Data Insights Take-Home

Marianne C. Halloran

October 14, 2017

IRS 2016 Form 990s from AWS, public dataset containing financial information about NPOs: https://aws.amazon.com/public-datasets/irs-990/

IRS Style Sheet for Form 990: IRS990 StyleSheet

A. Data Info and Access

- Vital information on the tax-exempt community, comprised mostly of 501(c)(3) organizations, but also includes 501(c),4947(a)(1) and 527.
- Does not include donor information or other personally identifiable information (important from ethics' perspective).

B. Data format and contents For the purposes of this exercise, I looked into the following fields:

1. **EIN** (EIN): Employer identification number, format = integer.
2. **Contract termination** (contract_term): organization has terminated its existence or ceased, format = [0,1] for [not terminated, terminated]
3. **Tax_status** (tax_status): refer to the tax-exemption character of the NPO. Data format is categorical (integer):
   0 - 501(c)(3) organizations;
   1 - 501(c)organizations;
   2 - 4947(a)(1)organizations;
   3 - 527 organizations;
   4 - Not answered.
4. **Organization Name** (org_name), **City** and **State**: format = string.
5. **Activity** (activity): short description of the NPO's activities and mission, format = string. *Could be used in a NLP framework*
6. **Year formed** (year_formed): year of establishment of NPO, format = integer.
7. **Volunteer and Employee counts** (volunteer_ct, employee_ct): number of total volunteers and employees, format = integer.
8. **Total Revenues and Expenses** (total_revenue, total_expenses): format = float In specific here, we look at:
9. **Revenues** from fundraising events, campaigns, membership dues, government grants, gifts and program services.
10. **Expenses** related to management, compensation, and service.

*** The idea is that, by understanding how a NPO obtains revenue and spends its funds, we will be better poised to understand its efficacy. It also answers the questions about the financial strength of the NPO (its ability to attract resources, level of reserves, financial accountability, etc).**

1. **Net Assets** (net_assets): format = float
2. **Political and Lobbying Activity** (pol_act,lob_act): categorical representation for political or lobbying activity. True/False=(1,0)
3. **Foreign Affairs: offices, fundraising and assistance** (foreign_office, foreign_fundraising,foreign_assist): categorical representation of foreign offices, fundraising or assistance to individuals. True/False(1,0)

In [1]:

```
#==============================================================
===#
# LIBRARIES
   #
#==============================================================
===#
from __future__ import print_function
import numpy as np
import seaborn as sb
import requests
import csv
import os
from io import StringIO
import pandas as pd
from bs4 import BeautifulSoup as bs
from IPython.display import FileLink, FileLinks
```

In [2]:

```
#==============================================================
===#
# LOAD DATASET INDEX
   #
#==============================================================
===#
# Index listings of available filings (JSON and CSV)
# https://s3.amazonaws.com/irs-form-990/index_2016.csv
# https://s3.amazonaws.com/irs-form-990/index_2016.json
# Use field OBJECT_ID to download forms

# Load and save for later
save_file_name = 'index_990_2016.csv'
url = 'https://s3.amazonaws.com/irs-form-990/index_2016.csv'

# Load for Pandas
download=requests.get(url).content
index_2016=pd.read_csv(StringIO(download.decode('utf-8')))

print(u"\u0011",'Retrieved %d NPOs names from: \n\n%s \nto
\n%s' %
      (len(index_2016.TAXPAYER_NAME),
       index_2016.TAXPAYER_NAME.iloc[0],
       index_2016.TAXPAYER_NAME.iloc[-1]    ))

index_2016.to_csv('input/index.csv', index=False)
del index_2016
```

```
▶ Retrieved 378420 NPOs names from:

HARRIET AND HARMON KELLEY FOUNDATION FOR THE ARTS
to
SOUTH TOMS RIVER VOLUNTEER FIRST AID SQUAD INC
```

