Final Project
Gus Lipkin
CAP 4763 Time Series Modelling and Forecasting
22 April 2021

Abstract

Time Series Modelling and Forecasting is a set of tools that allow us to use data from the past to create models of the past that can then be used to predict the future. In this paper, I use the data available for The Villages metropolitan statistical area (MSA) in Florida to try and predict the March 2021 numbers for the number of total employment for private employers and average weekly earnings. The data is gathered from the St Louis Federal Reserve Economic Data (FRED) website and analyzed with a series of Time Series tools in STATA.

Table of Contents

Introduction

In CAP 4763 Time Series Modelling and Forecasting we learned about what it takes to create a time series model and forecast one or more periods into the future using the models we've made. Our final project is the culmination of all we've learned in the class and the goal of our project is to predict two values for a single metropolitan statistical area (MSA) in Florida. I chose The Villages because of its status as the largest retirement community in the country. I thought it would be interesting to study the trends for total private employment and average weekly earnings for a community that is composed primarily of retirees. Because I knew nothing else about The Villages, I had no preconceived notions about what the data might show me or how I could best use the data available to estimate the variables in question.

Our task was to estimate the values for March 2021 which the St Louis FRED would release the real values for shortly after the paper is due. This would allow us to test our skills against real world data. FRED was a bit excited about the March data and released it ten days early which allows me to compare my forecasts against the real world data. Throughout the paper I'll discuss how this was made possible with summary statistics, auto and partial autocorrelograms, time series plots, global search regression, rolling window forecasts, and fan charts.

Data

To start, I downloaded all available data for private, non-seasonally adjusted, monthly data for all time for The Villages MSA from FRED. Data for private service employees and all private employees was available going all the way back to January 1990 and all the way through March 2021. Average hourly and weekly earnings and weekly average hours for all private employees is availble from January 2011 to March 2021. When the project started, February and March 2021 data was not available yet. Once they became available, I re-downloaded the data in TSV format from FRED.

To work with the data I first had to change the variables to something more usable with the `rename` command in STATA. Next, we had to change the data into something that is recognized as time series data. I used `generate` to create the appropriate monthly date variables and `tsset` the data so that it was recognized at starting at January 2021, the earliest datapoint in the data. Because I am predicting March 2021 and can't predict it with the data, I created a set of dummy columns for total private employment and weekly earnings that had the March data and dropped the March data for the main data columns. This will allow me to compare my forecast to the actual value.

Performing log transforms on all the variables was the next data manipulation step. Log transforms serve two purposes. The first is to normalize the data so that it forms a bell curve. The second is to force all models and forecasts to only produce positive values because unless something has gone very very wrong, employment and weekly earnings will both always be positive.

The number of total employment and service employment are both in the thousands of numbers while the number of hours is in hours and teh number of earnings is in dollars.

Summary Statistics

Calculating summary statistics for all variables can provide some valuable base insights that I can use to help shape my models.

Table 1 Summary Statistics for All Standard Variables

Variable	Obs	Mean	Std. Dev.	Min	Max
Count	374	14.18556	6.880684	5.3	28
WeekHours	123	36.88455	3.791817	28.3	45.8
HourlyEarnings	123	19.72	2.903968	15.01	24.6
WeeklyEarnings	122	719.6542	84.57241	503.79	916.1
ServiceCount	375	10.43387	5.959179	3.9	22.8

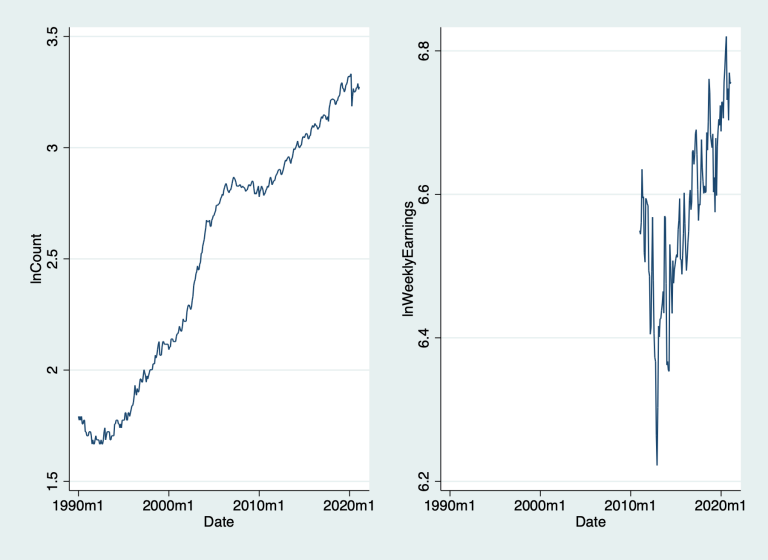
Of particular note in the non-transformed dat is that there is a wide variation in the number of private employment for The Villages has an ancredibly high range where the maximum value is more than five times greater than the minimum value. Meanwhile, hourly earnings have not increased by nearly the same factor. This indicates that The Villages are growing and supply is increasing to meet rising demand from continued development in The Villages MSA.

Table 2 Summary Statistics for All Log Transformed Variables

Variable	Obs	Mean	Std. Dev.	Min	Max
lnCount	374	2.5174	.5398403	1.667707	3.332205
lnWeekHours	123	3.602488	.10385	3.342862	3.824284
lnHourlyEarnings	123	2.970779	.1482819	2.708717	3.202746
lnWeeklyEarnings	122	6.571775	.1195694	6.222159	6.820126
lnServiceCount	375	2.172053	.5985689	1.360977	3.12676

To a more trained eye, the log transformed data may provide more insights, but I am not that good.

Figure 1 Time Series Plots of lnCount and lnWeeklyEarnings



The first step after creating summary statistics is to generate a time series line plot so we can see how the data is changing over time and give context to the summary statistics. On the left, we see the log transform of the number of private employment in The Villages. There is a solid upwards trend except for the market crashes in the early 1990s, late 2000s, and in 2020. Looking at lnWeeklyEarnings on the right, the data does not start until 2011 but the graph has been adjusted to show the same time period as lnCount. The data sharply decreases around 2012 and then slowly rises on average with large differences in month to month data. This could be due to The Villages being populated by old people who travel to Florida from their home state for the winter months, also known as "snow-birding."

Next, we want to make sure the data is stationary. Stationary data has a constant mean, variance, and autocorrelation through time which makes the data easier to work with, especially in time series modelling and forecasting. To solve this, we difference the data. Differencing is when you find the difference between each datapoint and the previous one. We can check to see if we need to difference our data by graphing the autocorrelogram and partial autocorrelogram. Below are the AC and PAC for lnCount and lnWeekly Earnings.

Figure 2 AC and PAC of lnCount

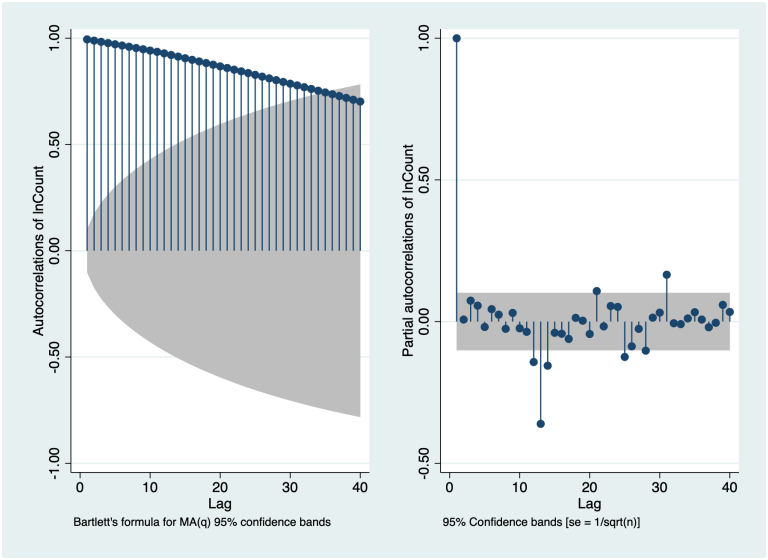
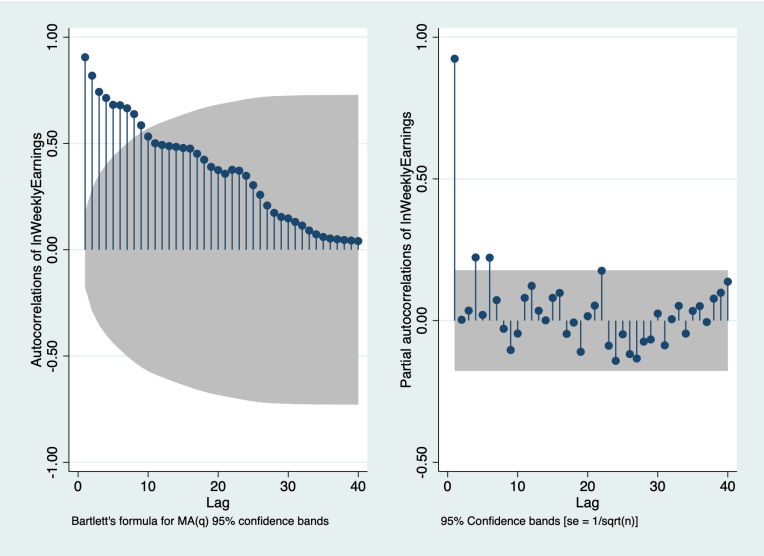


Figure 3 AC and PAC of lnWeeklyEarnings



Both AC charts show a high first term and then steadily decreasing terms thereafter which suggests that there is an autoregressive term in the data. Autoregressive data is data that is dependent, at least partially, on the data before it. Recursive formulas in geometric sequences are a good example. The PAC shows a significant value and then alternating positive and negative insignificant values, characteristic of a higher order moving average term. That is to say, the data has trends that repeat. In this case, it is likely seasonal trends such as increased employees during the holidays.

Model Estimation and Selection

Model estimation and selection is the keystone in time series modeling and forecasting. No matter how well prepared you are, if you cannot construct a good model, then all of your labor will be for nothing. Using a combination of intuition, global search regression, and rolling window forecasts, I was able to choose between a few different models that I thought would accurately model private employment and weekly earnings.

