FY2021 Analysis - Methodology for Fiscal Futures

Alea Wilbur

October 2022

Inspecting new FY21 data files

Data files for closed years have been obtained from IOC. Source spending data is at the fund-agency-object level and source revenue data is at the fund-agency-source level.

Revenue File:
- 671 Fund Numbers
- 1182 Revenue sources
- 84 Agencies

Expenditure File:
- 709 Funds
- 106 Agencies - 98 Division Numbers, 315 Division names

Variables

Using the comptrollers variables:

Detail Object is a department classification used by the State to group expenses.

Object: e.g. 1100 - Personal Services; 880 - Debt Service; 8813 - Current Maturities-Interest

Group: RE-reimbursements; TR-Transfers; 9-Other

Category: T - Taxable Bond Fund Payments, 4 - Nonprofit organizations grants; 7 - Personal Services Related

Type: T - Statutory Transfers; 1 - Operations; 6 - Permanent Improvements; 8-Debt Service; 9-Refunds

Class: ex. 402 - Income Tax Refunds, 407 - Sales Tax Refunds

Appropriation Category: 8800 - Debt Service; 1129 - Employee Retirement Paid State

Data Creation and Cleaning

1. Do the FOIA request
2. In a week or so, they send the expenditure and revenue data as excel files.
3. Checks whether there are any new agencies, re-used funds etc. Create a list of funds, agencies, fund names, etc. for the new year and compare it to the immediate prior year to identify new funds.
4. Update the funds_ab_in file which shows the use of funds. Use criteria to determine if the new funds should be in or out of the all-funds frame.
5. Then, download the excel files that are sent to you.

6. Open and change the names to be consistent with other files such as AGENCYNAME–> agency_name

Combine past years: All revenue files are in a `revenue` folder that I reference when I set the working directory. When adding new fiscal years, put the the newest year of data for revenue and expenditures in their respective folders.

Pre-FY2022

The code chunk below takes the .dta files for all fiscal years before FY 2022 and binds them together. Variable names were manually changed by past researchers so that they were consistent across years.

Reads in dta file and leaves fund as a character. No longer have to worry about preserving leading zeros in categories like the fund numbers. State code used to force fund, source, and from_fund to be 4 digits long and preserve leading zeros and fund was 3 digits long with leading zeros.

Code below reads in the csv files created in chunks above (allrevfiles.csv and allrexpfiles.csv). These files contain all years of data combined into one file BEFORE any recoding is done. Do not use this file for summing categories because it is just an in between step before recoding revenue and expenditure categories.

```
# combined in past chunks called create-rev-csv and create-exp-csv

allrevfiles <- read_csv("allrevfiles22.csv") #combined but not recoded
allexpfiles <- read_csv("allexpfiles22.csv") #combined but not recoded
```

Normally, when your receive the new fiscal year files from the Comptrollers office, you will need to change the variable names so that they are consistent with past years. This is an example of reading in the new file and changing the variable names.

For FY 2022 and after, .dta files can be avoided entirely and .csv files and R code will be used.All files before this year had been saved and passed on as .dta files for Stata code before the transition to R in Fall 2022

Example code below: Read in excel file and rename columns so that it plays well with the other years' files.

```
read_xlsx("Fis_Fut_Rev_2022.xlsx") %>%
  rename(fy = 'FISCAL YEAR',
         fund = 'FUND #',
         fund_name = 'FUND NAME',
         agency = 'AGENCY #',
         agency_name = 'AGENCY NAME',
         source = 'REVENUE SOURCE #',
         source_name = 'REV SRC NAME',
         receipts = 'REVENUE YTD AMOUNT'
  ) %>%
  # do these come from funds_ab_whatever file?
  mutate(fund_cat = FIND_COLUMN, #create fund_cat column
         fund_cat_name = FIND_NAME) # create fund_cat_name column

read_xlsx("Fis_Fut_Exp_2021.xlsx") %>%
  rename(fy = 'FISCAL YEAR',
         fund = 'FUND #',
         fund_name = 'FUND NAME',
         agency = 'AGENCY #',
         agency_name = 'AGENCY NAME',
         appr_org = 'DIVISION',
         org_name = 'DIVISION NAME',
         obj_seq_type = 'APPROPRIATION #',
```

```
        wh_approp_name = 'APPROPRIATION NAME',
        exp_net_xfer = 'NET OF TRANS AMOUNT',
        expenditure = 'EXPENDED THRI 7/26/22'
  ) %>%
  # do these come from funds_ab_whatever file?
  mutate(data_source = "exp IOC Oct 2021",
        object = ,
        seq = ,
        type = ,
        fund_cat = FIND_COLUMN, #create fund_cat column
        fund_cat_name = FIND_NAME) # create fund_cat_name column
```

Identify new and reused funds for newest fiscal year. Recode funds to take into account different fund numbers/names over the years. Update fund_ab_in_2021.xlsx with any changes from previous fiscal year.

Clarify and add steps for identifying new and reused funds.

For funds that were reused once, a 9 replaces the 0 as the first digit. If reused twice, then the first two values are 10.
- Ex. 0350 –> 9350 because its use changed.
- Ex. 0367 becomes 10367 because its use has changed twice now. There was fund 0367 originally, then its use changed and it was recoded as 9367, and now it changed again so it is a 10367.

```
# if first character is a 0, replace with a 9

rev_1998_2022 <- allrevfiles %>%
      mutate(fund = ifelse(fy < 2002 & fund %in% c("0730", "0241", "0350", "0367", "0381", "0382", "052

  mutate(fund = ifelse(fy < 2008 & fund %in% c("0027", "0033", "0037", "0058", "0062", "0066", "0075",
"0068", "0076", "0115", "0119", "0168", "0182", "0199", "0241", "0307", "0506", "0509", "0513"), str_rep

  mutate(fund = ifelse(fy < 2016 & fund %in% c("0263", "0399", "0409"), str_replace(fund, "0","9"), fun

  mutate(fund =  ifelse(fy < 2017 & fund == "0364", str_replace(fund, "0","9"), fund)) %>%

  mutate(fund =  ifelse(fy < 2018 & fund %in% c("0818", "0767", "0671", "0593", "0578"), str_replace(fu

  mutate(fund = ifelse(fy>1999 & fy < 2018 & fund == "0231", "10231", fund) ) %>%

  mutate(fund = ifelse(fy < 2019 & fund %in% c("0161", "0489", "0500", "0612", "0893", "0766"), str_rep

  mutate(fund =  ifelse(fy < 2020 & fund %in% c("0254", "0304", "0324", "0610", "0887", "0908", "0939",

  mutate(fund =  ifelse(fy < 2021 & fund %in% c("0255", "0325", "0348", "0967", "0972"), str_replace(fu
```

Expenditure recoding:

```
# if first character is a 0, replace with a 9

exp_1998_2022 <- allexpfiles %>%
      mutate(fund = ifelse(fy < 2002 & fund %in% c("0730", "0241", "0350", "0367", "0381", "0382", "052
```

```
  mutate(fund = ifelse(fy < 2008 & fund %in% c("0027", "0033", "0037", "0058", "0062", "0066", "0075",
"0068", "0076", "0115", "0119", "0168", "0182", "0199", "0241", "0307", "0506", "0509", "0513"), str_rep

  mutate(fund = ifelse(fy < 2016 & fund %in% c("0263", "0399", "0409"), str_replace(fund, "0","9"), fun

  mutate(fund =  ifelse(fy < 2017 & fund == "0364", str_replace(fund, "0","9"), fund)) %>%

  mutate(fund =  ifelse(fy < 2018 & fund %in% c("0818", "0767", "0671", "0593", "0578"), str_replace(fu
  mutate(fund = ifelse(fy>1999 & fy < 2018 & fund == "0231", "10231", fund) ) %>%

  mutate(fund = ifelse(fy < 2019 & fund %in% c("0161", "0489", "0500", "0612", "0893", "0766"), str_rep

  mutate(fund =  ifelse(fy < 2020 & fund %in% c("0254", "0304", "0324", "0610", "0887", "0908", "0939",

  mutate(fund =  ifelse(fy < 2021 & fund %in% c("0255", "0325", "0348", "0967", "0972"), str_replace(fu
```

```
funds_ab_in_2022 = readxl::read_excel("C:/Users/aleaw/OneDrive/Desktop/PhD Fall 2021 - Spring 2022/Merr


exp_temp <- exp_1998_2022 %>%
  arrange(fund, fy) %>%
  filter(expenditure != 0) %>% # keeps everything that is not zero
# join  funds_ab_in_2021  to exp_temp
 left_join(funds_ab_in_2022, by = "fund")  # matches most recent fund number
```

- the initial combined years of data are saved as dataframes named `exp_1998_2022` and `rev_1998_2022`. These are then saved as exp_temp and rev_temp while recoding variables. This is BEFORE category groups are created and cleaned below. Only a temporary file, do not use for analysis.

Update Agencies: Early agencies replaced by successors

```
# recodes old agency numbers to consistent agency number
exp_temp <- exp_temp %>%
  mutate(agency = case_when(
    (agency=="438"| agency=="475" |agency == "505") ~ "440",
    # financial institution &  professional regulation &
     # banks and real estate  --> coded as  financial and professional reg
    agency == "473" ~ "588", # nuclear safety moved into IEMA
    (agency =="531" | agency =="577") ~ "532", # coded as EPA
    (agency =="556" | agency == "538") ~ "406", # coded as agriculture
    agency == "560" ~ "592", # IL finance authority (fire trucks and agriculture stuff)to state fire ma
    agency == "570" & fund == "0011" ~ "494",   # city of Chicago road fund to transportation
    TRUE ~ (as.character(agency))))
```

Modify Expenditure File

Tax refunds

Aggregate expenditures: Save tax refunds as negative revenue. Code refunds to match the rev_type codes (02=income taxes, 03 = corporate income taxes, 06=sales tax, 09=motor fuel tax, 24=insurance taxes and fees, 35 = all other tax refunds)

```
## negative revenue becomes tax refunds

tax_refund_long <- exp_temp %>%
 # fund != "0401" # removes State Trust Funds
  filter(fund != "0401" & (object=="9910"|object=="9921"|object=="9923"|object=="9925")) %>%
  # keeps these objects which represent revenue, insurance, treasurer,and financial and professional re
  mutate(refund = case_when(
    fund=="0278" & sequence == "00" ~ "02",   # for income tax refund
    fund=="0278" & sequence == "01" ~ "03", # tax administration and enforcement and tax operations bec
     fund == "0278" & sequence == "02" ~ "02",
    object=="9921" ~ "21", # inheritance tax and estate tax refund appropriation
    object=="9923" ~ "09", # motor fuel tax refunds
    obj_seq_type == "99250055" ~ "06", # sales tax refund
    fund=="0378" & object=="9925" ~ "24", # insurance privilege tax refund
    fund=="0001" & object=="9925" ~ "35", #all other taxes
      T ~ "CHECK")) # if none of the items above apply to the observations, then code them as CHECK


exp_temp <- left_join(exp_temp, tax_refund_long) %>%
  mutate(refund = ifelse(is.na(refund),"not refund", as.character(refund)))

tax_refund <- tax_refund_long %>%
  group_by(refund, fy)%>%
  summarize(refund_amount = sum(expenditure, na.rm = TRUE)/1000000) %>%
  pivot_wider(names_from = refund, values_from = refund_amount, names_prefix = "ref_") %>%
  mutate_all(~replace_na(.,0)) %>%
  arrange(fy)


# remove the items we recoded in tax_refund_long
exp_temp <- exp_temp %>% filter(refund == "not refund")

#should be 156 fewer observations
```

`tax_refund` will ultimately be removed from expenditure totals and instead subtracted from revenue totals (since they were tax refunds).

Pension Expenditures

State payments to the following pension systems:

- Teachers Retirement System (TRS)
- New POB bond in 2019: Accelerated Bond Fund paid benefits in advance as lump sum
- State Employee Retirement System (SERS)
- State University Retirement System (SURS)
- Judges Retirement System (JRS)
- General Assembly Retirement System (GARS)

Old code comments: "You also must consider pension obligated bonds (POB) - funded contributions."
- Pension obligation bonds (POBs) are taxable bonds that some state and local governments have issued as part of an overall strategy to fund the unfunded portion of their pension liabilities by creating debt.

DONE: Change POB fund == 0325 to 0 in fund_ab_YEAR file to exclude it.

Check what is included in pensions:

- object = 4431 catches most pension items
- object = 1298 is purchase of investments and is excluded from analysis except for a couple exceptions during 2010 and 2011

Modify exp_temp and move all pension contributions to their own group (901):

```
exp_temp <-  exp_temp %>%
  arrange(fund) %>%
  mutate(pension = case_when(

  # objects were weird for 2010 and 2011 for teacher and judge retirement system
  (object=="4431" & fund=="0473" &  (fy==2010 | fy==2011)) ~ 3, # teachers retirement system,
  (object=="1298" & (fy==2010 | fy==2011) & (fund=="0477" | fund=="0479" | fund=="0481")) ~ 3, #judges

  (object=="4431" | (object>"1159" & object<"1166") & fund != "0183" & fund != "0193"  ) ~ 1, # 4431 =
   # objects 1159 to 1166 are all considered Retirement by Comptroller
  # object == 1167 also appears to be Other Retirement but isn't used yet
                                        TRUE ~ 0))

table(exp_temp$pension) # same number of total observations > 0 as pension_check
```

```
##
##      0      1      3
## 159009   8945      8
```

~~POB-funded contributions to JRS, SERS, GARS, and TRS must be accounted for in a different way:Pension = 2 represents retirement pension payments paid for with POB-funded contributions that were excluded from the fiscal futures analysis by default (in_ff was 0 because bond financed funds are not sustainable cash flows) but should be included and added to the revenue side under "Other Revenues" in later steps.~~ <– No longer done.

Description of Pension Obligation Acceleration Bond at this link

```
exp_temp <- exp_temp %>%
  # change object for 2010 and 2011, retirement expenditures were bond proceeds
  mutate(object = ifelse((pension == 3 & in_ff == "0"), "4431", object)) %>%
  # changes weird teacher & judge retirement system  pensions object to normal pension object 4431
  mutate(pension =  ifelse(pension ==1 & in_ff == "0", 2, pension)) %>% # coded as 2 if it was supposed
  mutate(in_ff = ifelse((pension ==2 | pension ==3 ), "1", in_ff))

table(exp_temp$pension)
```

```
##
##      0      1      2      3
## 159009   8812    133      8
```

```
# all other pensions objects (1 and 3) codes get agency code 901 for State Pension Contributions
exp_temp <- exp_temp %>%
  mutate(agency = ifelse(pension>0, "901", as.character(agency)),
         agency_name = ifelse(agency == "901", "State Pension Contributions", as.character(agency_name))
```

- *DONE* add "& fund !=0183" to teacher retirement pensions code below - FY22 AWM
  - fund = 0183 from 1999 to 2005, org_name = Serve America, fund_name is post traumatic stress awareness

Drop Interfund transfers

- object == 1993 is for interfund cash transfers
- agency == 799 is for statutory transfers
- object == 1298 is for purchase of investments and is not spending EXCEPT for pensions in 2010 and 2011 (and were recoded already to object == "4431"). Over 168,000 observations remain.

```
transfers_drop <- exp_temp %>% filter(
  agency == "799" | # statutory transfers
           object == "1993" |  # interfund cash transfers
           object == "1298") # purchase of investments

exp_temp <- anti_join(exp_temp, transfers_drop)
```

State employee healthcare costs

State Employee Health Care = Sum of expenditures for "health care coverage as elected by members per state employees group insurance act." The payments are made from the Health Insurance Reserve Fund. We subtract the share that came from employee contributions. Employee contributions are not considered a revenue source or an expenditure in our analysis.

Fund = 0457 is "Group insurance premium", in_ff = 1 Fund = 0193 is "Local govt health insurance reserve", in=ff = 0 fund = 0477 is "Community College Health Insurance", in=ff = 0.
- had large amount in early years Fund = 0907 = health insurance reserve, in_ff = 1 Fund = 9939 is "group self-insurers' insolv", in_ff = 1 Fund = 0940 is Self-Insurers security, in_ff = 0 Fund = 0739 is Group Workers Comp Pool Insol, in_ff = 1

Employer contributions for group insurance are excluded to avoid double counting the cost of healthcare.

All employer contributions are coded as object = 1180.