In [5]:

```
#================================================================
===#
# DOWNLOAD DATASET
   #
#================================================================
===#

# If loading dataset for the first time, uncomment line bellow
meta = pd.read_csv('input/index.csv')
NPO_meta = []; k=0

# Read xml
print(u"\u0011","Retrieving records.")
for xmlid in meta['OBJECT_ID']:
    try:
        url = "https://s3.amazonaws.com/irs-form-990/%d_public.
xml" % xmlid
        NPOxml = requests.get(url)
        NPOsoup = bs(NPOxml.text[3:], 'xml') # doing the [3:] t
akes care of some weird characters at front
    except requests.exceptions.Timeout:
        print("Timeout")
        pass
    except requests.exceptions.TooManyRedirects:
        print("Too Many Redirects")
        pass
    except requests.exceptions.RequestException as e:
        print("Request Exception: e")
        break


    #============================================================
=======#
    # GET NPO DATA
       #
    #============================================================
=======#

    ## EIN
    try:
        EIN = (NPOsoup.find('EIN').contents[0]).encode('utf-8')
    except AttributeError:
        EIN = 0

    ## Contract Termination
    # If the NPO discontinued operations or disposed of more th
an 25% of its assets
    try:
        contract_term =
(NPOsoup.find('ContractTerminationInd').contents[0]).encode('ut
f-8')
    except AttributeError:
        contract_term = 0

    ## Tax Exempt Status
    tax_status = None
    status_fields = ['Organization501c3Ind','Organization501cIn
d',
                      'Organization4947a1Ind','Organization527I
nd']
    for status_field in status_fields:
        try:
            tax_status =
(NPOsoup.find(status_field).contents[0]).encode('utf-8')
            if tax_status == 'X':
                tax_status = status_fields.index(status_field)
            break
        except AttributeError:
            pass
    if tax_status == None:
        tax_status = 4

    ## Name
    try:
        org_name = (NPOsoup.find('BusinessNameLine1Txt').conten
ts[0]).encode('utf-8')    ## City
    except AttributeError:
        try:
            org_name = (NPOsoup.find('Filer').BusinessName.BusinessNameLine1.contents[0]).encode('utf-8')
        except AttributeError:
            org_name = None

    ## City
    try:
        city = (NPOsoup.find('Filer').USAddress.CityNm.contents[0]).encode('utf-8')
    except AttributeError:
        try:
            city = (NPOsoup.find('Filer').USAddress.City.contents[0]).encode('utf-8')
        except AttributeError:
            try:
                city = (NPOsoup.find('City').contents[0]).encode('utf-8')
            except AttributeError:
                city = None

    ## State
    try:
        state = (NPOsoup.find('Filer').USAddress.StateAbbreviationCd.contents[0]).encode('utf-8')
    except AttributeError:
        try:
            state = (NPOsoup.find('Filer').USAddress.State.contents[0]).encode('utf-8')
        except AttributeError:
            try:
                state = (NPOsoup.find('State').contents[0]).encode('utf-8')
            except AttributeError:
                state = None

    ## Tax Year
    try:
        tax_year = (NPOsoup.find('TaxYr').contents[0]).encode('utf-8')
    except AttributeError:
        tax_year = 0

    ## Activity
    try:
        activity = (NPOsoup.find('ActivityOrMissionDesc').contents[0]).encode('utf-8')
    except AttributeError:
        activity = 0

    ## Year formed
    try:
        year_formed = (NPOsoup.find('FormationYr').contents[0]).encode('utf-8')
    except AttributeError:
        year_formed = 0

    ## Volunteers
    try:
        volunteer_ct = (NPOsoup.find('TotalVolunteersCnt').contents[0]).encode('utf-8')
    except AttributeError:
        volunteer_ct = 0

    ## Employee Cnt
    try:
        employee_ct = (NPOsoup.find('TotalEmployeeCnt').contents[0]).encode('utf-8')
    except AttributeError:
        employee_ct = 0


    ## REVENUES ##

    ## Campaigns (Part VIII, line 1a)
