NCCS Data Exploration

Mayleen

2025-05-20

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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
```

Loading Data

CORE Data Dictionary

```
data_dict <- read.csv("DATA_DICTS/CORE-HRMN_dd.csv")
head(data_dict)
```

```
##               variable_name                       variable_description
## 1                DUP_RTRN_X                 Indicates duplicate return
## 2                      EIN2                            Reformatted EIN
## 3 F9_00_EXEMPT_STAT_501C3_X         Indicates a 501(c)(3) organization
## 4    F9_00_GROUP_EXEMPT_NUM                     Group exemption number
## 5       F9_00_ORG_ADDR_CITY   Address of Filing Organization (US City)
## 6         F9_00_ORG_ADDR_L1 Address of Filing Organization (US Line 1)
##   variable_source                                          form_location
## 1            <NA>                                                   <NA>
## 2            <NA>                                                   <NA>
## 3              HD  F990-EZ-PART-00-SECTION-J;\nF990-PC-PART-00-SECTION-I
## 4              HD F990-EZ-PART-00-SECTION-F;\nF990-PC-PART-00-SECTION-HC
## 5              HD  F990-EZ-PART-00-SECTION-C;\nF990-PC-PART-00-SECTION-C
## 6              HD  F990-EZ-PART-00-SECTION-C;\nF990-PC-PART-00-SECTION-C
##   variable_coverage form_scope variable_datatype
## 1         2012-2022   PC-501C3           integer
## 2         2012-2022   PC-501C3         character
## 3         2012-2019   PC-501C3           integer
## 4         2012-2013   PC-501C3           integer
## 5         2012-2019   PC-501C3         character
## 6         2012-2019   PC-501C3         character
```

```r
#glimpse(data_dict)
#summarise(data_dict)
```

CORE Data

Year: 2022 Type: CHARITIES Scope: 990 + 990EZ filers

```r
core_data_2022 <- read_csv("CORE/CORE-2022-501C3-CHARITIES-PZ-HRMN.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 50844 Columns: 263
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (79): EIN2, F9_04_TAX_EXEMPT_BOND_ISSUER_X, F9_04_TRANSAC_PY_X, F9_04_L...
## dbl (182): BMF_SUBSECTION_CODE, F9_09_EXP_FEE_SVC_ACC_TOT, F9_10_LIAB_ACC_PA...
## lgl   (2): F9_05_DAF_EXCESS_BIZ_HOLDING_X, F9_04_HOSPITAL_AFS_X
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(core_data_2022)
```

```
## # A tibble: 6 x 263
##   EIN2         BMF_SUBSECTION_CODE F9_09_EXP_FEE_SVC_AC~1 F9_10_LIAB_ACC_PAYAB~2
##   <chr>                      <dbl>                  <dbl>                  <dbl>
## 1 EIN-01-0147~                   3                      0                   7057
## 2 EIN-01-0199~                   3                   4236                      0
## 3 EIN-01-0211~                   3                   7992                      0
## 4 EIN-01-0211~                   3                    900                      0
## 5 EIN-01-0211~                   3                   3500                      0
## 6 EIN-01-0211~                   3                  31045                1195273
## # i abbreviated names: 1: F9_09_EXP_FEE_SVC_ACC_TOT,
## #   2: F9_10_LIAB_ACC_PAYABLE_EOY
## # i 259 more variables: F9_10_ASSET_ACC_NET_EOY <dbl>,
## #   F9_04_TAX_EXEMPT_BOND_ISSUER_X <chr>, F9_09_EXP_AD_PROMO_TOT <dbl>,
## #   F9_04_TRANSAC_PY_X <chr>, F9_09_EXP_BEN_PAID_MEMB_TOT <dbl>,
## #   F9_10_NAFB_CAP_STCK_EOY <dbl>, F9_04_LTD_X <chr>,
## #   F9_04_SCHED_O_REQ_X <chr>, F9_09_EXP_COMP_DSQ_PERS_TOT <dbl>, ...
```

```r
#glimpse(core_data_2022)
#summarise(core_data_2022)
```

Year: 1989 Type: CHARITIES Scope: 990 + 990EZ filers

```r
core_data_1989 <- read_csv("CORE/CORE-1989-501C3-CHARITIES-PZ-HRMN.csv")
```

