March Forecast on Employment and Average Weekly Earnings by Industry within Tampa-St. Petersburg-Clearwater MSA

Submitted to
Dr. Jim Dewey
Florida Polytechnic University

Submitted by
Luiz Gustavo Fagundes Malpele
Department of Data Science
Florida Polytechnic University

Abstract: Florida's state government sought to obtain an empirical forecast on employment and earnings by industry within the region of Tampa-St. Petersburg-Clearwater MSA for the purpose of state level law enforcement budgeting and planning. In order to accomplish this, the employment and earnings were predicted by utilizing past unemployment rate, specific industrial sectors' data, and Florida's employment as factors. Then, different models were generated utilizing economic factors, and the best one was selected by applying validation techniques, such as the rolling window approach.

1. Introduction

The Florida state government desires to forecast two different subjects for state law enforcement budge and planning, which are average weekly earnings of all employees and total private employees in Tampa-St. Petersburg-Clearwater MSA. By preemptively estimating what will be standards of employment and earnings, it is possible to allocate resources to solve the problems at its source, this study will focus on the labor market and average weekly earnings before the Covid-19 pandemic gains strength and affects the American Labor Market.

For such prediction to be made, the best models were developed using the historical data, which included data starting on 1944 for state-wide predictors such as Florida's non-farm employees, and on 2007 for the most recent ones which are the subjects of the forecast: Average weekly hours and earnings, and average hourly earnings for Tampa's MSA.

2.1. Data

The data utilized in this project came from the FRED, the Federal Reserve of Economic Data, which provided data consisting of thousands of economic data time series from scores of national, international, public, and private sources. It includes the four variables which are the subjects of the forecast, are *tpa_aveweek_earn* and *tpa_aveweek_hour*, which stands for Average weekly hours and earnings, *tpa_avehour_earn* standing for Average hourly eanings, and *tpa_priv*, the total private employees in Tampa's MSA.

Furthermore, the data also includes extra variables which might be useful as predictors for the model, which are *tpa_serv, tpa_tech, tpa_bp, tpa_nonfarm, tpa_unemp, and fl_nonfarm*. More information regarding each variable can be referenced on **Table 2,** and **Appendix C** contains the data.

It is important to note that the data gathered from this source was wrangled and cleaned to be able to properly format it and investigate the best regression models. The data starts on different periods for each of variables, and the most recent ones start on January of 2007, which should not be a problem for a one-period ahead forecast. Lastly, there are three noticeable events on 2001, 2009 which correspond to the Dot-Com bubble and Global Financial Crisis, and the Covid-19 crisis which is in the beginning and gaining strength when this forecast is being launched, so there is currently a high presence of volatility on the industry.

Tables 1: Summary of the variables (Mean, Standard deviation, Minimum, and Maximum)

Variable	Obs	Mean	Std. Dev.	Min	Max
tpa_unemp	361	5.487812	2.237848	2.6	11.7
daten	974	7137.268	8562.49	-7670	21946
tpa_nonfarm	601	896.9854	313.706	311.2	1412.4
tpa_bp	386	1368.969	646.2005	279	3441
tpa_tech	277	83.46643	15.96183	55.5	123.2
tpa_serv	361	979.3163	143.2321	692.5	1258.8
tpa_aveweek_earn	157	810.5957	60.59111	711.83	946.21
tpa_avehour_earn	157	23.40153	1.869038	20.28	27.61
tpa_aveweek_hour	157	34.65478	.4744019	33.1	37
tpa_priv	361	975.6399	135.7733	711	1251.6
fl_nonfarm	973	3918.434	2788.313	359	9128.3
datec	974	7137.268	8562.49	-7670	21946
date	974	234.5	281.3139	-252	721
month	974	6.489733	3.457776	1	12
lnfl_nonfarm	973	7.905891	.9549767	5.883322	9.119135
lntpa_priv	361	6.872934	.1447663	6.566672	7.132178
lntpa_aveweek_hour	157	3.545343	.0136399	3.499533	3.610918
lntpa_avehour_earn	157	3.149772	.0772252	3.009635	3.318178
lntpa_aveweek_earn	157	6.695115	.0722956	6.567839	6.852465
lntpa_serv	361	6.875543	.1530702	6.540308	7.137914
lntpa_tech	277	4.406582	.1887783	4.016383	4.813809
lntpa_bp	386	7.105117	.5000402	5.631212	8.143517
lntpa_nonfarm	601	6.722691	.4153847	5.740436	7.253046
lntpa_unemp	361	1.62951	.3736546	.9555115	2.459589
tpa_totalweek_earn	157	872433.8	137703.8	711021.4	1184276
lntpa_total_earn	157	13.66744	.1504448	13.47446	13.98464

The number of observations for each variable is different because some the data started to be collected on different periods. It is important to point out some important variable that will direct impact the modelling process such as unemployment rate, it’s standard deviation is 2.23 and mean is 5.48%, which indicates that there is low unemployment and that most of the time this number usually does not drastically changes due since the standard deviation is 2.23, but the problem is to predict this factor amid a crisis such as the sub-prime crisis in 2009.

Table 2: Description of all variable (Storage type, Display format, Variable label)

Variable name	Storage Type	Display Format	Value Label
datestring	str10	%10s	fed string date
tpa_unemp	double	%10.0g	Unemployment within Tampa MSA
daten float		%td	numeric (daily) date
tpa_nonfarm	double	%10.0g	Total Nonfarm Employees in Tampa MSA
tpa_bp	double	%10.0g	New Private Housing Authorized by Building Permits - Tampa MSA
tpa_tech	double	%10.0g	Professional, Technical, and Scientific Employees - Tampa MSA
tpa_serv	double	%10.0g	Service-Providing Employees - Tampa MSA
tpa_aveweek_e~n	double	%10.0g	Average Weekly Earnings - Tampa MSA
tpa_avehour_e~n	double	%10.0g	Average Hourly Earnings - Tampa MSA
tpa_aveweek_h~r	double	%10.0g	Average Weekly Hours - Tampa MSA
tpa_priv	double	%10.0g	All Employees: Total Private in Tampa-St. Petersburg-Clearwater,
fl_nonfarm	double	%10.0g	All Employees: Total Nonfarm in Florida
datec float	float	%9.0g	
date	float	%tm	
month	float	%9.0g	
lnfl_nonfarm	float	%9.0g	
lntpa_priv	float	%9.0g	
lntpa_aveweek~r	float	%9.0g	
lntpa_avehour~n	float	%9.0g	
lntpa_aveweek~n	float	%9.0g	
lntpa_serv	float	%9.0g	
lntpa_tech	float	%9.0g	
lntpa_bp	float	%9.0g	
lntpa_nonfarm	float	%9.0g	
lntpa_unemp	float	%9.0g	
tpa_totalweek~n	float	%9.0g	Total Weekly Earnings (thousands) - Tampa MSA
lntpa_totalwe~n	float	%9.0g	Log of Total Weekly Earnings - Tampa MSA

This table includes information regarding the storage type of each variable (str10, float, and double), the display format for each of the variables, and the variable label, which is the variable name on the Excel file before importing it to Stata.

2.2. Total Private Employees Compared to other Explanatory variables

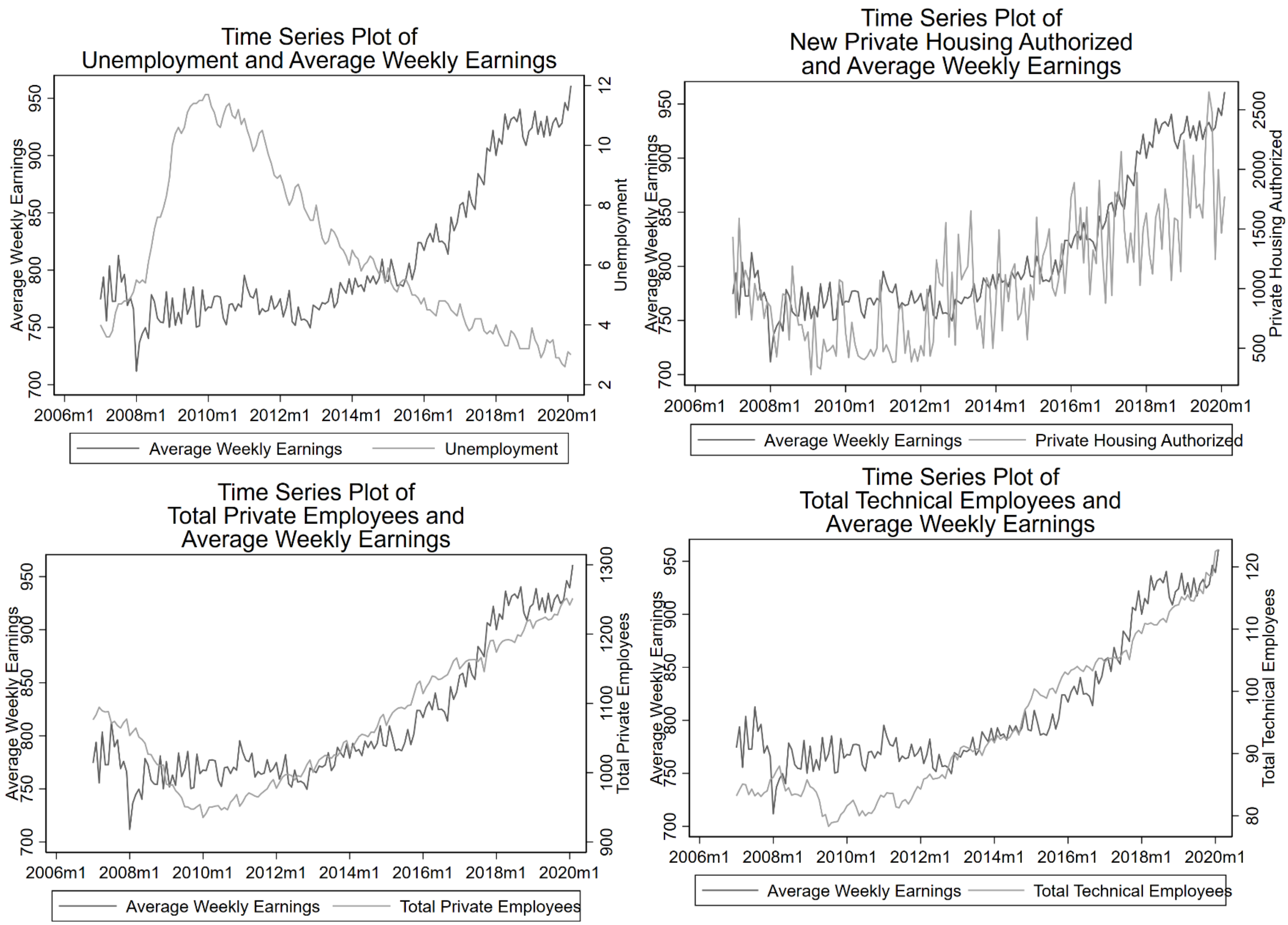
Figure 3: Two-way time series plot of Total Private Employees and Explanatory variables

The Total Private Employees variable will be subject to forecast on the next steps, so it is important to understand how the time series curve of the response variable compares to other explanatory variables. The chosen variables that are going to be subject to the forecast were represent in the two-way time series plots above, and they are: Service-Providing Employees, Total Florida Nonfarm Workers, Total Technical Employees, and Unemployment.