Model Estimation and Selection Total Private Employment

Before diving in with more advanced techniques, I devised a few models that I thought might work well based on the time series plots, AC, and PAC that I generated earlier. My first model regressed the differenced value of `lnCount` against the twelfth, twenty-fourth, thirty-sixth, and forty-eigth lags of the differenced `lnCount`. The lags mean that the independent variables are `lnCount` from those many months ago each to create an autoregressive model. The final model used in STATA was `reg d.lnCount l(12,24,36,48)d.lnCount`. When run through a rolling window forecast to find the optimal window size, I found that this inital model had an optimal window size of 132 months which resulted in a rolling window root mean square error of 0.0172128.

My next intuitive model for `lnCount` uses the fifth, twelfth, twenty-fourth, thirty-sixth, and forty-eighth lags of differenced `lnCount` and the fifth lag of different average weekly hours along with an indicator variable for May. The final regression in stata was `reg d.lnCount l(5,12,24,36,48)d.lnCount l(5)d.lnWeekHours m5`. With an optimal window size of eighty-four periods, I had a rolling window root mean square error of 0.01950911.

Begrudgingly, I moved on from my intuition to global search regression which allows me to input many different variables and run regressions on all combinations of the given variables. Again, I used the differenced `lnCount` as the dependent variable and the first twelve lags of the differenced `lnCount` along with the differenced twenty-fourth, thirty-sixth, and forty-eighth lags of `lnCount`. I chose to only use lagged versions of `lnCount` because service employment is just a subset of the total employment. The other variables I chose not to include because I wanted to make sure the search did not take too long to run and because the AC and PAC indicated high amounts of autoregression. I also fixed all month indicators.

Table 3 GSREG Models and Results for Total Private Employment

Model	Windows	RWRMSE
`reg d.lnCount l12d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	144	0.01824906
`reg d.lnCount l(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	144	0.01777071
`reg d.lnCount l4d.lnWeekHours l9d.lnWeekHours l8d.lnHourlyEarnings m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	12	0.0176238

Model Estimation and Selection Average Weekly Earnings

When modeling average weekly earnings, I used similar methods to total private employment. In a change of pace, however, I only guesstimated one model. I regressed the differenced `lnWeeklyEarnings` against the lagged and differenced `lnWeekHours` and `lnHourlyEarnings`. I was hoping that the combination of hours per week and earnings per hour would accurately predict the current weekly earnings. This could have worked better if I had weekly data but I do the best that I can. After I can the rolling window estimate, I found that the optimal window size was sixty months with a RWRMSE of 0.06145693.

For my global search regression, I used the same lags as with `lnCount` but instead did not use the forty-eighth lag for `lnWeeklyEarnings`. Even though I had the opportunity to use `lnWeekHours` and `lnHourlyEarnings` together as I suggested earlier, I chose to work with an autoregressive model again because as I also said before, the data is not weekly which makes those numbers a bit less valuable.

Table 4 GSREG Models and Results for Average Weekly Earnings

Model	Windows	RWRMSE
`reg d.lnWeeklyEarnings l3d.lnWeeklyEarnings l5d.lnWeeklyEarnings m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	84	0.06004448
`reg d.lnWeeklyEarnings l3d.lnWeeklyEarnings l5d.lnWeeklyEarnings l7d.lnWeeklyEarnings`	84	0.05250414

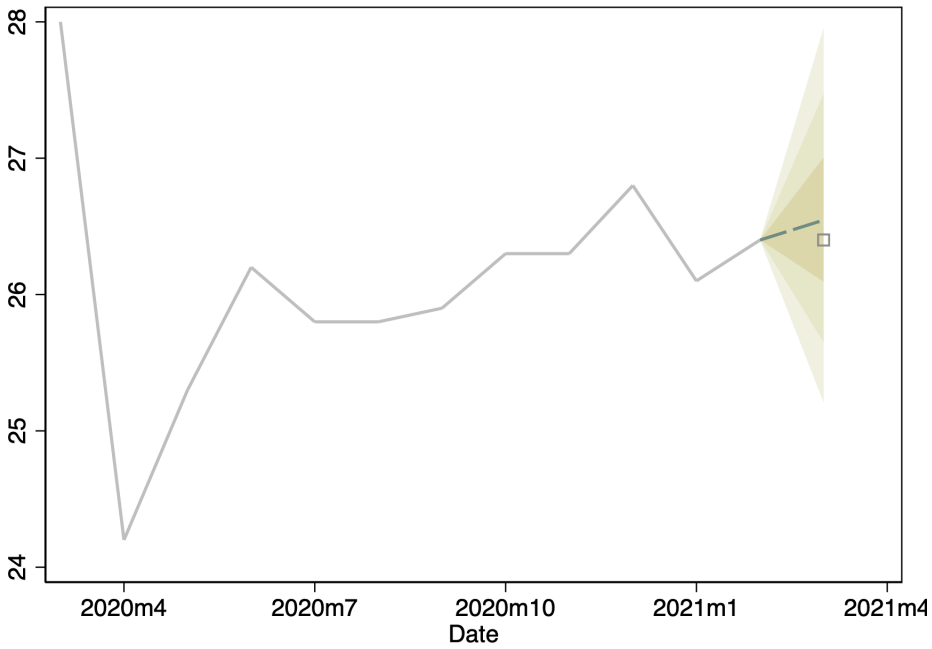
Final Results

This year is especially exciting to be a student of the magnificent Dr Dewey because I can now compare my models to the real value. When selecting the models, I did not use the March data available to me because it would cloud the models and of course I can predict March's values if I know what March is.

Final Results Total Private Employment

FRED says that the total private employment for March 2021 is $26,400$. My best model `reg d.lnCount l(12,24,36,48)d.lnCount`, predicted that the value would be $26,545$ which is $.5\%$ higher than the real value. With real world context added, it is possible that the number is lower than I expected because many people received a third COVID stimulus check in March and some people used that opportunity to leave their jobs and begin a search for a new job that is better for them.

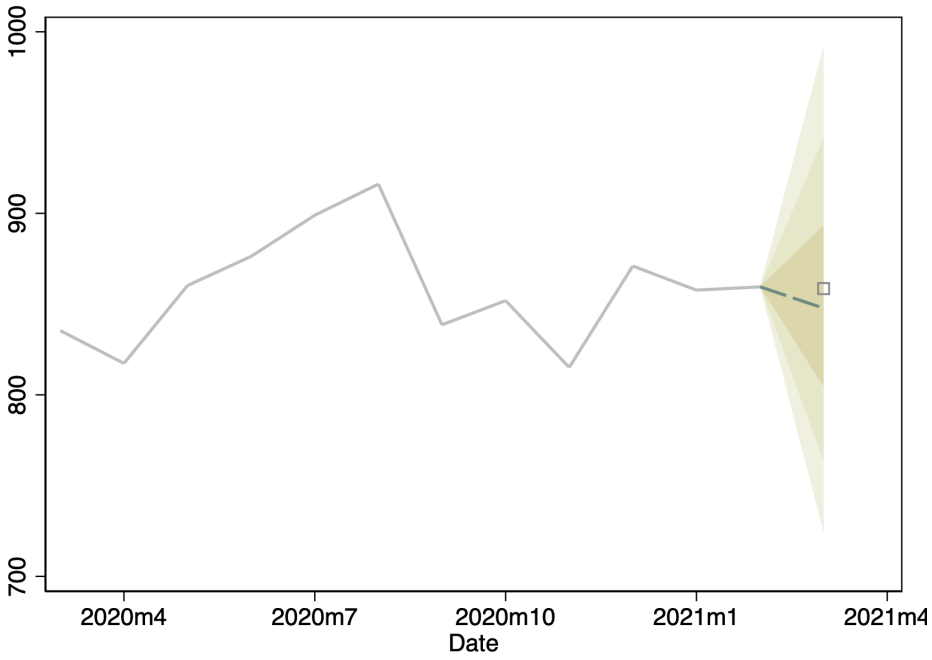
Figure 4 Fan Chart of Total Private Employment



Final Results Average Weekly Earnings

For average weekly earnings, FRED reported that in March 2021, the mean weekly income in The Villages was $\$858.52$. My estimate with `reg d.lnWeeklyEarnings l(3,5,7)d.lnWeeklyEarnings` was $\$847.7501$ which is a difference of 1.25%. I believe my autoregressive model was a poor choice because here it is clearly influenced by a dip in weekly earnings in the last six months.

Figure 5 Fan Chart of Average Weekly Earnings



Conclusion

Time series modelling and forecasting is a complicated art but not nearly as complicated as it may seem initially. After a brief examination of the data with summary statistics, I examined the data visually with time series plots, autocorrelograms, and partial autocorrelograms. The ACs and PACs indicated that the data was not stationary and needed to be differenced. Next, using a mix of guesstimation and global search regression, I identified several models for each variable that I wanted to forecast. I ran each model through a rolling window forecast to identify the optimal window width and then used that to forecast the values for March 2021. While my forecasts weren't perfect, they were pretty good and well within the 95% confidence interval of the fan chart. If I were to improve the models for the future, I would introduce new variables to the models so that they weren't purely autoregressive.

