Personal Income Forecasting Methods

Real Time Estimates of Changes in Wage & Salary Disbursements

Guillermo Roditi Dominguez, Managing Director
guillermo@newriverinvestments.com

Overview

On April 30th, 2020, the US Department of Labor reported 3.84 million initial claims for unemployment and 19.5 million continuing unemployment claims for the week ending Friday April 24, bringing the total of new claims during the period starting March 2020 to 30.6 million claims. For the month of March 2020, the Bureau of Economic Analysis reported a monthly sequential 2% decline in Personal Income and 7.55% decline in Personal Spending on a seasonally adjusted basis. The Wages and Salaries component declined by 3.1%, and proprietor's income, an income measure of sole-proprietorships excluding dividend and interest for non-financial businesses, and rental income for non-real-estate businesses, declined by 8.2%. Combined, these two measures accounted for approximately 59% of personal income during the months preceding the economic shock and largely represent the share of spendable income earned through participation in economic activity. The remainder is income earned through ownership of financial assets, transfer payments (e.g. social security, unemployment benefits, and Medicare and Medicaid), and supplements to wages and salaries like pension contributions, employer-paid payroll taxes, and other non-salary remuneration like health insurance.

The lag of the Bureau of Economic Analysis personal income is based on the Bureau of Labor Statistics Current Employment Survey, which references the pay period that includes the 12th calendar day of the month, at the end of the month following the reference month. In times where the economy is growing in nominal terms (first derivative is positive, second zero) the sample period can be expected to be marginally lower than the end of the month. Likewise, when the economy is shrinking in nominal terms (first derivative negative, second zero) the sample period can be expected to be marginally higher than the end of the month. In these cases, a simple period-over-period adjustment (yearly or monthly depending on application of seasonal adjustment) is sufficient to identify the trend and extrapolate it to the present-time. In cases where the second derivative (acceleration, deceleration) this measure will prove to be less precise, the further away from zero the second derivative is, the higher the error.

In the case of wages and salaries and proprietor's income forgone as a result of the work-closures resulting from mandatory and voluntary work-stoppages during the period beginning mid-March, the BEA has opted to adjust estimates of March wages and salaries from the Current Employment Survey using the Department of Labor's higher-frequency reports of initial claims for unemployment insurance with claims data for the weeks ending March 21, March 28, and April 4[1]. Additionally, the BEA opted to adjust estimates using a private estimate of workers they expect to have become unemployed but who were unable to submit a claim for unemployment insurance.

Over the last 8 years, I have supplemented this information with a real-time estimate derived from the Daily Treasury Statement[2], which reports, with a lag of one business day, deposits made to US Treasury accounts for income and employment taxes withheld (payroll taxes). Withheld income and employment taxes primarily represent taxes withheld for income taxes, Medicare, Medicaid and Social Security, the latter 3 of which are shared equally by employers and employees. Withheld income and employment taxes *does* not include payments made by businesses under the Federal Unemployment Tax Act. The monthly sum of taxes withheld can be seasonally adjusted and then compared to the BEA's measure of wages and salaries to derive an effective total tax rate. Likewise, the monthly sum of taxes withheld can be adjusted by subtracting the BEA's estimate of employer contributions for government social insurance (multiplied by 2) to estimate what portion of taxes withheld were income taxes. This estimate of income taxes can then be used to estimate the effective income tax rate. Combined, these measures can add valuable information that allows us to better understand the composition of increases and decreases in income. For example, a rising effective tax rate in the context of employment losses would suggest a disproportionate impact on populations with lower effective tax rates, most-likely low-income households[3]. During periods of relatively stable effective tax rates can be used to generate relatively accurate advanced estimates of wages and salaries disbursed. Because the treasury data is not a sample, there is no sample errors; however, due to the variations in when employers are required to make deposits relative to when payments to employees are made and differences of when employers pay employees relative to their work period, this measure will lack the accuracy of a survey measure when attempting to measure income at a specific point in time.

As part of the March 27th, *2020 Coronavirus Aid, Relief, and Economic Security Act (CARES) Act*, employers can—beginning with the pay period ending March 27th—defer employer contributions of payroll taxes until December 21 with certain exceptions related to recipients of Payroll Protection Program loans. As a result, the information from the Daily Treasury Statement can no longer be compared to pre-CARES Act period for forecasting purposes. This has led to my decision to publish this short note where I explain the methods used in the latest version of this model as well as a look at how our modeled estimates of personal income have progressed year to date. Finally, you will be able to find some of the unprocessed and processed data as well as a simplified version of the model's source code in a public Github repository. It is my expectation that some of the readers with a greater knowledge of statistical analysis will be able to improve on it or identify egregious abuses of multiple regression.