On the unemployment, technical, and total private employees it is noticeable the effect of the 2001 crisis knows as the dot-com bubble. Furthermore, the most noticeable impact on employment happened on 2009 due to the sub prime crisis, and the curves of the explanatory variables have a similar fluctuation behavior when compared to total private employment, this will be.

2.3. Average Weekly Earnings compared to other explanatory variables

Figure 4: Two-way time series plot of Total Private Employees and Explanatory variables

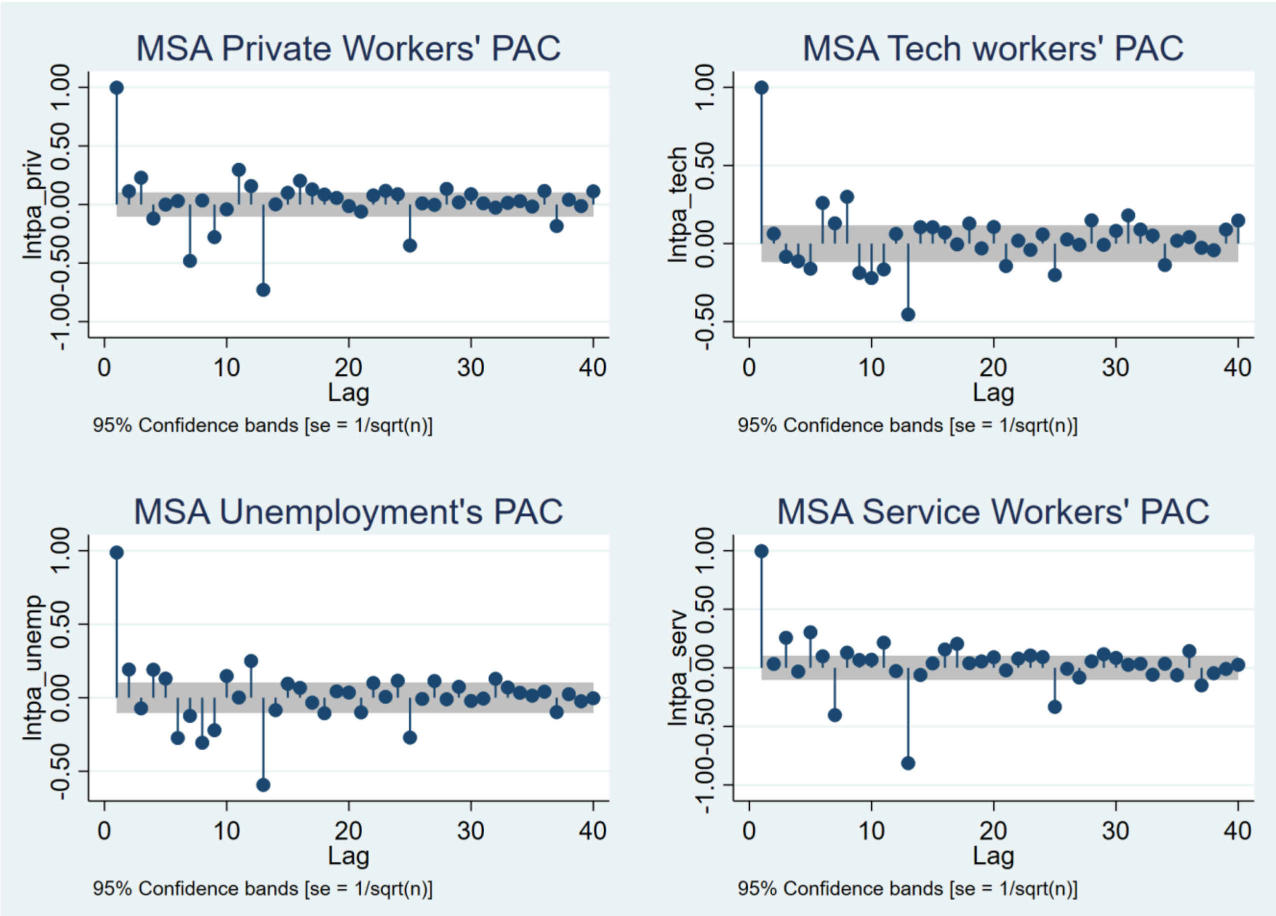


In order to forecast Average Weekly Earnings, four different variables were chosen: Total Private Employees, Total Technical Employees, New Private Housing Authorized, and Unemployment within the MSA. It is noticeable that the Average Weekly Earnings curve presents a more unpredictable noise structure when compared to private employment as its present clear annual cycles.

There is only one significant drop on the average weekly earnings curve, which is on January of 2008, and a significant growth starting on January 2016, also noticeable on other employment variables. Additionally, the four explanatory variables show a rapid increase on the past years happening on the MSA region, there is low-unemployment, intense growth in the technological, and labor market within the region, which could also directly reflect on the average weekly earnings of workers.

3.1 Variables' Partial Auto Correlogram

Figure 5: PAC for lntpa_priv and the Explanatory Variables



The partial auto correlogram shows that the four variables above (*lntpa_priv*, *lntpa_tech*, *lntpa_unemp, and lntpa_serv)* present a highly persistent time series, it can be observed that ρ's value is close to 1, so there is strong evidence that these four variables should be differenced. It is important that it happens, so the serial correlation of residuals within the model gradually decreases.

The Dickey-Fuller for unit root statistical test was used to formally confirm the null hypothesis that these four variables need to be differenced. The result is that there is not enough strong evidence to reject that $\rho = 1$, then these four variables will be differenced.

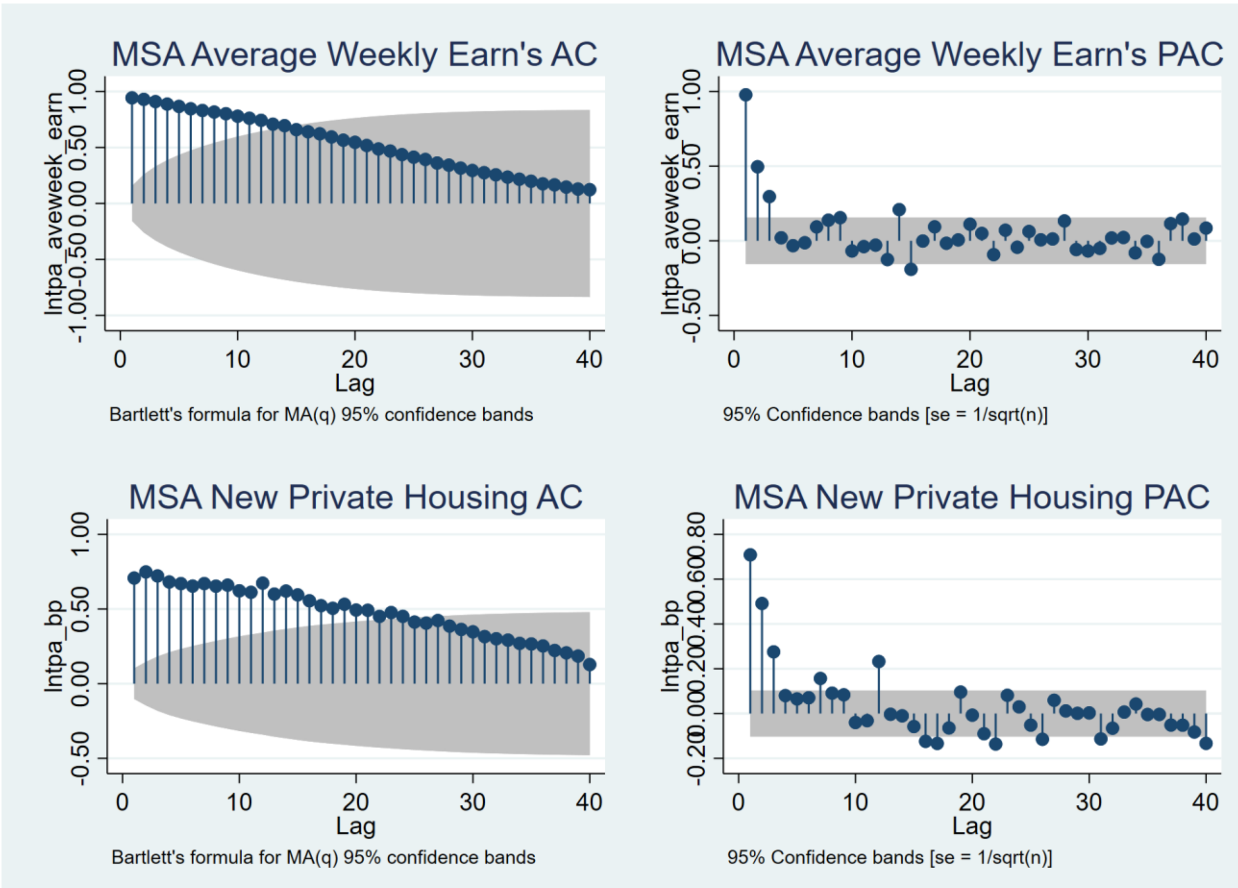
3.2 Variables' Auto Correlogram

Figure 5: AC for lntpa_priv and the Explanatory Variables

The same thing can be observed on the auto correlogram of the four variables presented above, the lagged effect takes more than 40 periods to completely disappear, so it shows that differencing is the right approach to conduct a forecast on with the selected variables. The only one that presents a slightly different behavior, is unemployment, but it still a highly-persistent time series.

3.3 AC and PAC for the Second Forecast Variables’

Figure 6: AC and PAC for lntpaave_week_earn and lntpa_bp



Lastly, it was necessary to conduct the same processes above, but with the variable that will also be used for the average weekly earnings forecast. The AC and PAC show that *lntpa_aveweek_earn* present a highly persistent time series, it can be observed that ρ’s value is close to 1, so there is strong evidence that it should be differenced. The Dickey-Fuller test was conducted and to formally confirmed that there is not enough strong evidence to reject that $\rho = 1$, then these four variables will be differenced.

On the other hand, lntpa_bp will not be differenced as its PAC is approximately 0.7, so the lagged effect dies relatively quickly, this was also confirmed by the DF unit-root test.

4.1 Tampa-St. Petersburg-Clearwater Private Employment Forecast

The 10 best models for forecasting private employment for Tampa-St. Petersburg-Clearwater were selected by using Global Search Regression method (GSREG). 24 months of data were reserved for out of sample's RMSE evaluation, 11-month indicator were fixed (from month 2 to month 12), and the criteria that will be used to evaluate the different models are AIC, BIC, and the out of sample root mean square error; GSREG went through 169765 models and ranked them according the selected criteria.

The selected models were the 1st, 2nd, 3rd, 4th, 5th, 6th, 11th, 18th, 25th, and 63rd. They were renamed as "Model 1" through 10 and the variables model are displayed below.

Table 7: GSREG's best 10 models

VAR \ MODEL	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
l1d.lntpa_priv							X			X
l2d.lntpa_priv										X
l3d.lntpa_priv	X	X	X	X	X	X	X	X	X	X
l6d.lntpa_priv	X			X	X	X	X			X
l12d.lntpa_priv	X	X	X	X	X		X	X	X	X
l24d.lntpa_priv			X		X				X	
l1d.lntpa_unemp										
l12d.lntpa_unemp	X	X	X		X	X	X	X	X	X
l1d.lntpa_tech								X	X	
l12d.lntpa_tech				X						
m2,3,4,5…,12	X	X	X	X	X	X	X	X	X	X

Firstly, the lntpa_serv variable did not appeared in any of the more relevant models, so this variable was discarded and the GSREG routine ran again, but with the tpa_tech variable, since the Technology industry had an important impact on the Tampa Bay area through the past years.