    try:
        rev_campaigns = (NPOsoup.find('FederatedCampaignsAmt').contents[0]).encode('utf-8')
    except AttributeError:
        rev_campaigns = 0

    ## Membership Dues (Part VIII, line 1b)
    try:
        rev_membership = (NPOsoup.find('MembershipDuesAmt').contents[0]).encode('utf-8')
    except AttributeError:
        rev_membership = 0

    ## Fundraising Events (Part VIII, line 1c)
    try:
        rev_fundraising = (NPOsoup.find('FundraisingAmt').conte
nts[0]).encode('utf-8')
    except AttributeError:
        rev_fundraising = 0

    ## Government Grants (Part VIII, line 1e)
    try:
        rev_govgrants = (NPOsoup.find('GovernmentGrantsAmt').co
ntents[0]).encode('utf-8')
    except AttributeError:
        rev_govgrants = 0

    ## Other gifts (Part VIII, line 1f)
    try:
        rev_other = (NPOsoup.find('AllOtherContributionsAmt').c
ontents[0]).encode('utf-8')
    except AttributeError:
        rev_other = 0

    ## Program Service Revenue (Part VIII, line 2g)
    try:
        rev_progserv = (NPOsoup.find('TotalProgramServiceRevenu
eAmt').contents[0]).encode('utf-8')
    except AttributeError:
        rev_progserv = 0


    ## Net from Fundraising Events (Part VIII, line 3c)
    try:
        rev_netfundraising = (NPOsoup.find('NetIncmFromFundrais
ingEvtGrp/TotalRevenueColumnAmt').contents[0]).encode('utf-8')
    except AttributeError:
        rev_netfundraising = 0


    ## CY Total Revenue (Part VIII, line 1c)
    total_revenue = None
    revenue_fields = ['TotalRevenueCurrentYear',
'TotalRevenue', 'TotalRevenueAmt','CYTotalRevenueAmt']
    for revenue_field in revenue_fields:
        try:
            total_revenue = (NPOsoup.find(revenue_field).conten
ts[0]).encode('utf-8')
            break
        except AttributeError:
            pass
    if total_revenue == None:
        total_revenue = 0

    ## PY Total Revenue (Part VIII, line 1c)
    try:
        total_revenuePY = (NPOsoup.find('PYTotalRevenueAmt').co
ntents[0]).encode('utf-8')
    except AttributeError:
        total_revenuePY = 0


    ## EXPENSES
    ## Total Grant Expenses  (Part IX, line 25B)
    try:
        exp_grants =
(NPOsoup.find('CYGrantsAndSimilarPaidAmt').contents[0]).encode('
tf-8')
    except AttributeError:
        exp_grants = 0

    ## Total Service Expenses  (Part IX, line 25B)
    try:
        exp_progserv = (NPOsoup.find('CYBenefitsPaidToMembersAm
t').contents[0]).encode('utf-8')
    except AttributeError:
        exp_progserv = 0


    ## Total Management Expenses (Part IX, line 25C)
    try:
        exp_management = (NPOsoup.find('CYSalariesCompEmpBnftPa
idAmt').contents[0]).encode('utf-8')
    except AttributeError:
        exp_management = 0

    ## Total Fundraising Expensens (Part IX, line 25D)
    try:
        exp_fundraising = (NPOsoup.find('CYTotalFundraisingExpe
nseAmt').contents[0]).encode('utf-8')
    except AttributeError:
        exp_fundraising = 0

    ## CY Total Expenses
    total_expenses = None
    expense_fields = ['CYTotalExpensesAmt','TotalExpenses','Tot
alExpensesAmt']
    for expense_field in expense_fields:
        try:
            total_expenses = (NPOsoup.find(expense_field).conte
nts[0]).encode('utf-8')
            break
        except AttributeError:
            pass
    if total_expenses == None:
        total_expenses = 0

    ## PY Total Expenses (Part VIII, column (A), line 25)
    try:
        total_expensesPY =
(NPOsoup.find('PYTotalExpensesAmt').contents[0]).encode('utf-
8')
    except AttributeError:
        total_expensesPY = 0


    ## COMPENSANTIONS

    ## Total Compensations (PART VII)
    try:
        total_compensations = (NPOsoup.find('TotalReportableCom
pFromOrgAmt').contents[0]).encode('utf-8')
    except AttributeError:
        total_compensations = 0

    ## Compensations more than $100k (Part VII, line 2)
    try:
        comp_more100k = (NPOsoup.find('IndivRcvdGreaterThan100K
Cnt').contents[0]).encode('utf-8')
    except AttributeError:
        comp_more100k = 0


    ## Net Assessts of Fund Balances *End of Year*