[1] How did BEA adjust March 2020 wages and salaries to account for the effects of COVID-19? https://www.bea.gov/help/faq/1411

[2] Daily Treasury Statement https://www.fiscal.treasury.gov/reports-statements/dts/

[3] This would be due to the positively sloped income tax curve.

Please feel free to let me know how stupid I am but do know that I am unlikely to produce a response more thoughtful than “¯_(ツ)_/¯”.

Daily Treasury Statement

The data and what they represent

The Daily Treasury Statement can be found online in excel, text, and PDF form. The line item referenced is “Withheld Income and Employment Taxes” and can be found in Section IV (line 194 on the excel file). Readers with Bloomberg access will be able to find it under “USCBFTWI Index” on the terminal. The Treasury updates the data at 4PM Eastern time every day that banks are open, Bloomberg typically updates a few minutes after that[4]. The unit used in the release is millions of dollars. Deposits are made through the Electronic Federal Tax Payment System (EFTPS), and a deposit must be scheduled prior to 8PM Eastern in order to be counted.

The Deposit Schedules

The date an employer’s payment must arrive by is governed by Publication 15[5] for non-agricultural employers and Publication 51[6] for agricultural employers. Generally, deposits are due on business days only and payments due on weekends or federal holidays can be paid on the first business day following the holiday. There is never a period longer than 3 consecutive calendars days without a business days unless a presidential proclamation (typically for the funerals of former presidents) extends an existing 3-day period into four days or that bridges a single day gap between a holiday and a weekend. There are various possible schedules for payments but only 3 that are relevant to this exercise:

- Monthly Schedule: Employers reporting less than $50,000 in taxes per quarter must deposit taxes once a month on the 15th calendar day of the month after a pay date
- Semiweekly Schedule: Employers with payrolls that report more than $50,000 in taxes per quarter must pay on a semiweekly basis on Wednesdays and Fridays. Taxes on paydays falling on Wednesday Thursday or Friday are due the Wednesday after the pay date. Taxes on paydays falling Saturday, Sunday, Monday, and Tuesday are due the Friday after the pay period.
- The $100,000 Next-Day rule: For any pay date in which taxes withheld exceed $100,000 the deposit is due the business day after the pay date.

[4] For years I manually updated these every day after a make-shift script broke and I decided it wasn’t that much work anyways. Finally, one day David Taggart of PDMacro.com pointed out to me the series was on Bloomberg, which led to me switching from my relatively short, hand-compiled series to the full data set that Bloomberg had. Which in turn led to me realizing how many dates were missing from the Bloomberg set. After multiple requests all the ones I’ve found have been fixed, but if you use data from before 2010 I wouldn’t expect it to be perfectly complete.

[5] Publication 15 https://www.irs.gov/publications/p15

[6] Publication 51 https://www.irs.gov/publications/p51

For periods where the due date is a holiday, next-day and monthly depositors have an additional day to make payment. For semiweekly depositors, any holiday[7] between the pay date and the due period will extend the due date window by a business day so that no due date is ever less than 3 days after the pay date. These patterns are because of the high degree of volatility in the daily numbers.

Complete Month Data

The easiest adjustment to DTS data is to aggregate it into monthly sums (by calendar sum, ignoring the pay period) and seasonally adjusting it[8]. A simple adjust process is to use the SEATS procedure, part of the X-13ARIMA-SEATS software available from the Census Bureau[9]. As can be seen in in the time series Figure 1 below, this simple adjustment effectively accounts for all the volatility of the estimates resulting from seasonal patterns or calendar effects[10]. The resulting seasonally adjusted monthly deposit rate can then be converted into a seasonally adjusted annual rate (SAAR) by multiplying by 12, which makes it automatically compatible with the personal income release from the BEA. With that we can estimate effective federal income tax rates and monthly changes in before- and after-tax income from wages and salaries, as illustrated in the scatterplot Figure 2 below[11].