Furthermore, the most common features that these 10 models presented were lag differences 3, 6, and 12 for private employment; lag 12 is important as it represents the impact of exactly one cycle before on the same period, lags 3 and 6 represent a trimester and semester impact of what happened on the past, on future's private employment which could be the reflect of companies' reviewing their growth policies and planning. The same explanation applies for the 24^{th} lag of private employment, but it appeared in only 30% of the selected models.

The last variable that had a significant impact on model selection was the 12^{nd} lag of unemployment as it stores what happened exactly 1 year before and in the same month. There were some variables that had a small appearance rate (between 10 and 20% of the models) were the 1^{st} and 12^{th} lagged-differences for Technical Workers, it was a significant explanatory variable as the technological industry is currently growing at a fast rate. Lastly, the 1st and 2nd lag-differences for lntpa_priv did not appeared in most of the models, the 3rd lag difference had a bigger importance in terms of forecasting and the month indicator probably captured the effect of these lags.

4.2 Calculating the Error Using the Rolling Window Procedure

The next step compared the models and the selected criterion was the Rolling Window routine, as it provides RMSE of the out of sample forecast by differencing the actual value and the forecast value to obtain forecast error. The optimal window was selected by choosing the one that provided the lowest RMSE, the lowest-tested window size had 60 months and the largest 120 (10 years of data), the optimal width was 72, but the RMSE for w=60 was also documented for the purpose of comparison. The results can be observed on the table below:

Table 8: Rolling Window Root Mean Square Error Table

Model	Number of Variables	Rolling Window RMSE w = 72	Rolling Window RMSE w = 60
Baseline Model	48	0.00563933	0.00888157
Model 1	16	0.00372879	0.00383464
Model 2	15	0.00374256	0.00381513
Model 3	15	0.00385698	0.00396325
Model 4	16	0.00384226	0.00391004
Model 5	17	0.00386631	0.00401386
Model 6	15	0.00365800	0.00368501
Model 7	16	0.00365800	0.00407800
Model 8	16	0.00392266	0.00383306
Model 9	17	0.00386714	0.00397699
Model 10	18	0.00382010	0.00402472

After obtaining the rolling window RMSE for the 10 models, the best 4 models were select for a rolling window validation on the entire sample, each of the models were tested with a window width of 60 and 72 months.

Table 9: Rolling Window RMSE on the entire sample

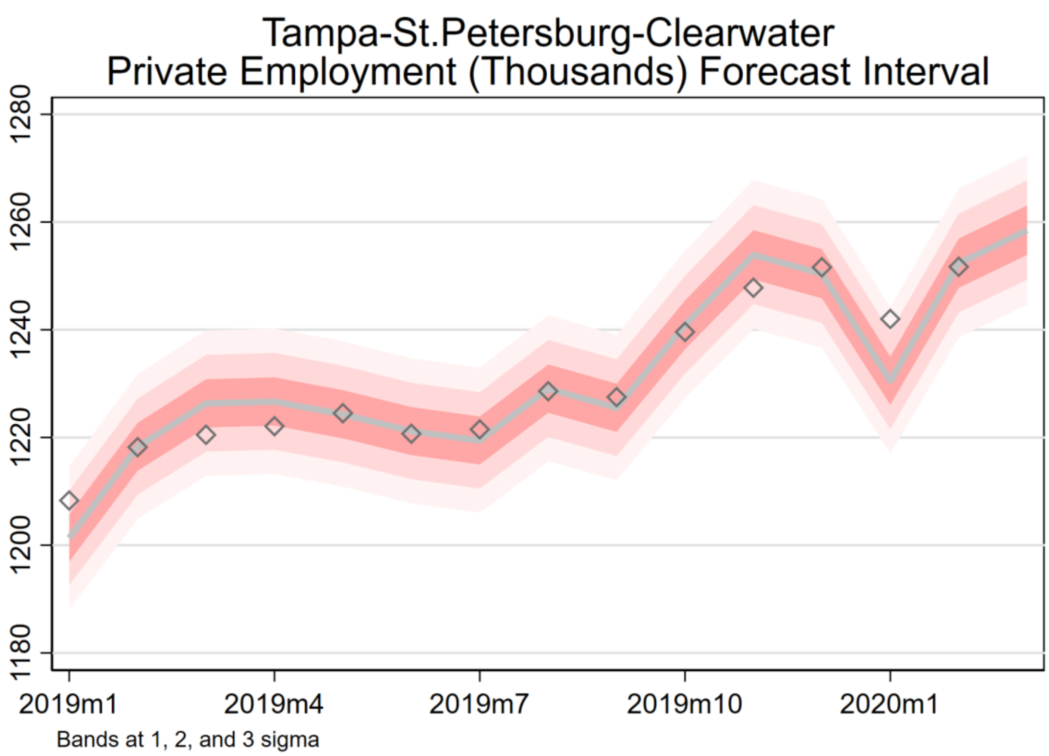
Model	Number of Variables	Rolling Window RMSE w = 72	Rolling Window RMSE = 60
Baseline Model	48	0.00519887	0.00777598
Model 1	16	0.00373772	0.00384119
Model 2	15	0.00381512	0.00391078
Model 6	15	0.00366889	0.00371705
Model 7	16	0.00385047	0.00402158

The result revealed that model 16, the was not only the more parsimonious of the forecasting models, but also the one that present the lowest forecasting error. This was the only model that did not present the 12th lag of private workers, instead it presented the 12th lag of unemployment which probably helped capturing the effect on private employees. Lastly, Model 6 only presented lags 3 and 6 of the response variable, and the month indicators, the fact that it had the lowest number of variables within all tested models could be determinant for having a the lowest rolling window RMSE.

4.3 Forecasting Private Employment

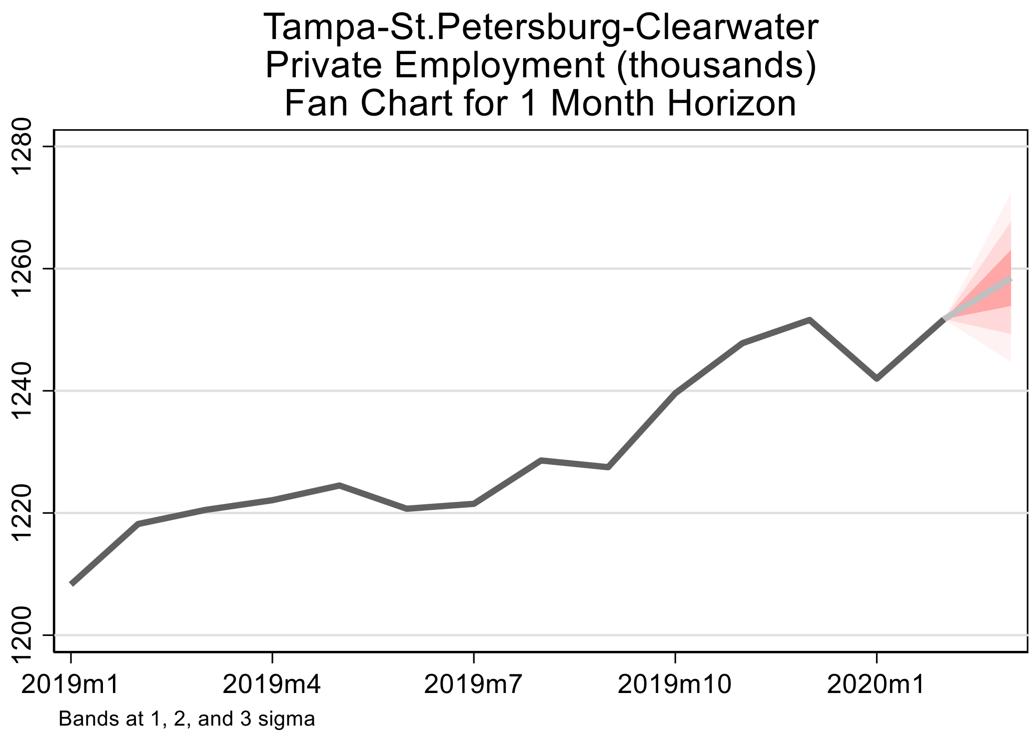
The forecasting interval for private employment within the MSA was calculated using the rolling window RMSE as it is a purely out of sample measurement. Furthermore, approximately normality was assumed as the forecasting interval will deal with 1, 2, and 3 standard deviations of the response variable. There was not a substantial difference between the empirical and gaussian forecasting intervals, so it is a fair assumption.

Figure 10: Forecasting Interval for tpa_priv



The forecasting interval on the figure above present a consistent result, the comparison between actual and forecast values show that 85% of the actual values fall within 1 standard deviation from the forecast, and the other 15% fall within 2σ. The future projection shows that private employment will likely increase for March of 2020, but the still the chance of decreasing, the fan chart on the next page better illustrates this prediction.

Figure 11: Fan Chart for 1 Month Horizon of tpa_priv



The forecast for shows growth perspective and it accurately predicted the in-sample actual values of private employment. If the result does not exactly follow the growth perspective as illustrated by the forecast as presented by this forecast, it is because of the Covid-19 influence on the labor market.

5.1 Tampa-St. Petersburg-Clearwater Average Weekly Earnings

The 20 best models for forecasting average weekly earnings for Tampa-St. Petersburg-Clearwater were selected by using Global Search Regression method (GSREG). 24 months of data were reserved for out of sample's RMSE evaluation, 11-month indicator were fixed (from month 2 to month 12), and the criteria that will be used to evaluate the different models are AIC, BIC, and the out of sample root mean square error; the GSREG routine ran twice and through went through 126,006 models and ranked them according the selected criteria.

The selected models were renamed as "Model 1" through 20, they were separated into two different groups as two GSREG routines were ran and the variables for each the models are displayed below on table below, the darker gray background means that the variable was not included on the GSREG for that model:

Table 12: GSREG's best 20 models

VAR \ MODEL	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
l1d.lntpa_ave_week_earn	X		X		X	X	X	X		X	X	X	X	X	X	X	X	X	X	X
l2d.lntpa_ave_week_earn	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
l3d.lntpa_ave_week_earn					X											X				
l4d.lntpa_ave_week_earn														X						
l6d.lntpa_ave_week_earn	X	X	X	X	X	X	X	X	X		X	X	X	X			X	X	X	X
l12d.lntpa_ave_week_earn																		X		
l24d.lntpa_ave_week_earn								X	X											
l36d.lntpa_ave_week_earn											X	X	X		X	X	X	X	X	X
l1d.lntpa_unemp											X		X	X		X	X	X	X	X
l2d.lntpa_unemp	X	X			X	X		X	X	X										
l2d.lntpa_tech						X	X													
l12d.lntpa_tech						X	X													
l1d.lntpa_priv	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X
l1d.lntpa_total_week_earn		X		X					X											
l1d.lntpa_bp																			X	
l2d.lntpa_bp																				X
l4d.lntpa_bp																	X			
m2,3,4,5…,12	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

The variables that had a most frequent appearance were the 1st, 2nd, 6th, 36th lagged differences of the average weekly earnings, the 1st, 2nd, and 6th lag-differences captured the impact of immediate changes on weekly earnings and the impact of a semester. The presence of the 36th lag as an efficient lag in terms of forecasting could be reasoned by the changes caused by past work contracts or political changes in the region due to elections.

