1 Notes on Matrix Derivation

Thursday, January 14, 2021 5:39 PM

Matrix Algebra for OLS Estimator

Big Picture

- Matrix algebra can produce compact notation
- some programs are matrix oriented
- Excel is a matrix

Dependent Variable

- dependent var is an $n\times 1$ column vector

$$Y=\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix}$$

- subscript = index
- use bold typeface

Independent Variables

- k independent vars and a constant term
- thus, $n\times(k+1)$ matrix size

Linear regression model

- define B as a $(k+1)\times 1$ vector of coefficients
  V as an $n\times 1$ vector of error terms

  ↳ linear multiple regression in matrix form: $Y=XB+V$
- keep track of the dimensions

First order condition of applying RSS

- OLS estimators are residual sum squares (RSS)

$$\frac{dRSS}{dB_j}=0 \Rightarrow \sum_{i=1}^{n} x_{ij}\hat{V}_i=0,\quad (j=0,1,\dots,k)$$

↳ $\hat{V}$ = residual

- system of k+1 equations written as $X'\hat{V}=0 \rightarrow (k+1)\times 1$ vector of 0

↳ transpose of X

OLS Estimators in Matrix Form

- $\hat{B}$ is a $(k+1)\times 1$ vector of OLS estimates

$$X'V=0$$
$$X'(Y-X\hat{B})=0$$
$$X'Y=(X'X)\hat{B}$$
$$\hat{B}=(X'X)^{-1}(X'Y)$$

An important result

$$\hat{B}=(X'X)^{-1}(X'Y)=(X'X)^{-1}(X'(XB+V))=B+(X'X)^{-1}(X'V)$$

- $\hat{B}$ in general differs from B due to the error V
- B is an unknown constant
- distribution of $\hat{B}$ is the sampling distribution

Statistical Properties of OLS estimator I

- under certain assumptions, the OLS estimator is unbiased

Statistical Properties of OLS estimator II

- most likely $\hat{B}$ is biased for two reasons:
  1) data is not independent
  2) $E(V|X)\neq 0$ which can be contributed to an omitted variable, simultaneity, and measurement error

Statistical Properties of OLS Estimator III

only valid if homoskedasticity holds

$$E((\hat{B}-B)(\hat{B}-B)'|X)=\sigma^2(X'X)^{-1}$$

Heteroskedasticity

$$E((\hat{B}-B)(\hat{B}-B)'|X)=(X'X)^{-1}(X'\Omega X)(X'X)^{-1}$$

Ω = diagonal matrix

White Sandwich Estimator

$$X'\hat{\Omega}X=\sum_{i=1}^{n}\hat{V}_i^2 x_i x_i'$$

$$(X'X)^{-1}(X'\hat{\Omega}X)(X'X)^{-1}$$

Predicted Values

$$P\equiv X(X'X)^{-1}X'$$

↳ projection matrix

$$P=P' \qquad PP=P \qquad PX=X$$

Residuals

$$\hat{V}=Y-\hat{Y}=(I-P)Y=MY$$

$$M\equiv I-P$$

$$M=M' \qquad MM=M \qquad PM=0$$

Frisch Waugh Theorem I

1.1 Population Regression Function

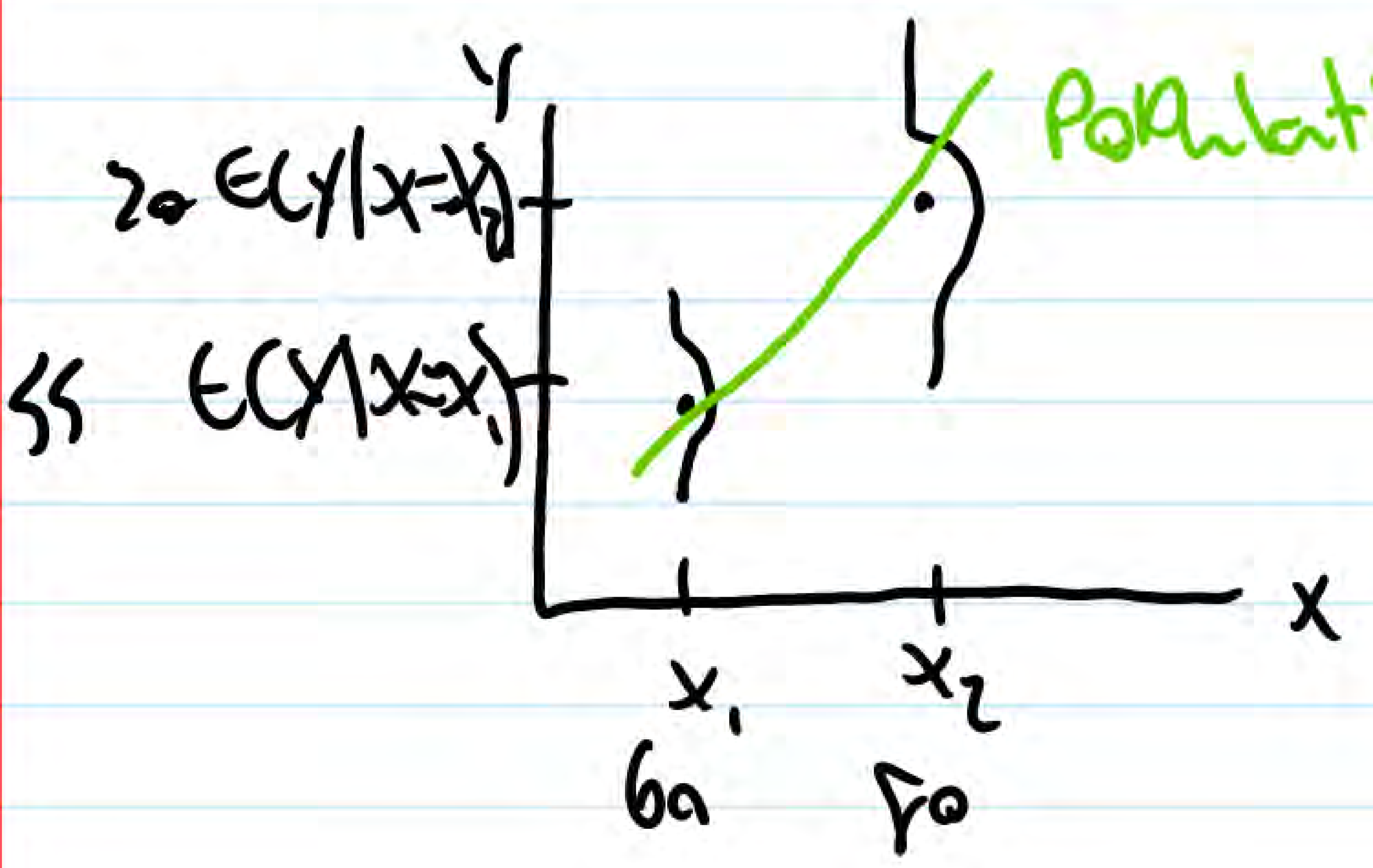
Wednesday, January 13, 2021 9:23 PM

PRF

Expected Value $= E(X) = \sum_i f_i x_i = \int_{-\infty}^{\infty} v f(v)\, dv$

Random Variable?

Probability density function



$Y_i = E(Y|X) +$ Residual

$E(Y|X) =$ linear function

$E(Y) = \beta_0 + \beta_1 X$

$= \beta_0 + \beta_1 X + \beta_2 X^2$

$= \beta_0 + \beta_1 X + \beta_2 \, {}^1/_X$

$E(\ln Y) = \beta_0 + \beta_1 \ln(X)$

Specification Error: shape of approx $\neq$ shape of PRF

1.1 Population Regression Function

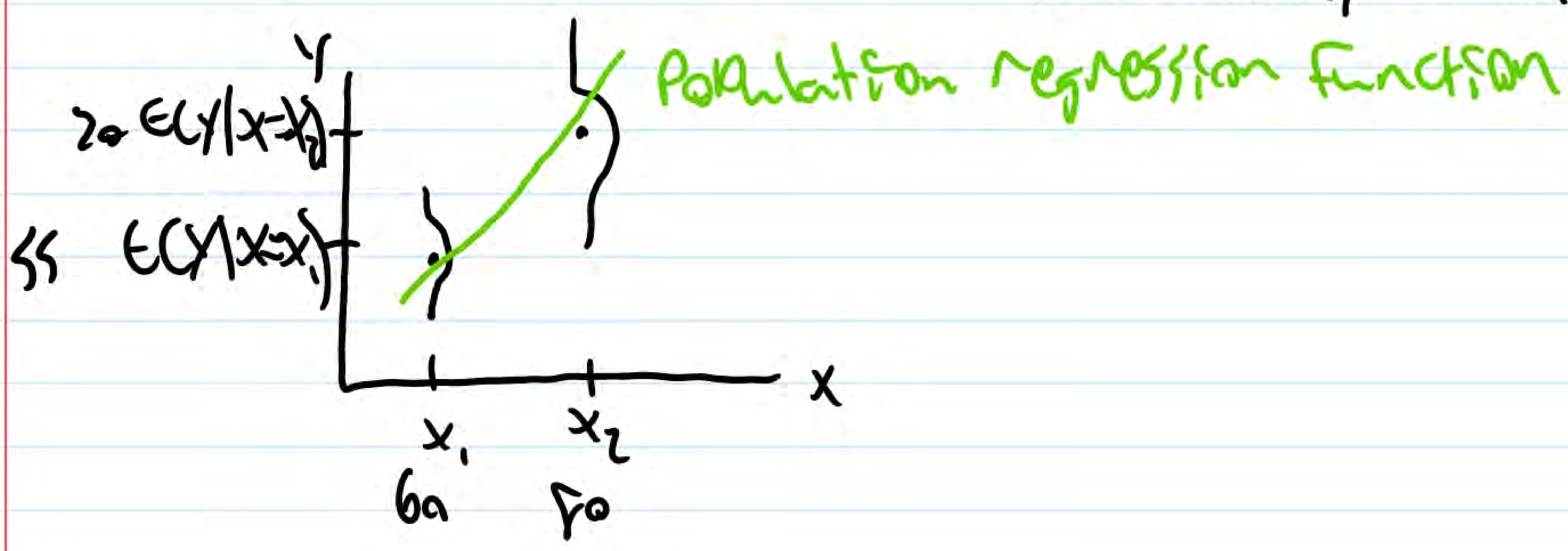
Wednesday, January 13, 2021 9:23 PM

PRF

Expected Value $= E(X) = \sum_i f_i x_i = \int_{-\infty}^{\infty} v f(v)\, dv$

Random Variable?

Probability density function



$Y_i = E(Y|X) + \text{Residual}$

$E(Y|X) = \text{linear function}$

$E(Y) = \beta_0 + \beta_1 X$

$= \beta_0 + \beta_1 X + \beta_2 X^2$

$= \beta_0 + \beta_1 X + \beta_2 \, 1/X$

$E(\ln Y) = \beta_0 + \beta_1 \ln(X)$

Specification Error: shape of approx $\neq$ shape of PRF

Data

$$
\begin{array}{c c}
 & j \\
 & \begin{array}{cc} Y & \begin{array}{ccccc} X_1 & X_2 & X_3 & \dots & X_k \end{array} \end{array} \\
i & \begin{bmatrix} 70 \\ 73 \\ 85 \\ 80 \\ \vdots \end{bmatrix} \begin{bmatrix} 60 & 50 & & & \\ 80 & 60 & & & \\ 75 & \vdots & & & \\ 90 & \vdots & & & \\ \vdots & & & & \end{bmatrix}
\end{array}
$$

$x_0 = 1$ for all

$Y \qquad X$

$$E(y_i | x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots$$

$$y_i = \underbrace{E(y_i | x_i)}_{\text{Predictable}} + r_i$$

$$E(y|x) = X\beta$$

↳ vector of unknowns
↳ x matrix

$$Y = X\beta + r$$

↳ residual matrix

$$
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots \\ 1 & x_{21} & \vdots & \\ 1 & \vdots & \vdots & \\ 1 & \vdots & \vdots & \\ 1 & x_{n1} & \vdots & \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}
$$

$i \in 1, 2, \dots n$

$Y = X\beta + r$

$y_1 = 1 \cdot \beta_0 + x_{11}\beta_1 + x_{12}\beta_2 + \dots x_{1k}\beta_k + r_1$

$\dots y_n = \dots + r_n$

1.3 Estimating the Betas

Wednesday, January 13, 2021 10:03 PM

$y_i = E(y|x) + r_i = \sum_j \beta_j x_{ij} + r_i$

$r_i = y_i - \sum_j \beta_j x_{ij}$

$r_i^2 = (y_i - \sum_j \beta_j x_{ij})^2$

$SSR = \sum_i r_i^2 = \sum_i (y_i - \sum_j \beta_j x_{ij})^2$

<u>Sample</u>

$\sum \hat{r}_i^2 = \sum_i (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \dots - \hat{\beta}_k x_{ik})^2$

$\hat{\beta}$ is best estimate of β

$\hat{y}$ is best prediction of y

$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \dots$

$\hat{r}_i = y_i - \hat{y}_i$

$\boxed{\text{Min Sample SSR}}$ OLS

1.4 Minimizing the SSR

Wednesday, January 13, 2021 10:13 PM

$$\frac{dSSR}{dB_0} = \left(2 \sum (y - \sum_j B_j X_j)\right)(-1) = 0$$

$$\sum_i Y_i = \sum_i \sum_j B_j X_{ij}$$

$$\frac{\sum_i Y_i}{n} = \sum \hat{Y}_i / n$$

$$\hat{Y}/n = \bar{Y}$$

$$\sum Y_i X_{i1} = \sum \hat{Y}_i X_{i1}$$

1.5 The Normal Equations

Wednesday, January 13, 2021 10:19 PM

$$\frac{dssr}{d\hat{\beta}_j} = -2\sum_i (y_i - \hat{y})\, x_{ij} = 0$$

$$\sum_i \hat{r}_i \cdot x_{ij} = 0$$

$$j = 0 \quad x_{i0} = 1$$

$$\sum \hat{r}_i = 0$$

$$Corr(x_{ij}, r_i) = 0$$

$$\hat{Y}/n = (\sum \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \ldots)/n$$

$$\bar{Y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{X}_1 + \hat{\beta}_2 \bar{X}_2 + \ldots$$

choose $\hat{\beta}_0$ so line passes through sample mean

1.6-9 Statistical Properties of the OLS Estimates

Monday, January 18, 2021 7:08 PM

β = Slope of Population regression function

$\sum_i y_i x_{ij} = \sum \hat{y}_i x_{ij}$

$\sum_i y_i x_{ik} = \sum \hat{y}_i x_{ik}$

$\vdots$

$x'y = x'\hat{y}$

$\hat{y} = x\hat{\beta}$ $\qquad$ $x' = x^{\text{Transpose}} = x^T$

$x'y = x'x\hat{\beta}$

$\begin{bmatrix} 4/10 & -1/10 \\ -1/5 & 3/10 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ → Identity matrix

$(x^Tx)^{-1}x^Ty = (x^Tx)^{-1}x^Tx\hat{\beta}$

$\hat{\beta} = (x^Tx)^{-1}x^Ty$

→ unknown Pop. Values

$E(\hat{\beta}) = ? = E[(x^Tx)^{-1}x^T(x\beta + r)]$

$= E[\beta + (x^Tx)^{-1}x^Tr]$

$= \beta + (x^Tx)^{-1}x^T E(r|x)$

↳ $= 0$ in Population

$E(\hat{\beta}) = \beta$ if no spec error!

$Var(\hat{\beta}) \rightarrow E((\hat{\beta} - E(\hat{\beta}))^2)$

$E((\hat{\beta} - \beta)^2) = E(\cancel{\beta} + (x^Tx)^{-1}x^Tr - \cancel{\beta})^2$

$E[(x^Tx^{-1})x^Trr^Tx(x^Tx^{-1})]$

$Var(\hat{\beta}) = (x^Tx)^{-1}x^T E(rr^T|x)x(x^Tx)^{-1}$

$rr^T = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} [r_1 r_2 \ldots r_n]$ = $n \times n$ matrix

$E(rr^T|x) = ?$

Estimating the Sigma Matrix

$E(r_i r_j|x) = 0$

1) No spec error
2) random sample

$\Sigma = E$[diagonal matrix]

$\hat{\Sigma}$ = [diagonal hat matrix] → Assume homoskedasticity $E(r_i^2) = E(r_j^2)$ $\forall\, i = j$

$E(r_i^2) = \sigma_i^2 = \sigma_j^2 = \sigma^2$

I = identity matrix

Homoskedasticity

$Var(\hat{\beta}) = (x^Tx)^{-1}x^T\sigma^2 Ix(x^Tx)^{-1}$

$= \sigma^2(x^Tx)^{-1}x^Tx(x^Tx)^{-1}$

$= \sigma^2(x^Tx)^{-1}$ → default

= diagonal = Variance + off diagonal = Covariance

Sandwich Estimator

Eicker-hube-white Estimator

$\widehat{Var}(\hat{\beta}) = (x^Tx)^{-1}x^T\hat{\Sigma}x(x^Tx)^{-1}$

hard to calculate

Corr factor $= \frac{n}{n-k-1}$

Heteroskedasticity Robust Variance Estimator

Assumptions:
1) Correct specification
2) random sample

2.1-2 Stata Introduction and Interpreting Regression Results

Thursday, January 21, 2021 9:06 PM

Set more off = don't pause after every command

2.1-2 Stata Introduction and Interpreting Regression Results

Thursday, January 21, 2021 9:06 PM

Set more off = don't Pause after every command

2.3 Understanding Coefficient Variance

Thursday, January 21, 2021 9:54 PM

$$\widehat{Var}(\hat{\beta}) = (X^TX)^{-1}X^T\Sigma X(X^TX)^{-1}$$

$$\begin{bmatrix} \widehat{Var}(\hat{\beta}_1) & & \\ & \widehat{Var}(\hat{\beta}_2) & \\ & & \widehat{Var}(\hat{\beta}_n) \end{bmatrix}$$

$\widehat{Var}(\hat{\beta}_j) = s^2/[(1-\hat{R}_j)SST_j]$ → assumes homoscedasticity

↳ $s^2 = [\sum_i r_i^2]/[N-k-1]$

↳ $\hat{R}_j$ = regress x_j on all other x_i $(i \neq j)$ to get r^2

↳ $SST_j = \sum_i (x_j - \bar{x}_j)^2$ → total variability in x_j

$1/(1-\hat{R}_j^2)$ → variance inflation factor

2.3 Understanding Coefficient Variance

Thursday, January 21, 2021 9:54 PM

$$\widehat{Var}(\hat{\beta}) = (X^TX)^{-1}X^T \Sigma X(X^TX)^{-1}$$

$$\begin{bmatrix} \widehat{Var}(\hat{\beta}_1) & & \\ & \widehat{Var}(\hat{\beta}_2) & \\ & & \widehat{Var}(\hat{\beta}_n) \end{bmatrix}$$

$\widehat{Var}(\hat{\beta}_j) = s^2 / [(1-\hat{R}_j)SST_j]$ → assumes homoscedasticity

↳ $s^2 = [\sum_i r_i^2] / [N-k-1]$

↳ $\hat{R}_j$ = regress x_j on all other x_i $(i \neq j)$ to get r^2

↳ $SST_j = \sum_i (x_j - \bar{x}_j)^2$ → total variability in x_j

$1/(1-\hat{R}_j^2)$ → variance inflation factor

clear -> clears everything

set more *off* -> doesn't pause after every command

cd "path" -> changes the directory

log *using* "file name" -> creates a log file with that name

import *delimited* "path" -> imports comma delimited files (csv, tsv, whatever)

/* block comment ***/**
********* single line comment

scatter *y-var x-var* -> creates a scatterplot with the x and y vars specified

regress *y-var x-var* -> creates a simple linear regression with the x and y vars specified

STOP -> isn't a command but will stop the do file because it will throw and error and stop

regress *y-var x-var,* **robust** -> ["Robust regression is an alternative to least squares regression when data is contaminated with outliers or influential observations and it can also be used for the purpose of detecting influential observations." (Links to an external site.)]()

regress *y-var* **c.***x-var***##i.***x-var2*, **robust** -> same thing as before. c. = continuous variable, i. = indicator/dummy variable, ## = complete interaction between the x variables

testparm i.*y-var* **i.***y-var***#c.***x-var* <- test parameters associated with y-var and y-var that's interacted with x-var

1. Is there a difference between # and ##?
2. Does it have to be y-var y-var or can it be y-var any-var?

gen *newName* = stuff -> creates a new variable called newName using anything put after the =

corr *var1 var2* -> Find the correlation between the variables

vif -> when run after **corr** it gives the variance inflation factor

predict *var* <- predicts var

predict *var*, **residuals** <- plots against residuals

reg -> same as **regress**

log close -> closes the log

translate *log_file_name new_pdf_name*

2.5 Root Mean Square Error

Thursday, January 21, 2021 10:34 PM

$(y_i - \hat{y}_i)^2 / N = MSE = (y_i - \hat{y}_i)^2 / (N-k-1)$

precision vs accuracy/bias

10 Book Notes

Friday, January 22, 2021 12:05 PM

10-1 The Nature of Time Series Data

Data must go in order

A series of random variables indexed by time is a stochastic process or time series process
↑ random

10-2 Examples of Time Series Regression models

10-2a Static models

a static model time has an immediate effect
good for judging tradeoffs between y and z

10-2b Finite Distributed Lag models

FDL model

one or more variables can affect y w/ a lag

δ_0 is the impact propensity or impact multiplier

Lag distribution summarizes dynamic effect that a temporary increase in z has on y

Long-run propensity/multiplier (LRP)

for any horizon h, we can define the cumulative effect

10-2c A convention about the time index

Time starts at t=1

10-3 Finite Sample Properties of OLS under classical assumptions

10-3a Unbiasedness of OLS

Assumptions:
1) Linear in parameters
2) No perfect collinearity
3) Zero conditional mean
4) Homoskedasticity
5) No serial correlation
6) Normality

10-3b The Variance of the OLS estimators and the Gauss-Markov Theorem

Homoskedasticity:
Conditional on x, variance of u_t is same for all t
↳ $Var(u_t) = \sigma^2, t=1,2,\ldots,n$

No serial correlation:
Conditional on x, errors in two times are uncorrelated
↳ $Corr(u_t, u_s | X) = 0$, for all $t \neq s$
↳ suffer from serial/auto correlation when false

10-3c Inference under the classical linear model Assumptions

Normality:
errors u_t are independent of X and are independentantly and identically distributed as Normal$(0, \sigma^2)$

10 Time Series Econometrics for the 21st Century

Friday, January 22, 2021 11:09 AM

1 Introduction

Econometrics started with Time series data

2 Overview

Core models to learn:

1) Autoregressive model (AR) → $y_t = \alpha + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + e_t$
2) Regression model → $y_t = \alpha + \delta_0 x_t + e_t$
3) Distributed Lag (DL) model
   ↳ $y_t = \alpha + \delta_0 x_t + \delta_1 x_{t-1} + \ldots + \delta_q x_{t-q} + e_t$
4) Autoregressive-distributed Lag (ADL) model
   ↳ $y_t = \alpha + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \delta_0 x_t + \delta_1 x_{t-1} + \ldots + \delta_q x_{t-q} + e_t$

p = # lags of dependent var y_t
q = # of lags of explanatory var x_t
e_t = mean-zero shock

14! core insights!

3 Standard Errors and t Statistics

Newey standard error appropriate for simple (non-dynamic) regressions and DL models
↳ distributed Lag
↳ to be used when serial correlation has not been modelled

4 Autoregressive Models

5 Distributed Lag models

Estimate impact of one var on another

6 Autoregressive Distributed Lag models

7 Model selection

"Lag selection is inherently a bias-variance trade-off"

8 Spurious Regression

9 Structural Change

be wary of major shifts within data that throw off estimates

10 Forecasting

10.2 Time Series Basics

Friday, January 22, 2021 1:55 PM

$\{X_t : t = 1, 2, \ldots\}$

Time series data is a random sequential process

observe one outcome at a point in time

1) Past can impact the future
2) Outcomes are random, but obs not independent

Static Time series model

$Y_t = \beta_0 + \beta_1 X_t + \mu_t$

$\frac{rev}{exp}_t = \beta_0 + \beta_1 rgdpc_t + \mu_t \longrightarrow rGDPC = gdp / gdpdef \cdot pop$

revenue / expenditure

exp$_t$: depend on what we want income↑, more depends on available revenue

rev$_t$: avg tax rate · tax base t

income

10.3 Time Trends

Sunday, January 31, 2021 6:16 PM

past affects future so std error isn't right

spurious correlations!

linear trends by regressing time and a variable

Time trends

$y_t = \alpha_0 + \alpha_1 t + r_{1t}$ $\qquad x_t = \gamma_0 + \gamma_1 t + r_{xt}$

$\hat{y}_t = \hat{\alpha}_0 + \hat{\alpha}_{1t}$

$\hat{r}_{it} = (y_t - \hat{\alpha}_0 - \hat{\alpha}_{1t})$

suppose $y_t = \beta_0 + \beta_1 x_t + \delta t + r_t$

How is y correlated with x?

detrending = put in t trend take residuals afterwards

differencing =

Lags:

$L.y_t = y_{t-1}$ $\qquad f.y_t = y_{t+1}$

$L2.y_t = y_{t-2}$

$D.y_t = y_t - y_{t-1}$

differencing

$y_t = \beta_0 + \beta_1 x_t + \delta t + r_t$

$y_{t-1} = \beta_0 + \beta_1 x_{t-1} + \delta(t-1) + r_{t-1}$

$\hookrightarrow d.y_t = y_t - y_{t-1} = \delta + \beta_1(x_t - x_{t-1}) + (r_t - r_{t-1})$

10.4 Seasonal Trends

Sunday, January 31, 2021 7:29 PM

add quarterly dummy variables to adjust for seasons

$$Y_t = \beta_0 + \beta_1 X_t + \delta t + \underbrace{\gamma_{sp} + \gamma_{sum} + \gamma_{fa}}_{\text{Seasons}} + u_t$$

Why does lag matter with seasons?

10.5-6

Sunday, January 31, 2021 5:27 PM

$\hat{y}_t = -0.46 - 0.22x_t + 0.65x_{t-1} - 0.003t + 0.04sp + .001sum + .012fall$

lag

spring summer fall

maybe lag matters more?

$\hat{y} = .15x_t + 1.56x_{t-1} + .67x_{t-2} - .63x_{t-3} - 1.5x_{t-4}$

4 lags is one cycle

ssc install estout

lag: l(x/y).

L time x to y dot

esttab → recommended for exporting

different # observations because data is burned to calculate lag

10.7

Sunday, January 31, 2021 6:05 PM

$Y_t = \beta_0 + \beta_1 X_t + \beta_2 X_{t-1} + \beta_3 X_{t-2}$

$= \alpha + \beta_0 X_{t-0} + \beta_1 X_{t-1} + \beta_3 X_{t-3} + \dots$

intercept

$= \alpha + \sum_{\ell} \beta_\ell X_{t-\ell}$

$Y_t =$ Total N

$X_t =$ Emp by industry of interest

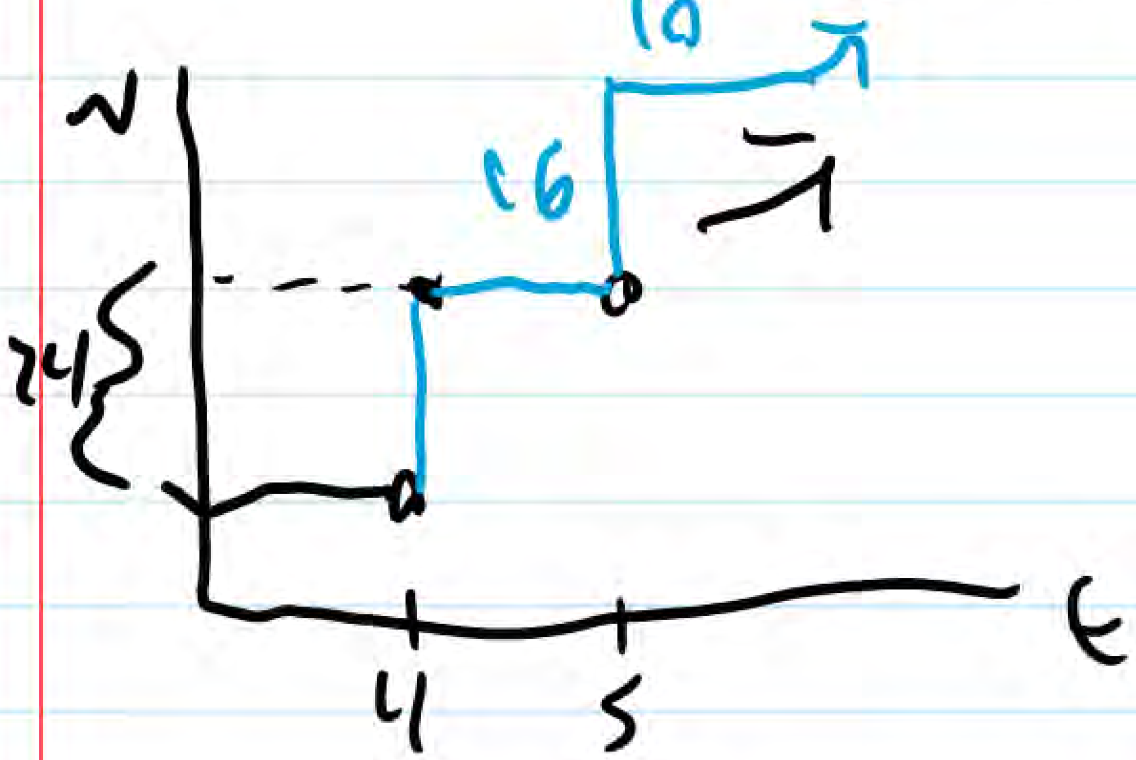
$N_t = \alpha + \beta_0 N_{It} + \beta_1 N_{It-1} + \beta_2 N_{It-2} + \dots$

takes 3 years

$N_t = 100 + 1.2 N_{It} + .8 N_{It-1} + .5 N_{It-2}$

$\Delta N_t = 1.2 \Delta N_{It} + .8 \Delta N_{It-1} + .5 \Delta N_{It-2}$

$t=4 \quad \Delta N_I = 20$



$\rightarrow \Delta N = 50$

Long run effect $= 1.2 + .8 + .5 = 2.5$

18 Book Notes

Wednesday, February 3, 2021 6:56 PM

Advanced Time Series Topics

18.3 Spurious regression

There's an extraneous reason for correlation

18.2 Testing For Unit roots

Unit root process w/ drift very different from one without

Unit root hypothesis

$H_0: \rho = 1$

$H_a: \rho < 1$

$\theta = \rho - 1$

$\Delta y_t = \alpha + \theta y_{t-1} + e_t$

asymptotic distribution of t statistic under Ho is Dickey-Fuller Distribution

Problem Set 1

Gus Lipkin

CAP 4763 Time Series Modelling and Forecasting

> Corrections are underlined

> All uncited quotes are from the Problem Set 1 official solution

Table of Contents

3 Static Model

3a

> Explain why the size of Florida's labor force, the prime age employment to population ratio, and Florida building permits, might be closely related to the number of nonfarm jobs in Florida in a static long run sense. You might want to make some time series plots to give your data context. (Perhaps where one variable is employment and the other, on the other axis, is one of the other variables.)

The size of Florida's labor force can only increase for a few reasons. People either grow up and get a job or people move into the state for one reason or another. These would increase the prime age employment to population ratio but those people need places to work. They could either work in construction or any affiliated field which handles building permits or they could work in a building being constructed by the people handling those permits. In the meantime, as farming becomes more efficient and reliant on technology, not as many people are needed to farm the same parcels of land. This leads to more people employed in non-farm jobs.

"We can think of the number employed as the product of the portion of those in the labor market that are employed and the number that want work and so are in the market. Then the log of total employment is the sum of the logs of those two pieces. From there:

- The number that want to work should closely track labor force in Florida.
- The fraction of those that want to be employed that are employed tracks the strength of the Florida economy, which closely tracks the strength of the national economy, for which the employment to population ratio is a good proxy.
- Construction is a large part of Florida's economic base, due to constant in-migration. So, variations in the strength of the economy may be reflected somewhat in building permits."

3b

> Estimate the static model relating monthly nonfarm employment in Florida to the other three variables (all in logs) without controlling for seasonal impacts or a time trend.

Source	SS	df	MS Number of obs =	396
	F(3, 392) =	5972.65		
Model	10.5356085	3	3.51186951 Prob > F =	0.0000
Residual	.230492978	392	.000587992 R-squared =	0.9786
	Adj R-squared =	0.9784		
Total	10.7661015	395	.027255953 Root MSE =	.02425
ln_fl_nonf~m	Coef.	Std. Err.	t P>t [95% Conf.	Interval]
ln_fl_lf	1.110504	.0092305	120.31 0.000 1.092356	1.128651
ln_us_epr	.6006702	.047797	12.57 0.000 .5066997	.6946407
ln_fl_bp	.0516831	.0028713	18.00 0.000 .0460379	.0573282
_cons	-11.78364	.2925244	-40.28 0.000 -12.35875	-11.20852

3c

> Estimate the static model with month indicators and a time trend.

Source	SS	df	MS Number of obs =	396
	F(15, 380) =	2935.69		
Model	10.6739911	15	.711599408 Prob > F =	0.0000
Residual	.092110398	380	.000242396 R-squared =	0.9914
	Adj R-squared =	0.9911		
Total	10.7661015	395	.027255953 Root MSE =	.01557
ln_fl_nonf~m	Coef.	Std. Err.	t P>t [95% Conf.	Interval]
ln_fl_lf	.9282631	.0413265	22.46 0.000 .8470059	1.00952

ln_us_epr	.9105558	.0514333	17.70 0.000 .8094263	1.011685
ln_fl_bp	.0466812	.0021579	21.63 0.000 .0424382	.0509242
month				
2	.0045623	.0038378	1.19 0.235 -.0029837	.0121084
3	-.001379	.003839	-0.36 0.720 -.0089274	.0061694
4	-.0029373	.0038393	-0.77 0.445 -.0104863	.0046116
5	-.0142748	.0038468	-3.71 0.000 -.0218384	-.0067112
6	-.0356123	.0038709	-9.20 0.000 -.0432234	-.0280012
7	-.0519102	.0038917	-13.34 0.000 -.0595622	-.0442582
8	-.0380965	.0038668	-9.85 0.000 -.0456995	-.0304936
9	-.026004	.0038581	-6.74 0.000 -.0335899	-.0184181
10	-.0215894	.0038763	-5.57 0.000 -.029211	-.0139678
11	-.0014672	.0039082	-0.38 0.708 -.0091517	.0062173
12	.0054514	.0038735	1.41 0.160 -.0021648	.0130675
date	.0003124	.0000637	4.90 0.000 .000187	.0004377
_cons	-10.26323	.498888	-20.57 0.000 -11.24416	-9.282304

3d

> Compare your results from b and c and interpret any differences. What do the seasonal and time trend variables contribute?

Adding the seasonal and time trend variables transform the data into true time series data and give context to the changes. From both you can see that there is a general increase in nonfarm employment. However, by adding the month indicators, you can see that nonfarm employment decreases ever so slightly from March to November, presumably due to prime farming season. "All three coefficients change slightly. The time trend controls for growth at a constant rate over time, while the month indicators control for seasonality. For example, construction employment varies with the weather, employment always varies with holidays, and in Florida employment also varies with tourist season. Presumably, controlling for these effects allows the model to better reveal the underlying relationships between the other variables. (The caveat is we have not checked this data for stationarity or weak dependence, which comes later.)"

3e

> Why should you be cautious using the results of these models for testing any hypotheses about the underlying relationships?

In time series data, the past affects the future and observations are not independent. Standard error and p-value assume that your data is independent which we just established time series data is not.

4 Finite Distributed Lag Model

4a

> Estimate the distributed lag model relating monthly nonfarm employment to lags 0 to 12 of the three predictor variables without month indicators and a time trend.

Source	SS	df	MS Number of obs =	384
	F(39, 344) =	1506.36		
Model	9.45063897	39	.242324076 Prob > F =	0.0000
Residual	.055338456	344	.000160868 R-squared =	0.9942
	Adj R-squared =	0.9935		
Total	9.50597742	383	.024819784 Root MSE =	.01268
ln_fl_nonf~m	Coef.	Std. Err.	t P>t [95% Conf.	Interval]
ln_fl_lf				
-.	-.3180953	.2192272	-1.45 0.148 -.7492898	.1130992
L1.	-.4936055	.2780395	-1.78 0.077 -1.040477	.0532661
L2.	.3085466	.27846	1.11 0.269 -.239152	.8562452
L3.	1.173922	.2948363	3.98 0.000 .5940134	1.753831
L4.	-.2346487	.2905929	-0.81 0.420 -.8062113	.3369138
L5.	.2808166	.2958343	0.95 0.343 -.3010552	.8626884
L6.	-.2076341	.3372426	-0.62 0.539 -.8709511	.4556829
L7.	.428488	.3391507	1.26 0.207 -.2385821	1.095558

L8.	.4803611	.3332665	1.44 0.150 -.1751354	1.135858
L9.	.2977526	.3112925	0.96 0.339 -.3145235	.9100288
L10.	-.00028	.3217814	-0.00 0.999 -.6331867	.6326267
L11.	-.5860114	.3256137	-1.80 0.073 -1.226456	.0544331
L12.	.0176351	.2499574	0.07 0.944 -.4740021	.5092724
ln_us_epr				
-.	1.180441	.1573579	7.50 0.000 .8709364	1.489946
L1.	.2435207	.202013	1.21 0.229 -.1538155	.6408569
L2.	-.1519264	.2015081	-0.75 0.451 -.5482695	.2444166
L3.	-.719111	.2119425	-3.39 0.001 -1.135977	-.3022447
L4.	.1877102	.2014654	0.93 0.352 -.2085489	.5839692
L5.	-.1596306	.206881	-0.77 0.441 -.5665414	.2472803
L6.	.4937537	.2396216	2.06 0.040 .0224458	.9650615
L7.	-.3031484	.236988	-1.28 0.202 -.7692764	.1629796
L8.	-.2995254	.2312056	-1.30 0.196 -.7542801	.1552293
L9.	.5953076	.2915942	2.04 0.042 .0217756	1.16884
L10.	-.1656984	.352639	-0.47 0.639 -.8592984	.5279015
L11.	.5326939	.3523697	1.51 0.132 -.1603764	1.225764
L12.	-.4280274	.2543508	-1.68 0.093 -.928306	.0722511
ln_fl_bp				
-.	.0177815	.0051888	3.43 0.001 .0075758	.0279872
L1.	.0056999	.0054688	1.04 0.298 -.0050566	.0164565
L2.	.0123023	.0056879	2.16 0.031 .0011149	.0234898
L3.	-.0005041	.0058381	-0.09 0.931 -.0119871	.0109788
L4.	-.0040248	.0058282	-0.69 0.490 -.0154881	.0074385

L5.	.0053648	.0058106	0.92 0.357 -.006064	.0167937
L6.	.0122019	.0057914	2.11 0.036 .0008108	.0235929
L7.	.0146252	.0057698	2.53 0.012 .0032766	.0259737
L8.	.0114715	.0057663	1.99 0.047 .0001299	.0228131
L9.	.0100892	.0057895	1.74 0.082 -.0012981	.0214765
L10.	-.0077443	.0056515	-1.37 0.171 -.0188601	.0033715
L11.	-.0129284	.0055227	-2.34 0.020 -.0237908	-.002066
L12.	-.0156324	.0052843	-2.96 0.003 -.0260261	-.0052388
_cons	-14.00483	.220126	-63.62 0.000 -14.43779	-13.57187

4b

> Estimate the model in (a) but add month indicators and a time trend.

Source	SS	df	MS Number of obs =	384
	F(51, 332) =	1880.48		
Model	9.47318331	51	.185748692 Prob > F =	0.0000
Residual	.03279411	332	.000098777 R-squared =	0.9966
	Adj R-squared =	0.9960		
Total	9.50597742	383	.024819784 Root MSE =	.00994
ln_fl_nonf~m	Coef.	Std. Err.	t P>t [95% Conf.	Interval]
ln_fl_lf				
-.	.1395258	.2167149	0.64 0.520 -.2867817	.5658333
L1.	-.0728475	.2909974	-0.25 0.802 -.6452787	.4995837
L2.	-.0401378	.2914261	-0.14 0.891 -.6134123	.5331367
L3.	.4941867	.3004728	1.64 0.101 -.096884	1.085257
L4.	.0243743	.3032608	0.08 0.936 -.5721806	.6209291

L5.	-.0515457	.3007867	-0.17 0.864 -.6432337	.5401424
L6.	.2645611	.3042172	0.87 0.385 -.3338753	.8629975
L7.	.3032209	.3064496	0.99 0.323 -.2996069	.9060486
L8.	.0945934	.3058001	0.31 0.757 -.5069567	.6961435
L9.	-.1097755	.3559336	-0.31 0.758 -.8099451	.590394
L10.	.1539543	.375505	0.41 0.682 -.5847148	.8926234
L11.	-.2776778	.3787638	-0.73 0.464 -1.022757	.4674017
L12.	-.0112724	.279864	-0.04 0.968 -.5618026	.5392579
ln_us_epr				
-.	.8902343	.1499136	5.94 0.000 .595334	1.185135
L1.	.0725186	.1976025	0.37 0.714 -.3161923	.4612294
L2.	.0146862	.1973291	0.07 0.941 -.3734868	.4028593
L3.	-.3099001	.2109421	-1.47 0.143 -.7248517	.1050514
L4.	.137028	.215249	0.64 0.525 -.2863958	.5604519
L5.	-.0073661	.2142714	-0.03 0.973 -.4288668	.4141346
L6.	.0293898	.2200462	0.13 0.894 -.4034709	.4622504
L7.	-.1397223	.2227059	-0.63 0.531 -.5778149	.2983702
L8.	-.0598893	.2228997	-0.27 0.788 -.4983631	.3785844
L9.	.4823653	.4060878	1.19 0.236 -.3164642	1.281195
L10.	.0335197	.4684115	0.07 0.943 -.887909	.9549485
L11.	.4443457	.4733678	0.94 0.349 -.4868327	1.375524
L12.	-.3652099	.3457533	-1.06 0.292 -1.045353	.3149335
ln_fl_bp				
-.	.0174185	.0043812	3.98 0.000 .0088	.0260369
L1.	.0097915	.0047176	2.08 0.039 .0005113	.0190717

L2.	.005989	.0048174	1.24 0.215 -.0034873	.0154654
L3.	.0067099	.0049382	1.36 0.175 -.0030042	.016424
L4.	.0015463	.0049663	0.31 0.756 -.0082232	.0113157
L5.	.0025978	.0049914	0.52 0.603 -.007221	.0124166
L6.	.006001	.0049798	1.21 0.229 -.0037949	.0157968
L7.	.0066017	.0049157	1.34 0.180 -.003068	.0162715
L8.	-.0015491	.0049371	-0.31 0.754 -.011261	.0081628
L9.	.0010036	.0048898	0.21 0.838 -.0086153	.0106225
L10.	-.0004773	.0047767	-0.10 0.920 -.0098737	.008919
L11.	-.0083937	.0046846	-1.79 0.074 -.017609	.0008216
L12.	-.0041455	.0044702	-0.93 0.354 -.0129391	.004648
month				
2	.0077995	.0048077	1.62 0.106 -.001658	.017257
3	.0052085	.0041637	1.25 0.212 -.0029821	.0133991
4	-.0010198	.0053356	-0.19 0.849 -.0115156	.009476
5	-.0012298	.0047478	-0.26 0.796 -.0105694	.0081098
6	-.0122415	.0055844	-2.19 0.029 -.0232267	-.0012563
7	-.0240128	.0047031	-5.11 0.000 -.0332644	-.0147612
8	-.0152756	.0052483	-2.91 0.004 -.0255997	-.0049514
9	-.0111308	.0045365	-2.45 0.015 -.0200548	-.0022068
10	-.0046899	.006722	-0.70 0.486 -.0179129	.0085332
11	.0076979	.0057763	1.33 0.184 -.0036649	.0190607
12	.0151789	.0059337	2.56 0.011 .0035065	.0268514
date	.0003695	.000047	7.86 0.000 .000277	.0004619
_cons	-11.28083	.391293	-28.83 0.000 -12.05055	-10.5111

4d

> Compare your results from a and c and interpret any differences. What do the seasonal and time trend variables contribute?

The model in 4a is accurate to the data it was given but does not make sense and has no practical application because the data is not organized in any way and does not account for the data being time series data. "This is largely the same as it was for question 3. The difference is that since we are controlling for one year ago, the lags themselves may capture some of the seasonal difference in the first model, and that adding seasonal effects purges that, changing the results potentially at all lags. This, though, it just more of the same basic thing."

4e

> Estimate two alternative models that contain month indicators and a time trend but that impose a more parsimonious lag structure for the predictor variables. Explain your choices.

4e Sampling each quarter

Source	SS	df	MS Number of obs =	384
	F(24, 359) =	3636.67		
Model	9.46703767	24	.394459903 Prob > F =	0.0000
Residual	.038939751	359	.000108467 R-squared =	0.9959
	Adj R-squared =	0.9956		
Total	9.50597742	383	.024819784 Root MSE =	.01041
ln_fl_nonf~m	Coef.	Std. Err.	t P>t [95% Conf.	Interval]
ln_fl_lf				
-.	.2198644	.118892	1.85 0.065 -.0139479	.4536767
L4.	.3640379	.1628088	2.24 0.026 .0438591	.6842168
L8.	.6241057	.1697337	3.68 0.000 .2903084	.957903
L12.	-.3365352	.1300465	-2.59 0.010 -.592284	-.0807864

ln_us_epr				
-.	.8706823	.0862833	10.09 0.000 .7009981	1.040367
L4.	.0186581	.1180743	0.16 0.875 -.213546	.2508623
L8.	-.1364675	.1363531	-1.00 0.318 -.4046187	.1316838
L12.	.4492816	.1055542	4.26 0.000 .2416993	.6568639
ln_fl_bp				
-.	.0288326	.0033225	8.68 0.000 .0222986	.0353666
L4.	.014784	.0040692	3.63 0.000 .0067816	.0227864
L8.	.0053046	.0040599	1.31 0.192 -.0026795	.0132888
L12.	-.0040886	.0034865	-1.17 0.242 -.0109452	.002768
month				
2	.003724	.0027268	1.37 0.173 -.0016384	.0090864
3	.003428	.0030747	1.11 0.266 -.0026188	.0094747
4	-.0013812	.0030302	-0.46 0.649 -.0073404	.0045779
5	-.0050709	.003101	-1.64 0.103 -.0111693	.0010275
6	-.0215379	.0030889	-6.97 0.000 -.0276125	-.0154633
7	-.0356678	.0033321	-10.70 0.000 -.0422208	-.0291149
8	-.0202856	.0032905	-6.16 0.000 -.0267567	-.0138145
9	-.0118143	.0031977	-3.69 0.000 -.0181028	-.0055257
10	-.0142884	.0031129	-4.59 0.000 -.0204102	-.0081666
11	-.0033333	.0030634	-1.09 0.277 -.0093578	.0026912
12	.0070509	.0028963	2.43 0.015 .001355	.0127468
date	.0004262	.0000476	8.96 0.000 .0003326	.0005197
_cons	-10.60852	.3857432	-27.50 0.000 -11.36712	-9.849916

4e True Quarters

Source	SS	df	MS Number of obs =	392
	F(27, 364) =	2505.01		
Model	10.2740552	27	.380520563 Prob > F =	0.0000
Residual	.055292923	364	.000151904 R-squared =	0.9946
	Adj R-squared =	0.9942		
Total	10.3293481	391	.02641777 Root MSE =	.01232
ln_fl_nonf~m	Coef.	Std. Err.	t P>t [95% Conf.	Interval]
ln_fl_lf				
-.	.2790757	.2536131	1.10 0.272 -.2196552	.7778065
L1.	.2956093	.3348151	0.88 0.378 -.3628055	.9540241
L2.	-.2641756	.3153608	-0.84 0.403 -.8843334	.3559822
L3.	.2832334	.3167687	0.89 0.372 -.339693	.9061598
L4.	.3220421	.2469667	1.30 0.193 -.1636186	.8077029
ln_us_epr				
-.	.869919	.1763402	4.93 0.000 .5231456	1.216693
L1.	-.1508318	.2303364	-0.65 0.513 -.6037889	.3021253
L2.	.1899043	.2170821	0.87 0.382 -.2369882	.6167968
L3.	-.2262386	.2208671	-1.02 0.306 -.6605744	.2080971
L4.	.3389032	.1751932	1.93 0.054 -.0056147	.6834212
ln_fl_bp				
-.	.0204443	.0051347	3.98 0.000 .010347	.0305417
L1.	.0107528	.0054657	1.97 0.050 4.39e-06	.0215012

L2.	.0026867	.0054899	0.49 0.625 -.0081091	.0134826
L3.	.0070439	.0054993	1.28 0.201 -.0037706	.0178583
L4.	.0071123	.0051692	1.38 0.170 -.003053	.0172777
month				
2	.0052225	.0038186	1.37 0.172 -.0022868	.0127318
3	.0086375	.0041006	2.11 0.036 .0005735	.0167014
4	.0012736	.0046541	0.27 0.785 -.0078787	.0104258
5	.0022027	.0038771	0.57 0.570 -.0054216	.0098269
6	-.0193223	.0040672	-4.75 0.000 -.0273206	-.0113241
7	-.0362039	.0038883	-9.31 0.000 -.0438502	-.0285575
8	-.0245188	.0043528	-5.63 0.000 -.0330787	-.015959
9	-.0171602	.0037189	-4.61 0.000 -.0244733	-.0098471
10	-.0193132	.0044175	-4.37 0.000 -.0280001	-.0106262
11	-.004866	.0041178	-1.18 0.238 -.0129637	.0032317
12	.0058531	.0039007	1.50 0.134 -.0018177	.0135238
dateQ	.0010375	.0001584	6.55 0.000 .0007259	.001349
_cons	-10.55687	.4171884	-25.30 0.000 -11.37727	-9.736468

4e Explanation

I was curious to know how sampling lag for a single month from each quarter for a year would compare to generating a new quarter date variable and using that for lag. Unfortunately, I don't think I did it right and I don't know how to get what I want. Instead, what I have for the second chart is quarterly dates but the lagged variables are now only lagged for the first four months of the year.

Based on the MSE of each model, the first one is a little bit better but I don't think either is great.

"The most important lags would seem to be the most recent month, and the same month a year ago."

Appendix A

```
clear
set more off

cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Problem Set 1"

*2b Load the data
import delimited "Assignment_1_Monthly.txt"

rename lnu02300000 us_epr
rename flnan fl_nonfarm
rename fllfn fl_lf
rename flbppriv fl_bp
rename date datestring

*2c Turn on a log file
log using "Problem Set 1", replace

*2d Generate a monthly date variable (make its display format monthly time, %tm)
gen datec=date(datestring, "YMD")
gen date=mofd(datec)
format date %tm

*2e tsset your data
tsset date

*2f
gen ln_us_epr=log(us_epr)
gen ln_fl_nonfarm=log(fl_nonfarm)
gen ln_fl_lf=log(fl_lf)
gen ln_fl_bp=log(fl_bp)

*3b Estimate the static model relating monthly nonfarm employment in Florida to the other three variables (all in logs) without controlling for seasonal impacts or a time trend.
regress ln_fl_nonfarm ln_fl_lf ln_us_epr ln_fl_bp

*3c Estimate the static model with month indicators and a time trend.
gen month=month(datec)
reg ln_fl_nonfarm ln_fl_lf ln_us_epr ln_fl_bp i.month date

*4a Estimate the distributed lag model relating monthly nonfarm employment to lags 0 to 12 of the three predictor variables without month indicators and a time trend.
regress ln_fl_nonfarm l(0/12).ln_fl_lf l(0/12).ln_us_epr l(0/12).ln_fl_bp

*4b Estimate the model in (a) but add month indicators and a time trend.
```

```
regress ln_fl_nonfarm l(0/12).ln_fl_lf l(0/12).ln_us_epr l(0/12).ln_fl_bp i.month
date

*4e Estimate two alternative models that contain month indicators and a time trend
but that impose a more parsimonious lag structure for the predictor variables.
Explain your choices.
regress ln_fl_nonfarm l(0,4,8,12).ln_fl_lf l(0,4,8,12).ln_us_epr l(0,4,8,12).ln_fl_bp
i.month date
gen dateQ = qofd(datec)
format dateQ %tq
regress ln_fl_nonfarm l(0/4).ln_fl_lf l(0/4).ln_us_epr l(0/4).ln_fl_bp i.month dateQ

log close
```

Appendix B

```
      name:  <unnamed>
       log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/P
> roblem Set 1/Problem Set 1.smcl
  log type:  smcl
 opened on:  11 Feb 2021, 19:36:36

.
. *2d Generate a monthly date variable (make its display format monthly time, %tm)
. gen datec=date(datestring, "YMD")

. gen date=mofd(datec)

. format date %tm

.
. *2e tsset your data
. tsset date
        time variable:  date, 1939m1 to 2020m12
                delta:  1 month

.
. *2f
. gen ln_us_epr=log(us_epr)
(108 missing values generated)

. gen ln_fl_nonfarm=log(fl_nonfarm)

. gen ln_fl_lf=log(fl_lf)
(444 missing values generated)

. gen ln_fl_bp=log(fl_bp)
(588 missing values generated)

.
. *3b Estimate the static model relating monthly nonfarm employment in Florida to the ot
> her three variables (all in logs) without controlling for seasonal impacts or a time t
> rend.
```

```
. regress ln_fl_nonfarm ln_fl_lf ln_us_epr ln_fl_bp

      Source |       SS           df       MS      Number of obs   =       396
-------------+----------------------------------   F(3, 392)       =   5972.65
       Model |  10.5356085         3  3.51186951   Prob > F        =    0.0000
    Residual |  .230492978       392  .000587992   R-squared       =    0.9786
-------------+----------------------------------   Adj R-squared   =    0.9784
       Total |  10.7661015       395  .027255953   Root MSE        =    .02425

------------------------------------------------------------------------------
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_lf |   1.110504   .0092305   120.31   0.000     1.092356    1.128651
   ln_us_epr |   .6006702    .047797    12.57   0.000     .5066997    .6946407
    ln_fl_bp |   .0516831   .0028713    18.00   0.000     .0460379    .0573282
       _cons |  -11.78364   .2925244   -40.28   0.000    -12.35875   -11.20852
------------------------------------------------------------------------------

. 
. *3c Estimate the static model with month indicators and a time trend.
. gen month=month(datec)

. reg ln_fl_nonfarm ln_fl_lf ln_us_epr ln_fl_bp i.month date

      Source |       SS           df       MS      Number of obs   =       396
-------------+----------------------------------   F(15, 380)      =   2935.69
       Model |  10.6739911        15  .711599408   Prob > F        =    0.0000
    Residual |  .092110398       380  .000242396   R-squared       =    0.9914
-------------+----------------------------------   Adj R-squared   =    0.9911
       Total |  10.7661015       395  .027255953   Root MSE        =    .01557

------------------------------------------------------------------------------
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_lf |   .9282631   .0413265    22.46   0.000     .8470059     1.00952
   ln_us_epr |   .9105558   .0514333    17.70   0.000     .8094263    1.011685
    ln_fl_bp |   .0466812   .0021579    21.63   0.000     .0424382    .0509242
             |
       month |
          2  |   .0045623   .0038378     1.19   0.235    -.0029837    .0121084
          3  |   -.001379    .003839    -0.36   0.720    -.0089274    .0061694
          4  |  -.0029373   .0038393    -0.77   0.445    -.0104863    .0046116
          5  |  -.0142748   .0038468    -3.71   0.000    -.0218384   -.0067112
          6  |  -.0356123   .0038709    -9.20   0.000    -.0432234   -.0280012
          7  |  -.0519102   .0038917   -13.34   0.000    -.0595622   -.0442582
          8  |  -.0380965   .0038668    -9.85   0.000    -.0456995   -.0304936
          9  |   -.026004   .0038581    -6.74   0.000    -.0335899   -.0184181
         10  |  -.0215894   .0038763    -5.57   0.000     -.029211   -.0139678
         11  |  -.0014672   .0039082    -0.38   0.708    -.0091517    .0062173
         12  |   .0054514   .0038735     1.41   0.160    -.0021648    .0130675
             |
        date |   .0003124   .0000637     4.90   0.000      .000187    .0004377
       _cons |  -10.26323    .498888   -20.57   0.000    -11.24416   -9.282304
------------------------------------------------------------------------------

. 
. *4a Estimate the distributed lag model relating monthly nonfarm employment to lags 0 t
```

> o 12 of the three predictor variables without month indicators and a time trend.
. regress ln_fl_nonfarm l(0/12).ln_fl_lf l(0/12).ln_us_epr l(0/12).ln_fl_bp

Source	SS	df	MS
Model	9.45063897	39	.242324076
Residual	.055338456	344	.000160868
Total	9.50597742	383	.024819784

Number of obs	=	384
F(39, 344)	=	1506.36
Prob > F	=	0.0000
R-squared	=	0.9942
Adj R-squared	=	0.9935
Root MSE	=	.01268

ln_fl_nonf~m	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
ln_fl_lf						
--.	-.3180953	.2192272	-1.45	0.148	-.7492898	.1130992
L1.	-.4936055	.2780395	-1.78	0.077	-1.040477	.0532661
L2.	.3085466	.27846	1.11	0.269	-.239152	.8562452
L3.	1.173922	.2948363	3.98	0.000	.5940134	1.753831
L4.	-.2346487	.2905929	-0.81	0.420	-.8062113	.3369138
L5.	.2808166	.2958343	0.95	0.343	-.3010552	.8626884
L6.	-.2076341	.3372426	-0.62	0.539	-.8709511	.4556829
L7.	.428488	.3391507	1.26	0.207	-.2385821	1.095558
L8.	.4803611	.3332665	1.44	0.150	-.1751354	1.135858
L9.	.2977526	.3112925	0.96	0.339	-.3145235	.9100288
L10.	-.00028	.3217814	-0.00	0.999	-.6331867	.6326267
L11.	-.5860114	.3256137	-1.80	0.073	-1.226456	.0544331
L12.	.0176351	.2499574	0.07	0.944	-.4740021	.5092724
ln_us_epr						
--.	1.180441	.1573579	7.50	0.000	.8709364	1.489946
L1.	.2435207	.202013	1.21	0.229	-.1538155	.6408569
L2.	-.1519264	.2015081	-0.75	0.451	-.5482695	.2444166
L3.	-.719111	.2119425	-3.39	0.001	-1.135977	-.3022447
L4.	.1877102	.2014654	0.93	0.352	-.2085489	.5839692
L5.	-.1596306	.206881	-0.77	0.441	-.5665414	.2472803
L6.	.4937537	.2396216	2.06	0.040	.0224458	.9650615
L7.	-.3031484	.236988	-1.28	0.202	-.7692764	.1629796
L8.	-.2995254	.2312056	-1.30	0.196	-.7542801	.1552293
L9.	.5953076	.2915942	2.04	0.042	.0217756	1.16884
L10.	-.1656984	.352639	-0.47	0.639	-.8592984	.5279015
L11.	.5326939	.3523697	1.51	0.132	-.1603764	1.225764
L12.	-.4280274	.2543508	-1.68	0.093	-.928306	.0722511
ln_fl_bp						
--.	.0177815	.0051888	3.43	0.001	.0075758	.0279872
L1.	.0056999	.0054688	1.04	0.298	-.0050566	.0164565
L2.	.0123023	.0056879	2.16	0.031	.0011149	.0234898
L3.	-.0005041	.0058381	-0.09	0.931	-.0119871	.0109788
L4.	-.0040248	.0058282	-0.69	0.490	-.0154881	.0074385
L5.	.0053648	.0058106	0.92	0.357	-.006064	.0167937
L6.	.0122019	.0057914	2.11	0.036	.0008108	.0235929
L7.	.0146252	.0057698	2.53	0.012	.0032766	.0259737
L8.	.0114715	.0057663	1.99	0.047	.0001299	.0228131
L9.	.0100892	.0057895	1.74	0.082	-.0012981	.0214765
L10.	-.0077443	.0056515	-1.37	0.171	-.0188601	.0033715
L11.	-.0129284	.0055227	-2.34	0.020	-.0237908	-.002066
L12.	-.0156324	.0052843	-2.96	0.003	-.0260261	-.0052388

```
       _cons |  -14.00483    .220126   -63.62   0.000    -14.43779   -13.57187
------------------------------------------------------------------------------

.
. *4b Estimate the model in (a) but add month indicators and a time trend.
. regress ln_fl_nonfarm l(0/12).ln_fl_lf l(0/12).ln_us_epr l(0/12).ln_fl_bp i.month date

      Source |       SS           df       MS      Number of obs   =       384
-------------+----------------------------------   F(51, 332)      =   1880.48
       Model |  9.47318331        51  .185748692   Prob > F        =    0.0000
    Residual |   .03279411       332  .000098777   R-squared       =    0.9966
-------------+----------------------------------   Adj R-squared   =    0.9960
       Total |  9.50597742       383  .024819784   Root MSE        =    .00994

------------------------------------------------------------------------------
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_lf |
         --. |   .1395258   .2167149     0.64   0.520    -.2867817    .5658333
         L1. |  -.0728475   .2909974    -0.25   0.802    -.6452787    .4995837
         L2. |  -.0401378   .2914261    -0.14   0.891    -.6134123    .5331367
         L3. |   .4941867   .3004728     1.64   0.101     -.096884    1.085257
         L4. |   .0243743   .3032608     0.08   0.936    -.5721806    .6209291
         L5. |  -.0515457   .3007867    -0.17   0.864    -.6432337    .5401424
         L6. |   .2645611   .3042172     0.87   0.385    -.3338753    .8629975
         L7. |   .3032209   .3064496     0.99   0.323    -.2996069    .9060486
         L8. |   .0945934   .3058001     0.31   0.757    -.5069567    .6961435
         L9. |  -.1097755   .3559336    -0.31   0.758    -.8099451     .590394
        L10. |   .1539543    .375505     0.41   0.682    -.5847148    .8926234
        L11. |  -.2776778   .3787638    -0.73   0.464    -1.022757    .4674017
        L12. |  -.0112724    .279864    -0.04   0.968    -.5618026    .5392579
             |
   ln_us_epr |
         --. |   .8902343   .1499136     5.94   0.000      .595334    1.185135
         L1. |   .0725186   .1976025     0.37   0.714    -.3161923    .4612294
         L2. |   .0146862   .1973291     0.07   0.941    -.3734868    .4028593
         L3. |  -.3099001   .2109421    -1.47   0.143    -.7248517    .1050514
         L4. |    .137028    .215249     0.64   0.525    -.2863958    .5604519
         L5. |  -.0073661   .2142714    -0.03   0.973    -.4288668    .4141346
         L6. |   .0293898   .2200462     0.13   0.894    -.4034709    .4622504
         L7. |  -.1397223   .2227059    -0.63   0.531    -.5778149    .2983702
         L8. |  -.0598893   .2228997    -0.27   0.788    -.4983631    .3785844
         L9. |   .4823653   .4060878     1.19   0.236    -.3164642    1.281195
        L10. |   .0335197   .4684115     0.07   0.943     -.887909    .9549485
        L11. |   .4443457   .4733678     0.94   0.349    -.4868327    1.375524
        L12. |  -.3652099   .3457533    -1.06   0.292    -1.045353    .3149335
             |
    ln_fl_bp |
         --. |   .0174185   .0043812     3.98   0.000        .0088    .0260369
         L1. |   .0097915   .0047176     2.08   0.039     .0005113    .0190717
         L2. |    .005989   .0048174     1.24   0.215    -.0034873    .0154654
         L3. |   .0067099   .0049382     1.36   0.175    -.0030042     .016424
         L4. |   .0015463   .0049663     0.31   0.756    -.0082232    .0113157
         L5. |   .0025978   .0049914     0.52   0.603     -.007221    .0124166
         L6. |    .006001   .0049798     1.21   0.229    -.0037949    .0157968
         L7. |   .0066017   .0049157     1.34   0.180     -.003068    .0162715
         L8. |  -.0015491   .0049371    -0.31   0.754     -.011261    .0081628
```

```
         L8. |  -.0015491   .0049371    -0.31   0.754     -.011261    .0081626
         L9. |   .0010036   .0048898     0.21   0.838    -.0086153    .0106225
        L10. |  -.0004773   .0047767    -0.10   0.920    -.0098737     .008919
        L11. |  -.0083937   .0046846    -1.79   0.074     -.017609    .0008216
        L12. |  -.0041455   .0044702    -0.93   0.354    -.0129391     .004648
             |
       month |
          2  |   .0077995   .0048077     1.62   0.106     -.001658     .017257
          3  |   .0052085   .0041637     1.25   0.212    -.0029821    .0133991
          4  |  -.0010198   .0053356    -0.19   0.849    -.0115156     .009476
          5  |  -.0012298   .0047478    -0.26   0.796    -.0105694    .0081098
          6  |  -.0122415   .0055844    -2.19   0.029    -.0232267   -.0012563
          7  |  -.0240128   .0047031    -5.11   0.000    -.0332644   -.0147612
          8  |  -.0152756   .0052483    -2.91   0.004    -.0255997   -.0049514
          9  |  -.0111308   .0045365    -2.45   0.015    -.0200548   -.0022068
         10  |  -.0046899    .006722    -0.70   0.486    -.0179129    .0085332
         11  |   .0076979   .0057763     1.33   0.184    -.0036649    .0190607
         12  |   .0151789   .0059337     2.56   0.011     .0035065    .0268514
             |
        date |   .0003695    .000047     7.86   0.000      .000277    .0004619
       _cons |  -11.28083    .391293   -28.83   0.000    -12.05055    -10.5111
------------------------------------------------------------------------------

.
. *4e Estimate two alternative models that contain month indicators and a time trend but
>  that impose a more parsimonious lag structure for the predictor variables. Explain yo
> ur choices.
. regress ln_fl_nonfarm l(0,4,8,12).ln_fl_lf l(0,4,8,12).ln_us_epr l(0,4,8,12).ln_fl_bp
> i.month date

      Source |       SS           df       MS      Number of obs   =       384
-------------+----------------------------------   F(24, 359)      =   3636.67
       Model |  9.46703767        24  .394459903   Prob > F        =    0.0000
    Residual |  .038939751       359  .000108467   R-squared       =    0.9959
-------------+----------------------------------   Adj R-squared   =    0.9956
       Total |  9.50597742       383  .024819784   Root MSE        =    .01041

------------------------------------------------------------------------------
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_lf |
         --. |   .2198644    .118892     1.85   0.065    -.0139479    .4536767
         L4. |   .3640379   .1628088     2.24   0.026     .0438591    .6842168
         L8. |   .6241057   .1697337     3.68   0.000     .2903084     .957903
        L12. |  -.3365352   .1300465    -2.59   0.010     -.592284   -.0807864
             |
   ln_us_epr |
         --. |   .8706823   .0862833    10.09   0.000     .7009981    1.040367
         L4. |   .0186581   .1180743     0.16   0.875     -.213546    .2508623
         L8. |  -.1364675   .1363531    -1.00   0.318    -.4046187    .1316838
        L12. |   .4492816   .1055542     4.26   0.000     .2416993    .6568639
             |
    ln_fl_bp |
         --. |   .0288326   .0033225     8.68   0.000     .0222986    .0353666
         L4. |    .014784   .0040692     3.63   0.000     .0067816    .0227864
         L8. |   .0053046   .0040599     1.31   0.192    -.0026795    .0132888
        L12. |  -.0040886   .0034865    -1.17   0.242    -.0109452     .002768
             |
```

```
       month |
          2  |    .003724   .0027268     1.37   0.173    -.0016384    .0090864
          3  |    .003428   .0030747     1.11   0.266    -.0026188    .0094747
          4  |  -.0013812   .0030302    -0.46   0.649    -.0073404    .0045779
          5  |  -.0050709    .003101    -1.64   0.103    -.0111693    .0010275
          6  |  -.0215379   .0030889    -6.97   0.000    -.0276125   -.0154633
          7  |  -.0356678   .0033321   -10.70   0.000    -.0422208   -.0291149
          8  |  -.0202856   .0032905    -6.16   0.000    -.0267567   -.0138145
          9  |  -.0118143   .0031977    -3.69   0.000    -.0181028   -.0055257
         10  |  -.0142884   .0031129    -4.59   0.000    -.0204102   -.0081666
         11  |  -.0033333   .0030634    -1.09   0.277    -.0093578    .0026912
         12  |   .0070509   .0028963     2.43   0.015      .001355    .0127468
             |
        date |   .0004262   .0000476     8.96   0.000     .0003326    .0005197
       _cons |  -10.60852   .3857432   -27.50   0.000    -11.36712   -9.849916
------------------------------------------------------------------------------
```

```
. gen dateQ = qofd(datec)

. format dateQ %tq

. regress ln_fl_nonfarm l(0/4).ln_fl_lf l(0/4).ln_us_epr l(0/4).ln_fl_bp i.month dateQ
```

```
      Source |       SS           df       MS      Number of obs   =       392
-------------+----------------------------------   F(27, 364)      =   2505.01
       Model |  10.2740552        27  .380520563   Prob > F        =    0.0000
    Residual |  .055292923       364  .000151904   R-squared       =    0.9946
-------------+----------------------------------   Adj R-squared   =    0.9942
       Total |  10.3293481       391  .02641777    Root MSE        =    .01232

------------------------------------------------------------------------------
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_lf |
         --. |   .2790757   .2536131     1.10   0.272    -.2196552    .7778065
         L1. |   .2956093   .3348151     0.88   0.378    -.3628055    .9540241
         L2. |  -.2641756   .3153608    -0.84   0.403    -.8843334    .3559822
         L3. |   .2832334   .3167687     0.89   0.372     -.339693    .9061598
         L4. |   .3220421   .2469667     1.30   0.193    -.1636186    .8077029
             |
   ln_us_epr |
         --. |    .869919   .1763402     4.93   0.000     .5231456    1.216693
         L1. |  -.1508318   .2303364    -0.65   0.513    -.6037889    .3021253
         L2. |   .1899043   .2170821     0.87   0.382    -.2369882    .6167968
         L3. |  -.2262386   .2208671    -1.02   0.306    -.6605744    .2080971
         L4. |   .3389032   .1751932     1.93   0.054    -.0056147    .6834212
             |
    ln_fl_bp |
         --. |   .0204443   .0051347     3.98   0.000      .010347    .0305417
         L1. |   .0107528   .0054657     1.97   0.050     4.39e-06    .0215012
         L2. |   .0026867   .0054899     0.49   0.625    -.0081091    .0134826
         L3. |   .0070439   .0054993     1.28   0.201    -.0037706    .0178583
         L4. |   .0071123   .0051692     1.38   0.170     -.003053    .0172777
             |
       month |
          2  |   .0052225   .0038186     1.37   0.172    -.0022868    .0127318
          3  |   .0086375   .0041006     2.11   0.036     .0005735    .0167014
```

```
           4  |   .0012736   .0046541     0.27   0.785    -.0078787    .0104258
           5  |   .0022027   .0038771     0.57   0.570    -.0054216    .0098269
           6  |  -.0193223   .0040672    -4.75   0.000    -.0273206   -.0113241
           7  |  -.0362039   .0038883    -9.31   0.000    -.0438502   -.0285575
           8  |  -.0245188   .0043528    -5.63   0.000    -.0330787    -.015959
           9  |  -.0171602   .0037189    -4.61   0.000    -.0244733   -.0098471
          10  |  -.0193132   .0044175    -4.37   0.000    -.0280001   -.0106262
          11  |   -.004866   .0041178    -1.18   0.238    -.0129637    .0032317
          12  |   .0058531   .0039007     1.50   0.134    -.0018177    .0135238
              |
       dateQ  |   .0010375   .0001584     6.55   0.000     .0007259     .001349
       _cons  |  -10.55687   .4171884   -25.30   0.000    -11.37727   -9.736468
------------------------------------------------------------------------------

.
.
. log close
      name:  <unnamed>
       log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/P
> roblem Set 1/Problem Set 1.smcl
  log type:  smcl
 closed on:  11 Feb 2021, 19:36:37
----------------------------------------------------------------------------------------
```

Problem Set 2

Gus Lipkin

CAP 4763 Time Series Modelling and Forecasting

> All underlined portions are the corrections

> All uncited quotes are from the problem set

Table of Contents

Part A

1. Writethemodel $y_t = \alpha + \delta t + \rho y_{t-1} + \beta x_{t-1} + r$ in first differences.

- $\Delta y_t = \delta + \rho \Delta y_{t-1} + \beta \Delta x_{t-1} + \Delta r_t$

2. Suppose after first differencing a model is $\Delta y_t = \delta - \varphi - 2\varphi t + \rho \Delta y_{t-1} + \beta \Delta x_{t-1} + \Delta r_t$. What was it before the first difference was taken? (Hint: both t and t^2 are in it.)

- $y_t = \delta t + \varphi t^2 + \varphi t - \varphi + \rho y_{t-1} + \beta x_{t-1} + r_t$ <- WRONG
- $\Delta y_t = \delta - \phi + 2\phi t + \rho \Delta t_{t-1} + \beta \Delta x_{t-1} + \Delta r_t$ <- RIGHT

3. Suppose you are originally interested in the model $y_t = \alpha + \delta t + \rho y_{t-1} + \beta x_{t-1} + r_t$, where $r_t = \gamma r_{t-1} + \varepsilon_t$ and ε_t is an independent random disturbance. Write the dynamically complete model in first differences. Hint: first substitute to make the model dynamically complete, and then take the first difference.

- $y_t = \alpha + \delta t + \rho y_{t-1} + \beta x_{t-1} + \gamma r_{t-1} + \varepsilon_t$ <- WRONG
- $\Delta y_t = \delta + \rho \Delta y_{t-1} + \beta \Delta x_{t-1} + \gamma \Delta r_{t-1} + \Delta \varepsilon_t$ <- WRONG
- $\Delta y_t = \delta(1-\gamma) + (\rho + \gamma)\Delta y_{t-1} - \gamma\rho \Delta y_{t-2} + \beta \Delta x_{t-2} + \varepsilon_t - \varepsilon_{t-1}$<- RIGHT

Part B

3. Autocorrelation and Weak Dependence

1. Obtain the correlation of each variable with its one period lag.

Variable	Correlation with Lag
lnflnonfarm	.9981
lnfllf	.9994
lnusepr	.9821
lnflbp	.9477

(obs=875)	corr ln_us_epr l1.ln_us_epr
	L.
	ln_us~r *ln_us*~r
ln_us_epr	
-.	1.0000
L1.	0.9758 1.0000

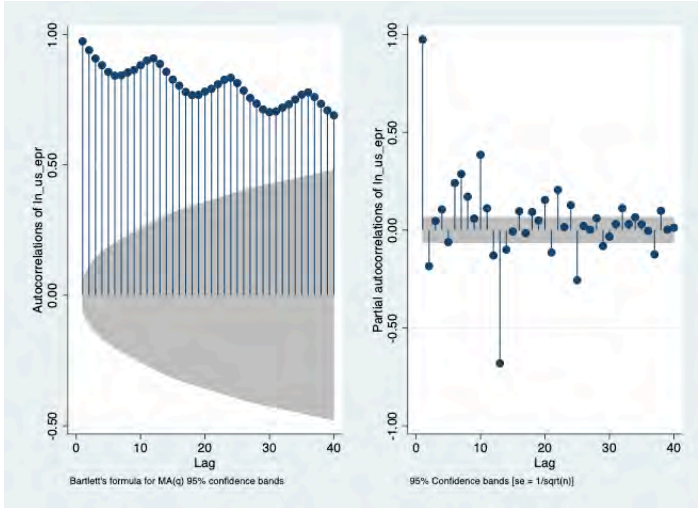
(obs=983)	**corr ln_fl_nonfarm l1.ln_fl_nonfarm**
	L.
	ln_fl~*m ln_fl*~m
ln_fl_nonf~m	
-.	1.0000
L1.	0.9999 1.0000

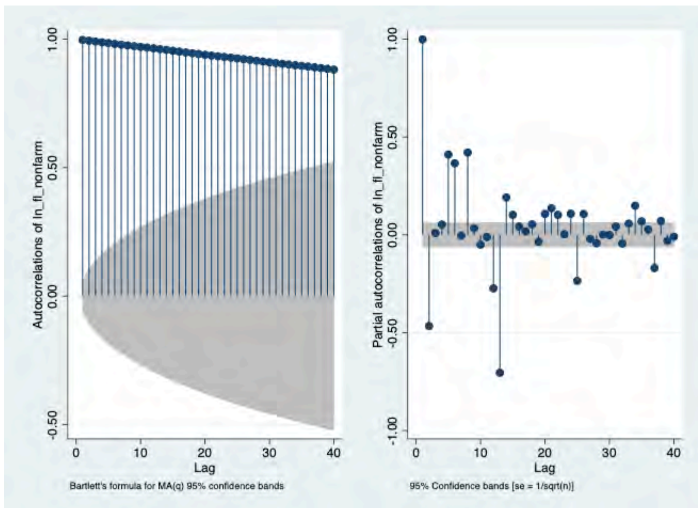
(obs=539)	**corr ln_fl_lf l1.ln_fl_lf**
	L.
	ln_fl_lf ln_fl_lf
ln_fl_lf	
-.	1.0000
L1.	0.9997 1.0000

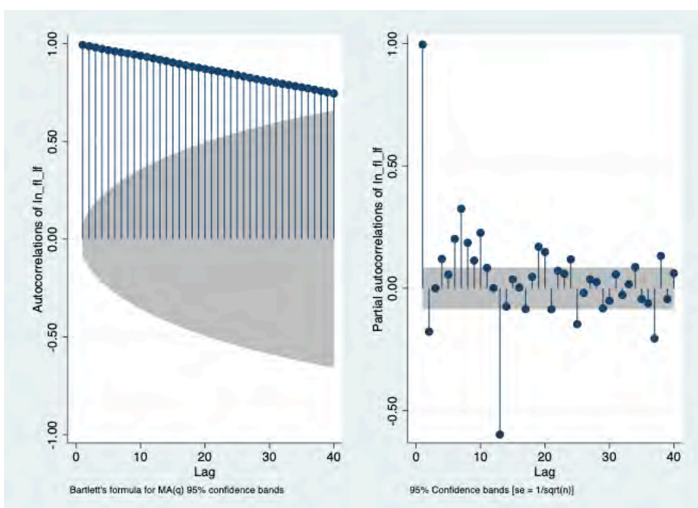
(obs=395)	**corr ln_fl_bp l1.ln_fl_bp**
	L.
	ln_fl_bp ln_fl_bp
ln_fl_bp	
-.	1.0000
L1.	0.9470 1.0000

- There appears to be very high correlation between the log form of each variable and its first lag. The highest is ln_fl_nonfarm with a correlation of .9999, followed by ln_fl_lf, ln_us_epr, and ln_fl_bp with .9997, .9758, and .9470 respectively.

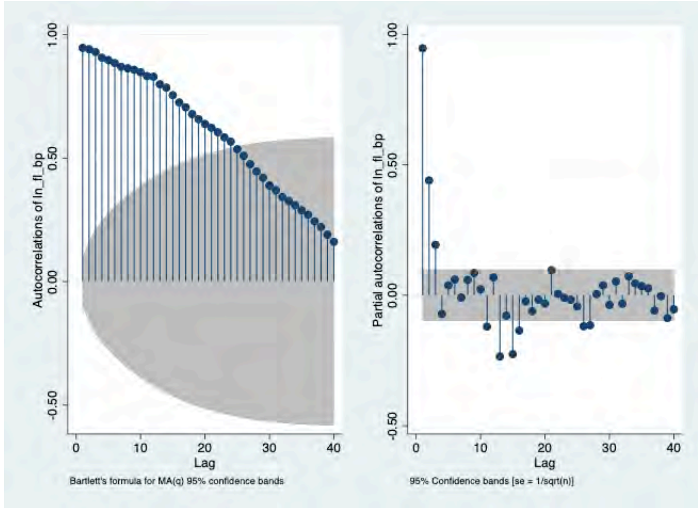
2. Obtain the autocorrelogram and partial autocorrelagram for each variable.







For the above three graphs, because all of the points are outside and above the cone, we can conclude that there is an autoregressive term in the data and should consult the partial autocorrelation graph. The PAC suggests that this is a higher order moving average.



For the last graph, the autocorrelation is not all outside of the confidence interval. When we look at the PAC we see that there are significant correlations in the first few terms followed by insignificant correlations in the rest. This suggests the order of the autoregressive term.

3. Conduct the Dickey-Fuller unit root rest for each variable.

Variable	**Dickey-Fuller p-value**
lnflnonfarm	.0328
lnfllf	.6285
lnusepr	.2246
lnflbp	.7774

```
. dfuller ln_us_epr, trend regress

Dickey-Fuller test for unit root                   Number of obs   =       875

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -4.020            -3.960            -3.410            -3.120
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0082

------------------------------------------------------------------------------
 D.ln_us_epr |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   ln_us_epr |
         L1. |  -.0392314   .0097585    -4.02   0.000    -.0583843   -.0200784
             |
      _trend |   4.02e-06   1.84e-06     2.18   0.030     3.99e-07    7.63e-06
       _cons |   .1583652   .0392952     4.03   0.000     .0812411    .2354894
------------------------------------------------------------------------------
```

```
. dfuller ln_fl_nonfarm, trend regress

Dickey-Fuller test for unit root                   Number of obs   =       983

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -0.653            -3.960            -3.410            -3.120
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.9761

------------------------------------------------------------------------------
D.           |
ln_fl_nonfarm|      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ln_fl_nonfarm|
          L1.|   -.001659   .0025399    -0.65   0.514    -.0066433    .0033253
             |
       _trend|   6.93e-07   8.56e-06     0.08   0.935    -.0000161    .0000175
        _cons|   .0159216   .0160282     0.99   0.321    -.0155318    .0473751
------------------------------------------------------------------------------

. dfuller ln_fl_lf, trend regress

Dickey-Fuller test for unit root                   Number of obs   =       539

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -1.724            -3.960            -3.410            -3.120
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.7400

------------------------------------------------------------------------------
   D.ln_fl_lf|      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     ln_fl_lf|
          L1.|  -.0076457   .0044337    -1.72   0.085    -.0163552    .0010639
             |
       _trend|   7.40e-06   8.61e-06     0.86   0.391    -9.52e-06    .0000243
        _cons|    .120517   .0676628     1.78   0.075    -.0123997    .2534337
------------------------------------------------------------------------------

. dfuller ln_fl_bp, trend regress

Dickey-Fuller test for unit root                   Number of obs   =       395

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -3.256            -3.984            -3.424            -3.130
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0738

------------------------------------------------------------------------------
   D.ln_fl_bp|      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     ln_fl_bp|
          L1.|  -.0545463   .0167509    -3.26   0.001    -.0874792   -.0216134
             |
       _trend|  -.0000375   .0000734    -0.51   0.609    -.0001817    .0001067
        _cons|   .5091679   .1583766     3.21   0.001     .1977942    .8205417
------------------------------------------------------------------------------
```

For both Dickey-Fuller of the ln_us_epr, the p-value is extremely low at .0082 and so we accept the null hypothesis. For all others, we fail to reject the null hypothesis. Especially ln_fl_nonfarm and ln_fl_lf.

4. "Looking at the AC and PAC, all four show strong enough first order autoregressive relationships to merit differencing. We can reject the null of an I(1) process for the log of non-farm employment. But, the partial autocorrelation coefficient is so close to one that we should difference anyway. The AC and PAC for the log difference of non-farm employment are bwlow, illustrating the differences are clearly not I(1)."

4. ARDL Model and Breusch-Godfrey Test

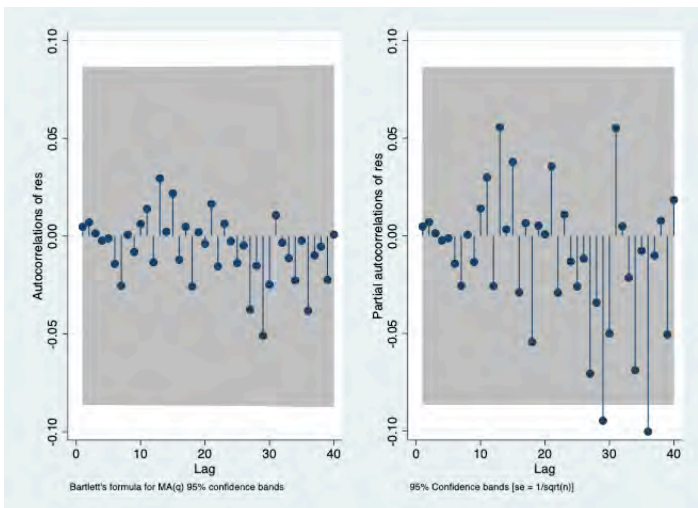
Given the results of the previous question, transform the data as needed and estimate a dynamically complete ARDL model for non-farm employment. Include at least one lag of the relevant dependent variable. How many additional lags of the dependent variable, and how many lags of which independent variables you include, are up to you. Looking back at what you did for Problem Set 1 might be informative, but don't be limited by it. Produce and interpret the AC and PAC for the residuals and the results of a Breusch-Godfrey test. In your write up, justify your specification and interpret the results.

.	regress d.ln_fl_nonfarm l(1/48)d.ln_fl_nonfarm	l(12/24)d.ln_us_epr	l(1/18, 24)d.ln_fl_lf	date
	Source SS df MS	Number of obs =	515	
	F(81, 433) =	15.48		
Model .050091055 81 .000618408	Prob > F =	0.0000		
Residual .01729335 433 .000039938	R-squared =	0.7434		
	Adj R-squared =	0.6954		
Total .067384405 514 .000131098	Root MSE =	.00632		
-------------	---------	---------	-----	-----
D.				
ln_fl_nonfarm	Coef.	Std. Err.	t	P>t
ln_fl_nonfarm				
LD.	-.1441103	.059346	-2.43	0.016
L2D.	-.1332106	.060728	-2.19	0.029
L3D.	.0520745	.060831	0.86	0.392
L4D.	.1139409	.0609067	1.87	0.062
L5D.	.066288	.0611891	1.08	0.279
L6D.	.1944856	.0614959	3.16	0.002
L7D.	.0759452	.0622902	1.22	0.223
L8D.	.0829208	.0631492	1.31	0.190
L9D.	.2532911	.0930772	2.72	0.007
L10D.	.1403499	.0960901	1.46	0.145
L11D.	.1893271	.0946093	2.00	0.046
L12D.	.4685154	.0957577	4.89	0.000

L13D.	.0758492	.1003991	0.76	0.450
L14D.	.0089228	.1008964	0.09	0.930
L15D.	.0490602	.1006788	0.49	0.626
L16D.	-.0187785	.1013922	-0.19	0.853
L17D.	.0547956	.1017669	0.54	0.591
L18D.	.0863921	.1011552	0.85	0.394
L19D.	-.25835	.1016689	-2.54	0.011
L20D.	-.1621826	.1009034	-1.61	0.109
L21D.	-.0839614	.1033319	-0.81	0.417
L22D.	-.1719582	.1017154	-1.69	0.092
L23D.	.0347504	.1011416	0.34	0.731
L24D.	.2927769	.0998811	2.93	0.004
L25D.	.1178616	.098203	1.20	0.231
L26D.	.0999885	.0980021	1.02	0.308
L27D.	-.1283723	.0980801	-1.31	0.191
L28D.	-.2031139	.0980964	-2.07	0.039
L29D.	-.2892074	.097907	-2.95	0.003
L30D.	-.5772115	.0991658	-5.82	0.000
L31D.	.6236058	.1020615	6.11	0.000
L32D.	.1870999	.1073141	1.74	0.082
L33D.	.1426809	.1091241	1.31	0.192
L34D.	.1068341	.1078243	0.99	0.322
L35D.	-.0794067	.1078368	-0.74	0.462
L36D.	.1327386	.1064489	1.25	0.213
L37D.	-.0639028	.099194	-0.64	0.520
L38D.	-.048562	.0984536	-0.49	0.622
L39D.	.0871388	.0975069	0.89	0.372
L40D.	-.1442082	.0974565	-1.48	0.140
L41D.	-.0032331	.0966638	-0.03	0.973
L42D.	.0938246	.0970599	0.97	0.334
L43D.	-.3559573	.0966539	-3.68	0.000
L44D.	-.0089124	.0978207	-0.09	0.927

L45D.	-.0882528	.0966085	-0.91	0.361
L46D.	.1086727	.091884	1.18	0.238
L47D.	.0313382	.091654	0.34	0.733
L48D.	.0609195	.091323	0.67	0.505
ln_us_epr				
L12D.	-.0155085	.1744885	-0.09	0.929
L13D.	-.3056076	.153451	-1.99	0.047
L14D.	-.5608006	.1545155	-3.63	0.000
L15D.	-.3645519	.1519838	-2.40	0.017
L16D.	.0029936	.1580302	0.02	0.985
L17D.	.0422232	.1559561	0.27	0.787
L18D.	.3199335	.1565006	2.04	0.042
L19D.	-.07463	.0988972	-0.75	0.451
L20D.	.0625226	.0999685	0.63	0.532
L21D.	-.0436852	.1002131	-0.44	0.663
L22D.	.2231831	.0985078	2.27	0.024
L23D.	-.0081188	.0960409	-0.08	0.933
L24D.	-.2688582	.1616447	-1.66	0.097
ln_fl_lf				
LD.	.1762398	.0704433	2.50	0.013
L2D.	-.1356975	.0715783	-1.90	0.059
L3D.	-.1659446	.0715828	-2.32	0.021
L4D.	-.0977864	.0709175	-1.38	0.169
L5D.	-.1364495	.0722069	-1.89	0.059
L6D.	-.2270642	.0723796	-3.14	0.002
L7D.	-.1332104	.0724525	-1.84	0.067
L8D.	-.2396185	.0727056	-3.30	0.001
L9D.	-.1256755	.079465	-1.58	0.114
L10D.	-.180737	.0797732	-2.27	0.024
L11D.	-.005726	.0808095	-0.07	0.944

L12D.	.0558537	.1334055	0.42	0.676
L13D.	.0173463	.1262683	0.14	0.891
L14D.	.2969825	.1275491	2.33	0.020
L15D.	.125207	.1266497	0.99	0.323
L16D.	-.0665773	.1288379	-0.52	0.606
L17D.	-.1292395	.1273895	-1.01	0.311
L18D.	-.2883037	.1278108	-2.26	0.025
L24D.	.2278015	.1255369	1.81	0.070
date	-8.65e-06	3.19e-06	-2.72	0.007
_cons	.0058495	.0022338	2.62	0.009



I don't think there's any correlation because almost everything is inside the interval.

. estat bgodfrey, lag(1/48)	
Breusch-Godfrey LM test for	autocorrelation

lags(p)	chi2	df	Prob > chi2
1	0.617	1	0.4321
2	1.630	2	0.4427

3	1.639	3	0.6506
4	1.665	4	0.7970
5	1.730	5	0.8850
6	2.757	6	0.8387
7	8.252	7	0.3109
8	8.536	8	0.3830
9	8.707	9	0.4648
10	8.803	10	0.5509
11	9.015	11	0.6205
12	10.913	12	0.5364
13	12.697	13	0.4715
14	12.775	14	0.5443
15	14.075	15	0.5198
16	15.212	16	0.5091
17	15.284	17	0.5751
18	18.315	18	0.4351
19	18.317	19	0.5014
20	19.893	20	0.4647
21	19.920	21	0.5263
22	20.203	22	0.5704
23	20.218	23	0.6287
24	20.362	24	0.6760
25	21.112	25	0.6864
26	21.381	26	0.7221
27	23.290	27	0.6693
28	24.359	28	0.6624
29	25.322	29	0.6615

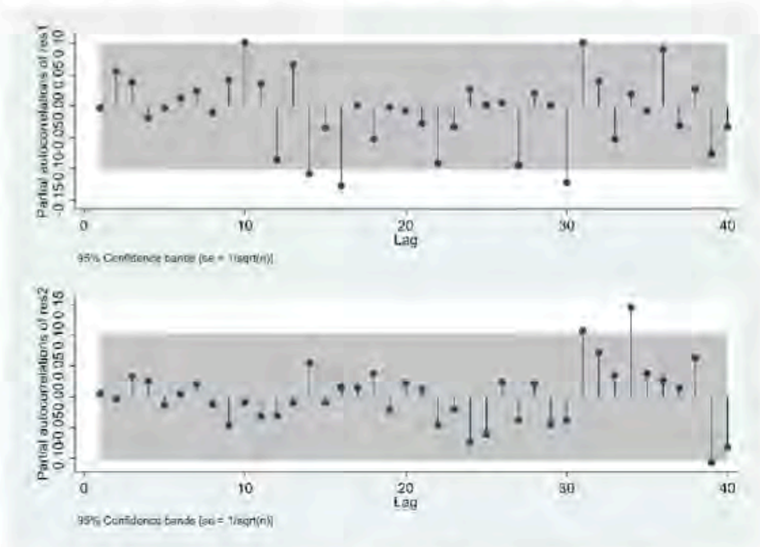
30	27.716	30	0.5855
31	28.706	31	0.5846
32	28.728	32	0.6330
33	29.272	33	0.6533
34	30.894	34	0.6207
35	30.897	35	0.6666
36	33.834	36	0.5720
37	35.071	37	0.5597
38	35.519	38	0.5847
39	38.229	39	0.5049
40	38.448	40	0.5402
41	38.548	41	0.5801
42	39.001	42	0.6034
43	39.107	43	0.6408
44	39.122	44	0.6804
45	39.431	45	0.7061
46	39.812	46	0.7278
47	40.011	47	0.7550
48	40.617	48	0.7664
H0: no serial correlation			

4) Given the results of the previous question, transform the data as needed and estimate a dynamically complete ARDL model for non-farm employment. Include at least one lag of the relevant dependent variable. How many additional lags of the dependent variable, and how many lags of which independent variables you include, are up to you. Looking back at what you did for Problem Set 1 might be informative, but don't be limited by it. Produce and interpret the AC and PAC for the residuals and the results of a Breusch-Godfrey test. In your write up, justify your specification and interpret the results.

I estimated two models, one with all lags back 12 months and one going back 24 months. Breush-Godfrey test results are in the table below. In the first case, the null of no serial correlation is rejected. For the second, the null can't be rejected at 24 lags, but it neither is it convincingly rejected (p=0.16). However, examining the PACs for the residuals in the figure below gives a bit more confidence in the second model. It also suggests some lags from year 3 and 4 may be worth including.

Breush-Godfrey tests for question 4

	$p >$ chi	
Lags	Model 1	Model 2
1	0.8812	0.4861
2	0.0332	0.778
3	0.0074	0.0585
4	0.0129	0.0386
5	0.0266	0.0709
6	0.0475	0.0774
7	0.0787	0.1049
8	0.0453	0.1426
9	0.068	0.1042
10	0.0467	0.1464
11	0.0688	0.1728
12	0.005	0.2121
13	0.0035	0.2321
14	0.0047	0.1206
15	0.0064	0.1304
16	0.0007	0.1483
17	0.0012	0.1816
18	0.0019	0.2039
19	0.0028	0.2119
20	0.0037	0.2138
21	0.0056	0.255
22	0.006	0.2065
23	0.0066	0.2485
24	0.0079	0.1618



A more parsimonious model, possibly with selected lags out further, might be a good idea. However, with some careful thought and exploration, I still have not come up with one that passed a Breusch-Godfrey test. Perhaps you did… Really, we will need more model selection tools to help us choose if we want to forecast. If we need to estimate parameters, we need to choose the appropriate model for the purpose, even if not dynamically complete, and then use appropriately adjusted standard errors. That is the point of the next problem.