Figure 1

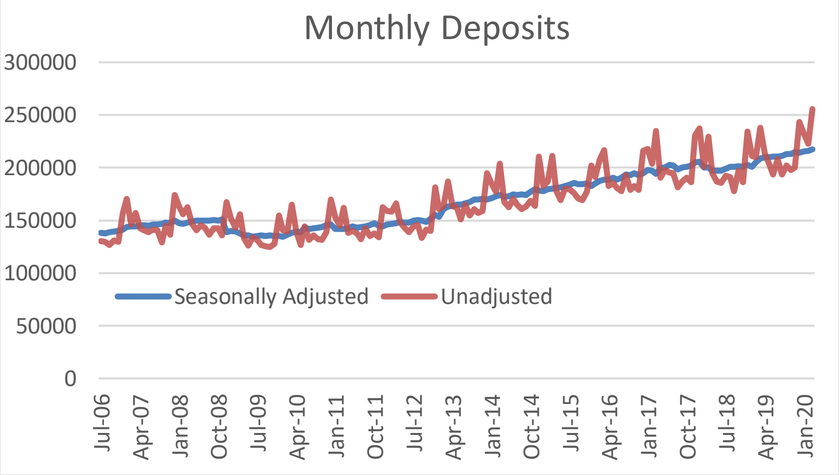
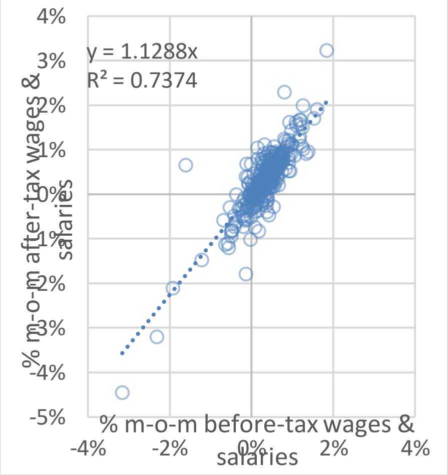


Figure 2



[7] This is the one rule I never got around to programing into the code, mostly because I just never got around to it
[8] Attentive readers will have already identified the problem here: part of the data for the month calendar month ends on the first business day of the month, another on the first semiweekly deposit date on the month, and another on the first business day after the 14th calendar day of the month. It is trivial to make these adjustments if you care about strictly matching the calendar pay periods, but SEATS will work with any time series that is broken-up into monthly periodicities, so for this example monthly deposits are used. In large part because the original scheduled release of this note was May 1 and that afforded me a complete month of April.
[9] X-13ARIMA-SEATS https://www.census.gov/srd/www/x13as/
[10] The resulting model will automatically use a log transformation, adjust for the leap years and major holidays and correct for the number and distribution of weekdays.
[11] The scatter plot includes all January, 1995 to March 2020 data

In addition to seasonally adjusting the data, the X-13 package can also forecast non-seasonally adjusted values for the periods following the input data, with a trend extrapolation for non-stationary series[12]. For large series with little volatility like macroeconomic aggregates, this is a valid approach with acceptable precision. Most of the inputs to the first estimate of GDP are just this.

Higher Frequency Estimates

Building a model

Starting with the known information about day-of-week and day-of-month patterns which were previously discussed we can begin to model the variability of smaller periodicities. The most obvious should be that Fridays and Wednesdays, or the following business day if that day is a holiday, will exhibit higher levels of deposits. Similarly, we can expect the 15th calendar day of a month, or the first business day after the 14th, to see similarly elevated levels. If some employees are always paid at month-end, then we would expect to see a similar pattern emerge there. These patterns present themselves as combinations of employer payroll periodicities and behaviors. Exploring explaining every single pattern that is measurable in the data is beyond the scope, but in the list below I will use the ones that a simple multiple linear regression found to be significant:

Daily and monthly patterns:

- The day of the week using 4 distinct binary values (“dummy variables”). I elected to use Tuesday as a base scenario and add 4 variables for the other days. In the case a day follows a holiday, the flags are raised for the actual weekday and the previous holiday weekday
- The first business day of the month (for payrolls that take place on the last day of the month)
- The last business day of the month
- The second business day of the month (for payrolls that take place on the first business day of the month)
- The eleventh and sixteenth business days of the month to account for biweekly and semimonthly payroll frequencies
- The first Wednesday and first Friday of the month (the first semiweekly due dates in the month)
- The third Monday of the month
- What percentage of the current month’s calendar days are also business days (for payrolls based on monthly periodicities)

Yearly Patterns. These were mostly data-mined after finding abnormal error terms after applying the patterns already mentioned:

- The first day of the year (for payrolls run the last business day of year)

[12] This is just a way of saying “the series has a drift” i.e. it tends to go up (inflation, growth, etc)

- The last day of the year
- March 16th and April 16th (I have no good explanation for these)
- The first business day of December
- The first business day after Christmas (holiday bonus season)
- The 17th and 18th day of April (I suspect these are related to tax filing due dates)

Once patterns are identified they can be used to calculate expected values for any day of the month. To account for seasonal variations and non-stationary data, the data is de-trended into a stationary series before applying seasonal regression by converting a day's dollar amount of deposits into a multiple of the average of the total percentage of deposits for that specific month[13]. Linear regression is applied to a dependent variable of the day's deposits expressed as a multiple of the mean deposits for that month[14] and 17 independent variables, 16 of which are binary values. The resulting summary of the regression for the period starting Feb 1, 2013[15] and ending Dec 31, 2020 can be found in **Exhibit 1** for readers who are interested.