Appendix A

```
clear
set more off

cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Final Project"
log using "Final Project.smcl", replace
import delimited "TS2020_Final_Project_Monthly.txt"
rename smu12455400500000001 Count
rename smu12455400500000002 WeekHours
rename smu12455400500000003 HourlyEarnings
rename smu12455400500000011 WeeklyEarnings
rename smu12455400800000001 ServiceCount


label variable Count "Count"
```

```
label variable WeekHours "WeekHours"
label variable HourlyEarnings "HourlyEarnings"
label variable WeeklyEarnings "WeeklyEarnings"
label variable ServiceCount "ServiceCount"


gen datec=date(date, "YMD")
gen Date=mofd(datec)
gen month=month(datec)
format Date %tm
tsset Date

gen withMarchCount = Count
gen withMarchEarnings = WeeklyEarnings
replace Count=. if tin(2021m3,)
replace WeeklyEarnings=. if tin(2021m3,)

gen lnCount = ln(Count)
gen lnWeekHours = ln(WeekHours)
gen lnHourlyEarnings = ln(HourlyEarnings)
gen lnWeeklyEarnings = ln(WeeklyEarnings)
gen lnServiceCount = ln(ServiceCount)

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

gen dlnCount=d.lnCount
gen l1dlnCount=l1d.lnCount
gen l2dlnCount=l2d.lnCount
```

```
gen l3dlnCount=l3d.lnCount
gen l4dlnCount=l4d.lnCount
gen l5dlnCount=l5d.lnCount
gen l6dlnCount=l6d.lnCount
gen l7dlnCount=l7d.lnCount
gen l8dlnCount=l8d.lnCount
gen l9dlnCount=l9d.lnCount
gen l10dlnCount=l10d.lnCount
gen l11dlnCount=l11d.lnCount
gen l12dlnCount=l12d.lnCount
gen l24dlnCount=l24d.lnCount
gen l36dlnCount=l36d.lnCount
gen l48dlnCount=l48d.lnCount

gen dlnWeekHours=d.lnWeekHours
gen l1dlnWeekHours=l1d.lnWeekHours
gen l2dlnWeekHours=l2d.lnWeekHours
gen l3dlnWeekHours=l3d.lnWeekHours
gen l4dlnWeekHours=l4d.lnWeekHours
gen l5dlnWeekHours=l5d.lnWeekHours
gen l6dlnWeekHours=l6d.lnWeekHours
gen l7dlnWeekHours=l7d.lnWeekHours
gen l8dlnWeekHours=l8d.lnWeekHours
gen l9dlnWeekHours=l9d.lnWeekHours
gen l10dlnWeekHours=l10d.lnWeekHours
gen l11dlnWeekHours=l11d.lnWeekHours
gen l12dlnWeekHours=l12d.lnWeekHours
gen l24dlnWeekHours=l24d.lnWeekHours
gen l36dlnWeekHours=l36d.lnWeekHours
gen l48dlnWeekHours=l48d.lnWeekHours

gen dlnHourlyEarnings=d.lnHourlyEarnings
gen l1dlnHourlyEarnings=l1d.lnHourlyEarnings
gen l2dlnHourlyEarnings=l2d.lnHourlyEarnings
gen l3dlnHourlyEarnings=l3d.lnHourlyEarnings
gen l4dlnHourlyEarnings=l4d.lnHourlyEarnings
gen l5dlnHourlyEarnings=l5d.lnHourlyEarnings
gen l6dlnHourlyEarnings=l6d.lnHourlyEarnings
gen l7dlnHourlyEarnings=l7d.lnHourlyEarnings
gen l8dlnHourlyEarnings=l8d.lnHourlyEarnings
gen l9dlnHourlyEarnings=l9d.lnHourlyEarnings
gen l10dlnHourlyEarnings=l10d.lnHourlyEarnings
gen l11dlnHourlyEarnings=l11d.lnHourlyEarnings
gen l12dlnHourlyEarnings=l12d.lnHourlyEarnings
gen l24dlnHourlyEarnings=l24d.lnHourlyEarnings
gen l36dlnHourlyEarnings=l36d.lnHourlyEarnings
gen l48dlnHourlyEarnings=l48d.lnHourlyEarnings

gen dlnWeeklyEarnings=d.lnWeeklyEarnings
gen l1dlnWeeklyEarnings=l1d.lnWeeklyEarnings
gen l2dlnWeeklyEarnings=l2d.lnWeeklyEarnings
```

```
gen l3dlnWeeklyEarnings=l3d.lnWeeklyEarnings
gen l4dlnWeeklyEarnings=l4d.lnWeeklyEarnings
gen l5dlnWeeklyEarnings=l5d.lnWeeklyEarnings
gen l6dlnWeeklyEarnings=l6d.lnWeeklyEarnings
gen l7dlnWeeklyEarnings=l7d.lnWeeklyEarnings
gen l8dlnWeeklyEarnings=l8d.lnWeeklyEarnings
gen l9dlnWeeklyEarnings=l9d.lnWeeklyEarnings
gen l10dlnWeeklyEarnings=l10d.lnWeeklyEarnings
gen l11dlnWeeklyEarnings=l11d.lnWeeklyEarnings
gen l12dlnWeeklyEarnings=l12d.lnWeeklyEarnings
gen l24dlnWeeklyEarnings=l24d.lnWeeklyEarnings
gen l36dlnWeeklyEarnings=l36d.lnWeeklyEarnings
gen l48dlnWeeklyEarnings=l48d.lnWeeklyEarnings

gen dlnServiceCount=d.lnServiceCount
gen l1dlnServiceCount=l1d.lnServiceCount
gen l2dlnServiceCount=l2d.lnServiceCount
gen l3dlnServiceCount=l3d.lnServiceCount
gen l4dlnServiceCount=l4d.lnServiceCount
gen l5dlnServiceCount=l5d.lnServiceCount
gen l6dlnServiceCount=l6d.lnServiceCount
gen l7dlnServiceCount=l7d.lnServiceCount
gen l8dlnServiceCount=l8d.lnServiceCount
gen l9dlnServiceCount=l9d.lnServiceCount
gen l10dlnServiceCount=l10d.lnServiceCount
gen l11dlnServiceCount=l11d.lnServiceCount
gen l12dlnServiceCount=l12d.lnServiceCount
gen l24dlnServiceCount=l24d.lnServiceCount
gen l36dlnServiceCount=l36d.lnServiceCount
gen l48dlnServiceCount=l48d.lnServiceCount

/*
The project is to forecast the March non-seasonally adjusted estimates of average
weekly earnings and total employment for private employers (total private) for a
Florida MSA of your choice and write up a professional report on your forecast.
*/
/* Count and WeeklyEarnings */

summ Count WeekHours HourlyEarnings WeeklyEarnings ServiceCount
summ lnCount lnWeekHours lnHourlyEarnings lnWeeklyEarnings lnServiceCount

ac lnCount, saving(lnCount_ac, replace)
pac lnCount, saving(lnCount_pac, replace)
graph combine lnCount_ac.gph lnCount_pac.gph, saving(lnCount_ac_pac, replace)
graph export "lnCount_ac_pac.png", replace
** Probably need to difference

ac lnWeeklyEarnings, saving(lnWeeklyEarnings_ac, replace)
pac lnWeeklyEarnings, saving(lnWeeklyEarnings_pac, replace)
graph combine lnWeeklyEarnings_ac.gph lnWeeklyEarnings_pac.gph,
saving(lnWeeklyEarnings_ac_pac, replace)
```

```
graph export "lnWeeklyEarnings_ac_pac.png", replace
** Probably need to difference

*starter models for count
*I used a pair plot to examine the rise and fall of variables with respect to each other
reg d.lnCount l(12,24,36,48)d.lnCount // .01637
scalar drop _all
quietly forval w=12(12)144 {
gen pred=.
gen nobs=.
  forval t=421/733 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnCount l12dlnCount l24dlnCount l36dlnCount l48dlnCount ///
    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.lnCount)^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
/*
RWmaxobs132 =        132
RWminobs132 =         12
RWrmse132 =   .0172128
*/

reg d.lnCount l(5,12,24,36,48)d.lnCount l(5)d.lnWeekHours m5 // .01711
scalar drop _all
quietly forval w=12(12)84 {
gen pred=.
gen nobs=.
  forval t=641/733 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnCount l5dlnCount l12dlnCount l24dlnCount l36dlnCount l48dlnCount l5dlnWeekHours m5 ///
    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.lnCount)^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
/*
RWmaxobs84 =         84
RWminobs84 =         23
RWrmse84 =  .01950911
*/

quietly gsreg dlnCount l1dlnCount l2dlnCount l3dlnCount l4dlnCount l5dlnCount
l6dlnCount ///
  l7dlnCount l8dlnCount l9dlnCount l10dlnCount l11dlnCount l12dlnCount ///
  l24dlnCount l36dlnCount l48dlnCount ///
  if tin(1990m1,2021m1), ///
  ncomb(1,12) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12) ///
  samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnCount) replace

*gsreg suggestions
reg d.lnCount l12d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
scalar drop _all
quietly forval w=12(12)144 {
gen pred=.
gen nobs=.
  forval t=385/733 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnCount l12dlnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
    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.lnCount)^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
/*
RWmaxobs144 =        144
```

```
RWminobs144 =         12
RWrmse144 =  .01824906
*/

reg d.lnCount l(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
scalar drop _all
quietly forval w=12(12)144 {
gen pred=.
gen nobs=.
  forval t=409/733 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnCount l12dlnCount l36dlnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
    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.lnCount)^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
/*
RWmaxobs144 =        144
RWminobs144 =         12
RWrmse144 =  .01777071
*/

reg d.lnCount l4d.lnWeekHours l9d.lnWeekHours l8d.lnHourlyEarnings m1 m2 m3 m4 m5 m6
m7 m8 m9 m10 m11
scalar drop _all
quietly forval w=12(12)84 {
gen pred=.
gen nobs=.
  forval t=624/733 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnCount l4dlnWeekHours l9dlnWeekHours l8dlnHourlyEarnings m1 m2 m3 m4 m5 m6
m7 m8 m9 m10 m11 ///
    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.lnCount)^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
/*
RWmaxobs12 =         12
RWminobs12 =          2
RWrmse12 =   .0176238
*/

scalar rwrmse = .0172128
reg d.lnCount l(12,24,36,48)d.lnCount if tin(,2021m2)
predict pd
gen pflcount=exp((rwrmse^2)/2)*exp(l.lnCount+pd) if Date==tm(2021m3)
gen ub1=exp((rwrmse^2)/2)*exp(l.lnCount+pd+1*rwrmse) if Date==tm(2021m3)
gen lb1=exp((rwrmse^2)/2)*exp(l.lnCount+pd-1*rwrmse) if Date==tm(2021m3)
gen ub2=exp((rwrmse^2)/2)*exp(l.lnCount+pd+2*rwrmse) if Date==tm(2021m3)
gen lb2=exp((rwrmse^2)/2)*exp(l.lnCount+pd-2*rwrmse) if Date==tm(2021m3)
gen ub3=exp((rwrmse^2)/2)*exp(l.lnCount+pd+3*rwrmse) if Date==tm(2021m3)
gen lb3=exp((rwrmse^2)/2)*exp(l.lnCount+pd-3*rwrmse) if Date==tm(2021m3)
drop pd

replace pflcount=Count if Date==tm(2021m2)
replace ub1=Count if Date==tm(2021m2)
replace ub2=Count if Date==tm(2021m2)
replace ub3=Count if Date==tm(2021m2)
replace lb1=Count if Date==tm(2021m2)
replace lb2=Count if Date==tm(2021m2)
replace lb3=Count if Date==tm(2021m2)

twoway (tsrline ub3 ub2 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
  (tsrline ub2 ub1 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
  (tsrline ub1 pflcount if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
  (tsrline pflcount lb1 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
  (tsrline lb1 lb2 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
  (tsrline lb2 lb3 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
  (tsline Count pflcount if tin(2020m3,2021m3) , ///
  lcolor(gs12 teal) lwidth(medthick medthick) ///
  lpattern(solid longdash)) ///
  (scatter withMarchCount Date if tin(2021m3,)), scheme(s1mono) legend(off)
```

```
graph export "CountFan.png", replace

/*-------------------------------------------------------------------------------*/

*starter models for weekly earnings
reg d.lnWeeklyEarnings l1d.lnWeekHours ld.lnHourlyEarnings
scalar drop _all
quietly forval w=12(12)84 {
gen pred=.
gen nobs=.
  forval t=616/733 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnWeeklyEarnings l1dlnWeekHours l1dlnHourlyEarnings ///
    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.lnWeeklyEarnings)^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
/*
RWmaxobs60 =         60
RWminobs60 =          2
RWrmse60 =  .06145693
*/

quietly gsreg dlnWeeklyEarnings l1dlnWeeklyEarnings l2dlnWeeklyEarnings
l3dlnWeeklyEarnings ///
  l4dlnWeeklyEarnings l5dlnWeeklyEarnings l6dlnWeeklyEarnings ///
  l7dlnWeeklyEarnings l8dlnWeeklyEarnings l9dlnWeeklyEarnings l10dlnWeeklyEarnings
///
  l11dlnWeeklyEarnings l12dlnWeeklyEarnings ///
  l24dlnWeeklyEarnings l36dlnWeeklyEarnings ///
  if tin(2011m1,2021m1), ///
  ncomb(1,12) aic outsample(24) ///
  samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnWeeklyEarnings)
replace

reg d.lnWeeklyEarnings l3d.lnWeeklyEarnings l5d.lnWeeklyEarnings m1 m2 m3 m4 m5 m6
m7 m8 m9 m10 m11
scalar drop _all
quietly forval w=12(12)84 {
```

```
gen pred=.
gen nobs=.
  forval t=620/733 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnWeeklyEarnings l3dlnWeeklyEarnings l5dlnWeeklyEarnings m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
    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.lnWeeklyEarnings)^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
/*
RWmaxobs84 =         84
RWminobs84 =          2
RWrmse84 =  .06004448
*/

reg d.lnWeeklyEarnings l3d.lnWeeklyEarnings l5d.lnWeeklyEarnings l7d.lnWeeklyEarnings
scalar drop _all
quietly forval w=12(12)84 {
gen pred=.
gen nobs=.
  forval t=622/733 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnWeeklyEarnings l3dlnWeeklyEarnings l5dlnWeeklyEarnings l7dlnWeeklyEarnings ///
    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.lnWeeklyEarnings)^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
/*
RWmaxobs84 =         84
RWminobs84 =          2
RWrmse84 =  .05250414
*/

drop pflcount ub1 ub2 ub3 lb1 lb2 lb3

scalar rwrmse = .05250414
reg d.lnWeeklyEarnings l(3,5,7)d.lnWeeklyEarnings if tin(,2021m2)
predict pd
gen pflcount=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd) if Date==tm(2021m3)
gen ub1=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+1*rwrmse) if Date==tm(2021m3)
gen lb1=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-1*rwrmse) if Date==tm(2021m3)
gen ub2=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+2*rwrmse) if Date==tm(2021m3)
gen lb2=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-2*rwrmse) if Date==tm(2021m3)
gen ub3=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+3*rwrmse) if Date==tm(2021m3)
gen lb3=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-3*rwrmse) if Date==tm(2021m3)
drop pd

replace pflcount=WeeklyEarnings if Date==tm(2021m2)
replace ub1=WeeklyEarnings if Date==tm(2021m2)
replace ub2=WeeklyEarnings if Date==tm(2021m2)
replace ub3=WeeklyEarnings if Date==tm(2021m2)
replace lb1=WeeklyEarnings if Date==tm(2021m2)
replace lb2=WeeklyEarnings if Date==tm(2021m2)
replace lb3=WeeklyEarnings if Date==tm(2021m2)

twoway (tsrline ub3 ub2 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
  (tsrline ub2 ub1 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
  (tsrline ub1 pflcount if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
  (tsrline pflcount lb1 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
  (tsrline lb1 lb2 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
  (tsrline lb2 lb3 if tin(2020m3,2021m3), ///
  recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
  (tsline WeeklyEarnings pflcount if tin(2020m3,2021m3) , ///
  lcolor(gs12 teal) lwidth(medthick medthick) ///
  lpattern(solid longdash)) ///
  (scatter withMarchEarnings Date if tin(2021m3,)), scheme(s1mono) legend(off)
graph export "WeeklyFan.png", replace

log close
translate "Final Project.smcl" "Final Project.txt", replace
```

```
510
```