    try:
        net_assets = (NPOsoup.find('NetAssetsOrFundBalancesEOYA
mt').contents[0]).encode('utf-8')
    except AttributeError:
        net_assets = 0

    ## Political Campaing Activity (NPO engage in direct or ind
irect political campaign
    #  activities on behalf of or in opposition to candidates f
or public office?)
    try:
        pol_act = (NPOsoup.find('PoliticalCampaignActyInd').con
tents[0]).encode('utf-8')
        if pol_act == 'false' or 'False':
            pol_act = 0
        if pol_act == 'true' or 'True' or 'X':
            pol_act = 1
        else:
            pol_act = 2
    except AttributeError:
        pol_act = 2 #Not reported

    ## Lobbying Activities (NPO engage in lobbying activities,
 or have a section
    #  501(h) election in effect during the tax year?)
    try:
        lob_act = (NPOsoup.find('LobbyingActivitiesInd').conten
ts[0]).encode('utf-8')
        if lob_act == 'false' or 'False':
            lob_act = 0
        if lob_act == 'true' or 'True' or 'X':
            lob_act = 1
        else:
            lob_act = 2
    except AttributeError:
        lob_act = 2 #Not reported


    ## Foreign office (NPO have office, employees, or agents ou
tside of the United States?)
    try:
        foreign_office = (NPOsoup.find('ForeignOfficeInd').cont
ents[0]).encode('utf-8')
        if foreign_office == 'false' or 'False':
            foreign_office = 0
        if foreign_office == 'true' or 'True' or 'X':
            foreign_office = 1
        else:
            foreign_office = 2
    except AttributeError:
        foreign_office = 2 #Not reported
    ## Foreign Fundraising
    #  NPO aggregate revenues or expenses of more than $10,000 from grantmaking, fundraising,
    #  business, investment, and program service activities outside the United States, or
    #  aggregate foreign investments valued at $100,000 or more?
    try:
        foreign_fundraising = (NPOsoup.find('ForeignActivitiesInd').contents[0]).encode('utf-8')
        if foreign_fundraising == 'false' or 'False':
            foreign_fundraising = 0
        if foreign_fundraising == 'true' or 'True' or 'X':
            foreign_fundraising = 1
        else:
            foreign_fundraising = 2
    except AttributeError:
        foreign_fundraising = 2 #Not reported


    ## Assistance to Foreign Individuals
    #  more than $5,000 of grants or other assistance to or for any foreign organization?
    #  more than $5,000 of aggregate grants or other assistance to or for foreign individuals?
    foreign_assist = None
    foreign_assist_fields= ['MoreThan5000KToOrgInd', 'MoreThan5000KToIndividualsInd']
    for foreign_assist_field in foreign_assist_fields:
        try:
            foreign_assist = (NPOsoup.find(foreign_assist_field).contents[0]).encode('utf-8')
            if foreign_assist == 'false' or 'False':
                foreign_assist = 0
            if foreign_assist == 'true' or 'True' or 'X':
                    foreign_assist = 1
            else:
                foreign_assist = 2
            break
        except AttributeError:
            pass
    if foreign_assist == None:
        foreign_assist = 2

    NPO_meta.append([EIN,contract_term, tax_status,org_name, city, state, tax_year, activity,
                     year_formed,volunteer_ct, employee_ct, rev_campaigns,
                     rev_membership, rev_fundraising, rev_govgrants, rev_other,
                     rev_progserv, rev_netfundraising, total_revenue, total_revenuePY,
                     exp_grants, exp_progserv, exp_management, exp_fundraising,
                     total_expenses, total_compensations, comp_more100k, net_assets,
                     pol_act,lob_act, foreign_office, foreign_fundraising,
                     foreign_assist])

    k+=1
    if k==1000:
        break

NPO_meta_df = pd.DataFrame(NPO_meta)
NPO_meta_df.to_csv('input/NPO_meta.csv', index=False)
print("Metafile saved to NPO_meta.csv")
```

```
▶ Retrieving records.
Metafile saved to NPO_meta.csv
```

In [ ]: