

# User Classification and Stats

William Lee

2023-07-05

## Introduction

Given the importance of carefully constructing the treatment and control groups for this research project, I figured it would be a good idea to write up the procedure and discuss the results all in one document, with particular emphasis on definitions for eligibility and sample sizes.

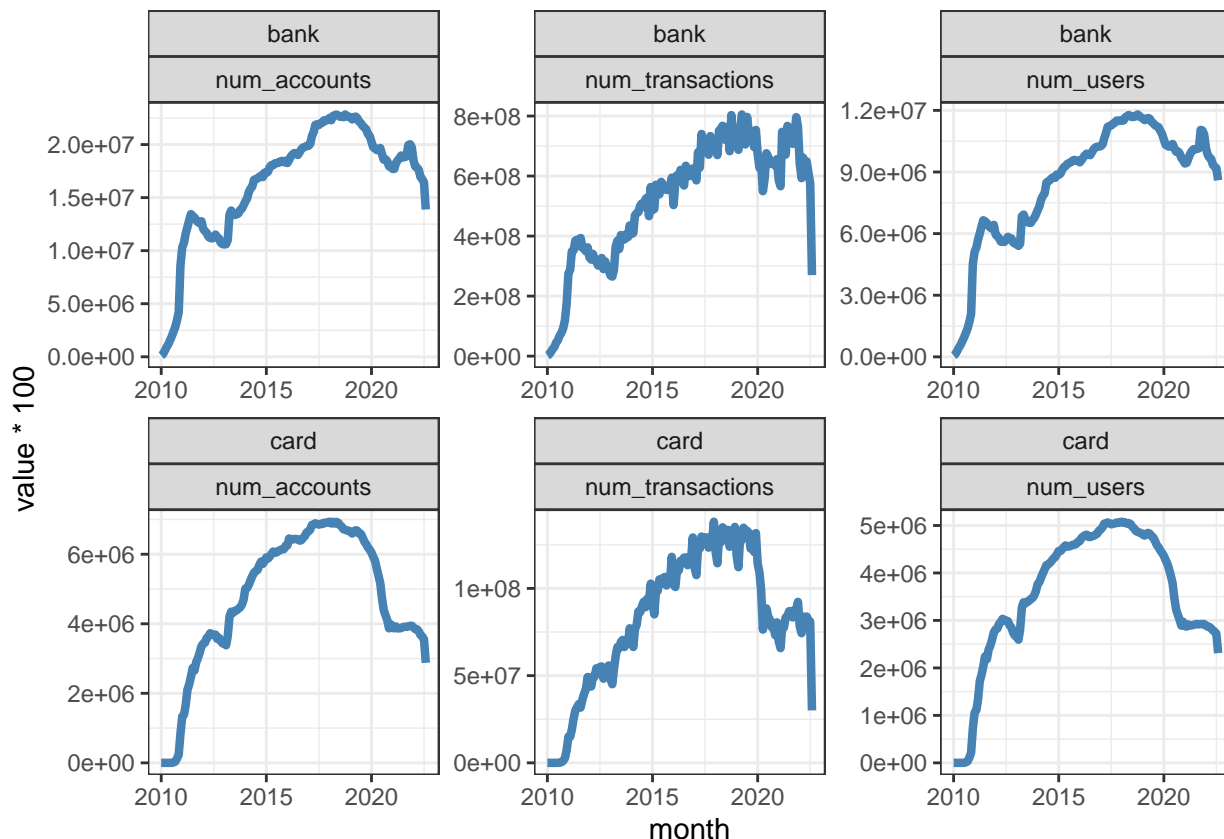
## Simple Data Set Statistics

The Yodlee data set is indexed by `unique_mem_ids` which are generated by Yodlee after merging the data from all of their subcontractors. We don't exactly know how the number of unique Yodlee members translates to unique human beings or households. Naturally, Yodlee doesn't reveal how they translate their raw data into unique member ids, however we will assume that each unique member id is truly one individual (or perhaps a joint account). Since Yodlee matches across data sets from different banks and credit card companies, they are likely matching on SSNs.

As of the August 2022 Yodlee batch, there are **56,474,837** distinct `unique_mem_ids` in the bank panel and **19,133,674** `unique_mem_ids` in the card panel. Below is the plot showing how the count of unique ids by month changes over time. The sample size grows considerably over time with a few minor dips presumably due to vendor contracts expiring. Throughout our analysis, we will have to keep in mind that we do not have a perfectly balanced sample, and that an individual's presence in the data is not necessarily random.<sup>1</sup>

---

<sup>1</sup>As an exercise for a later date, we can look to see how the composition of the users changes from regional/internet/brokerage/credit union/national bank using the `account_type` variable



Since we are not able to directly observe whether or not a given user was eligible/enrolled in the Payroll Tax Deferment, we have to infer eligibility from the transaction data and refine our sample so that we have high-quality treatment and control groups. So far, we are working under the assumption that an eligible federal worker is one who:

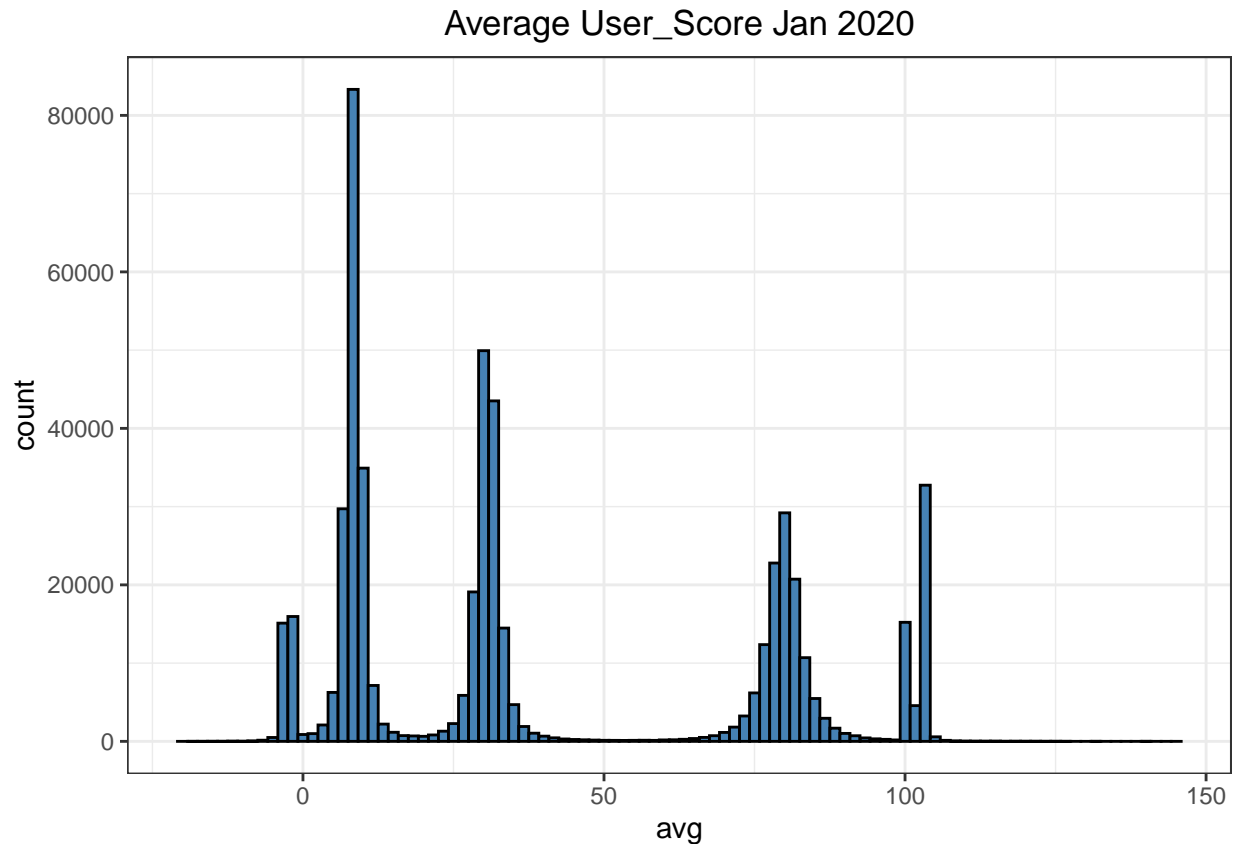
1. Has an average Yodlee Score of at least 6.5 from Jan 2020 - Jan 2020 . (This is Yodlee's suggested value for a 'stable' user.)
2. Has qualifying payment observations (based on description, primary\_merchant, and amount fields).<sup>2</sup>
3. Observe no more than 20% of total inflows from other sources of income (i.e. SSI, Venmo transactions, transfers from other accounts)
4. Has qualifying payment observations from a single employer at regular intervals (weekly, biweekly, or monthly) from Jan 2020 to Jan 2022.
5. Makes between \$2,500 - \$8666.67 per month (\$1,153.85 - \$4,000 for biweekly). We rule out individuals making too little as they are unlikely to be engaged in stable, full-time work. (Consider lowering the upper threshold by a certain amount to account for typical withholding). QUESTION: Does eligibility change during the deferral? For example, if you receive a scheduled pay raise in October 2020 that puts you above the threshold, do you still participate in the program?)
6. Observe no more than 35% volatility between paychecks to rule out employees with varying hours worked, travel reimbursements, etc.
7. Be able to link all credit card payments to credit cards in the Yodlee sample. Otherwise, we will not be observe debt-funded consumption.

<sup>2</sup>transaction must be greater than \$500, from an identified vendor, marked as Salary/Regular Income by Yodlee and not marked as a duplicate transaction

A state/local employee, will match the exact same definition but with a qualifying state/local string.

## Sample Size Estimates

In this section, I try to estimate what percent of users are disqualified at each successive component of the definition. Starting with a 1% random sample of the 20220816 Yodlee panel of transactions on or after ‘2018-08-01’, I have 306,646 unique individuals, 617,592 bank accounts 301,561,634 transactions.



## Filter for User Score

The first layer of filtering is for the user\_score. Below, I plot the histogram of the average monthly user\_score for January 2020 (a random month before Covid). I’m not sure how Yodlee decides these user\_scores, but there are some clear clusters of user\_scores. The user\_score varies a lot even within a single month (average range of user\_score was 45.55). I would love to know why user\_scores vary so much in a short amount of time since things like “User History, presence of a linked account, and number of merchants” shouldn’t change dramatically in a short amount of time.

After removing all individuals with an average user\_score of less than 6.5 over the sample period, we are down to 199,549 users, representing a 34.93 percent reduction in the sample. Perhaps it’s worth exploring a higher threshold, especially since the main analysis may involve daily time coefficients.

## Filter for Qualifying Strings

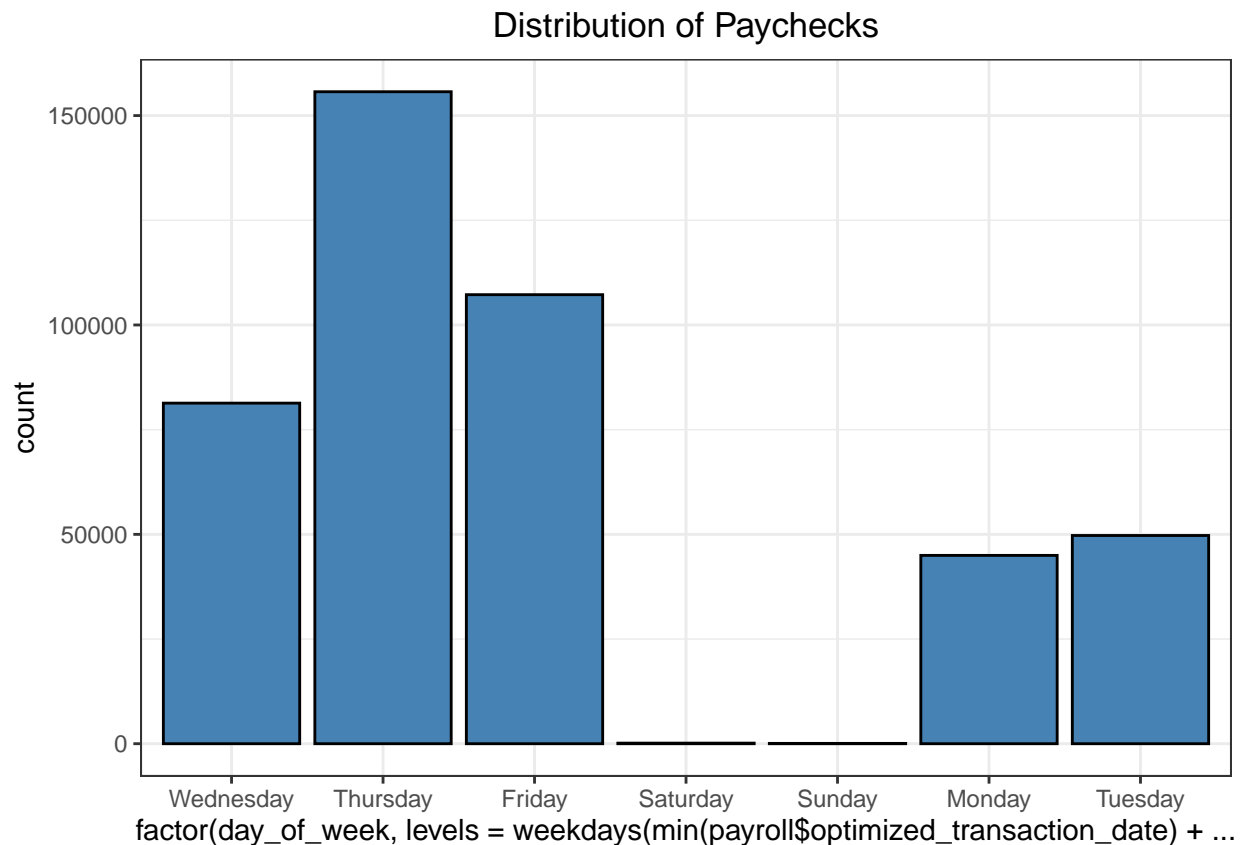
## Filter for Single Employer, Number of Paychecks

The second layer of filtering is to sort each individual into federal/state/local based on the description and primary merchant. The full .sql used is included in the appendix, but uses all of the patterns we have found up to this point. After filtering, we are left with 13,576 individuals which represents a 93.2 percent reduction in eligible individuals. 10508 are federal, 1260 are state, and 2091 are local.

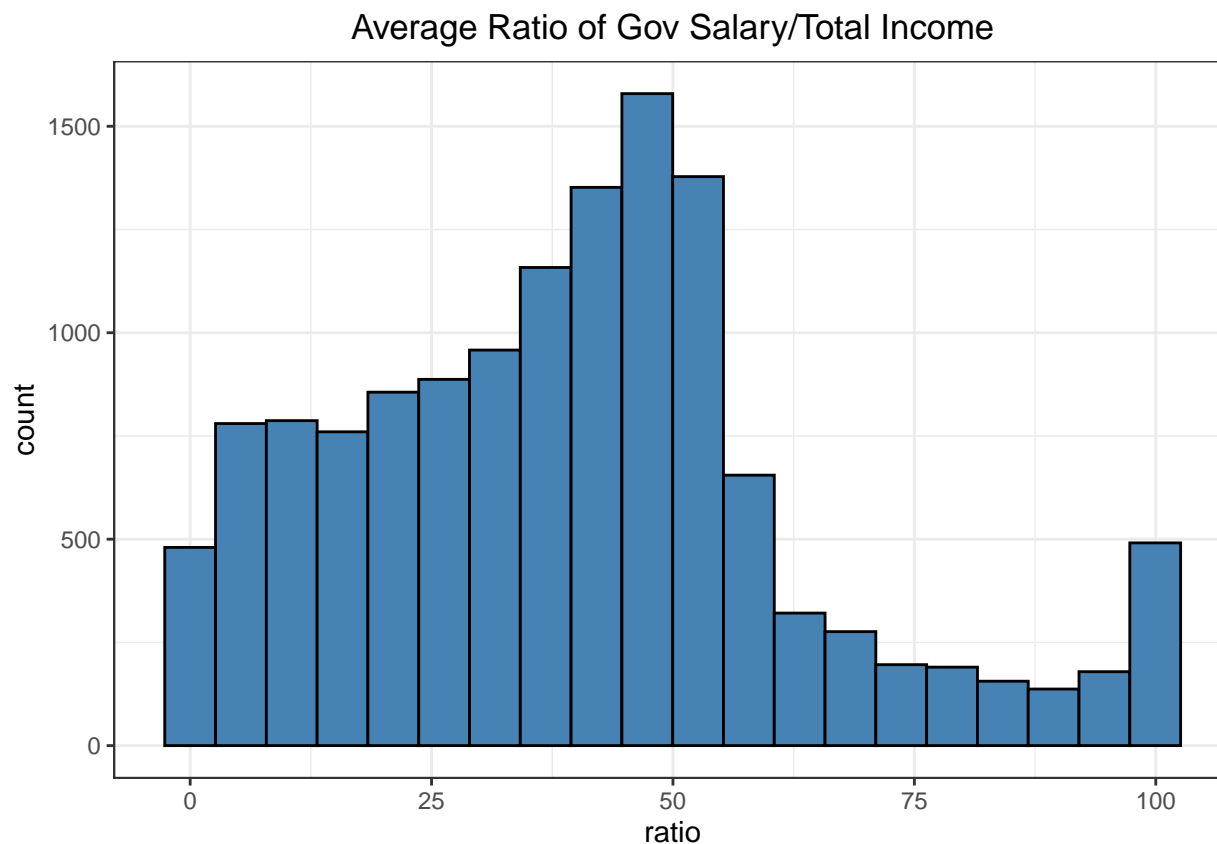
I tried to match these figures with outside estimates of the federal/state/local workforce. Despite the lack of a single definition for federal worker, this Brookings piece has similar estimates to our data (~15% in some sort of federal/state/local government, ~6% in federal).

Below is a histogram of the day of week our individuals receive paychecks. I would have thought many more would have been on Friday. We will have to make some decisions about how to handle the fact banks process payroll at different speeds. Even within the same employer and employees with similar pay cycles, there is substantial heterogeneity in when the paychecks are processed. Also, when we move to weekly level analysis later, we will have to decide what counts a week (i.e. Monday-Monday, Friday-Friday, 7 days since last pay, etc.) For now I am considering a week as Monday-Monday.

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



## Filter for Outside Sources of Income



After isolating the salary payments, I calculate the ratio of government salary/total income, and the histogram of the ratio is plotted above. Total Income = ‘Interest Income’ + ‘Other Income’ + ‘Salary/Regular Income’ + ‘Sales/Services Income’. If I followed the definition we discussed, I would be eliminating almost all of the individuals from the sample, so the rest of the analysis proceeds without actually making the cut. I will wait for further feedback before deciding what to do on this. I’m not sure why the ratio is so low for some individuals - are we missing salary strings or is “other income” in Yodlee simply far too broad. For example, if somebody sells their used car, it might show up as “other income” even though we wouldn’t normally consider that as income. Also, it could hint at the presence of a second earner or large outside transfers.