Appendix B

```
                                                      ___  ____  ____  ____
____(R)
                                                     /__    /   ____/   /   ____/
                                                    ___/   /   /___/   /   /___/
                                                      Statistics/Data analysis


      ----------------------------------------------------------------------------
--
             name:  <unnamed>
              log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Probl
      > em Sets/Final Project/Final Project.smcl
        log type:  smcl
       opened on:  19 Apr 2021, 20:43:14

     1 . import delimited "TS2020_Final_Project_Monthly.txt"
      (6 vars, 375 obs)

     2 . rename smu12455400500000001 Count

     3 . rename smu12455400500000002 WeekHours

     4 . rename smu12455400500000003 HourlyEarnings

     5 . rename smu12455400500000011 WeeklyEarnings

     6 . rename smu12455400800000001 ServiceCount

     7 .
     8 .
     9 . label variable Count "Count"

    10 . label variable WeekHours "WeekHours"

    11 . label variable HourlyEarnings "HourlyEarnings"

    12 . label variable WeeklyEarnings "WeeklyEarnings"

    13 . label variable ServiceCount "ServiceCount"

    14 .
    15 .
```

```
. gen datec=date(date, "YMD")

. gen Date=mofd(datec)

. gen month=month(datec)

. format Date %tm

. tsset Date
           time variable:  Date, 1990m1 to 2021m3
                   delta:  1 month

.
. gen withMarchCount = Count

. gen withMarchEarnings = WeeklyEarnings
  (252 missing values generated)

. replace Count=. if tin(2021m3,)
  (1 real change made, 1 to missing)

. replace WeeklyEarnings=. if tin(2021m3,)
  (1 real change made, 1 to missing)

.
. gen lnCount = ln(Count)
  (1 missing value generated)

. gen lnWeekHours = ln(WeekHours)
  (252 missing values generated)

. gen lnHourlyEarnings = ln(HourlyEarnings)
  (252 missing values generated)

. gen lnWeeklyEarnings = ln(WeeklyEarnings)
  (253 missing values generated)

. gen lnServiceCount = ln(ServiceCount)

.
. gen m1=0

. replace m1=1 if month==1
  (32 real changes made)

. gen m2=0

. replace m2=1 if month==2
  (32 real changes made)

. gen m3=0
```

```
91
92      38 . replace m3=1 if month==3
93        (32 real changes made)
94
95      39 . gen m4=0
96
97      40 . replace m4=1 if month==4
98        (31 real changes made)
99
100     41 . gen m5=0
101
102     42 . replace m5=1 if month==5
103       (31 real changes made)
104
105     43 . gen m6=0
106
107     44 . replace m6=1 if month==6
108       (31 real changes made)
109
110     45 . gen m7=0
111
112     46 . replace m7=1 if month==7
113       (31 real changes made)
114
115     47 . gen m8=0
116
117     48 . replace m8=1 if month==8
118       (31 real changes made)
119
120     49 . gen m9=0
121
122     50 . replace m9=1 if month==9
123       (31 real changes made)
124
125     51 . gen m10=0
126
127     52 . replace m10=1 if month==10
128       (31 real changes made)
129
130     53 . gen m11=0
131
132     54 . replace m11=1 if month==11
133       (31 real changes made)
134
135     55 . gen m12=0
136
137     56 . replace m12=1 if month==12
138       (31 real changes made)
139
140     57 .
141     58 . gen dlnCount=d.lnCount
```

```
  (2 missing values generated)

59 . gen l1dlnCount=l1d.lnCount
  (2 missing values generated)

60 . gen l2dlnCount=l2d.lnCount
  (3 missing values generated)

61 . gen l3dlnCount=l3d.lnCount
  (4 missing values generated)

62 . gen l4dlnCount=l4d.lnCount
  (5 missing values generated)

63 . gen l5dlnCount=l5d.lnCount
  (6 missing values generated)

64 . gen l6dlnCount=l6d.lnCount
  (7 missing values generated)

65 . gen l7dlnCount=l7d.lnCount
  (8 missing values generated)

66 . gen l8dlnCount=l8d.lnCount
  (9 missing values generated)

67 . gen l9dlnCount=l9d.lnCount
  (10 missing values generated)

68 . gen l10dlnCount=l10d.lnCount
  (11 missing values generated)

69 . gen l11dlnCount=l11d.lnCount
  (12 missing values generated)

70 . gen l12dlnCount=l12d.lnCount
  (13 missing values generated)

71 . gen l24dlnCount=l24d.lnCount
  (25 missing values generated)

72 . gen l36dlnCount=l36d.lnCount
  (37 missing values generated)

73 . gen l48dlnCount=l48d.lnCount
  (49 missing values generated)

74 .
75 . gen dlnWeekHours=d.lnWeekHours
  (253 missing values generated)