- eehc = 0 means it is NOT a state healthcare cost but it is an employer contribution of some type to some fund
- eehc = 1 means it is a state employee healthcare cost and it is an employer contribution to health insurance

if observation is a group insurance contribution, then the expenditure amount is set to $0 (essentially dropped from analysis)

```
#if observation is a group insurance contribution, then the expenditure amount is set to $0 (essentiall
exp_temp <- exp_temp %>%
  mutate(eehc = ifelse(
    # group insurance contributions for 1998-2005 and 2013-present
   fund == "0001" & (object == "1180" | object =="1900") & agency == "416" & appr_org=="20", 0, 1) )%>%
  mutate(eehc = ifelse(
    # group insurance contributions for 2006-2012
    fund == "0001" & object == "1180" & agency == "478" & appr_org=="80", 0, eehc) )%>%
  mutate(expenditure = ifelse(eehc=="0", 0, expenditure)) %>%
  mutate(agency = case_when(   # turns specific items into State Employee Healthcare (agency=904)
      fund=="0907" & (agency=="416" & appr_org=="20") ~ "904",   # central management Bureau of benefit
      fund=="0907" & (agency=="478" & appr_org=="80") ~ "904",   # agency = 478: healthcare & family se
```

```
      TRUE ~ as.character(agency))) %>%
  mutate(agency_name = ifelse(agency == "904", "STATE EMPLOYEE HEALTHCARE", as.character(agency_name)),
         in_ff = ifelse( agency == "904", 1, in_ff),
         group = ifelse(agency == "904", "904", as.character(agency)))  # creates group variable

# Default group = agency number

healthcare_costs <- exp_temp %>% filter(group == "904")

healthcare_costs %>% group_by(fy) %>% summarise(healthcare_cost = sum(expenditure)) %>% arrange(-fy)
```

```
## # A tibble: 25 x 2
##       fy healthcare_cost
##    <dbl>           <dbl>
##  1  2022     3005209490.
##  2  2021     2881401513.
##  3  2020     2981817668.
##  4  2019     3151879062.
##  5  2018     5851728454.
##  6  2017     3320902172.
##  7  2016      478673987.
##  8  2015     2454666392.
##  9  2014     2625894499.
## 10  2013     2174926175.
## # ... with 15 more rows
```

```
exp_temp <- anti_join(exp_temp, healthcare_costs)

healthcare_costs_yearly <- healthcare_costs %>% group_by(fy, group) %>% summarise(healthcare_cost = sum
```

> This code chunk above for dealing with group insurance means that healthcare costs need to be added to expenditures after other group names are assigned. Then employee contributions/insurance premiums from the revenue side need to be subtracted from the total cost of employee healthcare for the net cost.

Local Transfers

Separate transfers to local from parent agencies that come from DOR(492) or Transportation (494). Treats muni revenue transfers as expenditures, not negative revenue.

The share of certain taxes levied state-wide at a common rate and then transferred to local governments. (Purely local-option taxes levied by specific local governments with the state acting as collection agent are not included.)

The five corresponding revenue items are:

- Local share of Personal Income Tax
- Local share of General Sales Tax
- Personal Property Replacement Tax on Business Income
- Personal Property Replacement Tax on Public Utilities
- Local share of Motor Fuel Tax - Transportation Renewal Fund 0952

> Completed: Add the mft mentioned in GOMB email to code

```
exp_temp <- exp_temp %>% mutate(
  agency = case_when(fund=="0515" & object=="4470" & type=="08" ~ "971", # income tax
                     fund=="0515" & object=="4491" & type=="08" & sequence=="00" ~ "971",
                     fund=="0802" & object=="4491" ~ "972", #pprt transfer
                     fund=="0515" & object=="4491" & type=="08" & sequence=="01" ~ "976", #gst to local
                     fund=="0627" & object=="4472"~ "976" ,
                     fund=="0648" & object=="4472" ~ "976",
                     fund=="0515" & object=="4470" & type=="00" ~ "976",
                    object=="4491" & (fund=="0188"|fund=="0189") ~ "976",
                     fund=="0187" & object=="4470" ~ "976",
                     fund=="0186" & object=="4470" ~ "976",
                    object=="4491" & (fund=="0413"|fund=="0414"|fund=="0415")  ~ "975", #mft to local
                  fund == "0952"~ "975", # Added Sept 29 2022 AWM. Transportation Renewal MFT
                    TRUE ~ as.character(agency)),

  agency_name = case_when(agency == "971"~ "INCOME TAX 1/10 TO LOCAL",
                          agency == "972" ~ "PPRT TRANSFER TO LOCAL",
                          agency == "976" ~ "GST TO LOCAL",
                          agency == "975" ~ "MFT TO LOCAL",
                          TRUE~as.character(agency_name)),
  group = ifelse(agency>"970" & agency < "977", as.character(agency), as.character(group)))


table(exp_temp$group)
```

```
##
##   101   102   103   105   107   108   109   110   112   115   120   131   140
##   583     3   240   155    89   193   137   129   162   128    17   386     7
##   155   156   167   201   210   275   280   285   290   295   310   330   340
##    75   117   118  1345    15   399     1   234   470  1185   213   205   819
##   350   360   370   402   406   416   418   420   422   425   426   427   440
##  4098  1743   803  1829  4660  3932  2420 10975  9668  1038  7614   779  3705
##   442   444   445   446   448   452   458   466   478   482   492   493   494
##   596 11357    23  1119    22   610   305   587  3063  5524  4129  1924  9550
##   497   503   506   507   509   510   511   517   520   524   525   526   527
##  2519   421    17   332    33    26  8954   128     5  1126    28   174    40
##   528   529   532   533   534   537   540   541   542   546   548   554   555
##  1838    18  5746     2     5   192    64  1305   174   873   264    26    25
##   557   558   559   562   563   564   565   567   568   569   571   574   575
##   208   280   245    19   699    17   198   176     2   450    65    80    85
##   576   578   579   580   583   585   586   587   588   589   590   591   592
##     1   233   438   327    21    43  5297   683  2681   597   166   188  1070
##   593   598   601   608   612   616   620   628   636   644   664   676   684
##   151    10   720   177   131   141    99   147   115   182   271   462   895
##   691   692   693   695   901   971   972   975   976
##   934   786     8   197  8953    25    25    84  1174
```

```
transfers_long <- exp_temp %>%
  filter(group == "971" |group == "972" | group == "975" | group == "976")

transfers <- transfers_long %>%
  group_by(fy, group ) %>%
  summarize(sum_expenditure = sum(expenditure)/1000000) %>%
```

```
  pivot_wider(names_from = "group", values_from = "sum_expenditure", names_prefix = "exp_" )

exp_temp <- anti_join(exp_temp, transfers_long)


dropped_inff_0 <- exp_temp %>% filter(in_ff == 0)

exp_temp <- exp_temp %>% filter(in_ff == 1) # drops in_ff = 0 funds AFTER dealing with net-revenue abov
```

Debt Service

Debt Service expenditures include interest payment on both short-term and long-term debt. We do not include escrow or principal payments.

Decision from Sept 30 2022:

We are no longer including short term principal payments as a cost; only interest on short term borrowing is a cost. Pre FY22 and the FY21 correction, we did include an escrow payment and principle payments as costs but not bond proceeds as revenues. This caused expenditures to be inflated because we were essentially counting debt twice - the principle payment and whatever the money was spent on in other expenditure categories, which was incorrect.

- Include interest for long-term debt that likely funds capital projects.

8813 interest **INCLUDE AS COST**
8811 is for principle **EXCLUDE from analysis**
8841 is for escrow payments **EXCLUDE from analysis**
8800 is for capital projects (including the Tollway) **INCLUDE as cost - Note: debt principle and interest are both included because they are combined in the data observations; bond proceeds are not considered a revenue source**

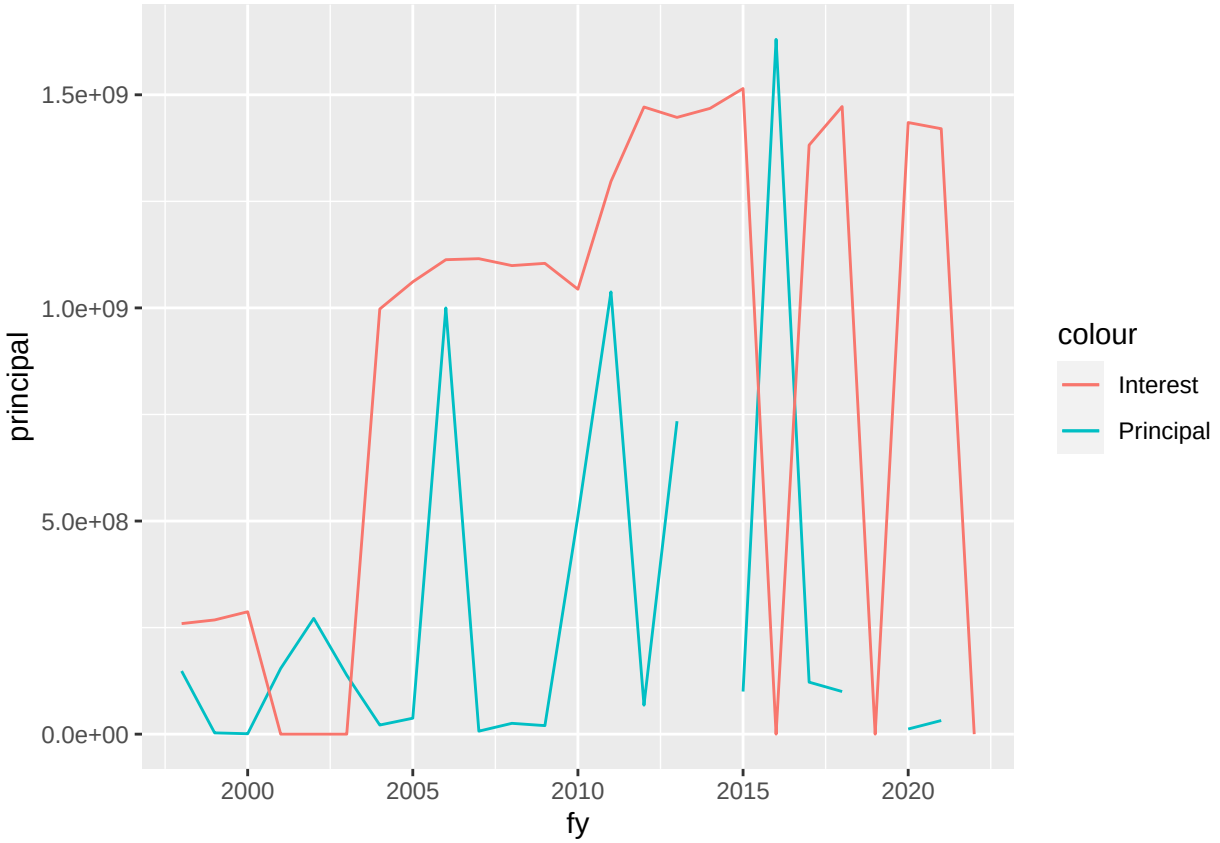
- ~~Exclude:~~ Bond principle payments: obj_seq_type == 88110008
  - **Exclude based on Merriman's meeting. Run it both ways to compare output to make sure we made the correct decision.**
- Exclude: Short term borrowing principle: obj_seq_type == 88110108
  - except for 2021 and 2022 which have GO Bond principle under this code????
  - based on the numbers, I think it is still short term principle with wrong appropriation name
- Include: General Obligation Bond Interest: obj_seq_type == 88130000 & 88130008
- Include: Interest for short-term borrowing: 88130108
- Exclude: Escrow payment == 88410008
- Include: Build IL Bonds principal AND interest
  - Tollway is obj_seq_type == 88000055, filter out fund == 0455 to avoid tollway debt
  - fund == 0455 is the IL State Toll Highway fund, items mostly for operations and maintenance

Filtering for interest on short term borrowing and GO bonds (8813_ _ _ _) and GO bond principal amounts (88130008).
- object == 8813 is for interest but obj_seq_type is used just to be more specific below.

```r
# GO bond principal and GO bond interest
GObond_debt <- exp_temp %>%
  filter(obj_seq_type == "88110008" |obj_seq_type == "88130000" | obj_seq_type == "88130008") %>%
  group_by(fy, obj_seq_type) %>%
  summarize(sum = sum(expenditure, na.rm=TRUE)) %>%
  pivot_wider(names_from = obj_seq_type, values_from = sum) %>%
  mutate(principal = `88110008`,
         interest = sum(`88130008`+`88130000`, na.rm = TRUE),
         ratio = (as.numeric(interest)/as.numeric(principal)))

GObond_debt %>% select(principal, interest, ratio) %>%
  mutate(across(principal:interest, ~format(., big.mark= ",", scientific = F)))
```

```
## # A tibble: 25 x 4
## # Groups:   fy [25]
##       fy principal     interest       ratio
##    <dbl> <chr>         <chr>          <dbl>
##  1  1998 148,066,200   259,385,877     1.75
##  2  1999 2,999,040     267,956,231    89.3
##  3  2000 1,000,000     287,154,654   287.
##  4  2001 154,166,026   0               0
##  5  2002 271,518,687   0               0
##  6  2003 138,351,231   0               0
##  7  2004 21,400,000    997,469,105    46.6
##  8  2005 37,400,000    1,061,205,268  28.4
##  9  2006 1,000,000,000 1,113,124,277   1.11
## 10  2007 6,995,000     1,115,485,186 159.
## # ... with 15 more rows
```

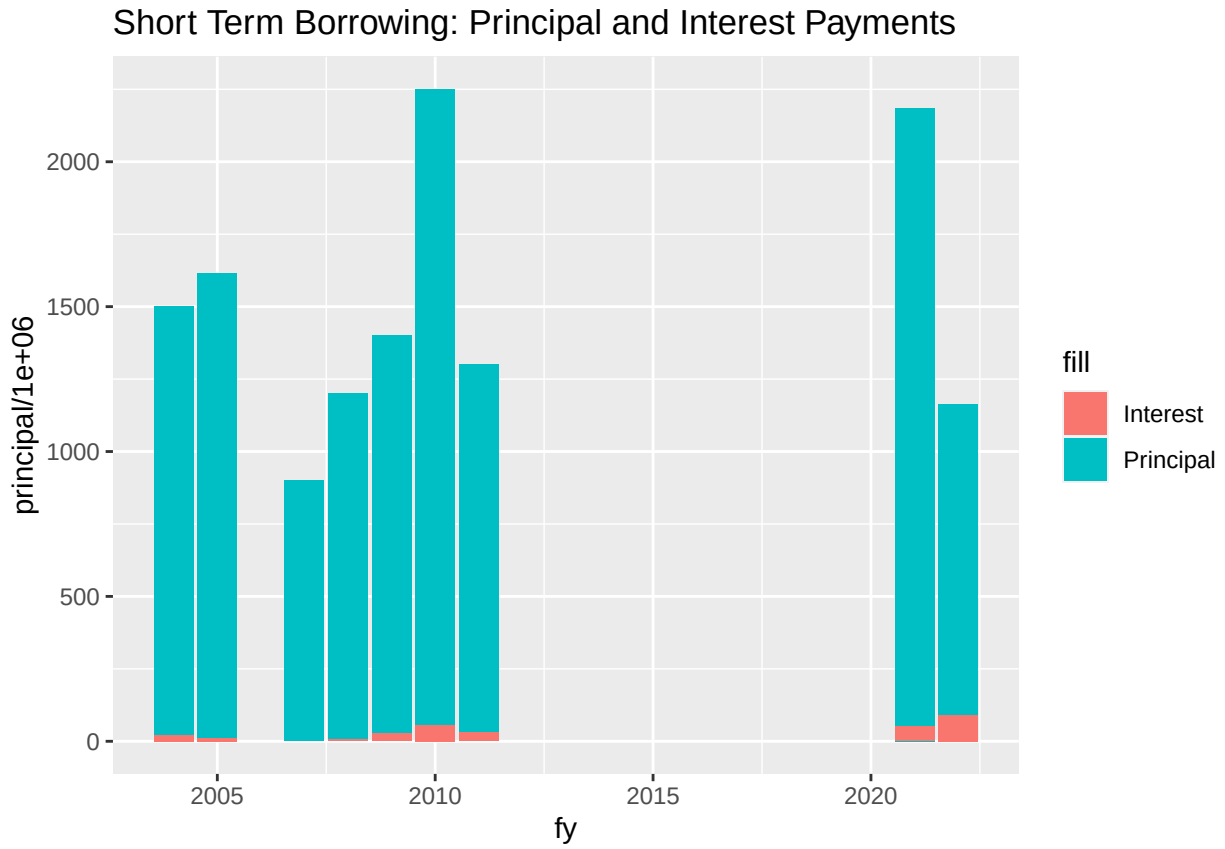
```r
GObond_debt %>% ggplot() +
  geom_line(aes(x=fy, y=principal, color = "Principal"))+
  geom_line(aes(x=fy, y=interest, color = "Interest"))
```




```
# short term borrowing, first observation is in 2004?
short_debt <- exp_temp %>%
  filter(obj_seq_type == 88110108 |obj_seq_type == 88130108) %>%
  group_by(fy, obj_seq_type) %>%
  summarize(sum = sum(expenditure, na.rm=TRUE)) %>%
  pivot_wider(names_from = obj_seq_type, values_from = sum) %>%
  mutate(principal = `88110108`,
         interest = `88130108`,
         ratio = (as.numeric(interest)/as.numeric(principal)))

short_debt %>% select(principal, interest, ratio) %>%
  mutate(across(principal:interest, ~format(., big.mark= ",",  scientific = F)))
```

```
## # A tibble: 9 x 4
## # Groups:   fy [9]
##      fy principal     interest      ratio
##   <dbl> <chr>         <chr>         <dbl>
## 1  2004 1,500,000,000 22,364,583  0.0149
## 2  2005 1,615,000,000 10,672,222  0.00661
## 3  2007 900,000,000   NA          NA
## 4  2008 1,200,000,000 6,233,333   0.00519
## 5  2009 1,400,000,000 26,675,000  0.0191
## 6  2010 2,250,000,000 55,277,778  0.0246
## 7  2011 1,300,000,000 30,975,000  0.0238
## 8  2021 2,184,745,000 51,007,557  0.0233
## 9  2022 1,164,255,000 90,437,183  0.0777
```

```
short_debt %>% ggplot() +
  geom_col(aes(x=fy, y=principal/1000000, fill = "Principal"))+
  geom_col(aes(x=fy, y=interest/1000000, fill = "Interest")) +
  labs(title = "Short Term Borrowing: Principal and Interest Payments")
```