After the third round of filtering, we are left with 5,818 individuals which represents a 57.14 percent reduction in eligible individuals. Most of the reduction is coming from the restriction on having four years of consistent paychecks from a single employer. Should we relax the definition a bit on the pre-period?

## Top Employers

Below is a list of the top 10 employers by each category:

- Some payments have no primary\_merchant but have FED SAL in the description. I’ve marked them as “FED SAL”
- We can only identify the branch of the military in a few situations (i.e. DFAS CLEVELAND = NAVY) but most of the descriptions are too censored.
- Do we want to keep obscure federal employers like the Bonneville Power Authority?

- I forgot what we decided on school boards and public education for local employers. I decided to keep for now
- For the state employers, I'm pretty certain we are missing something because California, Texas, and Florida should be the largest employers, not Maryland. Perhaps some states centralize payroll and others decentralize. Probably something worth figuring out at some point.
- Do we want the House of Representatives, US Senate strings

fed	primary_merchant_name_v2	n_unique_id	rank
federal	DFAS	3811	1
federal	U.S. Department Of The Treasury	685	2
federal	The U.S. Office Of Personnel Management	332	3
federal	United States Coast Guard	195	4
federal	USPS	73	5
federal	Agricultural Treasury Office	25	6
federal	FED SAL	19	7
federal	General Services Administration	10	8
federal	U.s. Department Of Health And Human Services	5	9
federal	US House	3	10
federal	Bonneville Power Administration	2	11
federal	Government Of Canada	1	12
local	City Of New York	20	1
local	Anne Arundel County Board	18	2
local	Montgomery County Government	15	3
local	Allegheny County	13	4
local	Baltimore County	11	5
local	City Of Chicago	10	6
local	City Of Columbus	8	7
local	Loudoun County Public Schools	8	8
local	Cook County	7	9
local	Gwinnett County Board Of Education	7	10
local	City Of Phila	5	11
local	Cumberland County	5	12
local	Franklin County	5	13
local	Fulton County Schools	5	14
local	Prince William County	5	15
state	State Of Maryland	46	1
state	Commonwealth of Pennsylvania	34	2
state	State Of California	17	3
state	State Of Florida	16	4
state	Washington State Treasurer	12	5
state	West Virginia State Treasurer	11	6
state	State Of Indiana	9	7
state	State Of North Carolina	9	8
state	State Of Arizona	8	9
state	State Of Missouri	8	10
state	State Of Alabama	7	11
state	State Treasurer	6	12
state	State of Connecticut	5	13
state	State Of Nebraska	5	14
state	State Of Alaska	4	15

Yodlee marks almost all Treas 310 transactions as US Treasury which is correct, but we might want to extract more information from each description. Below is a table of the sub-departments of the description.

I'm guessing that each code corresponds to the following: 015 (misc payments), COM2 (Dept of Commerce), DOI (Dept of Interior), DOT (Dept of Transportation), MCTF (some sort of military payment), STA (State Dept), and T (Treasury). I found a link to a full manual on Federal Payment services if we ever want to do more exploration. I do not know why there are two AGRI from different primary merchants or why there are two different OPM1's. <https://fiscal.treasury.gov/files/reference-guidance/green-book/greenbook-full.pdf>.

department	primary_merchant_name	n
015	U.S. Department Of The Treasury	1
AGRI	Agricultural Treasury Office	70
COM2	Us Treasury	1
DOI1	U.S. Department Of The Treasury	3
DOT4	U.S. Department Of The Treasury	1
MCTF	U.S. Department Of The Treasury	7768
OPM1	The U.S. Office Of Personnel Management	8852
STA	U.S. Department Of The Treasury	1
T	U.S. Department Of The Treasury	1
USCG	United States Coast Guard	132

I'd be interested to hear your thoughts on whether restricting our analysis to people who do not switch jobs or take side-jobs limits the external validity of our sample. On one hand, a refined sample will allow us to cleanly estimate the effects of the deferral on individual, but on the other hand, we will not be to make strong statements about the overall impact of the program.

### Filter for Income Thresholds

After the fourth layer of filtering which classes individuals by income thresholds, we are left with the following number of unique\_id\_mem's. The proportions seem about right, given that most government employees that have regular paychecks should make between 30,000 - 104,000 USD per year.

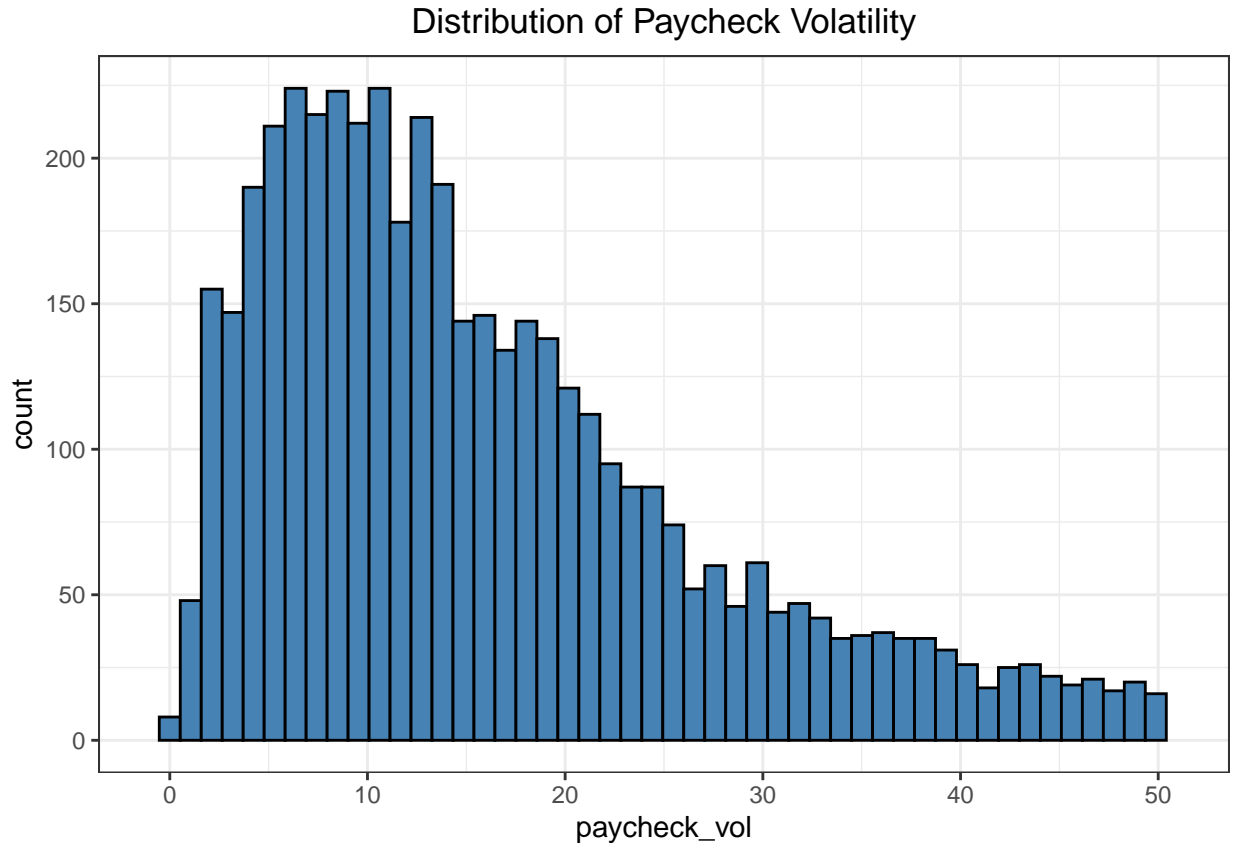
This is the section where we really need to be specific on dates, especially since a major result of this paper is to show how quickly the deferral translates into higher consumption.

[URGENT] How does the deferral work for people who have a partial pay period starting September 01, 2022? Do they get a partial benefit proportional to the pay cycle or does it only go into effect on whole paychecks? We should be able to figure this out by manually inspecting the data.

elig	federal	local	state
FALSE	403	7	4
TRUE	4007	291	133

### Filter for Paycheck Volatility

Below is a histogram of percentage volatility ( $\text{std}(\text{paycheck\_amt}) * 100 / \text{mean}(\text{paycheck\_amt})$ ). The threshold is entirely arbitrary, so I wanted to show the general distribution. I need to examine these individuals more to see why they have such volatility, but I think it's better to exclude them for now until we can better understand the trends. I'm guessing the common causes for paycheck volatility are pay raises, bonuses (are these common for government employees), changes in withholding, reimbursements, etc.



After the fifth round of filtering which removes individuals with high volatility in paychecks, we are left with the following number of individuals in each category.

elig	federal	local	state
FALSE	260	6	4
TRUE	3481	254	119

There are two more layers of filter: one for ruling out other sources of income and another for ensuring link-ability of Yodlee data. I have some ideas on how to do this efficiently, but I'd like your feedback first.