Diagnosing the model

Using the resulting model's output from the previous section we can test to see how well the daily estimates performed and how an extrapolation of the running sum of estimates compared to the actual final monthly total. Figure 3 is a scatterplot of the cumulative month-to-date forecasting error at a point-in-time. The x-axis is the observation number for the month and y-axis the percentage the true value exceeded the estimate by. The grey line represents the mean of the observations and the blue and red lines represent 2 standard deviations (-2, +2 respectively) of the forecasting error for the nth business of the month (81 months). Note that the last is generally meaningless because of smaller number of observations. There was only 5 business days with 23 days in the fitted data. There will be no discussion of hypotheses and tests. If you understand that language, then you can find and use the code. Table 1 is a table of the mean cumulative residual on the last day of each month in the sample[16]. In this case we would be looking for any kind of pattern we can find that clues us is onto any additional

[13] Please note that this introduces "hindsight bias" into the model. We are training the model using information that was not known at that point in time. This is more of a cardinal sin when building models for trading than for non-crucial economic purposes. A better solution, which I personally use, is to use the X-13 package with data up-to and including the last calendar month to output a forecast of non-seasonally adjusted monthly sums and use the current day's deposits as a percentage of the forecasted value. In theory, these approaches are an ocean apart, but in practice, this works well-enough and it greatly simplifies the code for building the model. Astute readers will find this modification almost trivial and, for readers with limited programming experience, this omission will greatly reduce the complexity and increase the readability of the code.

[14] Using a percentage of the month's total resulted in a lot of very small values for the coefficients and error terms which ultimately would have made the result hard to format due to too many digits left of the decimal point. You can yell at me on the internet if you are mad about it, I'm @newriverinvest on twitter.

[15] The Tax Relief, Unemployment Insurance Reauthorization, and Job Creation Act of 2010 included a one-year reduction in the FICA payroll tax that was subsequently extended until 2012. Feb 2013 is the first "clean" month

[16] The code in the repository has various ways to slice/pivot the data as examples of how to look for concentrations of errors, specifically chronic over or underestimations that may be repeating patterns

seasonality the data has. The most obvious one is that for the months of March to December, the mean residual is precisely the same for the years 2013 and 2019[17], suggesting possible multi-year cyclicality.