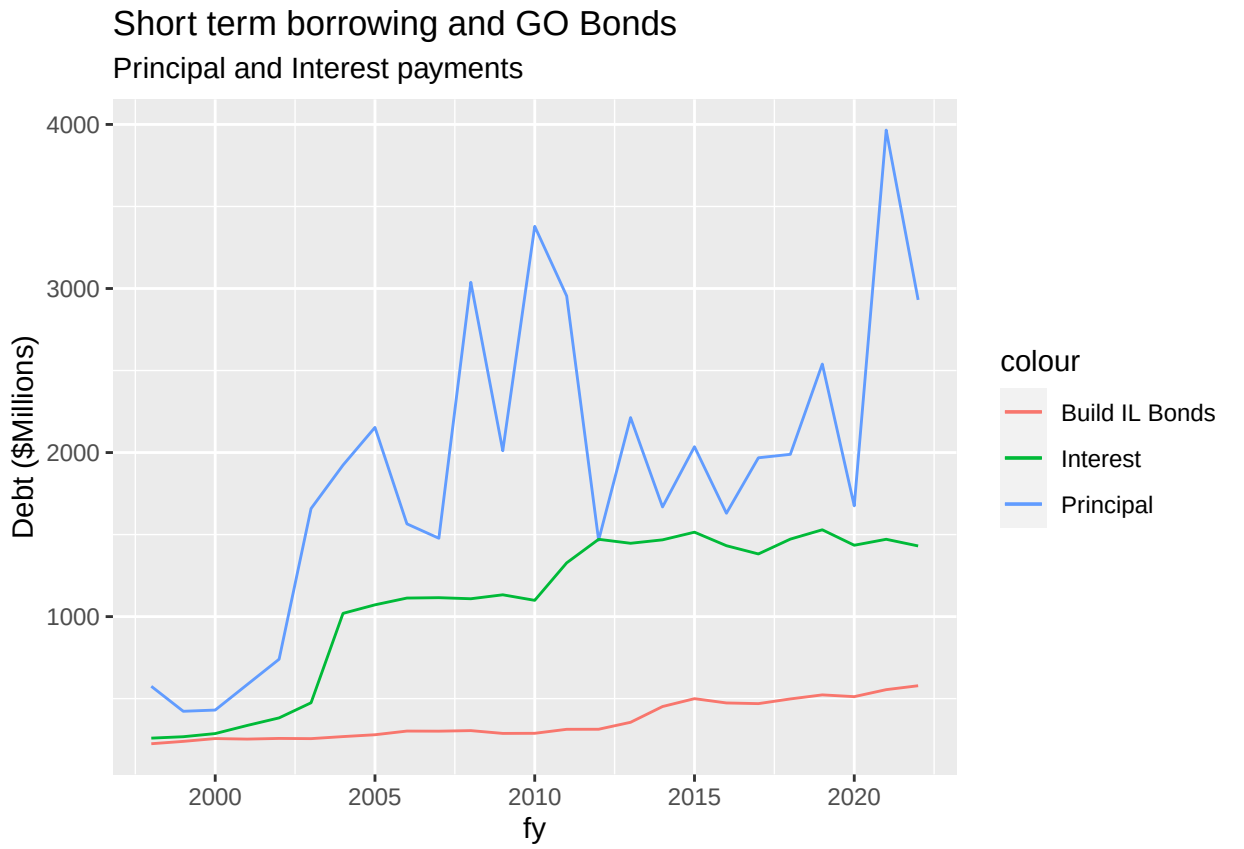



```
capitalprojects <- exp_temp %>%filter(object == "8800")


all_debt <- exp_temp %>%
  filter(fund != "0455" & (object == "8811" |object == "8813" | object == "8800") )%>%
  group_by(fy, object) %>%
  summarize(sum = sum(expenditure, na.rm=TRUE)) %>%
  pivot_wider(names_from = object, values_from = sum) %>%
  mutate(principal = `8811`,
         interest = `8813`,
         BuildIL = `8800`,
         ratio = (as.numeric(interest)/as.numeric(principal)))

all_debt %>% select(principal, interest, BuildIL, ratio) %>%
  mutate(across(principal:BuildIL, ~format(., big.mark= ",", scientific = F)))
```

```
## # A tibble: 25 x 5
## # Groups:   fy [25]
##       fy principal     interest      BuildIL     ratio
##    <dbl> <chr>         <chr>         <chr>       <dbl>
```

```
##  1  1998 575,371,231   259,385,877   225,147,958 0.451
##  2  1999 422,975,040   267,956,231   239,571,392 0.634
##  3  2000 430,464,406   287,154,654   256,685,875 0.667
##  4  2001 585,553,477   336,582,816   253,461,423 0.575
##  5  2002 740,370,036   382,634,975   257,622,074 0.517
##  6  2003 1,658,569,732 474,937,865   256,264,319 0.286
##  7  2004 1,923,668,657 1,019,833,688 268,750,574 0.530
##  8  2005 2,153,260,524 1,071,877,490 279,939,826 0.498
##  9  2006 1,565,449,887 1,113,124,277 302,328,995 0.711
## 10  2007 1,477,592,635 1,115,485,186 301,723,144 0.755
## # ... with 15 more rows
```

```
all_debt %>% ggplot() +
  geom_line(aes(x=fy, y=principal/1000000, color = "Principal"))+
  geom_line(aes(x=fy, y=interest/1000000, color = "Interest"))+
  geom_line(aes(x=fy, y = BuildIL / 1000000, color = "Build IL Bonds"))+
  labs(y = "Debt ($Millions)", title = "Short term borrowing and GO Bonds",
       subtitle = "Principal and Interest payments")
```



Capital projects include the IL Civic Center and Build Illinois Bonds. Tollway principal and interest has been dropped from Debt Service but is counted in Tollway Expenditure Cost.

```
debt_drop <- exp_temp %>%
  filter(object == "8841" |  object == "8811")
# escrow  OR  principle
```

```
#debt_drop %>% group_by(fy) %>% summarize(sum = sum(expenditure)) %>% arrange(-fy)


debt_keep <- exp_temp %>%
  filter(fund != "0455" & (object == "8813" | object == "8800" ))
# examine the debt costs we want to include

#debt_keep %>% group_by(fy) %>% summarize(sum = sum(expenditure)) %>% arrange(-fy)


exp_temp <- anti_join(exp_temp, debt_drop)
exp_temp <- anti_join(exp_temp, debt_keep)

debt_keep <- debt_keep %>%
  mutate(
    agency = ifelse(fund != "0455" & (object == "8813" | object == "8800"), "903", as.character(agency)
    group = ifelse(fund != "0455" & (object == "8813" | object == "8800"), "903", as.character(group)),
    in_ff = ifelse(group == "903", 1, as.character(in_ff)))

debt_keep_yearly <- debt_keep %>% group_by(fy, group) %>% summarize(debt_cost = sum(expenditure,na.rm=T
```

Medicaid

Medicaid. That portion of the Healthcare and Family Services (or Public Aid in earlier years, agency code 478) budget for Medical (appr_organization code 65) for awards and grants (object codes 4400 and 4900).

> State CURE will remain in the Medicaid expenditure category due to the nature of it being federal funds providing public health services and funding to locations that provide public services.

- Uses same appropriation name of "HEALTHCARE PROVIDER RELIEF" and fund == 0793 and obj_seq_type == 49000000. So can defend the "mistake" of including healthcare provider relief as Medicaid expenditure.

Add Fiscal Future group codes

```
exp_temp <- exp_temp %>%
  #mutate(agency = as.numeric(agency) ) %>%
  # arrange(agency)%>%
  mutate(
    group = case_when(
      agency>"100"& agency<"200" ~ "910", # legislative

      agency == "528"  | (agency>"200" & agency<"300") ~ "920", # judicial
      pension>0  ~ "901", # pensions
      (agency>"309" & agency<"400") ~ "930",    # elected officers

      agency == "586" ~ "959", # create new K-12 group

      agency=="402" | agency=="418" | agency=="478" | agency=="444" | agency=="482" ~ as.character(agen
      T ~ as.character(group))
```

```
  ) %>%


mutate(group = case_when(
  agency=="478" & (appr_org=="01" | appr_org == "65" | appr_org=="88") & (object=="4900" | object=="4

  agency == "586" & fund == "0355" ~ "945",  # 586 (Board of Edu) has special education which is part

  # OLD CODE: agency == "586" & appr_org == "18" ~ "945", # Spec. Edu Medicaid Matching

  agency=="425" | agency=="466" | agency=="546" | agency=="569" | agency=="578" | agency=="583" | agen

  agency=="420" | agency=="494" |  agency=="406" | agency=="557" ~ as.character(agency), # econ devt

  agency=="511" | agency=="554" | agency=="574" | agency=="598" ~ "946",  # Capital improvement

  agency=="422" | agency=="532" ~ as.character(agency), # environment & nat. resources

  agency=="440" | agency=="446" | agency=="524" | agency=="563"  ~ "944", # business regulation

  agency=="492" ~ "492", # revenue

  agency == "416" ~ "416", # central management services
  agency=="448" & fy > 2016 ~ "416", #add DoIT to central management

  T ~ as.character(group))) %>%


mutate(group = case_when(
  agency=="684" | agency=="691"  ~ as.character(agency),

  agency=="692" | agency=="695" | (agency>"599" & agency<"677") ~ "960", # higher education

  agency=="427"  ~ as.character(agency), # employment security

  agency=="507"|  agency=="442" | agency=="445" | agency=="452" |agency=="458" | agency=="497" ~ "948

  # other boards & Commissions
  agency=="503" | agency=="509" | agency=="510" | agency=="565" |agency=="517" | agency=="525" | agen

  # non-pension expenditures of retirement funds moved to "Other Departments"
  # should have removed pension expenditures already from exp_temp in Pensions step above
  agency=="131" | agency=="275" | agency=="589" |agency=="593"|agency=="594"|agency=="693" ~ "948",

  T ~ as.character(group))) %>%

mutate(group_name =
         case_when(
           group == "416" ~ "Central Management",
           group == "478" ~ "Healthcare and Family Services",
           group == "482" ~ "Public Health",
           group == "900" ~ "NOT IN FRAME",
           group == "901" ~ "STATE PENSION CONTRIBUTION",
```

```
            group == "903" ~ "DEBT SERVICE",
            group == "910" ~ "LEGISLATIVE"  ,
            group == "920" ~ "JUDICIAL" ,
            group == "930" ~ "ELECTED OFFICERS" ,
            group == "940" ~ "OTHER HEALTH-RELATED",
            group == "941" ~ "PUBLIC SAFETY" ,
            group == "942" ~ "ECON DEVT & INFRASTRUCTURE" ,
            group == "943" ~ "CENTRAL SERVICES",
            group == "944" ~ "BUS & PROFESSION REGULATION" ,
            group == "945" ~ "MEDICAID" ,
            group == "946" ~ "CAPITAL IMPROVEMENT" ,
            group == "948" ~ "OTHER DEPARTMENTS" ,
            group == "949" ~ "OTHER BOARDS & COMMISSIONS" ,
            group == "959" ~ "K-12 EDUCATION" ,
            group == "960" ~ "UNIVERSITY EDUCATION" ,
            group == agency ~ as.character(group),
            TRUE ~ "Check name"),
        year = fy)

exp_temp %>% filter(group_name == "Check name")
```

```
## # A tibble: 0 x 37
## # ... with 37 variables: fy <dbl>, fund <chr>, fund_name <chr>, agency <chr>,
## #   agency_name <chr>, appr_org <chr>, org_name <chr>, obj_seq_type <chr>,
## #   appn_net_xfer <dbl>, expenditure <dbl>, data_source <chr>, object <chr>,
## #   category <chr>, sequence <chr>, type <chr>, trans_agency <chr>,
## #   trans_type <chr>, wh_approp_name <chr>, fund_cat <chr>,
## #   fund_cat_name.x <chr>, expenditurethrough6302022 <dbl>, fund_ab <chr>,
## #   fund_ioc <chr>, fund_re <chr>, a_end <dbl>, in_ff <chr>, ...
```

```
#write_csv(exp_temp, "all_expenditures_recoded.csv")
```

All expenditures recoded but not aggregated: Allows for inspection of individual expenditures within larger categories. This stage of the data is extremely useful for investigating almost all questions we have about the data.

Note that these are the raw figures BEFORE we take the additional steps:

- Subtract employee insurance premiums from State Employee Healthcare expenditures
- Subtract tax refunds from tax revenues by revenue type.
- Subtract employee pension contributions (originally a dropped revenue) from State Pension expenditures
- **NOT DOING ANYMORE**: ~~Add employee health costs and certain pension contributions to All Other Revenues~~

Modify Revenue data

Revenue Categories not included in Fiscal Futures:
- 32. Garnishment-Levies. (State is fiduciary, not beneficiary.)

- 45. Student Fees-Universities. (Excluded from state-level budget.)
- 51. Retirement Contributions (of individuals and non-state entities).
- 66. Proceeds, Investment Maturities. (Not sustainable flow.)
- 72. Bond Issue Proceeds. (Not sustainable flow.)
- 75. Inter-Agency Receipts. (Except from Funds excluded from Fiscal Futures)
- 79. Cook County Intergovernmental Transfers. (State is not beneficiary.)
- 98. Prior Year Refunds.

All Other Sources

Expanded to include the following smaller sources:
- 30. Horse Racing Taxes & Fees.
- 60. Other Grants and Contracts.
- 63. Investment Income. - 75. Inter-Agency Receipts. **(Only from Funds excluded from Fiscal Futures)**

For aggregating revenue, use the rev_1998_2022 dataframe, join the funds_ab_in_2021 file to it, and then join the ioc_source_type file to the dataset.