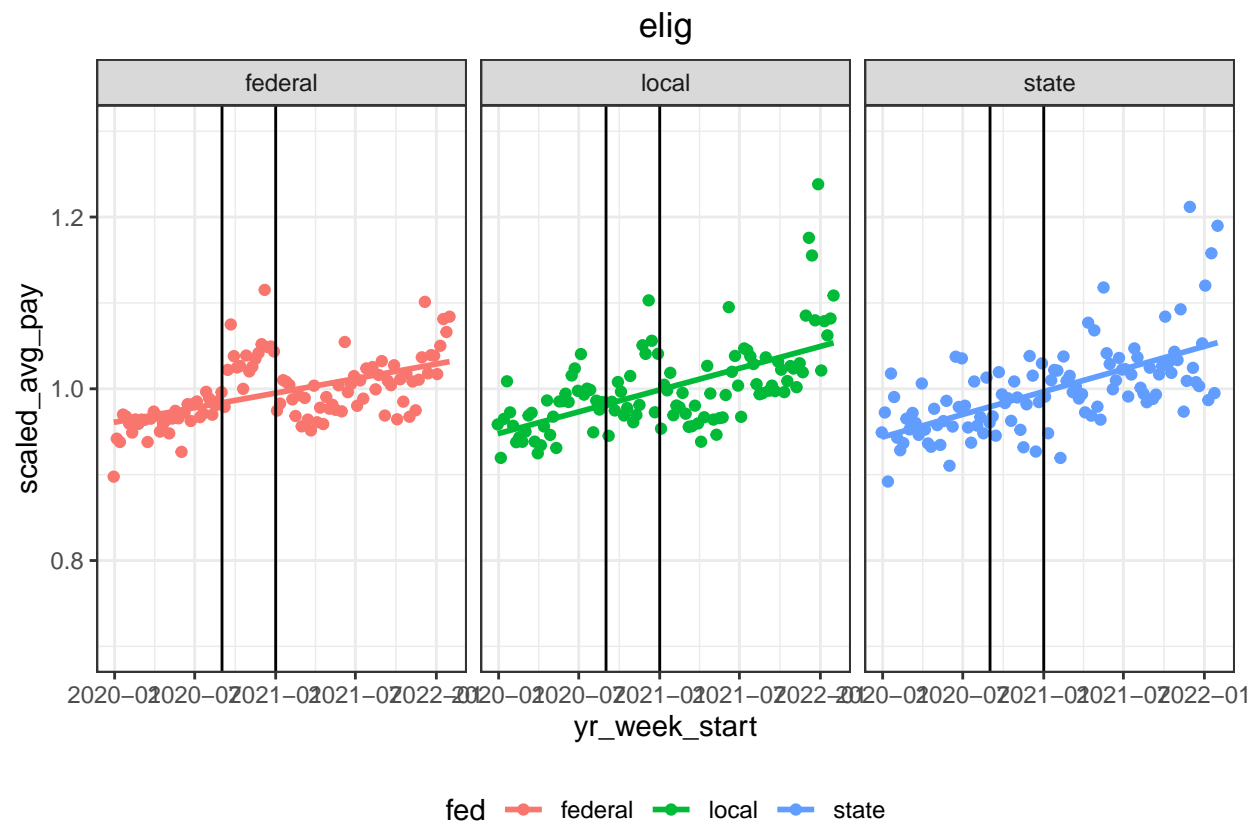
## Paycheck Graph

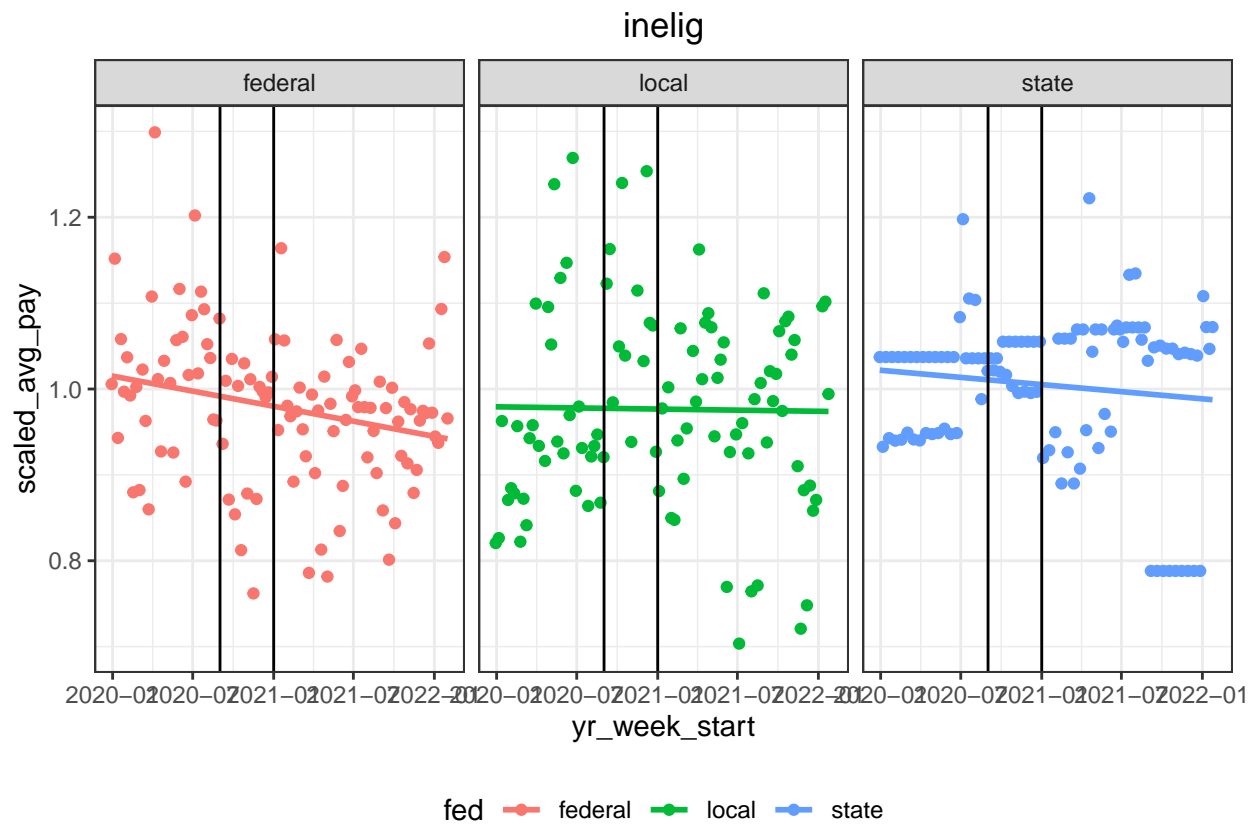
Here is the most important piece of the whole exercise. After all of this cleaning, we need to see if the deferral is visible in a simple weekly aggregate plot, and I think it looks pretty good.

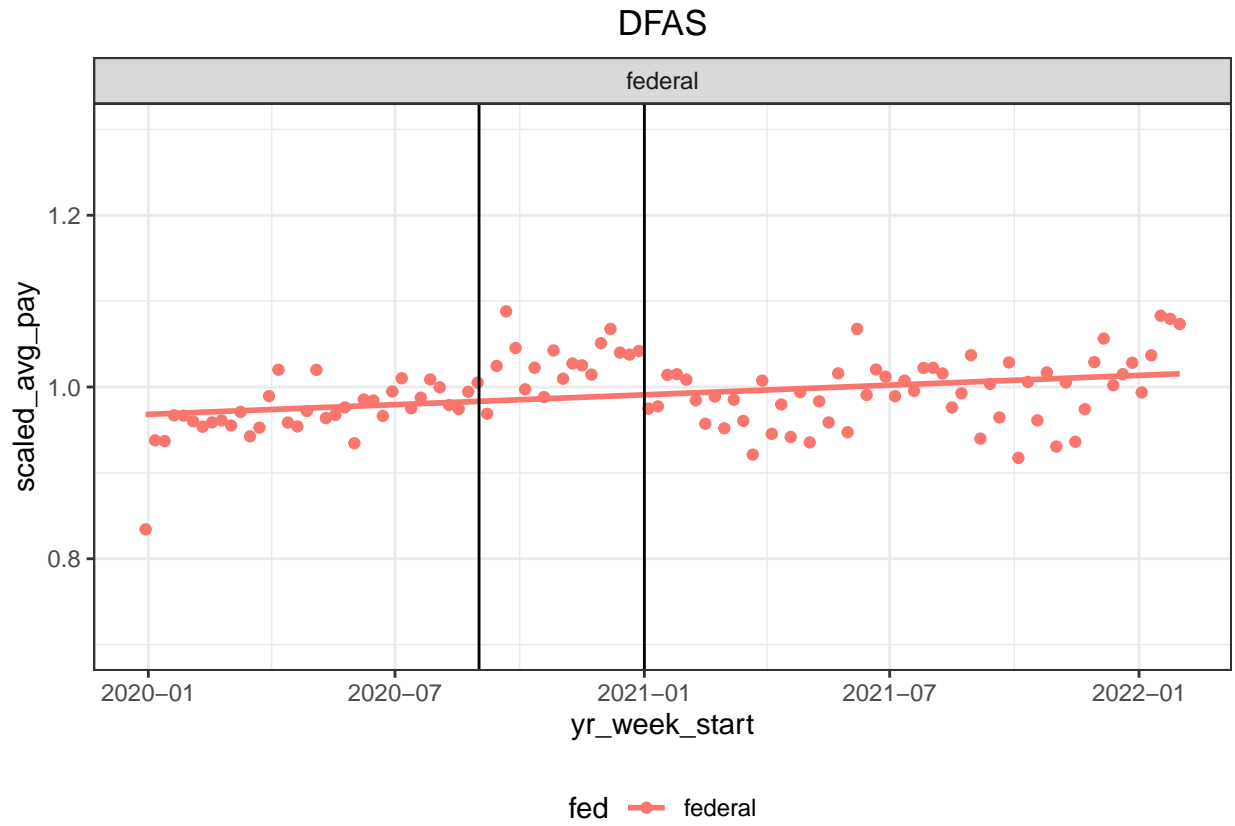
- The trend is clearly visible for eligible federal employees but absent for state and local employees.
- I'm concerned to see a small increase in the ineligible (due to the salary cutoff) federal employees during the payroll period. Perhaps the income thresholds are calculated over a longer period or I'm getting the annualization wrong. I need to explore this more.
- The dropoff in paychecks looks like it continues for a whole year (does this match what we were told from Adam's contacts)
- Clearly USPS is not included in the program
- Why are the trends so much stronger for DFAS than Agriculture and Treasury? Have I made a mistake on which payments to include in Federal? It's worrying that the clearest trends are in the military employees because those employees are the least comparable to state/local employees.

