Research Analyst III

Ian Stavros

2025-03-06

Below is a list of the dependencies I used for the code

First, I need to import the data to work with:

#Import the data to work with  
historical\_personal\_income\_tax\_return\_data <- read.csv("./data/Historical\_personal\_income\_tax\_return\_statistics\_by\_county\_and\_tax\_year\_20250306.csv", stringsAsFactors = FALSE)

Next, I determined which variables I wanted to work with, After Looking at the table I decided that a good set to show trends over time would be:s

variable\_table <- data.frame(  
 Trend\_Variables = c("Tax Year"),  
 Regional = c("County"),  
 "Breakdown Variables" = c("Filer Types % (Single, Joint, Separate)", "Total Exemptions in Thousands", "Average Exemptions Per Return", "Total Credits", "Average Credit Dollars")  
)  
  
kable(variable\_table)

| Trend\_Variables | Regional | Breakdown.Variables |
| --- | --- | --- |
| Tax Year | County | Filer Types % (Single, Joint, Separate) |
| Tax Year | County | Total Exemptions in Thousands |
| Tax Year | County | Average Exemptions Per Return |
| Tax Year | County | Total Credits |
| Tax Year | County | Average Credit Dollars |

Below is the generation of the data set’s I will be working with. Filer Type, Exemptions and Credits

# Select only necessary columns for return types  
filtered\_data <- historical\_personal\_income\_tax\_return\_data %>%  
 select(Tax.Year, County, Number.of.Returns, Return.Type..Single...., Return.Type..Joint....., Return.Type..Separate....)  
  
grouped\_data <- filtered\_data %>%  
 group\_by(County) %>%  
 arrange(Tax.Year, .by\_group = TRUE)  
  
formatted\_data <- filtered\_data %>%  
 pivot\_longer(cols = c(Return.Type..Single...., Return.Type..Joint....., Return.Type..Separate....),   
 names\_to = "Return.Type",   
 values\_to = "Percentage") %>%  
 mutate(Return.Type = case\_when(  
 Return.Type == "Return.Type..Single...." ~ "Single %",  
 Return.Type == "Return.Type..Joint....." ~ "Joint %",  
 Return.Type == "Return.Type..Separate...." ~ "Separated %"  
 ))  
  
# Group data by county and tax year, then extrapolate the average total exemptions and exeemptions per return.  
exemptions\_data <- historical\_personal\_income\_tax\_return\_data %>%  
 group\_by(County, Tax.Year) %>%  
 summarise(  
 Average\_Exemptions = mean(Average.Number.of.Exemptions, na.rm = TRUE),  
 Total\_Exemptions = sum(Total.Number.of.Exemptions..thousands.of.dollars., na.rm = TRUE),  
 Total\_Returns = sum(Number.of.Returns, na.rm = TRUE)  
 ) %>%  
 mutate(  
 Exemption\_Value\_Per\_Filing = Total\_Exemptions \* 1000 / Total\_Returns,  
 Exemption\_Value\_Per\_Filing\_Per\_Exemption = Exemption\_Value\_Per\_Filing/Average\_Exemptions  
 )

## `summarise()` has grouped output by 'County'. You can override using the  
## `.groups` argument.

#collecting the credits data  
credits\_data <- historical\_personal\_income\_tax\_return\_data %>%  
 group\_by(County, Tax.Year) %>%  
 summarise(  
 Total\_Credits\_In\_Thousands = Total.Credits..thousands.of.dollars.,  
 Average\_Credits\_dollars = Average.Credits..dollars.  
 )

## `summarise()` has grouped output by 'County'. You can override using the  
## `.groups` argument.

Next I generated a breakdown of for each variable into a graph to show me trends over time.

generate\_single\_line\_graph <- function(title, data\_frame, x\_axis\_title, x\_values, y\_axis\_title, y\_values) {  
 line\_graph <- ggplot(data\_frame, aes(x = .data[[x\_values]], y = .data[[y\_values]])) +  
 geom\_line(size = 1, color = "blue") +  
 geom\_point(size = 2, color = "lightgrey") + # Removed extra comma  
 labs(title = title,  
 x = x\_axis\_title,  
 y = y\_axis\_title) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(color = "lightgrey"),  
 axis.title = element\_text(color = "lightgrey"),  
 axis.text = element\_text(color = "lightgrey"),  
 legend.title = element\_text(color = "lightgrey"),  
 legend.text = element\_text(color = "lightgrey")  
 )  
 return(line\_graph)  
}  
  