You need to update the funds_ab_in and ioc_source_type file every year!

include how to do that later

```
# fund info to revenue for all years
rev_temp <- inner_join(rev_1998_2022, funds_ab_in_2022, by = "fund") %>% arrange(source)

# need to update the ioc_source_type file every year!
ioc_source_type <- readxl::read_xlsx("C:/Users/aleaw/OneDrive/Desktop/PhD Fall 2021 - Spring 2022/Merri

rev_temp <- left_join(rev_temp, ioc_source_type, by = "source")
# automatically used source, source name does not match for the join to work using source_name
```

Update Agencies: Early agencies replaced by successors

```
# recodes old agency numbers to consistent agency number
rev_temp <- rev_temp %>%
  mutate(agency = case_when(
    (agency=="438"| agency=="475" |agency == "505") ~ "440",
    # financial institution &  professional regulation &
     # banks and real estate  --> coded as  financial and professional reg
    agency == "473" ~ "588", # nuclear safety moved into IEMA
    (agency =="531" | agency =="577") ~ "532", # coded as EPA
    (agency =="556" | agency == "538") ~ "406", # coded as agriculture
    agency == "560" ~ "592", # IL finance authority (fire trucks and agriculture stuff)to state fire ma
    agency == "570" & fund == "0011" ~ "494",   # city of Chicago road fund to transportation
    TRUE ~ (as.character(agency))))
```

```
rev_temp <- rev_temp %>%
  mutate(
    rev_type = ifelse(rev_type=="57" & agency=="478" & (source=="0618"|source=="2364"|source=="0660"|so
    rev_type_name = ifelse(rev_type=="58", "FEDERAL TRANSPORTATION", rev_type_name),
    rev_type = ifelse(rev_type=="57" & agency=="494", "59", rev_type),
    rev_type_name = ifelse(rev_type=="59", "FEDERAL TRANSPORTATION", rev_type_name),
    rev_type_name = ifelse(rev_type=="57", "FEDERAL OTHER", rev_type_name),
    rev_type = ifelse(rev_type=="6", "06", rev_type),
```

```
    rev_type = ifelse(rev_type=="9", "09", rev_type))


rev_temp %>%
  group_by(fy, rev_type_name) %>%
  summarise(receipts = sum(receipts, na.rm = TRUE)/1000000) %>% pivot_wider(names_from = rev_type_name,
```

```
## # A tibble: 25 x 5
## # Groups:   fy [25]
##       fy      . `FEDERAL OTHER` `FEDERAL TRANSPORTATION`  `NA`
##    <dbl>  <dbl>           <dbl>                    <dbl> <dbl>
##  1  1998 35959.            3804.                    4174.    NA
##  2  1999 41945.            3976.                    4326.    NA
##  3  2000 53111.            4299.                    4691.    NA
##  4  2001 57543.            4409.                    5375.    NA
##  5  2002 59358.            4704.                    5480.    NA
##  6  2003 73997.            5047.                    5520.    NA
##  7  2004 73821.            5598.                    7358.    NA
##  8  2005 70333.            5357.                    6855.    NA
##  9  2006 76545.            5254.                    7138.    NA
## 10  2007 81296.            5585.                    7580.    NA
## # ... with 15 more rows
```

Pension Contributions

Employee contributions to pension are a revenue source for the state. In order to get the net cost of pensions for the state, employee contributions should be subtracted in order to calculate net costs.

- current year employee revenue source = 0573
- contributions by employee == 572 (stops at 2011)

```
#pension_rev_check <- rev_temp %>% filter(source == "0572" | source == "0573" | source == "0574" | sour

#write_csv(pension_rev_check, "pension_revenue_check.csv")


# current year employee revenue source = 0573, contributions by employee == 572 (stops at 2011)
pension_rev <- rev_temp %>% filter(rev_type == "51" & source == "0573" | source == "0572")


rev_type <- anti_join(rev_temp, pension_rev)

pension_rev_yearly <- rev_temp %>% filter(rev_type == "51" & source == "0573" | source == "0572") %>%
  group_by(fy, rev_type) %>%
  summarise(pension_rev_sum = sum(receipts, na.rm=TRUE)/1000000) %>% select(-rev_type)
```

`pension_rev` should be subtracted from the state pension expenditures. Employee contributions to pensions are a revenue source. We want net pension cost, therefore subtract employee contributions from pension costs.

Health Insurance Premiums from Employees

Employee insurance premiums for healthcare are a revenue source for the state. In order to get the true cost of State Employee Healthcare, total employee insurance premiums should be subtracted from the healthcare expenditure total (exp_904 - premiums). Gather all employee premiums into `opt_premiums` and subtract it from the Expenditure table after aggregating and pivoting steps.

0120 = ins prem-option life 0120 = ins prem-optional life/univ

0347 = optional health - HMO 0348 = optional health - dental 0349 = optional health - univ/local SI 0350 = optional health - univ/local 0351 = optional health - retirement 0352 = optional health - retirement SI 0353 = optional health - retire/dental 0354 = optional health - retirement hmo

2199-2209 = various HMOs, dental, health plans from Health Insurance Reserve (fund)

```
opt_premiums_CHECK <- rev_temp %>%
  filter((fund=="0907" | fund == "0457") & (source=="0120" | source=="0121" | (source>"0345" & source<"
         )

#collect optional insurance premiums to fund 0907 for use in eehc expenditure
rev_temp <- rev_temp %>%
  mutate(med_option_recent = ifelse(
    fund=="0907" & (source=="0120"| source=="0121"| (source>"0345" & source<"0357")|(source>"2199" & so

    # adds more rev_type codes
    rev_type = case_when(
      fund =="0427" ~ "12", # pub utility tax
      fund == "0742" | fund == "0473" ~ "24", # insurance and fees
      fund == "0976" ~ "36",# receipts from rev producing
      fund == "0392" |fund == "0723" ~ "39", # licenses and fees
      fund == "0656" ~ "78", #all other rev sources
      TRUE ~ as.character(rev_type)))
#if not mentioned, then rev_type as it was



# optional insurance premiums = employee insurance premiums
med_option_recent <- rev_temp %>%
  group_by(fy, med_option_recent) %>%
  summarize(med_option_amt_recent = sum(receipts)/1000000) %>%
  filter(med_option_recent == 1) %>%
  rename(year = fy) %>%
  select(-med_option_recent)

med_option_long <- rev_temp %>%  filter(med_option_recent == 1)
# 361 observations have med_option_recent == 1


rev_temp <- rev_temp %>% filter(med_option_recent != 1)
# could also do an antijoin with rev_temp and med_option dataframes
```

Still need to add med_option data to Other Revenues

Transfers in and Out: in_from_out <- c("0847", "0867", "1175", "1176", "1177", "1178", "1181", "1182", "1582", "1592", "1745", "1982", "2174", "2264")

SPORTS FACILITIES TAX TRUST REPLACEMENT VEHICLE TAX-ST EASTERN ILLINOIS UNIVERSITY GOVERNOR'S STATE UNIVERSITY NORTHEASTERN ILLINOIS UNIV WESTERN ILLINOIS UNIVERSITY 1579 SOUTHERN ILLINOIS UNIVERSITY 1580 UNIVERSITY OF ILLINOIS 1581 PUBLIC BUILDING FUND 1582 DRYCLEANER TRUST FUND LOCAL DEBT SERVICE ACCT 1745 STATE EMPLOY RETIREMENT SYSTEM 1982 CHILD SUPPORT ENFORCE TRUST PROTEST FUND

I don't have much faith in the transfers in and out steps- AWM

```
rev_temp <- rev_temp %>%
  filter(in_ff == 1) %>%
  mutate(local = ifelse(is.na(local), 0, local)) %>%
  filter(local != 1)

# 1175 doesnt exist?
in_from_out <- c("0847", "0867", "1175", "1176", "1177", "1178", "1181", "1182", "1582", "1592", "1745"

# what does this actually include:
in_out_df <- rev_type %>%
  mutate(infromout = ifelse(source %in% in_from_out, 1, 0)) %>%
  filter(infromout == 1)

rev_temp <- rev_temp %>%
  mutate(rev_type_new = ifelse(source %in% in_from_out, "76", rev_type))
# if source contains any of the codes in in_from_out, code them as 76 (all other rev).

# revenue types to drop
drop_type <- c("32", "45", "51",
               "66", "72", "75", "79", "98")

# drops Blank, Student Fees, Retirement contributions, proceeds/investments,
# bond issue proceeds, interagency receipts, cook IGT, Prior year refunds.


rev_temp <- rev_temp %>% filter(!rev_type_new %in% drop_type)
# keep observations that do not have a revenue type mentioned in drop_type

table(rev_temp$rev_type_new)
```

```
##
##    02    03    06    09    12    15    18    21    24    27    30    31    33
##   161   124   828   127   575   258    45  1420   450    76   659   124   130
##    35    36    39    42    48    54    57    58    59    60    63    76    78
##   660  5152  9067  2755    31  1239  6436   620   226   103  5081   154 10889
##    99
##   964
```

```
rev_temp %>%
  group_by(fy, rev_type_new) %>%
  summarize(total_reciepts = sum(receipts)/1000000) %>%
  pivot_wider(names_from = rev_type_new, values_from = total_reciepts, names_prefix = "rev_")
```

```
## # A tibble: 25 x 29
```

```
## # Groups:   fy [25]
##       fy rev_02 rev_03 rev_06 rev_09 rev_12 rev_15 rev_18 rev_21 rev_24 rev_27
##    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1  1998  6847.  1860.  7198.  1301.  1193.   464.   57.0   250.   132.   123.
##  2  1999  7226.  1855.  7647.  1329.  1423.   499.   57.2   347.   249.   121.
##  3  2000  7686.  2068.  8218.  1356.  1524.   467.  128.    348.   268.   144.
##  4  2001  7996.  1790.  8175.  1367.  1606.   473.  124.    361.   298.   151.
##  5  2002  7471.  1384.  8279.  1374.  1568.   469.  122.    329.   323.   165.
##  6  2003  7341.  1293.  8228.  1388.  1439.   700.  123.    237.   372.   147.
##  7  2004  7272.  1596.  8637.  1424.  1536.   760.  127.    222.   473.   169.
##  8  2005  7979.  1972.  8981.  1435.  1501.   656.  147.    310.   434.   190.
##  9  2006  8635.  2400.  9638.  1446.  1550.   640.  152.    272.   404.   189.
## 10  2007  9408.  2936.  9785.  1454.  1604.   639.  156.    264.   409.   201.
## # ... with 15 more rows, and 18 more variables: rev_30 <dbl>, rev_31 <dbl>,
## #   rev_33 <dbl>, rev_35 <dbl>, rev_36 <dbl>, rev_39 <dbl>, rev_42 <dbl>,
## #   rev_48 <dbl>, rev_54 <dbl>, rev_57 <dbl>, rev_58 <dbl>, rev_59 <dbl>,
## #   rev_60 <dbl>, rev_63 <dbl>, rev_76 <dbl>, rev_78 <dbl>, rev_99 <dbl>,
## #   rev_NA <dbl>
```

```
# combines smallest 4  categories to to "Other"
# they were the 4 smallest in past years, are they still the 4 smallest?

rev_temp <- rev_temp %>%
  mutate(rev_type_new = ifelse(rev_type=="30" | rev_type=="60" | rev_type=="63" | rev_type=="76" | rev_


#table(rev_temp$rev_type_new)  # check work


rm(rev_1998_2022)
rm(exp_1998_2022)
```

Aggregating and Merging

- State employee insurance premiums (eehc from eehc2_amt) should be subtracted from state employee healthcare expenditures. State employer group insurance contributions should be dropped to avoid double counting costs.
- Subtract employee insurance premiums from 904 (State Employee Healthcare Expenditures - Employee Premiums = Actual state healthcare costs. Subtract med_option_amt_recent in med_option_recent from exp_904 in ff_exp).
- Employee pension contributions should be subtracted from state pension costs.
  - exp_901 - rev_51? even though rev_type == 51 (retirement contributions) is dropped based on past code.
- Local Government Transfers (exp_970) should be on the expenditure side

Revenues

```
ff_rev <- rev_temp %>%
  group_by(rev_type_new, fy) %>%
  summarize(sum_receipts = sum(receipts, na.rm=TRUE)/1000000 ) %>%
  pivot_wider(names_from = "rev_type_new", values_from = "sum_receipts", names_prefix = "rev_")

ff_rev<- left_join(ff_rev, tax_refund)

#ff_rev <- left_join(ff_rev, pension2_fy22, by=c("fy" = "year"))

#ff_rev <- left_join(ff_rev, eehc2_amt)
ff_rev <- mutate_all(ff_rev, ~replace_na(.,0))


ff_rev <- ff_rev %>%
  mutate(rev_02 = rev_02 - ref_02,
         rev_03 = rev_03 - ref_03,
         rev_06 = rev_06 - ref_06,
         rev_09 = rev_09 - ref_09,
         rev_21 = rev_21 - ref_21,
         rev_24 = rev_24 - ref_24,
         rev_35 = rev_35 - ref_35,

         rev_78new = rev_78 #+ pension_amt #+ eehc
         ) %>%
  select(-c(ref_02:ref_35, rev_76, rev_78, rev_99, rev_NA#, pension_amt
           #  , eehc
             )) %>%  # drop unwanted columns
  rowwise() %>%
  mutate(rev_TOTALS = sum(across(rev_02:rev_78new)))

ff_rev
```

```
## # A tibble: 25 x 24
## # Rowwise:
##       fy rev_02 rev_03 rev_06 rev_09 rev_12 rev_15 rev_18 rev_21 rev_24 rev_27
##    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1  1998  6334.  1631.  7198.  1291.  1193.   464.   57.0   246.   132.   123.
##  2  1999  6609.  1577.  7647.  1316.  1423.   499.   57.2   343.   249.   121.
##  3  2000  6910.  1698.  8218.  1340.  1524.   467.  128.    341.   266.   144.
##  4  2001  7156.  1550.  8175.  1354.  1606.   473.  124.    354.   293.   151.
##  5  2002  6639.  1172.  8279.  1358.  1568.   469.  122.    322.   321.   165.
##  6  2003  6096.   936.  8228.  1374.  1439.   700.  123.    230.   370.   147.
##  7  2004  6170.   987.  8637.  1406.  1536.   760.  127.    215.   470.   169.
##  8  2005  6936.  1604.  8981.  1420.  1501.   656.  147.    301.   433.   190.
##  9  2006  7537.  2061.  9638.  1430.  1550.   640.  152.    263.   403.   189.
## 10  2007  8271.  2645.  9785.  1440.  1604.   639.  156.    255.   406.   201.
## # ... with 15 more rows, and 13 more variables: rev_31 <dbl>, rev_33 <dbl>,
## #   rev_35 <dbl>, rev_36 <dbl>, rev_39 <dbl>, rev_42 <dbl>, rev_48 <dbl>,
## #   rev_54 <dbl>, rev_57 <dbl>, rev_58 <dbl>, rev_59 <dbl>, rev_78new <dbl>,
## #   rev_TOTALS <dbl>
```

Since I already pivot_wider()ed the table in the previous code chunk, I now change each column's name by using rename() to set new variable names. Ideally the final dataframe would have both the variable name

and the variable label but I have not done that yet.

Expenditures

- Create exp_930 for the cost of debt interest
- Create state employee healthcare costs that reflects the health costs minus the insurance premiums paid by employees that came in (904_new = healthcare_cost - med_option_amt_recent).
- Create exp_901_new for pensions that reflect employees pension contributions. Subtract employee contributions from pension costs (exp_901 - pension_rev_sum).
- Create exp_970 for all local government transfers (exp_971 + exp_972 + exp_975 + exp_976).

```
ff_exp <- exp_temp %>%
  group_by(fy, group) %>%
  summarize(sum_expenditures = sum(expenditure, na.rm=TRUE)/1000000 ) %>%
  pivot_wider(names_from = "group", values_from = "sum_expenditures", names_prefix = "exp_")%>%

    left_join(debt_keep_yearly) %>%
  mutate(exp_903 = debt_cost) %>%

    left_join(healthcare_costs_yearly) %>%

  # join state employee healthcare and subtract employee premiums
  left_join(med_option_recent, by = c("fy" = "year")) %>%
  mutate(exp_904_new = (`healthcare_cost` - `med_option_amt_recent`)) %>% # state employee healthcare p

  left_join(pension_rev_yearly) %>%
      mutate(exp_901_new = exp_901 - pension_rev_sum) %>% #employee pension contributions


  # join local transfers and create exp_970
  left_join(transfers) %>%
  mutate(exp_970 = exp_971 + exp_972  + exp_975 + exp_976)

ff_exp<- ff_exp %>%
  select(-c(exp_901, med_option_amt_recent, debt_cost, healthcare_cost, pension_rev_sum, exp_971:exp_97
  rowwise() %>%
  mutate(exp_TOTALS = sum(across(exp_402:exp_970)))
```

- after aggregating expenditures and revenues, pivoting wider, and left_joining the additional mini dataframes (med_option_recent, debt_keep_yearly, pension_rev_yearly, healthcare_costs_yearly, & transfers), then I dropped the columns that I no longer want in my final tables. I also added a TOTALS column by summing across the columns.

Graphs!

In order to graph data in different ways, it is helpful to have data in both wide and long formats.

The chunk below creates the long format data of the recoded observations for expenditures and revenues (`exp_long` and `rev_long`). The category names are all upper case and long and based on Stata labels.

Nearly all expenditure categories and revenue sources are still separate in this stage. Later on, the largest ten-ish to fifteen-ish categories are kept separate and the the remaining smaller expenditure and revenue categories are aggregated to "All Other _____" for final summary tables that are used in the Fiscal Futures paper.