Figure 3

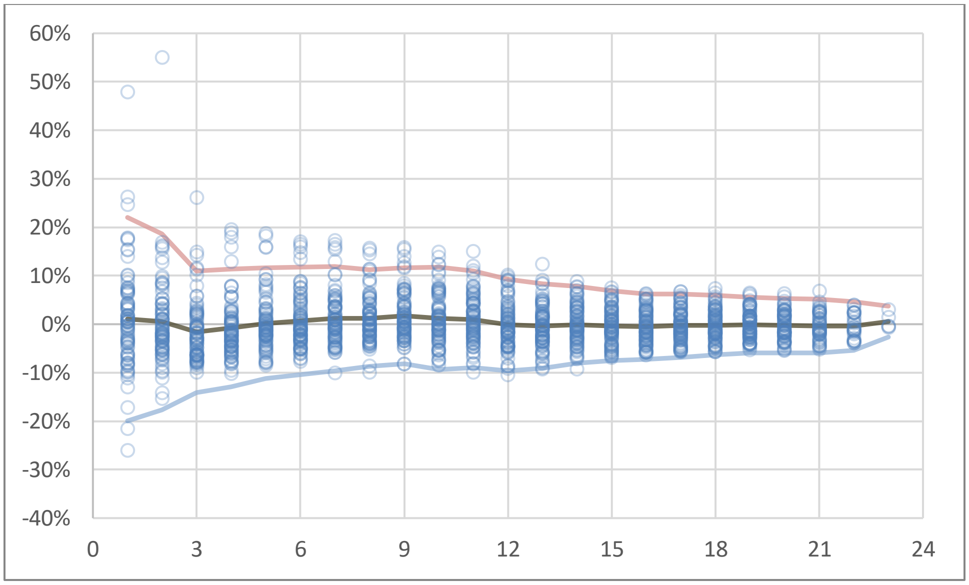


Table 1

	2013	2014	2015	2016	2017	2018	2019
1		-0.849%	1.987%	-1.352%	-5.586%	-4.579%	1.473%
2	-4.156%	-4.089%	-4.089%	-4.721%	-0.073%	-0.073%	-4.089%
3	-2.412%	-6.754%	-1.142%	0.758%	-1.362%	2.021%	-2.412%
4	-3.885%	-0.177%	1.609%	3.273%	3.643%	-3.342%	-3.885%
5	1.256%	4.567%	1.741%	0.837%	-2.304%	3.473%	1.256%
6	1.670%	-1.798%	-2.412%	0.712%	1.149%	-1.090%	1.670%
7	-2.304%	2.970%	0.444%	1.741%	-2.819%	0.231%	-2.304%
8	3.909%	1.545%	-2.797%	-2.962%	3.473%	0.444%	3.909%
9	-1.512%	-1.512%	2.087%	1.932%	-0.768%	1.744%	-1.512%
10	3.473%	1.256%	4.567%	-2.819%	0.997%	-2.304%	3.473%
11	-0.823%	1.394%	-1.606%	2.567%	2.011%	2.404%	-0.823%
12	1.522%	-1.650%	4.795%	5.252%	2.460%	-2.099%	1.522%

[17] If you have not figured it out yet, I will not ruin your fun until the next section. All I will tell you is that all the information you need to figure out why it starts in March and not February is in this document. Go back to the things that seemed like unnecessary detail.

Improving the model

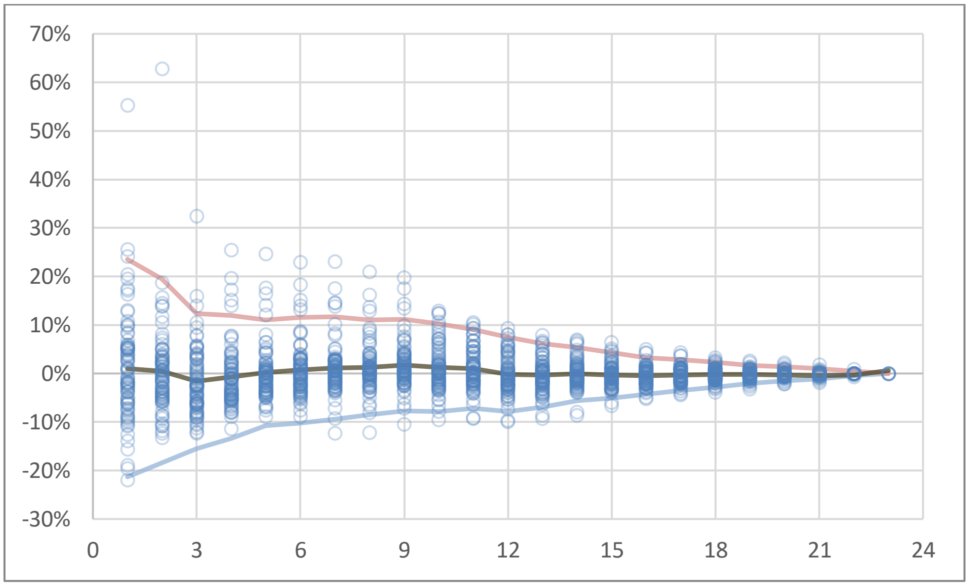
The cumulative errors represented by the data in Table 1 are, of course, nonsense. The reason for the repetition of the precise numbers 6 years apart is because in the 2013 and 2019 calendars the days of the week in which the days of the year fall are the same. The dependent variable of a model represents the relationship between a day's deposits and the average monthly deposits, therefore the cumulative fitted values for any month should equal the business-day count for the month. Because the model is based solely on the calendar, estimates for the daily pattern can be created months or years in advance; the only unknown being a federal holiday by presidential proclamation. Therefore, estimates can be transformed so that the final sum of monthly estimates accurately matches the known value without introducing a hindsight bias to the model. In case it is not crystal clear: the sum of all the monthly fitted values should be the number of business days in a month, the error in Table 1 should be obvious after seeing Table 2.

Table 2

	2013	**2014**	**2015**	**2016**	**2017**	**2018**	**2019**
1		21.17821	18.62246	19.25696	21.1173	21.96152	20.69059
2	17.28009	19.77688	19.77688	20.94421	19.01378	19.01378	19.77688
3	21.50645	22.41835	21.23979	22.82574	23.31335	21.55547	21.50645
4	22.85475	22.03885	21.64592	20.31261	19.27137	21.70181	22.85475
5	21.72362	20.04102	19.65181	20.82433	22.50693	21.23601	21.72362
6	19.66607	21.37764	22.53058	21.84327	21.74727	21.22883	19.66607
7	22.50693	21.34667	22.89789	19.65181	20.56372	20.95146	22.50693
8	21.14	20.67552	21.58742	23.6812	21.23601	22.89789	21.14
9	20.30249	20.30249	20.56174	20.59433	20.15368	18.66871	20.30249
10	21.23601	21.72362	20.04102	20.56372	19.80062	22.50693	21.23601
11	19.15633	17.74914	19.30513	19.48659	18.61783	19.51918	19.15633
12	20.68044	22.36304	19.99313	19.89713	19.50792	20.41983	20.68044

Depending on the amount of training data you have, you could try treating day count as a factor instead of an integer and add dummy variables for each different day count (18-23) or, more sanely, simply scale the daily estimates so that the monthly fitted values sum up to the day count. Figure 4 illustrates how the monthly estimates progress after scaling is applied.