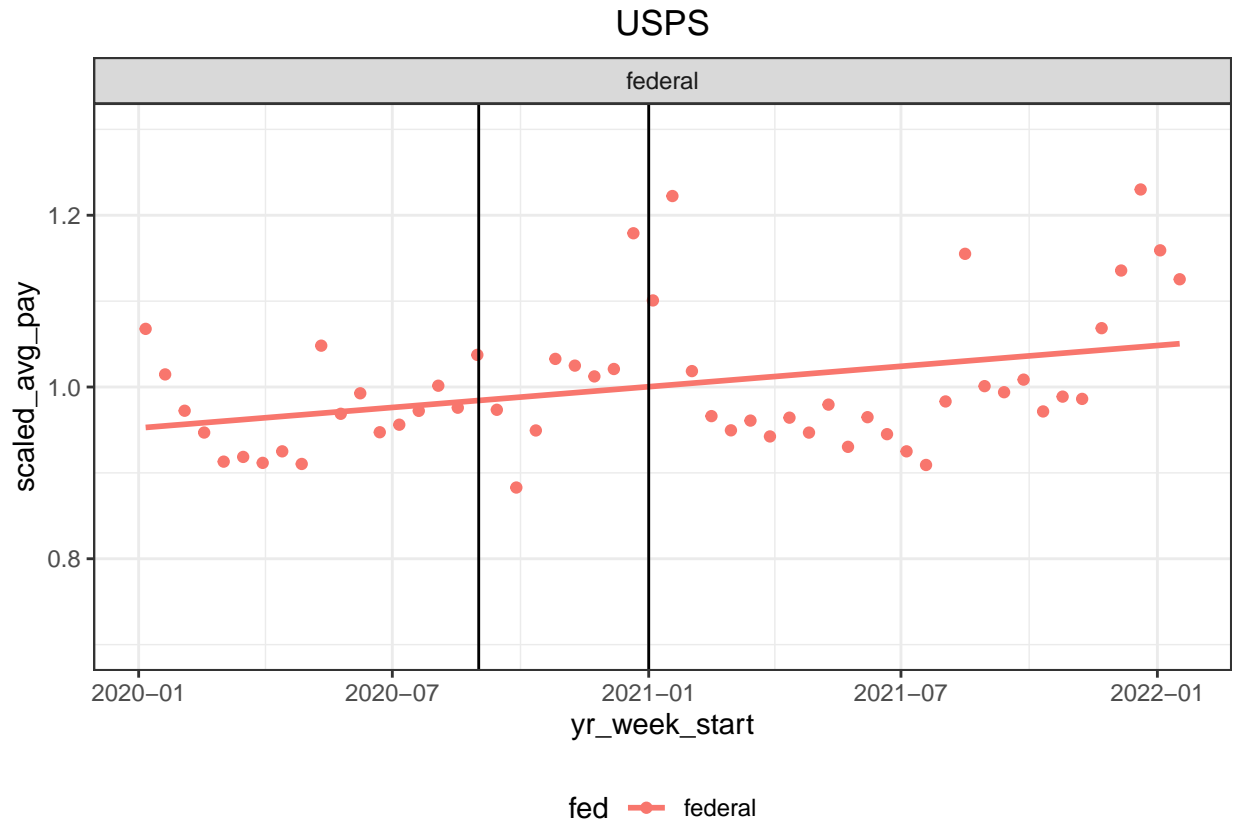


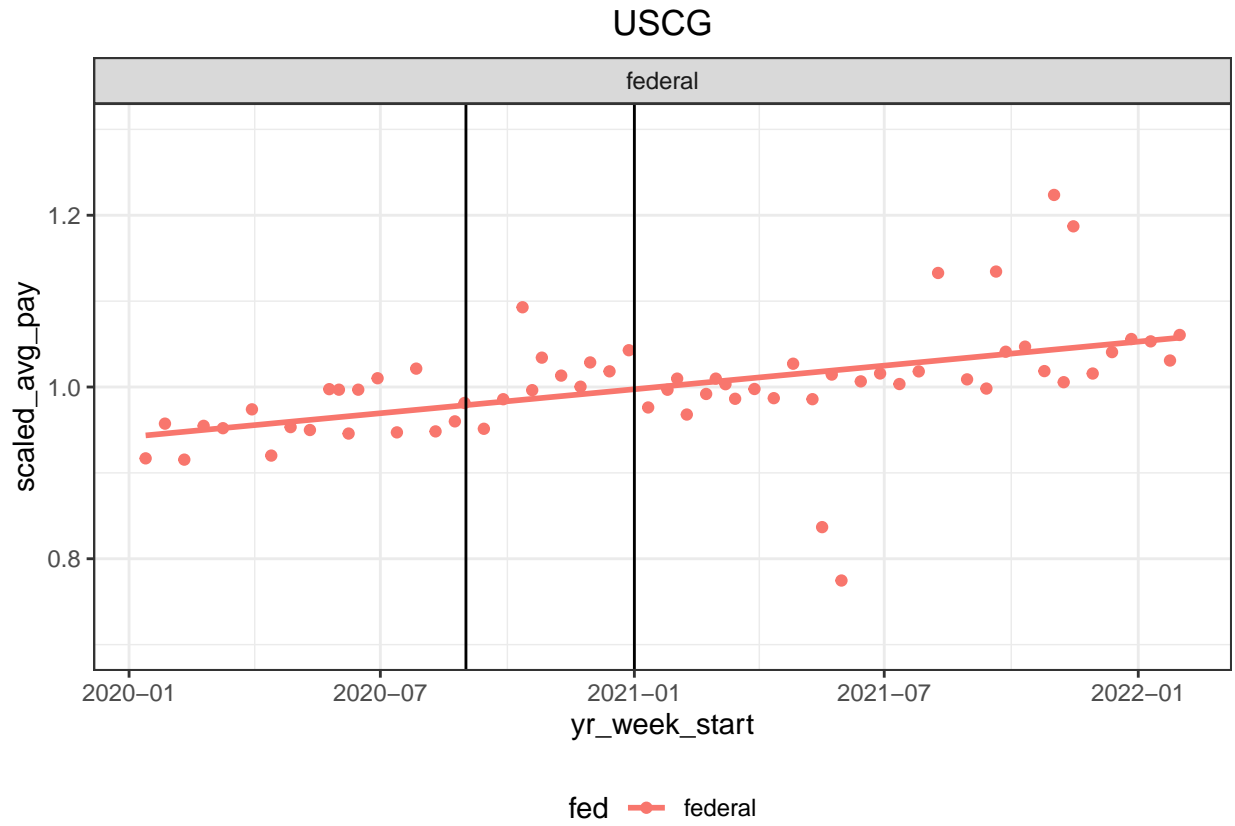
- Local employees have a clear increase at the very end of the year. This could be bonuses, overtime, or holiday pay incentives.