# Get a list of unique counties  
unique\_counties <- unique(formatted\_data$County)  
  
# Loop through each county and generate a separate plot  
for (county in unique\_counties) {  
 filer\_type\_county\_data <- formatted\_data %>% filter(County == county)  
 exemption\_county\_data <- exemptions\_data %>% filter(County == county)  
 credits\_county\_data <- credits\_data %>% filter(County == county)  
   
 # Generate plot for this county  
 filer\_type\_plot\_over\_time <- ggplot(filer\_type\_county\_data, aes(x = Tax.Year, y = Percentage, color = Return.Type, group = Return.Type)) +  
 geom\_line(size = 1) +  
 geom\_point(size = 2) +  
 labs(title = paste("Filing Status Trends for", county),  
 x = "Tax Year",  
 y = "Percentage of Returns",  
 color = "Filing Type") +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(color = "lightgrey"),  
 axis.title = element\_text(color = "lightgrey"),  
 axis.text = element\_text(color = "lightgrey"),  
 legend.title = element\_text(color = "lightgrey"),  
 legend.text = element\_text(color = "lightgrey")  
 )  
  
 returns\_title = paste("Total Tax Returns in", county)  
 returns\_plot\_over\_time <- generate\_single\_line\_graph(  
 title = returns\_title,  
 data\_frame = filer\_type\_county\_data,  
 x\_axis\_title = "Tax Year",  
 x\_values = "Tax.Year",  
 y\_axis\_title = "Total Number of Returns",  
 y\_values = "Number.of.Returns"  
 )  
   
 #Generate Exemption Data Plots Over Time  
 total\_exemption\_title = paste("Total $ Exemptions per year in", county)  
 total\_exemption\_plot\_over\_time <- generate\_single\_line\_graph(  
 title = total\_exemption\_title,  
 data\_frame = exemption\_county\_data,  
 x\_axis\_title = "Tax Year",  
 x\_values = "Tax.Year",  
 y\_axis\_title = "Exemptions Total (in thousands)",  
 y\_values = "Total\_Exemptions"  
 )  
   
 exemption\_total\_per\_return\_title = paste("Exemption Total per Tax Return in", county)  
 exemption\_value\_per\_filing\_plot\_over\_time <- generate\_single\_line\_graph(  
 title = exemption\_total\_per\_return\_title,  
 data\_frame = exemption\_county\_data,  
 x\_axis\_title = "Tax Year",  
 x\_values = "Tax.Year",  
 y\_axis\_title = "Dollar Value of Exemptions per Return",  
 y\_values = "Exemption\_Value\_Per\_Filing"  
 )  
   
 average\_exemption\_title = paste("Average Exemptions per Return in", county)  
 average\_exemption\_plot\_over\_time <- generate\_single\_line\_graph(  
 title = average\_exemption\_title,  
 data\_frame = exemption\_county\_data,  
 x\_axis\_title = "Tax Year",  
 x\_values = "Tax.Year",  
 y\_axis\_title = "Avg Exemptions Per Return",  
 y\_values = "Average\_Exemptions"  
 )  
   
 average\_exemption\_value\_tile = paste("Average Dollar Value per Exemption per Return in", county)  
 average\_exemption\_value\_per\_return\_over\_time <- generate\_single\_line\_graph(  
 title = average\_exemption\_value\_tile,  
 data\_frame = exemption\_county\_data,  
 x\_axis\_title = "Tax Year",  
 x\_values = "Tax.Year",  
 y\_axis\_title = "Average Dollar Value per Exemption",  
 y\_values = "Exemption\_Value\_Per\_Filing\_Per\_Exemption"  
 )  
   
 total\_credits\_title <- paste0("Total Credits per Return in", county)  
 total\_credits\_plot\_over\_time <- average\_exemption\_value\_per\_return\_over\_time <- generate\_single\_line\_graph(  
 title = total\_credits\_title,  
 data\_frame = credits\_county\_data,  
 x\_axis\_title = "Tax Year",  
 x\_values = "Tax.Year",  
 y\_axis\_title = "Credit Total (in thousands)",  
 y\_values = "Total\_Credits\_In\_Thousands"  
 )  
   
 #Generate Credit Plots Over Time  
 average\_credits\_title = paste("Average Credits per Return in", county)  
 average\_credits\_plot\_over\_time <- average\_exemption\_value\_per\_return\_over\_time <- generate\_single\_line\_graph(  
 title = average\_credits\_title,  
 data\_frame = credits\_county\_data,  
 x\_axis\_title = "Tax Year",  
 x\_values = "Tax.Year",  
 y\_axis\_title = "Average Dollar Value per Credits",  
 y\_values = "Average\_Credits\_dollars"  
 )  
}