5. Dynamically Complete Models and Newey-West Standard Errors

```
. reg d.ln_fl_nonfarm l(0/4)d.ln_fl_bp if tin(1948m1,2020m1)

      Source |       SS           df       MS      Number of obs   =       380
-------------+----------------------------------   F(5, 374)       =      2.97
       Model |  .00146591         5  .000293182   Prob > F        =    0.0122
    Residual |  .036972226       374  .000098856   R-squared       =    0.0381
-------------+----------------------------------   Adj R-squared   =    0.0253
       Total |  .038438136       379  .00010142    Root MSE        =    .00994

------------------------------------------------------------------------------
D.           |
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_bp |
         D1. |  -.0043445   .0035864    -1.21   0.227    -.0113965    .0027075
         LD. |  -.0115113   .0040594    -2.84   0.005    -.0194935   -.0035291
        L2D. |   .0019871   .0041056     0.48   0.629    -.0060858      .01006
        L3D. |  -.0011778   .0040768    -0.29   0.773    -.0091941    .0068385
        L4D. |  -.0028262   .0036121    -0.78   0.434    -.0099287    .0042763
             |
       _cons |   .0015358   .0005101     3.01   0.003     .0005328    .0025387
------------------------------------------------------------------------------

. newey d.ln_fl_nonfarm l(0/4)d.ln_fl_bp if tin(1948m1,2020m1), lag(4)

Regression with Newey-West standard errors      Number of obs     =        380
maximum lag: 4                                  F(  5,       374) =       4.01
                                                Prob > F          =     0.0015

------------------------------------------------------------------------------
D.           |             Newey-West
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_bp |
         D1. |  -.0043445    .003622    -1.20   0.231    -.0114665    .0027776
         LD. |  -.0115113   .0036606    -3.14   0.002    -.0187093   -.0043133
        L2D. |   .0019871   .0043475     0.46   0.648    -.0065616    .0105358
        L3D. |  -.0011778    .004813    -0.24   0.807    -.0106416     .008286
        L4D. |  -.0028262    .003664    -0.77   0.441    -.0100308    .0043783
             |
       _cons |   .0015358   .0004154     3.70   0.000     .0007189    .0023526
------------------------------------------------------------------------------
```

if fuller high, can't reject

5) Suppose you are interested in the relationship between the first difference in non-farm employment and the lags 0 to 4 of the differences of Florida building permits, controlling for seasonal impacts, but not controlling for any other variables or lags, including lags of employment. That is, you explicitly do not want to a dynamically complete model. (Don't worry about why, for this purpose.) Estimate the model both with and without Newey-West standard errors and discuss the difference that makes.

The results of interest are in the table at right. Note that the Newey-West standard errors are larger for the first three coefficients and smaller for the last. The regular standard errors are misleading regarding the precision of the estimates.

Models for question 5

Std Err	Regular	Newey-West
D.lnflbp	0.00820***	0.00820**
	(0.00203)	(0.00250)
LD.lnflbp	0.00793***	0.00793**
	(0.00236)	(0.00294)
L2D.lnflbp	0.00627*	0.00627
	(0.00244)	(0.00348)
L3D.lnflbp	0.00730**	0.00730*
	(0.00237)	(0.00300)
L4D.lnflbp	0.00430*	0.00430*
	(0.00204)	(0.00199)
N	379	379
R^2	0.764	

Standard errors in parentheses
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$
Constant, trend, and month coefficients omitted for space

Appendix A

```
clear
set more off

cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Problem Set 2"

*2a
*Done

*2b Load the data
import delimited "Assignment_1_Monthly.txt"

rename lnu02300000 us_epr
rename flnan fl_nonfarm
rename fllfn fl_lf
rename flbppriv fl_bp
rename date datestring
```

```
*2c Turn on a log file
log using "Problem Set 1", replace

*2d Generate a monthly date variable (make its display format monthly time, %tm)
gen datec=date(datestring, "YMD")
gen date=mofd(datec)
format date %tm

*2e tsset your data
tsset date

*2f
gen ln_us_epr=log(us_epr)
gen ln_fl_nonfarm=log(fl_nonfarm)
gen ln_fl_lf=log(fl_lf)
gen ln_fl_bp=log(fl_bp)

*3a
corr ln_us_epr l1.ln_us_epr
corr ln_fl_nonfarm l1.ln_fl_nonfarm
corr ln_fl_lf l1.ln_fl_lf
corr ln_fl_bp l1.ln_fl_bp

*3b
ac ln_us_epr, saving(ac_ln_us_epr.gph, replace)
pac ln_us_epr, saving(pac_ln_us_epr.gph, replace)
graph combine ac_ln_us_epr.gph pac_ln_us_epr.gph, saving(combo_ln_us_epr.gph,
replace)

ac ln_fl_nonfarm, saving(ac_ln_fl_nonfarm.gph, replace)
pac ln_fl_nonfarm, saving(pac_ln_fl_nonfarm.gph, replace)
graph combine ac_ln_fl_nonfarm.gph pac_ln_fl_nonfarm.gph,
saving(combo_ln_fl_nonfarm.gph, replace)

ac ln_fl_lf, saving(ac_ln_fl_lf.gph, replace)
pac ln_fl_lf, saving(pac_ln_fl_lf.gph, replace)
graph combine ac_ln_fl_lf.gph pac_ln_fl_lf.gph, saving(combo_ln_fl_lf.gph, replace)

ac ln_fl_bp, saving(ac_ln_fl_bp.gph, replace)
pac ln_fl_bp, saving(pac_ln_fl_bp.gph, replace)
graph combine ac_ln_fl_bp.gph pac_ln_fl_bp.gph, saving(combo_ln_fl_bp.gph, replace)

*3c
dfuller ln_us_epr, trend regress
dfuller ln_fl_nonfarm, trend regress
dfuller ln_fl_lf, trend regress
dfuller ln_fl_bp, trend regress

*4
```

```
regress d.ln_fl_nonfarm l(1/48)d.ln_fl_nonfarm l(12/24)d.ln_us_epr l(1/18,24)d.ln_fl_lf date
predict res, residual
ac res, saving(p4_ac.gph, replace)
pac res, saving(p4_pac.gph, replace)
graph combine p4_ac.gph p4_pac.gph, saving(p4_combo.gph, replace)
estat bgodfrey, lag(1/48)

*5
reg d.ln_fl_nonfarm l(0/4)d.ln_fl_bp if tin(1948m1,2020m1)
newey d.ln_fl_nonfarm l(0/4)d.ln_fl_bp if tin(1948m1,2020m1), lag(4)

log close
```

Appendix B

```
      name:  <unnamed>
       log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem S
> ets/Problem Set 2/Problem Set 1.smcl
  log type:  smcl
 opened on:  26 Feb 2021, 18:08:56

.
. *2d Generate a monthly date variable (make its display format monthly time, %tm)
. gen datec=date(datestring, "YMD")

. gen date=mofd(datec)

. format date %tm

.
. *2e tsset your data
. tsset date
        time variable:  date, 1939m1 to 2020m12
                delta:  1 month

.
. *2f
. gen ln_us_epr=log(us_epr)
(108 missing values generated)

. gen ln_fl_nonfarm=log(fl_nonfarm)

. gen ln_fl_lf=log(fl_lf)
(444 missing values generated)

. gen ln_fl_bp=log(fl_bp)
(588 missing values generated)

.
. *3a
. corr ln_us_epr l1.ln_us_epr
(obs=875)

             |                L.
             | ln_us_~r ln_us_~r
-------------+------------------
   ln_us_epr |
         --. |   1.0000
         L1. |   0.9758   1.0000


. corr ln_fl_nonfarm l1.ln_fl_nonfarm
(obs=983)

             |                L.
             | ln_fl_~m ln_fl_~m
-------------+------------------
ln_fl_nonf~m |
         --. |   1.0000
         L1. |   0.9999   1.0000


. corr ln_fl_lf l1.ln_fl_lf
(obs=539)

             |                L.
             | ln_fl_lf ln_fl_lf
-------------+------------------
    ln_fl_lf |
```

```
        --. |   1.0000
        L1. |   0.9997   1.0000


. corr ln_fl_bp l1.ln_fl_bp
(obs=395)

             |                 L.
             | ln_fl_bp ln_fl_bp
-------------+------------------
    ln_fl_bp |
        --. |   1.0000
        L1. |   0.9470   1.0000


.
. *3b
. ac ln_us_epr, saving(ac_ln_us_epr.gph, replace)
(file ac_ln_us_epr.gph saved)

. pac ln_us_epr, saving(pac_ln_us_epr.gph, replace)
(file pac_ln_us_epr.gph saved)

. graph combine ac_ln_us_epr.gph pac_ln_us_epr.gph, saving(combo_ln_us_epr.gph, rep
> lace)
(file combo_ln_us_epr.gph saved)

.
. ac ln_fl_nonfarm, saving(ac_ln_fl_nonfarm.gph, replace)
(file ac_ln_fl_nonfarm.gph saved)

. pac ln_fl_nonfarm, saving(pac_ln_fl_nonfarm.gph, replace)
(file pac_ln_fl_nonfarm.gph saved)

. graph combine ac_ln_fl_nonfarm.gph pac_ln_fl_nonfarm.gph, saving(combo_ln_fl_nonf
> arm.gph, replace)
(file combo_ln_fl_nonfarm.gph saved)

.
. ac ln_fl_lf, saving(ac_ln_fl_lf.gph, replace)
(file ac_ln_fl_lf.gph saved)

. pac ln_fl_lf, saving(pac_ln_fl_lf.gph, replace)
(file pac_ln_fl_lf.gph saved)

. graph combine ac_ln_fl_lf.gph pac_ln_fl_lf.gph, saving(combo_ln_fl_lf.gph, replac
> e)
(file combo_ln_fl_lf.gph saved)

.
. ac ln_fl_bp, saving(ac_ln_fl_bp.gph, replace)
(file ac_ln_fl_bp.gph saved)

. pac ln_fl_bp, saving(pac_ln_fl_bp.gph, replace)
(file pac_ln_fl_bp.gph saved)

. graph combine ac_ln_fl_bp.gph pac_ln_fl_bp.gph, saving(combo_ln_fl_bp.gph, replac
> e)
(file combo_ln_fl_bp.gph saved)

.
. *3c
. dfuller ln_us_epr, trend regress

Dickey-Fuller test for unit root                   Number of obs   =       875

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -4.020            -3.960            -3.410            -3.120
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.0082

------------------------------------------------------------------------------
D.ln_us_epr |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  ln_us_epr |
        L1. |  -.0392314   .0097585    -4.02   0.000    -.0583843   -.0200784
            |
     _trend |   4.02e-06   1.84e-06     2.18   0.030     3.99e-07    7.63e-06
      _cons |   .1583652   .0392952     4.03   0.000     .0812411    .2354894
------------------------------------------------------------------------------

. dfuller ln_fl_nonfarm, trend regress

Dickey-Fuller test for unit root                   Number of obs   =       983

                               ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(t)             -0.653            -3.960            -3.410            -3.120
------------------------------------------------------------------------------
MacKinnon approximate p-value for Z(t) = 0.9761
```

D. ln_fl_nonfarm	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
ln_fl_nonfarm						
L1.	-.001659	.0025399	-0.65	0.514	-.0066433	.0033253
_trend	6.93e-07	8.56e-06	0.08	0.935	-.0000161	.0000175
_cons	.0159216	.0160282	0.99	0.321	-.0155318	.0473751

```
. dfuller ln_fl_lf, trend regress
```

Dickey-Fuller test for unit root   Number of obs   =   539

	Test Statistic	Interpolated Dickey-Fuller 1% Critical Value	5% Critical Value	10% Critical Value
Z(t)	-1.724	-3.960	-3.410	-3.120

MacKinnon approximate p-value for Z(t) = **0.7400**

D.ln_fl_lf	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
ln_fl_lf						
L1.	-.0076457	.0044337	-1.72	0.085	-.0163552	.0010639
_trend	7.40e-06	8.61e-06	0.86	0.391	-9.52e-06	.0000243
_cons	.120517	.0676628	1.78	0.075	-.0123997	.2534337

```
. dfuller ln_fl_bp, trend regress
```

Dickey-Fuller test for unit root   Number of obs   =   395

	Test Statistic	Interpolated Dickey-Fuller 1% Critical Value	5% Critical Value	10% Critical Value
Z(t)	-3.256	-3.984	-3.424	-3.130

MacKinnon approximate p-value for Z(t) = **0.0738**

D.ln_fl_bp	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
ln_fl_bp						
L1.	-.0545463	.0167509	-3.26	0.001	-.0874792	-.0216134
_trend	-.0000375	.0000734	-0.51	0.609	-.0001817	.0001067
_cons	.5091679	.1583766	3.21	0.001	.1977942	.8205417

```
.
. *4
. regress d.ln_fl_nonfarm l(1/48)d.ln_fl_nonfarm l(12/24)d.ln_us_epr l(1/18, 24)d.l
> n_fl_lf date
```

Source	SS	df	MS
Model	.050091055	81	.000618408
Residual	.01729335	433	.000039938
Total	.067384405	514	.000131098

Number of obs	=	515
F(81, 433)	=	15.48
Prob > F	=	0.0000
R-squared	=	0.7434
Adj R-squared	=	0.6954
Root MSE	=	.00632

D. ln_fl_nonfarm	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
ln_fl_nonfarm						
LD.	-.1441103	.059346	-2.43	0.016	-.2607523	-.0274683
L2D.	-.1332106	.060728	-2.19	0.029	-.2525689	-.0138524
L3D.	.0520745	.060831	0.86	0.392	-.0674863	.1716354
L4D.	.1139409	.0609067	1.87	0.062	-.0057687	.2336504
L5D.	.066288	.0611891	1.08	0.279	-.0539766	.1865526
L6D.	.1944856	.0614959	3.16	0.002	.073618	.3153532
L7D.	.0759452	.0622902	1.22	0.223	-.0464836	.1983741
L8D.	.0829208	.0631492	1.31	0.190	-.0411963	.207038
L9D.	.2532911	.0930772	2.72	0.007	.0703519	.4362303
L10D.	.1403499	.0960901	1.46	0.145	-.0485112	.329211
L11D.	.1893271	.0946093	2.00	0.046	.0033766	.3752776
L12D.	.4685154	.0957577	4.89	0.000	.2803077	.6567232
L13D.	.0758492	.1003991	0.76	0.450	-.1214811	.2731795
L14D.	.0089228	.1008964	0.09	0.930	-.1893847	.2072303
L15D.	.0490602	.1006788	0.49	0.626	-.1488197	.24694
L16D.	-.0187785	.1013922	-0.19	0.853	-.2180605	.1805035
L17D.	.0547956	.1017669	0.54	0.591	-.1452228	.2548141
L18D.	.0863921	.1011552	0.85	0.394	-.1124241	.2852084
L19D.	-.25835	.1016689	-2.54	0.011	-.4581759	-.0585241
L20D.	-.1621826	.1009034	-1.61	0.109	-.360504	.0361389
L21D.	-.0839614	.1033319	-0.81	0.417	-.2870559	.1191331
L22D.	-.1719582	.1017154	-1.69	0.092	-.3718755	.0279592

L23D.	.0347504	.1011416	0.34	0.731	-.1640391	.2335399
L24D.	.2927769	.0998811	2.93	0.004	.0964647	.489089
L25D.	.1178616	.098203	1.20	0.231	-.0751523	.3108754
L26D.	.0999885	.0980021	1.02	0.308	-.0926304	.2926074
L27D.	-.1283723	.0980801	-1.31	0.191	-.3211445	.0644
L28D.	-.2031139	.0980964	-2.07	0.039	-.3959182	-.0103096
L29D.	-.2892074	.097907	-2.95	0.003	-.4816395	-.0967753
L30D.	-.5772115	.0991658	-5.82	0.000	-.7721176	-.3823054
L31D.	.6236058	.1020615	6.11	0.000	.4230083	.8242034
L32D.	.1870999	.1073141	1.74	0.082	-.0238215	.3980212
L33D.	.1426809	.1091241	1.31	0.192	-.0717978	.3571596
L34D.	.1068341	.1078243	0.99	0.322	-.1050899	.3187581
L35D.	-.0794067	.1078368	-0.74	0.462	-.2913554	.1325421
L36D.	.1327386	.1064489	1.25	0.213	-.0764823	.3419594
L37D.	-.0639028	.099194	-0.64	0.520	-.2588645	.1310589
L38D.	-.048562	.0984536	-0.49	0.622	-.2420684	.1449445
L39D.	.0871388	.0975069	0.89	0.372	-.104507	.2787845
L40D.	-.1442082	.0974565	-1.48	0.140	-.3357548	.0473384
L41D.	-.0032331	.0966638	-0.03	0.973	-.1932218	.1867555
L42D.	.0938246	.0970599	0.97	0.334	-.0969425	.2845917
L43D.	-.3559573	.0966539	-3.68	0.000	-.5459264	-.1659882
L44D.	-.0089124	.0978207	-0.09	0.927	-.2011749	.1833501
L45D.	-.0882528	.0966085	-0.91	0.361	-.2781327	.1016272
L46D.	.1086727	.091884	1.18	0.238	-.0719214	.2892668
L47D.	.0313382	.091654	0.34	0.733	-.1488038	.2114803
L48D.	.0609195	.091323	0.67	0.505	-.118572	.240411
ln_us_epr						
L12D.	-.0155085	.1744885	-0.09	0.929	-.3584584	.3274413
L13D.	-.3056076	.153451	-1.99	0.047	-.607209	-.0040062
L14D.	-.5608006	.1545155	-3.63	0.000	-.8644942	-.257107
L15D.	-.3645519	.1519838	-2.40	0.017	-.6632696	-.0658341
L16D.	.0029936	.1580302	0.02	0.985	-.3076081	.3135954
L17D.	.0422232	.1559561	0.27	0.787	-.264302	.3487483
L18D.	.3199335	.1565006	2.04	0.042	.0123381	.6275288
L19D.	-.07463	.0988972	-0.75	0.451	-.2690083	.1197484
L20D.	.0625226	.0999685	0.63	0.532	-.1339614	.2590065
L21D.	-.0436852	.1002131	-0.44	0.663	-.2406498	.1532795
L22D.	.2231831	.0985078	2.27	0.024	.0295703	.416796
L23D.	-.0081188	.0960409	-0.08	0.933	-.1968832	.1806456
L24D.	-.2688582	.1616447	-1.66	0.097	-.5865639	.0488476
ln_fl_lf						
LD.	.1762398	.0704433	2.50	0.013	.0377865	.3146932
L2D.	-.1356975	.0715783	-1.90	0.059	-.2763815	.0049866
L3D.	-.1659446	.0715828	-2.32	0.021	-.3066375	-.0252517
L4D.	-.0977864	.0709175	-1.38	0.169	-.2371718	.041599
L5D.	-.1364495	.0722069	-1.89	0.059	-.278369	.00547
L6D.	-.2270642	.0723796	-3.14	0.002	-.3693234	-.0848051
L7D.	-.1332104	.0724525	-1.84	0.067	-.2756127	.0091919
L8D.	-.2396185	.0727056	-3.30	0.001	-.3825182	-.0967187
L9D.	-.1256755	.079465	-1.58	0.114	-.2818605	.0305095
L10D.	-.180737	.0797732	-2.27	0.024	-.3375278	-.0239463
L11D.	-.005726	.0808095	-0.07	0.944	-.1645537	.1531017
L12D.	.0558537	.1334055	0.42	0.676	-.2063492	.3180565
L13D.	.0173463	.1262683	0.14	0.891	-.2308286	.2655213
L14D.	.2969825	.1275491	2.33	0.020	.0462901	.547675
L15D.	.125207	.1266497	0.99	0.323	-.1237177	.3741316
L16D.	-.0665773	.1288379	-0.52	0.606	-.3198027	.186648
L17D.	-.1292395	.1273895	-1.01	0.311	-.3796182	.1211391
L18D.	-.2883037	.1278108	-2.26	0.025	-.5395104	-.037097
L24D.	.2278015	.1255369	1.81	0.070	-.018936	.4745389
date	-8.65e-06	3.19e-06	-2.72	0.007	-.0000149	-2.39e-06
_cons	.0058495	.0022338	2.62	0.009	.0014592	.0102399

```
. predict res, residual
(469 missing values generated)

. ac res, saving(p4_ac.gph, replace)
(file p4_ac.gph saved)

. pac res, saving(p4_pac.gph, replace)
(file p4_pac.gph saved)

. graph combine p4_ac.gph p4_pac.gph, saving(p4_combo.gph, replace)
(file p4_combo.gph saved)

. estat bgodfrey, lag(1/48)
```

Breusch-Godfrey LM test for autocorrelation

lags(*p*)	chi2	df	Prob > chi2
1	0.617	1	0.4321
2	1.630	2	0.4427
3	1.639	3	0.6506
4	1.665	4	0.7970
5	1.730	5	0.8850
6	2.757	6	0.8387
7	8.252	7	0.3109
8	8.536	8	0.3830
9	8.707	9	0.4648
10	8.803	10	0.5509

```
      11 |        9.015          11              0.6205
      12 |       10.913          12              0.5364
      13 |       12.697          13              0.4715
      14 |       12.775          14              0.5443
      15 |       14.075          15              0.5198
      16 |       15.212          16              0.5091
      17 |       15.284          17              0.5751
      18 |       18.315          18              0.4351
      19 |       18.317          19              0.5014
      20 |       19.893          20              0.4647
      21 |       19.920          21              0.5263
      22 |       20.203          22              0.5704
      23 |       20.218          23              0.6287
      24 |       20.362          24              0.6760
      25 |       21.112          25              0.6864
      26 |       21.381          26              0.7221
      27 |       23.290          27              0.6693
      28 |       24.359          28              0.6624
      29 |       25.322          29              0.6615
      30 |       27.716          30              0.5855
      31 |       28.706          31              0.5846
      32 |       28.728          32              0.6330
      33 |       29.272          33              0.6533
      34 |       30.894          34              0.6207
      35 |       30.897          35              0.6666
      36 |       33.834          36              0.5720
      37 |       35.071          37              0.5597
      38 |       35.519          38              0.5847
      39 |       38.229          39              0.5049
      40 |       38.448          40              0.5402
      41 |       38.548          41              0.5801
      42 |       39.001          42              0.6034
      43 |       39.107          43              0.6408
      44 |       39.122          44              0.6804
      45 |       39.431          45              0.7061
      46 |       39.812          46              0.7278
      47 |       40.011          47              0.7550
      48 |       40.617          48              0.7664
-----------------------------------------------------------------
                    H0: no serial correlation

.
. *5
. reg d.ln_fl_nonfarm l(0/4)d.ln_fl_bp if tin(1948m1,2020m1)

      Source |       SS           df       MS      Number of obs   =       380
-------------+----------------------------------   F(5, 374)       =      2.97
       Model |  .00146591         5  .000293182   Prob > F        =    0.0122
    Residual |  .036972226       374  .000098856   R-squared       =    0.0381
-------------+----------------------------------   Adj R-squared   =    0.0253
       Total |  .038438136       379   .00010142   Root MSE        =    .00994

------------------------------------------------------------------------------
D.           |
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_bp |
         D1. |  -.0043445   .0035864    -1.21   0.227    -.0113965    .0027075
         LD. |  -.0115113   .0040594    -2.84   0.005    -.0194935   -.0035291
        L2D. |   .0019871   .0041056     0.48   0.629    -.0060858      .01006
        L3D. |  -.0011778   .0040768    -0.29   0.773    -.0091941    .0068385
        L4D. |  -.0028262   .0036121    -0.78   0.434    -.0099287    .0042763
             |
       _cons |   .0015358   .0005101     3.01   0.003     .0005328    .0025387
------------------------------------------------------------------------------

. newey d.ln_fl_nonfarm l(0/4)d.ln_fl_bp if tin(1948m1,2020m1), lag(4)

Regression with Newey-West standard errors      Number of obs     =        380
maximum lag: 4                                  F(  5,        374) =      4.01
                                                Prob > F          =     0.0015

------------------------------------------------------------------------------
D.           |             Newey-West
ln_fl_nonf~m |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    ln_fl_bp |
         D1. |  -.0043445    .003622    -1.20   0.231    -.0114665    .0027776
         LD. |  -.0115113   .0036606    -3.14   0.002    -.0187093   -.0043133
        L2D. |   .0019871   .0043475     0.46   0.648    -.0065616    .0105358
        L3D. |  -.0011778    .004813    -0.24   0.807    -.0106416     .008286
        L4D. |  -.0028262    .003664    -0.77   0.441    -.0100308    .0043783
             |
       _cons |   .0015358   .0004154     3.70   0.000     .0007189    .0023526
------------------------------------------------------------------------------

.
. log close
      name:  <unnamed>
       log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem S
> ets/Problem Set 2/Problem Set 1.smcl
  log type:  smcl
 closed on:  26 Feb 2021, 18:09:11
----------------------------------------------------------------------------------
```

Gus Lipkin - CAP 4763 Time Series Midterm

> **Corrections are marked by underlines**

> 1. Express the model above in first differences. Under what conditions would you need to work with the differenced model instead of the original?

- $List_t = \beta_0 + \beta_P lnPermits_{t-1} + \beta_{Int} lnInterest_{t-1} + \beta_{Inf} lnInflation_{t-1} + r_t$
- $\Delta List_t = \beta_0 + \beta_P \Delta lnPermits_{t-1} + \beta_{Int} \Delta lnInterest_{t-1} + \beta_{Inf} \Delta lnInflation_{t-1} + \Delta r_t$
- $\underline{\Delta List_t = \beta_0 + \beta_P \Delta lnPermits_{t-1} + \beta_{Int} \Delta lnInterest_{t-1} + \beta_{Inf} \Delta lnInflation_{t-1} + \Delta r_t}$

> 2. Suppose you think the residual, r_t, follows an AR(1) process with parameter ρ. Write the dynamically complete version of the original model ***and*** of the model in first differences.

- $List_t = \beta_0 + \beta_P lnPermits_{t-1} + \beta_{Int} lnInterest_{t-1} + \beta_{Inf} lnInflation_{t-1} + r_t$
- $\Delta List_t = \underline{\beta_0 +} \beta_P lnPermits_{t-1} + \beta_{Int} lnInterest_{t-1} + \beta_{Inf} lnInflation_{t-1} +$
  $\rho(List_{t-1} - \beta_0 - \beta_P lnPermits_{t-1} - \beta_{Int} lnInterest_{t-1} - \beta_{Inf} lnInflation_{t-1}) + \varepsilon_t$
- $\underline{\Delta List_t = \beta_0 + \beta_P lnPermits_{t-1} + \beta_{Int} lnInterest_{t-1} + \beta_{Inf} lnInflation_{t-1} +}$
  $\underline{\rho(List_{t-1} - \beta_0 - \beta_P lnPermits_{t-1} - \beta_{Int} lnInterest_{t-1} - \beta_{Inf} lnInflation_{t-1}) + \varepsilon_t}$
- That second equation can then be distributed out and further reduced.

> 3. What is the purpose of the commands on page 1 lines 25-32?

- In order the functions do as follows, renaming the *date* variable to *datestring*, generating a new variable called *datec* that is the *datestring* variable in YMD format, generating another variable, *date* that is *datec* in monthly data form, then formatting the *date* variable as a timeseries data-type, then setting the beginning of the time-series data, and finally generating a new *month* variable that is the month of the date.
- We do all of this to make sure that our data is in a format that STATA can work with and so that we have all of the variables we will need later ready to go at the beginning of the analysis.
- "These lines make sure Stata properly recognizes time in the data. A monthly date is set for time related calculations such as lags and differences and a categorical variable is created for month of year (1-12) for capturing seasonal effects."

> 4. What is the purpose of the commands and results from page 2 line 4 through page 3 line 30, and what conclusion should be drawn from the results of these commands?

- The *ac* command generates an autocorrelogram graph while the *pac* chart generates a partial autocorrelogram. *dfuller* runs a Dickey-Fuller test on the data. The Dickey-Fuller test has an option for *lag(12)* which lets us lag for 12 months (an entire year) of data.
- AC and PAC graphs can be used in conjunction to identify ARIMA models. If a point is significant, it extends beyond the shaded boundary. For the non-differenced models, we see that significance decreases as time passes. We can then look to the PAC chart and see that there is significant correlation in the first lag and correlations that are not significant. This suggests a higher order autorgressive term. In the differenced models, the AC graph spikes are right at the edges of the range. If they are significant, it suggests an autoregressive term, if they are not significant, it suggests a moving average term. The same goes for the PAC but instead the insignificant values suggest autoregressive while the significant

suggest moving average.

- The Dickey-Fuller test is interpreted by its p-value. I don't remember what it does and everything I have says that it tests to see if there is a unit root but I have no clue what a unit root is and can't figure it out. I found this article https://stats.stackexchange.com/questions/29121/intuitive-explanation-of-unit-root which has a very funny joke at the bottom of the accepted answer. If I'm understanding what A.A. Milne 2.0 is saying, the model does not have a unit root because the p-value is less than one, the data will converge back to the same spot.
- "The purpose is to determine whether lnList is exhibits high persistence or only weak dependence, since further analysis required the time series be stationary and weakly dependent. The AC and PAC are consistent with an I(1) process and indicate very high persistence, and the Dickey Fuller test cannot reject the null hypothesis of an I(1) process. The conclusion is that this series demonstrates high persistence and should be differenced before further analysis."

> 5. Four sets of models are estimates. What are the differences between the sets (**not** between the models in a given set)? Which set is better for the purpose at hand? Why?

- Model 1 is only lagged, model 2 is lagged with the date and month indicators, model three is lagged and differenced, model 4 is lagged and differenced with the date and month indicators.
- It's possible that these are AR/DL models in all four forms. None, autoregressive, distributed lag, and autoregressive distributed lag
- I'm going to choose Model Set 3. There's something bothering me suggesting maybe I should choose 4 instead but I'm going to stick with 3. I'm not super confident with why, but I feel okay with it and I'm running out of time. It's the only one where there's ever any enough evidence to reject the null hypothesis of the Breusch-Godfrey test.
- "The differences are a) whether month dummies and a time trend are included and b) whether the data is differenced before estimating the model. The PAC and AC and Dickey Fuller test discussed previously indicate differencing is necessary. Seasonal indicator variables should be used to deal with purely seasonal effects (e.g. weather impacts on home building) unless there is a clear reason not to. Hence, set 4, which does both, is the best of these."

> 6. There are three models within the set you chose. Each of those is estimated twice. What is the difference between the two sets of estimates? Does the difference matter? Why? Which is better? Why?

- The first set of models is estimated only using the first lag. The second set of models uses the first and second lags.
- The difference does matter because it changes how far back the model looks when making its predictions.
- I think the first option is better because it is taking all three variables into account rather than just two. Like I say below, it is best to include all hypothesis variables when testing.
- "For each model in the set, the first estimate uses the command **regress** which calculated default standard errors that assume no serial correlation and no heteroskedasticity in the residuals, while the second uses the command **newey** which calculates standard errors robust to autocorrelation and heteroskedasticity in the residuals. Thus, unless you are very confident your model has no autocorrelation or heteroskedasticity in the residuals, use the second estimate with Newey-West standard errors. Note the Breusch-Godfrey tests suggest autocorrelation remains."

> 7. For the set you chose as best, interpret the F-test for the first model in that set. That is, if set X is

> best, interpret the F-test that follows one of the two estimates of Model X.1. Again, there are two versions. Use the better one. Your answer to 6 should have made it clear what the difference is, which is better, and why.

- (I'm not entirely sure if *testparm* or *estat bgodfrey* is the F-test and I don't have enough time to figure it out. I'm going to assume it's *testparm*. I also don't know what *test* is either...)
Although the p-value is very close to .05, it is still just above it at .0566. This suggests that we our model does not fit the data as well as it could.
- "This is a test (using the **testparm** command for testing sets of parameters) of the null hypothesis that neither the first lag of inflation nor the first lag of the interest rate have predictive power. The second version is best because it uses calculations robust to autocorrelation and heteroskedasticity. The p-value of 0.3545 indicates there is very little evidence upon which to conclude these variables are predictive of list prices."

> 8. How do the three models in your chosen set relate to the model set out on the previous page and to questions 1 and 2?

- One model is like the original model given, one of them is like the first difference model from problem 1, and one of them is like the autoregressive dynamically complete model from question 2.
- "All are derived from the model set out in the background section. Model 4.1 is in first differences, like in question 1. Model 4.2 is the second equation in question 2, which would be dynamically complete if the residuals of the baseline model were a simple AR(1) process. The third model simply deleted inflation and interest as predictors from Model 4.2 following the test that shows no evidence they contribute predictive power. (If you incorrectly chose a non-differenced set, this would differ slightly)"

> 9. What assumption must be defended to apply a causal interpretation to the results of this model, as opposed to a purely predictive one?

- You must assume that all relevant variables are accounted for and that there are no *omitted variables*. You must also assume that there is no *multicollinearity* in the data. The first means you shouldn't leave out important data and the second means you shouldn't include two pieces of data that are correlated with each other such as the amount of cereal consumed and the amount of milk consumed. Most people consume those items together, so using them both can be redundant and detrimental to the model.
- "No omitted causes of list price are contemporaneously correlated with permits, interest, or inflation."

> 10. Within the set of models you chose as best, X, which model is best for predicting *List*? That is, X.1, X.2, or X.3? Why? Which is best for testing they hypotheses of interest? Why? If the two are different, why?

- I think model 3.3 is the best predictor but 3.2 also looks pretty good. The best for testing the hypothesis is 3.2 because it is the only one that takes the number of building permits, the interest rate, and the inflation rate into account. They could be different because when testing a hypothesis, it is important to test all of the variables discussed in your hypothesis. If I say "high consumption of pizza and beers leads to heart disease", I can't only test if pizza leads to heart disease, I have to test both. That said, it might turn out that pizza is a much better indicator than beers and that beers doesn't add much. In that case, the pizza only model would be better because there is less room for error.

- "For prediction, one could make an argument for either model 4.2 or model 4.3. Model 4.3 is more parsimonious, dropping variables that seem not to have predictive power. But if the content knowledge indicating they should be controlled for is strong, Model 4.2 is better and 4.3 is simply overfit to the data. Model 4.1 is not as useful because it lacks lagged variables that we see have predictive power.

  For causation, the three null hypotheses are that the three coefficients in the original model are zero. The alternative hypotheses are that the coefficient on permits is negative, the coefficient on interest rates is negative, and that the coefficient on inflation is positive. Model 4.3 does not contain all three coefficients, so it simply cannot test these hypotheses. As long as we use the Newey-West standard errors, and can defend the assumption discussed in question 9, we can make an argument for either model 4.1 or model 4.2. The coefficients of the first model are a bit more precisely estimated because the second lags of the three predictors in Model 4.2 don't appear to add anything useful, so I would probably take Model 4.1, which is the direct application of the hypothesized model after differencing. But, if you chose 4.2, that is not a bad answer."

13.2 Model Selection 2

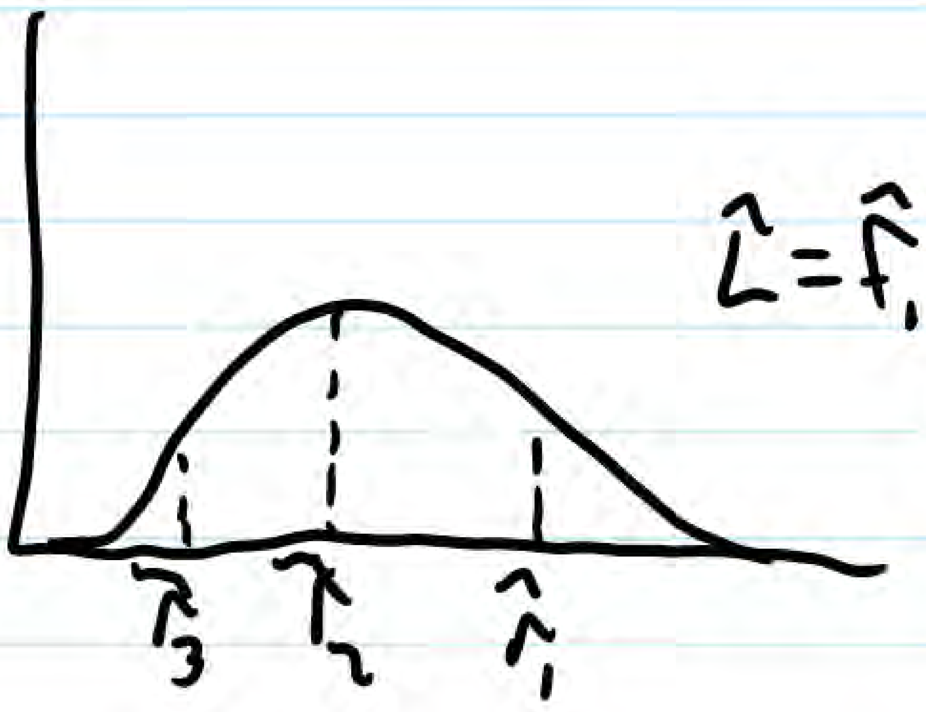
Wednesday, March 3, 2021 7:27 PM

1) Domain knowledge
+ joint hypothesis tests
2) measures that guard against overfitting

↳ $AIC = 2k - 2\ln\hat{L}$ → good

↳ Bad ↳ $\hat{L}$ = likelihood function

↳ smaller is better



$\hat{L} = \hat{f}_1 \hat{f}_2 \hat{f}_3 \rightarrow \ln\hat{L} = \ln\hat{f}_1 + \ln\hat{f}_2 + \ln\hat{f}_3$

↳ $\sum_{t=1}^{T} \ln(\hat{f}_t \mid \text{model})$

↳ $BIC = \ln(T)k - 2\ln\hat{L}$

↳ smaller is better

↳ finite set of models

↳ tend to pick the right one at the limit

AIC set of models grows w/ sample

3) Empirical Evidence on out of sample prediction

13.3 Model Selection 3

Saturday, March 6, 2021 6:02 PM

fit	test

→ Hold-out sample

↑ Estimation sample

Cross validation

↳ est on 1, eval on 2 then est on 2, eval on 1

↳ 2 fold

3 fold → est on 2 groups, eval on last

Can have as many folds as data size

↳ n-fold or leave one out → LOOCV

LOOCV w/ OLS can get the answer w/ a matrix calculation

Cross RMSE → $\sqrt{\frac{(\hat{y}_i - y_i)^2}{n}}$

w/ normal residuals, AIC → close to best model

BIC ↗

LOOCV is best model

↳ same if residuals are not normal

13.4 Model Selection 4

Saturday, March 6, 2021 6:26 PM

$$MSE_{f_1} = \frac{k}{n} \sum_{i \in f_1} (\hat{y}_i - y_i)^2$$

$$n_{f_1} = \frac{N}{k}$$

$$RMSE_{f_1} = \sqrt{\frac{k}{n} \sum_{i \in f_1} (\hat{y}_i - y_i)^2}$$

$$MSE_{all} = \sum_i (\hat{y}_i - y_i)^2 / N$$

$$RMSE_{all} = \sqrt{\frac{1}{k} \sum_{f=1}^{k} RMSE_f^2}$$

13.8 Forecasting 3

Wednesday, March 10, 2021 4:18 PM

Undoing differencing and log transforms

Y not ΔY, $\ln Y$, or $\Delta \ln Y$

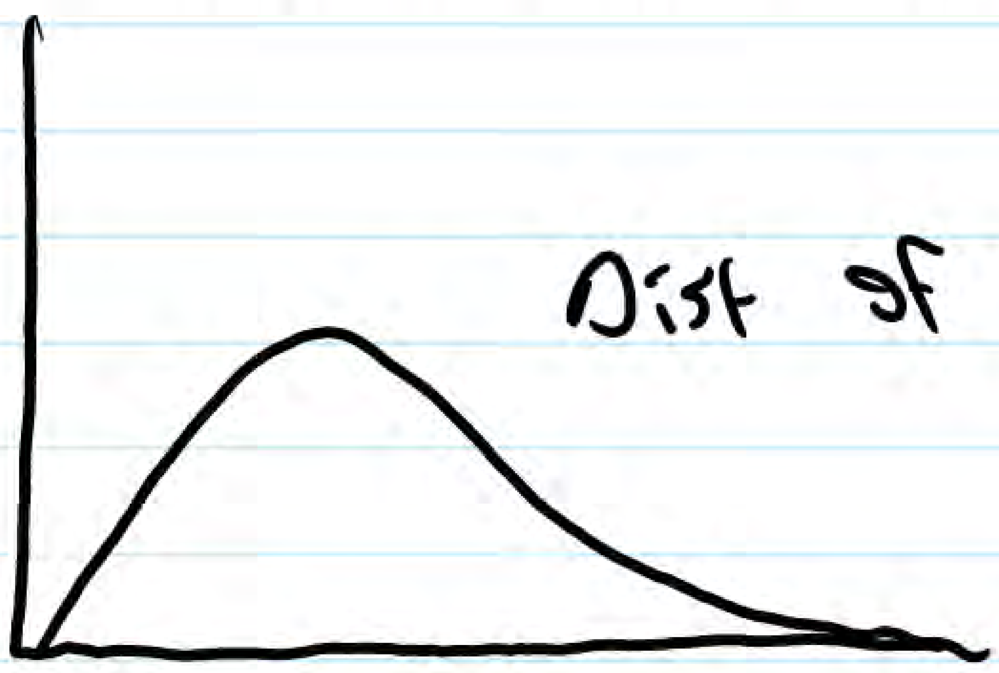
$\hat{f}_{t,h}$ $\hat{f}_{t,1}$ = forecast at t of $t+1$

$\Delta Y_{t+1} = Y_{t+1} - Y_t$

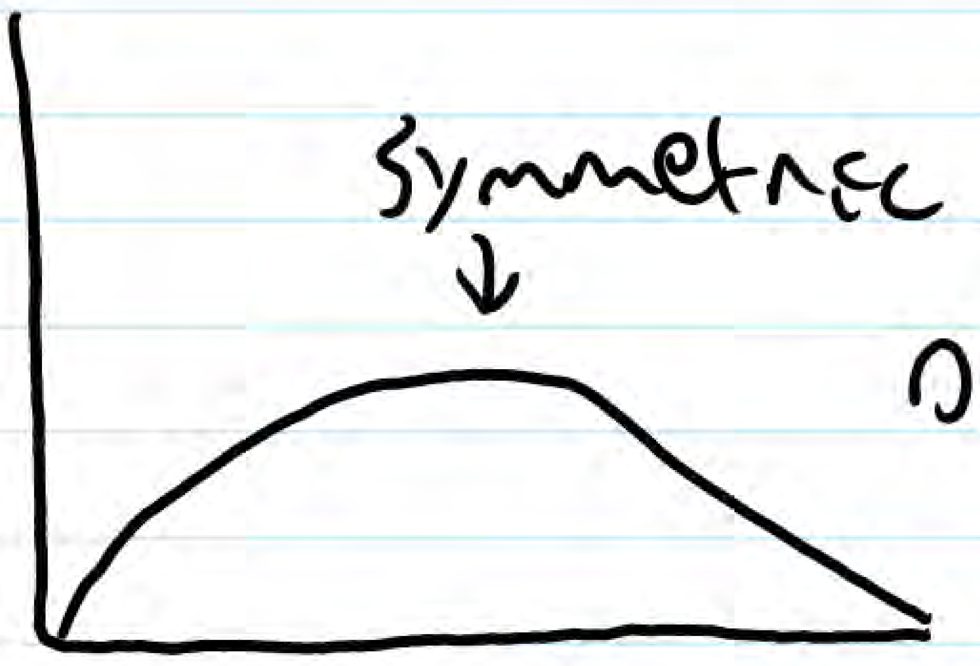
$Y_{t+1} = Y_t + \Delta Y_{t+1}$

$\ln Y_{t+1} \rightarrow Y_{t+1} = e^{\ln Y_{t+1}}$

$\Delta \ln Y_{t+1} = \ln Y_{t+1} - \ln Y_t$



Dist of Y → log normal



Dist of Y if $\ln Y$ is normal

$e^{\frac{1}{2} rmse^2}$ $e^{\ln Y_t + \widehat{\Delta \ln Y_{t+1}}}$

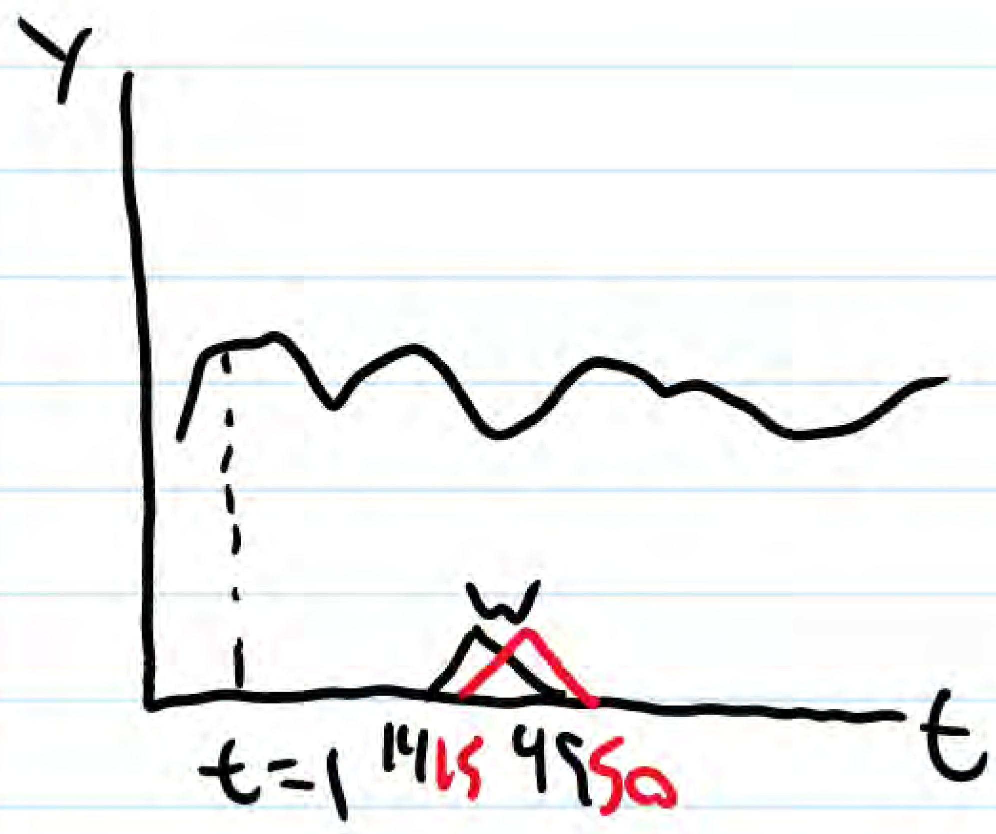
assumes Normality Empirical Version

$\sum_i e^{\hat{r}_i}/T$

$(e^{\ln Y_t + \widehat{\Delta \ln Y_{t+1}}})(\sum_i e^{\hat{r}_i}/T)$

19.1 Rolling Window Estimation

Friday, March 19, 2021 9:50 AM



Window width = w

$\hat{Y}_{50} = f(I_{49})$

$\hat{Y}_{51} = f(I_{50})$

$\hat{Y}_{52} = f(I_{51})$

...