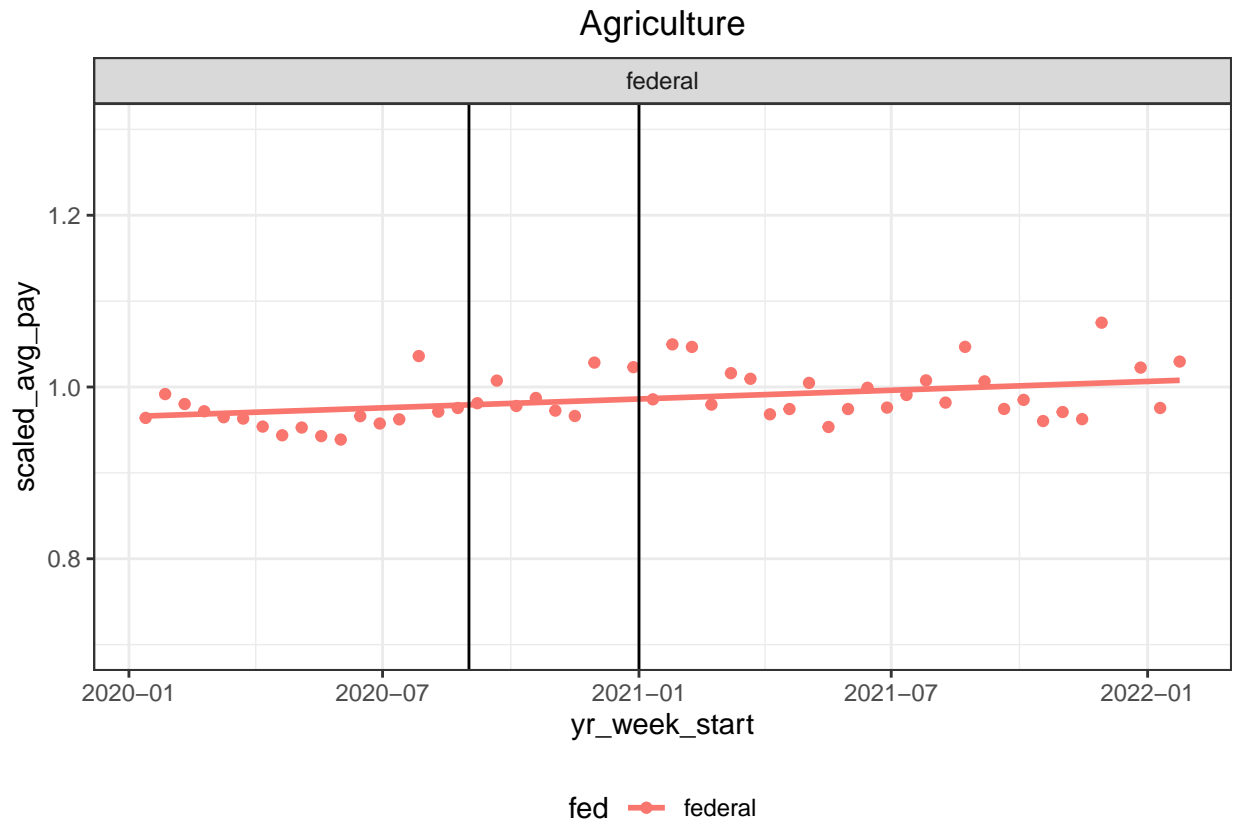


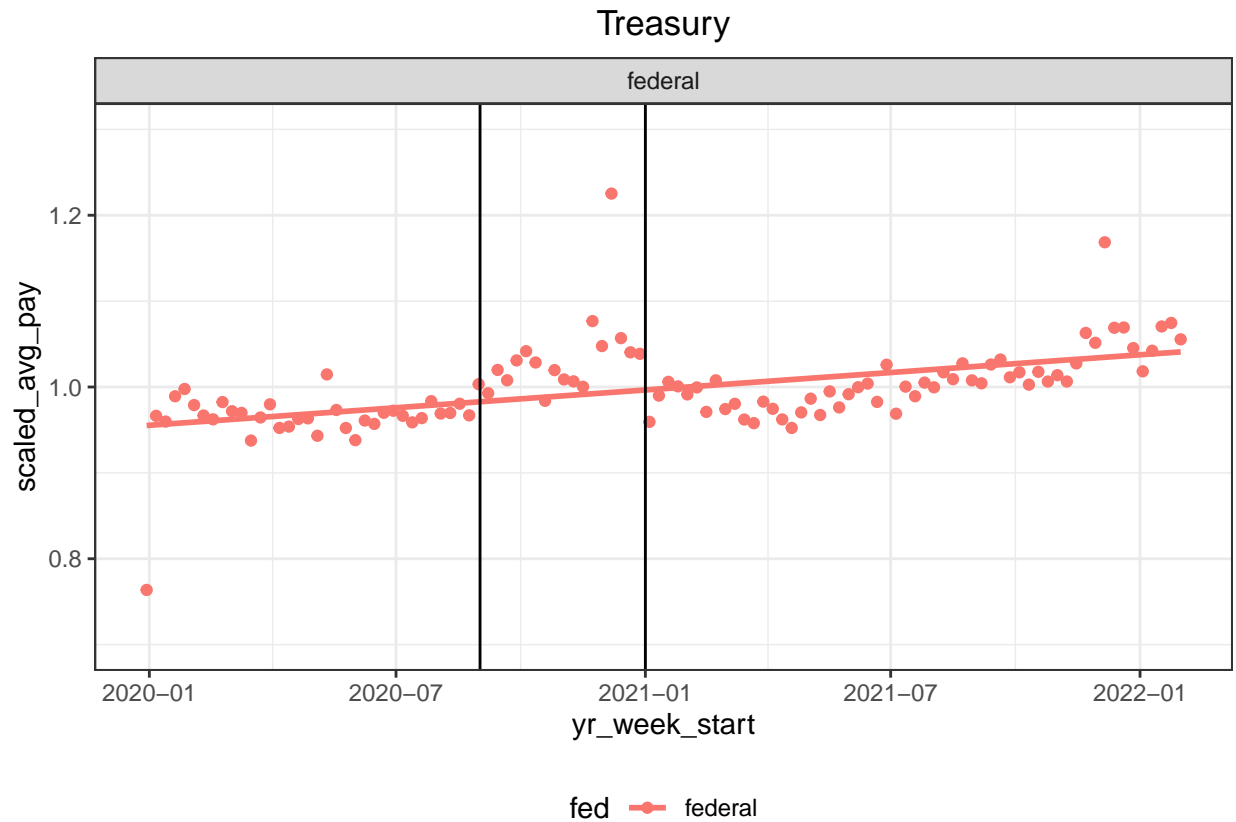








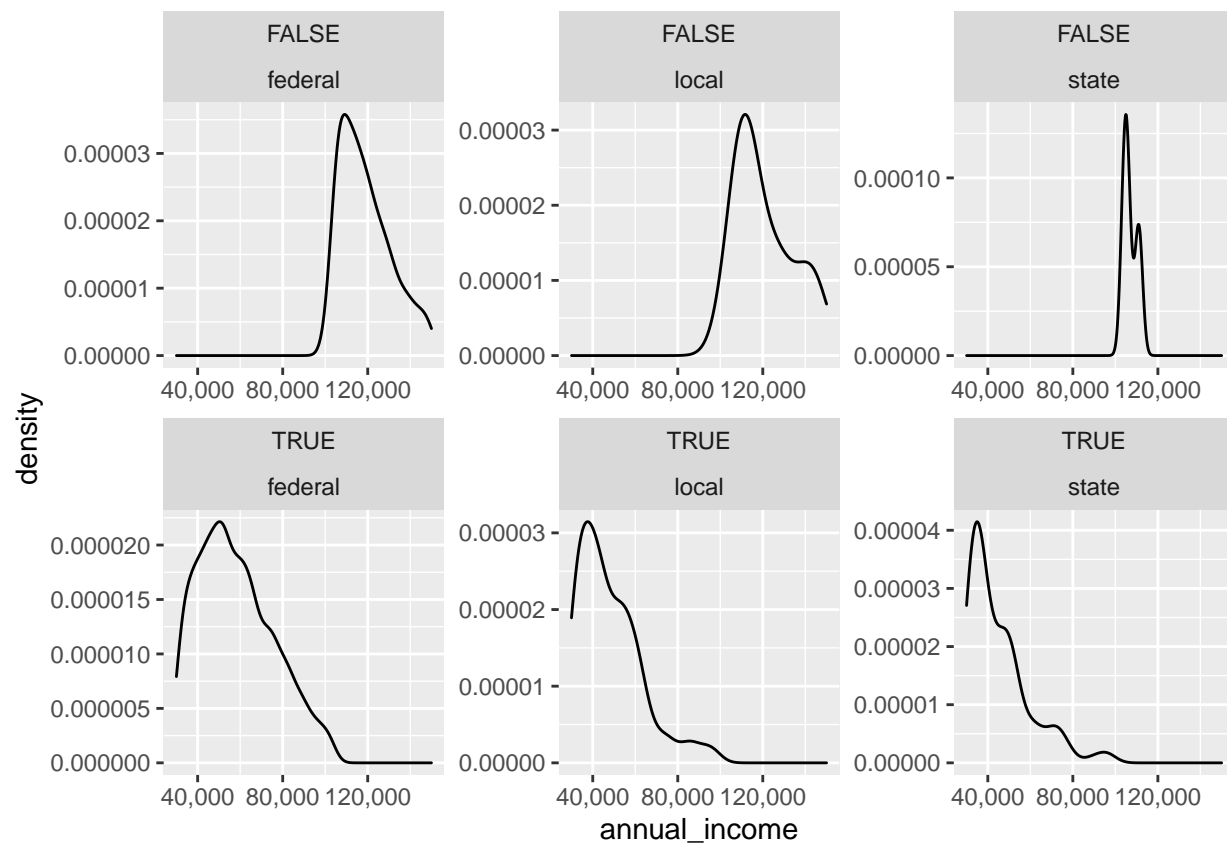




### Annual Income Stats

I wanted to make sure that annualizing the last paycheck yielded similar results to aggregating the last year of paychecks. It looks like annualizing the last paycheck is roughly correct, but there are still plenty of individuals who are “ineligible” judged by their last paycheck but clearly make less than 104k if we aggregate for the prior year. I’m also curious as to why there are some very highly paid state employees. I’m particularly concerned by the local