Figure 4



Using the model on new data

Using the coefficients from the model it is now possible to use the model to attempt to forecast data for months after the period the data was trained on. The daily model will extrapolate the deposits received month-to-date in the context of their timing patterns and output an estimate of the final sum of deposits for the month. That estimate can be compared to the SEATS forecast to look for clues of acceleration or deceleration. As the month progresses, the estimate will improve, and any excess or shortfall will become more meaningful information, especially any breach of the confidence interval bounds from the SEATS forecast. At this point, the current estimate of the monthly value can be used to produce seasonally adjusted data using the X-13 package, a technique known as concurrent adjustment[18]. Note that this could[19] have the undesirable effect of affecting the modeled seasonality and produce revisions of the more recent data. This is an unavoidable trade-off of this method of seasonality adjustment, but it fortunately reproduces the behavior in the data we are attempting to estimate. For an example of the real-time progression of the SAAR estimate for April 2020, see Figure 5.

For real-time observation for the purpose of identifying deviations from established trends, the daily deposits can be compared to an expected deposit schedule derived from the SEATS forecasts. The application is straight forward, the daily estimated seasonal pattern is applied to the monthly SEATS forecast to convert it into a daily value and compared to the actual daily value. The actual daily deposit minus the forecast daily deposit is the error, and a running sum of the error will illustrate the present data's deviation from the previously forecasted trend. Figure 6 is an illustration on the daily deposit

[18] It's not optimal to do this, but the if it is good enough for the BLS, it is good enough for me! https://www.bls.gov/cps/seasonal-adjustment-methodology.htm

[19] For a single data point in this specific series, it will absolutely not matter and SEATS will not dilute the effect of any changes or lead to a measurable revision, but generally speaking it is not a best practice

schedule from January 2, 2020 to April 30, 2020. A $45 billion (numbers are in millions) shortfall accumulated in the 27 business days from March 25 to April 30 would, using the January and February effective tax rate of 27.28%[20], correspond to a $165 billion reduction in the wages and salaries paid (NSA) and $167 billion on a seasonally adjusted monthly basis[21]. Of course, this is an upper limit on the losses due to the option for employers to defer payment of certain payroll taxes.