Our rough rule of thumb was to keep categories larger than $2 billion but that is not a strict rule at all. The other goal was to replicate tables from past years to allow for comparability over time.

```
###### clean revenue data in long format ####

# take ff_rev and pivot it wider so that all of the revenue sources are in one column and all values ar
# then apply labels  based on Stata code

rev_long <- pivot_longer(ff_rev, rev_02:rev_TOTALS, names_to = c("type","Category"), values_to = "Dolla
  rename(Year = fy) %>%
  filter(Year < 2022) %>%
  mutate(Category_name = case_when(
    Category == "02" ~ "INDIVIDUAL INCOME TAXES, gross local, net refunds" ,
    Category == "03" ~ "CORPORATE INCOME TAXES, gross of PPRT, net of refunds" ,
    Category == "06" ~ "SALES TAXES, gross of local share" ,
    Category == "09" ~ "MOTOR FUEL TAX, gross of local share, net of refunds" ,
    Category == "12" ~ "PUBLIC UTILITY TAXES, gross of PPRT" ,
    Category == "15" ~ "CIGARETTE TAXES" ,
    Category == "18" ~ "LIQUOR GALLONAGE TAXES" ,
    Category == "21" ~ "INHERITANCE TAX" ,
    Category == "24" ~ "INSURANCE TAXES&FEES&LICENSES, net of refunds " ,
    Category == "27" ~ "CORP FRANCHISE TAXES & FEES" ,
    Category == "30" ~ "HORSE RACING TAXES & FEES",  # in Other
    Category == "31" ~ "MEDICAL PROVIDER ASSESSMENTS" ,
    Category == "32" ~ "GARNISHMENT-LEVIES" , # dropped
    Category == "33" ~  "LOTTERY RECEIPTS" ,
    Category == "35" ~  "OTHER TAXES" ,
    Category == "36" ~  "RECEIPTS FROM REVENUE PRODUCNG",
    Category == "39" ~  "LICENSES, FEES & REGISTRATIONS" ,
    Category == "42" ~  "MOTOR VEHICLE AND OPERATORS" ,
    Category == "45" ~  "STUDENT FEES-UNIVERSITIES",   # dropped
    Category == "48" ~  "RIVERBOAT WAGERING TAXES" ,
    Category == "51" ~  "RETIREMENT CONTRIBUTIONS" , # dropped
    Category == "54" ~ "GIFTS AND BEQUESTS",
    Category == "57" ~  "FEDERAL OTHER" ,
    Category == "58" ~  "FEDERAL MEDICAID",
    Category == "59" ~  "FEDERAL TRANSPORTATION" ,
    Category == "60" ~  "OTHER GRANTS AND CONTRACTS", #other
    Category == "63" ~  "INVESTMENT INCOME", # other
    Category == "66" ~ "PROCEEDS,INVESTMENT MATURITIES" , #dropped
    Category == "72" ~ "BOND ISSUE PROCEEDS",  #dropped
    Category == "75" ~  "INTER-AGENCY RECEIPTS ",  #dropped
    Category == "76" ~  "TRANSFER IN FROM OUT FUNDS",  #other
    Category == "78new" ~  "ALL OTHER SOURCES" ,
    Category == "79" ~   "COOK COUNTY IGT", #dropped
    Category == "98" ~  "PRIOR YEAR REFUNDS", #dropped
    Category == "TOTALS"~ "Total Revenue",
                 T ~ "Check Me!"

  ) )
```

```
###### clean expenditure data in long format ####

# pivot ff_exp longer so that the only variables are Year, type, Category, Category_name, and Dollars
exp_long <- pivot_longer(ff_exp, exp_402:exp_TOTALS , names_to = c("type", "Category"), values_to = "Do
  rename(Year = fy ) %>%
    filter(Year < 2022) %>%

  mutate(Category_name =
           case_when(
             Category == "402" ~ "AGING" ,
             Category == "406" ~ "AGRICULTURE",
             Category == "416" ~ "CENTRAL MANAGEMENT",
             Category == "418" ~ "CHILDREN AND FAMILY SERVICES",
             Category == "420" ~ "COMMERCE AND ECONOMIC OPPORTUNITY",
             Category == "422" ~ "NATURAL RESOURCES" ,
             Category == "426" ~ "CORRECTIONS",
             Category == "427" ~ "EMPLOYMENT SECURITY" ,
             Category == "444" ~ "HUMAN SERVICES" ,
             Category == "448" ~ "Innovation and Technology", # AWM added fy2022
             Category == "478" ~ "HEALTHCARE & FAM SER NET OF MEDICAID",
             Category == "482" ~ "PUBLIC HEALTH",
             Category == "492" ~ "REVENUE",
             Category == "494" ~ "TRANSPORTATION" ,
             Category == "532" ~ "ENVIRONMENTAL PROTECT AGENCY" ,
             Category == "557" ~ "IL STATE TOLL HIGHWAY AUTH" ,
             Category == "684" ~ "IL COMMUNITY COLLEGE BOARD",
             Category == "691" ~ "IL STUDENT ASSISTANCE COMM" ,
             Category == "900" ~ "NOT IN FRAME",
             Category == "901" ~ "STATE PENSION CONTRIBUTION",
             Category == "903" ~ "DEBT SERVICE",
             Category == "904" ~ "State Employee Healthcare",
             Category == "910" ~ "LEGISLATIVE"  ,
             Category == "920" ~ "JUDICIAL" ,
             Category == "930" ~ "ELECTED OFFICERS" ,
             Category == "940" ~ "OTHER HEALTH-RELATED",
             Category == "941" ~ "PUBLIC SAFETY" ,
             Category == "942" ~ "ECON DEVT & INFRASTRUCTURE" ,
             Category == "943" ~ "CENTRAL SERVICES",
             Category == "944" ~ "BUS & PROFESSION REGULATION" ,
             Category == "945" ~ "MEDICAID" ,
             Category == "946" ~ "CAPITAL IMPROVEMENT" ,
             Category == "948" ~ "OTHER DEPARTMENTS" ,
             Category == "949" ~ "OTHER BOARDS & COMMISSIONS" ,
             Category == "959" ~ "K-12 EDUCATION" ,
             Category == "960" ~ "UNIVERSITY EDUCATION",
             Category == "970" ~ "Local Govt Transfers",
           Category == "TOTALS" ~ "Total Expenditure",
             T ~ "CHECK ME!")
           )
# uncomment to write long format data as CSVs for inspecting work

 # write_csv(exp_long, "expenditures_recoded_long_FY22.csv")
```

```
 # write_csv(rev_long, "revenue_recoded_long_FY22.csv")


###### All data combined into one long format. Can be easily pivoted in R or Excel by type and Category

aggregated_totals_long <- rbind(rev_long, exp_long) %>% filter(Category!= "TOTALS")


year_totals <- aggregated_totals_long %>%
  group_by(type, Year) %>%
  summarize(Dollars = sum(Dollars, na.rm = TRUE)) %>%
  pivot_wider(names_from = "type", values_from = Dollars) %>%
  rename(
         Expenditures = exp,
         Revenue = rev) %>%
  mutate(Gap = Revenue - Expenditures) %>% arrange(-Year)
# creates variable for the Gap each year

year_totals
```

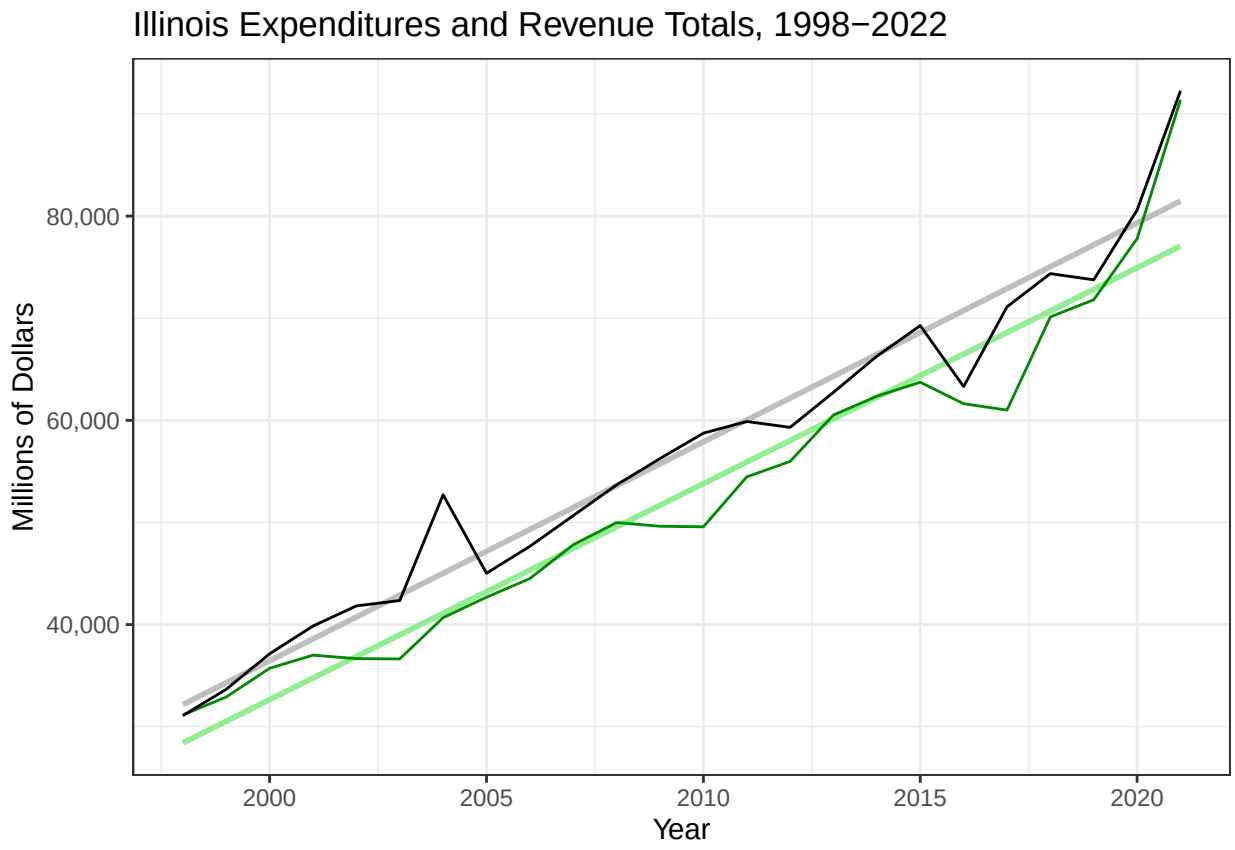
```
## # A tibble: 24 x 4
##     Year Expenditures Revenue     Gap
##    <dbl>        <dbl>   <dbl>   <dbl>
##  1  2021       92264.  91392.   -872.
##  2  2020       80600.  77792.  -2808.
##  3  2019       73760.  71807.  -1952.
##  4  2018       74366.  70127.  -4239.
##  5  2017       71118.  61002. -10115.
##  6  2016       63302.  61625.  -1677.
##  7  2015       69279.  63729.  -5549.
##  8  2014       66274.  62360.  -3914.
##  9  2013       62745.  60527.  -2217.
## 10  2012       59310.  55975.  -3335.
## # ... with 14 more rows
```

After pivoting_longer() and creating `rev_long` and `exp_long`, expenditures and revenues are in the same format and can be combined together for the totals and gap each year.

Graphs made from `yearly_totals` and `aggregated_totals_long` dataframe:

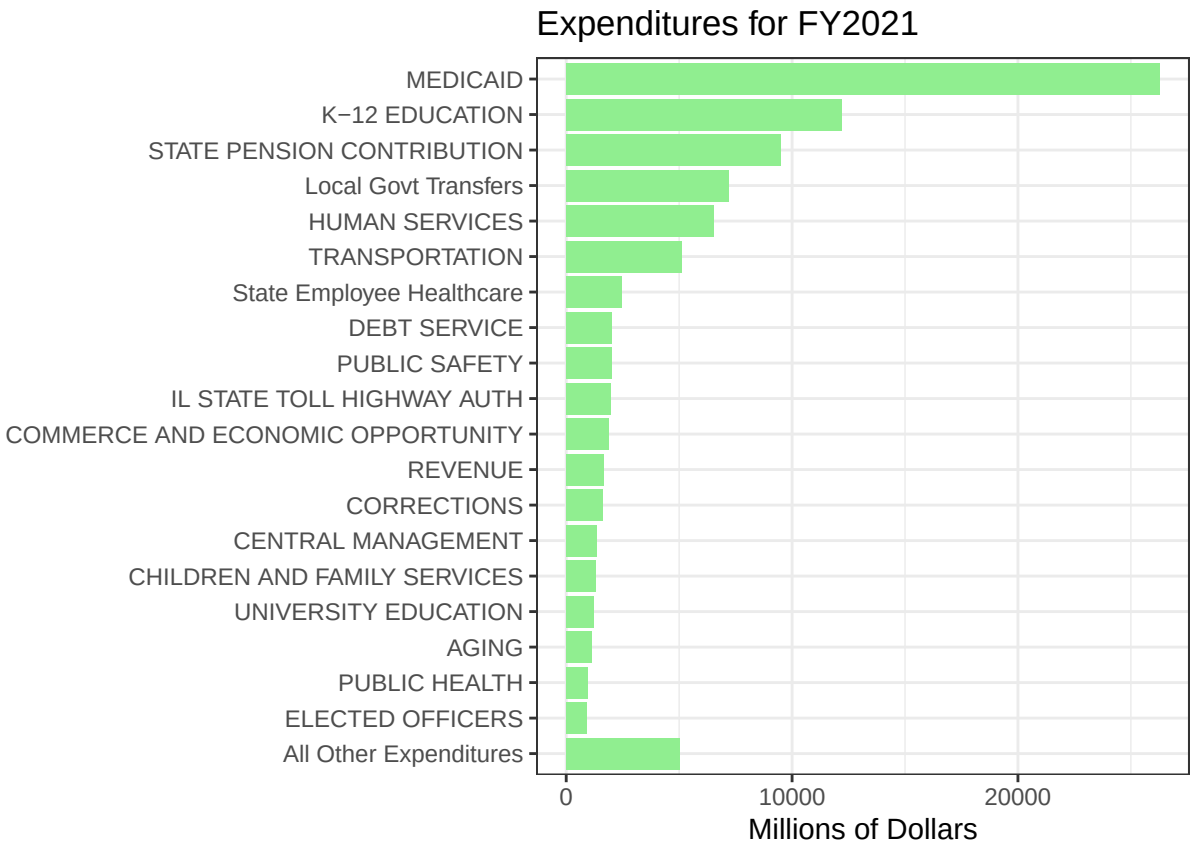
```
year_totals %>%
  ggplot() +
  # geom_smooth adds regression line, graphed first so it appears behind line graph
  geom_smooth(aes(x = Year, y = Revenue), color = "light green", method = "lm", se = FALSE) +
  geom_smooth(aes(x = Year, y = Expenditures), color = "gray", method = "lm", se = FALSE) +

  # line graph of revenue and expenditures
  geom_line(aes(x = Year, y = Revenue), color = "green4") +
  geom_line(aes(x = Year, y = Expenditures), color = "black") +

  # labels
    theme_bw() +
  scale_y_continuous(labels = comma)+
```

```
  xlab("Year") +
  ylab("Millions of Dollars")  +
  ggtitle("Illinois Expenditures and Revenue Totals, 1998-2022")
```