The other explanatory variable that had a remarkable appearance on the model selection process were the 1st and 2nd lagged-differences of unemployment that are understandable as a higher unemployment rate could cause earning to decrease, and the 1st lagged-difference of private employment in the region, since an immediate change in this variable would directly affect earnings.

Lastly, some variables such as 3rd, 4th, 12th, and 24th lagged-difference had a significantly lower appearance because the other lag structure of 1st, 2nd, 6th, and 36th most likely did a better job forecasting the response variable, than these four; the 12th lag strangely does not frequently appear, it is the exact same month one year before.

Other explanatory variables were also used for modeling, such as Technical Employees and New Authorized Private Housings, but these also appeared on some specific models.

5.2 Calculating the Error Using the Rolling Window Procedure

The next step compared the models and the selected criterion was the Rolling Window routine, as it provides RMSE of the out of sample forecast by differencing the actual value and the forecast value to obtain forecast error. The optimal window was selected by choosing the one that provided the lowest RMSE, the lowest-tested window size had 60 month and the largest 120 (10 years of data), the optimal width was 72, but the RMSE for w=60 was also documented for the purpose of comparison. The results can be observed on the table below:

Table 13: Rolling Window Root Mean Square Error Table

Model	Number of Variables	Rolling Window RMSE w = 72	Rolling Window RMSE w = 60
Baseline Model	60	0.02607599	0.03382393
Model 1	17	0.01515101	0.01638406
Model 2	17	0.01515100	0.01638404
Model 3	16	0.01468432	0.01570439
Model 4	16	0.01468431	0.01570437
Model 5	18	0.01508680	0.01645600
Model 6	19	0.01554505	0.01729146
Model 7	18	0.01788451	0.01862166
Model 8	18	0.01555186	0.01645905
Model 9	18	0.01555184	0.01645903
Model 10	17	0.01609510	0.01682444
Model 11	18	0.01468636	0.01537400
Model 12	17	0.01439983	0.01525857
Model 13	17	0.01410208	0.01410615
Model 14	18	0.01478661	0.01554007
Model 15	16	0.01450512	0.01501677
Model 16	18	0.01464437	0.01552508
Model 17	19	0.01473219	0.01550527
Model 18	19	0.01527138	0.01692899
Model 19	18	0.01529544	0.01578350
Model 20	18	0.01500541	0.01571638

After obtaining the rolling window RMSE for the 20 models, the best 5 models for each GSREG routine were select for a rolling window validation on the entire sample, each of the models were tested with a window width of 60 and 72 months.

Table 14: Rolling Window RMSE on the entire sample

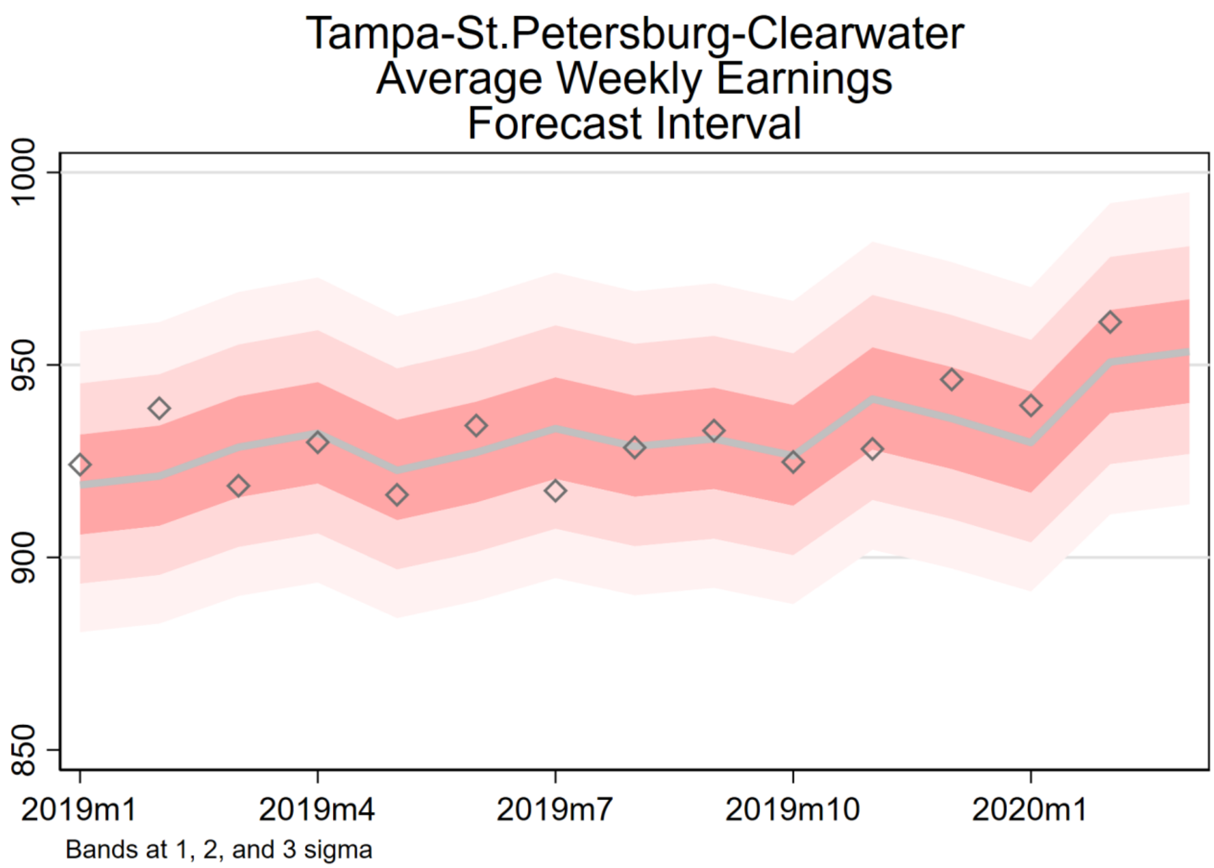
Model	Number of Variables	Rolling Window RMSE w = 72	Rolling Window RMSE = 60
Baseline Model	60	0.02628608	0.03430798
Model 1	17	0.01523177	0.01655930
Model 2	17	0.01523177	0.01655927
Model 3	16	0.01479594	0.01611956
Model 4	16	0.01479594	0.01611953
Model 5	18	0.01518784	0.01665850
Model 11	18	0.01469695	0.01528468
Model 12	17	0.01439978	0.01502587
Model 13	17	0.01415321	0.01425654
Model 15	16	0.0144085	0.01493113
Model 16	18	0.01474412	0.01613783

The result revealed that model 13, the only among the 20 models that did not include private employment, presented the lowest forecasting error. This was the only model included most of the common features, such as the 1^{st}, 2^{nd}, 6^{th}, and 36^{th} lagged-differences for the response variable that provided a significant better forecasting power with the inclusion of lag 36. Lastly, Model 13 only presented the month indicators and the first lag difference for unemployment, it had only one explanatory variable other the its own lags and month indicators, and the fact that it had a lower number of variables within all tested models could have been determinant for having the lowest rolling window RMSE.

5.3 Forecasting Private Employment

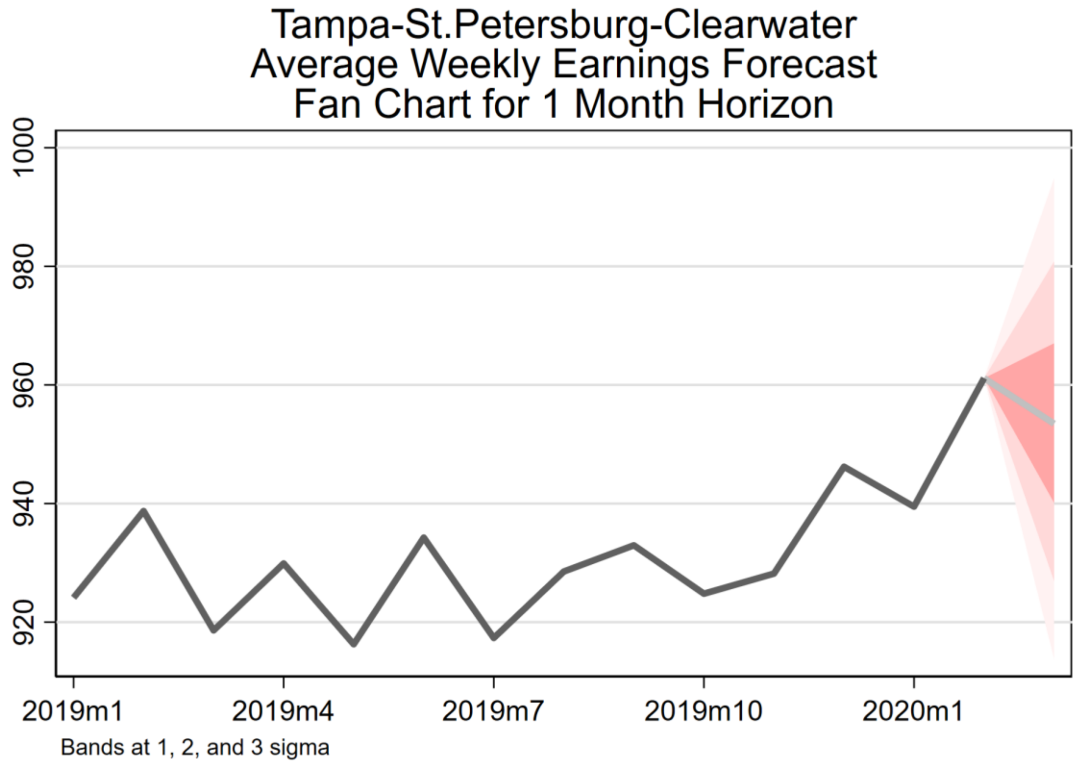
The forecasting interval for Average Weekly Earnings within the MSA was calculated using the rolling window RMSE as it is a purely out of sample measurement. Furthermore, approximately normality was assumed as the forecasting interval will deal with 1, 2, and 3 standard deviations of the response variable. There was not a substantial difference between the empirical and gaussian forecasting intervals, so it is a fair assumption.

Figure 15: Forecasting Interval for tpaave_week_earn



The forecasting interval on the figure above present a consistent result, the comparison between actual and forecast values show that 85% of the actual values fall within 1 standard deviation from the forecast, and the other 15% fall within 2σ. The future projection shows that average weekly earnings will likely stay stable for March of 2020, but with a slightly higher probability of decreasing. The rolling window RMSE was slight higher when compared to the Private Employment's prediction because the variability within Average Weekly Earnings was significantly higher and did not actually followed a definite pattern as it could be observed during the model selection process. The fan chart on the next page better illustrates this prediction.

Figure 16: Fan Chart for 1 Month Horizon of tpaaveweek_earn



The forecast for Average Weekly Earnings in Tampa-St. Petersburg-Clearwater shows a shrinking perspective and it accurately predicted the in-sample actual values of average weekly earnings between January of 2019 and February of 2020. The shrinking perspective of average weekly earnings could be increased by the Covid-19 influence on the labor market.