```
## Rows: 138982 Columns: 55
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (9): F9_00_ORG_ADDR_L1, F9_00_ORG_ADDR_CITY, F9_00_ORG_NAME_L1, MISSION...
## dbl (46): F9_10_ASSET_TOT_BOY, F9_10_ASSET_TOT_EOY, F9_09_EXP_COMP_DTK_TOT, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(core_data_1989)
```

```
## # A tibble: 6 x 55
##   F9_00_ORG_ADDR_L1  F9_10_ASSET_TOT_BOY F9_10_ASSET_TOT_EOY F9_00_ORG_ADDR_CITY
##   <chr>                            <dbl>               <dbl> <chr>
## 1 PO BOX 1441                     256700              297477 BANGOR
## 2 S HIGH ST                      5867159             5374950 BRIDGETON
## 3 309 BLACK POINT R~              122304              143839 PROUTS NECK
## 4 50 CONGRESS ST                  677232              670662 RUMFORD
## 5 PO BOX 417                     6121132             6271322 BOOTHBAY HARBOR
## 6 PO BOX 287                    14435587            15074348 BELFAST
## # i 51 more variables: F9_09_EXP_COMP_DTK_TOT <dbl>, F9_08_REV_CONTR_TOT <dbl>,
## #   F9_08_REV_CONTR_MEMBSHIP_DUE <dbl>, F9_00_ORG_EIN <dbl>,
## #   SC_02_EXP_GRASS_M_NONTAX_FILEORG <dbl>,
## #   SC_02_EXP_LOB_M_NONTAX_FILEORG <dbl>, F9_09_EXP_TOT_TOT <dbl>,
## #   F9_00_TAX_YEAR <dbl>, F9_01_NAFB_TOT_EOY <dbl>,
## #   F9_09_EXP_FEE_SVC_FUNDR_TOT <dbl>, F9_08_REV_OTH_EVNT_NET_TOT <dbl>,
## #   F9_00_GROUP_EXEMPT_NUM <dbl>, F9_08_REV_OTH_INV_COST_GOODS <dbl>, ...
```

```r
#glimpse(core_data_1989)
#summarise(core_data_1989)
#unique(core_data_1989$MISSION_NTEE)
```

Year: 2014 Type: CHARITIES Scope: 990 + 990EZ filers

```r
core_data_2014 <- read_csv("CORE/CORE-2014-501C3-CHARITIES-PZ-HRMN.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 362393 Columns: 289
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (18): EIN2, F9_04_TRANSAC_ENGAGED_X, F9_05_UBIZ_FORM_990T_FILED_X, F9_0...
## dbl (113): BMF_SUBSECTION_CODE, F9_09_EXP_COMP_DTK_TOT, F9_08_REV_OTH_INV_CO...
## lgl (158): F9_09_EXP_FEE_SVC_ACC_TOT, F9_10_LIAB_ACC_PAYABLE_EOY, F9_10_ASSE...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(core_data_1989)
```

```
## # A tibble: 6 x 55
##   F9_00_ORG_ADDR_L1  F9_10_ASSET_TOT_BOY F9_10_ASSET_TOT_EOY F9_00_ORG_ADDR_CITY
##   <chr>                            <dbl>               <dbl> <chr>
## 1 PO BOX 1441                     256700              297477 BANGOR
## 2 S HIGH ST                      5867159             5374950 BRIDGETON
## 3 309 BLACK POINT R~              122304              143839 PROUTS NECK
## 4 50 CONGRESS ST                  677232              670662 RUMFORD
## 5 PO BOX 417                     6121132             6271322 BOOTHBAY HARBOR
## 6 PO BOX 287                    14435587            15074348 BELFAST
## # i 51 more variables: F9_09_EXP_COMP_DTK_TOT <dbl>, F9_08_REV_CONTR_TOT <dbl>,
## #   F9_08_REV_CONTR_MEMBSHIP_DUE <dbl>, F9_00_ORG_EIN <dbl>,
## #   SC_02_EXP_GRASS_M_NONTAX_FILEORG <dbl>,
## #   SC_02_EXP_LOB_M_NONTAX_FILEORG <dbl>, F9_09_EXP_TOT_TOT <dbl>,
## #   F9_00_TAX_YEAR <dbl>, F9_01_NAFB_TOT_EOY <dbl>,
## #   F9_09_EXP_FEE_SVC_FUNDR_TOT <dbl>, F9_08_REV_OTH_EVNT_NET_TOT <dbl>,
## #   F9_00_GROUP_EXEMPT_NUM <dbl>, F9_08_REV_OTH_INV_COST_GOODS <dbl>, ...
```