```

```
. gen l1dlnWeekHours=l1d.lnWeekHours
(254 missing values generated)

. gen l2dlnWeekHours=l2d.lnWeekHours
(255 missing values generated)

. gen l3dlnWeekHours=l3d.lnWeekHours
(256 missing values generated)

. gen l4dlnWeekHours=l4d.lnWeekHours
(257 missing values generated)

. gen l5dlnWeekHours=l5d.lnWeekHours
(258 missing values generated)

. gen l6dlnWeekHours=l6d.lnWeekHours
(259 missing values generated)

. gen l7dlnWeekHours=l7d.lnWeekHours
(260 missing values generated)

. gen l8dlnWeekHours=l8d.lnWeekHours
(261 missing values generated)

. gen l9dlnWeekHours=l9d.lnWeekHours
(262 missing values generated)

. gen l10dlnWeekHours=l10d.lnWeekHours
(263 missing values generated)

. gen l11dlnWeekHours=l11d.lnWeekHours
(264 missing values generated)

. gen l12dlnWeekHours=l12d.lnWeekHours
(265 missing values generated)

. gen l24dlnWeekHours=l24d.lnWeekHours
(277 missing values generated)

. gen l36dlnWeekHours=l36d.lnWeekHours
(289 missing values generated)

. gen l48dlnWeekHours=l48d.lnWeekHours
(301 missing values generated)

.
. gen dlnHourlyEarnings=d.lnHourlyEarnings
(253 missing values generated)

. gen l1dlnHourlyEarnings=l1d.lnHourlyEarnings
(254 missing values generated)
```

```
 94 . gen l2dlnHourlyEarnings=l2d.lnHourlyEarnings
   (255 missing values generated)

 95 . gen l3dlnHourlyEarnings=l3d.lnHourlyEarnings
   (256 missing values generated)

 96 . gen l4dlnHourlyEarnings=l4d.lnHourlyEarnings
   (257 missing values generated)

 97 . gen l5dlnHourlyEarnings=l5d.lnHourlyEarnings
   (258 missing values generated)

 98 . gen l6dlnHourlyEarnings=l6d.lnHourlyEarnings
   (259 missing values generated)

 99 . gen l7dlnHourlyEarnings=l7d.lnHourlyEarnings
   (260 missing values generated)

100 . gen l8dlnHourlyEarnings=l8d.lnHourlyEarnings
   (261 missing values generated)

101 . gen l9dlnHourlyEarnings=l9d.lnHourlyEarnings
   (262 missing values generated)

102 . gen l10dlnHourlyEarnings=l10d.lnHourlyEarnings
   (263 missing values generated)

103 . gen l11dlnHourlyEarnings=l11d.lnHourlyEarnings
   (264 missing values generated)

104 . gen l12dlnHourlyEarnings=l12d.lnHourlyEarnings
   (265 missing values generated)

105 . gen l24dlnHourlyEarnings=l24d.lnHourlyEarnings
   (277 missing values generated)

106 . gen l36dlnHourlyEarnings=l36d.lnHourlyEarnings
   (289 missing values generated)

107 . gen l48dlnHourlyEarnings=l48d.lnHourlyEarnings
   (301 missing values generated)

108 .
109 . gen dlnWeeklyEarnings=d.lnWeeklyEarnings
   (254 missing values generated)

110 . gen l1dlnWeeklyEarnings=l1d.lnWeeklyEarnings
   (254 missing values generated)

111 . gen l2dlnWeeklyEarnings=l2d.lnWeeklyEarnings
```

```
      (255 missing values generated)

. gen l3dlnWeeklyEarnings=l3d.lnWeeklyEarnings
      (256 missing values generated)

. gen l4dlnWeeklyEarnings=l4d.lnWeeklyEarnings
      (257 missing values generated)

. gen l5dlnWeeklyEarnings=l5d.lnWeeklyEarnings
      (258 missing values generated)

. gen l6dlnWeeklyEarnings=l6d.lnWeeklyEarnings
      (259 missing values generated)

. gen l7dlnWeeklyEarnings=l7d.lnWeeklyEarnings
      (260 missing values generated)

. gen l8dlnWeeklyEarnings=l8d.lnWeeklyEarnings
      (261 missing values generated)

. gen l9dlnWeeklyEarnings=l9d.lnWeeklyEarnings
      (262 missing values generated)

. gen l10dlnWeeklyEarnings=l10d.lnWeeklyEarnings
      (263 missing values generated)

. gen l11dlnWeeklyEarnings=l11d.lnWeeklyEarnings
      (264 missing values generated)

. gen l12dlnWeeklyEarnings=l12d.lnWeeklyEarnings
      (265 missing values generated)

. gen l24dlnWeeklyEarnings=l24d.lnWeeklyEarnings
      (277 missing values generated)

. gen l36dlnWeeklyEarnings=l36d.lnWeeklyEarnings
      (289 missing values generated)

. gen l48dlnWeeklyEarnings=l48d.lnWeeklyEarnings
      (301 missing values generated)

.
. gen dlnServiceCount=d.lnServiceCount
      (1 missing value generated)

. gen l1dlnServiceCount=l1d.lnServiceCount
      (2 missing values generated)

. gen l2dlnServiceCount=l2d.lnServiceCount
      (3 missing values generated)

```

```
  129 . gen l3dlnServiceCount=l3d.lnServiceCount
     (4 missing values generated)

  130 . gen l4dlnServiceCount=l4d.lnServiceCount
     (5 missing values generated)

  131 . gen l5dlnServiceCount=l5d.lnServiceCount
     (6 missing values generated)

  132 . gen l6dlnServiceCount=l6d.lnServiceCount
     (7 missing values generated)

  133 . gen l7dlnServiceCount=l7d.lnServiceCount
     (8 missing values generated)

  134 . gen l8dlnServiceCount=l8d.lnServiceCount
     (9 missing values generated)

  135 . gen l9dlnServiceCount=l9d.lnServiceCount
     (10 missing values generated)

  136 . gen l10dlnServiceCount=l10d.lnServiceCount
     (11 missing values generated)

  137 . gen l11dlnServiceCount=l11d.lnServiceCount
     (12 missing values generated)

  138 . gen l12dlnServiceCount=l12d.lnServiceCount
     (13 missing values generated)

  139 . gen l24dlnServiceCount=l24d.lnServiceCount
     (25 missing values generated)

  140 . gen l36dlnServiceCount=l36d.lnServiceCount
     (37 missing values generated)

  141 . gen l48dlnServiceCount=l48d.lnServiceCount
     (49 missing values generated)

  142 .
  143 . /*
     > The project is to forecast the March non-seasonally adjusted estimates of
ave
     > rage weekly earnings and total employment for private employers (total
privat
     > e) for a Florida MSA of your choice and write up a professional report on
you
     > r forecast.
     > */
  144 . /* Count and WeeklyEarnings */
  145 .
```

```
   146 . summ Count WeekHours HourlyEarnings WeeklyEarnings ServiceCount

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
             Count |        374    14.18556    6.880684        5.3         28
         WeekHours |        123    36.88455    3.791817       28.3       45.8
      HourlyEarn~s |        123       19.72    2.903968      15.01       24.6
      WeeklyEarn~s |        122    719.6542    84.57241     503.79      916.1
      ServiceCount |        375    10.43387    5.959179        3.9       22.8

   147 . summ lnCount lnWeekHours lnHourlyEarnings lnWeeklyEarnings lnServiceCount

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
           lnCount |        374      2.5174    .5398403   1.667707   3.332205
       lnWeekHours |        123    3.602488      .10385   3.342862   3.824284
      lnHourlyEa~s |        123    2.970779    .1482819   2.708717   3.202746
      lnWeeklyEa~s |        122    6.571775    .1195694   6.222159   6.820126
      lnServiceC~t |        375    2.172053    .5985689   1.360977    3.12676

   148 .
   149 . ac lnCount, saving(lnCount_ac, replace)
      (file lnCount_ac.gph saved)

   150 . pac lnCount, saving(lnCount_pac, replace)
      (file lnCount_pac.gph saved)

   151 . graph combine lnCount_ac.gph lnCount_pac.gph, saving(lnCount_ac_pac,
replace)
      (file lnCount_ac_pac.gph saved)

   152 . graph export "lnCount_ac_pac.png", replace
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets
      > /Final Project/lnCount_ac_pac.png written in PNG format)

   153 . ** Probably need to difference
   154 .
   155 . ac lnWeeklyEarnings, saving(lnWeeklyEarnings_ac, replace)
      (file lnWeeklyEarnings_ac.gph saved)

   156 . pac lnWeeklyEarnings, saving(lnWeeklyEarnings_pac, replace)
      (file lnWeeklyEarnings_pac.gph saved)

   157 . graph combine lnWeeklyEarnings_ac.gph lnWeeklyEarnings_pac.gph,
saving(lnWeek
      > lyEarnings_ac_pac, replace)
      (file lnWeeklyEarnings_ac_pac.gph saved)

   158 . graph export "lnWeeklyEarnings_ac_pac.png", replace
```

```
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
      > /Final Project/lnWeeklyEarnings_ac_pac.png written in PNG format)

   159 . ** Probably need to difference
   160 .
   161 . *starter models for count
   162 . *I used a pair plot to examine the rise and fall of variables with respect to
      >  each other
   163 . reg d.lnCount l(12,24,36,48)d.lnCount // .01637

            Source |       SS           df       MS      Number of obs   =       325
      -------------+----------------------------------   F(4, 320)       =     16.54
             Model |  .017539188         4  .004384797   Prob > F        =    0.0000
          Residual |  .084856979       320  .000265178   R-squared       =    0.1713
      -------------+----------------------------------   Adj R-squared   =    0.1609
             Total |  .102396167       324  .000316038   Root MSE        =    .01628

      ------------------------------------------------------------------------------
         D.lnCount |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
           lnCount |
             L12D. |   .3609966   .0621085     5.81   0.000      .238804    .4831893
             L24D. |    .137848   .0617615     2.23   0.026      .016338     .259358
             L36D. |  -.0160136   .0614584    -0.26   0.795    -.1369272      .1049
             L48D. |   .1265117   .0585322     2.16   0.031     .0113551    .2416683
                   |
             _cons |   .0017116   .0009853     1.74   0.083    -.0002269    .0036502
      ------------------------------------------------------------------------------

   164 . scalar drop _all

   165 . quietly forval w=12(12)144 {

   166 . scalar list
```

```
      RWmaxobs144 =        144
      RWminobs144 =         12
       RWrmse144 =   .0172276
      RWmaxobs132 =        132
      RWminobs132 =         12
       RWrmse132 =   .0172128
      RWmaxobs120 =        120
      RWminobs120 =         12
       RWrmse120 =  .01721825
      RWmaxobs108 =        108
      RWminobs108 =         12
       RWrmse108 =  .01723674
      RWmaxobs96 =         96
      RWminobs96 =         12
        RWrmse96 =  .01722006
      RWmaxobs84 =         84
      RWminobs84 =         12
        RWrmse84 =  .01726063
      RWmaxobs72 =         72
      RWminobs72 =         12
        RWrmse72 =  .01722377
      RWmaxobs60 =         60
      RWminobs60 =         12
        RWrmse60 =   .0173443
      RWmaxobs48 =         48
      RWminobs48 =         12
        RWrmse48 =  .01755803
      RWmaxobs36 =         36
      RWminobs36 =         12
        RWrmse36 =  .01805924
      RWmaxobs24 =         24
      RWminobs24 =         12
        RWrmse24 =   .0185871
      RWmaxobs12 =         12
      RWminobs12 =         12
        RWrmse12 =  .02320505

  167 . /*
     > RWmaxobs132 =        132
     > RWminobs132 =         12
     > RWrmse132 =   .0172128
     > */
  168 .
  169 . reg d.lnCount l(5,12,24,36,48)d.lnCount l(5)d.lnWeekHours m5 // .01711

           Source |       SS           df       MS      Number of obs   =
116
     -------------+----------------------------------   F(7, 108)       =
5.94
            Model |  .012171566         7  .001738795   Prob > F        =
0.0000
```

```
          Residual |   .03162877       108  .000292859   R-squared       =
0.2779
      -------------+----------------------------------   Adj R-squared   =
0.2311
             Total |  .043800336       115  .000380872   Root MSE        =
.01711

      ------------------------------------------------------------------------
-
         D.lnCount |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
Interval]
      -------------+----------------------------------------------------------
-
           lnCount |
              L5D. |  -.1231921   .0845717    -1.46   0.148     -.290828
.0444438
             L12D. |   .5811114   .1685831     3.45   0.001     .2469504
.9152724
             L24D. |  -.1196017   .1627467    -0.73   0.464    -.4421938
.2029904
             L36D. |   .2532303   .1742525     1.45   0.149    -.0921684
.5986291
             L48D. |   .1341638   .1858633     0.72   0.472    -.2342495
.5025771
                   |
       lnWeekHours |
              L5D. |   .0170123   .0364906     0.47   0.642    -.0553184
.089343
                   |
                m5 |   .0067588   .0061605     1.10   0.275    -.0054524
.0189699
             _cons |   .0004279   .0018229     0.23   0.815    -.0031854
.0040412
      ------------------------------------------------------------------------
-

   170 . scalar drop _all

   171 . quietly forval w=12(12)84 {

   172 . scalar list
      RWmaxobs84 =         84
      RWminobs84 =         23
        RWrmse84 =  .01950911
      RWmaxobs72 =         72
      RWminobs72 =         23
        RWrmse72 =  .01949719
      RWmaxobs60 =         60
      RWminobs60 =         23
        RWrmse60 =   .0199438
      RWmaxobs48 =         48
```

```
      RWminobs48 =         23
        RWrmse48 =  .02035982
      RWmaxobs36 =         36
      RWminobs36 =         23
        RWrmse36 =  .02138785
      RWmaxobs24 =         24
      RWminobs24 =         23
        RWrmse24 =  .02268585
      RWmaxobs12 =         12
      RWminobs12 =         12
        RWrmse12 =  .05004898

   173 . /*
      > RWmaxobs84 =         84
      > RWminobs84 =         23
      > RWrmse84 =  .01950911
      > */
   174 .
   175 . quietly gsreg dlnCount l1dlnCount l2dlnCount l3dlnCount l4dlnCount
l5dlnCount
      >  l6dlnCount ///
      >         l7dlnCount l8dlnCount l9dlnCount l10dlnCount l11dlnCount
l12dlnCount
      > ///
      >         l24dlnCount l36dlnCount l48dlnCount ///
      >         if tin(1990m1,2021m1), ///
      >         ncomb(1,12) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
m11
      > m12) ///
      >         samesample nindex( -1 aic -1 bic -1 rmse_out)
results(gsreg_dlnCount)
      >  replace

   176 .
   177 . *gsreg suggestions
   178 . reg d.lnCount l12d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

            Source |       SS           df       MS      Number of obs   =
361
      -------------+----------------------------------   F(12, 348)      =
6.91
             Model |  .022616974        12  .001884748   Prob > F        =
0.0000
          Residual |  .094871179       348  .000272618   R-squared       =
0.1925
      -------------+----------------------------------   Adj R-squared   =
0.1647
             Total |  .117488153       360  .000326356   Root MSE        =
.01651

