. If there was no match then it would return"None" otherwise it would return the result back into the list made for the results. The r in `r'\beducation\b'` indicates that it is a regular expression and the at the beginning and end of education indicate word boundaries meaning that the match must be "education" and not another word with the phrase education within it such as "educational"

Appendix

Code used

Extract function

```
def extract(xmlfile):
    """
    Extract a dictionary containing the elements of interest
    """
    tree = etree.parse(xmlfile)
    fields990 = ["ActivityOrMissionDesc", "MissionDesc", "TotalEmployeeCnt", "TotalAssetsEOYAmt", "TotalContributionsAmt", "CYTotalRevenueAmt"]

    # Hold all the results
    result = {}
    for f in fields990:
        # Won't always be there
        try:
            result[f] = tree.xpath("/Return/ReturnData/IRS990/" + f + "/text()")[0]
        except:
            # xpath fails for some reason, so just give up!
            # A better way to handle this is to actually *look*
            # at this XML file, which may have a different structure.

            return None

    #extracts the file return date

    try:
        result["returnDateStamp"] = tree.xpath("/Return/ReturnHeader/ReturnTs/text()")[0]
    except:
        return None

    #extract the EIN for the business
    try:
        result["EIN"] = tree.xpath("/Return/ReturnHeader/Filer/EIN/text()")[0]
    except:
        return None

    #exctacts the company name
    try:
        result["BusinessName"] = tree.xpath("/Return/ReturnHeader/Filer/BusinessName/BusinessNameLine1Txt/text()")[0]
    except:
        return None

    #regular expression to filter out nonprofits for education
    pattern = r'\beducation\b'
    value = 'ActivityOrMissionDesc'

    if value in result and re.search(pattern, result[value], flags=re.IGNORECASE) is None:
            return None



    return result
```

Parallel Command

```
# 10 parallel workers
with Pool(10) as p:
    # Parallel map
    r = p.map(extract, all990)

    #filters out the None in Results
    output = list(filter(None,r))
```

Creating the Data Frame from output of running parallel on the extract function

```
#Creates a deepcopy and converts strings in certain areas to integers
from copy import deepcopy
outputint = deepcopy(output)

for x in outputint:
    for k, v in x.items():
        if k in {"TotalEmployeeCnt","TotalAssetsEOYAmt", "TotalContributionsAmt", "CYTotalRevenueAmt","EIN"}:
            x[k] = int(v)


#creates a subset for just a specific year to determine 5 most important nonprofits
outputyear20 = [x for x in outputint if "2020" in x["returnDateStamp"]]

#sorting based of CYTotalRevenueAmt

outputyear20.sort(key = lambda x: x["CYTotalRevenueAmt"])

#filterting to only find the top 5 based off the CYTotalRevenueAmt
outputyear5 = outputyear20[-5:]
outputEIN = {x["EIN"] for x in outputyear5}

#Filtering the original dataset to only find those that correspond with EIN

output5 = [x for x in outputint if x['EIN'] in outputEIN]

#conver the data into a data frame
import pandas as pd
outputdataframe = pd.DataFrame(output5)

#Converting the data frame to a csv file to export to R
outputdataframe.to_csv('nonprofits.csv',index=False)
```