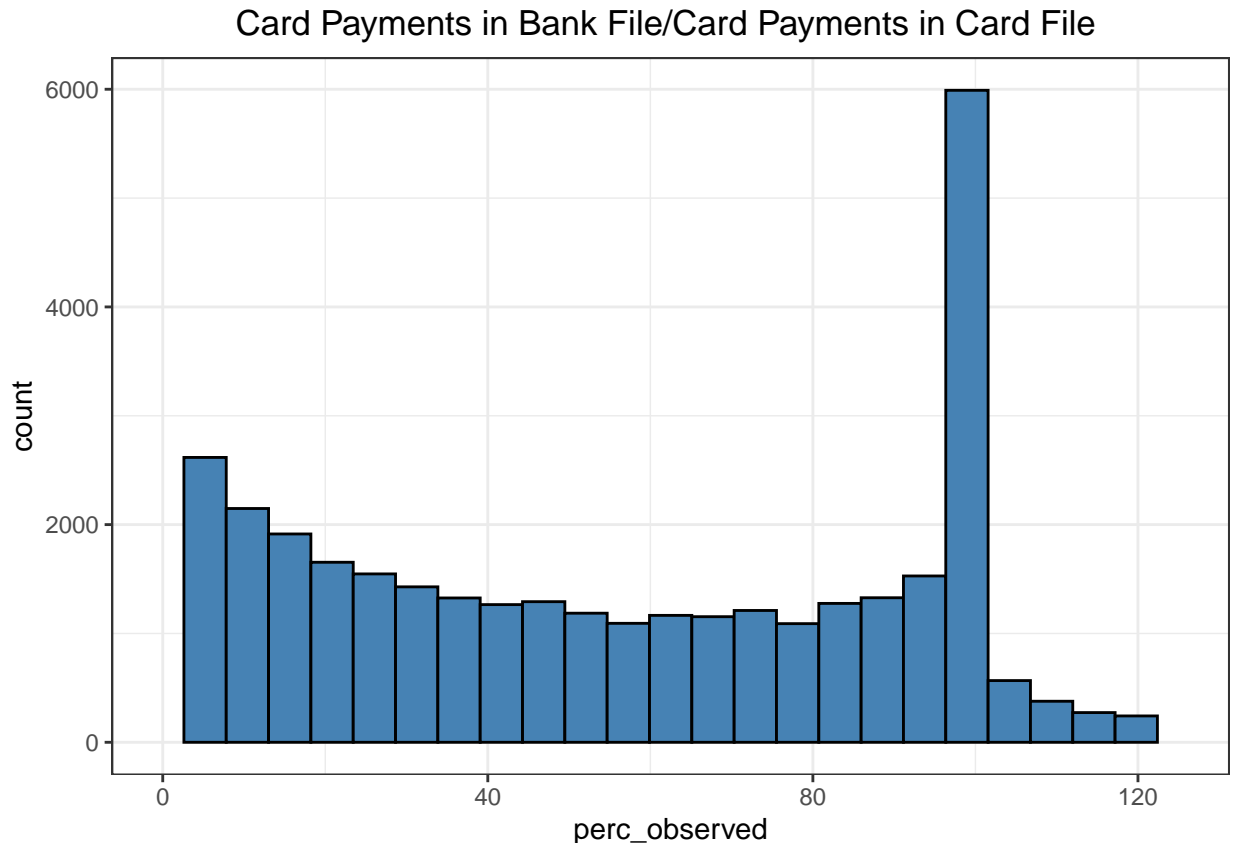
```
## Warning: Removed 34 rows containing non-finite values (stat_density).
```



```
## Warning: Removed 4637 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```





This is another section where I wasn't sure how you would want to handle it. If we were able to see all accounts, we would expect most people to be near 100 and definitely not over. There are a couple factors potentially at play:

- Yodlee is missing accounts
- Yodlee is bad at determining what is a credit card payment (DISCOVER CC PYMT is obvious a credit card but what about strings that are less obvious?)
- Shared accounts/business accounts are ruining the picture

## Appendix

You can find my code and outputs on my personal github page (<https://github.com/WilliamHLee104/PayrollTax>)[<https://github.com/WilliamHLee104/PayrollTax>].

```
-- Simple Count for Total Population
select count(distinct unique_mem_id) as num_users
from yi_xpanelov6_20220816.bank_panel

select count(distinct unique_mem_id) as num_users
from yi_xpanelov6_20220816.card_panel

-- Create 1% sample and store in my temp directory
CREATE TABLE temp_132.onepercsample AS (SELECT *
                                         FROM yi_xpanelov6_20220816.bank_panel
```

```

WHERE mod(unique_mem_id, 100) = 1
AND optimized_transaction_date >= '2018-08-01'
AND is_duplicate = 0)

-- Create 1% card sample and store in temp directory
CREATE TABLE temp_132.onepercsample_card AS (SELECT *
FROM yi_xpanelov6_20220816.card_panel
WHERE mod(unique_mem_id, 100) = 1
AND optimized_transaction_date >= '2018-08-01')

-- Count By Month (saved to count_by_month.csv)
(select count(distinct unique_mem_id) as num_users, count(distinct unique_bank_account_id) as num_accounts,
count(*) as num_transactions, 'bank' as source, month
from (select substring(optimized_transaction_date, 1, 7) as month, unique_mem_id, unique_bank_account_id
from yi_xpanelov6_20220816.bank_panel WHERE mod(unique_mem_id, 100) = 1 ) as month_create
GROUP BY month
ORDER BY month)
UNION ALL
(select count(distinct unique_mem_id) as num_users, count(distinct unique_card_account_id) as num_accounts,
count(*) as num_transactions, 'card' as source, month
from (select substring(optimized_transaction_date, 1, 7) as month, unique_mem_id, unique_card_account_id
from yi_xpanelov6_20220816.card_panel WHERE mod(unique_mem_id, 100) = 1) as month_create
GROUP BY month
ORDER BY month)

-- Count Users/Transactions in the 1% sample
select count(distinct unique_mem_id) as num_users,
count(distinct unique_bank_account_id),
count(*) as num_transactions
from temp_132.onepercsample

-- User Score Dist (saved to user_score_dist.csv)
select unique_mem_id, avg(user_score), min(user_score), max(user_score), count(*)
from temp_132.onepercsample
where optimized_transaction_date >= '2020-01-01'
AND optimized_transaction_date < '2020-02-01'
group by unique_mem_id

-- Filter By User Score
-- NB: Next round of filtering uses filter1 as a base table
create table temp_132.filter1 as (SELECT b.*
from (select unique_mem_id,
avg(user_score) as avg,
min(user_score) as min,
substring(min(optimized_transaction_date), 1, 7) as min_r,
substring(max(optimized_transaction_date), 1, 7) as max_r
from temp_132.onepercsample
where optimized_transaction_date >= '2020-01-01' and optimized_
group by unique_mem_id) a

```

```

                                inner join (select * from temp_132.onepercsample where optim
                                                on a.unique_mem_id = b.unique_mem_id
                                where avg > 6.5)

select count(distinct unique_mem_id)
from temp_132.filter1

-- Find all Qualifying Federal/State/Local Payroll Transactions
create table temp_132.payroll as (
select *
from (select *,
        CASE
            WHEN ((upper(primary_merchant_name) like '%DFAS%' OR
                  upper(primary_merchant_name) like '%U.S. DEPARTMENT OF THE TREASURY%' OR
                  upper(primary_merchant_name) like '%US TREASURY%' OR
                  upper(primary_merchant_name) like '%GOVERNMENT%' OR
                  upper(primary_merchant_name) like '%GSA%' OR
                  upper(primary_merchant_name) like '%THE GENERAL SERVICES ADMINISTRATION%' OR
                  upper(primary_merchant_name) like '%THE U.S. OFFICE OF PERSONNEL MANAGEMENT%' OR
                  upper(primary_merchant_name) like '%UNITED STATES COAST GUARD%' OR
                  upper(primary_merchant_name) like '%U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES%' OR
                  upper(primary_merchant_name) like '%AGRICULTURAL TREASURY OFFICE%' OR
                  upper(primary_merchant_name) like '%CENSUS%' OR
                  upper(primary_merchant_name) like '%SOCIAL SECURITY ADMINISTRATION%' OR
                  upper(primary_merchant_name) like '%FARM SERVICE AGENCY%' OR
                  description ilike '%FED SAL%'
            or description ilike '%FAA TREAS 310%'
            or description ilike '%EPA TREAS 310%'
            or description ilike '%GSA TREAS 310%'
            or description ilike '%DOI1 TREAS 310%'
            or description ilike '%DOT4 TREAS 310%'
            or description ilike '%NIH TREAS 310%' or description ilike '%NIH. TREAS 310%'
            or description ilike '%OPM1 TREAS 310%'
            or description ilike '%DHS TREAS 310%'
            or description ilike '%LOC1 TREAS 310%'
            or description ilike '%USSS TREAS 310%'
            or description ilike '%CBP TREAS 310%'
            or description ilike '%DOJ TREAS 310%'
            or description ilike '%USSS TREAS 310%'
            or description ilike '%US HOUSE OF REPR%'
            or description ilike '%US SENATE FED SAL%'
            or description ilike '%TENN VALLEY AUTH TRPDFEDSL%'
            or description ilike '%TENN VALLEY AUTH ACH: TRPDFEDSL%'
            or description ilike '%US SENATE FED SAL%'
            or description ilike '%USPS%'
            or description ilike '%IN AF PAY%'
            or description ilike '%IN ARMY ACT%'
            or description ilike '%IN AF PAY%'
            or description ilike '%IN AF RES%'
            or description ilike '%IN ARMY RC%')

```

```

or description ilike '%NAVY ACT%'
or description ilike '%NAVY ALT%'
or description ilike '%NAVY RES%')
AND description not ilike '%SSA TREAS 310%'
AND description not ilike '%SOC SEC%'
AND description not ilike '%VA BEN%'
AND description not ilike '%TREASURY PMN%'
AND description not ilike '%SERV F%'
AND description not ilike '%SUPP SEC%'
AND description not ilike '%US TREASURY CF%'
AND description not ilike '%TAX%'
AND description not ilike '%RET%'
AND description not ilike '%FED PAYMENT%'
AND description not ilike '%ALLT%'
AND description not ilike '%PPTAS%'
AND description not ilike '%BENEFIT PAYMENT%'
AND description not ilike '%TRAVEL PAY%'
AND description not ilike '%UI BEN%'
AND description not ilike '%USCIS%'
AND description not ilike '%VACP%'
AND description not ilike '%DCPS%'
AND description not ilike '%CASH%'
AND description not ilike '%IATS PAY%'
AND description not ilike '%MISC PAY%'
AND description not ilike '%NJ SDU%'
AND description not ilike '%TREAS 449%'
AND description not ilike '%SDP%'
AND description not ilike '%CHILD%'
AND description not ilike '%FAIRFAX%'
AND description not ilike '%GOVERNMENT SOLUTIONS%'
AND description not ilike '%GOVERNMENT SERVICES%'
AND description not ilike '%GOVERNMENT VI%'
AND description not ilike '%COUNTY%'
AND description not ilike '%EITX%'
AND description not ilike '%CITY%'
AND description not ilike '%ASI GOV%'
AND description not ilike '%STUDENT LN%'
AND description not ilike '%STATE%'
AND description not ilike '%POLICE%'
AND description not ilike '%NY %'
AND description not ilike '%OHIO%'
AND description not ilike '%AR.GOV%'
AND description not ilike '%NJMONT%'
AND description not ilike '%EDUCATION%'
AND description not ilike '%KANSAS%'
AND description not ilike '%NEWYORK%'
AND description not ilike '%SSA TREAS 310%'
AND description not ilike '%SBAD TREAS 310%'
AND description not ilike '%RRB TREAS 310%'
AND description not ilike '%RRB TREAS 310%'
AND description not ilike '%DOEP TREAS%'
AND description not ilike '%DFEC TREAS 310%'
AND primary_merchant_name not ilike '%COUNTY%'

```