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

These graphs can be seen at:  
graph\_repo: [Click here to visit RStudio](https://posit.co/)  
slide\_summary: [Click here to visit RStudio](https://posit.co/)

determine\_trend\_stats <- function(county, data, start\_year, end\_year, trend\_type, trend\_focus) {  
 # Build dynamic column names  
 p\_value\_key <- ifelse(is.na(trend\_type), "p\_value", paste0("p\_value\_", trend\_type))  
 slope\_key <- ifelse(is.na(trend\_type), "slope", paste0("slope\_", trend\_type))  
   
 # Create a default tibble with the proper column names  
 return\_data\_frame <- tibble(  
 !!p\_value\_key := NA,  
 !!slope\_key := NA  
 )  
   
 if (!is.na(start\_year) && !is.na(end\_year)) {  
 window\_data <- data %>%  
 filter(County == county, Tax.Year >= start\_year, Tax.Year <= end\_year)  
   
 if (nrow(window\_data) >= 2) {  
 model <- lm(Total\_Returns ~ Tax.Year, data = window\_data)  
 coefficients <- summary(model)$coefficients  
   
 return\_data\_frame <- tibble(  
 !!p\_value\_key := coefficients["Tax.Year", "Pr(>|t|)"],  
 !!slope\_key := coefficients["Tax.Year", trend\_focus]  
 )  
 }  
 }  
 return(return\_data\_frame)  
}  
  
  
get\_longest\_trend <- function(data, years, values, trend\_type, look\_ahead = 3, p\_threshold = 0.05) {  
 n <- length(values)  
 longest\_trend <- 0  
 trend\_start <- NA  
 trend\_end <- NA  
 current\_trend\_start <- NA  
   
 # Define the primary trend condition: for a "downward" trend, each value must be lower than its predecessor.  
 primary\_condition <- if (trend\_type == "downward") `<` else `>`  
   
 i <- 2  
 while (i <= n) {  
 if (primary\_condition(values[i], values[i - 1])) {  
 # Continuing primary trend: initialize the trend start if needed.  
 if (is.na(current\_trend\_start)) {  
 current\_trend\_start <- years[i - 1]  
 }  
 current\_length <- years[i] - current\_trend\_start  
 if (current\_length > longest\_trend) {  
 longest\_trend <- current\_length  
 trend\_start <- current\_trend\_start  
 trend\_end <- years[i]  
 }  
 i <- i + 1  
 } else {  
 # A reversal is detected at index i.  
 # Look ahead up to look\_ahead years to see if there are two consecutive reversal years.  
 reversal\_indices <- c()  
 for (j in i:min(n, i + look\_ahead - 1)) {  
 if (!primary\_condition(values[j], values[j - 1])) {  
 # Only count indices if they are consecutive.  
 if (length(reversal\_indices) == 0 || j == tail(reversal\_indices, 1) + 1) {  
 reversal\_indices <- c(reversal\_indices, j)  
 } else {  
 break # Non-consecutive break in reversal sequence.  
 }  
 } else {  
 break # Encountered a value that resumes the primary trend.  
 }  
 }  
   
 if (length(reversal\_indices) >= 2) {  
 # Two or more consecutive reversal years are available.  
 reversal\_years <- years[reversal\_indices]  
 window\_data <- data %>% filter(County == county, Tax.Year %in% reversal\_years)  
 if (nrow(window\_data) >= 2) {  
 model <- lm(Total\_Returns ~ Tax.Year, data = window\_data)  
 p\_value <- summary(model)$coefficients["Tax.Year", "Pr(>|t|)"]  
 if (!is.na(p\_value) && p\_value < p\_threshold) {  
 # The opposite trend in this reversal window is statistically significant.  
 # End the primary trend at the last year before the reversal window began.  
 break  
 } else {  
 # The reversal window, though two years long, is not statistically significant.  
 # Skip the reversal window and continue the primary trend.  
 i <- tail(reversal\_indices, 1) + 1  
 next  
 }  
 } else {  
 # Insufficient data in the reversal window—skip these years.  
 i <- tail(reversal\_indices, 1) + 1  
 next  
 }  
 } else {  
 # Only one reversal year is found: treat it as an outlier and continue the primary trend.  
 i <- i + 1  
 }  
 # Reset the current trend start because the primary trend was interrupted.  
 current\_trend\_start <- NA  
 }  
 }  
   