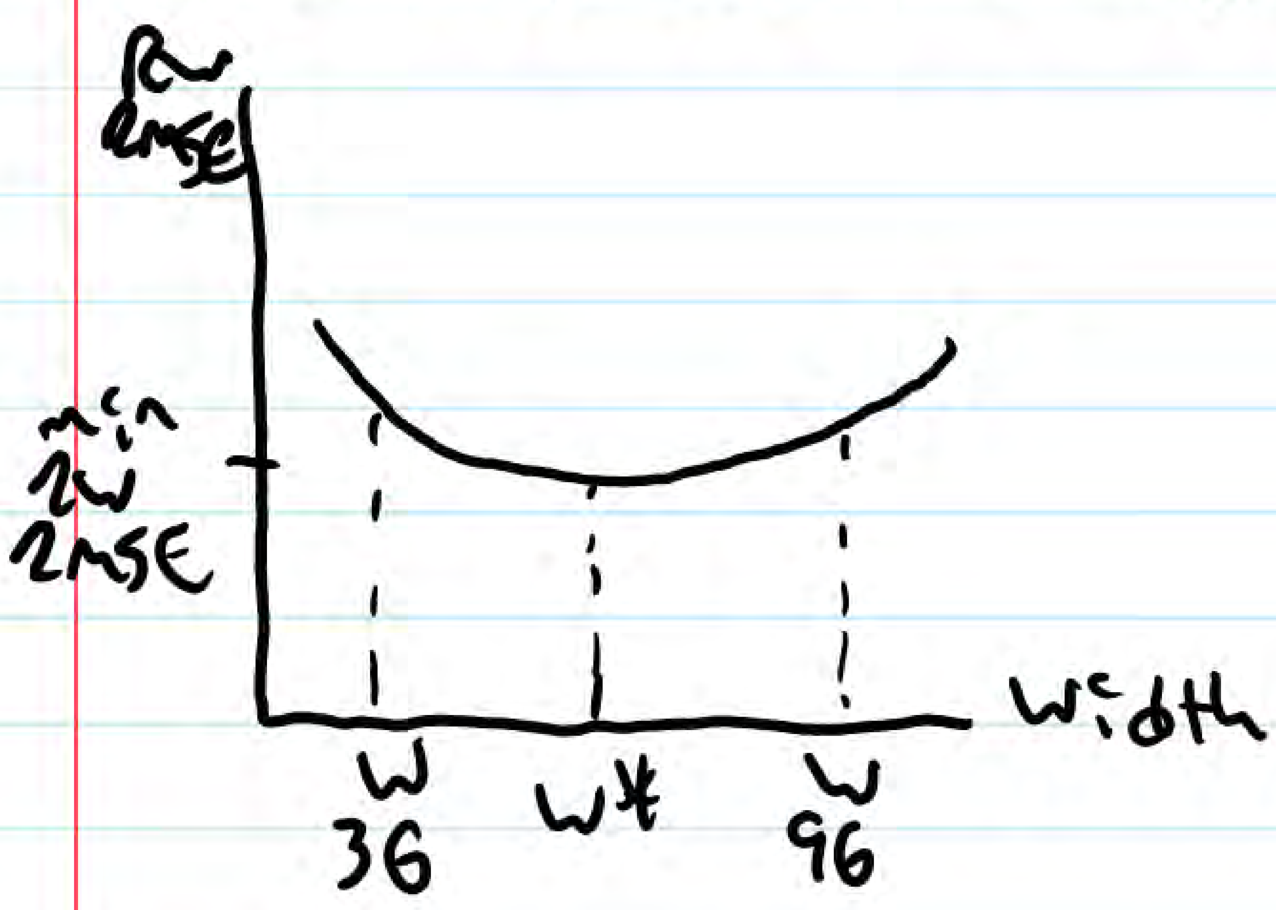
$\hat{Y}_{200} = f(I_{199})$

$e_{50} = Y_{50} - \hat{Y}_{50}$

$\hat{e} = y - \hat{y}$

$\sqrt{\Sigma e^2 / n}$ = Rolling window RMSE

forecast interval = $\hat{Y}_{max+1} \pm 1.96 \cdot RWRMSE$



22.2 Seasonal Adjustments and Differencing

Monday, April 12, 2021 2:43 PM

Additive or multiplicative seasonal adjustments

$$Y_t = \rho_1 Y_{t-1} + \beta_1 X_{t-1} + \underset{\text{coefficient}}{\phi_{July}} \cdot \underset{1 \text{ or } 0}{D_{July}} + \phi_{Dec} \cdot D_{Dec}$$

$$\ln Y_t = \rho_1 \ln Y_{t-1} + \beta_1 \ln X_{t-1} + \phi_{July} \cdot D_{July} + \phi_{Dec} \cdot D_{Dec}$$

Problem Set 3

> All corrections are underlined

1

A

30 Observations	df	AIC	BIC	K(10)RMSE	LOORMSE
Model 1	2	96.287195	98.951604	.95207289	1.3616366
Model 2	2	91.889926	94.4816	.90405284	**1.356985**
Model 3	2	90.028458	**92.544651**	.96983151	1.443163
Model 12	3	93.570654	97.458165	.98902607	1.3845023
Model 13	3	91.221146	94.995435	1.0090104	1.4675022
Model 23	3	**89.685273**	93.459562	.98595497	1.4442447
Model 123	4	90.613095	95.645481	**.90078153**	1.4592071

B

300 Observations	df	AIC	BIC	K(10)RMSE	LOORMSE
Model 1	2	922.28935	929.68354	.81262585	1.1338151
Model 2	2	909.51605	916.90352	.79089382	1.1218131
Model 3	2	926.4476	933.82832	.81547326	1.1579294
Model 12	3	873.69372	**884.77491**	**.74679534**	1.0577271
Model 13	3	914.62743	925.6985	.80468719	1.1355557
Model 23	3	906.59798	917.66906	.79506088	1.1173397
Model 123	4	**872.61172**	887.37316	.74912768	**1.0557316**

C

> 2. Compare model selection using the four criteria. Did they agree in both cases (n=30 and n=300)? For each criterion, are the differences between the “best” model and the next best more or less pronounced with more data?

Note: I know I probably should have set a seed for the data, but because the objective of the exercise is to compare how results compare between datasets of different sizes, I didn't.

In the smaller dataset, the best model differs greatly between the tests. In the larger dataset, the results are more consistent where the model with lags 1 and 2 and 1, 2, and 3 are split 50/50 between the four tests. In the smaller dataset, the results of each test are much closer for the AIC and BIC while for the crossfold and LOOCV the differences between results are similar for the small and large datasets. That said, while the spread is similar, the actual values are different.

2

Dataset	df	AIC	BIC	K(10)RMSE	LOORMSE
Model 1	64	-3105.8368	-2853.1626	.00579331	.00746943
Model 2	34	-3164.4562	-3029.3467	**.00489408**	**.00634841**
Model 3	33	-2874.5257	-2744.2405	.00603002	.00808926
Model 4	33	**-3612.8403**	**-3472.7828**	.00734835	.01062333

> Use the data you used for problem set 1 and 2, prepared for analysis in the same way. Make a table with the model selection measures for each model below. Based on the information in that table, which model do you think is best? Explain why.

Models 2 and 4 each have half the measurements on their side. Model 2 has the crossfold and LOOCV while model 4 has the AIC and BIC. I would choose model 2 because it has more degrees of freedom and the difference between model 2 and 4's AIC and BIC is smaller proportionally than the difference between the crossfold and LOOCV.

Appendix A

```
log using "Problem Set 3", replace
*1a
clear
set obs 30
gen t=[_n]
tsset t
gen r=rnormal()
gen y=r if t<4
```

```
replace y=0.5+0.5*l.y-0.1*l2.y+0.25*l3.y+r if t>=4
drop r

*model
reg d.y l1d.y
*aic/bic
estat ic
scalar define df1=el(r(S),1,4)
scalar define aic1=el(r(S),1,5)
scalar define bic1=el(r(S),1,6)
*10 fold cv rmse
crossfold reg d.y l1d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse1=(el(kMSE,1,1)/k)^.5
scalar list krmse1
matrix drop kMSE
scalar drop k
*loocv
loocv reg d.y l1d.y
scalar define loormse1=r(rmse)

reg d.y l2d.y
estat ic
scalar define df2=el(r(S),1,4)
scalar define aic2=el(r(S),1,5)
scalar define bic2=el(r(S),1,6)
crossfold reg d.y l2d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse2=(el(kMSE,1,1)/k)^.5
scalar list krmse2
matrix drop kMSE
scalar drop k
loocv reg d.y l2d.y
scalar define loormse2=r(rmse)

reg d.y l3d.y
estat ic
scalar define df3=el(r(S),1,4)
scalar define aic3=el(r(S),1,5)
scalar define bic3=el(r(S),1,6)
crossfold reg d.y l3d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse3=(el(kMSE,1,1)/k)^.5
scalar list krmse3
matrix drop kMSE
```

```
scalar drop k
loocv reg d.y l3d.y
scalar define loormse3=r(rmse)

reg d.y l1d.y l2d.y
estat ic
scalar define df12=el(r(S),1,4)
scalar define aic12=el(r(S),1,5)
scalar define bic12=el(r(S),1,6)
crossfold reg d.y l1d.y l2d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse12=(el(kMSE,1,1)/k)^.5
scalar list krmse12
matrix drop kMSE
scalar drop k
loocv reg d.y l1d.y l2d.y
scalar define loormse12=r(rmse)

reg d.y l1d.y l3d.y
estat ic
scalar define df13=el(r(S),1,4)
scalar define aic13=el(r(S),1,5)
scalar define bic13=el(r(S),1,6)
crossfold reg d.y l1d.y l3d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse13=(el(kMSE,1,1)/k)^.5
scalar list krmse13
matrix drop kMSE
scalar drop k
loocv reg d.y l1d.y l3d.y
scalar define loormse13=r(rmse)

reg d.y       l2d.y l3d.y
estat ic
scalar define df23=el(r(S),1,4)
scalar define aic23=el(r(S),1,5)
scalar define bic23=el(r(S),1,6)
crossfold reg d.y       l2d.y l3d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse23=(el(kMSE,1,1)/k)^.5
scalar list krmse23
matrix drop kMSE
scalar drop k
loocv reg d.y       l2d.y l3d.y
scalar define loormse23=r(rmse)
```

```
reg d.y l1d.y l2d.y l3d.y
estat ic
scalar define df123=el(r(S),1,4)
scalar define aic123=el(r(S),1,5)
scalar define bic123=el(r(S),1,6)
crossfold reg d.y l1d.y l2d.y l3d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse123=(el(kMSE,1,1)/k)^.5
scalar list krmse123
matrix drop kMSE
scalar drop k
loocv reg d.y l1d.y l2d.y l3d.y
scalar define loormse123=r(rmse)

matrix drop _all
matrix fit1=(df1,aic1,bic1,krmse1,loormse1)
matrix fit2=(df2,aic2,bic2,krmse2,loormse2)
matrix fit3=(df3,aic3,bic3,krmse3,loormse3)
matrix fit12=(df12,aic12,bic12,krmse12,loormse12)
matrix fit13=(df13,aic13,bic13,krmse13,loormse13)
matrix fit23=(df23,aic23,bic23,krmse23,loormse23)
matrix fit123=(df123,aic123,bic123,krmse123,loormse123)
matrix FIT=fit1\fit2\fit3\fit12\fit13\fit23\fit123
matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 12" "Model 13" "Model 23" "Model 123"
matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE
matrix list FIT

*1b
clear
set obs 300
gen t=[_n]
tsset t
gen r=rnormal()
gen y=r if t<4
replace y=0.5+0.5*l.y-0.1*l2.y+0.25*l3.y+r if t>=4
drop r

*model
reg d.y l1d.y
*aic/bic
estat ic
scalar define df1=el(r(S),1,4)
scalar define aic1=el(r(S),1,5)
scalar define bic1=el(r(S),1,6)
*10 fold cv rmse
```

```
crossfold reg d.y l1d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse1=(el(kMSE,1,1)/k)^.5
scalar list krmse1
matrix drop kMSE
scalar drop k
*loocv
loocv reg d.y l1d.y
scalar define loormse1=r(rmse)

reg d.y l2d.y
estat ic
scalar define df2=el(r(S),1,4)
scalar define aic2=el(r(S),1,5)
scalar define bic2=el(r(S),1,6)
crossfold reg d.y l2d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse2=(el(kMSE,1,1)/k)^.5
scalar list krmse2
matrix drop kMSE
scalar drop k
loocv reg d.y l2d.y
scalar define loormse2=r(rmse)

reg d.y l3d.y
estat ic
scalar define df3=el(r(S),1,4)
scalar define aic3=el(r(S),1,5)
scalar define bic3=el(r(S),1,6)
crossfold reg d.y l3d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse3=(el(kMSE,1,1)/k)^.5
scalar list krmse3
matrix drop kMSE
scalar drop k
loocv reg d.y l3d.y
scalar define loormse3=r(rmse)

reg d.y l1d.y l2d.y
estat ic
scalar define df12=el(r(S),1,4)
scalar define aic12=el(r(S),1,5)
scalar define bic12=el(r(S),1,6)
crossfold reg d.y l1d.y l2d.y
scalar define k=10
```

```
matrix kMSE=r(est)'*r(est)
scalar krmse12=(el(kMSE,1,1)/k)^.5
scalar list krmse12
matrix drop kMSE
scalar drop k
loocv reg d.y l1d.y l2d.y
scalar define loormse12=r(rmse)

reg d.y l1d.y l3d.y
estat ic
scalar define df13=el(r(S),1,4)
scalar define aic13=el(r(S),1,5)
scalar define bic13=el(r(S),1,6)
crossfold reg d.y l1d.y l3d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse13=(el(kMSE,1,1)/k)^.5
scalar list krmse13
matrix drop kMSE
scalar drop k
loocv reg d.y l1d.y l3d.y
scalar define loormse13=r(rmse)

reg d.y       l2d.y l3d.y
estat ic
scalar define df23=el(r(S),1,4)
scalar define aic23=el(r(S),1,5)
scalar define bic23=el(r(S),1,6)
crossfold reg d.y       l2d.y l3d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse23=(el(kMSE,1,1)/k)^.5
scalar list krmse23
matrix drop kMSE
scalar drop k
loocv reg d.y       l2d.y l3d.y
scalar define loormse23=r(rmse)

reg d.y l1d.y l2d.y l3d.y
estat ic
scalar define df123=el(r(S),1,4)
scalar define aic123=el(r(S),1,5)
scalar define bic123=el(r(S),1,6)
crossfold reg d.y l1d.y l2d.y l3d.y
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse123=(el(kMSE,1,1)/k)^.5
scalar list krmse123
```

```
matrix drop kMSE
scalar drop k
loocv reg d.y l1d.y l2d.y l3d.y
scalar define loormse123=r(rmse)

matrix drop _all
matrix fit1=(df1,aic1,bic1,krmse1,loormse1)
matrix fit2=(df2,aic2,bic2,krmse2,loormse2)
matrix fit3=(df3,aic3,bic3,krmse3,loormse3)
matrix fit12=(df12,aic12,bic12,krmse12,loormse12)
matrix fit13=(df13,aic13,bic13,krmse13,loormse13)
matrix fit23=(df23,aic23,bic23,krmse23,loormse23)
matrix fit123=(df123,aic123,bic123,krmse123,loormse123)
matrix FIT=fit1\fit2\fit3\fit12\fit13\fit23\fit123
matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 12" "Model 13" "Model 23" "Model 123"
matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE
matrix list FIT

*2
clear
cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Problem Set 3"
import delimited "Assignment_1_Monthly.txt"

rename lnu02300000 us_epr
rename flnan fl_nonfarm
rename fllfn fl_lf
rename flbppriv fl_bp
rename date datestring

*2d Generate a monthly date variable (make its display format monthly time, %tm)
gen datec=date(datestring, "YMD")
gen date=mofd(datec)
format date %tm

*2e tsset your data
tsset date
gen month=month(datec)

*2f
gen lnusepr=log(us_epr)
gen lnflnonfarm=log(fl_nonfarm)
gen lnfllf=log(fl_lf)
gen lnflbp=log(fl_bp)

*model
```

```
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf l(0/12)d.lnusepr
l(0/12)d.lnflbp i.month date
*aic/bic
estat ic
scalar define df1=el(r(S),1,4)
scalar define aic1=el(r(S),1,5)
scalar define bic1=el(r(S),1,6)
*10 fold cv rmse
crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf
l(0/12)d.lnusepr l(0/12)d.lnflbp i.month date
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse1=(el(kMSE,1,1)/k)^.5
scalar list krmse1
matrix drop kMSE
scalar drop k
*loocv
loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf l(0/12)d.lnusepr
l(0/12)d.lnflbp i.month date
scalar define loormse1=r(rmse)

reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr
l(0/2)d.lnflbp i.month date
estat ic
scalar define df2=el(r(S),1,4)
scalar define aic2=el(r(S),1,5)
scalar define bic2=el(r(S),1,6)
crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr
l(0/2)d.lnflbp i.month date
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse2=(el(kMSE,1,1)/k)^.5
scalar list krmse2
matrix drop kMSE
scalar drop k
loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr
l(0/2)d.lnflbp i.month date
scalar define loormse2=r(rmse)

reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf l(0/2,12)d.lnflbp
i.month date
estat ic
scalar define df3=el(r(S),1,4)
scalar define aic3=el(r(S),1,5)
scalar define bic3=el(r(S),1,6)
crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf
l(0/2,12)d.lnflbp i.month date
scalar define k=10
```

```
matrix kMSE=r(est)'*r(est)
scalar krmse3=(el(kMSE,1,1)/k)^.5
scalar list krmse3
matrix drop kMSE
scalar drop k
loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf l(0/2,12)d.lnflbp
i.month date
scalar define loormse3=r(rmse)

reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month
estat ic
scalar define df4=el(r(S),1,4)
scalar define aic4=el(r(S),1,5)
scalar define bic4=el(r(S),1,6)
crossfold reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month
scalar define k=10
matrix kMSE=r(est)'*r(est)
scalar krmse4=(el(kMSE,1,1)/k)^.5
scalar list krmse4
matrix drop kMSE
scalar drop k
loocv reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month
scalar define loormse4=r(rmse)

matrix drop _all
matrix fit1=(df1,aic1,bic1,krmse1,loormse1)
matrix fit2=(df2,aic2,bic2,krmse2,loormse2)
matrix fit3=(df3,aic3,bic3,krmse3,loormse3)
matrix fit4=(df4,aic4,bic4,krmse4,loormse4)
matrix FIT=fit1\fit2\fit3\fit4
matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 4"
matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE
matrix list FIT

log close
```

Appendix B

```
      name:  <unnamed>
       log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 - Time Series/Problem Sets/Problem Set 3/Problem Set 3.smcl
  log type:  smcl
 opened on:  15 Mar 2021, 19:19:30

. *1a
. clear

. set obs 30
number of observations (_N) was 0, now 30
```

```
. gen t=[_n]

. tsset t
        time variable:  t, 1 to 30
                delta:  1 unit

. gen r=rnormal()

. gen y=r if t<4
(27 missing values generated)

. replace y=0.5+0.5*l.y-0.1*l2.y+0.25*l3.y+r if t>=4
(27 real changes made)

. drop r

.
. *model
. reg d.y l1d.y

      Source |       SS           df       MS      Number of obs   =        28
-------------+----------------------------------   F(1, 26)        =      7.29
       Model |  5.98982946         1  5.98982946   Prob > F        =    0.0120
    Residual |  21.3591043        26  .821504011   R-squared       =    0.2190
-------------+----------------------------------   Adj R-squared   =    0.1890
       Total |  27.3489337        27  1.01292347   Root MSE        =    .90637

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
         LD. |  -.4480631   .1659346    -2.70   0.012    -.7891465   -.1069798
             |
       _cons |   .0946765   .1720709     0.55   0.587    -.2590203    .4483734
------------------------------------------------------------------------------

. *aic/bic
. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |         28   -39.4009  -35.94011       2    75.88022   78.54463
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df1=el(r(S),1,4)

. scalar define aic1=el(r(S),1,5)

. scalar define bic1=el(r(S),1,6)

. *10 fold cv rmse
. crossfold reg d.y l1d.y

             |      RMSE
-------------+-----------
        est1 |  .7602895
        est2 |  .6312166
        est3 |  1.110403
        est4 |  1.269966
        est5 |  .4469951

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse1=(el(kMSE,1,1)/k)^.5

. scalar list krmse1
    krmse1 =  .63419929

. matrix drop kMSE

. scalar drop k

. *loocv
. loocv reg d.y l1d.y

 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   .95384454
Mean Absolute Errors     |   .77314162
Pseudo-R2                |   .0999433
-----------------------------------------

. scalar define loormse1=r(rmse)
```

```
.
. reg d.y l2d.y

      Source |       SS           df       MS      Number of obs   =        27
-------------+----------------------------------   F(1, 25)        =      0.14
       Model |  .141091522         1  .141091522   Prob > F        =    0.7148
    Residual |  25.8136069        25  1.03254427   R-squared       =    0.0054
-------------+----------------------------------   Adj R-squared   =   -0.0343
       Total |  25.9546984        26  .99825763    Root MSE        =    1.0161

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
        L2D. |    -.06877   .1860386    -0.37   0.715    -.4519236    .3143836
             |
       _cons |   .1002526   .1964497     0.51   0.614    -.3043431    .5048482
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |         27   -37.7783  -37.70472       2    79.40943   82.00111
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df2=el(r(S),1,4)

. scalar define aic2=el(r(S),1,5)

. scalar define bic2=el(r(S),1,6)

. crossfold reg d.y l2d.y

             |      RMSE
-------------+-----------
        est1 |  .9021655
        est2 |  .7459198
        est3 |  1.167232
        est4 |  1.243048
        est5 |  1.326711

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse2=(el(kMSE,1,1)/k)^.5

. scalar list krmse2
    krmse2 =  .77704956

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l2d.y

 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   1.110311
Mean Absolute Errors     |   .8857642
Pseudo-R2                |   .30515038
-----------------------------------------

. scalar define loormse2=r(rmse)

.
. reg d.y l3d.y

      Source |       SS           df       MS      Number of obs   =        26
-------------+----------------------------------   F(1, 24)        =      0.09
       Model |  .100885437         1  .100885437   Prob > F        =    0.7619
    Residual |   25.778986        24  1.07412442   R-squared       =    0.0039
-------------+----------------------------------   Adj R-squared   =   -0.0376
       Total |  25.8798714        25  1.03519486   Root MSE        =    1.0364

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
        L3D. |   .0592101   .1932007     0.31   0.762    -.3395366    .4579567
             |
       _cons |   .0747593   .2050307     0.36   0.719    -.3484032    .4979218
------------------------------------------------------------------------------
```

```
. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |         26   -36.8322  -36.78142       2    77.56284   80.07904
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df3=el(r(S),1,4)

. scalar define aic3=el(r(S),1,5)

. scalar define bic3=el(r(S),1,6)

. crossfold reg d.y l3d.y

             |      RMSE
-------------+-----------
        est1 |  1.260278
        est2 |  .5783534
        est3 |  .4936832
        est4 |  1.409555
        est5 |  1.043313

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse3=(el(kMSE,1,1)/k)^.5

. scalar list krmse3
    krmse3 =  .72400711

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l3d.y


 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   1.1233063
Mean Absolute Errors     |   .89965294
Pseudo-R2                |   .59352903
-----------------------------------------

. scalar define loormse3=r(rmse)

.
. reg d.y l1d.y l2d.y

      Source |       SS           df       MS      Number of obs   =        27
-------------+----------------------------------   F(2, 24)        =      4.75
       Model |  7.35816653         2  3.67908326   Prob > F        =    0.0183
    Residual |  18.5965319        24  .774855494   R-squared       =    0.2835
-------------+----------------------------------   Adj R-squared   =    0.2238
       Total |  25.9546984        26  .99825763   Root MSE        =    .88026

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
         LD. |  -.5821105   .1907371    -3.05   0.005    -.9757726   -.1884485
        L2D. |  -.3293629   .1823834    -1.81   0.083    -.7057837     .047058
             |
       _cons |   .1501327   .1709626     0.88   0.389    -.2027167     .502982
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |         27   -37.7783  -33.27771       3    72.55541   76.44292
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df12=el(r(S),1,4)

. scalar define aic12=el(r(S),1,5)

. scalar define bic12=el(r(S),1,6)

. crossfold reg d.y l1d.y l2d.y
```

```
. crossfold reg d.y l1d.y l2d.y

             |      RMSE
-------------+-----------
        est1 |  .2755685
        est2 |  1.280924
        est3 |  .8959287
        est4 |  .8905425
        est5 |   .782406

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse12=(el(kMSE,1,1)/k)^.5

. scalar list krmse12
   krmse12 =  .62646722

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l1d.y l2d.y


 Leave-One-Out Cross-Validation Results 
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   .97005343
Mean Absolute Errors     |   .76316726
Pseudo-R2                |   .13999295
-----------------------------------------

. scalar define loormse12=r(rmse)

. 
. reg d.y l1d.y l3d.y

      Source |       SS           df       MS      Number of obs   =        26
-------------+----------------------------------   F(2, 23)        =      2.64
       Model |  4.83662497         2  2.41831248   Prob > F        =    0.0926
    Residual |  21.0432464        23  .914923758   R-squared       =    0.1869
-------------+----------------------------------   Adj R-squared   =    0.1162
       Total |  25.8798714        25  1.03519486   Root MSE        =    .95652

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
         LD. |  -.4285231   .1883531    -2.28   0.033    -.8181613    -.038885
        L3D. |   .0320856   .1787074     0.18   0.859    -.3375988      .40177
             |
       _cons |   .1149156   .1900488     0.60   0.551    -.2782303    .5080615
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |         26   -36.8322  -34.14268       3    74.28537   78.05965
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df13=el(r(S),1,4)

. scalar define aic13=el(r(S),1,5)

. scalar define bic13=el(r(S),1,6)

. crossfold reg d.y l1d.y l3d.y

             |      RMSE
-------------+-----------
        est1 |  .8978038
        est2 |  .7710074
        est3 |  1.453003
        est4 |  .9490036
        est5 |  .5381048

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse13=(el(kMSE,1,1)/k)^.5

. scalar list krmse13
   krmse13 =  .68570296
```

```
. matrix drop kMSE

. scalar drop k

. loocv reg d.y l1d.y l3d.y

 Leave-One-Out Cross-Validation Results
-----------------------------------------------
         Method          |    Value
-------------------------+---------------------
Root Mean Squared Errors |    1.0320119
Mean Absolute Errors     |    .83683368
Pseudo-R2                |    .07234527
-----------------------------------------------

. scalar define loormse13=r(rmse)

.
. reg d.y       l2d.y l3d.y

      Source |       SS           df       MS      Number of obs   =        26
-------------+----------------------------------   F(2, 23)        =      0.11
       Model |  .237741921         2  .118870961   Prob > F        =    0.8993
    Residual |  25.6421295        23   1.1148752   R-squared       =    0.0092
-------------+----------------------------------   Adj R-squared   =   -0.0770
       Total |  25.8798714        25  1.03519486   Root MSE        =    1.0559

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
        L2D. |  -.0805558   .2299203    -0.35   0.729    -.5561822    .3950705
        L3D. |   .0218562   .2238511     0.10   0.923     -.441215    .4849274
             |
       _cons |   .0832065   .2102705     0.40   0.696    -.3517713    .5181842
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |         26   -36.8322  -36.71222       3    79.42445   83.19874
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df23=el(r(S),1,4)

. scalar define aic23=el(r(S),1,5)

. scalar define bic23=el(r(S),1,6)

. crossfold reg d.y       l2d.y l3d.y

             |      RMSE
-------------+-----------
        est1 |  1.202326
        est2 |  1.176127
        est3 |  .7737328
        est4 |  1.260123
        est5 |   1.24396

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse23=(el(kMSE,1,1)/k)^.5

. scalar list krmse23
   krmse23 =  .81011538

. matrix drop kMSE

. scalar drop k

. loocv reg d.y       l2d.y l3d.y

 Leave-One-Out Cross-Validation Results
-----------------------------------------------
         Method          |    Value
-------------------------+---------------------
Root Mean Squared Errors |    1.1743793
Mean Absolute Errors     |    .93166705
Pseudo-R2                |    .6235484
-----------------------------------------------

. scalar define loormse23=r(rmse)
```

```
.
. reg d.y l1d.y l2d.y l3d.y

      Source |       SS           df       MS      Number of obs   =        26
-------------+----------------------------------   F(3, 22)        =      3.31
       Model |  8.04353552         3  2.68117851   Prob > F        =    0.0390
    Residual |  17.8363359        22   .81074254   R-squared       =    0.3108
-------------+----------------------------------   Adj R-squared   =    0.2168
       Total |  25.8798714        25  1.03519486   Root MSE        =    .90041

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
         LD. |  -.6480901   .2088663    -3.10   0.005    -1.081252   -.2149279
        L2D. |  -.4593609   .2309682    -1.99   0.059    -.9383596    .0196377
        L3D. |  -.1948192   .2032633    -0.96   0.348    -.6163615     .226723
             |
       _cons |     .18366     .18221     1.01   0.324    -.1942205    .5615405
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |         26   -36.8322  -31.99324       4    71.98647   77.01886
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df123=el(r(S),1,4)

. scalar define aic123=el(r(S),1,5)

. scalar define bic123=el(r(S),1,6)

. crossfold reg d.y l1d.y l2d.y l3d.y

             |      RMSE
-------------+-----------
        est1 |  1.277871
        est2 |  .4126419
        est3 |  1.187153
        est4 |  .7160594
        est5 |  .7569776

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse123=(el(kMSE,1,1)/k)^.5

. scalar list krmse123
  krmse123 =  .65561544

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l1d.y l2d.y l3d.y

 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   1.0025941
Mean Absolute Errors     |   .78300616
Pseudo-R2                |   .14309964
-----------------------------------------

. scalar define loormse123=r(rmse)

.
. matrix drop _all

. matrix fit1=(df1,aic1,bic1,krmse1,loormse1)

. matrix fit2=(df2,aic2,bic2,krmse2,loormse2)

. matrix fit3=(df3,aic3,bic3,krmse3,loormse3)

. matrix fit12=(df12,aic12,bic12,krmse12,loormse12)

. matrix fit13=(df13,aic13,bic13,krmse13,loormse13)

. matrix fit23=(df23,aic23,bic23,krmse23,loormse23)

. matrix fit123=(df123,aic123,bic123,krmse123,loormse123)
```

```
. matrix FIT=fit1\fit2\fit3\fit12\fit13\fit23\fit123

. matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 12" "Model 13" "Model 23" "Model 123"

. matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE

. matrix list FIT

FIT[7,5]
                 df        AIC        BIC  K(10)RMSE    LOORMSE
  Model 1         2  75.880218  78.544627  .63419929  .95384454
  Model 2         2  79.409433  82.001106  .77704956   1.110311
  Model 3         2  77.562845  80.079038  .72400711  1.1233063
 Model 12         3  72.555413  76.442924  .62646722  .97005343
 Model 13         3  74.285365  78.059655  .68570296  1.0320119
 Model 23         3  79.424447  83.198737  .81011538  1.1743793
Model 123         4  71.986474  77.018861  .65561544  1.0025941

.
. *1b
. clear

. set obs 300
number of observations (_N) was 0, now 300

. gen t=[_n]

. tsset t
        time variable:  t, 1 to 300
                delta:  1 unit

. gen r=rnormal()

. gen y=r if t<4
(297 missing values generated)

. replace y=0.5+0.5*l.y-0.1*l2.y+0.25*l3.y+r if t>=4
(297 real changes made)

. drop r

.
. *model
. reg d.y l1d.y

      Source |       SS           df       MS      Number of obs   =       298
-------------+----------------------------------   F(1, 296)       =     30.29
       Model |  40.2287435         1  40.2287435   Prob > F        =    0.0000
    Residual |   393.11806       296  1.32810155   R-squared       =    0.0928
-------------+----------------------------------   Adj R-squared   =    0.0898
       Total |  433.346804       297  1.45908015   Root MSE        =    1.1524

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
         LD. |  -.3047742   .0553765    -5.50   0.000    -.4137557   -.1957926
             |
       _cons |   .0055929   .0667596     0.08   0.933    -.1257907    .1369765
------------------------------------------------------------------------------

. *aic/bic
. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        298   -478.636  -464.1191       2    932.2383   939.6325
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df1=el(r(S),1,4)

. scalar define aic1=el(r(S),1,5)

. scalar define bic1=el(r(S),1,6)

. *10 fold cv rmse
. crossfold reg d.y l1d.y

             |      RMSE
-------------+-----------
        est1 |  1.070865
        est2 |  1.307489
        est3 |  1.076643
        est4 |  1.144266
        est5 |  1.153222

. scalar define k=10
```

```
. matrix kMSE=r(est)'*r(est)

. scalar krmse1=(el(kMSE,1,1)/k)^.5

. scalar list krmse1
    krmse1 =  .81576385

. matrix drop kMSE

. scalar drop k

. *loocv
. loocv reg d.y l1d.y

 Leave-One-Out Cross-Validation Results

         Method          |    Value
-------------------------+--------------
Root Mean Squared Errors |  1.1602627
Mean Absolute Errors     |   .9370285
Pseudo-R2                |   .08032407
-------------------------+--------------

. scalar define loormse1=r(rmse)

.
. reg d.y l2d.y

      Source |       SS           df       MS      Number of obs   =       297
-------------+----------------------------------   F(1, 295)       =     28.21
       Model |  37.8014875         1  37.8014875   Prob > F        =    0.0000
    Residual |  395.272758       295  1.33990765   R-squared       =    0.0873
-------------+----------------------------------   Adj R-squared   =    0.0842
       Total |  433.074246       296  1.46308867   Root MSE        =    1.1575

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
        L2D. |  -.2956623   .0556646    -5.31   0.000    -.4052123   -.1861123
             |
       _cons |   .0080967   .0671695     0.12   0.904    -.1240954    .1402888
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        297  -477.4355  -463.8726       2    931.7451   939.1326
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df2=el(r(S),1,4)

. scalar define aic2=el(r(S),1,5)

. scalar define bic2=el(r(S),1,6)

. crossfold reg d.y l2d.y

             |      RMSE
-------------+-----------
        est1 |  1.155929
        est2 |  1.318321
        est3 |  1.105786
        est4 |  1.208012
        est5 |  1.006752

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse2=(el(kMSE,1,1)/k)^.5

. scalar list krmse2
    krmse2 =  .82278473

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l2d.y

 Leave-One-Out Cross-Validation Results

         Method          |    Value
-------------------------+--------------
```

```
Root Mean Squared Errors  |   1.164344
Mean Absolute Errors      |   .96184136
Pseudo-R2                 |   .07534286
-------------------------------------------

. scalar define loormse2=r(rmse)

.
. reg d.y l3d.y

      Source |       SS           df       MS      Number of obs   =       296
-------------+----------------------------------   F(1, 294)       =     14.20
       Model |  19.9598466         1  19.9598466   Prob > F        =    0.0002
    Residual |  413.112855       294  1.40514577   R-squared       =    0.0461
-------------+----------------------------------   Adj R-squared   =    0.0428
       Total |  433.072702       295  1.46804306   Root MSE        =    1.1854

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
        L3D. |    .214962   .0570353     3.77   0.000     .1027128    .3272113
             |
       _cons |   .0037681   .0689004     0.05   0.956    -.1318325    .1393686
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        296  -476.3266  -469.3433       2    942.6866   950.0673
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df3=el(r(S),1,4)

. scalar define aic3=el(r(S),1,5)

. scalar define bic3=el(r(S),1,6)

. crossfold reg d.y l3d.y

             |      RMSE
-------------+-----------
        est1 |  1.220885
        est2 |  1.300472
        est3 |  1.206425
        est4 |  1.047378
        est5 |  1.156072

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse3=(el(kMSE,1,1)/k)^.5

. scalar list krmse3
    krmse3 =  .84087751

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l3d.y

 Leave-One-Out Cross-Validation Results
-------------------------------------------
         Method           |    Value
--------------------------+----------------
Root Mean Squared Errors  |   1.1927634
Mean Absolute Errors      |   .97432674
Pseudo-R2                 |   .03508471
-------------------------------------------

. scalar define loormse3=r(rmse)

.
. reg d.y l1d.y l2d.y

      Source |       SS           df       MS      Number of obs   =       297
-------------+----------------------------------   F(2, 294)       =     51.48
       Model |   112.32631         2   56.163155   Prob > F        =    0.0000
    Residual |  320.747936       294  1.09097937   R-squared       =    0.2594
-------------+----------------------------------   Adj R-squared   =    0.2543
       Total |  433.074246       296  1.46308867   Root MSE        =    1.0445

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
```

```
             |
           y |
         LD. |  -.4357756   .0527255    -8.26   0.000    -.5395429   -.3320084
        L2D. |  -.4291487   .0527612    -8.13   0.000    -.5329862   -.3253111
             |
       _cons |   .0117439   .0606114     0.19   0.846    -.1075433    .1310311
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        297  -477.4355  -432.8479       3    871.6957   882.7769
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df12=el(r(S),1,4)

. scalar define aic12=el(r(S),1,5)

. scalar define bic12=el(r(S),1,6)

. crossfold reg d.y l1d.y l2d.y

             |      RMSE
-------------+-----------
        est1 |  .9054228
        est2 |  .9939704
        est3 |  1.153992
        est4 |  1.044548
        est5 |  1.095283

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse12=(el(kMSE,1,1)/k)^.5

. scalar list krmse12
   krmse12 =  .73689822

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l1d.y l2d.y


 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   1.0505398
Mean Absolute Errors     |    .8495225
Pseudo-R2                |    .2471428
-----------------------------------------

. scalar define loormse12=r(rmse)

.
. reg d.y l1d.y l3d.y

      Source |       SS           df       MS      Number of obs   =       296
-------------+----------------------------------   F(2, 293)       =     18.07
       Model |  47.5617458         2  23.7808729   Prob > F        =    0.0000
    Residual |  385.510956       293  1.31573705   R-squared       =    0.1098
-------------+----------------------------------   Adj R-squared   =    0.1037
       Total |  433.072702       295  1.46804306   Root MSE        =    1.1471

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
         LD. |  -.2642997   .0577048    -4.58   0.000    -.3778681   -.1507313
        L3D. |   .1369759   .0577577     2.37   0.018     .0233034    .2506484
             |
       _cons |   .0062375   .0666745     0.09   0.926    -.1249842    .1374591
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        296  -476.3266  -459.1089       3    924.2178   935.2889
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.
```

```
. scalar define df13=el(r(S),1,4)

. scalar define aic13=el(r(S),1,5)

. scalar define bic13=el(r(S),1,6)

. crossfold reg d.y l1d.y l3d.y

             |      RMSE
-------------+-----------
        est1 |  1.144654
        est2 |  1.043574
        est3 |  1.244819
        est4 |   1.14139
        est5 |  1.214143

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse13=(el(kMSE,1,1)/k)^.5

. scalar list krmse13
   krmse13 =  .82010777

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l1d.y l3d.y


 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   1.1555101
Mean Absolute Errors     |    .9348069
Pseudo-R2                |   .09434506
-----------------------------------------

. scalar define loormse13=r(rmse)

.
. reg d.y        l2d.y l3d.y

      Source |       SS           df       MS      Number of obs   =       296
-------------+----------------------------------   F(2, 293)       =     17.08
       Model |   45.215903         2  22.6079515   Prob > F        =    0.0000
    Residual |  387.856799       293  1.32374334   R-squared       =    0.1044
-------------+----------------------------------   Adj R-squared   =    0.0983
       Total |  433.072702       295  1.46804306   Root MSE        =    1.1505

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
        L2D. |  -.2538017    .058105    -4.37   0.000    -.3681577   -.1394456
        L3D. |   .1374621   .0581324     2.36   0.019      .023052    .2518721
             |
       _cons |   .0064038   .0668776     0.10   0.924    -.1252176    .1380252
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        296  -476.3266  -460.0068       3    926.0135   937.0846
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df23=el(r(S),1,4)

. scalar define aic23=el(r(S),1,5)

. scalar define bic23=el(r(S),1,6)

. crossfold reg d.y        l2d.y l3d.y

             |      RMSE
-------------+-----------
        est1 |  1.261762
        est2 |  1.151197
        est3 |  1.047751
        est4 |  1.195764
        est5 |    1.0922

. scalar define k=10
```

```
. matrix kMSE=r(est)'*r(est)

. scalar krmse23=(el(kMSE,1,1)/k)^.5

. scalar list krmse23
   krmse23 =  .81472913

. matrix drop kMSE

. scalar drop k

. loocv reg d.y       l2d.y l3d.y


 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   1.1606538
Mean Absolute Errors     |   .94993411
Pseudo-R2                |   .08627031
-----------------------------------------

. scalar define loormse23=r(rmse)

.
. reg d.y l1d.y l2d.y l3d.y

      Source |       SS           df       MS      Number of obs   =       296
-------------+----------------------------------   F(3, 292)       =     34.57
       Model |  113.494499         3  37.8314998   Prob > F        =    0.0000
    Residual |  319.578203       292   1.0944459   R-squared       =    0.2621
-------------+----------------------------------   Adj R-squared   =    0.2545
       Total |  433.072702       295  1.46804306   Root MSE        =    1.0462

------------------------------------------------------------------------------
         D.y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           y |
         LD. |  -.4616699   .0584503    -7.90   0.000    -.5767071   -.3466327
        L2D. |  -.4554328   .0586774    -7.76   0.000    -.5709171   -.3399486
        L3D. |  -.0603309   .0584902    -1.03   0.303    -.1754466    .0547848
             |
       _cons |   .0128112   .0608155     0.21   0.833    -.1068812    .1325036
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        296  -476.3266  -431.3489       4    870.6978   885.4592
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df123=el(r(S),1,4)

. scalar define aic123=el(r(S),1,5)

. scalar define bic123=el(r(S),1,6)

. crossfold reg d.y l1d.y l2d.y l3d.y

             |      RMSE
-------------+-----------
        est1 |  1.193648
        est2 |  .9485697
        est3 |  1.119405
        est4 |  1.049818
        est5 |  .9232315

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse123=(el(kMSE,1,1)/k)^.5

. scalar list krmse123
  krmse123 =  .74378241

. matrix drop kMSE

. scalar drop k

. loocv reg d.y l1d.y l2d.y l3d.y


 Leave-One-Out Cross-Validation Results
-----------------------------------------
```