```

```
------------------------------------------------------------------------------
         D.lnCount |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------------+----------------------------------------------------------------
           lnCount |
             L12D. |   .1748571   .0594184     2.94   0.003     .0579928    .2917215
                   |
               m1 |  -.0099477   .0043004    -2.31   0.021    -.0184056   -.0014897
               m2 |   .0009939   .0042297     0.23   0.814    -.0073251    .0093129
               m3 |   .0030247   .0042759     0.71   0.480    -.0053851    .0114345
               m4 |  -.0071933   .0042648    -1.69   0.093    -.0155814    .0011948
               m5 |  -.0098194   .0043178    -2.27   0.024    -.0183118   -.0013271
               m6 |  -.0133285   .0043874    -3.04   0.003    -.0219576   -.0046994
               m7 |  -.0091828   .0042967    -2.14   0.033    -.0176336    -.000732
               m8 |  -.0017998   .0042632    -0.42   0.673    -.0101846    .006585
               m9 |   -.006737   .0042824    -1.57   0.117    -.0151597    .0016858
              m10 |   .0062149   .0042795     1.45   0.147    -.0022021    .0146319
              m11 |   .0042124   .0042811     0.98   0.326    -.0042078    .0126325
            _cons |   .0072199   .0030452     2.37   0.018     .0012306    .0132093
------------------------------------------------------------------------------

  179 . scalar drop _all

  180 . quietly forval w=12(12)144 {

  181 . scalar list
     RWmaxobs144 =        144
     RWminobs144 =         12
      RWrmse144 =  .01824906
     RWmaxobs132 =        132
     RWminobs132 =         12
      RWrmse132 =  .01832173
     RWmaxobs120 =        120
     RWminobs120 =         12
      RWrmse120 =  .01833557
```

```
      RWmaxobs108 =        108
      RWminobs108 =         12
       RWrmse108 =  .01841089
      RWmaxobs96 =         96
      RWminobs96 =         12
        RWrmse96 =  .01836974
      RWmaxobs84 =         84
      RWminobs84 =         12
        RWrmse84 =  .01849267
      RWmaxobs72 =         72
      RWminobs72 =         12
        RWrmse72 =  .01861349
      RWmaxobs60 =         60
      RWminobs60 =         12
        RWrmse60 =  .01911515
      RWmaxobs48 =         48
      RWminobs48 =         12
        RWrmse48 =  .01922268
      RWmaxobs36 =         36
      RWminobs36 =         12
        RWrmse36 =  .01991683
      RWmaxobs24 =         24
      RWminobs24 =         12
        RWrmse24 =  .02022186
      RWmaxobs12 =         12
      RWminobs12 =         12
        RWrmse12 =  .02009249

   182 . /*
      > RWmaxobs144 =        144
      > RWminobs144 =         12
      > RWrmse144 =  .01824906
      > */
   183 .
   184 . reg d.lnCount l(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

            Source |       SS           df       MS      Number of obs   =
337
      -------------+----------------------------------   F(13, 323)      =
5.68
             Model |  .019946057        13  .001534312   Prob > F        =
0.0000
          Residual |  .087185203       323  .000269923   R-squared       =
0.1862
      -------------+----------------------------------   Adj R-squared   =
0.1534
             Total |  .107131259       336  .000318843   Root MSE        =
.01643

      ------------------------------------------------------------------------
-
```

```
         D.lnCount |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
Interval]
      -------------+----------------------------------------------------------
-
           lnCount |
             L12D. |   .1849403   .0636401     2.91   0.004      .059739
.3101417
             L36D. |   -.049332   .0606582    -0.81   0.417    -.1686671
.0700031
                   |
               m1 |  -.0073418   .0044769    -1.64   0.102    -.0161493
.0014658
               m2 |   .0022711   .0043559     0.52   0.602    -.0062984
.0108407
               m3 |   .0043593    .004416     0.99   0.324    -.0043285
.0130471
               m4 |  -.0065438   .0043922    -1.49   0.137    -.0151847
.002097
               m5 |  -.0089215   .0045194    -1.97   0.049    -.0178126
-.0000304
               m6 |  -.0133453   .0046241    -2.89   0.004    -.0224425
-.004248
               m7 |  -.0085154    .004457    -1.91   0.057    -.0172839
.0002531
               m8 |  -.0004554    .004392    -0.10   0.917    -.0090959
.0081852
               m9 |  -.0056625   .0044299    -1.28   0.202    -.0143775
.0030526
              m10 |   .0071688   .0044386     1.62   0.107    -.0015635
.0159011
              m11 |   .0042074   .0044259     0.95   0.343    -.0044998
.0129146
            _cons |   .0067355   .0031722     2.12   0.034     .0004948
.0129762
      ------------------------------------------------------------------------
-

   185 . scalar drop _all

   186 . quietly forval w=12(12)144 {

   187 . scalar list
      RWmaxobs144 =        144
      RWminobs144 =         12
       RWrmse144 =  .01777071
      RWmaxobs132 =        132
      RWminobs132 =         12
       RWrmse132 =  .01782557
      RWmaxobs120 =        120
      RWminobs120 =         12
       RWrmse120 =  .01785253
```

```
      RWmaxobs108 =        108
      RWminobs108 =         12
       RWrmse108 =  .01794692
      RWmaxobs96 =         96
      RWminobs96 =         12
        RWrmse96 =  .01793358
      RWmaxobs84 =         84
      RWminobs84 =         12
        RWrmse84 =  .01803355
      RWmaxobs72 =         72
      RWminobs72 =         12
        RWrmse72 =  .01807408
      RWmaxobs60 =         60
      RWminobs60 =         12
        RWrmse60 =  .01843535
      RWmaxobs48 =         48
      RWminobs48 =         12
        RWrmse48 =  .01835092
      RWmaxobs36 =         36
      RWminobs36 =         12
        RWrmse36 =  .01863303
      RWmaxobs24 =         24
      RWminobs24 =         12
        RWrmse24 =   .0196745
      RWmaxobs12 =         12
      RWminobs12 =         12
        RWrmse12 =  .01880291

   188 . /*
      > RWmaxobs144 =        144
      > RWminobs144 =         12
      > RWrmse144 =  .01777071
      > */
   189 .
   190 . reg d.lnCount l4d.lnWeekHours l9d.lnWeekHours l8d.lnHourlyEarnings m1 m2
m3 m
      > 4 m5 m6 m7 m8 m9 m10 m11

            Source |       SS           df       MS      Number of obs   =
112
      -------------+----------------------------------   F(14, 97)       =
3.24
             Model |  .013917393        14    .0009941   Prob > F        =
0.0003
          Residual |  .029798432        97    .0003072   R-squared       =
0.3184
      -------------+----------------------------------   Adj R-squared   =
0.2200
             Total |  .043715825       111  .000393836   Root MSE        =
.01753