Figure 5

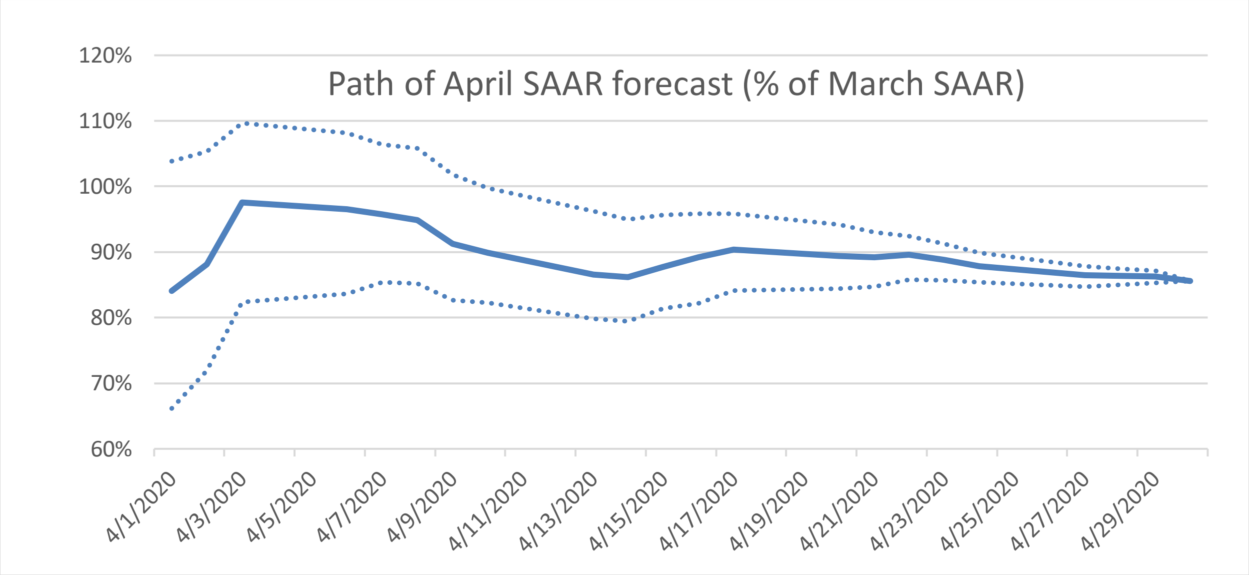
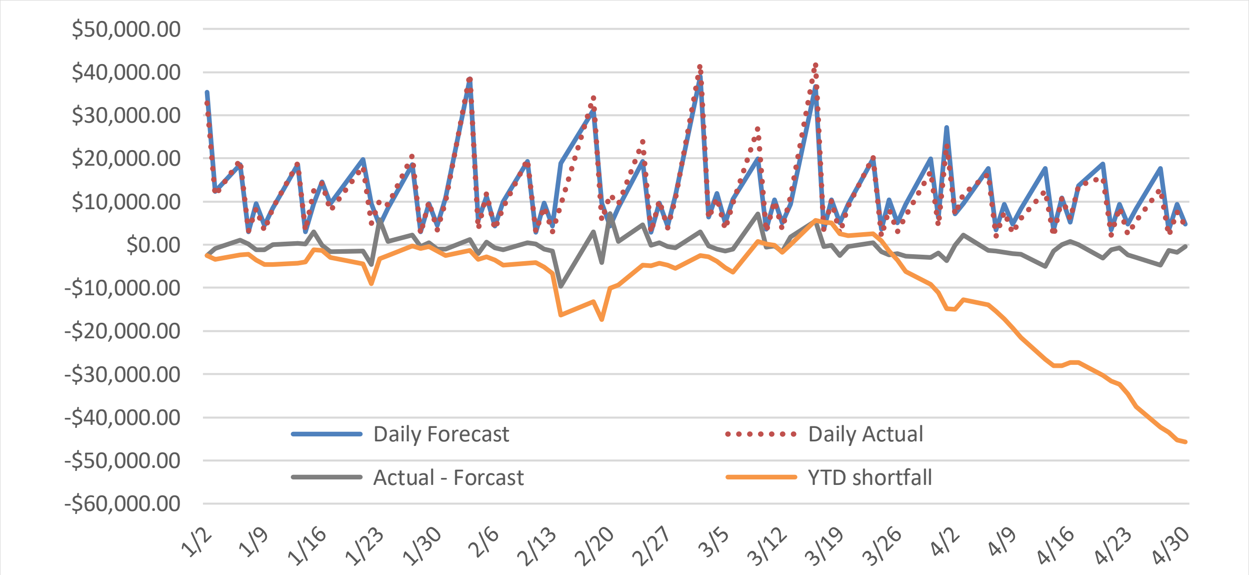


Figure 6



[20] See the last paragraph in page 2

[21] Using a total adjustment factor of 0.98826

Conclusion

I hope this has been a helpful overview of how to create working real-time advance forecasts of macroeconomic aggregates during macroeconomic shocks. For more resources please see **Appendix A**. The source code written in generating this exercise is included in the repository but do understand that it was written to be a “quick hack” and not production-quality work. If you intend to re-use the model, you should be proficient in the tools used. While I might be able to incorporate improvements suggested, I will not be able to answer programing-related questions due to time constraints. If you do re-use it, I would appreciate being credited for the work I have done. Lastly, this was a labor of boredom and an attempt to show younger people in the field of economics with less experience how to *do* economics in colloquial language starting with data methodology understanding and through methods of statistical analysis, not a tutorial for recreating this model. It is my hope that people newer to the field will remember the intuition and steps more than the actual model accompanying this write up. It was deliberately not meant to be written in academic style or full of potentially intimidating equations or code because the target audience is not the people who seek those qualities. Personally, I think this would make a suitable starting point for a group project in an undergraduate-level statistics or econometrics class and I wish I had done something similar when I was a student. If you are a graduate student teaching a class in econometrics and want to do that, please do, I will help to the extent that I can. For anyone else that was hoping for a free model to use for trading or investing purposes: you get what you pay for.