Method	Value
Root Mean Squared Errors	1.0551773
Mean Absolute Errors	.85532304
Pseudo-R2	.24451175

```
. scalar define loormse123=r(rmse)

.
. matrix drop _all

. matrix fit1=(df1,aic1,bic1,krmse1,loormse1)

. matrix fit2=(df2,aic2,bic2,krmse2,loormse2)

. matrix fit3=(df3,aic3,bic3,krmse3,loormse3)

. matrix fit12=(df12,aic12,bic12,krmse12,loormse12)

. matrix fit13=(df13,aic13,bic13,krmse13,loormse13)

. matrix fit23=(df23,aic23,bic23,krmse23,loormse23)

. matrix fit123=(df123,aic123,bic123,krmse123,loormse123)

. matrix FIT=fit1\fit2\fit3\fit12\fit13\fit23\fit123

. matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 12" "Model 13" "Model 23" "Model 123"

. matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE

. matrix list FIT

FIT[7,5]
                 df        AIC        BIC  K(10)RMSE    LOORMSE
  Model 1         2  932.23828  939.63247  .81576385  1.1602627
  Model 2         2  931.74513   939.1326  .82278473   1.164344
  Model 3         2  942.68657  950.06729  .84087751  1.1927634
 Model 12         3  871.69575  882.77694  .73689822  1.0505398
 Model 13         3  924.21782   935.2889  .82010777  1.1555101
 Model 23         3  926.01353  937.08461  .81472913  1.1606538
Model 123         4  870.69781  885.45925  .74378241  1.0551773

.
. *2
. clear

. cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 - Time Series/Problem Sets/Problem Set 3"
/Users/guslipkin/Documents/Spring2020/CAP 4763 - Time Series/Problem Sets/Problem Set 3

. import delimited "Assignment_1_Monthly.txt"
(5 vars, 984 obs)

.
. rename lnu02300000 us_epr

. rename flnan fl_nonfarm

. rename fllfn fl_lf

. rename flbppriv fl_bp

. rename date datestring

.
. *2d Generate a monthly date variable (make its display format monthly time, %tm)
. gen datec=date(datestring, "YMD")

. gen date=mofd(datec)

. format date %tm

.
. *2e tsset your data
. tsset date
        time variable:  date, 1939m1 to 2020m12
                delta:  1 month

. gen month=month(datec)

.
. *2f
. gen lnusepr=log(us_epr)
(108 missing values generated)

. gen lnflnonfarm=log(fl_nonfarm)

. gen lnfllf=log(fl_lf)
(444 missing values generated)

. gen lnflbp=log(fl_bp)
```

```
(588 missing values generated)

.
. *model
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf l(0/12)d.lnusepr l(0/12)d.lnflbp i.month date

      Source |       SS           df       MS      Number of obs   =       383
-------------+----------------------------------   F(63, 319)      =     53.24
       Model |  .050768432        63  .000805848   Prob > F        =    0.0000
    Residual |  .004828206       319  .000015135   R-squared       =    0.9132
-------------+----------------------------------   Adj R-squared   =    0.8960
       Total |  .055596638       382  .000145541   Root MSE        =    .00389

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.3187209   .0520581    -6.12   0.000    -.4211416   -.2163003
        L2D. |  -.1993953   .0546776    -3.65   0.000    -.3069696    -.091821
        L3D. |    .004615   .0556328     0.08   0.934    -.1048386    .1140686
        L4D. |   .0520114   .0553102     0.94   0.348    -.0568075    .1608302
        L5D. |  -.0290364   .0550957    -0.53   0.599    -.1374333    .0793605
        L6D. |  -.0411661   .0575557    -0.72   0.475    -.1544029    .0720706
        L7D. |  -.0589717   .0560994    -1.05   0.294    -.1693433       .0514
        L8D. |    -.06692   .0555654    -1.20   0.229    -.1762409     .042401
        L9D. |   .0735794   .0592884     1.24   0.216    -.0430663     .190225
       L10D. |  -.1280257   .0580919    -2.20   0.028    -.2423174    -.013734
       L11D. |  -.0562037   .0581423    -0.97   0.334    -.1705946    .0581872
       L12D. |   .4026367   .0587393     6.85   0.000     .2870713    .5182021
             |
      lnfllf |
         D1. |  -.4678501    .091836    -5.09   0.000    -.6485309   -.2871693
         LD. |  -.1874345   .0955731    -1.96   0.051    -.3754677    .0005987
        L2D. |  -.1532159   .0957994    -1.60   0.111    -.3416944    .0352626
        L3D. |    .186449   .0979739     1.90   0.058    -.0063077    .3792057
        L4D. |   .0760726   .0990118     0.77   0.443     -.118726    .2708713
        L5D. |   .0351072    .099036     0.35   0.723    -.1597391    .2299535
        L6D. |   .0231131   .0990097     0.23   0.816    -.1716814    .2179075
        L7D. |   .1259544   .0974149     1.29   0.197    -.0657025    .3176112
        L8D. |    .024699   .0982434     0.25   0.802    -.1685878    .2179858
        L9D. |   .1150141   .1100678     1.04   0.297    -.1015364    .3315646
       L10D. |   .2080239   .1108839     1.88   0.062    -.0101322      .42618
       L11D. |  -.0752819   .1117702    -0.67   0.501    -.2951818    .1446179
       L12D. |  -.1248653   .1111426    -1.12   0.262    -.3435303    .0937998
             |
     lnusepr |
         D1. |   1.058897   .0618861    17.11   0.000      .937141    1.180654
         LD. |    .376984    .084734     4.45   0.000     .2102759    .5436921
        L2D. |   .2272904   .0866516     2.62   0.009     .0568095    .3977713
        L3D. |  -.1902081    .087627    -2.17   0.031    -.3626078   -.0178083
        L4D. |     .00045   .0885154     0.01   0.996    -.1736977    .1745976
        L5D. |  -.0261134   .0874342    -0.30   0.765    -.1981339    .1459071
        L6D. |   .1609474   .0889456     1.81   0.071    -.0140467    .3359415
        L7D. |   .0157551   .0888144     0.18   0.859    -.1589808     .190491
        L8D. |   .0633536   .0901157     0.70   0.483    -.1139427    .2406498
        L9D. |   .0217037   .1470597     0.15   0.883    -.2676257    .3110332
       L10D. |  -.1641384   .1480045    -1.11   0.268    -.4553266    .1270498
       L11D. |   .2149393   .1460384     1.47   0.142    -.0723808    .5022593
       L12D. |   .0161647   .1416189     0.11   0.909    -.2624602    .2947897
             |
      lnflbp |
         D1. |   .0043768   .0017545     2.49   0.013      .000925    .0078286
         LD. |   .0049935   .0020942     2.38   0.018     .0008732    .0091137
        L2D. |   .0051854   .0022195     2.34   0.020     .0008187     .009552
        L3D. |   .0065461   .0022284     2.94   0.004      .002162    .0109302
        L4D. |   .0055029   .0022297     2.47   0.014      .001116    .0098898
        L5D. |   .0067131   .0022143     3.03   0.003     .0023567    .0110696
        L6D. |   .0071018   .0022379     3.17   0.002      .002699    .0115047
        L7D. |    .007351    .002247     3.27   0.001     .0029301    .0117719
        L8D. |   .0054368   .0023044     2.36   0.019     .0009031    .0099705
        L9D. |    .004466   .0022983     1.94   0.053    -.0000558    .0089878
       L10D. |   .0061294   .0022689     2.70   0.007     .0016656    .0105932
       L11D. |   .0032694   .0021733     1.50   0.133    -.0010063    .0075452
       L12D. |   .0026534    .001789     1.48   0.139    -.0008664    .0061731
             |
       month |
          2  |  -.0019614   .0032441    -0.60   0.546    -.0083439    .0044211
          3  |  -.0024974   .0033763    -0.74   0.460     -.00914     .0041452
          4  |   -.007529   .0038677    -1.95   0.052    -.0151384    .0000805
          5  |  -.0039676   .0033266    -1.19   0.234    -.0105125    .0025772
          6  |  -.0144882   .0036057    -4.02   0.000    -.0215823   -.0073942
          7  |  -.0097851   .0032285    -3.03   0.003     -.016137   -.0034332
          8  |  -.0000542   .0035161    -0.02   0.988    -.0069719    .0068635
          9  |   .0051214   .0032399     1.58   0.115    -.0012529    .0114956
         10  |   .0025358   .0044185     0.57   0.566    -.0061572    .0112288
         11  |  -.0004647   .0037668    -0.12   0.902    -.0078757    .0069462
         12  |   .0024219    .003508     0.69   0.490    -.0044799    .0093237
             |
        date |  -5.37e-06   2.17e-06    -2.48   0.014    -9.64e-06   -1.11e-06
       _cons |   .0079011   .0030662     2.58   0.010     .0018685    .0139337
------------------------------------------------------------------------------
```

```
. *aic/bic
. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        383    1148.96   1616.918      64   -3105.837  -2853.163
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df1=el(r(S),1,4)

. scalar define aic1=el(r(S),1,5)

. scalar define bic1=el(r(S),1,6)

. *10 fold cv rmse
. crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf l(0/12)d.lnusepr l(0/12)d.lnflbp i.month date

             |      RMSE
-------------+-----------
        est1 |  .0041889
        est2 |  .0108909
        est3 |  .0039031
        est4 |  .0059736
        est5 |  .0061815

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse1=(el(kMSE,1,1)/k)^.5

. scalar list krmse1
    krmse1 =  .00474646

. matrix drop kMSE

. scalar drop k

. *loocv
. loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf l(0/12)d.lnusepr l(0/12)d.lnflbp i.month date


 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method           |    Value
--------------------------+--------------
Root Mean Squared Errors  |   .00746943
Mean Absolute Errors      |   .00370951
Pseudo-R2                 |   .62103068
-----------------------------------------

. scalar define loormse1=r(rmse)

.
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr l(0/2)d.lnflbp i.month date

      Source |       SS           df       MS      Number of obs   =       393
-------------+----------------------------------   F(33, 359)      =     89.36
       Model |  .050628703        33  .001534203   Prob > F        =    0.0000
    Residual |  .006163276       359  .000017168   R-squared       =    0.8915
-------------+----------------------------------   Adj R-squared   =    0.8815
       Total |  .056791979       392  .000144877   Root MSE        =    .00414

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.2879019   .0460794    -6.25   0.000    -.3785214   -.1972825
        L2D. |  -.1514457   .0463813    -3.27   0.001     -.242659   -.0602325
        L3D. |  -.0567902   .0294329    -1.93   0.054    -.1146727    .0010924
        L4D. |   .0805401    .029035     2.77   0.006     .0234401    .1376402
        L5D. |   .0148106   .0285969     0.52   0.605    -.0414279    .0710492
        L6D. |   .1156142   .0294928     3.92   0.000     .0576138    .1736146
        L7D. |   .0569047   .0295656     1.92   0.055     -.001239    .1150483
        L8D. |   .0333213   .0299318     1.11   0.266    -.0255424     .092185
        L9D. |   .0904818   .0496354     1.82   0.069    -.0071309    .1880945
       L10D. |  -.0920517   .0486864    -1.89   0.059    -.1877981    .0036947
       L11D. |   -.041793   .0502562    -0.83   0.406    -.1406266    .0570406
       L12D. |   .5067995   .0521856     9.71   0.000     .4041716    .6094273
             |
      lnfllf |
         D1. |  -.5263049    .089119    -5.91   0.000    -.7015658   -.3510439
         LD. |  -.2330839   .0937837    -2.49   0.013    -.4175183   -.0486495
        L2D. |  -.1589497   .0933952    -1.70   0.090    -.3426201    .0247208
             |
     lnusepr |
```

```
         D1.   1.060432   .0618714    17.14   0.000     .9387559    1.182108
         LD.   .3727363   .0796402     4.68   0.000     .2161164    .5293561
        L2D.     .18183   .0794706     2.29   0.023     .0255436    .3381164
             |
      lnflbp |
         D1.   .0039846   .0017113     2.33   0.020     .0006191      .00735
         LD.    .002868   .0019597     1.46   0.144     -.000986     .006722
        L2D.   .0007752   .0017454     0.44   0.657    -.0026574    .0042078
             |
       month |
          2   -.0067672   .0019876    -3.40   0.001    -.0106759   -.0028585
          3   -.0075077   .0021433    -3.50   0.001    -.0117227   -.0032928
          4   -.0145828   .0023379    -6.24   0.000    -.0191805   -.0099851
          5   -.0115104   .0023047    -4.99   0.000    -.0160428   -.0069781
          6   -.0201266   .0021525    -9.35   0.000    -.0243597   -.0158934
          7   -.0165043   .0020312    -8.13   0.000    -.0204989   -.0125097
          8   -.0060822   .0023503    -2.59   0.010    -.0107044     -.00146
          9   -.0003452   .0020707    -0.17   0.868    -.0044175    .0037271
         10   -.0031356   .0022726    -1.38   0.169     -.007605    .0013337
         11   -.0021778   .0022087    -0.99   0.325    -.0065215    .0021659
         12   -.0005171   .0019811    -0.26   0.794    -.0044131    .0033788
             |
        date  -3.85e-06   1.97e-06    -1.95   0.052    -7.73e-06    2.76e-08
       _cons   .0119425   .0019422     6.15   0.000      .008123     .015762
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        393   1179.844   1616.228      34   -3164.456  -3029.347
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df2=el(r(S),1,4)

. scalar define aic2=el(r(S),1,5)

. scalar define bic2=el(r(S),1,6)

. crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr l(0/2)d.lnflbp i.month date

             |      RMSE
-------------+-----------
        est1 |  .0083028
        est2 |  .0107218
        est3 |  .0037125
        est4 |  .0057848
        est5 |  .0054343

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse2=(el(kMSE,1,1)/k)^.5

. scalar list krmse2
    krmse2 =   .0051056

. matrix drop kMSE

. scalar drop k

. loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr l(0/2)d.lnflbp i.month date


 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   .00634841
Mean Absolute Errors     |   .00352834
Pseudo-R2                |   .7235841
-----------------------------------------

. scalar define loormse2=r(rmse)

.
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf l(0/2,12)d.lnflbp i.month date

      Source |       SS           df       MS      Number of obs   =       383
-------------+----------------------------------   F(32, 350)      =     47.62
       Model |  .045212283        32  .001412884   Prob > F        =    0.0000
    Residual |  .010384356       350  .00002967    R-squared       =    0.8132
-------------+----------------------------------   Adj R-squared   =    0.7961
       Total |  .055596638       382  .000145541   Root MSE        =    .00545

------------------------------------------------------------------------------
D.           |
```

```
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.1740883   .0474513    -3.67   0.000    -.2674139   -.0807628
        L2D. |  -.1796728   .0482708    -3.72   0.000    -.2746102   -.0847355
        L3D. |   .0482014    .037623     1.28   0.201    -.0257942     .122197
        L4D. |   .0632059   .0373781     1.69   0.092    -.0103081    .1367199
        L5D. |  -.0028745   .0368744    -0.08   0.938    -.0753979    .0696488
        L6D. |  -.0245775   .0368577    -0.67   0.505     -.097068    .0479131
        L7D. |   .0020704   .0365831     0.06   0.955    -.0698801    .0740208
        L8D. |  -.0138142   .0369521    -0.37   0.709    -.0864904     .058862
        L9D. |   .0905825   .0658306     1.38   0.170    -.0388908    .2200557
       L10D. |   .0013538   .0624801     0.02   0.983    -.1215299    .1242374
       L11D. |   .0933649   .0634182     1.47   0.142    -.0313638    .2180935
       L12D. |   .5476684   .0667124     8.21   0.000     .4164608     .678876
             |
      lnfllf |
         D1. |    .928238   .0488357    19.01   0.000     .8321896    1.024286
         LD. |   .2156929   .0625792     3.45   0.001     .0926143    .3387715
        L2D. |   -.031619   .0633368    -0.50   0.618    -.1561876    .0929497
       L12D. |  -.3699158   .0765149    -4.83   0.000    -.5204027    -.219429
             |
      lnflbp |
         D1. |   .0063989   .0022569     2.84   0.005     .0019601    .0108378
         LD. |   .0023471   .0025799     0.91   0.364    -.0027269    .0074211
        L2D. |   .0019136   .0023019     0.83   0.406    -.0026138    .0064409
       L12D. |  -.0005871   .0019573    -0.30   0.764    -.0044366    .0032625
             |
       month |
          2  |   .0038169   .0025894     1.47   0.141    -.0012758    .0089095
          3  |   .0021932    .002851     0.77   0.442    -.0034141    .0078005
          4  |   .0060381   .0027741     2.18   0.030      .000582    .0114941
          5  |   .0021218   .0028366     0.75   0.455    -.0034572    .0077007
          6  |  -.0044018   .0025573    -1.72   0.086    -.0094314    .0006277
          7  |   -.007066   .0025453    -2.78   0.006    -.0120719     -.00206
          8  |   .0048701   .0029279     1.66   0.097    -.0008884    .0106287
          9  |    .006203    .002778     2.23   0.026     .0007394    .0116666
         10  |   .0122795   .0027643     4.44   0.000     .0068429    .0177162
         11  |   .0128943   .0026646     4.84   0.000     .0076537    .0181349
         12  |   .0111956   .0024931     4.49   0.000     .0062924    .0160989
             |
        date |  -8.62e-08   2.63e-06    -0.03   0.974    -5.27e-06    5.09e-06
       _cons |  -.0044995   .0022885    -1.97   0.050    -.0090005    1.56e-06
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        383    1148.96   1470.263      33   -2874.526  -2744.241
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df3=el(r(S),1,4)

. scalar define aic3=el(r(S),1,5)

. scalar define bic3=el(r(S),1,6)

. crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf l(0/2,12)d.lnflbp i.month date

             |      RMSE 
-------------+-----------
        est1 |  .0049165 
        est2 |  .0059758 
        est3 |  .0057349 
        est4 |  .0069113 
        est5 |  .0156106 

. scalar define k=10

. matrix kMSE=r(est)'*r(est)

. scalar krmse3=(el(kMSE,1,1)/k)^.5

. scalar list krmse3
    krmse3 =   .00619861

. matrix drop kMSE

. scalar drop k

. loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf l(0/2,12)d.lnflbp i.month date


 Leave-One-Out Cross-Validation Results 
-----------------------------------------
         Method          |    Value
-------------------------+---------------
```

```
Root Mean Squared Errors |   .00808926
Mean Absolute Errors     |   .00433674
Pseudo-R2                |   .55858806
-----------------------------------------

. scalar define loormse3=r(rmse)

.
. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month

      Source |       SS           df       MS      Number of obs   =       515
-------------+----------------------------------   F(32, 482)      =     27.54
       Model |  .043561062        32  .001361283   Prob > F        =    0.0000
    Residual |  .023823343       482  .000049426   R-squared       =    0.6465
-------------+----------------------------------   Adj R-squared   =    0.6230
       Total |  .067384405       514  .000131098   Root MSE        =    .00703

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.0532095   .0602857    -0.88   0.378    -.1716647    .0652456
        L2D. |  -.0454893   .0610899    -0.74   0.457    -.1655247     .074546
        L3D. |   -.002843   .0435375    -0.07   0.948    -.0883896    .0827037
        L4D. |   .0464336   .0433763     1.07   0.285    -.0387964    .1316636
        L5D. |  -.0013076   .0431507    -0.03   0.976    -.0860943    .0834792
        L6D. |   .0200798   .0443166     0.45   0.651    -.0669978    .1071574
        L7D. |   .0224034   .0456573     0.49   0.624    -.0673086    .1121153
        L8D. |  -.0328449   .0459038    -0.72   0.475    -.1230412    .0573514
        L9D. |   .0396597    .069747     0.57   0.570    -.0973861    .1767054
       L10D. |  -.0055183   .0689159    -0.08   0.936     -.140931    .1298945
       L11D. |   .0773034   .0698507     1.11   0.269    -.0599461    .2145529
       L12D. |     .52293   .0889544     5.88   0.000     .3481438    .6977163
       L24D. |   .1727222   .0828627     2.08   0.038     .0099055    .3355389
             |
      lnfllf |
         LD. |   .0987675   .1177628     0.84   0.402    -.1326242    .3301593
        L2D. |  -.0771596   .1180294    -0.65   0.514    -.3090752    .1547561
       L12D. |    .037179   .1402792     0.27   0.791    -.2384553    .3128133
       L24D. |   .2524612     .13061     1.93   0.054    -.0041741    .5090966
             |
     lnusepr |
         LD. |    .011417   .1047318     0.11   0.913    -.1943703    .2172043
        L2D. |  -.0696112   .1055857    -0.66   0.510    -.2770762    .1378538
       L12D. |   .0186543   .1853382     0.10   0.920    -.3455162    .3828249
       L24D. |  -.4320365   .1733143    -2.49   0.013    -.7725814   -.0914916
             |
       month |
          2  |   .0101444   .0037078     2.74   0.006     .0028589      .01743
          3  |   .0082944   .0040339     2.06   0.040     .0003681    .0162207
          4  |   .0073699   .0043026     1.71   0.087    -.0010844    .0158242
          5  |   .0099262   .0038417     2.58   0.010     .0023776    .0174749
          6  |   .0069505   .0039997     1.74   0.083    -.0009085    .0148094
          7  |   .0039967   .0035296     1.13   0.258    -.0029385     .010932
          8  |   .0087545   .0035631     2.46   0.014     .0017533    .0157557
          9  |     .00794   .0031884     2.49   0.013     .0016752    .0142048
         10  |   .0129886   .0042968     3.02   0.003     .0045458    .0214313
         11  |   .0119932   .0037839     3.17   0.002     .0045582    .0194281
         12  |   .0116251   .0031936     3.64   0.000       .00535    .0179002
             |
       _cons |  -.0086323   .0031025    -2.78   0.006    -.0147284   -.0025362
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        515   1571.685    1839.42      33    -3612.84  -3472.783
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df4=el(r(S),1,4)

. scalar define aic4=el(r(S),1,5)

. scalar define bic4=el(r(S),1,6)

. crossfold reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month

             |       RMSE
-------------+-----------
        est1 |   .0041409
        est2 |   .0152337
        est3 |   .0066411
        est4 |   .0048066
        est5 |    .006126

. scalar define k=10
```

```
. matrix kMSE=r(est)'*r(est)

. scalar krmse4=(el(kMSE,1,1)/k)^.5

. scalar list krmse4
    krmse4 =  .00594936

. matrix drop kMSE

. scalar drop k

. loocv reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month


 Leave-One-Out Cross-Validation Results
 -----------------------------------------
         Method          |      Value
 -------------------------+---------------
 Root Mean Squared Errors |   .01062333
 Mean Absolute Errors     |   .00394285
 Pseudo-R2                |   .31761928
 -----------------------------------------

. scalar define loormse4=r(rmse)

.
. matrix drop _all

. matrix fit1=(df1,aic1,bic1,krmse1,loormse1)

. matrix fit2=(df2,aic2,bic2,krmse2,loormse2)

. matrix fit3=(df3,aic3,bic3,krmse3,loormse3)

. matrix fit4=(df4,aic4,bic4,krmse4,loormse4)

. matrix FIT=fit1\fit2\fit3\fit4

. matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 4"

. matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE

. matrix list FIT

FIT[4,5]
                df         AIC          BIC   K(10)RMSE     LOORMSE
Model 1         64  -3105.8368  -2853.1626   .00474646   .00746943
Model 2         34  -3164.4562  -3029.3467    .0051056   .00634841
Model 3         33  -2874.5257  -2744.2405   .00619861   .00808926
Model 4         33  -3612.8403  -3472.7828   .00594936   .01062333

.
. log close
      name:  <unnamed>
       log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 - Time Series/Problem Sets/Problem Set 3/Problem Set 3.smcl
  log type:  smcl
 closed on:  15 Mar 2021, 19:20:04
-------------------------------------------------------------------------------------------------------------------------------
```

Problem Set 4

Gus Lipkin

> All corrections are underlined

Problems

1. Drop any observations after December 2019.

```
drop if tin(2020m1,)
```

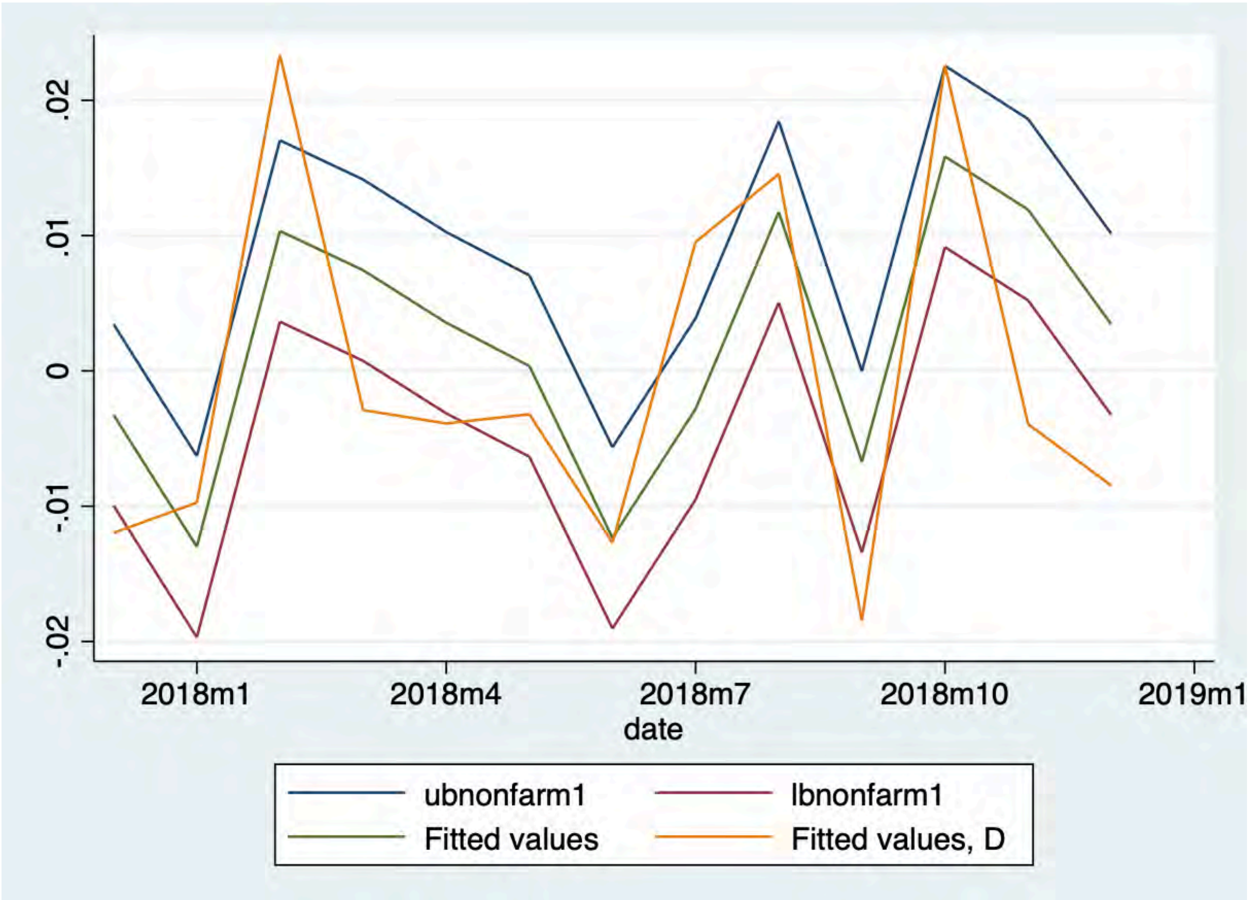
2. Refer to homework 3, question 2. Adapt the four models used there so they will be appropriate for making a one period ahead forecast.

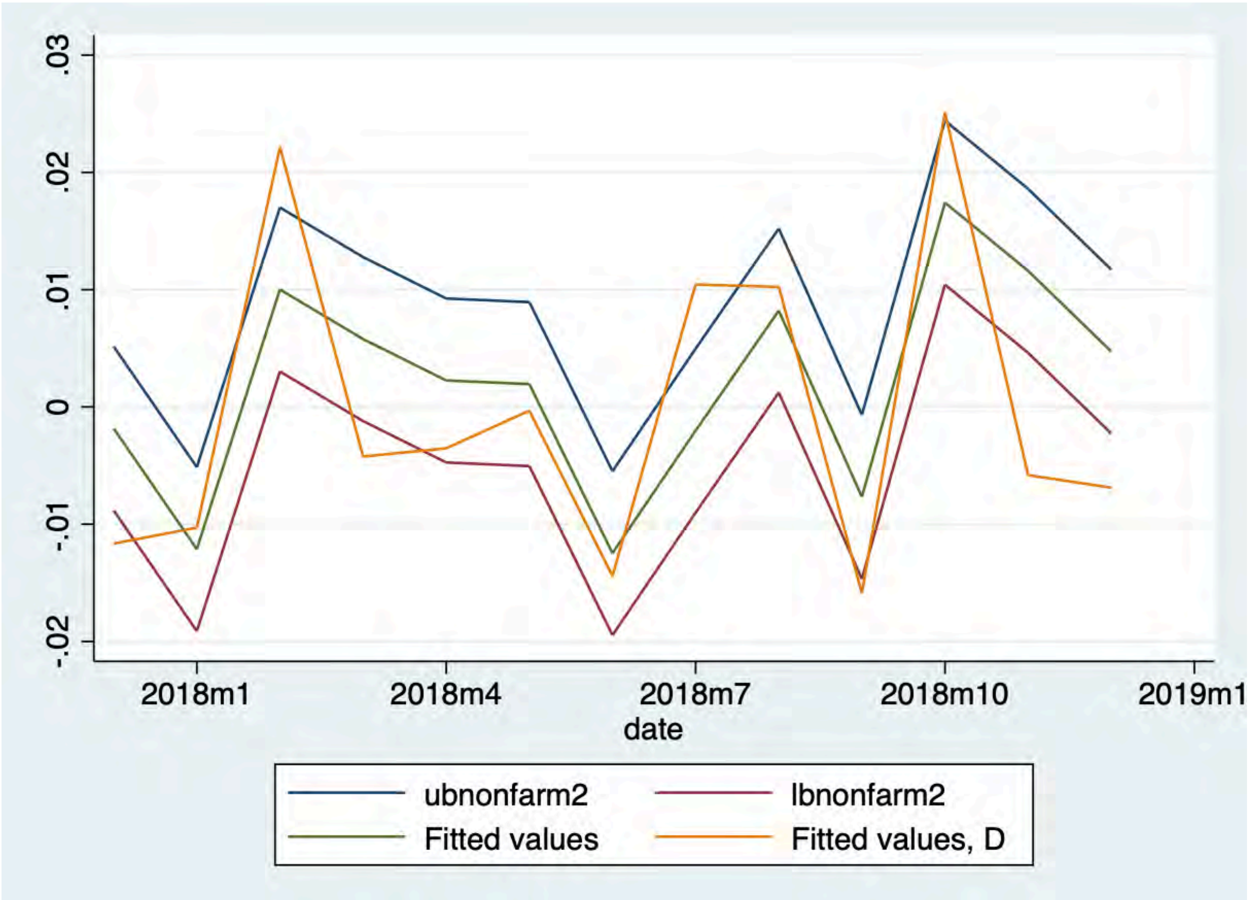
```
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month date
reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month
```

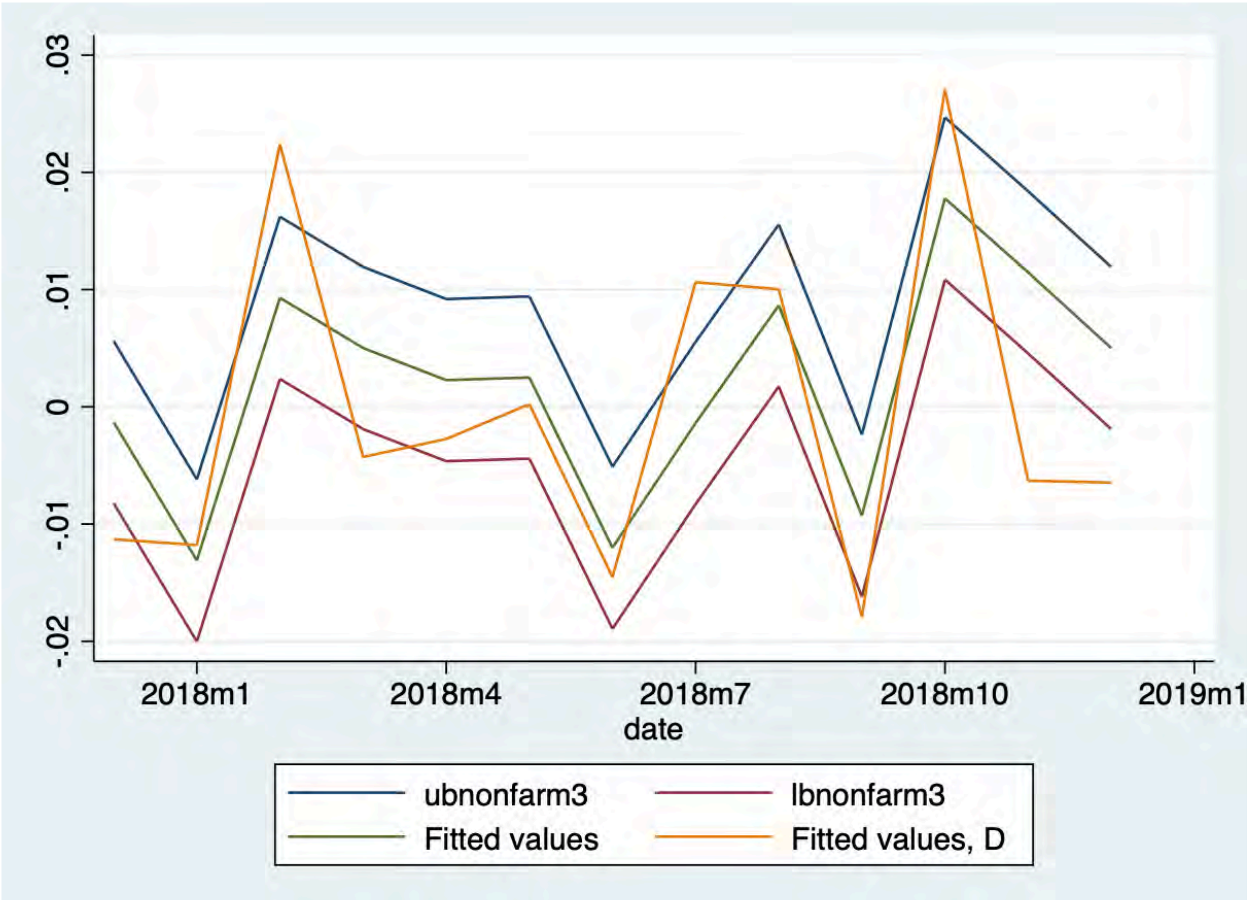
3. For each model, calculate the out of sample RMSE for the last year of observations (last 12 observations). To do this, you must not include these observations in the model estimation.

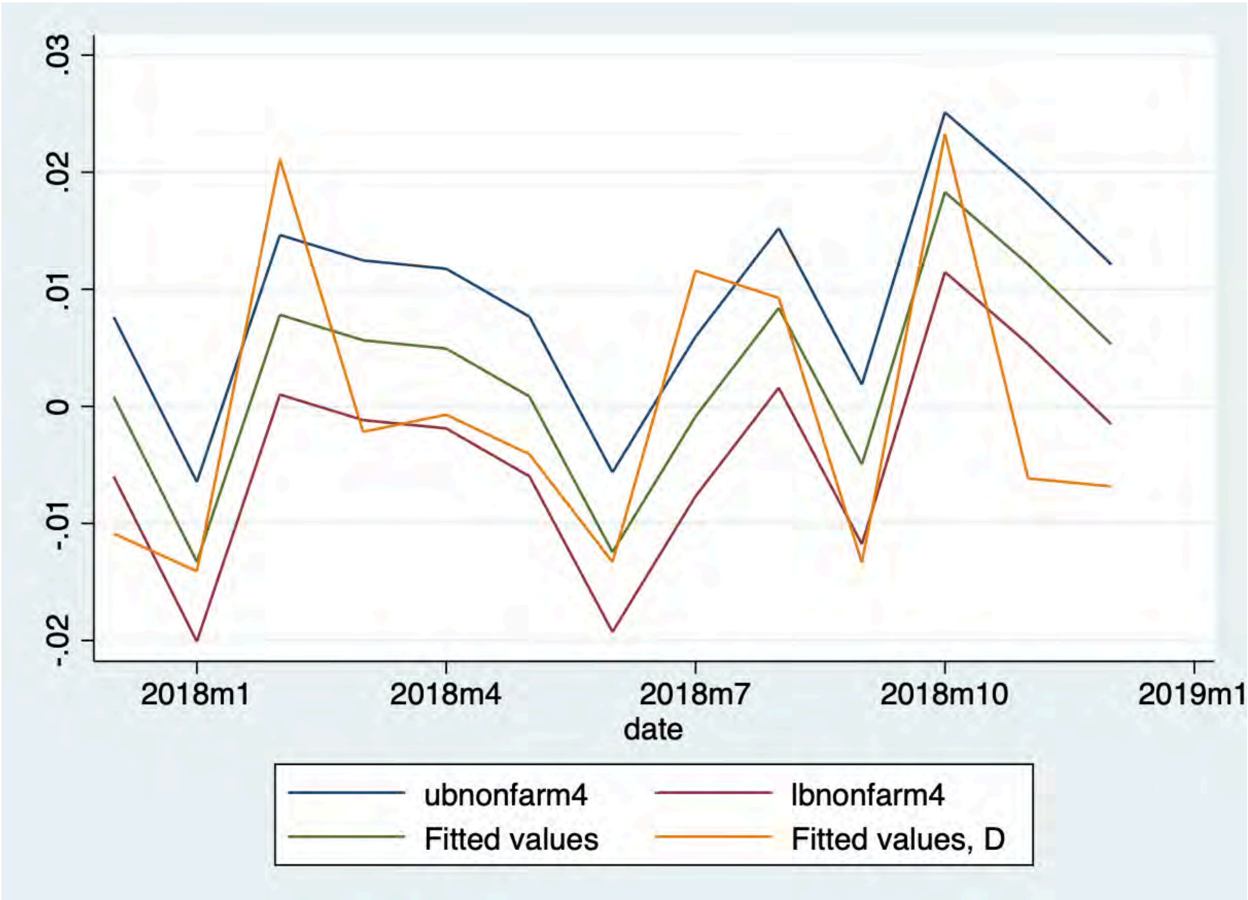
	c1	c2	c3	c4
r1	.00341856	.00356652	.00352901	.00347889

4. For each model, prepare a figure with the actual change in the log of nonfarm employment for the last 24 months, and for the last year the point forecast and the forecast interval, again using the model fit excluding the last 12 months.


.02
.01
0
-.01
-.02
2018m1
2018m4
2018m7
2018m10
2019m1
date
ubnonfarm1
lbnonfarm1
Fitted values
Fitted values, D


.03
.02
.01
0
-.01
-.02
2018m1
2018m4
2018m7
2018m10
2019m1
date
ubnonfarm2
lbnonfarm2
Fitted values
Fitted values, D


.03
.02
.01
0
-.01
-.02
2018m1
2018m4
2018m7
2018m10
2019m1
date
ubnonfarm3
lbnonfarm3
Fitted values
Fitted values, D

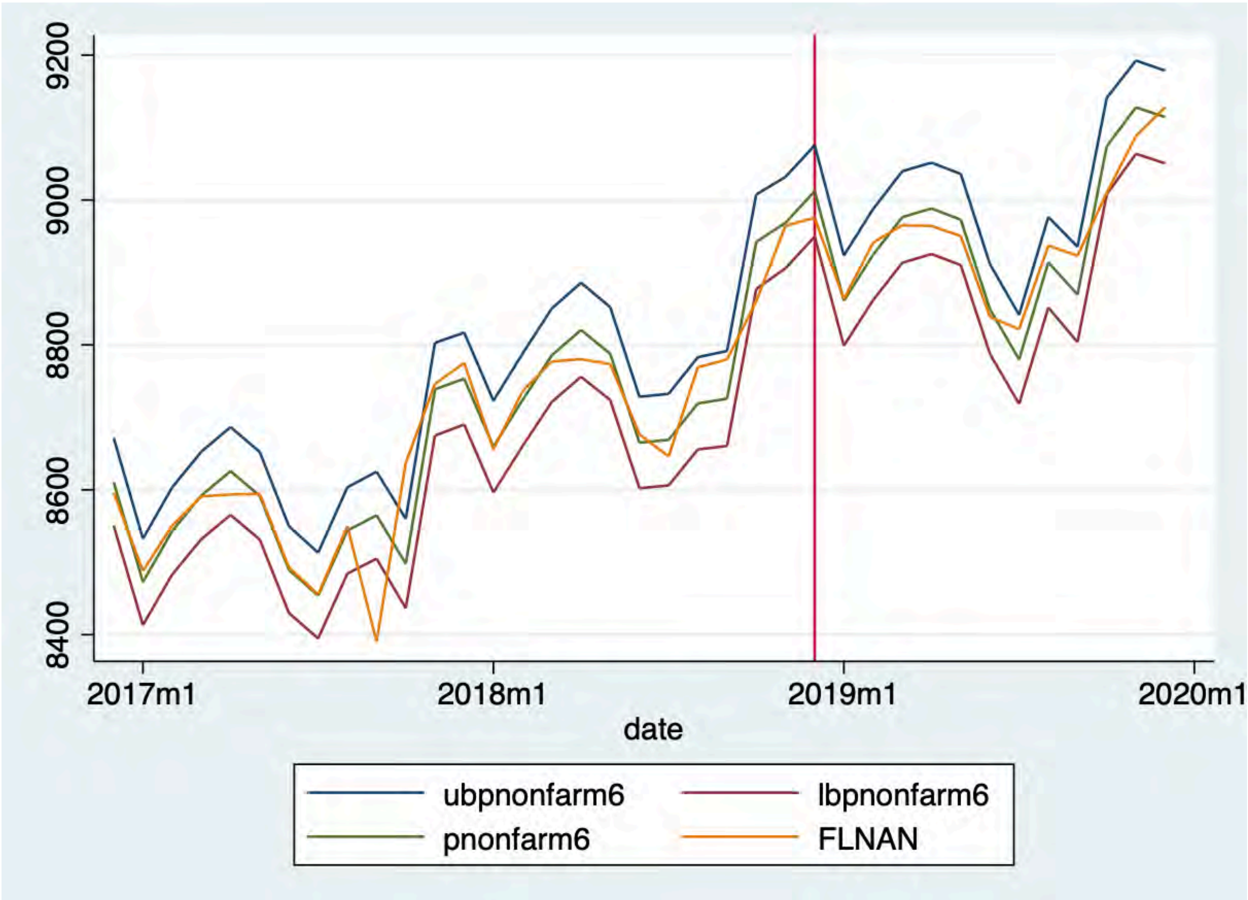


5. Select the best model for forecasting purposes based on AIC, BIC, LOOCV, and out of sample RMSE for the final year of data. Justify your choice.

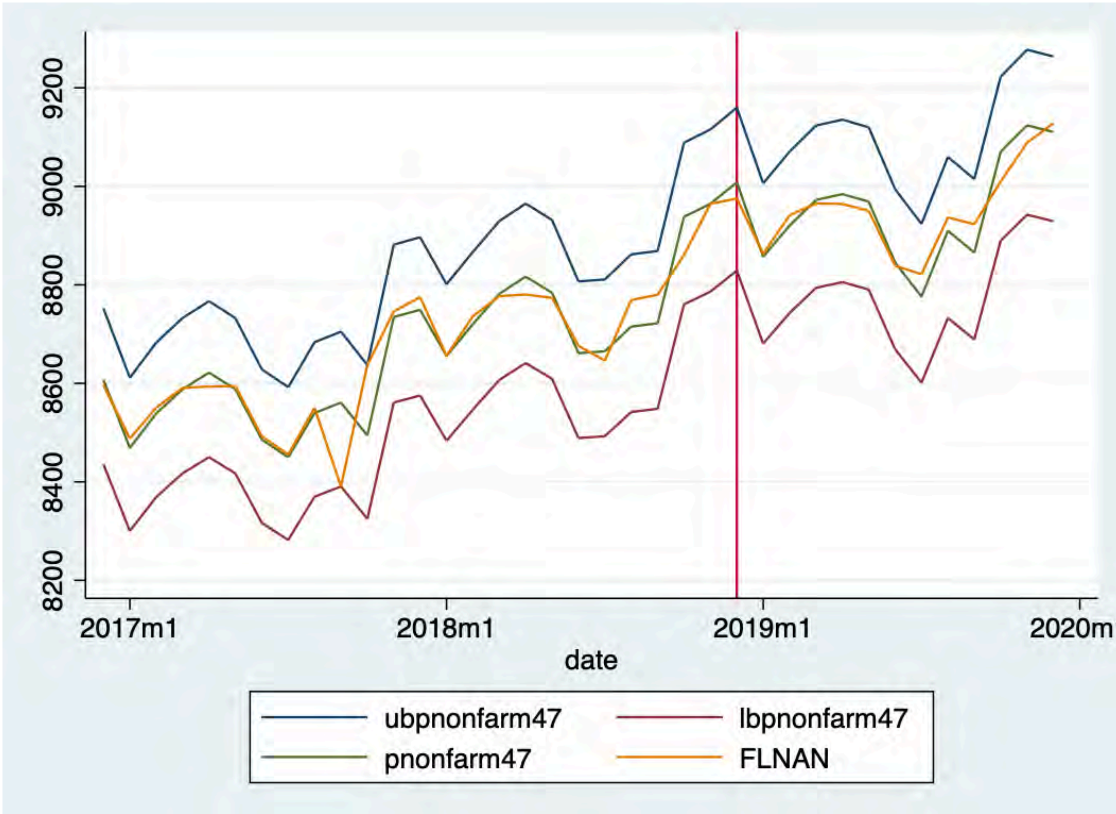
	df	AIC	BIC	RMSE	LOORMSE
Model 1	61	-3114.0527	-2875.1644	**.00341856**	.00380836
Model 2	31	-3190.9568	-3068.7301	.00356652	.00371849
Model 3	31	-3115.1451	-2993.7429	.00352901	.00375319
Model 4	33	**-4236.6004**	**-4097.3209**	.00347889	**.00355785**

I chose model four because the AIC, BIC, and LOORMSE were the lowest. The only difference was that the lowest RMSE was model 1. However, the difference in RMSE for model 1 and model 4 is very low so I'm comfortable choosing model 4 over model 3.

6. For the best model, transform the values appropriately and prepare a figure with the actual level of nonfarm employment (not the log) for the last 24 months, and for the last 12 months the point forecast and the forecast interval for nonfarm employment. For the interval forecast, assume approximate normality, and use the standard error of the forecast.



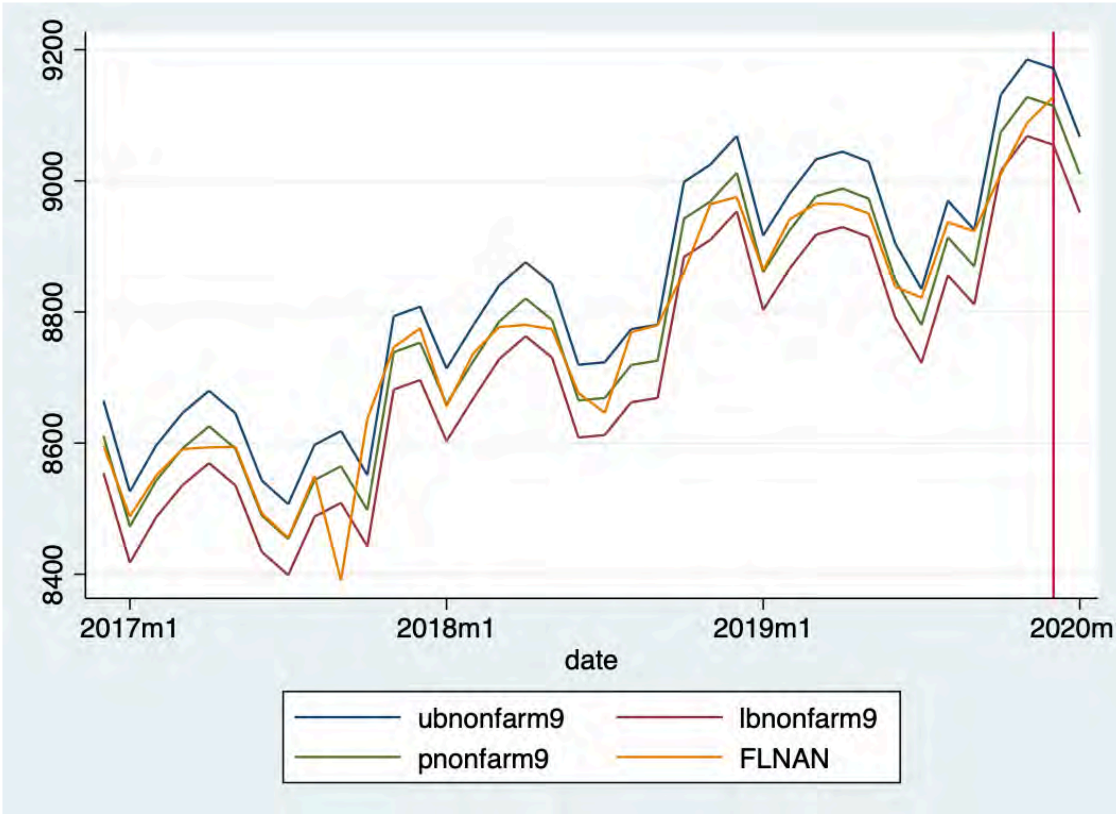
7. Now prepare another figure, again for the best model, with the actual level of nonfarm employment for the last 24 months, and for the last 12 months the point forecast and the forecast interval for nonfarm employment. This time, use the empirical approach, based on the data used to fit the model, to construct the forecast interval.



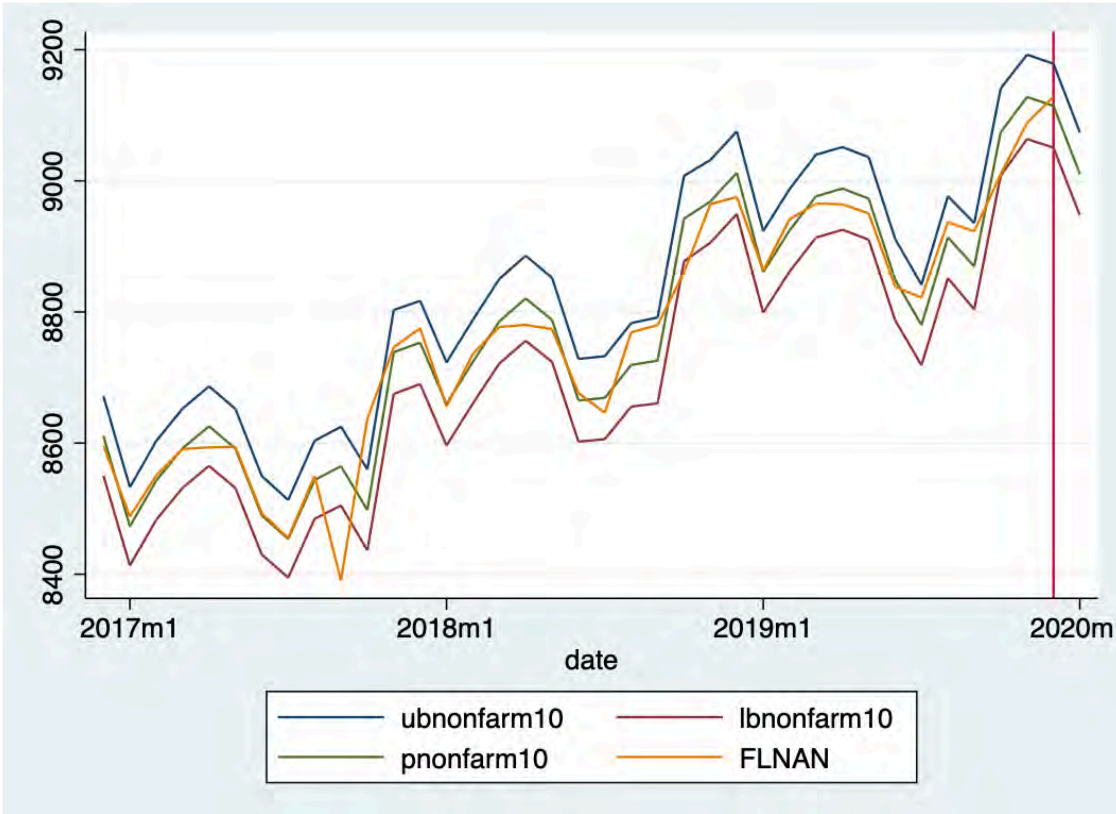
8. Run these commands to add January 2020 to the data (for which you will generate a forecast) and fill in the corresponding values for year and month:
   tsappend, add(1)
   replace month=month(dofm(date)) if month==.

```
tsappend, add(1)
replace month=month(dofm(date)) if month==.
```

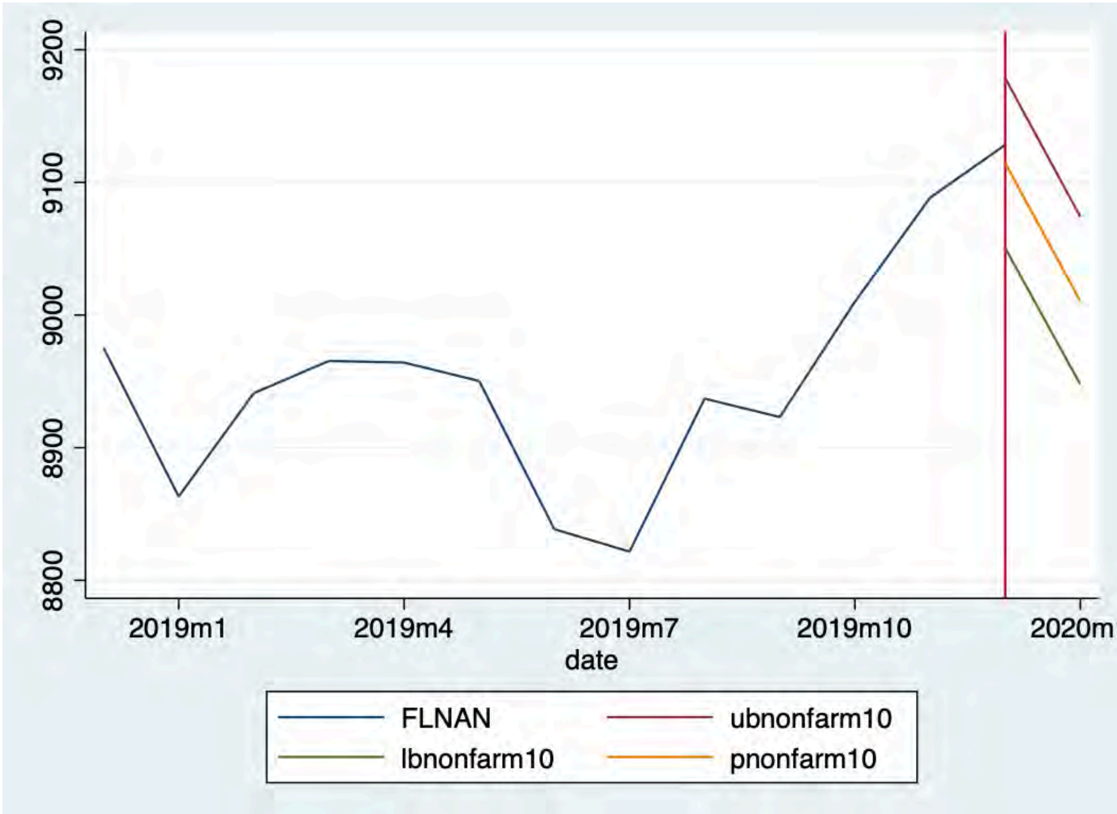
9. Run your selected model on the full sample and use it to forecast January 2020. Create point and interval forecasts for the change in the log of non-farm employment. Use the empirical approach.