6. Conclusion

In conclusion, the selected models for forecasting purpose for both Average Weekly Earnings and Private Employment present a consistent result while forecasting their respective response variables, the actual value and point forecast fall no more than 1 standard deviation away from each other in 85% of the cases, and in every instance the distance was not greater than 2σ; it is also important to point out that the Standard Errors for both forecasts were not high, so the results were solid.

The limitations of involving these two forecasts are that they fail to capture the beginning effects of Covid-19 on the American Labor Market during March of 2020, but it still consistent and will most likely predict what happens one-period ahead. Lastly, this study did not try to forecast more than one period ahead since the World is currently passing through a moment of volatility and the impact of the Covid-19 pandemic will probably be more dramatic after March and an empirical model will not be able to capture the impact of the pandemic.

Appendix A: Do-file-for-Project-PrivEmployment

```
*t-initial = 1997 due tpa_tech

*Baseline Model  fro RW - w = 60

scalar drop _all

quietly forval w=60(12)120 {

gen pred=.

gen nobs=.

                    forval t=619/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1

                    reg d.lntpa_priv d.l(1/12)lntpa_priv d.l(1/12)lntpa_unemp
                    d.l(1/12)lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_priv)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5

summ nobs

scalar RWminobs`w'=r(min)

scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


*Selected models for RW - w = 60

*1

scalar drop _all

quietly forval w=60(12)120 {

gen pred=.

gen nobs=.
```

```
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv l12d.lntpa_priv
                    l12d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*2
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l12d.lntpa_priv l12d.lntpa_unemp
                    m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
```

```
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*3
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l12d.lntpa_priv l24d.lntpa_priv
                    l12d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*4
```

```
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv l12d.lntpa_priv
                    l12d.lntpa_unemp l12d.lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
                    m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*5
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv l12d.lntpa_priv
                    l24d.lntpa_priv l12d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
                    m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
```

```
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*6
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv l12d.lntpa_unemp m2
                    m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
```

```
}
scalar list

*11
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l1d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv
                    l12d.lntpa_priv l12d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
                    m12  if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*18
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
```

```
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l12d.lntpa_priv l12d.lntpa_unemp
                    l1d.lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12  if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*25
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
    reg d.lntpa_priv l3d.lntpa_priv l12d.lntpa_priv l24d.lntpa_priv l12d.lntpa_unemp
                    l1d.lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12  if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
```

```
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*63
scalar drop _all
quietly forval w=60(12)120 {
gen pred=.
gen nobs=.
                    forval t=619/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
    reg d.lntpa_priv l1d.lntpa_priv l2d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv
                    l12d.lntpa_priv l12d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
                    m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

* Fixed W = 72
```

```
*1
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=571/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv l12d.lntpa_priv
                    l12d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*2
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=571/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l12d.lntpa_priv l12d.lntpa_unemp
                    m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
```

```
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*6
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=571/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv l12d.lntpa_unemp m2
                    m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
```

```
}
scalar list

*11

scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=571/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l1d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv
                    l12d.lntpa_priv l12d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
                    m12  if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Baseline w =72

scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
```

```
gen nobs=.
                    forval t=571/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv d.l(1/12)lntpa_priv d.l(1/12)lntpa_unemp
                    d.l(1/12)lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 6
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=571/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_priv l3d.lntpa_priv l6d.lntpa_priv l12d.lntpa_unemp m2
                    m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
```

```
gen errsq=(pred-d.lntpa_priv)^2
summ errsq
gen rwpred = pred
scalar RWrmse`w'=r(mean)^.5
summ nobs rwpred
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

scalar rwrmse1 = 0.00366889

*Constructing a empirical interval - w = 72
*reg dlnfl_nonfarm l1dlnfl_nonfarm l2dlnfl_nonfarm l3dlnfl_nonfarm l5dlnfl_nonfarm
                    l6dlnfl_nonfarm l12dlnfl_nonfarm l2dlnfl_lf l2dlnus_epr m2 m3 m4
                    m5 m6 m7 m8 m9 m10 m11 m12 if tin(2014m2,2020m2)

gen residual=(d.lntpa_priv-rwpred)
gen expres=exp(residual)
summ expres
scalar meanexpres=r(mean)
_pctile residual, percentiles(2.5,97.5)
gen pye=meanexpres*exp(l.lntpa_priv+rwpred)
gen ubye=meanexpres*exp(l.lntpa_priv+rwpred+r(r2))
gen lbye=meanexpres*exp(l.lntpa_priv+rwpred+r(r1))

twoway (tsline tpa_priv if tin(2019m1,2020m4)) ///
                    (tsline pye ubye lbye if tin(2018m1,2020m4)), ///
                    title("Actual and Empirical Forecast Florida for Private Workers
                    for Tampa-St.Pt-Cl.") ytitle("") xtitle("") legend(label(1
                    "Actual") label(2 "Forecast") ///
                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m3yemp,
                    replace)

twoway (tsline tpa_priv if tin(2019m1,2020m4)) ///
                    (tsline pye ubye lbye if tin(2019m1,2020m4)), ///
```

```
                    title("Empirical Forecast") ytitle("") xtitle("") legend(label(1
                    "Actual") label(2 "Forecast") ///
                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m3yemp,
                    replace)

*Constructing a Gaussian intervar - w = 72

gen pyn=exp(l.lntpa_priv+rwpred+(rwrmse1^2)/2)
gen ubyn=exp(l.lntpa_priv+rwpred+1.96*rwrmse1+(rwrmse1^2)/2)
gen lbyn=exp(l.lntpa_priv+rwpred-1.96*rwrmse1+(rwrmse1^2)/2)
twoway (tsline tpa_priv if tin(2019m1,2020m2)) ///
                    (tsline pyn ubyn lbyn if tin(2019m1,2020m3)), ///
                    title("Actual and Approx. Normal Forecast Florida for Nonfarm-
                    Workers") ytitle("") xtitle("") legend(label(1 "Actual") label(2
                    "Forecast") ///
                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m3ynorm,
                    replace)

twoway (tsline tpa_priv if tin(2019m1,2020m2)) ///
                    (tsline pyn ubyn lbyn if tin(2019m1,2020m3)), ///
                    title("Approximately Normal Forecast") ytitle("") xtitle("")
                    legend(label(1 "Actual") label(2 "Forecast") ///
                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m3ynorm,
                    replace)

twoway (tsline tpa_priv if tin(2018m1,2020m2)) ///
                    (tsline pyn ubyn lbyn if tin(2019m1,2020m3)), ///
                    title("Approximately Normal Forecast") ytitle("") xtitle("")
                    legend(label(1 "Actual") label(2 "Forecast") ///
                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m3ynorm2,
                    replace)

graph combine m3ynorm.gph m3yemp.gph , ///
                    saving(m3yen, replace)

*Chart one month ahead - Empirical
gen fub=ubye if tin(2020m3,)
gen flb=lbye if tin(2020m3,)
gen fcst=pye if tin(2020m3,)
replace fcst=tpa_priv if tin(2020m2,2020m2)
```

```
replace fub=tpa_priv if tin(2020m2,2020m2)

replace flb=tpa_priv if tin(2020m2,2020m2)


*Chart one month ahead - Normal

twoway(tsline tpa_priv if tin(2019m1,2020m2))(tsline fub flb fcst if
                    tin(2020m2,2020m3) ), title("Empirical Forecast") ytitle("")
                    xtitle("") legend(label(1 "Actual") label(2 "Upper Bound") ///

                    label(3 "Lower Bound") label(4 "Forecast")) saving(fcste, replace)


replace fub=ubyn if tin(2020m3,)

replace flb=lbyn if tin(2020m3,)

replace fcst=pyn if tin(2020m3,)

replace fcst=tpa_priv if tin(2020m2,2020m2)

replace fub=tpa_priv if tin(2020m2,2020m2)

replace flb=tpa_priv if tin(2020m2,2020m2)


twoway(tsline tpa_priv if tin(2019m1,2020m2))(tsline fub flb fcst if
                    tin(2020m2,2020m3) ), title("Approximately Normal Forecast")
                    ytitle("") xtitle("") legend(label(1 "Actual") label(2 "Upper
                    Bound") ///

                    label(3 "Lower Bound") label(4 "Forecast")) saving(fcstn, replace)


graph combine fcstn.gph fcste.gph , ///

                    saving(fcts, replace)


*FAN CHART


*H=1

scalar rwrmse2 = 0.00366889

gen ptpae=exp((rwrmse2^2)/2)*exp(l.lntpa_priv+rwpred)

gen ub1=exp((rwrmse2^2)/2)*exp(l.lntpa_priv+rwpred+1*rwrmse2)

gen lb1=exp((rwrmse2^2)/2)*exp(l.lntpa_priv+rwpred-1*rwrmse2)

gen ub2=exp((rwrmse2^2)/2)*exp(l.lntpa_priv+rwpred+2*rwrmse2)

gen lb2=exp((rwrmse2^2)/2)*exp(l.lntpa_priv+rwpred-2*rwrmse2)

gen ub3=exp((rwrmse2^2)/2)*exp(l.lntpa_priv+rwpred+3*rwrmse2)

gen lb3=exp((rwrmse2^2)/2)*exp(l.lntpa_priv+rwpred-3*rwrmse2)
```

```
*Fan Charts

twoway (tsrline ub3 ub2 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(5) lwidth(none) ) ///
                    (tsrline ub2 ub1 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(15) lwidth(none) ) ///
                    (tsrline ub1 ptpae if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(35) lwidth(none) ) ///
                    (tsrline ptpae lb1 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(35) lwidth(none) ) ///
                    (tsrline lb1 lb2 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(15) lwidth(none) ) ///
                    (tsrline lb2 lb3 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(5) lwidth(none) ) ///
                    (tsline ptpae if tin(2019m1,2020m3) , ///
                    lcolor(gs12) lwidth(thick thick) ) ///
                    (scatter tpa_priv date if tin(2019m1,2020m3), lcolor(gs6)), ///
                    scheme(s1mono) legend(off) ///
                    title("Tampa-St.Petersburg-Clearwater"  ///
                    "Private Employment (Thousands) Forecast Interval") legend(off)
                    ///
                    xtitle("") ylabel(,grid)  ///
                    note("Bands at 1, 2, and 3 sigma")

gen fptpae=tpa_priv if tin(2020m2,2020m2)
gen fub1=tpa_priv if tin(2020m2,2020m2)
gen fub2=tpa_priv if tin(2020m2,2020m2)
gen fub3=tpa_priv if tin(2020m2,2020m2)
gen flb1=tpa_priv if tin(2020m2,2020m2)
gen flb2=tpa_priv if tin(2020m2,2020m2)
gen flb3=tpa_priv if tin(2020m2,2020m2)

replace fptpae=ptpae if tin(2020m3,2020m3)
replace fub1=ub1 if tin(2020m3,2020m3)
replace fub2=ub2 if tin(2020m3,2020m3)
replace fub3=ub3 if tin(2020m3,2020m3)
```

```
replace flb1=lb1 if tin(2020m3,2020m3)
replace flb2=lb2 if tin(2020m3,2020m3)
replace flb3=lb3 if tin(2020m3,2020m3)

twoway (tsrline fub3 fub2 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(5) lwidth(none) ) ///
                    (tsrline fub2 fub1 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(15) lwidth(none) ) ///
                    (tsrline fub1 fptpae if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(35) lwidth(none) ) ///
                    (tsrline fptpae flb1 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(35) lwidth(none) ) ///
                    (tsrline flb1 flb2 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(15) lwidth(none) ) ///
                    (tsrline flb2 flb3 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(5) lwidth(none) ) ///
                    (tsline fptpae if tin(2020m2,2020m3) , ///
                    lcolor(gs12) lwidth(thick thick) ) ///
                    (tsline tpa_priv if tin(2019m1,2020m3) , ///
                    lcolor(gs6) lwidth(thick thick) ), scheme(s1mono) legend(off) ///
                    title("Tampa-St.Petersburg-Clearwater"  ///
                    "Private Employment (thousands)" ///
                    "Fan Chart for 1 Month Horizon") legend(off) ///
                    xtitle("") ylabel(,grid)  ///
                    note("Bands at 1, 2, and 3 sigma")
```