```
#glimpse(core_data_1989)
#summarise(core_data_1989)
#unique(core_data_1989$MISSION_NTEE)
```

Unified BMF Data

```
unified_bmf <- read_csv("CORE/BMF_UNIFIED_V1.1.csv")
```

```
## Rows: 3462997 Columns: 49
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (22): EIN2, NTEE_IRS, NTEE_NCCS, NTEEV2, NCCS_LEVEL_1, NCCS_LEVEL_2, NCC...
## dbl (27): EIN, F990_TOTAL_REVENUE_RECENT, F990_TOTAL_INCOME_RECENT, F990_TOT...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(unified_bmf)
```

```
## # A tibble: 6 x 49
##   EIN2      EIN NTEE_IRS NTEE_NCCS NTEEV2 NCCS_LEVEL_1 NCCS_LEVEL_2 NCCS_LEVEL_3
##   <chr>   <dbl> <chr>    <chr>     <chr>  <chr>        <chr>        <chr>
## 1 EIN-00~     0 Z99      <NA>      <NA>   501CX NONPR~ O            UN
## 2 EIN-00~     1 B43      B43       UNI-B~ 501C3 CHARI~ O            ED
## 3 EIN-00~     4 B90      B90       EDU-B~ 501C3 CHARI~ O            ED
## 4 EIN-00~     5 A50      A50       ART-A~ 501C3 CHARI~ O            AR
## 5 EIN-00~  3154 C36      C36       ENV-C~ 501C3 CHARI~ O            EN
## 6 EIN-00~  4101 N65      N65       HMS-N~ 501C3 PRIVA~ S            HS
## # i 41 more variables: F990_TOTAL_REVENUE_RECENT <dbl>,
## #   F990_TOTAL_INCOME_RECENT <dbl>, F990_TOTAL_ASSETS_RECENT <dbl>,
## #   F990_ORG_ADDR_CITY <chr>, F990_ORG_ADDR_STATE <chr>,
## #   F990_ORG_ADDR_ZIP <chr>, F990_ORG_ADDR_STREET <chr>,
## #   CENSUS_CBSA_FIPS <dbl>, CENSUS_CBSA_NAME <chr>, CENSUS_BLOCK_FIPS <dbl>,
## #   CENSUS_URBAN_AREA <chr>, CENSUS_STATE_ABBR <chr>, CENSUS_COUNTY_NAME <chr>,
## #   ORG_ADDR_FULL <chr>, ORG_ADDR_MATCH <chr>, LATITUDE <dbl>, ...
```

```
#glimpse(unified_bmf)
#summarise(unified_bmf)
```

Merging Unified BMF to CORE

Checking for duplicates

- in bmf, how many of those duplicates have the same organization name?
- in bmf duplicates, what are the patterns in which columns they differ in value?
- figure out if we care about that info–if its not info we care about, then we can keep one of the two without worry?
- do I need to merge right now? How important is the info in BMF file to current task?

```
test <- unified_bmf |>
    count(EIN2) |>
    filter(n > 2)
```

```
unified_bmf |>
    filter(EIN2 == "EIN-00-0000000")
```

```
## # A tibble: 16 x 49
##    EIN2     EIN NTEE_IRS NTEE_NCCS NTEEV2 NCCS_LEVEL_1 NCCS_LEVEL_2 NCCS_LEVEL_3
##    <chr>  <dbl> <chr>    <chr>     <chr>  <chr>        <chr>        <chr>
##  1 EIN-0~     0 Z99      <NA>      <NA>   501CX NONPR~ O            UN
##  2 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
##  3 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
##  4 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
##  5 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
##  6 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
##  7 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
##  8 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
##  9 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
## 10 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
## 11 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
## 12 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
## 13 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
## 14 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
## 15 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
## 16 EIN-0~     0 <NA>     <NA>      <NA>   <NA>         <NA>         <NA>
## # i 41 more variables: F990_TOTAL_REVENUE_RECENT <dbl>,
## #   F990_TOTAL_INCOME_RECENT <dbl>, F990_TOTAL_ASSETS_RECENT <dbl>,
## #   F990_ORG_ADDR_CITY <chr>, F990_ORG_ADDR_STATE <chr>,
## #   F990_ORG_ADDR_ZIP <chr>, F990_ORG_ADDR_STREET <chr>,
## #   CENSUS_CBSA_FIPS <dbl>, CENSUS_CBSA_NAME <chr>, CENSUS_BLOCK_FIPS <dbl>,
## #   CENSUS_URBAN_AREA <chr>, CENSUS_STATE_ABBR <chr>, CENSUS_COUNTY_NAME <chr>,
## #   ORG_ADDR_FULL <chr>, ORG_ADDR_MATCH <chr>, LATITUDE <dbl>, ...
```