10. Transform the point and interval forecast of the January 2020 change in the log of non-farm employment to create a point and interval forecast of non-farm employment for January 2020.



11. Generate a figure showing the last 12 months of non-farm employment and the January 2020 point and interval forecasts. (The figure shows actual for January 2019 through December 2019 and then the point and interval forecast for January 2020.)



The point forecast is 9093.688, and the empirical interval is 8923.5 to 9215.6.

Appendix A

```
clear
set more off

cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Problem Set 4"

*2a
*Done

*2b Load the data
import delimited "Assignment_1_Monthly.txt"

rename lnu02300000 us_epr
rename flnan fl_nonfarm
rename fllfn fl_lf
rename flbppriv fl_bp
rename date datestring

*2c Turn on a log file
log using "Problem Set 4", replace
```

```
*2d Generate a monthly date variable (make its display format monthly time, %tm)
gen datec=date(datestring, "YMD")
gen date=mofd(datec)
gen month=month(datec)
format date %tm

*2e tsset your data
tsset date

*2f
gen lnusepr=log(us_epr)
gen lnflnonfarm=log(fl_nonfarm)
gen lnfllf=log(fl_lf)
gen lnflbp=log(fl_bp)

*1
drop if tin(2020m1,)

*2
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month date
reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month

*3
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date if tin(,2018m12)
scalar define rmse1=e(rmse)
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date if tin(,2018m12)
scalar define rmse2=e(rmse)
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month date if tin(,2018m12)
scalar define rmse3=e(rmse)
reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
scalar define rmse4=e(rmse)

matrix drop _all
matrix row=(rmse1, rmse2, rmse3, rmse4)
matrix RMSE = row
matrix list RMSE
```

```
*4
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr
l(1/12)d.lnflbp i.month date if tin(,2018m12)
predict nonfarm1
gen ubnonfarm1=nonfarm1+1.96*e(rmse)
gen lbnonfarm1=nonfarm1-1.96*e(rmse)
tsline ubnonfarm1 lbnonfarm1 nonfarm1 d.nonfarm1 if tin(2017m12, 2018m12)
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr
l(1/2)d.lnflbp i.month date if tin(,2018m12)
predict nonfarm2
gen ubnonfarm2=nonfarm2+1.96*e(rmse)
gen lbnonfarm2=nonfarm2-1.96*e(rmse)
tsline ubnonfarm2 lbnonfarm2 nonfarm2 d.nonfarm2 if tin(2017m12, 2018m12)
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp
i.month date if tin(,2018m12)
predict nonfarm3
gen ubnonfarm3=nonfarm3+1.96*e(rmse)
gen lbnonfarm3=nonfarm3-1.96*e(rmse)
tsline ubnonfarm3 lbnonfarm3 nonfarm3 d.nonfarm3 if tin(2017m12, 2018m12)
reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
predict nonfarm4
gen ubnonfarm4=nonfarm4+1.96*e(rmse)
gen lbnonfarm4=nonfarm4-1.96*e(rmse)
tsline ubnonfarm4 lbnonfarm4 nonfarm4 d.nonfarm4 if tin(2017m12, 2018m12)

*5
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr
l(1/12)d.lnflbp i.month date
estat ic
scalar define df1=el(r(S),1,4)
scalar define aic1=el(r(S),1,5)
scalar define bic1=el(r(S),1,6)
loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr
l(1/12)d.lnflbp i.month date
scalar define loormse1=r(rmse)

reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr
l(1/2)d.lnflbp i.month date
estat ic
scalar define df2=el(r(S),1,4)
scalar define aic2=el(r(S),1,5)
scalar define bic2=el(r(S),1,6)
loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr
l(1/2)d.lnflbp i.month date
scalar define loormse2=r(rmse)
```

```
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp
i.month date
estat ic
scalar define df3=el(r(S),1,4)
scalar define aic3=el(r(S),1,5)
scalar define bic3=el(r(S),1,6)
loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp
i.month date
scalar define loormse3=r(rmse)

reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month
estat ic
scalar define df4=el(r(S),1,4)
scalar define aic4=el(r(S),1,5)
scalar define bic4=el(r(S),1,6)
loocv reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month
scalar define loormse4=r(rmse)

matrix drop _all
matrix fit1=(df1,aic1,bic1,rmse1,loormse1)
matrix fit2=(df2,aic2,bic2,rmse2,loormse2)
matrix fit3=(df3,aic3,bic3,rmse3,loormse3)
matrix fit4=(df4,aic4,bic4,rmse4,loormse4)
matrix FIT=fit1\fit2\fit3\fit4
matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 4"
matrix colnames FIT=df AIC BIC RMSE LOORMSE
matrix list FIT

*6
reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
predict nonfarm6
predict stdfore6, stdf
gen pnonfarm6=exp(l.lnflnonfarm+nonfarm6)*exp(.5*e(rmse)^2)
gen ubpnonfarm6=exp(l.lnflnonfarm+nonfarm6+1.96*stdfore6)*exp(.5*e(rmse)^2)
gen lbpnonfarm6=exp(l.lnflnonfarm+nonfarm6-1.96*stdfore6)*exp(.5*e(rmse)^2)
tsline ubpnonfarm6 lbpnonfarm6 pnonfarm6 fl_nonfarm if tin(2016m12,2019m12),
tline(2018m12)

*7
reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
predict nonfarm47
predict pres47 if tin(2016m12,2018m12), residual
gen exppres47=exp(pres47) if tin(2016m12,2018m12)
summ exppres47
```

```
gen pnonfarm47=r(mean)*exp(l.lnflnonfarm+nonfarm47)
_pctile exppres47, percentile(2.5,97.5)
gen lbpnonfarm47=r(r1)*exp(l.lnflnonfarm+nonfarm47)
gen ubpnonfarm47=r(r2)*exp(l.lnflnonfarm+nonfarm47)
tsline ubpnonfarm47 lbpnonfarm47 pnonfarm47 fl_nonfarm if tin(2016m12,2019m12),
tline(2018m12)

*8
tsappend, add(1)
replace month=month(dofm(date)) if month==.

*9
reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
predict nonfarm9
predict pres9 if tin(,2019m12), residual
gen exppres9=exp(pres9) if tin(,2019m12)
summ exppres9
gen pnonfarm9=r(mean)*exp(l.lnflnonfarm+nonfarm9)
_pctile exppres9, percentile(2.5,97.5)
gen lbnonfarm9=r(r1)*exp(l.lnflnonfarm+nonfarm9)
gen ubnonfarm9=r(r2)*exp(l.lnflnonfarm+nonfarm9)
tsline ubnonfarm9 lbnonfarm9  pnonfarm9 fl_nonfarm if tin(2016m12,2020m1),
tline(2019m12)

*10
reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
predict nonfarm10
predict stdfore10, stdf
gen pnonfarm10=exp(l.lnflnonfarm+nonfarm10)*exp(.5*e(rmse)^2)
gen ubnonfarm10=exp(l.lnflnonfarm+nonfarm10+1.96*stdfore10)*exp(.5*e(rmse)^2)
gen lbnonfarm10=exp(l.lnflnonfarm+nonfarm10-1.96*stdfore10)*exp(.5*e(rmse)^2)
tsline ubnonfarm10 lbnonfarm10 pnonfarm10 fl_nonfarm if tin(2016m12,2020m1),
tline(2019m12)

*11
tsline fl_nonfarm if tin(2018m12,2020m1) || tsline  ubnonfarm10 lbnonfarm10
pnonfarm10 if tin(2019m12,), tline(2019m12)

log close
```

Appendix B

```
      name:  <unnamed>
       log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 - Time Series/Problem Sets/Problem Set 4/Problem Set 4.smcl
  log type:  smcl
 opened on:  23 Mar 2021, 21:24:45

.
```

```
. *2d Generate a monthly date variable (make its display format monthly time, %tm)
. gen datec=date(datestring, "YMD")

. gen date=mofd(datec)

. gen month=month(datec)

. format date %tm

.
. *2e tsset your data
. tsset date
        time variable:  date, 1939m1 to 2020m12
                delta:  1 month

.
. *2f
. gen lnusepr=log(us_epr)
(108 missing values generated)

. gen lnflnonfarm=log(fl_nonfarm)

. gen lnfllf=log(fl_lf)
(444 missing values generated)

. gen lnflbp=log(fl_bp)
(588 missing values generated)

.
. *1
. drop if tin(2020m1,)
(12 observations deleted)

.
. *2
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date

      Source |       SS           df       MS      Number of obs   =       371
-------------+----------------------------------   F(60, 310)      =     49.11
       Model |  .033628663        60  .000560478   Prob > F        =    0.0000
    Residual |  .003538017       310  .000011413   R-squared       =    0.9048
-------------+----------------------------------   Adj R-squared   =    0.8864
       Total |   .03716668       370   .00010045   Root MSE        =    .00338

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.1709382   .0544484    -3.14   0.002    -.2780734    -.063803
        L2D. |  -.1482615   .0563272    -2.63   0.009    -.2590935   -.0374296
        L3D. |    .162612    .057373     2.83   0.005     .0497221    .2755018
        L4D. |   .1338157   .0578492     2.31   0.021     .0199889    .2476424
        L5D. |   .0485865   .0586353     0.83   0.408     -.066787    .1639599
        L6D. |   .1034826   .0583736     1.77   0.077    -.0113761    .2183412
        L7D. |  -.0017194   .0587605    -0.03   0.977    -.1173393    .1139005
        L8D. |  -.0661616   .0589687    -1.12   0.263    -.1821911     .049868
        L9D. |   .0703003   .0570451     1.23   0.219    -.0419442    .1825448
       L10D. |  -.2131464   .0557377    -3.82   0.000    -.3228184   -.1034745
       L11D. |  -.0492711   .0559229    -0.88   0.379    -.1593075    .0607653
       L12D. |   .3160254   .0547464     5.77   0.000     .2083039    .4237469
             |
      lnfllf |
         LD. |  -.1446453   .0983821    -1.47   0.143    -.3382263    .0489358
        L2D. |  -.1262282   .0987995    -1.28   0.202    -.3206305    .0681742
        L3D. |  -.0938871   .0998186    -0.94   0.348    -.2902948    .1025206
        L4D. |  -.0237969   .1021375    -0.23   0.816    -.2247673    .1771735
        L5D. |    .009078   .1014128     0.09   0.929    -.1904666    .2086226
        L6D. |  -.1064011   .1005346    -1.06   0.291    -.3042177    .0914155
        L7D. |  -.0363673   .1001261    -0.36   0.717    -.2333801    .1606454
        L8D. |  -.0227101   .0999056    -0.23   0.820     -.219289    .1738688
        L9D. |   .1419096   .0991963     1.43   0.154    -.0532736    .3370929
       L10D. |   .2356432   .0995816     2.37   0.019     .0397018    .4315847
       L11D. |  -.0156215   .1005373    -0.16   0.877    -.2134433    .1822003
       L12D. |  -.1496896   .0991977    -1.51   0.132    -.3448754    .0454963
             |
     lnusepr |
         LD. |   .2160403   .1318859     1.64   0.102    -.0434644    .4755451
        L2D. |   .0596176   .1342172     0.44   0.657    -.2044743    .3237096
        L3D. |   .1365192   .1332945     1.02   0.307    -.1257572    .3987956
        L4D. |    .140566   .1327682     1.06   0.291    -.1206748    .4018068
        L5D. |  -.0429764   .1313909    -0.33   0.744    -.3015071    .2155544
        L6D. |   .3174743   .1307471     2.43   0.016     .0602104    .5747382
        L7D. |   .1643272   .1319288     1.25   0.214    -.0952619    .4239164
        L8D. |  -.0363099   .1308906    -0.28   0.782    -.2938562    .2212364
        L9D. |  -.1104225   .1311797    -0.84   0.401    -.3685377    .1476927
       L10D. |  -.3143018   .1317767    -2.39   0.018    -.5735916    -.055012
       L11D. |   .1870841   .1321575     1.42   0.158     -.072955    .4471232
       L12D. |   .1595031    .127988     1.25   0.214     -.092332    .4113382
             |
      lnflbp |
         LD. |   .0014899   .0015781     0.94   0.346    -.0016152     .004595
        L2D. |   .0048244   .0018575     2.60   0.010     .0011694    .0084793
        L3D. |   .0065472   .0019601     3.34   0.001     .0026904     .010404
        L4D. |   .0049279    .001984     2.48   0.014     .0010242    .0088316
        L5D. |   .0050421   .0019821     2.54   0.011      .001142    .0089422
        L6D. |   .0049581   .0020017     2.48   0.014     .0010194    .0088969
        L7D. |   .0043549   .0020066     2.17   0.031     .0004067    .0083031
        L8D. |   .0036407   .0020487     1.78   0.077    -.0003905    .0076719
        L9D. |   .0033308   .0020403     1.63   0.104    -.0006839    .0073454
       L10D. |    .004421   .0020019     2.21   0.028      .000482      .00836
       L11D. |   .0031925   .0019124     1.67   0.096    -.0005706    .0069555
       L12D. |   .0030942   .0015709     1.97   0.050     3.18e-06    .0061852
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
month						
2	.0096471	.0035242	2.74	0.007	.0027127	.0165816
3	.0085261	.00368	2.32	0.021	.0012851	.0157671
4	.0092646	.0040143	2.31	0.022	.0013658	.0171635
5	.0035453	.0031449	1.13	0.260	-.0026427	.0097333
6	-.0026169	.0037248	-0.70	0.483	-.0099459	.0047121
7	.0042003	.0035921	1.17	0.243	-.0028678	.0112683
8	.0133789	.0037305	3.59	0.000	.0060387	.0207191
9	.0108298	.0031587	3.43	0.001	.0046146	.017045
10	.0165185	.0040273	4.10	0.000	.0085942	.0244428
11	.0090475	.0037037	2.44	0.015	.0017599	.0163351
12	.0154964	.0034704	4.47	0.000	.0086679	.022325
date	-3.66e-06	1.92e-06	-1.90	0.058	-7.43e-06	1.22e-07
_cons	-.0044454	.0028196	-1.58	0.116	-.0099933	.0011025

. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date

Source	SS	df	MS			
				Number of obs	=	381
				F(30, 350)	=	90.75
Model	.033990108	30	.001133004	Prob > F	=	0.0000
Residual	.004369869	350	.000012485	R-squared	=	0.8861
				Adj R-squared	=	0.8763
Total	.038359977	380	.000100947	Root MSE	=	.00353

D. lnflnonfarm	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
lnflnonfarm						
LD.	-.1252994	.0491434	-2.55	0.011	-.2219529	-.0286459
L2D.	-.0958602	.0518133	-1.85	0.065	-.1977648	.0060443
L3D.	.2398422	.0515342	4.65	0.000	.1384866	.3411978
L4D.	.1709118	.0489573	3.49	0.001	.0746243	.2671992
L5D.	.1515931	.0495961	3.06	0.002	.0540493	.249137
L6D.	.1371627	.0507474	2.70	0.007	.0373545	.2369708
L7D.	.0720218	.0527616	1.37	0.173	-.0317478	.1757914
L8D.	.0100807	.0513147	0.20	0.844	-.0908432	.1110047
L9D.	.0796876	.0488054	1.63	0.103	-.0163013	.1756764
L10D.	-.1886804	.047519	-3.97	0.000	-.2821391	-.0952217
L11D.	-.0844174	.0497603	-1.70	0.091	-.1822842	.0134493
L12D.	.388731	.0497483	7.81	0.000	.2908879	.4865741
lnfllf						
LD.	-.19426	.0941693	-2.06	0.040	-.3794688	-.0090513
L2D.	-.1402653	.0944467	-1.49	0.138	-.3260199	.0454893
lnusepr						
LD.	.2273847	.1224594	1.86	0.064	-.0134641	.4682334
L2D.	.0344122	.123157	0.28	0.780	-.2078086	.276633
lnflbp						
LD.	.0002646	.0014455	0.18	0.855	-.0025784	.0031076
L2D.	.0014358	.0014491	0.99	0.322	-.0014143	.0042858
month						
2	.0089123	.0022594	3.94	0.000	.0044686	.0133561
3	.0063439	.0027078	2.34	0.020	.0010183	.0116696
4	.0081387	.0027662	2.94	0.003	.0026983	.0135792
5	.002026	.0026956	0.75	0.453	-.0032757	.0073277
6	.0005971	.0024291	0.25	0.806	-.0041804	.0053746
7	.0012385	.0021082	0.59	0.557	-.0029078	.0053847
8	.011499	.0024647	4.67	0.000	.0066515	.0163465
9	.0150089	.002545	5.90	0.000	.0100036	.0200143
10	.0207091	.0025776	8.03	0.000	.0156395	.0257787
11	.0124833	.0024281	5.14	0.000	.0077077	.0172588
12	.0125059	.0019735	6.34	0.000	.0086245	.0163873
date	-9.91e-07	1.73e-06	-0.57	0.568	-4.40e-06	2.42e-06
_cons	-.0069309	.0019081	-3.63	0.000	-.0106838	-.0031781

. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month date

Source	SS	df	MS			
				Number of obs	=	371
				F(30, 340)	=	90.24
Model	.033019839	30	.001100661	Prob > F	=	0.0000
Residual	.004146841	340	.000012197	R-squared	=	0.8884
				Adj R-squared	=	0.8786
Total	.03716668	370	.00010045	Root MSE	=	.00349

D. lnflnonfarm	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
lnflnonfarm						
LD.	-.1007022	.0496694	-2.03	0.043	-.1984002	-.0030042
L2D.	-.0776112	.0500347	-1.55	0.122	-.1760278	.0208054
L3D.	.2495718	.0489276	5.10	0.000	.153333	.3458107
L4D.	.1659053	.0490998	3.38	0.001	.0693277	.2624828
L5D.	.1576085	.0494045	3.19	0.002	.0604315	.2547854
L6D.	.1606861	.0504142	3.19	0.002	.0615231	.2598491
L7D.	.0920205	.0512561	1.80	0.073	-.0087985	.1928394
L8D.	.0257082	.0504855	0.51	0.611	-.073595	.1250115
L9D.	.0816557	.0485929	1.68	0.094	-.013925	.1772363
L10D.	-.1982253	.0474658	-4.18	0.000	-.2915889	-.1048616
L11D.	-.1022931	.049168	-2.08	0.038	-.199005	-.0055813
L12D.	.3672239	.0495416	7.41	0.000	.2697773	.4646706
lnfllf						

```
         LD.  -.0415721   .0517745    -0.80   0.423    -.1434108    .0602666
        L2D.  -.1119071   .0515073    -2.17   0.030    -.2132203   -.0105939
       L12D.  -.1255582   .0492626    -2.55   0.011    -.2224562   -.0286603
              
      lnflbp  
         LD.   .0007479   .0014387     0.52   0.604    -.0020821    .0035778
        L2D.   .0019096   .0014405     1.33   0.186    -.0009239     .004743
       L12D.   .0016474   .0012703     1.30   0.196    -.0008512    .0041461
              
       month  
          2    .0086958   .0020531     4.24   0.000     .0046573    .0127342
          3    .0077648   .0024066     3.23   0.001     .0030311    .0124985
          4    .0089026   .0027229     3.27   0.001     .0035469    .0142584
          5    .0038752   .0026664     1.45   0.147    -.0013695    .0091199
          6    .0010548   .0023999     0.44   0.661    -.0036658    .0057754
          7    .0040191   .0020623     1.95   0.052    -.0000374    .0080755
          8    .0127434   .0023612     5.40   0.000      .008099    .0173878
          9    .0156048   .0025314     6.16   0.000     .0106256    .0205839
         10    .0204691   .0025295     8.09   0.000     .0154936    .0254446
         11    .0136586   .0022928     5.96   0.000     .0091487    .0181685
         12    .0133801    .001901     7.04   0.000     .0096408    .0171194
              
        date  -6.31e-07   1.73e-06    -0.37   0.715    -4.03e-06    2.76e-06
       _cons  -.0082167   .0018377    -4.47   0.000    -.0118314   -.0046019
```

```
. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month

      Source |       SS           df       MS      Number of obs   =       503
-------------+----------------------------------   F(32, 470)      =    111.72
       Model |  .043188967        32  .001349655   Prob > F        =    0.0000
    Residual |  .005677954       470  .000012081   R-squared       =    0.8838
-------------+----------------------------------   Adj R-squared   =    0.8759
       Total |  .048866921       502  .000097344   Root MSE        =    .00348

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.0255499     .03766    -0.68   0.498    -.0995527    .0484528
        L2D. |  -.0022421    .038345    -0.06   0.953    -.0775909    .0731068
        L3D. |   .1749722   .0380384     4.60   0.000     .1002259    .2497186
        L4D. |   .1177758   .0368875     3.19   0.002      .045291    .1902607
        L5D. |   .1127275   .0389003     2.90   0.004     .0362875    .1891675
        L6D. |    .090976   .0406874     2.24   0.026     .0110242    .1709278
        L7D. |   .0090996   .0405277     0.22   0.822    -.0705384    .0887376
        L8D. |  -.0366038   .0392974    -0.93   0.352    -.1138241    .0406165
        L9D. |   .0766282   .0372788     2.06   0.040     .0033744    .1498819
       L10D. |  -.1156119   .0372913    -3.10   0.002    -.1888902   -.0423337
       L11D. |   -.015338   .0383706    -0.40   0.690     -.090737    .0600611
       L12D. |   .3692339   .0461555     8.00   0.000     .2785372    .4599307
       L24D. |   .1970316   .0414188     4.76   0.000     .1156426    .2784205
             |
      lnfllf |
         LD. |  -.1283352    .068539    -1.87   0.062     -.263016    .0063457
        L2D. |  -.2120246   .0677924    -3.13   0.002    -.3452383   -.0788109
       L12D. |   -.032049   .0699833    -0.46   0.647    -.1695679      .10547
       L24D. |   .2125143   .0650669     3.27   0.001     .0846562    .3403724
             |
     lnusepr |
         LD. |   .1405225   .0907752     1.55   0.122    -.0378531     .318898
        L2D. |   .2347858   .0904615     2.60   0.010     .0570268    .4125448
       L12D. |  -.0236969   .0938261    -0.25   0.801    -.2080674    .1606736
       L24D. |  -.4061036   .0863262    -4.70   0.000    -.5757366   -.2364706
             |
       month |
          2  |   .0130388   .0021912     5.95   0.000     .0087331    .0173445
          3  |   .0143267   .0024097     5.95   0.000     .0095916    .0190618
          4  |   .0133539   .0024757     5.39   0.000      .008489    .0182188
          5  |   .0079333   .0022518     3.52   0.000     .0035084    .0123582
          6  |   .0089776   .0023234     3.86   0.000      .004412    .0135431
          7  |    .007944   .0019343     4.11   0.000     .0041431     .011745
          8  |   .0120699   .0019881     6.07   0.000     .0081632    .0159766
          9  |   .0139913   .0019658     7.12   0.000     .0101286    .0178541
         10  |   .0232609   .0024957     9.32   0.000     .0183568     .028165
         11  |   .0157873   .0022757     6.94   0.000     .0113154    .0202591
         12  |   .0128709   .0018317     7.03   0.000     .0092715    .0164702
             |
       _cons |    -.01163   .0016815    -6.92   0.000    -.0149341   -.0083258
------------------------------------------------------------------------------
```

```
.
. *3
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date if tin(,2018m12)

      Source |       SS           df       MS      Number of obs   =       359
-------------+----------------------------------   F(60, 298)      =     46.98
       Model |  .032942734        60  .000549046   Prob > F        =    0.0000
    Residual |  .003482599       298  .000011687   R-squared       =    0.9044
-------------+----------------------------------   Adj R-squared   =    0.8851
       Total |  .036425333       358  .000101747   Root MSE        =    .00342

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.1785286   .0557606    -3.20   0.002    -.2882631   -.0687941
        L2D. |  -.1456054   .0575841    -2.53   0.012    -.2589285   -.0322823
        L3D. |   .1670832   .0584604     2.86   0.005     .0520358    .2821307
        L4D. |    .135403   .0590676     2.29   0.023     .0191604    .2516456
```

L5D.	.0424067	.0600114	0.71	0.480	-.0756932	.1605066
L6D.	.0976656	.0600038	1.63	0.105	-.0204193	.2157504
L7D.	-.0007543	.0605067	-0.01	0.990	-.1198289	.1183204
L8D.	-.0676289	.0607766	-1.11	0.267	-.1872346	.0519769
L9D.	.0724574	.0588007	1.23	0.219	-.0432598	.1881745
L10D.	-.2140965	.0571111	-3.75	0.000	-.3264888	-.1017043
L11D.	-.0560776	.0572337	-0.98	0.328	-.168711	.0565558
L12D.	.3073789	.0560997	5.48	0.000	.1969771	.4177807
lnfllf						
LD.	-.14462	.1008735	-1.43	0.153	-.3431346	.0538946
L2D.	-.1332598	.1011166	-1.32	0.189	-.3322529	.0657332
L3D.	-.075505	.1025333	-0.74	0.462	-.277286	.126276
L4D.	.0025049	.1051978	0.02	0.981	-.2045197	.2095296
L5D.	.0257598	.1045744	0.25	0.806	-.1800381	.2315577
L6D.	-.1051399	.1040557	-1.01	0.313	-.309917	.0996372
L7D.	-.0521678	.1035685	-0.50	0.615	-.2559861	.1516505
L8D.	-.0353629	.1034667	-0.34	0.733	-.238981	.1682551
L9D.	.1257419	.1027424	1.22	0.222	-.0764507	.3279346
L10D.	.2264423	.102592	2.21	0.028	.0245458	.4283389
L11D.	-.0115823	.1034951	-0.11	0.911	-.2152562	.1920917
L12D.	-.1366552	.1019729	-1.34	0.181	-.3373335	.0640231
lnusepr						
LD.	.2199635	.1351476	1.63	0.105	-.0460011	.4859281
L2D.	.0824764	.1374525	0.60	0.549	-.1880241	.3529769
L3D.	.1238781	.1364468	0.91	0.365	-.1446433	.3923995
L4D.	.1231339	.135751	0.91	0.365	-.1440181	.390286
L5D.	-.0608881	.1344611	-0.45	0.651	-.3255017	.2037254
L6D.	.3128809	.1348904	2.32	0.021	.0474224	.5783393
L7D.	.1899949	.1365232	1.39	0.165	-.0786769	.4586667
L8D.	-.0190292	.1356228	-0.14	0.889	-.285929	.2478707
L9D.	-.0831357	.1358476	-0.61	0.541	-.3504779	.1842066
L10D.	-.3110896	.1355495	-2.30	0.022	-.5778451	-.0443342
L11D.	.186931	.1358319	1.38	0.170	-.0803804	.4542423
L12D.	.1507345	.1311892	1.15	0.251	-.1074402	.4089092
lnflbp						
LD.	.0018889	.0016344	1.16	0.249	-.0013275	.0051053
L2D.	.0052101	.0019412	2.68	0.008	.00139	.0090303
L3D.	.0065958	.0020468	3.22	0.001	.0025678	.0106239
L4D.	.0046902	.0020526	2.28	0.023	.0006508	.0087297
L5D.	.0048948	.0020293	2.41	0.016	.0009013	.0088882
L6D.	.0051713	.0020424	2.53	0.012	.0011519	.0091908
L7D.	.0045419	.0020457	2.22	0.027	.000516	.0085678
L8D.	.0034397	.002088	1.65	0.101	-.0006694	.0075488
L9D.	.0030855	.0020819	1.48	0.139	-.0010115	.0071825
L10D.	.0043761	.0020469	2.14	0.033	.0003479	.0084043
L11D.	.0033752	.0019615	1.72	0.086	-.000485	.0072353
L12D.	.0034007	.0016181	2.10	0.036	.0002164	.0065849
month						
2	.0099863	.0036814	2.71	0.007	.0027416	.0172311
3	.0093392	.0038138	2.45	0.015	.0018338	.0168446
4	.0098405	.0042053	2.34	0.020	.0015647	.0181163
5	.0035298	.0032546	1.08	0.279	-.0028751	.0099346
6	-.0029861	.0038815	-0.77	0.442	-.0106247	.0046525
7	.0039106	.0036724	1.06	0.288	-.0033164	.0111377
8	.0138735	.0038896	3.57	0.000	.0062188	.0215281
9	.0114154	.0032689	3.49	0.001	.0049823	.0178485
10	.017388	.004233	4.11	0.000	.0090577	.0257183
11	.0094232	.0038365	2.46	0.015	.0018732	.0169733
12	.0153423	.0036493	4.20	0.000	.0081607	.0225239
date	-3.47e-06	2.01e-06	-1.73	0.085	-7.42e-06	4.85e-07
_cons	-.0047857	.0029217	-1.64	0.102	-.0105354	.0009641

```
. scalar define rmse1=e(rmse)

. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date if tin(,2018m12)
```

Source	SS	df	MS
Model	.033319208	30	.00111064
Residual	.004299377	338	.00001272
Total	.037618585	368	.000102224

Number of obs	=	369
F(30, 338)	=	87.31
Prob > F	=	0.0000
R-squared	=	0.8857
Adj R-squared	=	0.8756
Root MSE	=	.00357

D. lnflnonfarm	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
lnflnonfarm						
LD.	-.133326	.0503841	-2.65	0.009	-.2324319	-.03422
L2D.	-.0952298	.0529421	-1.80	0.073	-.1993672	.0089076
L3D.	.2422019	.0525438	4.61	0.000	.1388477	.345556
L4D.	.1739116	.0500588	3.47	0.001	.0754456	.2723777
L5D.	.1496013	.0509462	2.94	0.004	.0493898	.2498129
L6D.	.1332644	.0521479	2.56	0.011	.0306891	.2358398
L7D.	.0730712	.0540717	1.35	0.177	-.0332882	.1794305
L8D.	.0161971	.0524574	0.31	0.758	-.086987	.1193812
L9D.	.0861744	.0499333	1.73	0.085	-.0120448	.1843937
L10D.	-.1843262	.048657	-3.79	0.000	-.2800348	-.0886175
L11D.	-.0933289	.0508022	-1.84	0.067	-.1932573	.0065995
L12D.	.3768875	.0508592	7.41	0.000	.276847	.476928
lnfllf						
LD.	-.1958536	.096245	-2.03	0.043	-.3851683	-.0065389
L2D.	-.1507978	.0964317	-1.56	0.119	-.3404796	.038884
lnusepr						

```
         LD. |    .231098   .1251978     1.85   0.066    -.0151669     .477363
        L2D. |   .0576833   .1260438     0.46   0.648    -.1902458    .3056124
             |
      lnflbp |
         LD. |   .0006178   .0014964     0.41   0.680    -.0023257    .0035613
        L2D. |   .0017963   .0014942     1.20   0.230    -.0011428    .0047354
             |
       month |
          2  |   .0092417   .0023322     3.96   0.000     .0046542    .0138292
          3  |   .0070536   .0028086     2.51   0.012     .0015291    .0125782
          4  |   .0084227   .0028514     2.95   0.003     .0028141    .0140314
          5  |   .0020872   .0027842     0.75   0.454    -.0033893    .0075638
          6  |   .0002948   .0025009     0.12   0.906    -.0046246    .0052141
          7  |   .0008253   .0021584     0.38   0.702    -.0034204    .0050709
          8  |    .011443   .0025191     4.54   0.000     .0064878    .0163981
          9  |   .0154884    .002613     5.93   0.000     .0103487    .0206282
         10  |   .0212612   .0026556     8.01   0.000     .0160376    .0264847
         11  |   .0130249   .0025092     5.19   0.000     .0080894    .0179605
         12  |   .0123947   .0020234     6.13   0.000     .0084147    .0163747
             |
        date |  -1.01e-06   1.84e-06    -0.55   0.581    -4.62e-06    2.60e-06
       _cons |  -.0070518   .0019605    -3.60   0.000     -.010908   -.0031955
------------------------------------------------------------------------------

. scalar define rmse2=e(rmse)

. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month date if tin(,2018m12)

      Source |       SS           df       MS      Number of obs   =       359
-------------+----------------------------------   F(30, 328)      =     86.56
       Model |  .032340453        30  .001078015   Prob > F        =    0.0000
    Residual |   .00408488       328  .000012454   R-squared       =    0.8879
-------------+----------------------------------   Adj R-squared   =    0.8776
       Total |  .036425333       358  .000101747   Root MSE        =    .00353

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.1087415    .050957    -2.13   0.034    -.2089852   -.0084978
        L2D. |  -.0783122   .0511855    -1.53   0.127    -.1790055    .0223811
        L3D. |   .2541334   .0499623     5.09   0.000     .1558465    .3524202
        L4D. |   .1689731   .0502832     3.36   0.001     .0700548    .2678914
        L5D. |   .1560811   .0507562     3.08   0.002     .0562324    .2559298
        L6D. |   .1559908   .0518295     3.01   0.003     .0540308    .2579509
        L7D. |    .094377   .0527031     1.79   0.074    -.0093018    .1980557
        L8D. |   .0330789   .0518367     0.64   0.524    -.0688954    .1350533
        L9D. |   .0893751   .0498407     1.79   0.074    -.0086727    .1874229
       L10D. |  -.1928458   .0486049    -3.97   0.000    -.2884625   -.0972291
       L11D. |  -.1106264   .0502767    -2.20   0.028    -.2095318   -.0117209
       L12D. |   .3558321    .050687     7.02   0.000     .2561195    .4555448
             |
      lnfllf |
         LD. |  -.0438007   .0531291    -0.82   0.410    -.1483174    .0607159
        L2D. |  -.1078659    .052828    -2.04   0.042    -.2117903   -.0039414
       L12D. |   -.121337   .0512128    -2.37   0.018    -.2220841    -.02059
             |
      lnflbp |
         LD. |   .0010857   .0014897     0.73   0.467     -.001845    .0040163
        L2D. |   .0022613   .0014885     1.52   0.130    -.0006669    .0051896
       L12D. |   .0017602   .0012989     1.36   0.176    -.0007949    .0043154
             |
       month |
          2  |   .0088945   .0021089     4.22   0.000     .0047459    .0130431
          3  |   .0081923   .0024954     3.28   0.001     .0032832    .0131013
          4  |   .0093013   .0028146     3.30   0.001     .0037643    .0148382
          5  |   .0039203   .0027567     1.42   0.156    -.0015028    .0093433
          6  |   .0008107   .0024749     0.33   0.743     -.004058    .0056794
          7  |    .003539   .0021131     1.67   0.095    -.0006179    .0076959
          8  |   .0127909   .0024143     5.30   0.000     .0080414    .0175403
          9  |   .0160119      .0026     6.16   0.000     .0108971    .0211268
         10  |   .0209367   .0026079     8.03   0.000     .0158063     .026067
         11  |   .0141242   .0023658     5.97   0.000     .0094702    .0187783
         12  |   .0133079   .0019471     6.83   0.000     .0094775    .0171383
             |
        date |  -6.06e-07   1.84e-06    -0.33   0.742    -4.22e-06    3.01e-06
       _cons |  -.0083676   .0018948    -4.42   0.000     -.012095   -.0046401
------------------------------------------------------------------------------

. scalar define rmse3=e(rmse)

. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)

      Source |       SS           df       MS      Number of obs   =       491
-------------+----------------------------------   F(32, 458)      =    109.93
       Model |  .042575775        32  .001330493   Prob > F        =    0.0000
    Residual |  .005543023       458  .000012103   R-squared       =    0.8848
-------------+----------------------------------   Adj R-squared   =    0.8768
       Total |  .048118799       490  .000098202   Root MSE        =    .00348

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.0311684   .0380987    -0.82   0.414    -.1060383    .0437014
        L2D. |   .0011509   .0387186     0.03   0.976    -.0749371     .077239
        L3D. |   .1811019   .0383354     4.72   0.000     .1057668    .2564371
        L4D. |    .122168   .0372877     3.28   0.001     .0488919     .195444
        L5D. |   .1131335   .0394764     2.87   0.004     .0355562    .1907107
        L6D. |   .0897111   .0411525     2.18   0.030     .0088401    .1705822
```

```
        L7D.  |   .0046318   .0409493     0.11   0.910    -.0758401    .0851037
        L8D.  |  -.0345244   .0396846    -0.87   0.385    -.1125109     .043462
        L9D.  |   .0809663   .0376451     2.15   0.032     .0069878    .1549448
       L10D.  |  -.1108451   .0375797    -2.95   0.003    -.1846951   -.0369951
       L11D.  |  -.0221662     .03865    -0.57   0.567    -.0981196    .0537872
       L12D.  |   .3376832   .0476368     7.09   0.000     .2440695    .4312969
       L24D.  |   .2364096     .04478     5.28   0.000       .14841    .3244093
              |
      lnfllf  |
         LD.  |  -.1300766   .0692757    -1.88   0.061    -.2662142    .0060609
        L2D.  |  -.2188669   .0684865    -3.20   0.001    -.3534536   -.0842802
       L12D.  |  -.0319706    .070685    -0.45   0.651    -.1708776    .1069365
       L24D.  |   .2248965     .06562     3.43   0.001     .0959428    .3538501
              |
     lnusepr  |
         LD.  |   .1537422    .092086     1.67   0.096    -.0272212    .3347056
        L2D.  |   .2542549   .0916164     2.78   0.006     .0742143    .4342955
       L12D.  |  -.0218538   .0949163    -0.23   0.818    -.2083793    .1646718
       L24D.  |  -.4432243   .0880674    -5.03   0.000    -.6162905   -.2701581
              |
       month  |
          2   |   .0133765    .002229     6.00   0.000     .0089962    .0177569
          3   |   .0150211   .0024611     6.10   0.000     .0101847    .0198576
          4   |   .0140381   .0025223     5.57   0.000     .0090814    .0189949
          5   |   .0082786   .0023045     3.59   0.000     .0037499    .0128072
          6   |   .0092759   .0023797     3.90   0.000     .0045993    .0139524
          7   |   .0081629   .0019705     4.14   0.000     .0042905    .0120353
          8   |   .0119157   .0020208     5.90   0.000     .0079446    .0158868
          9   |   .0141217   .0020005     7.06   0.000     .0101904     .018053
         10   |   .0242635   .0025397     9.55   0.000     .0192726    .0292543
         11   |   .0162918   .0023257     7.01   0.000     .0117215    .0208621
         12   |    .012759   .0018654     6.84   0.000     .0090932    .0164248
              |
       _cons  |  -.0119962   .0017171    -6.99   0.000    -.0153705   -.0086219
------------------------------------------------------------------------------

. scalar define rmse4=e(rmse)

.
. matrix drop _all

. matrix row=(rmse1, rmse2, rmse3, rmse4)

. matrix RMSE = row

. matrix list RMSE

RMSE[1,4]
           c1         c2         c3         c4
r1  .00341856  .00356652  .00352901  .00347889

.
. *4
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date if tin(,2018m12)

      Source |       SS           df       MS      Number of obs   =       359
-------------+----------------------------------   F(60, 298)      =     46.98
       Model |  .032942734        60  .000549046   Prob > F        =    0.0000
    Residual |  .003482599       298  .000011687   R-squared       =    0.9044
-------------+----------------------------------   Adj R-squared   =    0.8851
       Total |  .036425333       358  .000101747   Root MSE        =    .00342

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.1785286   .0557606    -3.20   0.002    -.2882631   -.0687941
        L2D. |  -.1456054   .0575841    -2.53   0.012    -.2589285   -.0322823
        L3D. |   .1670832   .0584604     2.86   0.005     .0520358    .2821307
        L4D. |    .135403   .0590676     2.29   0.023     .0191604    .2516456
        L5D. |   .0424067   .0600114     0.71   0.480    -.0756932    .1605066
        L6D. |   .0976656   .0600038     1.63   0.105    -.0204193    .2157504
        L7D. |  -.0007543   .0605067    -0.01   0.990    -.1198289    .1183204
        L8D. |  -.0676289   .0607766    -1.11   0.267    -.1872346    .0519769
        L9D. |   .0724574   .0588007     1.23   0.219    -.0432598    .1881745
       L10D. |  -.2140965   .0571111    -3.75   0.000    -.3264888   -.1017043
       L11D. |  -.0560776   .0572337    -0.98   0.328     -.168711    .0565558
       L12D. |   .3073789   .0560997     5.48   0.000     .1969771    .4177807
             |
      lnfllf |
         LD. |    -.14462   .1008735    -1.43   0.153    -.3431346    .0538946
        L2D. |  -.1332598   .1011166    -1.32   0.189    -.3322529    .0657332
        L3D. |   -.075505   .1025333    -0.74   0.462     -.277286     .126276
        L4D. |   .0025049   .1051978     0.02   0.981    -.2045197    .2095296
        L5D. |   .0257598   .1045744     0.25   0.806    -.1800381    .2315577
        L6D. |  -.1051399   .1040557    -1.01   0.313     -.309917    .0996372
        L7D. |  -.0521678   .1035685    -0.50   0.615    -.2559861    .1516505
        L8D. |  -.0353629   .1034667    -0.34   0.733     -.238981    .1682551
        L9D. |   .1257419   .1027424     1.22   0.222    -.0764507    .3279346
       L10D. |   .2264423    .102592     2.21   0.028     .0245458    .4283389
       L11D. |  -.0115823   .1034951    -0.11   0.911    -.2152562    .1920917
       L12D. |  -.1366552   .1019729    -1.34   0.181    -.3373335    .0640231
             |
     lnusepr |
         LD. |   .2199635   .1351476     1.63   0.105    -.0460011    .4859281
        L2D. |   .0824764   .1374525     0.60   0.549    -.1880241    .3529769
        L3D. |   .1238781   .1364468     0.91   0.365    -.1446433    .3923995
        L4D. |   .1231339    .135751     0.91   0.365    -.1440181     .390286
        L5D. |  -.0608881   .1344611    -0.45   0.651    -.3255017    .2037254
        L6D. |   .3128809   .1348904     2.32   0.021     .0474224    .5783393
        L7D. |   .1899949   .1365232     1.39   0.165    -.0786769    .4586667
```

```
        L8D.  |  -.0190292   .1356228    -0.14   0.889     -.285929    .2478707
        L9D.  |  -.0831357   .1358476    -0.61   0.541    -.3504779    .1842066
       L10D.  |  -.3110896   .1355495    -2.30   0.022    -.5778451   -.0443342
       L11D.  |    .186931   .1358319     1.38   0.170    -.0803804    .4542423
       L12D.  |   .1507345   .1311892     1.15   0.251    -.1074402    .4089092
              |
     lnflbp   |
         LD.  |   .0018889   .0016344     1.16   0.249    -.0013275    .0051053
        L2D.  |   .0052101   .0019412     2.68   0.008       .00139    .0090303
        L3D.  |   .0065958   .0020468     3.22   0.001     .0025678    .0106239
        L4D.  |   .0046902   .0020526     2.28   0.023     .0006508    .0087297
        L5D.  |   .0048948   .0020293     2.41   0.016     .0009013    .0088882
        L6D.  |   .0051713   .0020424     2.53   0.012     .0011519    .0091908
        L7D.  |   .0045419   .0020457     2.22   0.027      .000516    .0085678
        L8D.  |   .0034397    .002088     1.65   0.101    -.0006694    .0075488
        L9D.  |   .0030855   .0020819     1.48   0.139    -.0010115    .0071825
       L10D.  |   .0043761   .0020469     2.14   0.033     .0003479    .0084043
       L11D.  |   .0033752   .0019615     1.72   0.086     -.000485    .0072353
       L12D.  |   .0034007   .0016181     2.10   0.036     .0002164    .0065849
              |
      month   |
          2   |   .0099863   .0036814     2.71   0.007     .0027416    .0172311
          3   |   .0093392   .0038138     2.45   0.015     .0018338    .0168446
          4   |   .0098405   .0042053     2.34   0.020     .0015647    .0181163
          5   |   .0035298   .0032546     1.08   0.279    -.0028751    .0099346
          6   |  -.0029861   .0038815    -0.77   0.442    -.0106247    .0046525
          7   |   .0039106   .0036724     1.06   0.288    -.0033164    .0111377
          8   |   .0138735   .0038896     3.57   0.000     .0062188    .0215281
          9   |   .0114154   .0032689     3.49   0.001     .0049823    .0178485
         10   |    .017388    .004233     4.11   0.000     .0090577    .0257183
         11   |   .0094232   .0038365     2.46   0.015     .0018732    .0169733
         12   |   .0153423   .0036493     4.20   0.000     .0081607    .0225239
              |
       date   |  -3.47e-06   2.01e-06    -1.73   0.085    -7.42e-06    4.85e-07
      _cons   |  -.0047857   .0029217    -1.64   0.102    -.0105354    .0009641
------------------------------------------------------------------------------

. predict nonfarm1
(option xb assumed; fitted values)
(601 missing values generated)

. gen ubnonfarm1=nonfarm1+1.96*e(rmse)
(601 missing values generated)

. gen lbnonfarm1=nonfarm1-1.96*e(rmse)
(601 missing values generated)

. tsline ubnonfarm1 lbnonfarm1 nonfarm1 d.nonfarm1 if tin(2017m12, 2018m12)

. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date if tin(,2018m12)

      Source  |       SS           df       MS      Number of obs   =       369
--------------+----------------------------------   F(30, 338)      =     87.31
       Model  |  .033319208        30  .00111064    Prob > F        =    0.0000
    Residual  |  .004299377       338  .00001272    R-squared       =    0.8857
--------------+----------------------------------   Adj R-squared   =    0.8756
       Total  |  .037618585       368  .000102224   Root MSE        =    .00357

------------------------------------------------------------------------------
D.            |
 lnflnonfarm  |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
--------------+----------------------------------------------------------------
 lnflnonfarm  |
         LD.  |   -.133326   .0503841    -2.65   0.009    -.2324319    -.03422
        L2D.  |  -.0952298   .0529421    -1.80   0.073    -.1993672    .0089076
        L3D.  |   .2422019   .0525438     4.61   0.000     .1388477     .345556
        L4D.  |   .1739116   .0500588     3.47   0.001     .0754456    .2723777
        L5D.  |   .1496013   .0509462     2.94   0.004     .0493898    .2498129
        L6D.  |   .1332644   .0521479     2.56   0.011     .0306891    .2358398
        L7D.  |   .0730712   .0540717     1.35   0.177    -.0332882    .1794305
        L8D.  |   .0161971   .0524574     0.31   0.758     -.086987    .1193812
        L9D.  |   .0861744   .0499333     1.73   0.085    -.0120448    .1843937
       L10D.  |  -.1843262    .048657    -3.79   0.000    -.2800348   -.0886175
       L11D.  |  -.0933289   .0508022    -1.84   0.067    -.1932573    .0065995
       L12D.  |   .3768875   .0508592     7.41   0.000      .276847     .476928
              |
      lnfllf  |
         LD.  |  -.1958536    .096245    -2.03   0.043    -.3851683   -.0065389
        L2D.  |  -.1507978   .0964317    -1.56   0.119    -.3404796     .038884
              |
     lnusepr  |
         LD.  |    .231098   .1251978     1.85   0.066    -.0151669     .477363
        L2D.  |   .0576833   .1260438     0.46   0.648    -.1902458    .3056124
              |
      lnflbp  |
         LD.  |   .0006178   .0014964     0.41   0.680    -.0023257    .0035613
        L2D.  |   .0017963   .0014942     1.20   0.230    -.0011428    .0047354
              |
       month  |
          2   |   .0092417   .0023322     3.96   0.000     .0046542    .0138292
          3   |   .0070536   .0028086     2.51   0.012     .0015291    .0125782
          4   |   .0084227   .0028514     2.95   0.003     .0028141    .0140314
          5   |   .0020872   .0027842     0.75   0.454    -.0033893    .0075638
          6   |   .0002948   .0025009     0.12   0.906    -.0046246    .0052141
          7   |   .0008253   .0021584     0.38   0.702    -.0034204    .0050709
          8   |    .011443   .0025191     4.54   0.000     .0064878    .0163981
          9   |   .0154884    .002613     5.93   0.000     .0103487    .0206282
         10   |   .0212612   .0026556     8.01   0.000     .0160376    .0264847
         11   |   .0130249   .0025092     5.19   0.000     .0080894    .0179605
         12   |   .0123947   .0020234     6.13   0.000     .0084147    .0163747
              |
        date  |  -1.01e-06   1.84e-06    -0.55   0.581    -4.62e-06    2.60e-06
```

```
       _cons |  -.0070518   .0019605    -3.60   0.000     -.010908   -.0031955
------------------------------------------------------------------------------