```

AND primary_merchant_name not ilike '%LOUISIANA%'
AND primary_merchant_name not ilike '%ACCO BRANDS%'
AND primary_merchant_name not ilike '%GOVERNMENT SOLUTIONS%'
AND primary_merchant_name not ilike '%ASCENSUS TRUST%'
AND primary_merchant_name not ilike '%US NAVY NSA PC MORALE WELFARE & RECREATION%'
AND primary_merchant_name not ilike '%SOCIAL SEC%'
AND primary_merchant_name not ilike '%GOVERNMENT SERVICES%'
AND primary_merchant_name not ilike '%FEDERAL RESERVE%') THEN 'federal'
WHEN ((upper(primary_merchant_name) like '%STATE TREASUR%' or
upper(primary_merchant_name) like '%STATE COMPTROL%' or
upper(primary_merchant_name) like '%STATE CONTROLLER%' or
upper(primary_merchant_name) like '%ST OF%' or
upper(primary_merchant_name) like '%STATEOF%' or
upper(primary_merchant_name) like '%STATE OF%' or
upper(primary_merchant_name) like '%COMMONWEALTH OF%' or
upper(primary_merchant_name) like '%DEPARTMENT OF%' or
upper(primary_merchant_name) like '%STATE DEPARTMENT%' or
(upper(primary_merchant_name) like '%STATE OF COLORADO' and
description not ilike '%COLORADO STATE U%') or
(upper(primary_merchant_name) like '%STATE OF ILLINOIS' and description ilike '%
upper(primary_merchant_name) like '%LOUISIANA GOVERNMENT%')
and description not ilike '%TAX%'
and description not ilike '%UI%'
and description not ilike '%UNEMP%'
and description not ilike '%DSS%'
and description not ilike '%REFUND%'
and description not ilike '%BENEFIT%'
and description not ilike '%CHILD%'
and description not ilike '%EITX%'
and description not ilike '%SUPP%'
and upper(primary_merchant_name) not like '%U.S. DEPARTMENT OF THE TREASURY%'
and upper(primary_merchant_name) not like '%US TREASURY%'
and upper(primary_merchant_name) not like '%US DEPARTMENT OF EDUCATION%'
and upper(primary_merchant_name) not like '%U.S. DEPARTMENT OF HEALTH AND HUMAN SE
and upper(primary_merchant_name) not like '%DEPARTMENT OF VETERAN%'
and upper(primary_merchant_name) not like '%UNITED STATES%'
and upper(primary_merchant_name) not like '%STAR OF%'
and upper(primary_merchant_name) not like '%TAX%'
and upper(primary_merchant_name) not like '%BLUE CROSS%'
and upper(primary_merchant_name) not like '%POWER%'
and upper(primary_merchant_name) not like '%HEALTH%'
and upper(primary_merchant_name) not like '%MEDIC%'
and upper(primary_merchant_name) not like '%UNIV%'
and upper(primary_merchant_name) not like '%ELECTRIC%'
and upper(primary_merchant_name) not like '%CORP%'
and upper(primary_merchant_name) not like '%AIRLINE%'
and upper(primary_merchant_name) not like '%PACIFIC%'
and upper(primary_merchant_name) not like '%TOOL%'
and upper(primary_merchant_name) not like '%CLEANER%'
and upper(primary_merchant_name) not like '%DINER%'
and upper(primary_merchant_name) not like '%EDISON%'
and upper(primary_merchant_name) not like '%EXPRESS%'
and upper(primary_merchant_name) not like '%SOUTHERN%'

```

```

and upper(primary_merchant_name) not like '%LOTTERY%'
and upper(primary_merchant_name) not like '%PRIME%'
and upper(primary_merchant_name) not like '%VISION%'
and upper(primary_merchant_name) not like '%EMC%'
and upper(primary_merchant_name) not like '%CENTRAL%'
and upper(primary_merchant_name) not like '%LIFE%'
and upper(primary_merchant_name) not like '%INSUR%'
and upper(primary_merchant_name) not like '%BERGEN%'
and upper(primary_merchant_name) not like '%ROADHOUSE%'
and upper(primary_merchant_name) not like '%CHILD%'
and upper(primary_merchant_name) not like '%TECH%'
and upper(primary_merchant_name) not like '%SPCA%'
and upper(primary_merchant_name) not like '%TOYOTA%'
and upper(primary_merchant_name) not like '%TIMES%'
and upper(primary_merchant_name) not like '%ZOO%'
and upper(primary_merchant_name) not like '%CENTER%'
and upper(primary_merchant_name) not like '%MADE%'
and upper(primary_merchant_name) not like '%DENT%'
and upper(primary_merchant_name) not like '%CNG%'
and upper(primary_merchant_name) not like '%SOURCE%'
and upper(primary_merchant_name) not like '%HOTEL%'
and upper(primary_merchant_name) not like '%RAILR%'
and upper(primary_merchant_name) not like '%FRESH%'
and upper(primary_merchant_name) not like '%YORKER%'
and upper(primary_merchant_name) not like '%THEATRE%'
and upper(primary_merchant_name) not like '%GRILL%'
and upper(primary_merchant_name) not like '%GENESEE%'
and upper(primary_merchant_name) not like '%FURNITURE%'
and upper(primary_merchant_name) not like '%EAST%'
and upper(primary_merchant_name) not like '%COLLEGE%'
and upper(primary_merchant_name) not like '%GAS%'
and upper(primary_merchant_name) not like '%UTILIT%'
and upper(primary_merchant_name) not like '%COFFEE%'
and upper(primary_merchant_name) not like '%HOSPITAL%'
and upper(primary_merchant_name) not like '%RETIR%'
and upper(primary_merchant_name) not like '%REVENUE%') THEN 'state'
WHEN ((upper(primary_merchant_name) like '%COUNTY OF%' or
      upper(primary_merchant_name) like '%COUNTY%' or
      upper(primary_merchant_name) like '%CITY OF%' or
      upper(primary_merchant_name) like '%DEPARTMENT OF%' or
      upper(primary_merchant_name) like '%CITY DEPARTMENT%' or
      upper(primary_merchant_name) like 'PUBLIC SCHOOLS')
and description not ilike '%TAX%'
and description not ilike '%UI%'
and description not ilike '%DSS%'
and description not ilike '%UNEMP%'
and description not ilike '%REFUND%'
and description not ilike '%BENEFIT%'
and description not ilike '%CHILD%'
and upper(primary_merchant_name) not like '%U.S. DEPARTMENT OF THE TREASURY%'
and upper(primary_merchant_name) not like '%US DEPARTMENT OF EDUCATION%'
and upper(primary_merchant_name) not like '%U.S. DEPARTMENT OF HEALTH AND HUMAN SER
and upper(primary_merchant_name) not like '%US TREASURY%'

```

```

        and upper(primary_merchant_name) not like '%DEPARTMENT OF VETERAN AFFAIRS%'
        and upper(primary_merchant_name) not like '%DEPARTMENT OF VETERANS AFFAIRS%'
        and upper(primary_merchant_name) not like '%ELECTRIC%'
        and upper(primary_merchant_name) not like '%GAS%'
        and upper(primary_merchant_name) not like '%UTILIT%'
        and upper(primary_merchant_name) not like '%Prince William County Service Authority%'
        and upper(primary_merchant_name) not like '%UNIVERSITY%'
        and upper(primary_merchant_name) not like '%HOSPITAL%'
        and upper(primary_merchant_name) not like '%RETIR%'
        and upper(primary_merchant_name) not like '%REVENUE%') THEN 'local'
    ELSE 'other' END as fed
from temp_132.filter1
where transaction_base_type = 'credit'
    and transaction_category_name = 'Salary/Regular Income'
    and amount > 500)
where fed not like 'other')

-- Raw_Payroll (export to R for cleaning)
select a.*, b.total_income
from temp_132.payroll a
    inner join (select sum(amount) as total_income, unique_mem_id
from temp_132.onepercsample
where transaction_base_type = 'credit'
    and transaction_category_name in
        ('Interest Income', 'Other Income', 'Salary/Regular Income', 'Sales/Services')
    and is_duplicate = 0
group by unique_mem_id) b
on a.unique_mem_id = b.unique_mem_id

-- After running script in R, build up final list of individuals
-- Run payroll_ids.sql to create sql of ids. You may need to drop the table
-- drop table temp_132.payroll_ids
create table temp_132.payroll_ids (
    unique_mem_id numeric(30),
    fed varchar(10),
    elig varchar(7)
);

```