 return(data.frame(  
 window = paste(trend\_start, '-', trend\_end),  
 trend\_start = trend\_start,  
 trend\_end = trend\_end,  
 length = longest\_trend  
 ))  
}  
  
  
downtrend\_summary <- exemptions\_data %>%  
 arrange(County, Tax.Year) %>%  
 group\_by(County) %>%  
 summarise(  
 res = list(get\_longest\_trend(exemptions\_data, Tax.Year, Total\_Returns, 'downward'))  
 ) %>%  
 unnest(res)  
  
downtrend\_summary\_signif <- downtrend\_summary %>%  
 rowwise() %>%  
 mutate(  
 trend\_down = list(determine\_trend\_stats(County, exemptions\_data, trend\_start, trend\_end, NA, "Estimate"))  
 ) %>%  
 unnest(c(trend\_down), names\_repair = "unique") %>%  
 ungroup() %>%  
 filter(  
 !is.na(p\_value), p\_value < 0.05, slope < 0,  
 ) %>%  
 select(County, window, p\_value, slope)  
  
kable(downtrend\_summary\_signif)

| County | window | p\_value | slope |
| --- | --- | --- | --- |

# Create a dataset combining exemptions and number of filings  
exemptions\_with\_filing\_type <- historical\_personal\_income\_tax\_return\_data %>%  
 group\_by(County, Tax.Year) %>%  
 summarise(  
 Average\_Exemptions = mean(Average.Number.of.Exemptions, na.rm = TRUE),  
 Total\_Filings = sum(Number.of.Returns, na.rm = TRUE),  
 Single\_Percent = mean(Return.Type..Single...., na.rm = TRUE),  
 Joint\_Percent = mean(Return.Type..Joint....., na.rm = TRUE),  
 Separate\_Percent = mean(Return.Type..Separate...., na.rm = TRUE),  
 Exemptions\_Total\_Thousands = sum(Total.Number.of.Exemptions..thousands.of.dollars., na.rm = TRUE)  
 ) %>%  
 ungroup()

## `summarise()` has grouped output by 'County'. You can override using the  
## `.groups` argument.

# Linear regression including filing type  
lm\_model\_filing <- lm(Average\_Exemptions ~ Tax.Year + Total\_Filings + Single\_Percent + Joint\_Percent + Separate\_Percent,   
 data = exemptions\_with\_filing\_type)  
  
summary(lm\_model\_filing) # Reports p-values

##   
## Call:  
## lm(formula = Average\_Exemptions ~ Tax.Year + Total\_Filings +   
## Single\_Percent + Joint\_Percent + Separate\_Percent, data = exemptions\_with\_filing\_type)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.159109 -0.051437 -0.008503 0.047768 0.168746   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.160e+00 1.237e+00 -1.745 0.0814 .   
## Tax.Year 3.222e-03 6.084e-04 5.297 1.57e-07 \*\*\*  
## Total\_Filings 5.364e-07 4.529e-08 11.844 < 2e-16 \*\*\*  
## Single\_Percent -4.204e+00 1.107e-01 -37.966 < 2e-16 \*\*\*  
## Joint\_Percent -1.093e+00 1.098e-01 -9.956 < 2e-16 \*\*\*  
## Separate\_Percent -1.029e+00 6.733e-01 -1.528 0.1269   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07001 on 713 degrees of freedom  
## (19 observations deleted due to missingness)  
## Multiple R-squared: 0.7905, Adjusted R-squared: 0.789   
## F-statistic: 538 on 5 and 713 DF, p-value: < 2.2e-16

# Conclusions

The regression model from the previous slide indicates that over time, Average Exemptions tend to increase. However, higher percentages of Single and Joint filings are associated with a decrease in Average Exemptions, while Separate filings do not show a significant impact. Despite these findings, the model’s R² of 0.79 suggests that about 21% of the variability in Average Exemptions remains unexplained—highlighting the need for further research to identify additional contributing factors.

## Further Analysis

* A more granular analysis is currently limited by the lack of detailed return-level data. Having access to this information would enable us to pinpoint the specific job sectors most affected in each county and explain why some counties took longer to recover from the 2008 financial crisis.
* To gain a deeper understanding of revenue dynamics, I propose merging this data with datasets on Social Determinants of Health (SDOH), SOGI, and REALD. Integrating these sources may reveal key indicators that can provide a more detailed understanding of where revenue is being collected, from whom, and why, thereby enriching our overall analysis.