```

```
      ------------------------------------------------------------------------------
      ---
             D.lnCount |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interv
      > al]
      -----------------+------------------------------------------------------------
      ---
           lnWeekHours |
                  L4D. |  -.0013847   .0384012    -0.04   0.971    -.0776005    .074
      > 831
                  L9D. |   .0397686   .0385964     1.03   0.305    -.0368346   .1163
      > 718
                       |
      lnHourlyEarnings |
                  L8D. |   -.039029   .0414024    -0.94   0.348    -.1212014   .0431
      > 433
                       |
                    m1 |  -.0097045   .0078517    -1.24   0.219    -.0252879   .0058
      > 789
                    m2 |   .0000949   .0079445     0.01   0.990    -.0156727   .0158
      > 626
                    m3 |   -.004712   .0083585    -0.56   0.574    -.0213013   .0118
      > 773
                    m4 |  -.0273667   .0081729    -3.35   0.001    -.0435876   -.0111
      > 459
                    m5 |  -.0076836   .0081259    -0.95   0.347    -.0238112   .008
      > 444
                    m6 |   -.020254   .0081465    -2.49   0.015    -.0364227   -.0040
      > 854
                    m7 |  -.0130812   .0081852    -1.60   0.113    -.0293265   .0031
      > 642
                    m8 |   .0041701   .0081051     0.51   0.608    -.0119164   .0202
      > 565
                    m9 |  -.0089171   .0082764    -1.08   0.284    -.0253435   .0075
      > 093
                   m10 |   .0153608   .0081153     1.89   0.061    -.0007459   .0314
```

```
      > 674
                    m11 |   .0040463   .0079619     0.51   0.612    -.0117559
.0198
      > 485
                  _cons |   .0094122   .0056462     1.67   0.099    -.0017939
.0206
      > 183
      ------------------------------------------------------------------------------
--
      ---

   191 . scalar drop _all

   192 . quietly forval w=12(12)84 {

   193 . scalar list
      RWmaxobs84 =         84
      RWminobs84 =          2
        RWrmse84 =  .01847546
      RWmaxobs72 =         72
      RWminobs72 =          2
        RWrmse72 =  .01855448
      RWmaxobs60 =         60
      RWminobs60 =          2
        RWrmse60 =  .01850723
      RWmaxobs48 =         48
      RWminobs48 =          2
        RWrmse48 =  .01850217
      RWmaxobs36 =         36
      RWminobs36 =          2
        RWrmse36 =  .01942535
      RWmaxobs24 =         24
      RWminobs24 =          2
        RWrmse24 =  .02208272
      RWmaxobs12 =         12
      RWminobs12 =          2
        RWrmse12 =   .0176238

   194 . /*
      > RWmaxobs12 =         12
      > RWminobs12 =          2
      > RWrmse12 =   .0176238
      > */
   195 .
   196 . scalar rwrmse = .0172128

   197 . reg d.lnCount l(12,24,36,48)d.lnCount if tin(,2021m2)

           Source |       SS           df       MS      Number of obs   =
325
```

```
      -------------+----------------------------------   F(4, 320)       =
16.54
             Model |  .017539188         4  .004384797   Prob > F        =
0.0000
          Residual |  .084856979       320  .000265178   R-squared       =
0.1713
      -------------+----------------------------------   Adj R-squared   =
0.1609
             Total |  .102396167       324  .000316038   Root MSE        =
.01628

      ------------------------------------------------------------------------
-
         D.lnCount |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
Interval]
      -------------+----------------------------------------------------------
-
           lnCount |
             L12D. |   .3609966   .0621085     5.81   0.000      .238804
.4831893
             L24D. |    .137848   .0617615     2.23   0.026      .016338
.259358
             L36D. |  -.0160136   .0614584    -0.26   0.795    -.1369272
.1049
             L48D. |   .1265117   .0585322     2.16   0.031     .0113551
.2416683
                   |
             _cons |   .0017116   .0009853     1.74   0.083    -.0002269
.0036502
      ------------------------------------------------------------------------
-

   198 . predict pd
      (option xb assumed; fitted values)
      (49 missing values generated)

   199 . gen pflcount=exp((rwrmse^2)/2)*exp(l.lnCount+pd) if Date==tm(2021m3)
      (374 missing values generated)

   200 . gen ub1=exp((rwrmse^2)/2)*exp(l.lnCount+pd+1*rwrmse) if Date==tm(2021m3)
      (374 missing values generated)

   201 . gen lb1=exp((rwrmse^2)/2)*exp(l.lnCount+pd-1*rwrmse) if Date==tm(2021m3)
      (374 missing values generated)

   202 . gen ub2=exp((rwrmse^2)/2)*exp(l.lnCount+pd+2*rwrmse) if Date==tm(2021m3)
      (374 missing values generated)

   203 . gen lb2=exp((rwrmse^2)/2)*exp(l.lnCount+pd-2*rwrmse) if Date==tm(2021m3)
      (374 missing values generated)
```

```
  204 . gen ub3=exp((rwrmse^2)/2)*exp(l.lnCount+pd+3*rwrmse) if Date==tm(2021m3)
     (374 missing values generated)

  205 . gen lb3=exp((rwrmse^2)/2)*exp(l.lnCount+pd-3*rwrmse) if Date==tm(2021m3)
     (374 missing values generated)

  206 . drop pd

  207 .
  208 . replace pflcount=Count if Date==tm(2021m2)
     (1 real change made)

  209 . replace ub1=Count if Date==tm(2021m2)
     (1 real change made)

  210 . replace ub2=Count if Date==tm(2021m2)
     (1 real change made)

  211 . replace ub3=Count if Date==tm(2021m2)
     (1 real change made)

  212 . replace lb1=Count if Date==tm(2021m2)
     (1 real change made)

  213 . replace lb2=Count if Date==tm(2021m2)
     (1 real change made)

  214 . replace lb3=Count if Date==tm(2021m2)
     (1 real change made)

  215 .
  216 . twoway (tsrline ub3 ub2 if tin(2020m3,2021m3), ///
     >         recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
     >         (tsrline ub2 ub1 if tin(2020m3,2021m3), ///
     >         recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
     >         (tsrline ub1 pflcount if tin(2020m3,2021m3), ///
     >         recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
     >         (tsrline pflcount lb1 if tin(2020m3,2021m3), ///
     >         recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
     >         (tsrline lb1 lb2 if tin(2020m3,2021m3), ///
     >         recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
     >         (tsrline lb2 lb3 if tin(2020m3,2021m3), ///
     >         recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
     >         (tsline Count pflcount if tin(2020m3,2021m3) , ///
     >         lcolor(gs12 teal) lwidth(medthick medthick) ///
     >         lpattern(solid longdash)) ///
     >         (scatter withMarchCount Date if tin(2021m3,)), scheme(s1mono)
legend(
     > off)

  217 . graph export "CountFan.png", replace
```

```
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets
      > /Final Project/CountFan.png written in PNG format)

   218 .
   219 . /*---------------------------------------------------------------------
---
      > -*/
   220 .
   221 . *starter models for weekly earnings
   222 . reg d.lnWeeklyEarnings l1d.lnWeekHours ld.lnHourlyEarnings

            Source |       SS           df       MS      Number of obs   =
120
      -------------+----------------------------------   F(2, 117)       =
1.48
             Model |    .0071986         2    .0035993   Prob > F        =
0.2316
          Residual |  .284290844       117  .002429836   R-squared       =
0.0247
      -------------+----------------------------------   Adj R-squared   =
0.0080
             Total |  .291489443       119  .002449491   Root MSE        =
.04929

      ------------------------------------------------------------------------
--
      ---
      D.                |
      lnWeeklyEarnings |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
Interv
      > al]
      -----------------+------------------------------------------------------
--
      ---
           lnWeekHours |
                   LD. |  -.1334776   .1058563    -1.26   0.210    -.3431204
.0761
      > 651
                       |
      lnHourlyEarnings |
                   LD. |   .0746807   .1166933     0.64   0.523    -.1564244
.3057
      > 857
                       |
                 _cons |   .0015108   .0045071     0.34   0.738    -.0074152
.0104
      > 368
      ------------------------------------------------------------------------
--
      ---
```

```
959
960     223 . scalar drop _all
961
962     224 . quietly forval w=12(12)84 {
963
964     225 . scalar list
965        RWmaxobs84 =         84
966        RWminobs84 =          2
967          RWrmse84 =  .06183191
968        RWmaxobs72 =         72
969        RWminobs72 =          2
970          RWrmse72 =  .06162109
971        RWmaxobs60 =         60
972        RWminobs60 =          2
973          RWrmse60 =  .06144232
974        RWmaxobs48 =         48
975        RWminobs48 =          2
976          RWrmse48 =   .0618403
977        RWmaxobs36 =         36
978        RWminobs36 =          2
979          RWrmse36 =  .06201409
980        RWmaxobs24 =         24
981        RWminobs24 =          2
982          RWrmse24 =  .06224974
983        RWmaxobs12 =         12
984        RWminobs12 =          2
985          RWrmse12 =  .06583082
986
987     226 . /*
988        > RWmaxobs60 =         60
989        > RWminobs60 =          2
990        > RWrmse60 =  .06145693
991        > */
992     227 .
993     228 . quietly gsreg dlnWeeklyEarnings l1dlnWeeklyEarnings l2dlnWeeklyEarnings
    l3dln
994        > WeeklyEarnings ///
995        >         l4dlnWeeklyEarnings l5dlnWeeklyEarnings l6dlnWeeklyEarnings ///
996        >         l7dlnWeeklyEarnings l8dlnWeeklyEarnings l9dlnWeeklyEarnings
    l10dlnWee
997        > klyEarnings ///
998        >         l11dlnWeeklyEarnings l12dlnWeeklyEarnings ///
999        >         l24dlnWeeklyEarnings l36dlnWeeklyEarnings ///
1000       >         if tin(2011m1,2021m1), ///
1001       >         ncomb(1,12) aic outsample(24) ///
1002       >         samesample nindex( -1 aic -1 bic -1 rmse_out)
    results(gsreg_dlnWeekly
1003       > Earnings) replace
1004
1005    229 .
```

```
  230 . reg d.lnWeeklyEarnings l3d.lnWeeklyEarnings l5d.lnWeeklyEarnings m1 m2 m3 m4
      > m5 m6 m7 m8 m9 m10 m11

            Source |       SS           df       MS      Number of obs   =       116
      -------------+----------------------------------   F(13, 102)      =      2.16
             Model |  .061304493        13   .00471573   Prob > F        =    0.0166
          Residual |  .222983103       102  .002186109   R-squared       =    0.2156
      -------------+----------------------------------   Adj R-squared   =    0.1157
             Total |  .284287596       115  .002472066   Root MSE        =    .04676

      ------------------------------------------------------------------------------
      ---
      D.                |
      lnWeeklyEarnings |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interv