```
# which EIN2 have more than one entry?
core_data_2022 |>
    count(EIN2) |>
    filter(n > 1)
```

```
## # A tibble: 6 x 2
##   EIN2               n
##   <chr>          <int>
## 1 EIN-47-4469864     2
## 2 EIN-63-1226692     2
## 3 EIN-82-5453790     2
## 4 EIN-85-0306835     2
## 5 EIN-85-0772092     2
## 6 EIN-86-3470829     2
```

```
test <- core_data_1989 |>
    count(EIN2) |>
    filter(n > 2)

#head(test)
#typeof(test)
```

```
#test$EIN2
```

Exploring duplicates: BMF

Goal: Script to check what kinds of columns are different in duplicate rows

Return a data frame with the following variables: - col_name : variable name from data file - instances : number of times that a duplicate row differs in this column

SEE FILE: reviewing_duplicate_EIN.R for this script

```
# differences in logical comparison operators
test1 <- c(NA, TRUE, FALSE, TRUE, FALSE, TRUE)
test2 <- c(NA, NA, NA, TRUE, FALSE, FALSE)
test3 <- c(NA, NA, NA, TRUE, FALSE, FALSE)
test_compare2 <- test1 == test2
test_compare3 <- ((test1 == test2) & (test1 == test3) & (test2 == test3))

test_compare2_is <- is_equal(test1, test2)
test_compare3_is <- (is_equal(test1, test2) & is_equal(test1, test3) & is_equal(test2,test3))
```

```
library(tidyverse)
library(data.table)
library(dplyr)
source("reviewing_duplicate_EIN.R")

unified_bmf <- read.csv("CORE/BMF_UNIFIED_V1.1.csv")

vars_to_keep <- c("EIN2", "NTEE_NCCS", "NTEEV2", "NCCS_LEVEL_1") # "NTEE_IRS", "NCCS_LEVEL_2", "NCCS_LE

# Replace any empty strings '' with NA values
unified_bmf <- unified_bmf |> mutate_if(is.character, ~na_if(.,''))
info_unified_bmf <- duplicateEIN2_info(unified_bmf)
head(info_unified_bmf)

bmf_subset <- unified_bmf[vars_to_keep]
head(bmf_subset)
```

```
# drop rows with EIN = 00-0000000
bmf_subset <- bmf_subset[!(bmf_subset$EIN2 %in% "EIN-00-0000000"),]
n_before_removing_dupes <- nrow(bmf_subset)

# get list of EIN2 that are repeated
dupe_list <- bmf_subset |>
  count(EIN2) |>
  filter(n > 1)

n_dupes <- nrow(dupe_list)

# info on repeated EINs before dropping duplicated rows
# bmf_dupe_info_before_rem_dup <- duplicateEIN2_info(bmf_subset)
# head(bmf_dupe_info_before_rem_dup, 10)

# remove any rows that are exactly duplicated
bmf_sub_table <- data.table(bmf_subset)
setkeyv(bmf_sub_table, "EIN2")
```

```r
uniq_bmf_subset <- subset(unique(bmf_sub_table))
n_after_removing_dupes <- nrow(uniq_bmf_subset)

# info on repeated EINs after dropping duplicated rows
bmf_dupe_info <- duplicateEIN2_info(uniq_bmf_subset)
head(bmf_dupe_info, 10)
```

```r
library(dplyr)

# Find duplicates and only keep rows with duplicate EIN2
ein2_dups <- uniq_bmf_subset %>%
  group_by(EIN2) %>%
  filter(n() > 1)
```

Exploring duplicates: CORE