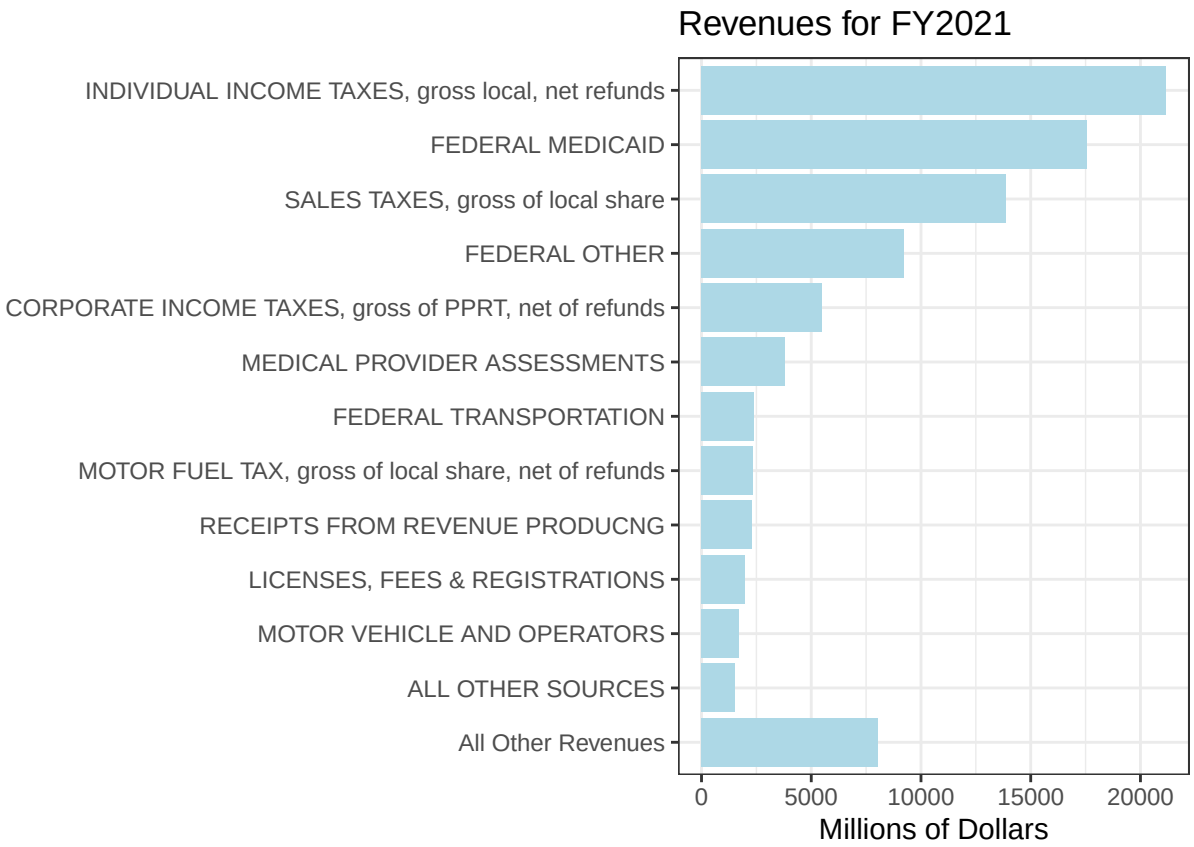


Expenditure and revenues when focusing on largest categories and combining others into "All Other Expenditures(Revenues)":

```
## Code is for top largest categories

exp_long %>%
  filter( Year == 2021 & Category_name != "Total Expenditure") %>%
  mutate(rank = rank(Dollars),
         # keep 13 largest Categories separate and combine the rest
        Category_name = ifelse(rank > 13, Category_name, 'All Other Expenditures')) %>%
 # select(-c(Year, Dollars, rank)) %>%
  arrange(desc(Dollars)) %>%
  ggplot() +
  geom_col(aes(x = fct_reorder(Category_name, `Dollars`), y = `Dollars`), fill = "light green")+
  coord_flip() +
      theme_bw() +
    labs(title = "Expenditures for FY2021") +
    xlab("") +
  ylab("Millions of Dollars")
```




```
rev_long %>%
  filter( Year == 2021 & Category_name != "Total Revenue") %>%
  mutate(rank = rank(Dollars),
         # Keeps 10 largest categories and combines the rest
        Category_name = ifelse(rank > 10, Category_name, 'All Other Revenues')) %>%
 # select(-c(Year, Dollars, rank)) %>%
  arrange(desc(Dollars)) %>%
  ggplot() +
  geom_col(aes(x = fct_reorder(Category_name, `Dollars`), y = `Dollars`), fill = "light blue")+
  coord_flip() +
      theme_bw() +
    labs(title = "Revenues for FY2021") +
    xlab("") +
  ylab("Millions of Dollars")
```



Summary Tables - All Categories

Each year, you will need to update the CAGR formulas!

`calc_cagr` is a function created for calculating the CAGRs for different spans of time.

```
# function for calculating the CAGR
calc_cagr <- function(df, n) {
  df <- exp_long %>%
    select(-type) %>%
    arrange(Category_name, Year) %>%
    group_by(Category_name) %>%
    mutate(cagr = ((`Dollars` / lag(`Dollars`, n)) ^ (1 / n)) - 1)

  return(df)
}

# This works for one variable at a time
cagr_23 <- calc_cagr(exp_long, 23) %>%
  # group_by(Category) %>%
  summarize(cagr_23 = round(sum(cagr*100, na.rm = TRUE), 2))

cagr23_precovid <- exp_long %>%
  filter(Year <= 2019) %>%
  calc_cagr(21) %>%
```

```
  summarize(cagr_21 = round(sum(cagr*100, na.rm = TRUE), 2))

cagr_10 <- calc_cagr(exp_long, 10) %>%
  filter(Year == 2021) %>%
  summarize(cagr_10 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_5 <- calc_cagr(exp_long, 5) %>%
  filter(Year == 2021) %>%
  summarize(cagr_5 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_3 <- calc_cagr(exp_long, 3) %>%
  filter(Year == 2021) %>%
  summarize(cagr_3 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_2 <- calc_cagr(exp_long, 2) %>%
  filter(Year == 2021) %>%
  summarize(cagr_2 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_1 <- calc_cagr(exp_long, 1) %>%
  filter(Year == 2021) %>%
  summarize(cagr_1 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

CAGR_expenditures_summary_allcats <- data.frame(cagr_1, cagr_2, cagr_3, cagr_5, cagr_10, cagr_23 ) %>%
  select(-c(Category_name.1, Category_name.2, Category_name.3, Category_name.4, Category_name.5 )) %>%
  rename("Expenditure Category" = Category_name, "1 Year CAGR" = cagr_1, "2 Year CAGR" = cagr_2, "3 Yea

CAGR_expenditures_summary_allcats
```

```
##                    Expenditure Category 1 Year CAGR 2 Year CAGR 3 Year CAGR
## 1                                 AGING        7.37        7.60        6.23
## 2                            AGRICULTURE       -7.33       -6.19       -7.75
## 3           BUS & PROFESSION REGULATION        2.68        0.18        0.92
## 4                    CAPITAL IMPROVEMENT       47.65       33.13       26.68
## 5                     CENTRAL MANAGEMENT       14.48       20.65       10.93
## 6           CHILDREN AND FAMILY SERVICES        5.21        6.30        6.30
## 7      COMMERCE AND ECONOMIC OPPORTUNITY      205.88       81.45       56.14
## 8                             CORRECTIONS        4.98        1.00       -6.10
## 9                            DEBT SERVICE        4.06       -0.64        0.93
## 10                     ELECTED OFFICERS        5.29        0.54        3.51
## 11                   EMPLOYMENT SECURITY       38.42       21.62       11.45
## 12          ENVIRONMENTAL PROTECT AGENCY       -6.39      -10.75       -9.63
## 13 HEALTHCARE & FAM SER NET OF MEDICAID       13.01      -12.28       -2.22
## 14                          HUMAN SERVICES        9.26        7.94        6.43
## 15           IL COMMUNITY COLLEGE BOARD        3.58        7.46        6.03
## 16           IL STATE TOLL HIGHWAY AUTH        2.38        5.89       10.19
## 17           IL STUDENT ASSISTANCE COMM       -4.94        1.44        0.64
## 18                                JUDICIAL        8.55       11.65        6.40
## 19                       K-12 EDUCATION        7.73        6.99        5.20
## 20                          LEGISLATIVE        4.66        6.56        5.81
## 21                 Local Govt Transfers       11.20        4.93        5.11
## 22                              MEDICAID       17.89       17.52       12.23
## 23                    NATURAL RESOURCES        4.07        0.99        4.23
## 24           OTHER BOARDS & COMMISSIONS       17.91        3.83        4.15
```

```
## 25                    OTHER DEPARTMENTS       37.95        3.96       -2.15
## 26                        PUBLIC HEALTH       69.52       47.52       35.85
## 27                        PUBLIC SAFETY       35.19       41.13       30.90
## 28                              REVENUE       49.75       68.74       48.44
## 29           State Employee Healthcare        -6.28       -6.24      -23.51
## 30          STATE PENSION CONTRIBUTION         6.52        8.17        8.66
## 31                   Total Expenditure       14.47       11.84        7.45
## 32                      TRANSPORTATION       32.40       25.59       14.58
## 33                 UNIVERSITY EDUCATION        1.76        3.29        2.75
##    5 Year CAGR 10 Year CAGR 23 Year CAGR
## 1         9.99         4.80         7.54
## 2         3.56        -0.85        -0.46
## 3        -4.06        -3.07         0.23
## 4        37.78         2.16         2.50
## 5        13.64         5.90         5.21
## 6         3.69         0.72         0.05
## 7        29.92         5.06         6.32
## 8        11.10         2.42         2.28
## 9         1.23         2.13         6.42
## 10        6.03         3.68         3.72
## 11        9.77         1.34         2.74
## 12       -3.87        -1.42         3.36
## 13        2.01        -3.15         5.26
## 14        5.29         1.37         2.27
## 15       28.63         0.06         1.57
## 16       -2.66        10.25         7.56
## 17        9.48        -2.20         0.78
## 18        5.97         3.55         2.99
## 19        5.46         2.45         3.88
## 20        3.68         2.38         2.54
## 21        2.87         2.64         3.20
## 22        9.06         6.38         7.12
## 23       10.93         2.02         1.48
## 24        2.99        -4.40         4.21
## 25        5.37        21.67         3.49
## 26       22.34        11.58         8.02
## 27       22.39         9.60         6.94
## 28       33.00         8.53         6.22
## 29       83.61         4.37         6.27
## 30        6.11        11.72        10.71
## 31        7.83         4.42         4.85
## 32        4.23         1.58         4.25
## 33       25.91        -1.12        -0.32
```

```
expenditure_change_allcats <- exp_long %>%
  select(-c(type,Category)) %>%
  filter(Year > 2019) %>%
  pivot_wider(names_from = Year , values_from = Dollars,   names_prefix = "Dollars_") %>%
  mutate("FY 2021 Expenditures ($ billions)" = round(Dollars_2021/1000, digits = 1),
  #  "Change from 2020 to 2021" = Dollars_2021 - Dollars_2020,
         "Percent Change from 2020 to 2021" = round((Dollars_2021 -Dollars_2020)/Dollars_2020*100, digi
  left_join(CAGR_expenditures_summary_allcats, by = c("Category_name" = "Expenditure Category")) %>%
  arrange(-`FY 2021 Expenditures ($ billions)`)%>%
  select(-c(Dollars_2020, Dollars_2021, `1 Year CAGR`:`10 Year CAGR`)) %>%
```

```
  rename( "Compound Annual Growth, 1998-2021*" = `23 Year CAGR`,
          "FY2021 Expenditure Category" = Category_name )

expenditure_change_allcats
```

```
## # A tibble: 33 x 4
##    `FY2021 Expenditure Cate~` `FY 2021 Expen~` `Percent Chang~` `Compound Annu~`
##    <chr>                                <dbl>            <dbl>            <dbl>
##  1 Total Expenditure                     92.3            14.5              4.85
##  2 MEDICAID                              26.3            17.9              7.12
##  3 K-12 EDUCATION                        12.2             7.73             3.88
##  4 STATE PENSION CONTRIBUTION             9.5             6.52            10.7
##  5 Local Govt Transfers                   7.2            11.2              3.2
##  6 HUMAN SERVICES                         6.5             9.26             2.27
##  7 TRANSPORTATION                         5.1            32.4              4.25
##  8 State Employee Healthcare              2.5            -6.28             6.27
##  9 IL STATE TOLL HIGHWAY AUTH             2               2.38             7.56
## 10 PUBLIC SAFETY                          2              35.2              6.94
## # ... with 23 more rows
```

```
calc_cagr <- function(df, n) {
  df <- rev_long %>%
    arrange(Category_name, Year) %>%
    group_by(Category_name) %>%
    mutate(cagr = ((Dollars / lag(Dollars, n)) ^ (1 / n)) - 1)

  return(df)
}

# This works for one variable at a time
cagr_23 <- calc_cagr(rev_long, 23) %>%
  summarize(cagr_23 = round(sum(cagr*100, na.rm = TRUE), 2))

cagr_10 <- calc_cagr(rev_long, 10) %>%
  filter(Year == 2021) %>%
  summarize(cagr_10 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_5 <- calc_cagr(rev_long, 5) %>%
  filter(Year == 2021) %>%
  summarize(cagr_5 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_3 <- calc_cagr(rev_long, 3) %>%
  filter(Year == 2021) %>%
  summarize(cagr_3 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_2 <- calc_cagr(rev_long, 2) %>%
  filter(Year == 2021) %>%
  summarize(cagr_2 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

 cagr_1 <- calc_cagr(rev_long, 1) %>%
  filter(Year == 2021) %>%
  summarize(cagr_1 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))
```

```r
CAGR_revenue_summary_allcats <- data.frame(cagr_1, cagr_2, cagr_3, cagr_5, cagr_10, cagr_23) %>%
  select(-c(Category_name.1, Category_name.2, Category_name.3, Category_name.4, Category_name.5 )) %>%
  rename("Revenue Category" = Category_name, "1 Year CAGR" = cagr_1, "2 Year CAGR" = cagr_2, "3 Year CA

CAGR_revenue_summary_allcats
```

```
##                                       Revenue Category 1 Year CAGR
## 1                                    ALL OTHER SOURCES      -12.53
## 2                                      CIGARETTE TAXES        7.82
## 3                           CORP FRANCHISE TAXES & FEES       51.55
## 4  CORPORATE INCOME TAXES, gross of PPRT, net of refunds       68.97
## 5                                    FEDERAL MEDICAID       26.83
## 6                                      FEDERAL OTHER       -5.10
## 7                                FEDERAL TRANSPORTATION       33.42
## 8                                    GIFTS AND BEQUESTS       63.20
## 9      INDIVIDUAL INCOME TAXES, gross local, net refunds       20.22
## 10                                     INHERITANCE TAX       61.51
## 11        INSURANCE TAXES&FEES&LICENSES, net of refunds        31.65
## 12                      LICENSES, FEES & REGISTRATIONS       38.90
## 13                               LIQUOR GALLONAGE TAXES        3.08
## 14                                   LOTTERY RECEIPTS       28.06
## 15                         MEDICAL PROVIDER ASSESSMENTS        9.66
## 16  MOTOR FUEL TAX, gross of local share, net of refunds        2.63
## 17                         MOTOR VEHICLE AND OPERATORS       16.03
## 18                                         OTHER TAXES        7.51
## 19                PUBLIC UTILITY TAXES, gross of PPRT       -3.84
## 20                     RECEIPTS FROM REVENUE PRODUCNG        6.58
## 21                           RIVERBOAT WAGERING TAXES      -45.81
## 22                  SALES TAXES, gross of local share       13.17
## 23                                      Total Revenue       17.48
##    2 Year CAGR 3 Year CAGR 5 Year CAGR 10 Year CAGR 23 Year CAGR
## 1         1.74       -4.14        4.22         1.25         2.84
## 2         9.17        6.26        1.65         4.54         3.00
## 3        13.74       15.44        9.02         4.44         4.42
## 4        22.22       24.19       13.47         6.34         5.41
## 5        20.62        9.22        9.31         8.01         7.48
## 6        24.50       16.65        9.30         0.95         3.99
## 7        32.15       14.31        5.22         2.19         4.66
## 8        15.94        7.49        9.84         5.08        10.92
## 9         7.62        7.95       12.12         8.12         5.39
## 10        7.63        8.26        8.44        13.59         2.53
## 11        9.79        3.91        3.91         4.07         7.01
## 12       29.35       17.38       10.08         8.57         8.43
## 13        2.47        1.76        1.63         5.53         7.67
## 14        5.77        5.58        2.63         3.12         2.53
## 15       26.62       20.00       14.31         8.83         8.84
## 16       32.68       20.42       11.98         6.13         2.64
## 17        2.85        4.49        1.74         1.25         3.62
## 18       -0.68        2.57        4.94        12.10         5.93
## 19       -3.62       -1.46       -0.52        -1.69         0.60
## 20       -5.41       -2.19        1.07         5.75         5.16
## 21      -35.33      -25.85      -16.78        -8.78        -0.76
## 22        5.51        5.69        4.16         3.84         2.89
```

```
## 23       12.82        9.23        8.20         5.31         4.79
```

```
revenue_change_allcats <- rev_long %>%
  select(-c(type,Category)) %>%
  filter(Year > 2019) %>%
  pivot_wider(names_from = Year , values_from = Dollars,   names_prefix = "Dollars_") %>%
  mutate(
    "FY 2021 Revenues ($ billions)" = round(Dollars_2021/1000, digits = 1),
#    "Change from 2020 to 2021" = round(Dollars_2021 - Dollars_2020, digits = 2),
        "Percent Change from 2020 to 2021" = round(((Dollars_2021 -Dollars_2020)/Dollars_2020*100), di
  left_join(CAGR_revenue_summary_allcats, by = c("Category_name" = "Revenue Category")) %>%
    arrange(-`FY 2021 Revenues ($ billions)`)%>%
  rename( "Compound Annual Growth, 1998-2021*" = `23 Year CAGR`,
          "FY2021 Revenue Category" = Category_name ) %>%
  select(-c(Dollars_2020, Dollars_2021, `1 Year CAGR`:`10 Year CAGR`))

revenue_change_allcats
```

```
## # A tibble: 23 x 4
##    `FY2021 Revenue Category`  `FY 2021 Reven~` `Percent Chang~` `Compound Annu~`
##    <chr>                                <dbl>            <dbl>            <dbl>
##  1 Total Revenue                         91.4             17.5             4.79
##  2 INDIVIDUAL INCOME TAXES, ~            21.2             20.2             5.39
##  3 FEDERAL MEDICAID                      17.6             26.8             7.48
##  4 SALES TAXES, gross of loc~            13.9             13.2             2.89
##  5 FEDERAL OTHER                          9.2             -5.1             3.99
##  6 CORPORATE INCOME TAXES, g~             5.5             69.0             5.41
##  7 MEDICAL PROVIDER ASSESSME~             3.8              9.66            8.84
##  8 MOTOR FUEL TAX, gross of ~             2.4              2.63            2.64
##  9 FEDERAL TRANSPORTATION                 2.4             33.4             4.66
## 10 RECEIPTS FROM REVENUE PRO~             2.3              6.58            5.16
## # ... with 13 more rows
```

Another way to examine expenditure and revenue change over time is to use a lag formula like is done in the chunk below. Should be kind of similar to 23 year CAGR.