      > al]
      -----------------+------------------------------------------------------------
      ---
      lnWeeklyEarnings |
                  L3D. |  -.2267216   .0950679    -2.38   0.019    -.4152883   -.0381
      > 549
                  L5D. |  -.1621197    .095104    -1.70   0.091     -.350758    .0265
      > 186
                       |
                    m1 |   -.013308   .0213233    -0.62   0.534    -.0556027    .0289
      > 866
                    m2 |    .020775   .0212478     0.98   0.331      -.02137     .06
      > 292
                    m3 |  -.0123903   .0220875    -0.56   0.576    -.0562008    .0314
      > 201
                    m4 |   .0105198   .0219037     0.48   0.632    -.0329261    .0539
      > 657
                    m5 |   .0377285   .0216445     1.74   0.084    -.0052032    .0806
      > 602
```

```
                    m6 |   .0272631   .0216181     1.26   0.210    -.0156164
.0701
      > 426
                    m7 |  -.0220653   .0214504    -1.03   0.306     -.064612
.0204
      > 813
                    m8 |   .0152172   .0210597     0.72   0.472    -.0265547
.0569
      > 891
                    m9 |   .0201901   .0215988     0.93   0.352    -.0226509
.0630
      > 312
                   m10 |   .0207722    .021844     0.95   0.344    -.0225553
.0640
      > 997
                   m11 |   .0084712   .0217091     0.39   0.697    -.0345888
.0515
      > 312
                 _cons |  -.0073031   .0151547    -0.48   0.631    -.0373625
.0227
      > 563
      ------------------------------------------------------------------------------
--
      ---

   231 . scalar drop _all

   232 . quietly forval w=12(12)84 {

   233 . scalar list
      RWmaxobs84 =         84
      RWminobs84 =          2
        RWrmse84 =  .06011868
      RWmaxobs72 =         72
      RWminobs72 =          2
        RWrmse72 =  .06057642
      RWmaxobs60 =         60
      RWminobs60 =          2
        RWrmse60 =  .06071208
      RWmaxobs48 =         48
      RWminobs48 =          2
        RWrmse48 =  .06042055
      RWmaxobs36 =         36
      RWminobs36 =          2
        RWrmse36 =  .06125152
      RWmaxobs24 =         24
      RWminobs24 =          2
        RWrmse24 =  .06537943
      RWmaxobs12 =         12
      RWminobs12 =          2
        RWrmse12 =   .0702019
```

```
1082
1083     234 . /*
1084        > RWmaxobs84 =         84
1085        > RWminobs84 =          2
1086        > RWrmse84 =  .06004448
1087        > */
1088     235 .
1089     236 . reg d.lnWeeklyEarnings l3d.lnWeeklyEarnings l5d.lnWeeklyEarnings
      l7d.lnWeekly
1090        > Earnings
1091
1092               Source |       SS           df       MS      Number of obs   =
      114
1093      -------------+----------------------------------   F(3, 110)       =
      3.85
1094               Model |  .026396014         3  .008798671   Prob > F        =
      0.0115
1095            Residual |  .251283445       110  .002284395   R-squared       =
      0.0951
1096      -------------+----------------------------------   Adj R-squared   =
      0.0704
1097               Total |  .277679459       113   .00245734   Root MSE        =
      .0478
1098
1099      ------------------------------------------------------------------------------
      --
1100      ---
1101      D.                |
1102      lnWeeklyEarnings |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
      Interv
1103      > al]
1104      -----------------+------------------------------------------------------------
      --
1105      ---
1106      lnWeeklyEarnings |
1107                   L3D. |  -.2408947   .0906673    -2.66   0.009    -.4205761
      -.0612
1108      > 133
1109                   L5D. |  -.1892527   .0903206    -2.10   0.038    -.3682468
      -.0102
1110      > 585
1111                   L7D. |   .0639647   .0913902     0.70   0.485    -.1171492
      .2450
1112      > 786
1113                        |
1114                 _cons |   .0025724   .0044848     0.57   0.567    -.0063154
      .0114
1115      > 602
1116      ------------------------------------------------------------------------------
      --
1117      ---
```

```
1118
1119     237 . scalar drop _all
1120
1121     238 . quietly forval w=12(12)84 {
1122
1123     239 . scalar list
1124        RWmaxobs84 =         84
1125        RWminobs84 =          2
1126          RWrmse84 =  .05259823
1127        RWmaxobs72 =         72
1128        RWminobs72 =          2
1129          RWrmse72 =  .05283772
1130        RWmaxobs60 =         60
1131        RWminobs60 =          2
1132          RWrmse60 =  .05314168
1133        RWmaxobs48 =         48
1134        RWminobs48 =          2
1135          RWrmse48 =   .0530381
1136        RWmaxobs36 =         36
1137        RWminobs36 =          2
1138          RWrmse36 =  .05353125
1139        RWmaxobs24 =         24
1140        RWminobs24 =          2
1141          RWrmse24 =  .05282122
1142        RWmaxobs12 =         12
1143        RWminobs12 =          2
1144          RWrmse12 =  .06036464
1145
1146     240 . /*
1147        > RWmaxobs84 =         84
1148        > RWminobs84 =          2
1149        > RWrmse84 =  .05250414
1150        > */
1151     241 .
1152     242 . drop pflcount ub1 ub2 ub3 lb1 lb2 lb3
1153
1154     243 .
1155     244 . scalar rwrmse = .05250414
1156
1157     245 . reg d.lnWeeklyEarnings l(3,5,7)d.lnWeeklyEarnings if tin(,2021m2)
1158
1159             Source |       SS           df       MS      Number of obs   =
     114
1160       -------------+----------------------------------   F(3, 110)       =
     3.85
1161              Model |  .026396014         3  .008798671   Prob > F        =
     0.0115
1162           Residual |  .251283445       110  .002284395   R-squared       =
     0.0951
1163       -------------+----------------------------------   Adj R-squared   =
     0.0704
```

```
1164              Total |  .277679459       113   .00245734   Root MSE        =
      .0478
1165
1166       ------------------------------------------------------------------------------
      --
1167       ---
1168       D.               |
1169       lnWeeklyEarnings |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
      Interv
1170       > al]
1171       -----------------+------------------------------------------------------------
      --
1172       ---
1173       lnWeeklyEarnings |
1174                   L3D. |  -.2408947   .0906673    -2.66   0.009    -.4205761
      -.0612
1175       > 133
1176                   L5D. |  -.1892527   .0903206    -2.10   0.038    -.3682468
      -.0102
1177       > 585
1178                   L7D. |   .0639647   .0913902     0.70   0.485    -.1171492
      .2450
1179       > 786
1180                        |
1181                 _cons |   .0025724   .0044848     0.57   0.567    -.0063154
      .0114
1182       > 602
1183       ------------------------------------------------------------------------------
      --
1184       ---
1185
1186     246 . predict pd
1187         (option xb assumed; fitted values)
1188         (260 missing values generated)
1189
1190     247 . gen pflcount=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd) if
      Date==tm(2021m3)
1191         (374 missing values generated)
1192
1193     248 . gen ub1=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+1*rwrmse) if
      Date==tm(202
1194         > 1m3)
1195         (374 missing values generated)
1196
1197     249 . gen lb1=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-1*rwrmse) if
      Date==tm(202
1198         > 1m3)
1199         (374 missing values generated)
1200
1201     250 . gen ub2=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+2*rwrmse) if
      Date==tm(202
```

```
1202        > 1m3)
1203        (374 missing values generated)
1204
1205     251 . gen lb2=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-2*rwrmse) if
    Date==tm(202
1206        > 1m3)
1207        (374 missing values generated)
1208
1209     252 . gen ub3=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+3*rwrmse) if
    Date==tm(202
1210        > 1m3)
1211        (374 missing values generated)
1212
1213     253 . gen lb3=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-3*rwrmse) if
    Date==tm(202
1214        > 1m3)
1215        (374 missing values generated)
1216
1217     254 . drop pd
1218
1219     255 .
1220     256 . replace pflcount=WeeklyEarnings if Date==tm(2021m2)
1221        (1 real change made)
1222
1223     257 . replace ub1=WeeklyEarnings if Date==tm(2021m2)
1224        (1 real change made)
1225
1226     258 . replace ub2=WeeklyEarnings if Date==tm(2021m2)
1227        (1 real change made)
1228
1229     259 . replace ub3=WeeklyEarnings if Date==tm(2021m2)
1230        (1 real change made)
1231
1232     260 . replace lb1=WeeklyEarnings if Date==tm(2021m2)
1233        (1 real change made)
1234
1235     261 . replace lb2=WeeklyEarnings if Date==tm(2021m2)
1236        (1 real change made)
1237
1238     262 . replace lb3=WeeklyEarnings if Date==tm(2021m2)
1239        (1 real change made)
1240
1241     263 .
1242     264 . twoway (tsrline ub3 ub2 if tin(2020m3,2021m3), ///
1243        >         recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
1244        >         (tsrline ub2 ub1 if tin(2020m3,2021m3), ///
1245        >         recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
1246        >         (tsrline ub1 pflcount if tin(2020m3,2021m3), ///
1247        >         recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
1248        >         (tsrline pflcount lb1 if tin(2020m3,2021m3), ///
1249        >         recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
```

```
1250        >         (tsrline lb1 lb2 if tin(2020m3,2021m3), ///
1251        >         recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
1252        >         (tsrline lb2 lb3 if tin(2020m3,2021m3), ///
1253        >         recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
1254        >         (tsline WeeklyEarnings pflcount if tin(2020m3,2021m3) , ///
1255        >         lcolor(gs12 teal) lwidth(medthick medthick) ///
1256        >         lpattern(solid longdash)) ///
1257        >         (scatter withMarchEarnings Date if tin(2021m3,)), scheme(s1mono)
      lege
1258        > nd(off)
1259
1260     265 . graph export "WeeklyFan.png", replace
1261        (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
      Sets
1262        > /Final Project/WeeklyFan.png written in PNG format)
1263
1264     266 .
1265     267 . log close
1266              name:  <unnamed>
1267               log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
      Series/Probl
1268        > em Sets/Final Project/Final Project.smcl
1269          log type:  smcl
1270         closed on:  19 Apr 2021, 21:08:41
1271        ------------------------------------------------------------------------------
      --
```