Appendix A: Do-file-for-Project-AveWeekEarns

```
* Variable will be loaded by using FRED use, so the most recent data can be captured

freduse FLNAN SMU12453000500000001 SMU12453000500000002 SMU12453000500000003
                    SMU12453000500000011 SMU12453000700000001 SMU12453006054000001
                    TAMP312BPPRIV TAMP312NAN TAMP312URN

*Datestring generation
rename date datestring
gen datec=date(datestring,"YMD")
gen date=mofd(datec)
format date %tm
tsset date

*Adjusting Observations
keep if tin(2007m1,)
tsappend, add(1)
tsset date

* Month indicators
generate month=month(datec)
replace month=month(dofm(date)) if month==.
gen m1=0
replace m1=1 if month==1
gen m2=0
replace m2=1 if month==2
gen m3=0
replace m3=1 if month==3
gen m4=0
replace m4=1 if month==4
gen m5=0
replace m5=1 if month==5
gen m6=0
replace m6=1 if month==6
gen m7=0
replace m7=1 if month==7
```

```
gen m8=0
replace m8=1 if month==8
gen m9=0
replace m9=1 if month==9
gen m10=0
replace m10=1 if month==10
gen m11=0
replace m11=1 if month==11
gen m12=0
replace m12=1 if month==12

* FLNAN = Florida Non Farm Employees
rename FLNAN fl_nonfarm
gen lnfl_nonfarm=ln(fl_nonfarm)

*SMU12453000500000001 = Total Private Employees in Tampa-St. Petersburg-Clearwater, FL
                    (MSA)
rename SMU12453000500000001 tpa_priv
gen lntpa_priv=ln(tpa_priv)
label variable tpa_priv "Total Private Employees"

*SMU12453000500000002 = Average Weekly Hours of All Employees: Total Private in Tampa-
                    St.    Petersburg-Clearwater, FL (MSA)
rename SMU12453000500000002 tpa_aveweek_hour
label variable tpa_aveweek_hour "Average Weekly Hours"
gen lntpa_aveweek_hour=ln(tpa_aveweek_hour)

*SMU12453000500000003 = Average Hourly Earnings of All Employees: Total Private in
                    Tampa-St.  Petersburg-Clearwater, FL (MSA)
rename SMU12453000500000003 tpa_avehour_earn
label variable tpa_avehour_earn "Average Hourly Earnings"
gen lntpa_avehour_earn=ln(tpa_avehour_earn)

*SMU12453000500000011 = Average Weekly Earnings of All Employees: Total Private in
                    Tampa-St.  Petersburg-Clearwater, FL (MSA)
rename SMU12453000500000011 tpa_aveweek_earn
label variable tpa_aveweek_earn "Average Weekly Earnings"
```

```
gen lntpa_aveweek_earn=ln(tpa_aveweek_earn)


* SMU12453000700000001 = All Employees: Service-Providing in Tampa-St. Petersburg-
                    Clearwater, FL (MSA)

rename SMU12453000700000001 tpa_serv

gen lntpa_serv=ln(tpa_serv)

label variable tpa_serv "Service-Providing Employees"


* SMU12453006054000001 = All Employees: Professional, Scientific, and Technical
                    Services in    Tampa-St. Petersburg-Clearwater, FL (MSA)

rename SMU12453006054000001 tpa_tech

gen lntpa_tech=ln(tpa_tech)

label variable tpa_tech "Total Technical Employees"


* TAMP312BPPRIV = New Private Housing Units Authorized by Building Permits for Tampa-
                    St. Petersburg-Clearwater, FL (MSA)

rename TAMP312BPPRIV tpa_bp

gen lntpa_bp=ln(tpa_bp)

label variable tpa_bp "Private Housing Authorized"


* TAMP312NAN  = All Employees: Total Nonfarm in Tampa-St. Petersburg-Clearwater, FL
                    (MSA)

rename TAMP312NAN tpa_nonfarm

gen lntpa_nonfarm=ln(tpa_nonfarm)

label variable tpa_nonfarm "Total Nonfarm Employees"


* TAMP312URN = Unemployment Rate in Tampa-St. Petersburg-Clearwater, FL (MSA)

rename TAMP312URN tpa_unemp

gen lntpa_unemp=ln(tpa_unemp)

label variable tpa_unemp "Unemployment"


* Total Weekly earning

gen tpa_totalweek_earn = tpa_priv*tpa_aveweek_earn

label variable tpa_totalweek_earn "Total Weekly Earnings (thousands)"

gen lntpa_totalweek_earn = ln(tpa_totalweek_earn)

label variable lntpa_totalweek_earn "Log of Total Weekly Earnings"
```

```
* Summary of all variables
summarize *

* Variables description
describe *

* Tsline for predictors
twoway (tsline tpa_aveweek_earn if tin(2007m1,2020m2) , ///
                    lcolor(gs6)) ///
                    (tsline tpa_unemp, yaxis(2)), ///
                    scheme(s1mono) ///
                    title("Time Series Plot of"  ///
                    "Unemployment and Average Weekly Earnings") legend(on) xtitle("")
                    saving(var1, replace)

twoway (tsline tpa_aveweek_earn if tin(2007m1,2020m2) , ///
                    lcolor(gs6)) ///
                    (tsline tpa_tech, yaxis(2)), ///
                    scheme(s1mono) ///
                    title("Time Series Plot of"  ///
                    "Total Technical Employees and" ///
                    "Average Weekly Earnings") legend(on) xtitle("") saving(var2,
                    replace)

twoway (tsline tpa_aveweek_earn if tin(2007m1,2020m2) , ///
                    lcolor(gs6)) ///
                    (tsline tpa_priv, yaxis(2)), ///
                    scheme(s1mono) ///
                    title("Time Series Plot of"  ///
                    "Total Private Employees and" ///
                    "Average Weekly Earnings") legend(on) xtitle("") saving(var3,
                    replace)

twoway (tsline tpa_aveweek_earn if tin(2007m1,2020m2) , ///
                    lcolor(gs6)) ///
                    (tsline tpa_bp, yaxis(2)), ///
                    scheme(s1mono) legend(off) ///
```

```
                    title("Time Series Plot of"  ///
                    "New Private Housing Authorized" ///
                    "and Average Weekly Earnings") legend(on) xtitle("") saving(var4,
                    replace)

twoway (tsline lntpa_aveweek_hour) if tin(2007m1,)
twoway (tsline lntpa_aveweek_earn) if tin(2007m1,)
twoway (tsline tpa_avehour_earn) if tin(2007m1,)
twoway (tsline tpa_totalweek_earn) if tin(2007m1,)
twoway (tsline tpa_aveweek_earn) if tin(2007m1,)
twoway (tsline lntpa_priv) if tin(1990m1,)
*Extra explanatory variables
twoway (tsline lntpa_tech) if tin(1997m1,)
twoway (tsline lntpa_unemp)  if tin(1990m1,)
twoway (tsline lntpa_bp) if tin(1990m1,)
twoway (tsline lntpa_serv) if tin(1990m1,)
twoway (tsline lntpa_nonfarm) if tin(1990m1,)
twoway (tsline fl_nonfarm) if tin(1990m1,)

graph combine var1.gph var2.gph var3.gph var4.gph , ///
                    saving(vars, replace)

*Predicting lntpa_aveweek_earn
set seed 22045
reg d.lntpa_aveweek_earn d.l(1/12,24,36)lntpa_aveweek_earn d.l(1,2,12)tpa_unemp
                    d.l(1,2,12)lntpa_tech d.l(1,2,12)lntpa_totalweek_earn
                    d.l(1,2,12)lntpa_priv m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
predict res1 if e(sample)==1, residual
pac res1
bgodfrey, lag(1/24)
drop res1

reg d.lntpa_aveweek_earn d.l(1/12,24,36)lntpa_aveweek_earn d.l(1,2,4,12)tpa_unemp
                    d.l(1,2,4,12)lntpa_priv d.l(1,2,4,12,24)lntpa_totalweek_earn m2 m3
                    m4 m5 m6 m7 m8 m9 m10 m11 m12
predict res2 if e(sample)==1, residual
pac res2
bgodfrey, lag(1/24)
```

```
drop res2


reg d.lntpa_aveweek_earn d.l(1/12,36)lntpa_aveweek_earn d.l(1,2,4,12)tpa_unemp
                    d.l(1,2,4,12)lntpa_priv d.l(1,2,4,12)lntpa_bp m2 m3 m4 m5 m6 m7 m8
                    m9 m10 m11 m12

predict res3 if e(sample)==1, residual

pac res3

bgodfrey, lag(1/24)

drop res3


reg d.lntpa_aveweek_earn d.l(1/12,36)lntpa_aveweek_earn d.l(1,2,4,12)tpa_unemp
                    d.l(1,2,4,12)lntpa_priv d.l(1,2,4,12)lntpa_bp m2 m3 m4 m5 m6 m7 m8
                    m9 m10 m11 m12

predict res3 if e(sample)==1, residual

pac res3

bgodfrey, lag(1/24)

drop res3


*Generating dummy variables


gen dlntpa_avehour_earn = d.lntpa_totalweek_earn

gen l1dlntpa_avehour_earn = l1d.lntpa_totalweek_earn

gen l2dlntpa_avehour_earn = l2d.lntpa_totalweek_earn

gen l3dlntpa_avehour_earn = l3d.lntpa_totalweek_earn

gen l4dlntpa_avehour_earn = l4d.lntpa_totalweek_earn

gen l5dlntpa_avehour_earn = l5d.lntpa_totalweek_earn

gen l6dlntpa_avehour_earn = l6d.lntpa_totalweek_earn

gen l7dlntpa_avehour_earn = l7d.lntpa_totalweek_earn

gen l8dlntpa_avehour_earn = l8d.lntpa_totalweek_earn

gen l9dlntpa_avehour_earn = l9d.lntpa_totalweek_earn

gen l10dlntpa_avehour_earn = l10d.lntpa_totalweek_earn

gen l11dlntpa_avehour_earn = l11d.lntpa_totalweek_earn

gen l12dlntpa_avehour_earn = l12d.lntpa_totalweek_earn

gen l24dlntpa_avehour_earn = l24d.lntpa_totalweek_earn

gen l36dlntpa_avehour_earn = l36d.lntpa_totalweek_earn


gen l1dlntpa_totalweek_earn = l1d.lntpa_totalweek_earn
```

```
gen l2dlntpa_totalweek_earn = l2d.lntpa_totalweek_earn
gen l4dlntpa_totalweek_earn = l4d.lntpa_totalweek_earn
gen l12dlntpa_totalweek_earn = l12d.lntpa_totalweek_earn

gen l1dlntpa_priv = l1d.lntpa_priv
gen l2dlntpa_priv = l2d.lntpa_priv
gen l4dlntpa_priv = l4d.lntpa_priv
gen l12dlntpa_priv = l12d.lntpa_priv

gen l1dlntpa_unemp = l1d.lntpa_unemp
gen l2dlntpa_unemp = l2d.lntpa_unemp
gen l4dlntpa_unemp = l4d.lntpa_unemp
gen l12dlntpa_unemp = l12d.lntpa_unemp

gen l1dlntpa_tech = l1d.lntpa_tech
gen l2dlntpa_tech = l2d.lntpa_tech
gen l4dlntpa_tech = l4d.lntpa_tech
gen l12dlntpa_tech = l12d.lntpa_tech

gen l1dlntpa_bp = l1d.lntpa_bp
gen l2dlntpa_bp = l2d.lntpa_bp
gen l4dlntpa_bp = l4d.lntpa_bp
gen l12dlntpa_bp = l12d.lntpa_bp

*GSREG
gsreg dlntpa_avehour_earn l1dlntpa_avehour_earn l2dlntpa_avehour_earn
                    l3dlntpa_avehour_earn ///
/*
                    l4dlntpa_avehour_earn l6dlntpa_avehour_earn  ///
                    l11dlntpa_avehour_earn l12dlntpa_avehour_earn
                    l24dlntpa_avehour_earn ///
                     l1dlntpa_totalweek_earn ///
                    l12dlntpa_totalweek_earn l1dlntpa_priv l12dlntpa_priv ///
                    l1dlntpa_unemp l2dlntpa_unemp l12dlntpa_unemp l1dlntpa_tech
                    l2dlntpa_tech ///
                    l12dlntpa_tech if tin(2007m1, 2020m3), ///
```

```
                    ncomb(1,7) aic outsample(24) fix(m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
                    m12) samesample ///

                    nindex( -0.3 aic -0.3 bic -0.4 rmse_out)
                    results(gsreg_dlntpa_earn) replace

*/

*GSREG With new variables and the most recurrent ones

gsreg dlntpa_avehour_earn l1dlntpa_avehour_earn l2dlntpa_avehour_earn
                    l3dlntpa_avehour_earn ///

/*

                    l4dlntpa_avehour_earn l6dlntpa_avehour_earn  ///

                     l12dlntpa_avehour_earn l36dlntpa_avehour_earn ///

    l1dlntpa_priv l2dlntpa_priv l4dlntpa_priv l12dlntpa_priv ///

                    l1dlntpa_unemp l2dlntpa_unemp l12dlntpa_unemp ///

                    l1dlntpa_bp l2dlntpa_bp l4dlntpa_bp l12dlntpa_bp if tin(2007m1,
                    2020m3), ///

                    ncomb(1,7) aic outsample(24) fix(m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
                    m12) samesample ///

                    nindex( -0.3 aic -0.3 bic -0.4 rmse_out)
                    results(gsreg_dlntpa_earn2) replace

                    */

*Best models

*1 - M1

reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_unemp m2 m3 m4 m5
                    m6 m7 m8 m9 m10 m11 m12

*2 - M2

reg d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn
                    l1d.lntpa_totalweek_earn l1d.lntpa_priv l2d.lntpa_unemp m2 m3 m4
                    m5 m6 m7 m8 m9 m10 m11 m12

*4 - M3

reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv m2 m3 m4 m5 m6 m7 m8 m9 m10
                    m11 m12

*5 - M4

reg d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_totalweek_earn m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12

*17 - M5

reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l3d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12

*Good BIC and AIC

*24 - M6
```

```
reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_unemp