XOXO,

Guillermo Roditi Dominguez
guillermo@newriverinvestments.com

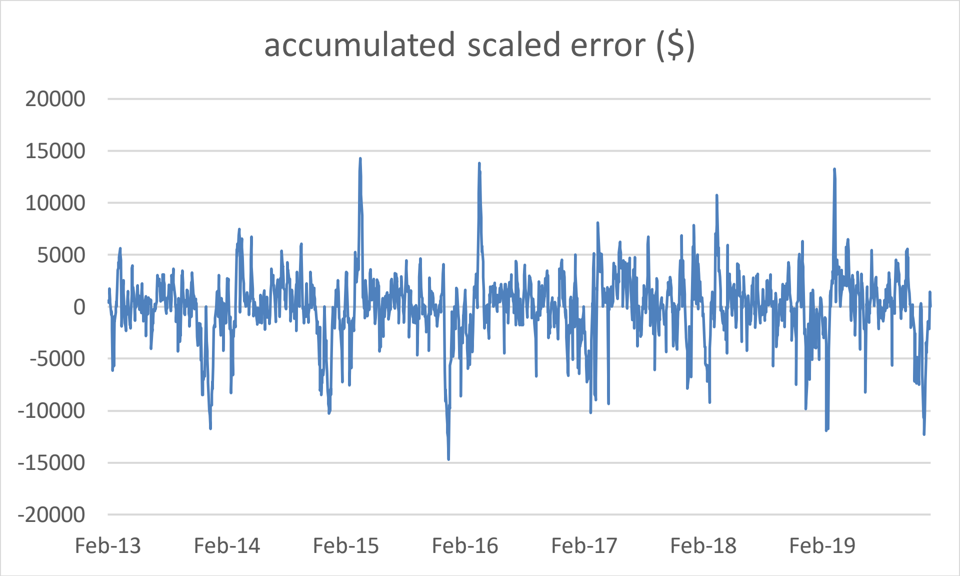
Exhibit 1

```
Call:
lm(formula = pct.deposit ~ is.monday + is.wednesday + is.thursday +
    is.friday + is.sixteen + is.eleventh + is.second + is.month.start +
    is.month.end + is.year.start + is.year.end + is.march.sixteen +
    is.april.sixteen + is.tax.season + is.post.thanksgiving +
    is.xmas + business.day.pct + is.third.monday + is.first.wednesday +
    is.first.friday, data = pre_covid)

Residuals:
     Min       1Q   Median       3Q      Max
-1.44611 -0.10859 -0.01917  0.10440  1.14340

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -0.51965    0.11120  -4.673 3.20e-06 ***
is.monday             1.39956    0.01768  79.160  < 2e-16 ***
is.wednesday          0.58361    0.01772  32.930  < 2e-16 ***
is.thursday           0.12860    0.01625   7.913 4.47e-15 ***
is.friday             0.48761    0.01788  27.269  < 2e-16 ***
is.sixteen            0.76310    0.02801  27.245  < 2e-16 ***
is.eleventh           0.14453    0.02769   5.219 2.01e-07 ***
is.second             0.23786    0.02549   9.330  < 2e-16 ***
is.month.start        1.62589    0.02751  59.108  < 2e-16 ***
is.month.end          0.22381    0.02608   8.581  < 2e-16 ***
is.year.start         0.44880    0.09331   4.810 1.64e-06 ***
is.year.end           0.49518    0.08609   5.752 1.04e-08 ***
is.march.sixteen      0.41546    0.07263   5.720 1.25e-08 ***
is.april.sixteen     -0.71888    0.07768  -9.254  < 2e-16 ***
is.tax.season         0.52151    0.06445   8.092 1.10e-15 ***
is.post.thanksgiving -0.90743    0.08637 -10.506  < 2e-16 ***
is.xmas               0.26875    0.08259   3.254 0.001160 **
business.day.pct      1.16701    0.16010   7.289 4.74e-13 ***
is.third.monday       0.10520    0.02796   3.762 0.000174 ***
is.first.wednesday    0.12717    0.02804   4.535 6.15e-06 ***
is.first.friday       0.11037    0.02800   3.942 8.40e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2177 on 1708 degrees of freedom
Multiple R-squared:  0.9103,	Adjusted R-squared:  0.9093
F-statistic: 866.9 on 20 and 1708 DF,  p-value: < 2.2e-16
```

Appendix A

Residual seasonality

The model as currently presented is a relatively accurate predictor, but there remains some stubborn residual seasonality that was not accurately captured with only 7 years for training. Specifically, there is a recurring underestimation overestimation in the middle of February, an overestimation around the third week of March, and a chronic overestimation in the second or third week of December. Fixing this is left as an exercise for the reader. I am tired of this project



Charts and data sources

All the visual aids in this document come from the “word-file-supports.xlsx” file which can be found in the “etc” folder of the GitHub repository

GitHub Repository

I just want to preface this by saying that while I really do appreciate your interest and that you are welcome to fork and edit the code as you please, I cannot possibly with my other obligations maintain this codebase in the future. If you have improvements you want to push, I will do my best to tend to that, but please do not be mad at me if I neglect this, I have a lot of stuff on my plate.

The repository: https://github.com/groditi/DTS-withholding