Expenditure and Revenue Growth using a lag formula:

```
 exp_long %>%
  group_by(Category_name) %>%
  mutate(Growth = ((Dollars) - lag(Dollars))/lag(Dollars) *100) %>%
  summarize(Growth = round(mean(Growth, na.rm = TRUE), 2))
```

```
## # A tibble: 33 x 2
##    Category_name                     Growth
##    <chr>                              <dbl>
##  1 AGING                               9.33
##  2 AGRICULTURE                         0.34
##  3 BUS & PROFESSION REGULATION         0.89
##  4 CAPITAL IMPROVEMENT                17.0
##  5 CENTRAL MANAGEMENT                  6.22
##  6 CHILDREN AND FAMILY SERVICES        0.09
##  7 COMMERCE AND ECONOMIC OPPORTUNITY  15.2
```

```
##  8 CORRECTIONS                          3.36
##  9 DEBT SERVICE                         7.29
## 10 ELECTED OFFICERS                     3.94
## # ... with 23 more rows
```

```
rev_long %>%
 group_by(Category_name) %>%
 mutate(Growth = ((Dollars) - lag(Dollars))/lag(Dollars) *100) %>%
 summarize(Growth = round(mean(Growth, na.rm = TRUE), 2))
```

```
## # A tibble: 23 x 2
##    Category_name                                          Growth
##    <chr>                                                   <dbl>
##  1 ALL OTHER SOURCES                                        4.21
##  2 CIGARETTE TAXES                                          3.8
##  3 CORP FRANCHISE TAXES & FEES                              5.19
##  4 CORPORATE INCOME TAXES, gross of PPRT, net of refunds    8.3
##  5 FEDERAL MEDICAID                                         8.27
##  6 FEDERAL OTHER                                            5.19
##  7 FEDERAL TRANSPORTATION                                   5.92
##  8 GIFTS AND BEQUESTS                                      14.2
##  9 INDIVIDUAL INCOME TAXES, gross local, net refunds        6.48
## 10 INHERITANCE TAX                                          6.97
## # ... with 13 more rows
```

Summary Tables - Largest Categories

The 10 largest revenue sources and 13 largest expenditure sources remain separate categories and all other smaller sources/expenditures are combined into "All Other _____". These condensed tables are typically used in the Fiscal Futures articles. They were manually created in past years but this hopefully automates the process a bit until final formatting stages.

- take ff_rev and ff_exp data frames, which were in wide format, pivot them longer and mutate the Category_name variable to nicer labels. Keep largest categories separate and aggregate the rest.

```
revenue_wide <- pivot_longer(ff_rev, rev_02:rev_TOTALS, names_to = c("type","Category"), values_to = "D
  rename(Year = fy) %>%
  filter(Year <2022) %>%

  mutate(Category_name = case_when(
    Category == "02" ~ "Income Tax" ,
    Category == "03" ~ "Corporate Income Tax" ,
    Category == "06" ~ "Sales Tax" ,
    Category == "09" ~ "Motor Fuel Taxes" ,
#   Category == "12" ~ "PUBLIC UTILITY TAXES, gross of PPRT" ,
 #  Category == "15" ~ "CIGARETTE TAXES" ,
#   Category == "18" ~ "LIQUOR GALLONAGE TAXES" ,
#  Category == "21" ~ "INHERITANCE TAX" ,
 #  Category == "24" ~ "INSURANCE TAXES&FEES&LICENSES, net of refunds " ,
  # Category == "27" ~ "CORP FRANCHISE TAXES & FEES" ,
#   Category == "30" ~ "HORSE RACING TAXES & FEES",  # in Other
```

```
    Category == "31" ~ "Medical Provider Assessments" ,
  #  Category == "32" ~ "GARNISHMENT-LEVIES" , # dropped
  #  Category == "33" ~  "LOTTERY RECEIPTS" ,
   # Category == "35" ~  "OTHER TAXES" ,
    Category == "36" ~  "Receipts from Revenue Producing",
    Category == "39" ~  "Licenses, Fees, Registration" ,
   # Category == "42" ~  "MOTOR VEHICLE AND OPERATORS" ,
#    Category == "45" ~  "STUDENT FEES-UNIVERSITIES",   # dropped
#    Category == "48" ~  "RIVERBOAT WAGERING TAXES" ,
  #  Category == "51" ~  "RETIREMENT CONTRIBUTIONS" , # dropped
   # Category == "54" ~ "GIFTS AND BEQUESTS",
    Category == "57" ~  "Federal Other" ,
    Category == "58" ~  "Federal Medicaid Reimbursements",
    Category == "59" ~  "Federal Transportation" ,
 #   Category == "60" ~  "OTHER GRANTS AND CONTRACTS", #other
#    Category == "63" ~  "INVESTMENT INCOME", # other
 #   Category == "66" ~ "PROCEEDS,INVESTMENT MATURITIES" , #dropped
 #   Category == "72" ~ "BOND ISSUE PROCEEDS",  #dropped
 #   Category == "75" ~  "INTER-AGENCY RECEIPTS ",  #dropped
 #   Category == "76" ~  "TRANSFER IN FROM OUT FUNDS",  #other
   # Category == "78new" ~  "ALL OTHER SOURCES" ,
   # Category == "79" ~   "COOK COUNTY IGT", #dropped
 #   Category == "98" ~  "PRIOR YEAR REFUNDS", #dropped

Category == "TOTALS" ~ "Total Revenue",
T ~ "Other Revenue Sources **" # any other Category number that was not specifically referenced is cobi

  ) ) %>%
  select(-type, -Category) %>%  # drop extra columns type and Category number
  group_by(Year, Category_name) %>%
  summarise(Dollars= round(sum(Dollars),digits=2))

# revenue_wide # not actually in wide format yet.
# has 10 largest rev sources separate and combined all others to Other in long data format.


# creates wide version of table where each revenue source is a column
revenue_wide2 <- revenue_wide %>% pivot_wider(names_from = Category_name,
              values_from = Dollars) %>%
  relocate("Other Revenue Sources **", .after = last_col()) %>%
  relocate("Total Revenue", .after =  last_col())



#### example for pivoting the other way:
# has Category Names as rows and years as columns. Easier to read source variable labels

# revenue_wide %>% pivot_wider(names_from = Year, values_from = Dollars) %>% select(Category_name, '202
```

```
expenditure_wide <- pivot_longer(ff_exp, exp_402:exp_TOTALS , names_to = c("type", "Category"), values_
  rename(Year = fy ) %>%
    filter(Year < 2022) %>%
```

```
  mutate(Category_name =
           case_when(
            # Category == "402" ~ "AGING" ,
           #  Category == "406" ~ "AGRICULTURE",
             Category == "416" ~ "Central Management",
            # Category == "418" ~ "CHILDREN AND FAMILY SERVICES",
             Category == "420" ~ "Commerce & Economic Opportunity",
           #  Category == "422" ~ "NATURAL RESOURCES" ,
            # Category == "426" ~ "CORRECTIONS",
           #  Category == "427" ~ "EMPLOYMENT SECURITY" ,
             Category == "444" ~ "Human Services" ,
           #  Category == "478" ~ "HEALTHCARE & FAM SER NET OF MEDICAID",
           #  Category == "482" ~ "PUBLIC HEALTH",
           #  Category == "492" ~ "REVENUE",
             Category == "494" ~ "Transportation" ,
           #  Category == "532" ~ "ENVIRONMENTAL PROTECT AGENCY" ,
             Category == "557" ~ "Tollway" ,
           #  Category == "684" ~ "IL COMMUNITY COLLEGE BOARD",
            # Category == "691" ~ "IL STUDENT ASSISTANCE COMM" ,
           #  Category == "900" ~ "NOT IN FRAME",
             Category == "901" ~ "State Pension Contribution",
             Category == "903" ~ "Debt Service",
             Category == "904" ~ "State Employee Healthcare",
            #  Category == "910" ~ "LEGISLATIVE"  ,
           #   Category == "920" ~ "JUDICIAL" ,
           #   Category == "930" ~ "ELECTED OFFICERS" ,
             # Category == "940" ~ "OTHER HEALTH-RELATED",
             Category == "941" ~ "Public Safety" ,
            #  Category == "942" ~ "ECON DEVT & INFRASTRUCTURE" ,
            #  Category == "943" ~ "CENTRAL SERVICES",
            #  Category == "944" ~ "BUS & PROFESSION REGULATION" ,
             Category == "945" ~ "Medicaid" ,
             Category == "946" ~ "Capital Improvement" ,
            #  Category == "948" ~ "OTHER DEPARTMENTS" ,
            # Category == "949" ~ "OTHER BOARDS & COMMISSIONS" ,
             Category == "959" ~ "K-12 Education" ,
            #  Category == "960" ~ "UNIVERSITY EDUCATION",
             Category == "970" ~ "Local Government Revenue Sharing",
          Category == "TOTALS" ~ "Total Expenditure",
             T ~ "All Other Expenditures **")
           ) %>%
  select(-type, -Category) %>%
  group_by(Year, Category_name) %>%
  summarise(Dollars= round(sum(Dollars),digits=2))

expenditure_wide2 <- expenditure_wide%>%
  pivot_wider(names_from = Category_name,
              values_from = Dollars) %>%
  relocate("All Other Expenditures **", .after = last_col()) %>%
  relocate("Total Expenditure", .after =  last_col())

expenditure_wide %>% pivot_wider(names_from = Year,
              values_from = Dollars)
```

```
## # A tibble: 15 x 25
##    Category_name  `1998` `1999` `2000` `2001` `2002` `2003` `2004` `2005` `2006`
##    <chr>           <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 All Other Exp~  7339.  7737.  8175.  8849.  9576.  9220.  8881.  8791.  9359.
##  2 Capital Impro~   256.   400.   828.   890.  1150.  1076.   833.   582.   493.
##  3 Central Manag~   424.   470.   483.   489.   485.   493.   579.   710.   689.
##  4 Commerce & Ec~   464.   524.   705.   763.   811.   830.   926.   422.   594.
##  5 Debt Service     485.   508.   544.   590.   640.   731.  1289.  1352.  1415.
##  6 Human Services  3889.  4060.  4167.  4483.  4495.  4409.  4599.  4688.  4811.
##  7 K-12 Education  5070.  5589.  6026.  6392.  6632.  6692.  7128.  7573.  7877.
##  8 Local Governm~  3476.  4032.  4366.  4472.  4300.  4119.  4257.  4674.  5204.
##  9 Medicaid        5398.  5767.  6442.  7187.  7292.  7901. 10061.  9530. 10466.
## 10 Public Safety    426.   446.   469.   491.   489.   502.   473.   531.   600.
## 11 State Employe~   609.   650.   795.   850.   896.  1139.  1243.  1330.  1421.
## 12 State Pension~   915.  1148.  1247.  1363.  1476.  1621.  9051.  1738.   890.
## 13 Tollway          367.   461.   435.   346.   393.   361.   417.   479.  1074.
## 14 Total Expendi~ 31075. 33657. 37132. 39850. 41821. 42350. 52708. 45024. 47655.
## 15 Transportation  1957.  1865.  2448.  2683.  3187.  3256.  2972.  2626.  2763.
## # ... with 15 more variables: `2007` <dbl>, `2008` <dbl>, `2009` <dbl>,
## #   `2010` <dbl>, `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,
## #   `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,
## #   `2020` <dbl>, `2021` <dbl>
```

Before I created the CAGR values for all categories and sources. Now we want CAGR values for the aggregated "All Other ___" group. We have to recalculate the 23 year CAGR and then left_join it to our top ten sources and top 13-ish expenditures:

If we want Tables 4a and 4b for the Fiscal Futures paper to match the categories used in Table 1 and 2, then we want to calculate the CAGR values for the combined expenditure and revenue sources. The CAGR/Growth section above calculates values for all categories.

Top 13 Expenditures

```
# CAGR values for largest expenditure categories and combined All Other Expenditures

calc_cagr <- function(df, n) {
  df <- expenditure_wide %>% # actually in long form, not wide yet. expenditure_wide2 is wide version
    arrange(Category_name, Year) %>%
    group_by(Category_name) %>%
    mutate(cagr = ((`Dollars` / lag(`Dollars`, n)) ^ (1 / n)) - 1)

  return(df)
}

cagr_23 <- calc_cagr(expenditure_wide, 23) %>%
  # group_by(Category) %>%
  summarize(cagr_23 = round(sum(cagr*100, na.rm = TRUE), 2))

cagr23_precovid <- expenditure_wide %>%
  filter(Year <= 2019) %>%
  calc_cagr(21) %>%
  summarize(cagr_21 = round(sum(cagr*100, na.rm = TRUE), 2))
```

```r
cagr_10 <- calc_cagr(expenditure_wide, 10) %>%
  filter(Year == 2021) %>%
  summarize(cagr_10 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_5 <- calc_cagr(expenditure_wide, 5) %>%
  filter(Year == 2021) %>%
  summarize(cagr_5 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_3 <- calc_cagr(expenditure_wide, 3) %>%
  filter(Year == 2021) %>%
  summarize(cagr_3 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_2 <- calc_cagr(expenditure_wide, 2) %>%
  filter(Year == 2021) %>%
  summarize(cagr_2 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_1 <- calc_cagr(expenditure_wide, 1) %>%
  filter(Year == 2021) %>%
  summarize(cagr_1 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

CAGR_expenditures_summary_majorcats <- data.frame(cagr_1, cagr_2, cagr_3, cagr_5, cagr_10, cagr_23 ) %>%
  select(-c(Category_name.1, Category_name.2, Category_name.3, Category_name.4, Category_name.5 )) %>%
  rename("Expenditure Category" = Category_name, "1 Year CAGR" = cagr_1, "2 Year CAGR" = cagr_2, "3 Year

CAGR_expenditures_summary_majorcats
```