. predict nonfarm2
(option xb assumed; fitted values)
(591 missing values generated)

. gen ubnonfarm2=nonfarm2+1.96*e(rmse)
(591 missing values generated)

. gen lbnonfarm2=nonfarm2-1.96*e(rmse)
(591 missing values generated)

. tsline ubnonfarm2 lbnonfarm2 nonfarm2 d.nonfarm2 if tin(2017m12, 2018m12)

. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month date if tin(,2018m12)

      Source |       SS           df       MS      Number of obs   =       359
-------------+----------------------------------   F(30, 328)      =     86.56
       Model |  .032340453        30  .001078015   Prob > F        =    0.0000
    Residual |   .00408488       328  .000012454   R-squared       =    0.8879
-------------+----------------------------------   Adj R-squared   =    0.8776
       Total |  .036425333       358  .000101747   Root MSE        =    .00353

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.1087415    .050957    -2.13   0.034    -.2089852   -.0084978
        L2D. |  -.0783122   .0511855    -1.53   0.127    -.1790055    .0223811
        L3D. |   .2541334   .0499623     5.09   0.000     .1558465    .3524202
        L4D. |   .1689731   .0502832     3.36   0.001     .0700548    .2678914
        L5D. |   .1560811   .0507562     3.08   0.002     .0562324    .2559298
        L6D. |   .1559908   .0518295     3.01   0.003     .0540308    .2579509
        L7D. |    .094377   .0527031     1.79   0.074    -.0093018    .1980557
        L8D. |   .0330789   .0518367     0.64   0.524    -.0688954    .1350533
        L9D. |   .0893751   .0498407     1.79   0.074    -.0086727    .1874229
       L10D. |  -.1928458   .0486049    -3.97   0.000    -.2884625   -.0972291
       L11D. |  -.1106264   .0502767    -2.20   0.028    -.2095318   -.0117209
       L12D. |   .3558321    .050687     7.02   0.000     .2561195    .4555448
             |
      lnfllf |
         LD. |  -.0438007   .0531291    -0.82   0.410    -.1483174    .0607159
        L2D. |  -.1078659    .052828    -2.04   0.042    -.2117903   -.0039414
       L12D. |   -.121337   .0512128    -2.37   0.018    -.2220841    -.02059
             |
      lnflbp |
         LD. |   .0010857   .0014897     0.73   0.467     -.001845    .0040163
        L2D. |   .0022613   .0014885     1.52   0.130    -.0006669    .0051896
       L12D. |   .0017602   .0012989     1.36   0.176    -.0007949    .0043154
             |
       month |
          2  |   .0088945   .0021089     4.22   0.000     .0047459    .0130431
          3  |   .0081923   .0024954     3.28   0.001     .0032832    .0131013
          4  |   .0093013   .0028146     3.30   0.001     .0037643    .0148382
          5  |   .0039203   .0027567     1.42   0.156    -.0015028    .0093433
          6  |   .0008107   .0024749     0.33   0.743     -.004058    .0056794
          7  |    .003539   .0021131     1.67   0.095    -.0006179    .0076959
          8  |   .0127909   .0024143     5.30   0.000     .0080414    .0175403
          9  |   .0160119      .0026     6.16   0.000     .0108971    .0211268
         10  |   .0209367   .0026079     8.03   0.000     .0158063     .026067
         11  |   .0141242   .0023658     5.97   0.000     .0094702    .0187783
         12  |   .0133079   .0019471     6.83   0.000     .0094775    .0171383
             |
        date |  -6.06e-07   1.84e-06    -0.33   0.742    -4.22e-06    3.01e-06
       _cons |  -.0083676   .0018948    -4.42   0.000     -.012095   -.0046401
------------------------------------------------------------------------------

. predict nonfarm3
(option xb assumed; fitted values)
(601 missing values generated)

. gen ubnonfarm3=nonfarm3+1.96*e(rmse)
(601 missing values generated)

. gen lbnonfarm3=nonfarm3-1.96*e(rmse)
(601 missing values generated)

. tsline ubnonfarm3 lbnonfarm3 nonfarm3 d.nonfarm3 if tin(2017m12, 2018m12)

. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)

      Source |       SS           df       MS      Number of obs   =       491
-------------+----------------------------------   F(32, 458)      =    109.93
       Model |  .042575775        32  .001330493   Prob > F        =    0.0000
    Residual |  .005543023       458  .000012103   R-squared       =    0.8848
-------------+----------------------------------   Adj R-squared   =    0.8768
       Total |  .048118799       490  .000098202   Root MSE        =    .00348

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.0311684   .0380987    -0.82   0.414    -.1060383    .0437014
        L2D. |   .0011509   .0387186     0.03   0.976    -.0749371     .077239
        L3D. |   .1811019   .0383354     4.72   0.000     .1057668    .2564371
        L4D. |    .122168   .0372877     3.28   0.001     .0488919     .195444
        L5D. |   .1131335   .0394764     2.87   0.004     .0355562    .1907107
        L6D. |   .0897111   .0411525     2.18   0.030     .0088401    .1705822
        L7D. |   .0046318   .0409493     0.11   0.910    -.0758401    .0851037
        L8D. |  -.0345344   .0396846    -0.87   0.385    -.1125109     .043462
```

```
         L8D.  |  -.0345244   .0396846    -0.87   0.385    -.1125109     .043462
         L9D.  |   .0809663   .0376451     2.15   0.032     .0069878    .1549448
        L10D.  |  -.1108451   .0375797    -2.95   0.003    -.1846951   -.0369951
        L11D.  |  -.0221662     .03865    -0.57   0.567    -.0981196    .0537872
        L12D.  |   .3376832   .0476368     7.09   0.000     .2440695    .4312969
        L24D.  |   .2364096     .04478     5.28   0.000       .14841    .3244093
               |
       lnfllf  |
          LD.  |  -.1300766   .0692757    -1.88   0.061    -.2662142    .0060609
         L2D.  |  -.2188669   .0684865    -3.20   0.001    -.3534536   -.0842802
        L12D.  |  -.0319706    .070685    -0.45   0.651    -.1708776    .1069365
        L24D.  |   .2248965     .06562     3.43   0.001     .0959428    .3538501
               |
      lnusepr  |
          LD.  |   .1537422    .092086     1.67   0.096    -.0272212    .3347056
         L2D.  |   .2542549   .0916164     2.78   0.006     .0742143    .4342955
        L12D.  |  -.0218538   .0949163    -0.23   0.818    -.2083793    .1646718
        L24D.  |  -.4432243   .0880674    -5.03   0.000    -.6162905   -.2701581
               |
        month  |
            2  |   .0133765    .002229     6.00   0.000     .0089962    .0177569
            3  |   .0150211   .0024611     6.10   0.000     .0101847    .0198576
            4  |   .0140381   .0025223     5.57   0.000     .0090814    .0189949
            5  |   .0082786   .0023045     3.59   0.000     .0037499    .0128072
            6  |   .0092759   .0023797     3.90   0.000     .0045993    .0139524
            7  |   .0081629   .0019705     4.14   0.000     .0042905    .0120353
            8  |   .0119157   .0020208     5.90   0.000     .0079446    .0158868
            9  |   .0141217   .0020005     7.06   0.000     .0101904     .018053
           10  |   .0242635   .0025397     9.55   0.000     .0192726    .0292543
           11  |   .0162918   .0023257     7.01   0.000     .0117215    .0208621
           12  |    .012759   .0018654     6.84   0.000     .0090932    .0164248
               |
        _cons  |  -.0119962   .0017171    -6.99   0.000    -.0153705   -.0086219
------------------------------------------------------------------------------

. predict nonfarm4
(option xb assumed; fitted values)
(469 missing values generated)

. gen ubnonfarm4=nonfarm4+1.96*e(rmse)
(469 missing values generated)

. gen lbnonfarm4=nonfarm4-1.96*e(rmse)
(469 missing values generated)

. tsline ubnonfarm4 lbnonfarm4 nonfarm4 d.nonfarm4 if tin(2017m12, 2018m12)

.
. *5
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date

      Source |       SS           df       MS      Number of obs   =       371
-------------+----------------------------------   F(60, 310)      =     49.11
       Model |  .033628663        60  .000560478   Prob > F        =    0.0000
    Residual |  .003538017       310  .000011413   R-squared       =    0.9048
-------------+----------------------------------   Adj R-squared   =    0.8864
       Total |   .03716668       370   .00010045   Root MSE        =    .00338

------------------------------------------------------------------------------
D.             |
 lnflnonfarm   |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------------+----------------------------------------------------------------
 lnflnonfarm   |
          LD.  |  -.1709382   .0544484    -3.14   0.002    -.2780734    -.063803
         L2D.  |  -.1482615   .0563272    -2.63   0.009    -.2590935   -.0374296
         L3D.  |    .162612    .057373     2.83   0.005     .0497221    .2755018
         L4D.  |   .1338157   .0578492     2.31   0.021     .0199889    .2476424
         L5D.  |   .0485865   .0586353     0.83   0.408     -.066787    .1639599
         L6D.  |   .1034826   .0583736     1.77   0.077    -.0113761    .2183412
         L7D.  |  -.0017194   .0587605    -0.03   0.977    -.1173393    .1139005
         L8D.  |  -.0661616   .0589687    -1.12   0.263    -.1821911     .049868
         L9D.  |   .0703003   .0570451     1.23   0.219    -.0419442    .1825448
        L10D.  |  -.2131464   .0557377    -3.82   0.000    -.3228184   -.1034745
        L11D.  |  -.0492711   .0559229    -0.88   0.379    -.1593075    .0607653
        L12D.  |   .3160254   .0547464     5.77   0.000     .2083039    .4237469
               |
       lnfllf  |
          LD.  |  -.1446453   .0983821    -1.47   0.143    -.3382263    .0489358
         L2D.  |  -.1262282   .0987995    -1.28   0.202    -.3206305    .0681742
         L3D.  |  -.0938871   .0998186    -0.94   0.348    -.2902948    .1025206
         L4D.  |  -.0237969   .1021375    -0.23   0.816    -.2247673    .1771735
         L5D.  |    .009078   .1014128     0.09   0.929    -.1904666    .2086226
         L6D.  |  -.1064011   .1005346    -1.06   0.291    -.3042177    .0914155
         L7D.  |  -.0363673   .1001261    -0.36   0.717    -.2333801    .1606454
         L8D.  |  -.0227101   .0999056    -0.23   0.820     -.219289    .1738688
         L9D.  |   .1419096   .0991963     1.43   0.154    -.0532736    .3370929
        L10D.  |   .2356432   .0995816     2.37   0.019     .0397018    .4315847
        L11D.  |  -.0156215   .1005373    -0.16   0.877    -.2134433    .1822003
        L12D.  |  -.1496896   .0991977    -1.51   0.132    -.3448754    .0454963
               |
      lnusepr  |
          LD.  |   .2160403   .1318859     1.64   0.102    -.0434644    .4755451
         L2D.  |   .0596176   .1342172     0.44   0.657    -.2044743    .3237096
         L3D.  |   .1365192   .1332945     1.02   0.307    -.1257572    .3987956
         L4D.  |    .140566   .1327682     1.06   0.291    -.1206748    .4018068
         L5D.  |  -.0429764   .1313909    -0.33   0.744    -.3015071    .2155544
         L6D.  |   .3174743   .1307471     2.43   0.016     .0602104    .5747382
         L7D.  |   .1643272   .1319288     1.25   0.214    -.0952619    .4239164
         L8D.  |  -.0363099   .1308906    -0.28   0.782    -.2938562    .2212364
         L9D.  |  -.1104225   .1311797    -0.84   0.401    -.3685377    .1476927
        L10D.  |  -.3143018   .1317767    -2.39   0.018    -.5735916    -.055012
        L11D.  |   .1870841   .1321575     1.42   0.158     -.072955    .4471232
        L12D.  |   .1595031    .127988     1.25   0.214     -.092332    .4113382
```

```
     lnflbp |
        LD. |   .0014899   .0015781     0.94   0.346    -.0016152     .004595
       L2D. |   .0048244   .0018575     2.60   0.010     .0011694    .0084793
       L3D. |   .0065472   .0019601     3.34   0.001     .0026904     .010404
       L4D. |   .0049279    .001984     2.48   0.014     .0010242    .0088316
       L5D. |   .0050421   .0019821     2.54   0.011      .001142    .0089422
       L6D. |   .0049581   .0020017     2.48   0.014     .0010194    .0088969
       L7D. |   .0043549   .0020066     2.17   0.031     .0004067    .0083031
       L8D. |   .0036407   .0020487     1.78   0.077    -.0003905    .0076719
       L9D. |   .0033308   .0020403     1.63   0.104    -.0006839    .0073454
      L10D. |    .004421   .0020019     2.21   0.028      .000482      .00836
      L11D. |   .0031925   .0019124     1.67   0.096    -.0005706    .0069555
      L12D. |   .0030942   .0015709     1.97   0.050     3.18e-06    .0061852
            |
      month |
         2  |   .0096471   .0035242     2.74   0.007     .0027127    .0165816
         3  |   .0085261     .00368     2.32   0.021     .0012851    .0157671
         4  |   .0092646   .0040143     2.31   0.022     .0013658    .0171635
         5  |   .0035453   .0031449     1.13   0.260    -.0026427    .0097333
         6  |  -.0026169   .0037248    -0.70   0.483    -.0099459    .0047121
         7  |   .0042003   .0035921     1.17   0.243    -.0028678    .0112683
         8  |   .0133789   .0037305     3.59   0.000     .0060387    .0207191
         9  |   .0108298   .0031587     3.43   0.001     .0046146     .017045
        10  |   .0165185   .0040273     4.10   0.000     .0085942    .0244428
        11  |   .0090475   .0037037     2.44   0.015     .0017599    .0163351
        12  |   .0154964   .0034704     4.47   0.000     .0086679     .022325
            |
       date |  -3.66e-06   1.92e-06    -1.90   0.058    -7.43e-06    1.22e-07
      _cons |  -.0044454   .0028196    -1.58   0.116    -.0099933    .0011025
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        371   1181.759   1618.026      61   -3114.053  -2875.164
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df1=el(r(S),1,4)

. scalar define aic1=el(r(S),1,5)

. scalar define bic1=el(r(S),1,6)

. loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date


 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   .00380836
Mean Absolute Errors     |   .00278549
Pseudo-R2                |   .85500572
-----------------------------------------

. scalar define loormse1=r(rmse)

.
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date

      Source |       SS           df       MS      Number of obs   =       381
-------------+----------------------------------   F(30, 350)      =     90.75
       Model |  .033990108        30  .001133004   Prob > F        =    0.0000
    Residual |  .004369869       350  .000012485   R-squared       =    0.8861
-------------+----------------------------------   Adj R-squared   =    0.8763
       Total |  .038359977       380  .000100947   Root MSE        =    .00353

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.1252994   .0491434    -2.55   0.011    -.2219529   -.0286459
        L2D. |  -.0958602   .0518133    -1.85   0.065    -.1977648    .0060443
        L3D. |   .2398422   .0515342     4.65   0.000     .1384866    .3411978
        L4D. |   .1709118   .0489573     3.49   0.001     .0746243    .2671992
        L5D. |   .1515931   .0495961     3.06   0.002     .0540493     .249137
        L6D. |   .1371627   .0507474     2.70   0.007     .0373545    .2369708
        L7D. |   .0720218   .0527616     1.37   0.173    -.0317478    .1757914
        L8D. |   .0100807   .0513147     0.20   0.844    -.0908432    .1110047
        L9D. |   .0796876   .0488054     1.63   0.103    -.0163013    .1756764
       L10D. |  -.1886804    .047519    -3.97   0.000    -.2821391   -.0952217
       L11D. |  -.0844174   .0497603    -1.70   0.091    -.1822842    .0134493
       L12D. |    .388731   .0497483     7.81   0.000     .2908879    .4865741
             |
      lnfllf |
         LD. |    -.19426   .0941693    -2.06   0.040    -.3794688   -.0090513
        L2D. |  -.1402653   .0944467    -1.49   0.138    -.3260199    .0454893
             |
     lnusepr |
         LD. |   .2273847   .1224594     1.86   0.064    -.0134641    .4682334
        L2D. |   .0344122    .123157     0.28   0.780    -.2078086     .276633
             |
      lnflbp |
         LD. |   .0002646   .0014455     0.18   0.855    -.0025784    .0031076
        L2D. |   .0014358   .0014491     0.99   0.322    -.0014143    .0042858
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
month						
2	.0089123	.0022594	3.94	0.000	.0044686	.0133561
3	.0063439	.0027078	2.34	0.020	.0010183	.0116696
4	.0081387	.0027662	2.94	0.003	.0026983	.0135792
5	.002026	.0026956	0.75	0.453	-.0032757	.0073277
6	.0005971	.0024291	0.25	0.806	-.0041804	.0053746
7	.0012385	.0021082	0.59	0.557	-.0029078	.0053847
8	.011499	.0024647	4.67	0.000	.0066515	.0163465
9	.0150089	.002545	5.90	0.000	.0100036	.0200143
10	.0207091	.0025776	8.03	0.000	.0156395	.0257787
11	.0124833	.0024281	5.14	0.000	.0077077	.0172588
12	.0125059	.0019735	6.34	0.000	.0086245	.0163873
date	-9.91e-07	1.73e-06	-0.57	0.568	-4.40e-06	2.42e-06
_cons	-.0069309	.0019081	-3.63	0.000	-.0106838	-.0031781

```
. estat ic
```

Akaike's information criterion and Bayesian information criterion

Model	N	ll(null)	ll(model)	df	AIC	BIC
.	381	1212.659	1626.478	31	-3190.957	-3068.73

Note: BIC uses N = number of observations. See [R] BIC note.

```
. scalar define df2=el(r(S),1,4)

. scalar define aic2=el(r(S),1,5)

. scalar define bic2=el(r(S),1,6)

. loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date
```

Leave-One-Out Cross-Validation Results

Method	Value
Root Mean Squared Errors	.00371849
Mean Absolute Errors	.00266976
Pseudo-R2	.86242412

```
. scalar define loormse2=r(rmse)

.
. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month date
```

Source	SS	df	MS
Model	.033019839	30	.001100661
Residual	.004146841	340	.000012197
Total	.03716668	370	.00010045

Number of obs	=	371
F(30, 340)	=	90.24
Prob > F	=	0.0000
R-squared	=	0.8884
Adj R-squared	=	0.8786
Root MSE	=	.00349

D. lnflnonfarm	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
lnflnonfarm						
LD.	-.1007022	.0496694	-2.03	0.043	-.1984002	-.0030042
L2D.	-.0776112	.0500347	-1.55	0.122	-.1760278	.0208054
L3D.	.2495718	.0489276	5.10	0.000	.153333	.3458107
L4D.	.1659053	.0490998	3.38	0.001	.0693277	.2624828
L5D.	.1576085	.0494045	3.19	0.002	.0604315	.2547854
L6D.	.1606861	.0504142	3.19	0.002	.0615231	.2598491
L7D.	.0920205	.0512561	1.80	0.073	-.0087985	.1928394
L8D.	.0257082	.0504855	0.51	0.611	-.073595	.1250115
L9D.	.0816557	.0485929	1.68	0.094	-.013925	.1772363
L10D.	-.1982253	.0474658	-4.18	0.000	-.2915889	-.1048616
L11D.	-.1022931	.049168	-2.08	0.038	-.199005	-.0055813
L12D.	.3672239	.0495416	7.41	0.000	.2697773	.4646706
lnfllf						
LD.	-.0415721	.0517745	-0.80	0.423	-.1434108	.0602666
L2D.	-.1119071	.0515073	-2.17	0.030	-.2132203	-.0105939
L12D.	-.1255582	.0492626	-2.55	0.011	-.2224562	-.0286603
lnflbp						
LD.	.0007479	.0014387	0.52	0.604	-.0020821	.0035778
L2D.	.0019096	.0014405	1.33	0.186	-.0009239	.004743
L12D.	.0016474	.0012703	1.30	0.196	-.0008512	.0041461
month						
2	.0086958	.0020531	4.24	0.000	.0046573	.0127342
3	.0077648	.0024066	3.23	0.001	.0030311	.0124985
4	.0089026	.0027229	3.27	0.001	.0035469	.0142584
5	.0038752	.0026664	1.45	0.147	-.0013695	.0091199
6	.0010548	.0023999	0.44	0.661	-.0036658	.0057754
7	.0040191	.0020623	1.95	0.052	-.0000374	.0080755
8	.0127434	.0023612	5.40	0.000	.008099	.0173878
9	.0156048	.0025314	6.16	0.000	.0106256	.0205839
10	.0204691	.0025295	8.09	0.000	.0154936	.0254446
11	.0136586	.0022928	5.96	0.000	.0091487	.0181685
12	.0133801	.001901	7.04	0.000	.0096408	.0171194
date	-6.31e-07	1.73e-06	-0.37	0.715	-4.03e-06	2.76e-06
_cons	-.0082167	.0018377	-4.47	0.000	-.0118314	-.0046019

```
. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        371   1181.759   1588.573      31   -3115.145  -2993.743
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df3=el(r(S),1,4)

. scalar define aic3=el(r(S),1,5)

. scalar define bic3=el(r(S),1,6)

. loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month date

 Leave-One-Out Cross-Validation Results
-----------------------------------------
         Method          |    Value
-------------------------+---------------
Root Mean Squared Errors |   .00375319
Mean Absolute Errors     |   .0026647
Pseudo-R2                |   .85864965
-----------------------------------------

. scalar define loormse3=r(rmse)

.
. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month

      Source |       SS           df       MS      Number of obs   =       503
-------------+----------------------------------   F(32, 470)      =    111.72
       Model |  .043188967        32  .001349655   Prob > F        =    0.0000
    Residual |  .005677954       470  .000012081   R-squared       =    0.8838
-------------+----------------------------------   Adj R-squared   =    0.8759
       Total |  .048866921       502  .000097344   Root MSE        =    .00348

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.0255499     .03766    -0.68   0.498    -.0995527    .0484528
        L2D. |  -.0022421    .038345    -0.06   0.953    -.0775909    .0731068
        L3D. |   .1749722   .0380384     4.60   0.000     .1002259    .2497186
        L4D. |   .1177758   .0368875     3.19   0.002      .045291    .1902607
        L5D. |   .1127275   .0389003     2.90   0.004     .0362875    .1891675
        L6D. |    .090976   .0406874     2.24   0.026     .0110242    .1709278
        L7D. |   .0090996   .0405277     0.22   0.822    -.0705384    .0887376
        L8D. |  -.0366038   .0392974    -0.93   0.352    -.1138241    .0406165
        L9D. |   .0766282   .0372788     2.06   0.040     .0033744    .1498819
       L10D. |  -.1156119   .0372913    -3.10   0.002    -.1888902   -.0423337
       L11D. |   -.015338   .0383706    -0.40   0.690     -.090737    .0600611
       L12D. |   .3692339   .0461555     8.00   0.000     .2785372    .4599307
       L24D. |   .1970316   .0414188     4.76   0.000     .1156426    .2784205
             |
      lnfllf |
         LD. |  -.1283352    .068539    -1.87   0.062     -.263016    .0063457
        L2D. |  -.2120246   .0677924    -3.13   0.002    -.3452383   -.0788109
       L12D. |   -.032049   .0699833    -0.46   0.647    -.1695679      .10547
       L24D. |   .2125143   .0650669     3.27   0.001     .0846562    .3403724
             |
     lnusepr |
         LD. |   .1405225   .0907762     1.55   0.122    -.0378531     .318898
        L2D. |   .2347858   .0904615     2.60   0.010     .0570268    .4125448
       L12D. |  -.0236969   .0938261    -0.25   0.801    -.2080674    .1606736
       L24D. |  -.4061036   .0863262    -4.70   0.000    -.5757366   -.2364706
             |
       month |
          2  |   .0130388   .0021912     5.95   0.000     .0087331    .0173445
          3  |   .0143267   .0024097     5.95   0.000     .0095916    .0190618
          4  |   .0133539   .0024757     5.39   0.000      .008489    .0182188
          5  |   .0079333   .0022518     3.52   0.000     .0035084    .0123582
          6  |   .0089776   .0023234     3.86   0.000      .004412    .0135431
          7  |    .007944   .0019343     4.11   0.000     .0041431     .011745
          8  |   .0120699   .0019881     6.07   0.000     .0081632    .0159766
          9  |   .0139913   .0019658     7.12   0.000     .0101286    .0178541
         10  |   .0232609   .0024957     9.32   0.000     .0183568     .028165
         11  |   .0157873   .0022757     6.94   0.000     .0113154    .0202591
         12  |   .0128709   .0018317     7.03   0.000     .0092715    .0164702
             |
       _cons |    -.01163   .0016815    -6.92   0.000    -.0149341   -.0083258
------------------------------------------------------------------------------

. estat ic

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |          N   ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
           . |        503   1609.944     2151.3      33     -4236.6  -4097.321
-----------------------------------------------------------------------------
Note: BIC uses N = number of observations. See [R] BIC note.

. scalar define df4=el(r(S),1,4)
```

```
. scalar define aic4=el(r(S),1,5)

. scalar define bic4=el(r(S),1,6)

. loocv reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month

 Leave-One-Out Cross-Validation Results
─────────────────────────────────────────────
         Method          │      Value
─────────────────────────┼───────────────────
Root Mean Squared Errors │    .00355785
Mean Absolute Errors     │    .00260948
Pseudo-R2                │    .86890552
─────────────────────────────────────────────

. scalar define loormse4=r(rmse)

.
. matrix drop _all

. matrix fit1=(df1,aic1,bic1,rmse1,loormse1)

. matrix fit2=(df2,aic2,bic2,rmse2,loormse2)

. matrix fit3=(df3,aic3,bic3,rmse3,loormse3)

. matrix fit4=(df4,aic4,bic4,rmse4,loormse4)

. matrix FIT=fit1\fit2\fit3\fit4

. matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 4"

. matrix colnames FIT=df AIC BIC RMSE LOORMSE

. matrix list FIT

FIT[4,5]
                df         AIC         BIC        RMSE     LOORMSE
Model 1         61  -3114.0527  -2875.1644   .00341856   .00380836
Model 2         31  -3190.9568  -3068.7301   .00356652   .00371849
Model 3         31  -3115.1451  -2993.7429   .00352901   .00375319
Model 4         33  -4236.6004  -4097.3209   .00347889   .00355785

.
. *6
. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)

      Source │       SS           df       MS      Number of obs   =       491
─────────────┼──────────────────────────────────   F(32, 458)      =    109.93
       Model │  .042575775        32  .001330493   Prob > F        =    0.0000
    Residual │  .005543023       458  .000012103   R-squared       =    0.8848
─────────────┼──────────────────────────────────   Adj R-squared   =    0.8768
       Total │  .048118799       490  .000098202   Root MSE        =    .00348

─────────────┬────────────────────────────────────────────────────────────────
D.           │
 lnflnonfarm │      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
─────────────┼────────────────────────────────────────────────────────────────
 lnflnonfarm │
         LD. │  -.0311684   .0380987    -0.82   0.414    -.1060383    .0437014
        L2D. │   .0011509   .0387186     0.03   0.976    -.0749371     .077239
        L3D. │   .1811019   .0383354     4.72   0.000     .1057668    .2564371
        L4D. │    .122168   .0372877     3.28   0.001     .0488919     .195444
        L5D. │   .1131335   .0394764     2.87   0.004     .0355562    .1907107
        L6D. │   .0897111   .0411525     2.18   0.030     .0088401    .1705822
        L7D. │   .0046318   .0409493     0.11   0.910    -.0758401    .0851037
        L8D. │  -.0345244   .0396846    -0.87   0.385    -.1125109     .043462
        L9D. │   .0809663   .0376451     2.15   0.032     .0069878    .1549448
       L10D. │  -.1108451   .0375797    -2.95   0.003    -.1846951   -.0369951
       L11D. │  -.0221662     .03865    -0.57   0.567    -.0981196    .0537872
       L12D. │   .3376832   .0476368     7.09   0.000     .2440695    .4312969
       L24D. │   .2364096     .04478     5.28   0.000       .14841    .3244093
             │
      lnfllf │
         LD. │  -.1300766   .0692757    -1.88   0.061    -.2662142    .0060609
        L2D. │  -.2188669   .0684865    -3.20   0.001    -.3534536   -.0842802
       L12D. │  -.0319706    .070685    -0.45   0.651    -.1708776    .1069365
       L24D. │   .2248965     .06562     3.43   0.001     .0959428    .3538501
             │
     lnusepr │
         LD. │   .1537422    .092086     1.67   0.096    -.0272212    .3347056
        L2D. │   .2542549   .0916164     2.78   0.006     .0742143    .4342955
       L12D. │  -.0218538   .0949163    -0.23   0.818    -.2083793    .1646718
       L24D. │  -.4432243   .0880674    -5.03   0.000    -.6162905   -.2701581
             │
       month │
          2  │   .0133765    .002229     6.00   0.000     .0089962    .0177569
          3  │   .0150211   .0024611     6.10   0.000     .0101847    .0198576
          4  │   .0140381   .0025223     5.57   0.000     .0090814    .0189949
          5  │   .0082786   .0023045     3.59   0.000     .0037499    .0128072
          6  │   .0092759   .0023797     3.90   0.000     .0045993    .0139524
          7  │   .0081629   .0019705     4.14   0.000     .0042905    .0120353
          8  │   .0119157   .0020208     5.90   0.000     .0079446    .0158868
          9  │   .0141217   .0020005     7.06   0.000     .0101904     .018053
         10  │   .0242635   .0025397     9.55   0.000     .0192726    .0292543
         11  │   .0162918   .0023257     7.01   0.000     .0117215    .0208621
         12  │    .012759   .0018654     6.84   0.000     .0090932    .0164248
             │
       _cons │  -.0119962   .0017171    -6.99   0.000    -.0153705   -.0086219
─────────────┴────────────────────────────────────────────────────────────────
```

```
. predict nonfarm6
(option xb assumed; fitted values)
(469 missing values generated)

. predict stdfore6, stdf
(469 missing values generated)

. gen pnonfarm6=exp(l.lnflnonfarm+nonfarm6)*exp(.5*e(rmse)^2)
(469 missing values generated)

. gen ubpnonfarm6=exp(l.lnflnonfarm+nonfarm6+1.96*stdfore6)*exp(.5*e(rmse)^2)
(469 missing values generated)

. gen lbpnonfarm6=exp(l.lnflnonfarm+nonfarm6-1.96*stdfore6)*exp(.5*e(rmse)^2)
(469 missing values generated)

. tsline ubpnonfarm6 lbpnonfarm6 pnonfarm6 fl_nonfarm if tin(2016m12,2019m12), tline(2018m12)

.
. *7
. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)

      Source |       SS           df       MS      Number of obs   =       491
-------------+----------------------------------   F(32, 458)      =    109.93
       Model |  .042575775        32  .001330493   Prob > F        =    0.0000
    Residual |  .005543023       458  .000012103   R-squared       =    0.8848
-------------+----------------------------------   Adj R-squared   =    0.8768
       Total |  .048118799       490  .000098202   Root MSE        =    .00348

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.0311684   .0380987    -0.82   0.414    -.1060383    .0437014
        L2D. |   .0011509   .0387186     0.03   0.976    -.0749371     .077239
        L3D. |   .1811019   .0383354     4.72   0.000     .1057668    .2564371
        L4D. |    .122168   .0372877     3.28   0.001     .0488919     .195444
        L5D. |   .1131335   .0394764     2.87   0.004     .0355562    .1907107
        L6D. |   .0897111   .0411525     2.18   0.030     .0088401    .1705822
        L7D. |   .0046318   .0409493     0.11   0.910    -.0758401    .0851037
        L8D. |  -.0345244   .0396846    -0.87   0.385    -.1125109     .043462
        L9D. |   .0809663   .0376451     2.15   0.032     .0069878    .1549448
       L10D. |  -.1108451   .0375797    -2.95   0.003    -.1846951   -.0369951
       L11D. |  -.0221662     .03865    -0.57   0.567    -.0981196    .0537872
       L12D. |   .3376832   .0476368     7.09   0.000     .2440695    .4312969
       L24D. |   .2364096     .04478     5.28   0.000       .14841    .3244093
             |
      lnfllf |
         LD. |  -.1300766   .0692757    -1.88   0.061    -.2662142    .0060609
        L2D. |  -.2188669   .0684865    -3.20   0.001    -.3534536   -.0842802
       L12D. |  -.0319706    .070685    -0.45   0.651    -.1708776    .1069365
       L24D. |   .2248965     .06562     3.43   0.001     .0959428    .3538501
             |
     lnusepr |
         LD. |   .1537422    .092086     1.67   0.096    -.0272212    .3347056
        L2D. |   .2542549   .0916164     2.78   0.006     .0742143    .4342955
       L12D. |  -.0218538   .0949163    -0.23   0.818    -.2083793    .1646718
       L24D. |  -.4432243   .0880674    -5.03   0.000    -.6162905   -.2701581
             |
       month |
          2  |   .0133765    .002229     6.00   0.000     .0089962    .0177569
          3  |   .0150211   .0024611     6.10   0.000     .0101847    .0198576
          4  |   .0140381   .0025223     5.57   0.000     .0090814    .0189949
          5  |   .0082786   .0023045     3.59   0.000     .0037499    .0128072
          6  |   .0092759   .0023797     3.90   0.000     .0045993    .0139524
          7  |   .0081629   .0019705     4.14   0.000     .0042905    .0120353
          8  |   .0119157   .0020208     5.90   0.000     .0079446    .0158868
          9  |   .0141217   .0020005     7.06   0.000     .0101904     .018053
         10  |   .0242635   .0025397     9.55   0.000     .0192726    .0292543
         11  |   .0162918   .0023257     7.01   0.000     .0117215    .0208621
         12  |    .012759   .0018654     6.84   0.000     .0090932    .0164248
             |
       _cons |  -.0119962   .0017171    -6.99   0.000    -.0153705   -.0086219
------------------------------------------------------------------------------

. predict nonfarm47
(option xb assumed; fitted values)
(469 missing values generated)

. predict pres47 if tin(2016m12,2018m12), residual
(947 missing values generated)

. gen exppres47=exp(pres47) if tin(2016m12,2018m12)
(947 missing values generated)

. summ exppres47

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
   exppres47 |         25    .9995604    .0061667   .9796714   1.016379

. gen pnonfarm47=r(mean)*exp(l.lnflnonfarm+nonfarm47)
(469 missing values generated)

. _pctile exppres47, percentile(2.5,97.5)

. gen lbpnonfarm47=r(r1)*exp(l.lnflnonfarm+nonfarm47)
(469 missing values generated)

. gen ubpnonfarm47=r(r2)*exp(l.lnflnonfarm+nonfarm47)
```

```
(469 missing values generated)

. tsline ubpnonfarm47 lbpnonfarm47 pnonfarm47 fl_nonfarm if tin(2016m12,2019m12), tline(2018m12)

.
. *8
. tsappend, add(1)

. replace month=month(dofm(date)) if month==.
(1 real change made)

.
. *9
. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)

      Source |       SS           df       MS      Number of obs   =       491
-------------+----------------------------------   F(32, 458)      =    109.93
       Model |  .042575775        32  .001330493   Prob > F        =    0.0000
    Residual |  .005543023       458  .000012103   R-squared       =    0.8848
-------------+----------------------------------   Adj R-squared   =    0.8768
       Total |  .048118799       490  .000098202   Root MSE        =    .00348

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.0311684   .0380987    -0.82   0.414    -.1060383    .0437014
        L2D. |   .0011509   .0387186     0.03   0.976    -.0749371     .077239
        L3D. |   .1811019   .0383354     4.72   0.000     .1057668    .2564371
        L4D. |    .122168   .0372877     3.28   0.001     .0488919     .195444
        L5D. |   .1131335   .0394764     2.87   0.004     .0355562    .1907107
        L6D. |   .0897111   .0411525     2.18   0.030     .0088401    .1705822
        L7D. |   .0046318   .0409493     0.11   0.910    -.0758401    .0851037
        L8D. |  -.0345244   .0396846    -0.87   0.385    -.1125109     .043462
        L9D. |   .0809663   .0376451     2.15   0.032     .0069878    .1549448
       L10D. |  -.1108451   .0375797    -2.95   0.003    -.1846951   -.0369951
       L11D. |  -.0221662     .03865    -0.57   0.567    -.0981196    .0537872
       L12D. |   .3376832   .0476368     7.09   0.000     .2440695    .4312969
       L24D. |   .2364096     .04478     5.28   0.000       .14841    .3244093
             |
      lnfllf |
         LD. |  -.1300766   .0692757    -1.88   0.061    -.2662142    .0060609
        L2D. |  -.2188669   .0684865    -3.20   0.001    -.3534536   -.0842802
       L12D. |  -.0319706    .070685    -0.45   0.651    -.1708776    .1069365
       L24D. |   .2248965     .06562     3.43   0.001     .0959428    .3538501
             |
     lnusepr |
         LD. |   .1537422    .092086     1.67   0.096    -.0272212    .3347056
        L2D. |   .2542549   .0916164     2.78   0.006     .0742143    .4342955
       L12D. |  -.0218538   .0949163    -0.23   0.818    -.2083793    .1646718
       L24D. |  -.4432243   .0880674    -5.03   0.000    -.6162905   -.2701581
             |
       month |
          2  |   .0133765    .002229     6.00   0.000     .0089962    .0177569
          3  |   .0150211   .0024611     6.10   0.000     .0101847    .0198576
          4  |   .0140381   .0025223     5.57   0.000     .0090814    .0189949
          5  |   .0082786   .0023045     3.59   0.000     .0037499    .0128072
          6  |   .0092759   .0023797     3.90   0.000     .0045993    .0139524
          7  |   .0081629   .0019705     4.14   0.000     .0042905    .0120353
          8  |   .0119157   .0020208     5.90   0.000     .0079446    .0158868
          9  |   .0141217   .0020005     7.06   0.000     .0101904     .018053
         10  |   .0242635   .0025397     9.55   0.000     .0192726    .0292543
         11  |   .0162918   .0023257     7.01   0.000     .0117215    .0208621
         12  |    .012759   .0018654     6.84   0.000     .0090932    .0164248
             |
       _cons |  -.0119962   .0017171    -6.99   0.000    -.0153705   -.0086219
------------------------------------------------------------------------------

. predict nonfarm9
(option xb assumed; fitted values)
(469 missing values generated)

. predict pres9 if tin(,2019m12), residual
(470 missing values generated)

. gen exppres9=exp(pres9) if tin(,2019m12)
(470 missing values generated)

. summ exppres9

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
    exppres9 |        503    1.000002    .0033683   .9796714   1.016379

. gen pnonfarm9=r(mean)*exp(l.lnflnonfarm+nonfarm9)
(469 missing values generated)

. _pctile exppres9, percentile(2.5,97.5)

. gen lbnonfarm9=r(r1)*exp(l.lnflnonfarm+nonfarm9)
(469 missing values generated)

. gen ubnonfarm9=r(r2)*exp(l.lnflnonfarm+nonfarm9)
(469 missing values generated)

. tsline ubnonfarm9 lbnonfarm9  pnonfarm9 fl_nonfarm if tin(2016m12,2020m1), tline(2019m12)

.
. *10
. reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
```

```
      Source |       SS           df       MS      Number of obs   =       491
-------------+----------------------------------   F(32, 458)      =    109.93
       Model |  .042575775        32  .001330493   Prob > F        =    0.0000
    Residual |  .005543023       458  .000012103   R-squared       =    0.8848
-------------+----------------------------------   Adj R-squared   =    0.8768
       Total |  .048118799       490  .000098202   Root MSE        =    .00348

------------------------------------------------------------------------------
D.           |
 lnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
 lnflnonfarm |
         LD. |  -.0311684   .0380987    -0.82   0.414    -.1060383    .0437014
        L2D. |   .0011509   .0387186     0.03   0.976    -.0749371     .077239
        L3D. |   .1811019   .0383354     4.72   0.000     .1057668    .2564371
        L4D. |    .122168   .0372877     3.28   0.001     .0488919     .195444
        L5D. |   .1131335   .0394764     2.87   0.004     .0355562    .1907107
        L6D. |   .0897111   .0411525     2.18   0.030     .0088401    .1705822
        L7D. |   .0046318   .0409493     0.11   0.910    -.0758401    .0851037
        L8D. |  -.0345244   .0396846    -0.87   0.385    -.1125109     .043462
        L9D. |   .0809663   .0376451     2.15   0.032     .0069878    .1549448
       L10D. |  -.1108451   .0375797    -2.95   0.003    -.1846951   -.0369951
       L11D. |  -.0221662     .03865    -0.57   0.567    -.0981196    .0537872
       L12D. |   .3376832   .0476368     7.09   0.000     .2440695    .4312969
       L24D. |   .2364096     .04478     5.28   0.000       .14841    .3244093
             |
      lnfl1f |
         LD. |  -.1300766   .0692757    -1.88   0.061    -.2662142    .0060609
        L2D. |  -.2188669   .0684865    -3.20   0.001    -.3534536   -.0842802
       L12D. |  -.0319706    .070685    -0.45   0.651    -.1708776    .1069365
       L24D. |   .2248965     .06562     3.43   0.001     .0959428    .3538501
             |
     lnusepr |
         LD. |   .1537422    .092086     1.67   0.096    -.0272212    .3347056
        L2D. |   .2542549   .0916164     2.78   0.006     .0742143    .4342955
       L12D. |  -.0218538   .0949163    -0.23   0.818    -.2083793    .1646718
       L24D. |  -.4432243   .0880674    -5.03   0.000    -.6162905   -.2701581
             |
       month |
          2  |   .0133765    .002229     6.00   0.000     .0089962    .0177569
          3  |   .0150211   .0024611     6.10   0.000     .0101847    .0198576
          4  |   .0140381   .0025223     5.57   0.000     .0090814    .0189949
          5  |   .0082786   .0023045     3.59   0.000     .0037499    .0128072
          6  |   .0092759   .0023797     3.90   0.000     .0045993    .0139524
          7  |   .0081629   .0019705     4.14   0.000     .0042905    .0120353
          8  |   .0119157   .0020208     5.90   0.000     .0079446    .0158868
          9  |   .0141217   .0020005     7.06   0.000     .0101904     .018053
         10  |   .0242635   .0025397     9.55   0.000     .0192726    .0292543
         11  |   .0162918   .0023257     7.01   0.000     .0117215    .0208621
         12  |    .012759   .0018654     6.84   0.000     .0090932    .0164248
             |
       _cons |  -.0119962   .0017171    -6.99   0.000    -.0153705   -.0086219
------------------------------------------------------------------------------

. predict nonfarm10
(option xb assumed; fitted values)
(469 missing values generated)

. predict stdfore10, stdf
(469 missing values generated)

. gen pnonfarm10=exp(l.lnflnonfarm+nonfarm10)*exp(.5*e(rmse)^2)
(469 missing values generated)

. gen ubnonfarm10=exp(l.lnflnonfarm+nonfarm10+1.96*stdfore10)*exp(.5*e(rmse)^2)
(469 missing values generated)

. gen lbnonfarm10=exp(l.lnflnonfarm+nonfarm10-1.96*stdfore10)*exp(.5*e(rmse)^2)
(469 missing values generated)

. tsline ubnonfarm10 lbnonfarm10 pnonfarm10 fl_nonfarm if tin(2016m12,2020m1), tline(2019m12)

.
. *11
. tsline fl_nonfarm if tin(2018m12,2020m1) || tsline  ubnonfarm10 lbnonfarm10 pnonfarm10 if tin(2019m12,), tline(2019m12)

.
. log close
      name:  <unnamed>
       log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 - Time Series/Problem Sets/Problem Set 4/Problem Set 4.smcl
  log type:  smcl
 closed on:  23 Mar 2021, 21:25:10
```

Problem Set 5

Gus Lipkin

CAP 4763 Time Series Modelling and Forecasting

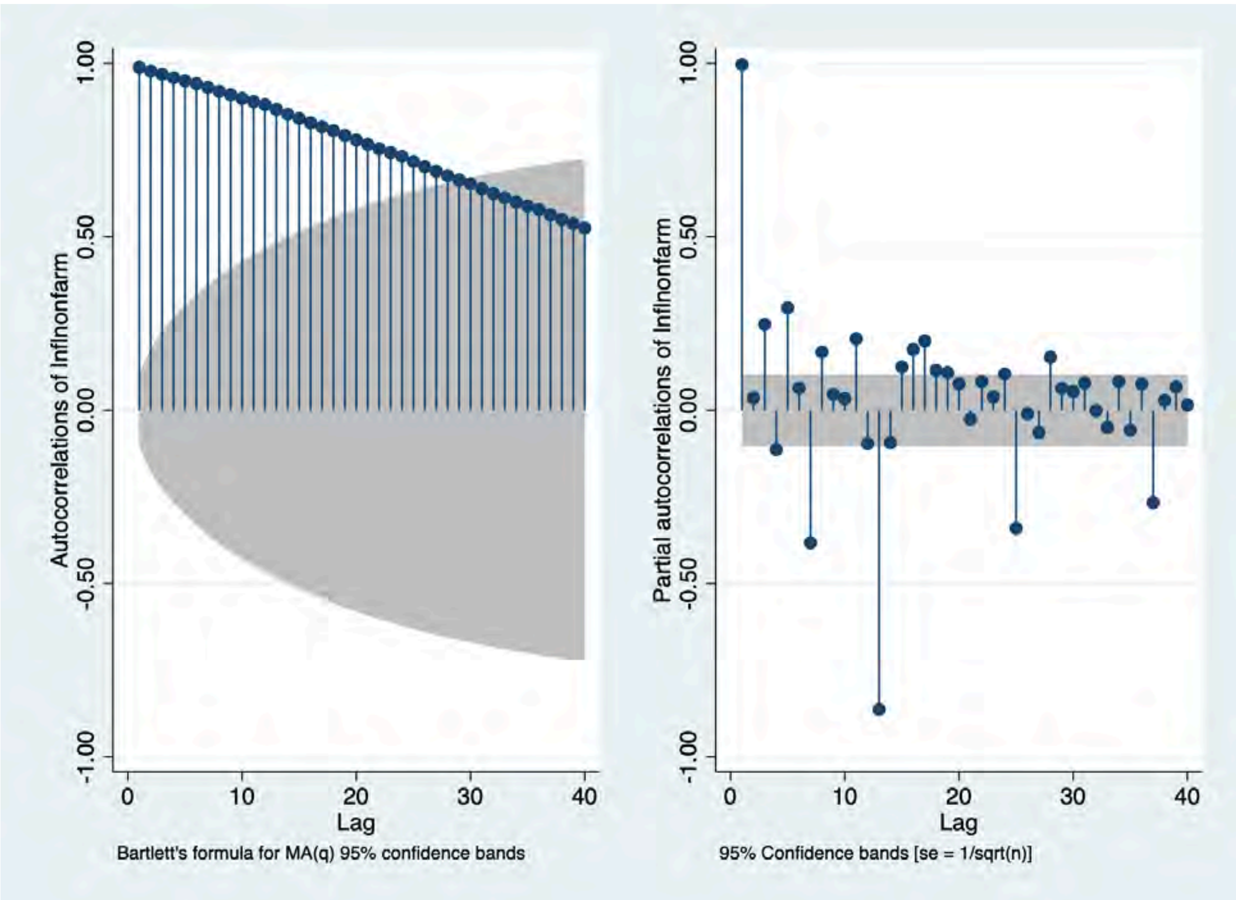
> All corrections are underlined

Table of Contents

Introduction

Time Series Modelling and Forecasting allows us to be better prepared for the future by using past data to predict future trends. In this case, we are trying to predict total nonfarm employment for the state of Florida for January 2020.

First, we create log transforms of all the variables. Because the variables cannot be negative, conducting a log transform ensures that they will not become negative later on. Next, we can generate monthly indicators so that if data trends are tied to a specific month or are affected seasonally, the indicator values will show the correlations. The last step before conducting the time series modelling and forecasting is to determine the dataset's stationarity in time. To do this, we construct an autocorrelogram and partial-autocorrelogram.



The high partial autocorrelation of the first lag indicates that we should use the first difference of the data. This corrects for stationarity.