                    l2d.lntpa_tech l12.lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12

*27 - M7

reg d.lntpa_aveweek_earn l1.lntpa_totalweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_tech
                    l12.lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12

*Good RMSE out

*35 - M8

reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l24d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12

*36 -M9

reg d.lntpa_aveweek_earn l1d.lntpa_totalweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l24d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12

*337-M10

reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp l12d.lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
                    m12


*GSREG 2 - Selected models


*1st - M11

reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12


*2nd - M12

reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn l1d.lntpa_priv m2 m3 m4 m5 m6
                    m7 m8 m9 m10 m11 m12


*3rd - M13

reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn l1d.lntpa_unemp m2 m3 m4 m5
                    m6 m7 m8 m9 m10 m11 m12


*4th - M14

reg d.lntpa_aveweek_earn d.l(1,2,4,6,36)lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12


*5th - M15

reg d.lntpa_aveweek_earn d.l(1,2,36)lntpa_aveweek_earn l1d.lntpa_priv m2 m3 m4 m5 m6
                    m7 m8 m9 m10 m11 m12
```

```
*11 - M16

reg d.lntpa_aveweek_earn d.l(1,2,3,6,36)lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12


*13 - M17

reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_unemp l4d.lntpa_bp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12


*14 - M18

reg d.lntpa_aveweek_earn d.l(1,2,6,12,36)lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12


*17 - M19

reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_unemp l1d.lntpa_bp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12


*18 - M20

reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_unemp l2d.lntpa_bp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12


*t-initial = 2007 due tpa_tech

*Baseline Model for RW

scalar drop _all

quietly forval w=60(12)84 {

gen pred=.

gen nobs=.

                    forval t=648/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1

                    reg d.lntpa_aveweek_earn d.l(1/12)lntpa_aveweek_earn
                    d.l(1/12)tpa_unemp d.l(1/12)lntpa_priv
                    d.l(1/12)lntpa_totalweek_earn m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }
```

```
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Selected models for RW - w = 60(12)84
*Model 1
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list
```

```
*Model 2
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_totalweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 3
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
```

```
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv m2 m3
                    m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 4
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l1d.lntpa_totalweek_earn m2
                    m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
```

```
summm nobs
```

```
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 5
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l3d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_unemp m2 m3 m4 m5
                    m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 6
scalar drop _all
quietly forval w=60(12)84 {
```

```
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp l2d.lntpa_tech l12.lntpa_tech m2 m3 m4 m5 m6 m7 m8
                    m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 7
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1.lntpa_totalweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_tech l12.lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
```

```
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 8
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn
                    l24d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_unemp m2 m3 m4 m5
                    m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
```

```
}
scalar list

*Model 9
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_totalweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn
                    l24d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_unemp m2 m3 m4 m5
                    m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 10

scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
```

```
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_unemp
                    l12d.lntpa_tech m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

* Fixed W = 72

*Baseline Model for RW
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=636/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1/12)lntpa_aveweek_earn
                    d.l(1/12)tpa_unemp d.l(1/12)lntpa_priv
                    d.l(1/12)lntpa_totalweek_earn m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
```

```
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 1
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=636/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
```

```
scalar list


*Model 2

scalar drop _all

quietly forval w=72(12)72 {

gen pred=.

gen nobs=.

                    forval t=636/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1

                    reg d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_totalweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5

summ nobs

scalar RWminobs`w'=r(min)

scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


*Model 3

scalar drop _all

quietly forval w=72(12)72 {

gen pred=.

gen nobs=.

                    forval t=636/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1
```

```
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv m2 m3
                    m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5

summ nobs

scalar RWminobs`w'=r(min)

scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


*Model 4

scalar drop _all

quietly forval w=72(12)72 {

gen pred=.

gen nobs=.

                    forval t=636/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1

                    reg d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l1d.lntpa_totalweek_earn m2
                    m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5
```

```
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 5
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=636/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l3d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_unemp m2 m3 m4 m5
                    m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Fixed W =60
```

```
*Baseline Model for RW
scalar drop _all
quietly forval w=60(12)60 {
gen pred=.
gen nobs=.
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1/12)lntpa_aveweek_earn
                    d.l(1/12)tpa_unemp d.l(1/12)lntpa_priv
                    d.l(1/12)lntpa_totalweek_earn m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


scalar drop _all
quietly forval w=60(12)60 {
gen pred=.
gen nobs=.
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
```

```
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 2
scalar drop _all
quietly forval w=60(12)60 {
gen pred=.
gen nobs=.
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_totalweek_earn l1d.lntpa_priv
                    l2d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
```

```
scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


*Model 3

scalar drop _all

quietly forval w=60(12)60 {

gen pred=.

gen nobs=.

                    forval t=624/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1

                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l6d.lntpa_aveweek_earn l1d.lntpa_priv m2 m3
                    m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5

summ nobs

scalar RWminobs`w'=r(min)

scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


*Model 4

scalar drop _all

quietly forval w=60(12)60 {

gen pred=.

gen nobs=.
```

```
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l2d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l1d.lntpa_totalweek_earn m2
                    m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 5
scalar drop _all
quietly forval w=60(12)60 {
gen pred=.
gen nobs=.
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn l1d.lntpa_aveweek_earn
                    l2d.lntpa_aveweek_earn l3d.lntpa_aveweek_earn
                    l6d.lntpa_aveweek_earn l1d.lntpa_priv l2d.lntpa_unemp m2 m3 m4 m5
                    m6 m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
```

```
gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5

summ nobs

scalar RWminobs`w'=r(min)

scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


* GSREG 2 - ROLLING WINDOWS


*t-initial = 2007

*Baseline Model for RW


*Selected models for RW - w = 60(12)84

*Model 11

scalar drop _all

quietly forval w=60(12)84 {

gen pred=.

gen nobs=.

                    forval t=648/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1

                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5

summ nobs
```

```
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 12
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart
                    & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 13
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
```

```
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 14
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,4,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
```

```
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 15
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,36)lntpa_aveweek_earn
                    l1d.lntpa_priv m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart
                    & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list
```

```
*Model 16
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,3,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 17
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
```

```
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp l4d.lntpa_bp m2 m3 m4 m5 m6 m7 m8
                    m9 m10 m11 m12 if date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5

summ nobs

scalar RWminobs`w'=r(min)

scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


*Model 18

scalar drop _all

quietly forval w=60(12)84 {

gen pred=.

gen nobs=.

                    forval t=648/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1

                    reg d.lntpa_aveweek_earn d.l(1,2,6,12,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5
```

```
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 19
scalar drop _all
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp l1d.lntpa_bp m2 m3 m4 m5 m6 m7 m8
                    m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 20

scalar drop _all
```

```
quietly forval w=60(12)84 {
gen pred=.
gen nobs=.
                    forval t=648/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp l2d.lntpa_bp m2 m3 m4 m5 m6 m7 m8
                    m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Fixed Windows

* Fixed W = 72

*Model 11
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=636/722 {
                    gen wstart=`t'-`w'
```

```
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 12
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=636/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart
                    & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
```

```