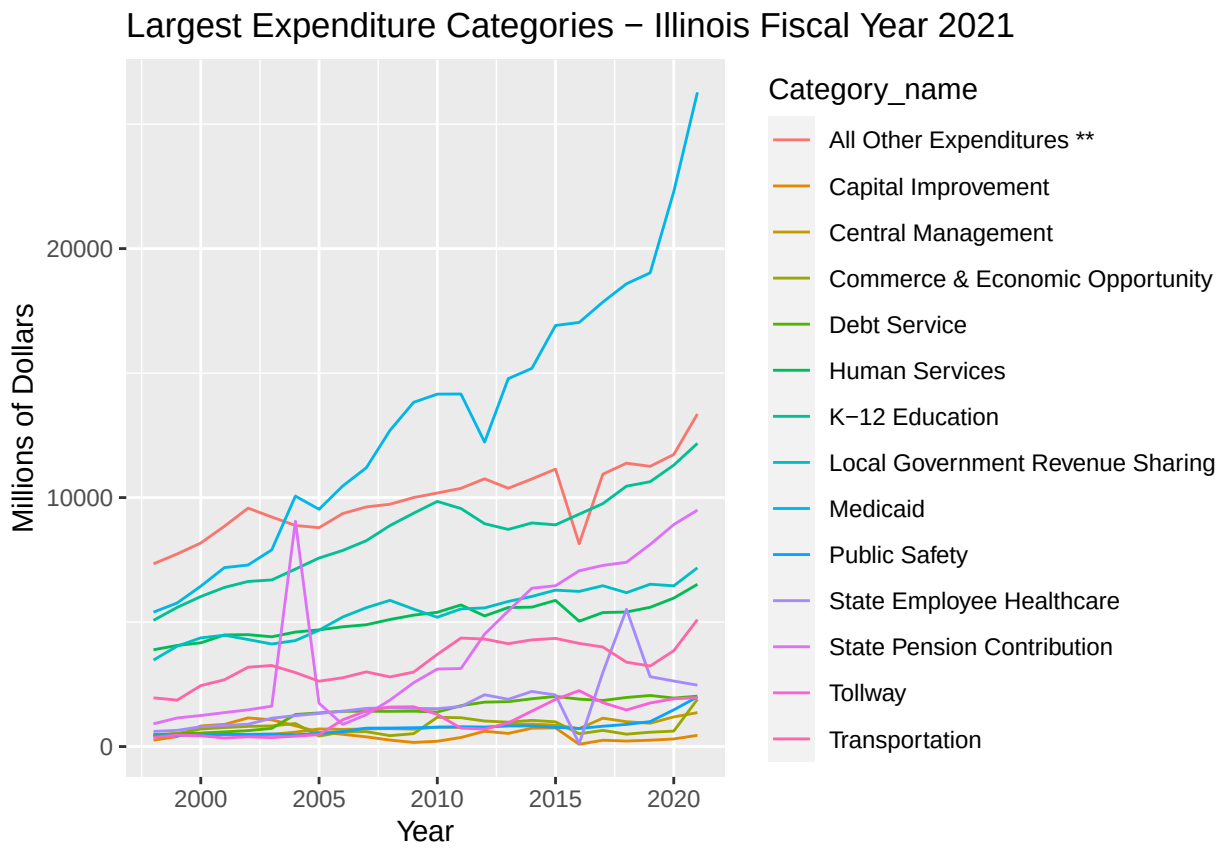
```
##                Expenditure Category 1 Year CAGR 2 Year CAGR 3 Year CAGR
## 1          All Other Expenditures **       13.85        8.94        5.47
## 2               Capital Improvement       47.65       33.13       26.68
## 3                Central Management       14.48       20.65       10.93
## 4    Commerce & Economic Opportunity      205.88       81.45       56.14
## 5                      Debt Service        4.06       -0.64        0.93
## 6                    Human Services        9.26        7.94        6.43
## 7                    K-12 Education        7.73        6.99        5.20
## 8  Local Government Revenue Sharing       11.20        4.93        5.11
## 9                           Medicaid       17.89       17.52       12.23
## 10                   Public Safety       35.19       41.13       30.90
## 11        State Employee Healthcare       -6.28       -6.24      -23.51
## 12       State Pension Contribution        6.52        8.17        8.66
## 13                          Tollway        2.38        5.89       10.19
## 14                Total Expenditure       14.47       11.84        7.45
## 15                   Transportation       32.40       25.59       14.58
##    5 Year CAGR 10 Year CAGR 23 Year CAGR
## 1        10.40         2.56         2.64
## 2        37.78         2.16         2.50
## 3        13.64         5.90         5.21
## 4        29.92         5.06         6.32
## 5         1.23         2.13         6.42
## 6         5.29         1.37         2.27
## 7         5.46         2.45         3.88
## 8         2.87         2.64         3.20
## 9         9.06         6.38         7.12
## 10       22.39         9.60         6.94
```

```
## 11       83.61         4.37         6.27
## 12        6.11        11.72        10.71
## 13       -2.66        10.25         7.56
## 14        7.83         4.42         4.85
## 15        4.23         1.58         4.25
```

```
# Yearly change for Top 13 largest expenditure categories
expenditure_change2 <- expenditure_wide %>%
  filter(Year > 2019) %>%
  pivot_wider(names_from = Year , values_from = Dollars,   names_prefix = "Dollars_") %>%
  mutate("FY 2021 Expenditures ($ billions)" = round(Dollars_2021/1000, digits = 1),
  #  "Change from 2020 to 2021" = Dollars_2021 - Dollars_2020,
         "Percent Change from 2020 to 2021" = round((Dollars_2021 -Dollars_2020)/Dollars_2020*100, digi
  left_join(cagr_23) %>%  # add long term CAGR
  arrange(-`FY 2021 Expenditures ($ billions)`)%>%
  select(-c(Dollars_2020, Dollars_2021)) %>%
  rename( "Compound Annual Growth, 1998-2021*" = `cagr_23`,
          "FY2021 Expenditure Category" = Category_name)

expenditure_change2
```

```
## # A tibble: 15 x 4
##    `FY2021 Expenditure Cate~` `FY 2021 Expen~` `Percent Chang~` `Compound Annu~`
##    <chr>                                <dbl>            <dbl>            <dbl>
##  1 Total Expenditure                     92.3             14.5             4.85
##  2 Medicaid                              26.3             17.9             7.12
##  3 All Other Expenditures **             13.4             13.8             2.64
##  4 K-12 Education                        12.2              7.73            3.88
##  5 State Pension Contribution             9.5              6.52           10.7
##  6 Local Government Revenue ~             7.2             11.2             3.2
##  7 Human Services                         6.5              9.26            2.27
##  8 Transportation                         5.1             32.4             4.25
##  9 State Employee Healthcare              2.5             -6.28            6.27
## 10 Debt Service                           2                4.06            6.42
## 11 Public Safety                          2               35.2             6.94
## 12 Tollway                                2                2.38            7.56
## 13 Commerce & Economic Oppor~             1.9            206.              6.32
## 14 Central Management                     1.4             14.5             5.21
## 15 Capital Improvement                    0.5             47.6             2.5
```

```
expenditure_wide %>%
  filter(Category_name!= "Total Expenditure") %>%
  ggplot(aes(x = Year, y = Dollars, group = Category_name, color=Category_name)) +
  geom_line()+
    xlab("Year") +
    ylab("Millions of Dollars")  +
    ggtitle("Largest Expenditure Categories - Illinois Fiscal Year 2021")
```



Top 10 Revenue Sources

Top 10 revenue sources CAGRs and Yearly Change Tables:

```
##### Top 10 revenue CAGRs: ####
# calculate just the 23 year cagr for our Yearly Change table that has the long run CAGR
calc_cagr <- function(df, n) {
  df <- revenue_wide %>% # actually in long format still but has small groups combined together
    arrange(Category_name, Year) %>%
    group_by(Category_name) %>%
    mutate(cagr = ((`Dollars` / lag(`Dollars`, n)) ^ (1 / n)) - 1)

  return(df)
}

cagr_23 <- calc_cagr(revenue_wide, 23) %>%
  summarize(cagr_23 = round(sum(cagr*100, na.rm = TRUE), 2))

cagr_10 <- calc_cagr(revenue_wide, 10) %>%
  filter(Year == 2021) %>%
  summarize(cagr_10 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_5 <- calc_cagr(revenue_wide, 5) %>%
  filter(Year == 2021) %>%
  summarize(cagr_5 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))
```

```
cagr_3 <- calc_cagr(revenue_wide, 3) %>%
  filter(Year == 2021) %>%
  summarize(cagr_3 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

cagr_2 <- calc_cagr(revenue_wide, 2) %>%
  filter(Year == 2021) %>%
  summarize(cagr_2 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

 cagr_1 <- calc_cagr(revenue_wide, 1) %>%
  filter(Year == 2021) %>%
  summarize(cagr_1 = case_when(Year == 2021 ~ round(sum(cagr*100, na.rm = TRUE), 2)))

CAGR_revenue_summary_majorcats <- data.frame(cagr_1, cagr_2, cagr_3, cagr_5, cagr_10, cagr_23) %>%
  select(-c(Category_name.1, Category_name.2, Category_name.3, Category_name.4, Category_name.5 )) %>%
  rename("Revenue Category" = Category_name, "1 Year CAGR" = cagr_1, "2 Year CAGR" = cagr_2, "3 Year CA

CAGR_revenue_summary_majorcats
```

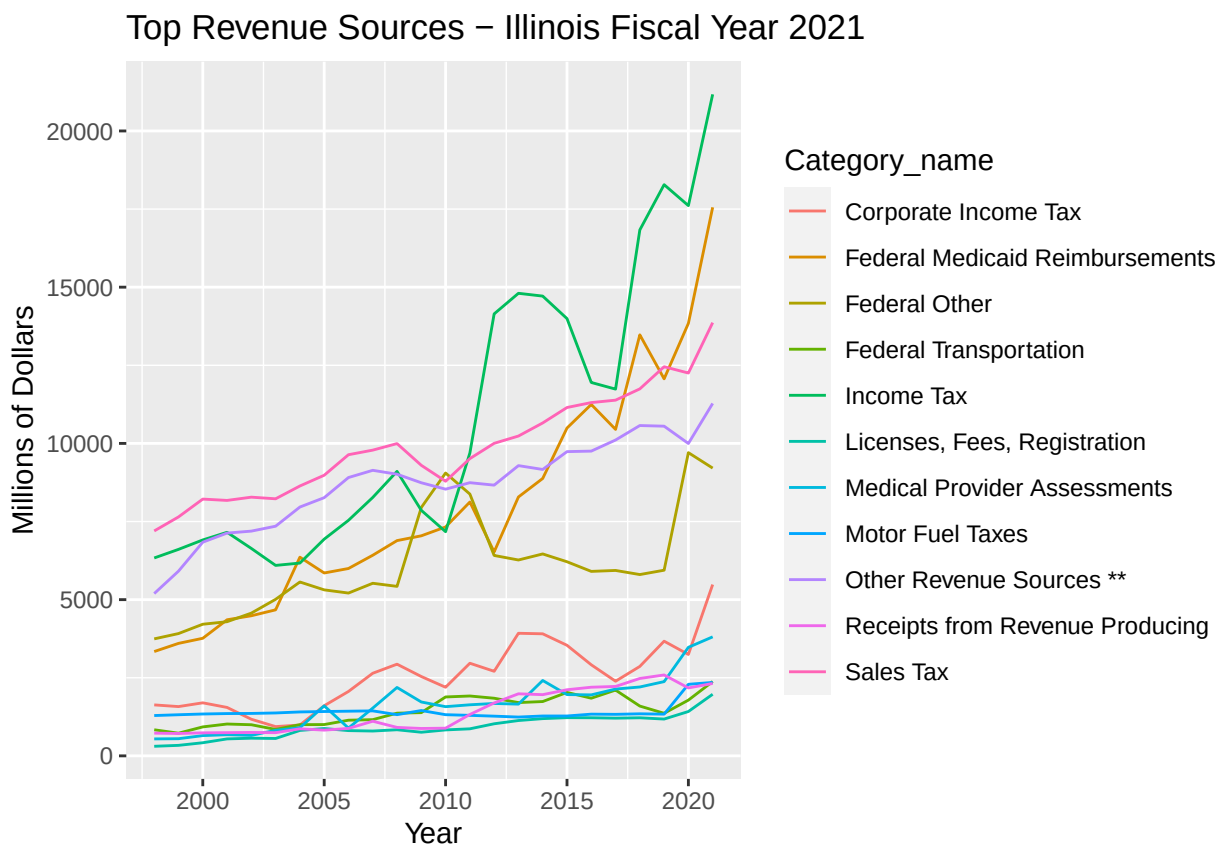
```
##                   Revenue Category 1 Year CAGR 2 Year CAGR 3 Year CAGR
## 1              Corporate Income Tax       68.97       22.22       24.19
## 2  Federal Medicaid Reimbursements       26.83       20.62        9.22
## 3                     Federal Other       -5.10       24.50       16.65
## 4           Federal Transportation       33.42       32.15       14.31
## 5                        Income Tax       20.22        7.62        7.95
## 6     Licenses, Fees, Registration       38.90       29.35       17.38
## 7     Medical Provider Assessments        9.66       26.62       20.00
## 8                 Motor Fuel Taxes        2.63       32.68       20.42
## 9         Other Revenue Sources **       12.82        3.39        2.18
## 10 Receipts from Revenue Producing        6.58       -5.41       -2.19
## 11                       Sales Tax       13.17        5.51        5.69
## 12                   Total Revenue       17.48       12.82        9.23
##    5 Year CAGR 10 Year CAGR 23 Year CAGR
## 1        13.47         6.34         5.41
## 2         9.31         8.01         7.48
## 3         9.30         0.95         3.99
## 4         5.22         2.19         4.66
## 5        12.12         8.12         5.39
## 6        10.08         8.57         8.43
## 7        14.31         8.83         8.84
## 8        11.98         6.13         2.64
## 9         2.95         2.58         3.43
## 10        1.07         5.75         5.16
## 11        4.16         3.84         2.89
## 12        8.20         5.31         4.79
```

```
###### Yearly change summary table for Top 10 Revenues #####
revenue_change2 <- revenue_wide %>%
  filter(Year > 2019) %>%
  pivot_wider(names_from = Year , values_from = Dollars,   names_prefix = "Dollars_") %>%
  mutate(
    "FY 2021 Revenues ($ billions)" = round(Dollars_2021/1000, digits = 1),
#    "Change from 2020 to 2021" = round(Dollars_2021 - Dollars_2020, digits = 2),
        "Percent Change from 2020 to 2021" = round(((Dollars_2021 -Dollars_2020)/Dollars_2020*100), di
```

```
  left_join(cagr_23) %>%
    arrange(-`FY 2021 Revenues ($ billions)`) %>%
  rename( "Compound Annual Growth, 1998-2021*" = `cagr_23`, # give nice labels for table output
          "FY2021 Revenue Category" = Category_name ) %>%
  select(-c(Dollars_2020, Dollars_2021))

revenue_change2
```

```
## # A tibble: 12 x 4
##    `FY2021 Revenue Category`  `FY 2021 Reven~` `Percent Chang~` `Compound Annu~`
##    <chr>                                <dbl>            <dbl>            <dbl>
##  1 Total Revenue                         91.4             17.5             4.79
##  2 Income Tax                            21.2             20.2             5.39
##  3 Federal Medicaid Reimburs~            17.6             26.8             7.48
##  4 Sales Tax                             13.9             13.2             2.89
##  5 Other Revenue Sources **              11.3             12.8             3.43
##  6 Federal Other                          9.2             -5.1             3.99
##  7 Corporate Income Tax                   5.5             69.0             5.41
##  8 Medical Provider Assessme~             3.8              9.66            8.84
##  9 Federal Transportation                 2.4             33.4             4.66
## 10 Motor Fuel Taxes                       2.4              2.63            2.64
## 11 Receipts from Revenue Pro~             2.3              6.58            5.16
## 12 Licenses, Fees, Registrat~             2               38.9             8.43
```

```
revenue_wide %>%
  filter(Category_name != "Total Revenue") %>% #uses only revenues
  ggplot(aes(x = Year, y = Dollars, group = Category_name, color = Category_name)) +
  geom_line()+
    xlab("Year") +
    ylab("Millions of Dollars")  +
    ggtitle("Top Revenue Sources - Illinois Fiscal Year 2021")
```



Export Summary Files

Saves main items in one excel file named `summary_file.xlsx`. Delete `eval=FALSE` to run on local computer.

```
#install.packages("openxlsx")
library(openxlsx)

dataset_names <- list('Aggregate Revenues' = revenue_wide2, # Top Categories aggregated, nice labels
                      'Aggregate Expenditures' = expenditure_wide2,


                      'Table 1' = expenditure_change2, #Top categories with yearly change, 23 yr cagr
                      'Table 2' = revenue_change2,

                      'Table 4.a' = CAGR_revenue_summary_majorcats, # Categories Match Table 1 in paper
                      'Table 4.b' = CAGR_expenditures_summary_majorcats,

                      'Table 1-AllCats' = expenditure_change_allcats,  # All Categories by Year
                      'Table 2-AllCats' = revenue_change_allcats,

                      'Table 4.a-AllCats' = CAGR_revenue_summary_allcats,
                      'Table 4.b-AllCats' = CAGR_expenditures_summary_allcats,

                      'year_totals' = year_totals,    # Total Revenue, Expenditure, and Fiscal gap per
```

```
                        'aggregated_totals_long' = aggregated_totals_long # all data in long format. Good
                        )

write.xlsx(dataset_names, file = 'summary_file_AWM_Oct10v2.xlsx')
```