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 13
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=636/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 15
scalar drop _all
```

```
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=636/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1

reg d.lntpa_aveweek_earn d.l(1,2,36)lntpa_aveweek_earn l1d.lntpa_priv m2 m3 m4 m5 m6
                    m7 m8 m9 m10 m11 m12 if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 16
scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
                    forval t=636/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,3,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
```

```
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Fixed W =60

*Model 11

scalar drop _all
quietly forval w=60(12)60 {
gen pred=.
gen nobs=.
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
                    if date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
```

```
scalar RWminobs`w'=r(min)

scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


*Model 12

scalar drop _all

quietly forval w=60(12)60 {

gen pred=.

gen nobs=.

                    forval t=624/722 {

                    gen wstart=`t'-`w'

                    gen wend=`t'-1

                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_priv m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart
                    & date<=wend

                    replace nobs=e(N) if date==`t'

                    predict ptemp

                    replace pred=ptemp if date==`t'

                    drop ptemp wstart wend

                    }

gen errsq=(pred-d.lntpa_aveweek_earn)^2

summ errsq

scalar RWrmse`w'=r(mean)^.5

summ nobs

scalar RWminobs`w'=r(min)

scalar RWmaxobs`w'=r(max)

drop errsq pred nobs

}

scalar list


*Model 3

scalar drop _all

quietly forval w=60(12)60 {

gen pred=.
```

```
gen nobs=.
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list


*Model 15
scalar drop _all
quietly forval w=60(12)60 {
gen pred=.
gen nobs=.
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,36)lntpa_aveweek_earn
                    l1d.lntpa_priv m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if date>=wstart
                    & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
```

```
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 16
scalar drop _all
quietly forval w=60(12)60 {
gen pred=.
gen nobs=.
                    forval t=624/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1

reg d.lntpa_aveweek_earn d.l(1,2,3,6,36)lntpa_aveweek_earn l1d.lntpa_priv
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12  if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
```

```
scalar list

*SELECTED ONE
*MODEL 13

scalar drop _all
quietly forval w=96(12)96 {
gen pred=.
gen nobs=.
                    forval t=662/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

*Model 13

scalar drop _all
quietly forval w=72(12)72 {
gen pred=.
gen nobs=.
```

```
                    forval t=636/722 {
                    gen wstart=`t'-`w'
                    gen wend=`t'-1
                    reg d.lntpa_aveweek_earn d.l(1,2,6,36)lntpa_aveweek_earn
                    l1d.lntpa_unemp m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 if
                    date>=wstart & date<=wend
                    replace nobs=e(N) if date==`t'
                    predict ptemp
                    replace pred=ptemp if date==`t'
                    drop ptemp wstart wend
                    }
gen errsq=(pred-d.lntpa_aveweek_earn)^2
summ errsq
gen rwpred = pred
scalar RWrmse`w'=r(mean)^.5
summ nobs rwpred
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

scalar rwrmse2 = 0.01415321

*Constructing a empirical interval - w = 72

gen residual=(d.lntpa_aveweek_earn-rwpred)
gen expres=exp(residual)
summ expres
scalar meanexpres=r(mean)
_pctile residual, percentiles(2.5,97.5)
gen pye=meanexpres*exp(l.lntpa_aveweek_earn+rwpred)
gen ubye=meanexpres*exp(l.lntpa_aveweek_earn+rwpred+r(r2))
gen lbye=meanexpres*exp(l.lntpa_aveweek_earn+rwpred+r(r1))

twoway (tsline tpa_aveweek_earn if tin(2018m1,2020m4)) ///
```

```
                    (tsline pye ubye lbye if tin(2019m1,2020m4)), ///

                    title("Actual and Empirical Forecast Florida for MSA's Average
                    Weekly Earnings") ytitle("") xtitle("") legend(label(1 "Actual")
                    label(2 "Forecast") ///

                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m4yemp,
                    replace)


twoway (tsline tpa_aveweek_earn if tin(2019m1,2020m4)) ///

                    (tsline pye ubye lbye if tin(2019m1,2020m4)), ///

                    title("Empirical Forecast") ytitle("") xtitle("") legend(label(1
                    "Actual") label(2 "Forecast") ///

                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m4yemp,
                    replace)


*Constructing a Gaussian intervar - w = 72


gen pyn=exp(l.lntpa_aveweek_earn+rwpred+(rwrmse2^2)/2)

gen ubyn=exp(l.lntpa_aveweek_earn+rwpred+1.96*rwrmse2+(rwrmse2^2)/2)

gen lbyn=exp(l.lntpa_aveweek_earn+rwpred-1.96*rwrmse2+(rwrmse2^2)/2)

twoway (tsline tpa_aveweek_earn if tin(2019m1,2020m2)) ///

                    (tsline pyn ubyn lbyn if tin(2019m1,2020m3)), ///

                    title("Actual and Approx. Normal Forecast for MSA's Average Weekly
                    Earnings") ytitle("") xtitle("") legend(label(1 "Actual") label(2
                    "Forecast") ///

                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m4ynorm,
                    replace)


twoway (tsline tpa_aveweek_earn if tin(2019m1,2020m2)) ///

                    (tsline pyn ubyn lbyn if tin(2019m1,2020m3)), ///

                    title("Approximately Normal Forecast") ytitle("") xtitle("")
                    legend(label(1 "Actual") label(2 "Forecast") ///

                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m4ynorm,
                    replace)


twoway (tsline tpa_aveweek_earn if tin(2018m1,2020m2)) ///

                    (tsline pyn ubyn lbyn if tin(2019m1,2020m3)), ///

                    title("Actual and Gaussian Forecast Florida for MSA's Average
                    Weekly Earnings") ytitle("") xtitle("") legend(label(1 "Actual")
                    label(2 "Forecast") ///

                    label(3 "Upper Bound") label(4 "Lower Bound")) saving(m4ynorm2,
                    replace)
```

```
graph combine m4ynorm.gph m4yemp.gph , ///
                    saving(m4yen, replace)

*Chart one month ahead - Empirical
gen fub=ubye if tin(2020m3,)
gen flb=lbye if tin(2020m3,)
gen fcst=pye if tin(2020m3,)
replace fcst=tpa_aveweek_earn if tin(2020m2,2020m2)
replace fub=tpa_aveweek_earn if tin(2020m2,2020m2)
replace flb=tpa_aveweek_earn if tin(2020m2,2020m2)

*Chart one month ahead - Normal
twoway(tsline tpa_aveweek_earn if tin(2019m1,2020m2))(tsline fub flb fcst if
                    tin(2020m2,2020m3) ), title("Empirical Forecast") ytitle("")
                    xtitle("") legend(label(1 "Actual") label(2 "Upper Bound") ///
                    label(3 "Lower Bound") label(4 "Forecast")) saving(fcste, replace)

replace fub=ubyn if tin(2020m3,)
replace flb=lbyn if tin(2020m3,)
replace fcst=pyn if tin(2020m3,)
replace fcst=tpa_aveweek_earn if tin(2020m2,2020m2)
replace fub=tpa_aveweek_earn if tin(2020m2,2020m2)
replace flb=tpa_aveweek_earn if tin(2020m2,2020m2)

twoway(tsline tpa_aveweek_earn if tin(2019m1,2020m2))(tsline fub flb fcst if
                    tin(2020m2,2020m3) ), title("Aproximately Normal Forecast")
                    ytitle("") xtitle("") legend(label(1 "Actual") label(2 "Upper
                    Bound") ///
                    label(3 "Lower Bound") label(4 "Forecast")) saving(fcstn, replace)

graph combine fcstn.gph fcste.gph , ///
                    saving(fcts, replace)

*FAN CHART
*H=1

gen ptpae=exp((rwrmse2^2)/2)*exp(l.lntpa_aveweek_earn+rwpred)
```

```
gen ub1=exp((rwrmse2^2)/2)*exp(l.lntpa_aveweek_earn+rwpred+1*rwrmse2)

gen lb1=exp((rwrmse2^2)/2)*exp(l.lntpa_aveweek_earn+rwpred-1*rwrmse2)

gen ub2=exp((rwrmse2^2)/2)*exp(l.lntpa_aveweek_earn+rwpred+2*rwrmse2)

gen lb2=exp((rwrmse2^2)/2)*exp(l.lntpa_aveweek_earn+rwpred-2*rwrmse2)

gen ub3=exp((rwrmse2^2)/2)*exp(l.lntpa_aveweek_earn+rwpred+3*rwrmse2)

gen lb3=exp((rwrmse2^2)/2)*exp(l.lntpa_aveweek_earn+rwpred-3*rwrmse2)



*Fan Charts

twoway (tsrline ub3 ub2 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(5) lwidth(none) ) ///
                    (tsrline ub2 ub1 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(15) lwidth(none) ) ///
                    (tsrline ub1 ptpae if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(35) lwidth(none) ) ///
                    (tsrline ptpae lb1 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(35) lwidth(none) ) ///
                    (tsrline lb1 lb2 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(15) lwidth(none) ) ///
                    (tsrline lb2 lb3 if tin(2019m1,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(5) lwidth(none) ) ///
                    (tsline ptpae if tin(2019m1,2020m3) , ///
                    lcolor(gs12) lwidth(thick thick) ) ///
                    (scatter tpa_aveweek_earn date if tin(2019m1,2020m3),
                    lcolor(gs6)), ///
                    scheme(s1mono) legend(off) ///
                    title("Tampa-St.Petersburg-Clearwater"  ///
                    "Average Weekly Earnings" ///
                    "Forecast Interval") legend(off) ///
                    xtitle("") ylabel(,grid)  ///
                    note("Bands at 1, 2, and 3 sigma")

gen fptpae=tpa_aveweek_earn if tin(2020m2,2020m2)

gen fub1=tpa_aveweek_earn if tin(2020m2,2020m2)

gen fub2=tpa_aveweek_earn if tin(2020m2,2020m2)
```

```
gen fub3=tpa_aveweek_earn if tin(2020m2,2020m2)
gen flb1=tpa_aveweek_earn if tin(2020m2,2020m2)
gen flb2=tpa_aveweek_earn if tin(2020m2,2020m2)
gen flb3=tpa_aveweek_earn if tin(2020m2,2020m2)

replace fptpae=ptpae if tin(2020m3,2020m3)
replace fub1=ub1 if tin(2020m3,2020m3)
replace fub2=ub2 if tin(2020m3,2020m3)
replace fub3=ub3 if tin(2020m3,2020m3)
replace flb1=lb1 if tin(2020m3,2020m3)
replace flb2=lb2 if tin(2020m3,2020m3)
replace flb3=lb3 if tin(2020m3,2020m3)

twoway (tsrline fub3 fub2 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(5) lwidth(none) ) ///
                    (tsrline fub2 fub1 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(15) lwidth(none) ) ///
                    (tsrline fub1 fptpae if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(35) lwidth(none) ) ///
                    (tsrline fptpae flb1 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(35) lwidth(none) ) ///
                    (tsrline flb1 flb2 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(15) lwidth(none) ) ///
                    (tsrline flb2 flb3 if tin(2020m2,2020m3), ///
                    recast(rarea) fcolor(red) fintensity(5) lwidth(none) ) ///
                    (tsline fptpae if tin(2020m2,2020m3) , ///
                    lcolor(gs12) lwidth(thick thick) ) ///
                    (tsline tpa_aveweek_earn if tin(2019m1,2020m3) , ///
                    lcolor(gs6) lwidth(thick thick) ), scheme(s1mono) legend(off) ///
                    title("Tampa-St.Petersburg-Clearwater"  ///
                    "Average Weekly Earnings Forecast" ///
                    "Fan Chart for 1 Month Horizon") legend(off) ///
                    xtitle("") ylabel(,grid)  ///
                    note("Bands at 1, 2, and 3 sigma")
```

```
graph export "Fan Chart.pdf", replace
```