GSREG, Rolling Window, and Choosing Models

GSREG

Because GSREG runs through every possible combination of variables fed to it up to a maximum number of variables per regression, it is necessary to limit the number of variables. How these variables are chosen is up to the person running the analysis. When I ran mine, I decided to include the first, third, sixth, ninth, twelfth, and twenty-fourth lags of each differenced variable for `lnflnonfarm`, `lnfllf`, and `lnusepr`. I also fixed the monthly indicators for January, March, June, and September. I chose to include all variables because while one variable may not have as heavy an influence, it is important to consider everything and conduct an analysis before dismissing any variables. I chose the lags and monthly indicators I did because while the data is monthly, I wanted to reduce the amount of variations that GSREG needs to go through without removing too many data points and without keeping too many variables that would cause the command to take too long to run. As for the twenty-fourth lag, I thought that there was a chance that long-term change would provide a grounding-point for the model so that it does not diverge too much. I understand that this can cause issues in times of immediate and rapid change such as the onset of COVID-19 but such events are few and far between.

The resulting best model suggested by GSREG is `reg d.lnflnonfarm l3d.lnflnonfarm l6d.lnflnonfarm l12d.lnflnonfarm l24d.lnflnonfarm l24d.lnfllf l6d.lnusepr m1 m3 m6 m9`. The models with the least variables had the four fixed month indicators and two other variables. The best of these is `reg d.lnflnonfarm l12d.lnflnonfarm m1 m3 m6 m9`. The third model I chose was of an average length at eight variables: `reg d.lnflnonfarm l3d.lnflnonfarm l12d.lnflnonfarm l24d.lnflnonfarm l24d.lnflnonfarm l6d.lnusepr m1 m3 m6 m9`.

Rolling Window

Running each model against the whole dataset in a Rolling Window model has a Root Mean Square Error of $.00388844$, $.00423688$, and $.00406403$, respectively. The first and third models had window widths of 108 observations while the second had a window width of 120. With the lowest RMSE, I decided to move forward with the first model. Calculating the percentiles for 2.5 and 97.5 of the distribution gives $-.0074653569608927$ and $.0065394379198551$.

Using the Best Model to Forecast January 2020

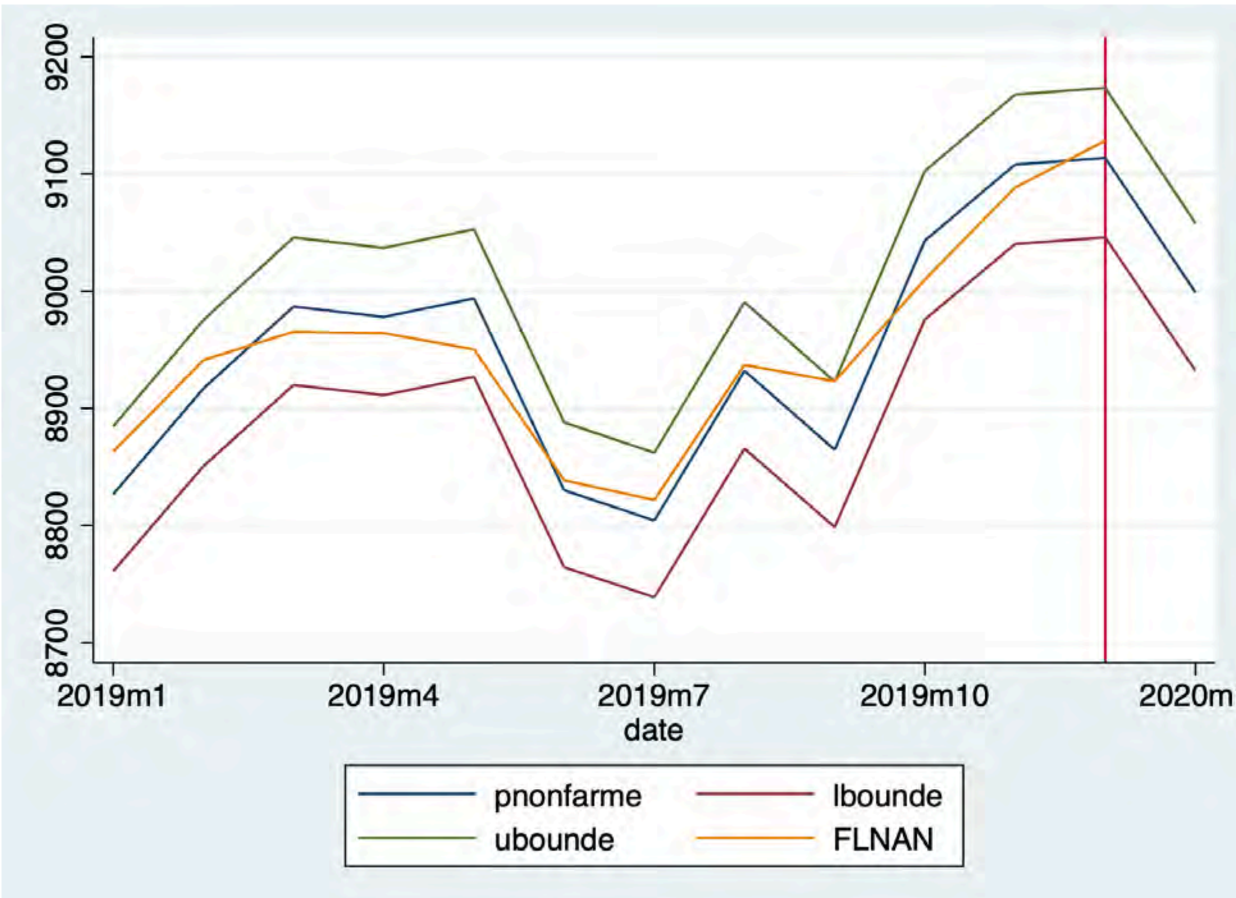
We can then use our best model and window width to forecast. If we limit our model data to several data points in the past, we can forecast the most recent past datapoints to check our model's accuracy.

month	pnonfarm	lbound	ubound
1	9001.077	8933.007	9069.667

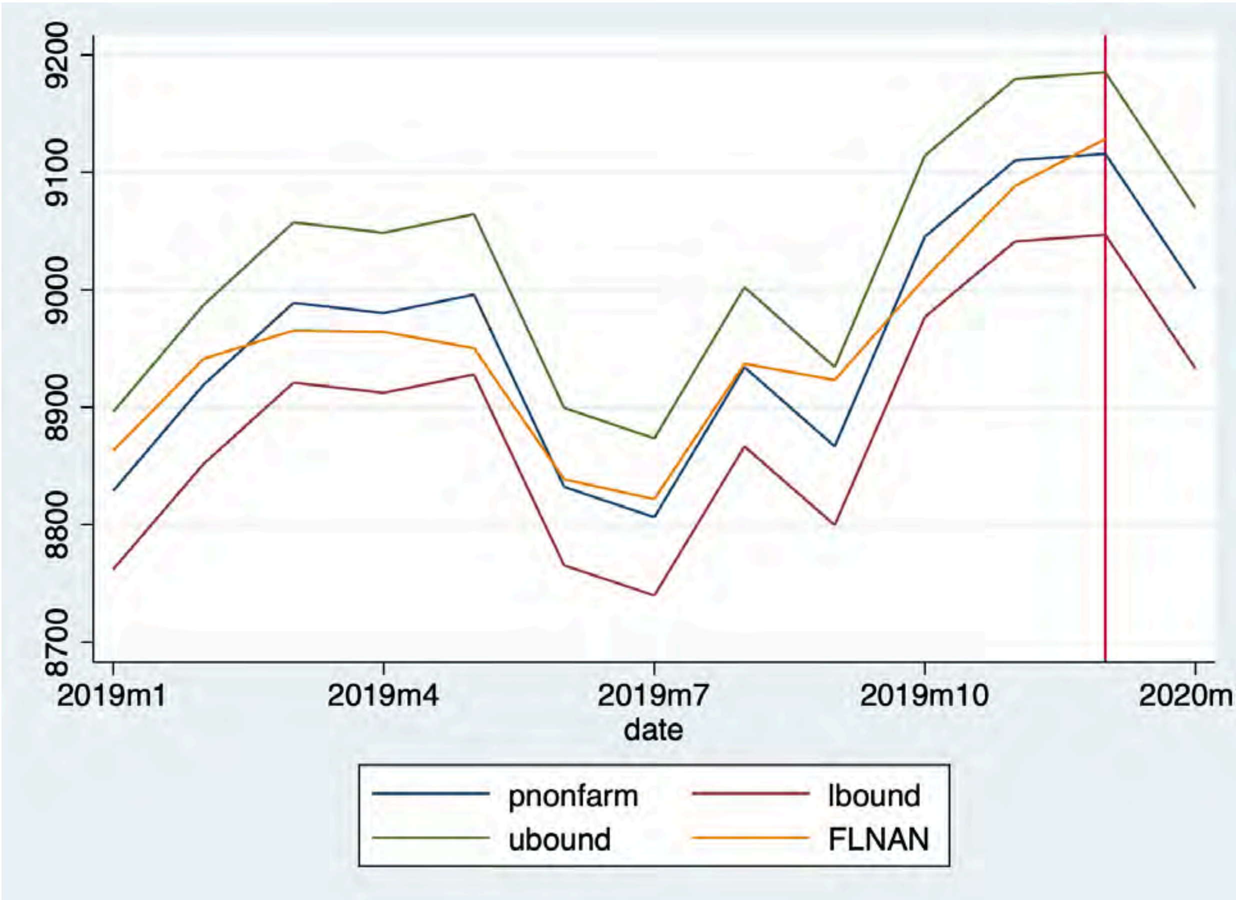
> Table 1: One Month Ahead Forecast Predicting Non-farm Employment for January 2020

When we consult the actual data for January 2020, we see that nonfarm employment was at 9050.4. While our forecast is low, it is still within the bounds set.

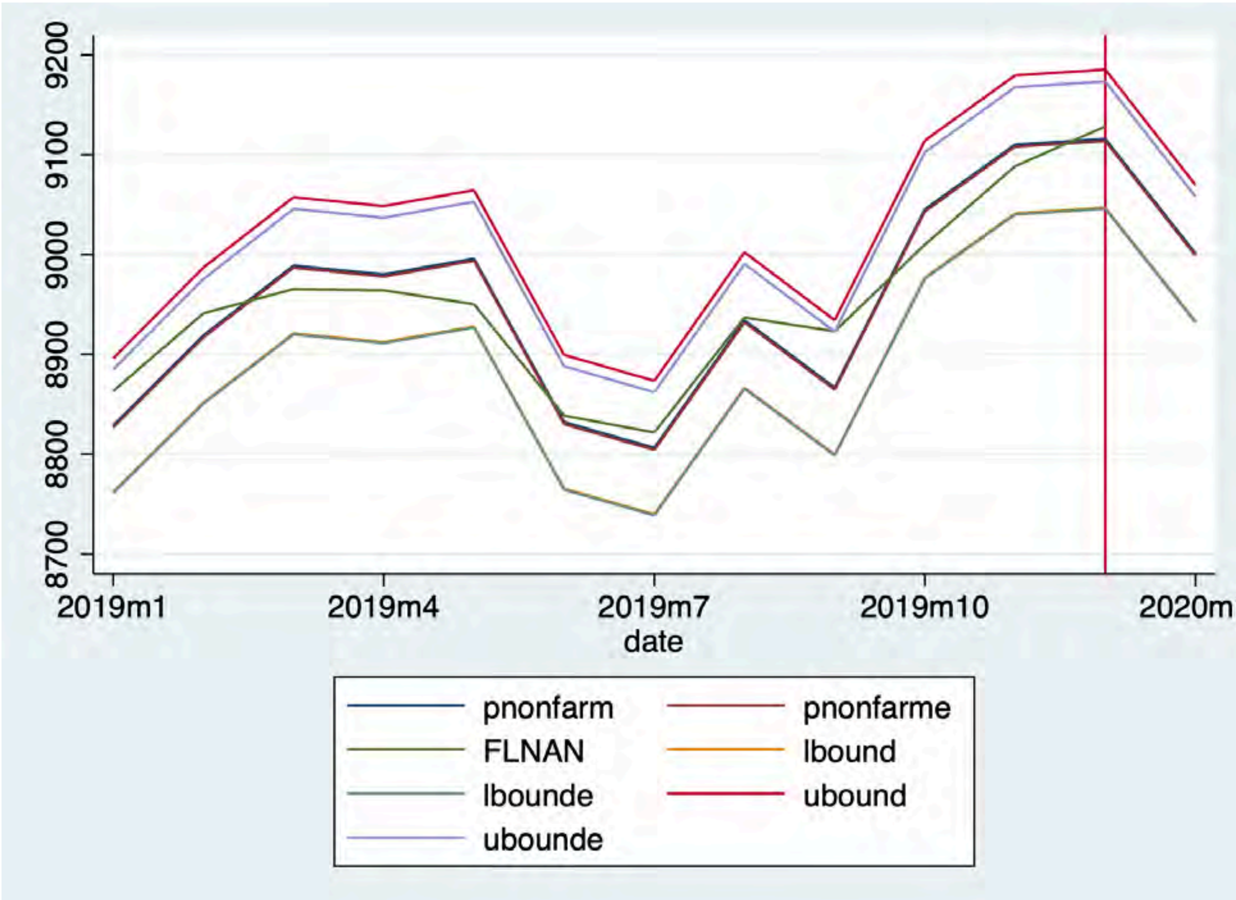
Illustrations and Interpretations of the Models



> Nonfarm Empirical



Nonfarm Normal



Nonfarm Normal vs Empirical

Conclusion

With proper adjustment of variables through log transforms, dummy variables, and differencing, we can create time series models of the past and use it to forecast the future with relative accuracy leading to potential insights or predictions about the future.

Appendix A (Code)

```
clear
set more off

cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Problem Set 5"
log using "Problem Set 5", replace
import delimited "Assignment_1_Monthly.txt"

rename lnu02300000 us_epr
rename flnan fl_nonfarm
rename fllfn fl_lf
rename flbppriv fl_bp
rename date datestring

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

tsset date

gen lnusepr=log(us_epr)
gen lnflnonfarm=log(fl_nonfarm)
gen lnfllf=log(fl_lf)
gen lnflbp=log(fl_bp)

*1
drop if !tin(1990m1,2019m12)

*2
tsset date
tsappend, add(1)
replace month=month(dofm(date)) if month==.

*interlude
ac lnflnonfarm, saving("ac_lnflnonfarm.gph", replace)
pac lnflnonfarm, saving("pac_lnflnonfarm.gph", replace)
graph combine ac_lnflnonfarm.gph pac_lnflnonfarm.gph
graph export ac_pac_lnflnonfarm.png, replace

*3
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 dlnflnonfarm=d.lnflnonfarm
gen l1dlnflnonfarm=l1d.lnflnonfarm
gen l2dlnflnonfarm=l2d.lnflnonfarm
gen l3dlnflnonfarm=l3d.lnflnonfarm
gen l4dlnflnonfarm=l4d.lnflnonfarm
gen l5dlnflnonfarm=l5d.lnflnonfarm
gen l6dlnflnonfarm=l6d.lnflnonfarm
gen l7dlnflnonfarm=l7d.lnflnonfarm
gen l8dlnflnonfarm=l8d.lnflnonfarm
gen l9dlnflnonfarm=l9d.lnflnonfarm
gen l10dlnflnonfarm=l10d.lnflnonfarm
gen l11dlnflnonfarm=l11d.lnflnonfarm
gen l12dlnflnonfarm=l12d.lnflnonfarm
gen l24dlnflnonfarm=l24d.lnflnonfarm

gen dlnfllf=d.lnfllf
gen l1dlnfllf=l1d.lnfllf
gen l2dlnfllf=l2d.lnfllf
gen l3dlnfllf=l3d.lnfllf
gen l4dlnfllf=l4d.lnfllf
gen l5dlnfllf=l5d.lnfllf
gen l6dlnfllf=l6d.lnfllf
gen l7dlnfllf=l7d.lnfllf
gen l8dlnfllf=l8d.lnfllf
gen l9dlnfllf=l9d.lnfllf
gen l10dlnfllf=l10d.lnfllf
gen l11dlnfllf=l11d.lnfllf
gen l12dlnfllf=l12d.lnfllf
```

```
gen l24dlnfllf=l24d.lnfllf

gen dlnusepr=d.lnusepr
gen l1dlnusepr=l1d.lnusepr
gen l2dlnusepr=l2d.lnusepr
gen l3dlnusepr=l3d.lnusepr
gen l4dlnusepr=l4d.lnusepr
gen l5dlnusepr=l5d.lnusepr
gen l6dlnusepr=l6d.lnusepr
gen l7dlnusepr=l7d.lnusepr
gen l8dlnusepr=l8d.lnusepr
gen l9dlnusepr=l9d.lnusepr
gen l10dlnusepr=l10d.lnusepr
gen l11dlnusepr=l11d.lnusepr
gen l12dlnusepr=l12d.lnusepr
gen l24dlnusepr=l24d.lnusepr


gsreg dlnflnonfarm l1dlnflnonfarm l3dlnflnonfarm l6dlnflnonfarm l9dlnflnonfarm ///
      l12dlnflnonfarm l24dlnflnonfarm ///
    l1dlnfllf l3dlnfllf l6dlnfllf l9dlnfllf ///
      l12dlnfllf l24dlnfllf ///
    l1dlnusepr l3dlnusepr l6dlnusepr l9dlnusepr ///
      l12dlnusepr l24dlnusepr if tin(1990m1,2019m12), ///
  ncomb(1,6) aic outsample(24) fix(m1 m3 m6 m9) ///
  samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnrer) replace


*5
/*
Best model
reg dlnflnonfarm l3dlnflnonfarm l6dlnflnonfarm l12dlnflnonfarm l24dlnflnonfarm
  l24dlnfllf l6dlnusepr m1 m3 m6 m9
*/
scalar drop _all
quietly forval w=48(12)144 {
gen pred=.
gen nobs=.
  forval t=529/720 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg d.lnflnonfarm l3d.lnflnonfarm l6d.lnflnonfarm l12d.lnflnonfarm l24d.lnflnonfarm ///
    l24d.lnfllf l6d.lnusepr m1 m3 m6 m9 ///
    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.lnflnonfarm)^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
/*
RWmaxobs108 = 108
RWminobs108 = 108
RWrmse108 = .00388844
*/

/*
Smallest / best model
reg dlnflnonfarm l12dlnflnonfarm m1 m3 m6 m9
*/
scalar drop _all
quietly forval w=48(12)144 {
gen pred=.
gen nobs=.
  forval t=529/720 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnflnonfarm l12dlnflnonfarm m1 m3 m6 m9 ///
    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.lnflnonfarm)^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
/*
RWmaxobs120 = 120
RWminobs120 = 120
```

```
RWrmse120 = .00423688

*/

/*
Best medium length model
reg dlnflnonfarm l3dlnflnonfarm l12dlnflnonfarm l24dlnflnonfarm l6dlnusepr
  m1 m3 m6 m9
*/
scalar drop _all
quietly forval w=48(12)144 {
gen pred=.
gen nobs=.
  forval t=529/720 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg dlnflnonfarm l3dlnflnonfarm l12dlnflnonfarm l24dlnflnonfarm l6dlnusepr ///
    m1 m3 m6 m9 ///
    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.lnflnonfarm)^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
/*
RWmaxobs108 = 108
RWminobs108 = 108
RWrmse108 = .00406403
*/

*6
/*
RWmaxobs108 = 108
RWminobs108 = 108
RWrmse108 = .00388844
*/
scalar drop _all
quietly forval w=156(12)156 {
gen pred=.
```

```
gen nobs=.
  forval t=432/720 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg d.lnflnonfarm l3d.lnflnonfarm l6d.lnflnonfarm l12d.lnflnonfarm
l24d.lnflnonfarm ///
    l24d.lnfllf l6d.lnusepr m1 m3 m6 m9 ///
    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.lnflnonfarm)^2
}
summ nobs // checking all had a full window
*get error info for normal interval
summ errsq
scalar rwrmse=r(mean)^0.5
scalar list rwrmse
gen res=(d.lnflnonfarm-pred)
_pctile res, percentile(2.5,97.5)
return list

*7
predict temp if tin(2020m1,2020m1)
replace pred=temp if tin(2020m1,2020m1)
drop temp
gen pnonfarm=exp(l.lnflnonfarm+pred+(rwrmse^2)/2)
gen ubound=exp(l.lnflnonfarm+pred+1.96*rwrmse+(rwrmse^2)/2)
gen lbound=exp(l.lnflnonfarm+pred-1.96*rwrmse+(rwrmse^2)/2)
list month pnonfarm lbound ubound if tin(2020m1,2020m1)
tsline pnonfarm lbound ubound fl_nonfarm if tin(2019m1,2020m1), tline(2019m12)
saving("Nonfarm_Normal", replace)
graph export "nonfarm_normal.png", replace

*8
*Empirical
drop res
gen res=(d.lnflnonfarm-pred)
gen expres=exp(res)
summ expres
scalar meanexpres=r(mean)
gen pnonfarme=exp(l.lnflnonfarm+pred)*meanexpres
_pctile res, percentile(2.5,97.5)
return list
gen lbounde=exp(l.lnflnonfarm+pred+r(r1))*meanexpres
gen ubounde=exp(l.lnflnonfarm+pred+r(r2))*meanexpres
```

```
list month pnonfarme lbounde ubounde if tin(2020m1,2020m1)
tsline pnonfarme lbounde ubounde fl_nonfarm if tin(2019m1,2020m1), ///
  tline(2019m12) saving("Nonfarm_Epirical", replace)
graph export "nonfarm_empirical.png", replace

*9
tsline pnonfarm pnonfarme fl_nonfarm lbound lbounde ubound ubounde ///
 if tin(2019m1,2020m1), tline(2019m12) saving("Normal_vs_Empirical", replace)
graph export "normal_vs_empirical.png", replace

translate "Problem Set 5.smcl" "Problem Set 5.txt", replace
log close
```

Appendix B (STATA Output)

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

      -------------------------------------------------------------------------
-----
            name:  <unnamed>
             log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time 
Series/Probl
      > em Sets/Problem Set 5/Problem Set 5.smcl
        log type:  smcl
       opened on:   3 Apr 2021, 21:50:15

     1 . import delimited "Assignment_1_Monthly.txt"
      (5 vars, 984 obs)

     2 .
     3 . rename lnu02300000 us_epr

     4 . rename flnan fl_nonfarm

     5 . rename fllfn fl_lf

     6 . rename flbppriv fl_bp

     7 . rename date datestring

     8 .
     9 . gen datec=date(datestring, "YMD")

    10 . gen date=mofd(datec)

    11 . gen month=month(datec)

    12 . format date %tm

    13 .
    14 . tsset date
               time variable:  date, 1939m1 to 2020m12
```

```
                       delta:  1 month

    15 .
    16 . gen lnusepr=log(us_epr)
      (108 missing values generated)

    17 . gen lnflnonfarm=log(fl_nonfarm)

    18 . gen lnfllf=log(fl_lf)
      (444 missing values generated)

    19 . gen lnflbp=log(fl_bp)
      (588 missing values generated)

    20 .
    21 . *1
    22 . drop if !tin(1990m1,2019m12)
      (624 observations deleted)

    23 .
    24 . *2
    25 . tsset date
               time variable:  date, 1990m1 to 2019m12
                       delta:  1 month

    26 . tsappend, add(1)

    27 . replace month=month(dofm(date)) if month==.
      (1 real change made)

    28 .
    29 . *interlude
    30 . ac lnflnonfarm, saving("ac_lnflnonfarm.gph", replace)
      (file ac_lnflnonfarm.gph saved)

    31 . pac lnflnonfarm, saving("pac_lnflnonfarm.gph", replace)
      (file pac_lnflnonfarm.gph saved)

    32 . graph combine ac_lnflnonfarm.gph pac_lnflnonfarm.gph

    33 . graph export ac_pac_lnflnonfarm.png, replace
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
      > /Problem Set 5/ac_pac_lnflnonfarm.png written in PNG format)

    34 .
    35 . *3
    36 . gen m1=0
```

```
    37 . replace m1=1 if month==1
      (31 real changes made)

    38 . gen m2=0

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

    40 . gen m3=0

    41 . replace m3=1 if month==3
      (30 real changes made)

    42 . gen m4=0

    43 . replace m4=1 if month==4
      (30 real changes made)

    44 . gen m5=0

    45 . replace m5=1 if month==5
      (30 real changes made)

    46 . gen m6=0

    47 . replace m6=1 if month==6
      (30 real changes made)

    48 . gen m7=0

    49 . replace m7=1 if month==7
      (30 real changes made)

    50 . gen m8=0

    51 . replace m8=1 if month==8
      (30 real changes made)

    52 . gen m9=0

    53 . replace m9=1 if month==9
      (30 real changes made)

    54 . gen m10=0

    55 . replace m10=1 if month==10
      (30 real changes made)
```

```
    56 . gen m11=0

    57 . replace m11=1 if month==11
      (30 real changes made)

    58 .
    59 . gen dlnflnonfarm=d.lnflnonfarm
      (2 missing values generated)

    60 . gen l1dlnflnonfarm=l1d.lnflnonfarm
      (2 missing values generated)

    61 . gen l2dlnflnonfarm=l2d.lnflnonfarm
      (3 missing values generated)

    62 . gen l3dlnflnonfarm=l3d.lnflnonfarm
      (4 missing values generated)

    63 . gen l4dlnflnonfarm=l4d.lnflnonfarm
      (5 missing values generated)

    64 . gen l5dlnflnonfarm=l5d.lnflnonfarm
      (6 missing values generated)

    65 . gen l6dlnflnonfarm=l6d.lnflnonfarm
      (7 missing values generated)

    66 . gen l7dlnflnonfarm=l7d.lnflnonfarm
      (8 missing values generated)

    67 . gen l8dlnflnonfarm=l8d.lnflnonfarm
      (9 missing values generated)

    68 . gen l9dlnflnonfarm=l9d.lnflnonfarm
      (10 missing values generated)

    69 . gen l10dlnflnonfarm=l10d.lnflnonfarm
      (11 missing values generated)

    70 . gen l11dlnflnonfarm=l11d.lnflnonfarm
      (12 missing values generated)

    71 . gen l12dlnflnonfarm=l12d.lnflnonfarm
      (13 missing values generated)

    72 . gen l24dlnflnonfarm=l24d.lnflnonfarm
      (25 missing values generated)
```

```

    73 .
    74 . gen dlnfllf=d.lnfllf
      (2 missing values generated)

    75 . gen l1dlnfllf=l1d.lnfllf
      (2 missing values generated)

    76 . gen l2dlnfllf=l2d.lnfllf
      (3 missing values generated)

    77 . gen l3dlnfllf=l3d.lnfllf
      (4 missing values generated)

    78 . gen l4dlnfllf=l4d.lnfllf
      (5 missing values generated)

    79 . gen l5dlnfllf=l5d.lnfllf
      (6 missing values generated)

    80 . gen l6dlnfllf=l6d.lnfllf
      (7 missing values generated)

    81 . gen l7dlnfllf=l7d.lnfllf
      (8 missing values generated)

    82 . gen l8dlnfllf=l8d.lnfllf
      (9 missing values generated)

    83 . gen l9dlnfllf=l9d.lnfllf
      (10 missing values generated)

    84 . gen l10dlnfllf=l10d.lnfllf
      (11 missing values generated)

    85 . gen l11dlnfllf=l11d.lnfllf
      (12 missing values generated)

    86 . gen l12dlnfllf=l12d.lnfllf
      (13 missing values generated)

    87 . gen l24dlnfllf=l24d.lnfllf
      (25 missing values generated)

    88 .
    89 . gen dlnusepr=d.lnusepr
      (2 missing values generated)

```

```
    90 . gen l1dlnusepr=l1d.lnusepr
      (2 missing values generated)

    91 . gen l2dlnusepr=l2d.lnusepr
      (3 missing values generated)

    92 . gen l3dlnusepr=l3d.lnusepr
      (4 missing values generated)

    93 . gen l4dlnusepr=l4d.lnusepr
      (5 missing values generated)

    94 . gen l5dlnusepr=l5d.lnusepr
      (6 missing values generated)

    95 . gen l6dlnusepr=l6d.lnusepr
      (7 missing values generated)

    96 . gen l7dlnusepr=l7d.lnusepr
      (8 missing values generated)

    97 . gen l8dlnusepr=l8d.lnusepr
      (9 missing values generated)

    98 . gen l9dlnusepr=l9d.lnusepr
      (10 missing values generated)

    99 . gen l10dlnusepr=l10d.lnusepr
      (11 missing values generated)

   100 . gen l11dlnusepr=l11d.lnusepr
      (12 missing values generated)

   101 . gen l12dlnusepr=l12d.lnusepr
      (13 missing values generated)

   102 . gen l24dlnusepr=l24d.lnusepr
      (25 missing values generated)

   103 .
   104 .
   105 . gsreg dlnflnonfarm l1dlnflnonfarm l3dlnflnonfarm l6dlnflnonfarm
l9dlnflnonfar
      > m ///
      >       l12dlnflnonfarm l24dlnflnonfarm ///
      >           l1dlnfllf l3dlnfllf l6dlnfllf l9dlnfllf ///
      >       l12dlnfllf l24dlnfllf ///
      >           l1dlnusepr l3dlnusepr l6dlnusepr l9dlnusepr ///
```

```
      >       l12dlnusepr l24dlnusepr if tin(1990m1,2019m12), ///
      >         ncomb(1,6) aic outsample(24) fix(m1 m3 m6 m9) ///
      >         samesample nindex( -1 aic -1 bic -1 rmse_out)
results(gsreg_dlnrer) r
      > eplace
      ----------------------------------------------------
      Total Number of Estimations: 31179
      ----------------------------------------------------
      ----------------------------------------------------
      Warning: Estimation could take about 14 minutes
      ----------------------------------------------------

      Computing combinations...
      Preparing regression list...
      Doing regressions...
      Saving results...
      file gsreg_dlnrer.dta saved
      ----------------------------------------------------
      Best estimation in terms of -1 aic -1 bic -1 rmse_out
      Estimation number 19215
      ----------------------------------------------------

            Source |       SS           df       MS      Number of obs   =
312
      -------------+----------------------------------   F(10, 301)      =
192.38
             Model |  .027480005        10     .002748   Prob > F        =
0.0000
          Residual |  .004299465       301  .000014284   R-squared       =
0.8647
      -------------+----------------------------------   Adj R-squared   =
0.8602
             Total |   .03177947       311  .000102185   Root MSE        =
.00378

      ----------------------------------------------------------------------------
-----
      --
         dlnflnonfarm |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
Interva
      > l]
      ----------------+-----------------------------------------------------------
-----
      --
                   m1 |  -.0068526   .0013736    -4.99   0.000    -.0095558
-.00414
      > 94
```

```
                   m3 |   .0009126   .0008591     1.06   0.289    -.0007781
.00260
      > 33
                   m6 |  -.0044233   .0010173    -4.35   0.000    -.0064252
-.00242
      > 14
                   m9 |  -.0008321   .0008569    -0.97   0.332    -.0025184
.00085
      > 41
       l3dlnflnonfarm |   .0992292   .0282426     3.51   0.001     .0436512
.15480
      > 71
       l6dlnflnonfarm |   .0741191   .0284776     2.60   0.010     .0180786
.13015
      > 96
      l12dlnflnonfarm |   .5446389    .067997     8.01   0.000     .4108292
.67844
      > 87
      l24dlnflnonfarm |   .1684682   .0598024     2.82   0.005     .0507844
.2861
      > 52
          l24dlnfllf |  -.1553647   .0508596    -3.05   0.002    -.2554501
-.05527
      > 93
          l6dlnusepr |   .1231808   .0528785     2.33   0.020     .0191224
.22723
      > 92
               _cons |   .0014172   .0003147     4.50   0.000     .0007979
.00203
      > 65
      --------------------------------------------------------------------------
-----
      --

   106 .
   107 .
   108 . *5
   109 . /*
      > Best model
      > reg dlnflnonfarm l3dlnflnonfarm l6dlnflnonfarm l12dlnflnonfarm
l24dlnflnonfar
      > m
      >         l24dlnfllf l6dlnusepr m1 m3 m6 m9
      > */
   110 . scalar drop _all

   111 . quietly forval w=48(12)144 {

```

```
112 . scalar list
   RWmaxobs144 =        144
   RWminobs144 =        144
    RWrmse144 =  .00396645
   RWmaxobs132 =        132
   RWminobs132 =        132
    RWrmse132 =  .00390407
   RWmaxobs120 =        120
   RWminobs120 =        120
    RWrmse120 =  .00388926
   RWmaxobs108 =        108
   RWminobs108 =        108
    RWrmse108 =  .00388844
   RWmaxobs96 =         96
   RWminobs96 =         96
     RWrmse96 =  .00403691
   RWmaxobs84 =         84
   RWminobs84 =         84
     RWrmse84 =  .00406426
   RWmaxobs72 =         72
   RWminobs72 =         72
     RWrmse72 =  .00411873
   RWmaxobs60 =         60
   RWminobs60 =         60
     RWrmse60 =  .00431692
   RWmaxobs48 =         48
   RWminobs48 =         48
     RWrmse48 =  .00460352

113 . /*
   > RWmaxobs108 = 108
   > RWminobs108 = 108
   > RWrmse108 = .00388844
   > */
114 .
115 . /*
   > Smallest / best model
   > reg dlnflnonfarm l12dlnflnonfarm m1 m3 m6 m9
   > */
116 . scalar drop _all

117 . quietly forval w=48(12)144 {

118 . scalar list
   RWmaxobs144 =        144
   RWminobs144 =        144
    RWrmse144 =  .00431666
   RWmaxobs132 =        132
```

```
      RWminobs132 =        132
       RWrmse132 =  .00426742
      RWmaxobs120 =        120
      RWminobs120 =        120
       RWrmse120 =  .00423688
      RWmaxobs108 =        108
      RWminobs108 =        108
       RWrmse108 =  .00428159
      RWmaxobs96 =         96
      RWminobs96 =         96
        RWrmse96 =  .00436091
      RWmaxobs84 =         84
      RWminobs84 =         84
        RWrmse84 =  .00439555
      RWmaxobs72 =         72
      RWminobs72 =         72
        RWrmse72 =  .00443487
      RWmaxobs60 =         60
      RWminobs60 =         60
        RWrmse60 =  .00453048
      RWmaxobs48 =         48
      RWminobs48 =         48
        RWrmse48 =  .00458215

   119 . /*
      > RWmaxobs120 = 120
      > RWminobs120 = 120
      > RWrmse120 = .00423688
      >
      > */
   120 .
   121 . /*
      > Best medium length model
      > reg dlnflnonfarm l3dlnflnonfarm l12dlnflnonfarm l24dlnflnonfarm l6dlnusepr
      >          m1 m3 m6 m9
      > */
   122 . scalar drop _all

   123 . quietly forval w=48(12)144 {

   124 . scalar list
      RWmaxobs144 =        144
      RWminobs144 =        144
       RWrmse144 =  .00412303
      RWmaxobs132 =        132
      RWminobs132 =        132
       RWrmse132 =  .00407538
```

```
      RWmaxobs120 =        120
      RWminobs120 =        120
       RWrmse120 =  .00406735
      RWmaxobs108 =        108
      RWminobs108 =        108
       RWrmse108 =  .00406403
      RWmaxobs96 =         96
      RWminobs96 =         96
        RWrmse96 =  .00419684
      RWmaxobs84 =         84
      RWminobs84 =         84
        RWrmse84 =  .00423362
      RWmaxobs72 =         72
      RWminobs72 =         72
        RWrmse72 =  .00429113
      RWmaxobs60 =         60
      RWminobs60 =         60
        RWrmse60 =  .00448591
      RWmaxobs48 =         48
      RWminobs48 =         48
        RWrmse48 =  .00478837

125 . /*
   > RWmaxobs108 = 108
   > RWminobs108 = 108
   > RWrmse108 = .00406403
   > */
126 .
127 . *6
128 . /*
   > RWmaxobs108 = 108
   > RWminobs108 = 108
   > RWrmse108 = .00388844
   > */
129 . scalar drop _all

130 . quietly forval w=156(12)156 {

131 . summ nobs // checking all had a full window

       Variable |        Obs        Mean    Std. Dev.       Min        Max
   -------------+---------------------------------------------------------
           nobs |        289    135.2561    32.98122         47        156

132 . *get error info for normal interval
133 . summ errsq

       Variable |        Obs        Mean    Std. Dev.       Min        Max
```

```
    -------------+---------------------------------------------------------
           errsq |        288     .000015    .0000471   1.10e-12   .0005431

. scalar rwrmse=r(mean)^0.5

. scalar list rwrmse
    rwrmse =  .00387308

. gen res=(d.lnflnonfarm-pred)
(73 missing values generated)

. _pctile res, percentile(2.5,97.5)

. return list

scalars:
                 r(r1) =  -.0074653569608927
                 r(r2) =  .0065394379198551

.
. *7
. predict temp if tin(2020m1,2020m1)
(option xb assumed; fitted values)
(360 missing values generated)

. replace pred=temp if tin(2020m1,2020m1)
(0 real changes made)

. drop temp

. gen pnonfarm=exp(l.lnflnonfarm+pred+(rwrmse^2)/2)
(72 missing values generated)

. gen ubound=exp(l.lnflnonfarm+pred+1.96*rwrmse+(rwrmse^2)/2)
(72 missing values generated)

. gen lbound=exp(l.lnflnonfarm+pred-1.96*rwrmse+(rwrmse^2)/2)
(72 missing values generated)

. list month pnonfarm lbound ubound if tin(2020m1,2020m1)

     +-----------------------------------------+
     | month   pnonfarm     lbound     ubound |
     |-----------------------------------------|
361. |     1   9001.077   8933.007   9069.667 |
     +-----------------------------------------+

```

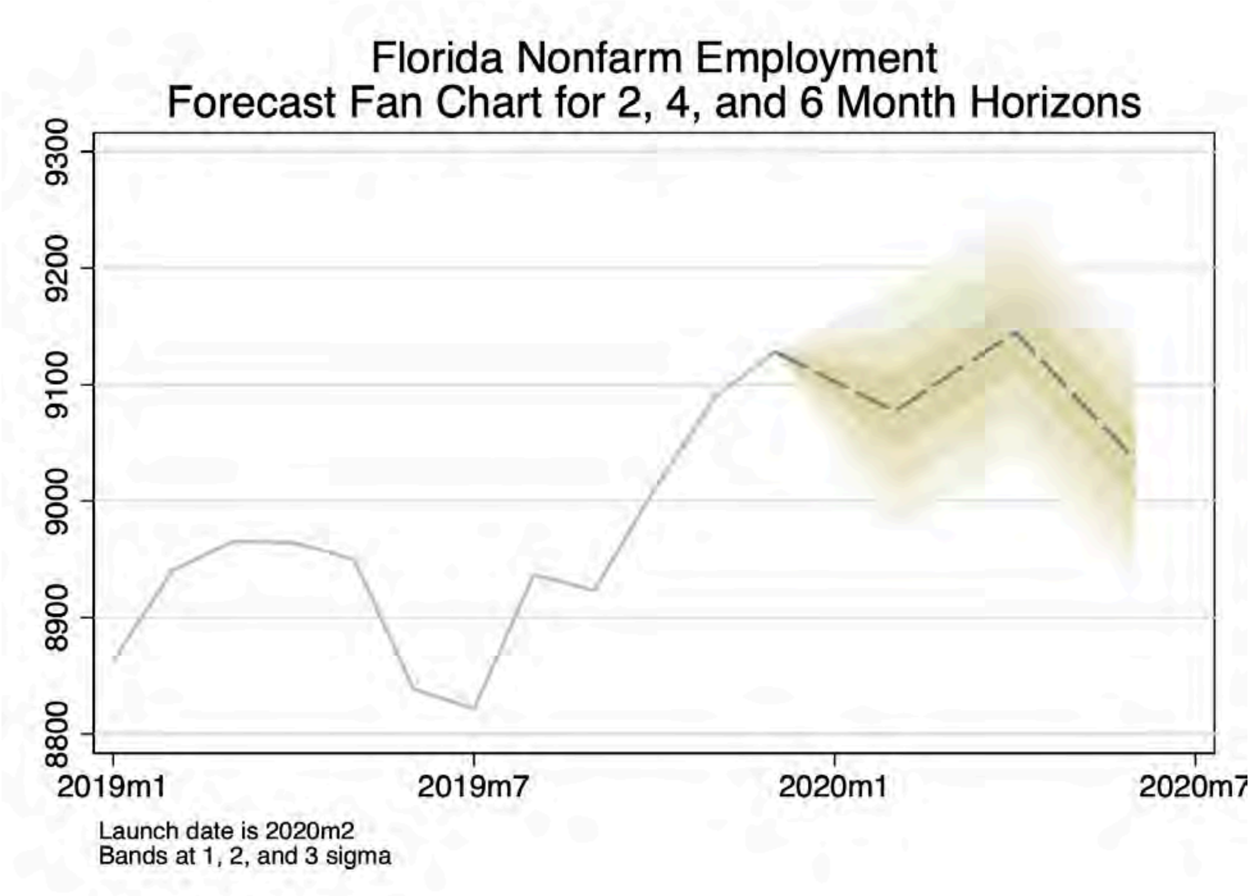
```
  148 . tsline pnonfarm lbound ubound fl_nonfarm if tin(2019m1,2020m1),
tline(2019m12
     > ) saving("Nonfarm_Normal", replace)
     (file Nonfarm_Normal.gph saved)

  149 . graph export "nonfarm_normal.png", replace
     (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets
     > /Problem Set 5/nonfarm_normal.png written in PNG format)

  150 .
  151 . *8
  152 . *Empirical
  153 . drop res

  154 . gen res=(d.lnflnonfarm-pred)
     (73 missing values generated)

  155 . gen expres=exp(res)
     (73 missing values generated)

  156 . summ expres

         Variable |        Obs        Mean    Std. Dev.       Min        Max
     -------------+---------------------------------------------------------
           expres |        288    .9997746    .0038735   .9777296   1.023579

  157 . scalar meanexpres=r(mean)

  158 . gen pnonfarme=exp(l.lnflnonfarm+pred)*meanexpres
     (72 missing values generated)

  159 . _pctile res, percentile(2.5,97.5)

  160 . return list

     scalars:
                      r(r1) =  -.0074653569608927
                      r(r2) =  .0065394379198551

  161 . gen lbounde=exp(l.lnflnonfarm+pred+r(r1))*meanexpres
     (72 missing values generated)

  162 . gen ubounde=exp(l.lnflnonfarm+pred+r(r2))*meanexpres
     (72 missing values generated)

  163 . list month pnonfarme lbounde ubounde if tin(2020m1,2020m1)

```

```
           +----------------------------------------+
           | month   pnonfa~e    lbounde    ubounde |
           |----------------------------------------|
      361. |     1   8998.981   8932.051   9058.022 |
           +----------------------------------------+

   164 . tsline pnonfarme lbounde ubounde fl_nonfarm if tin(2019m1,2020m1), ///
      >         tline(2019m12) saving("Nonfarm_Epirical", replace)
      (file Nonfarm_Epirical.gph saved)

   165 . graph export "nonfarm_empirical.png", replace
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
      > /Problem Set 5/nonfarm_empirical.png written in PNG format)

   166 .
   167 . *9
   168 . tsline pnonfarm pnonfarme fl_nonfarm lbound lbounde ubound ubounde ///
      >  if tin(2019m1,2020m1), tline(2019m12) saving("Normal_vs_Empirical", replace)
      (file Normal_vs_Empirical.gph saved)

   169 . graph export "normal_vs_empirical.png", replace
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
      > /Problem Set 5/normal_vs_empirical.png written in PNG format)

   170 .
```

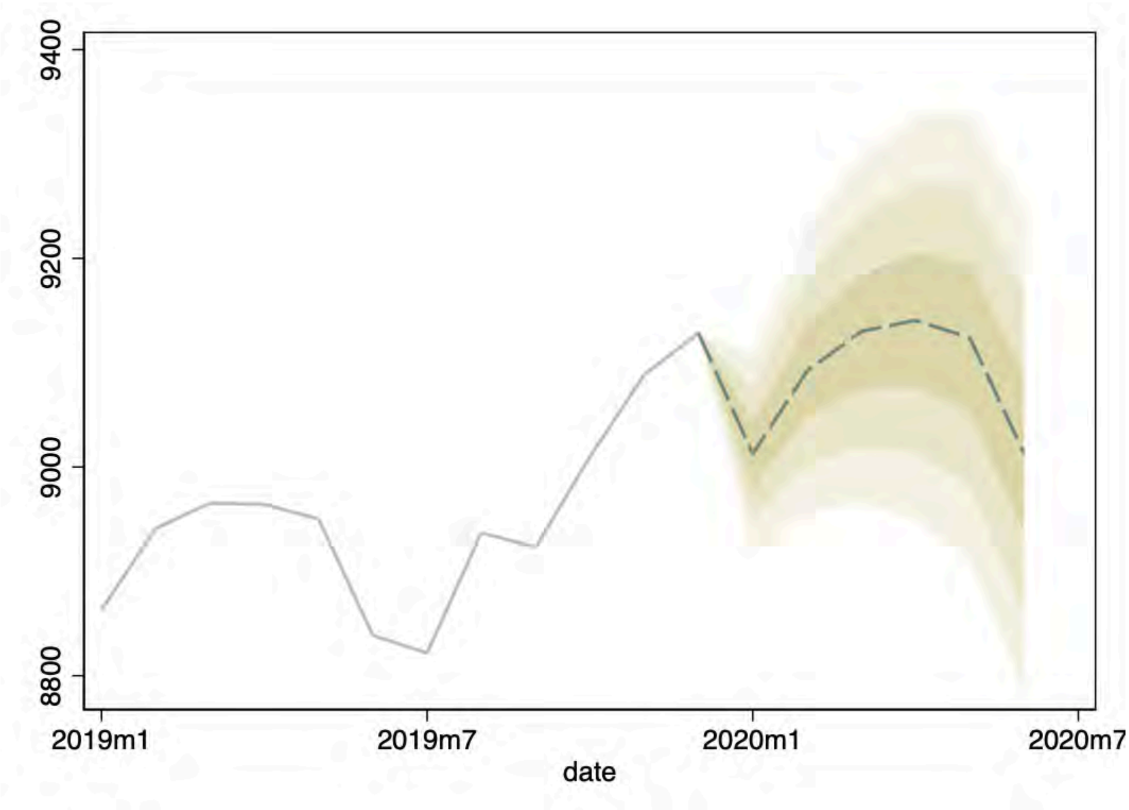
Problem Set 6

> Corrections are underlined. All corrections are from the official solutions.

ARDL



AR



Discussion

The ARDL model is much more "linear" than the AR model. While they both have the same general model, it looks like the AR model drew much more from the peak in 2019 than the ARDL did. I think this is because the AR model is only using nonfarm data so it's much more likely to replicate itself while the ARDL which uses more than one variable.

The-3 sigma interval corresponds to a 99.7% CI under the assumption the errors are normal, and such an interval would contain the actual outcome at least 89% of the time regardless of the underlying distribution. Hence, regardless of the actual distribution, it is unlikely employment will lie outside of the 3-sigma band.

Since the dynamic forecasting method is more subject to compounding errors, we would expect the forecast interval to be wider, but it is not. From the direct estimate of the 6 month change, the RMSE (for the log change) was 0.0115. From the dynamic model, 6 months out it was 0.0079 (adding the successive MSEs and taking the square root). This seems too low! What follows was not really required for full credit, but is worth thinking through carefully.

I ran the rolling window procedure for the entire sample period, using the OLS version of the model, to get the RMSE for forecasting the log change out one period. The RMSE was 0.0048. So, 6 months out, rmse should be about 0.0048*6^.5=0.0118. This is higher than 0.0115 from estimating the 6 month change directly, as it should be. And that does not even factor in what would happen if we ran through this 6 times dynamically with Rolling window, which would likely give a yet larger RMSE due to compounding the modeling error component, not just the residual component. The true RMSE would be even higher.

So, at the 6-month mark, the dynamic approach is underestimating the 95% ci upper bound by a factor BIGGER than exp(2*(0.0118-0.0079)) = 1.0078, and the lower interval lower by a factor smaller than 1/1.0078. With employment at the end of 2019 at 9.204 M, the CI upper bound should be at least 0.072M higher and the lower bound at least 0.072 lower.

Bottom line: is the CI is quite a bit too small with the dynamic technique! A quick dynamic forecast is fine if the stakes are low, and the fanchart is very suggestive. However, if the stakes are high, take the time to find the best model for the times of interest and use rolling window estimation and validation.

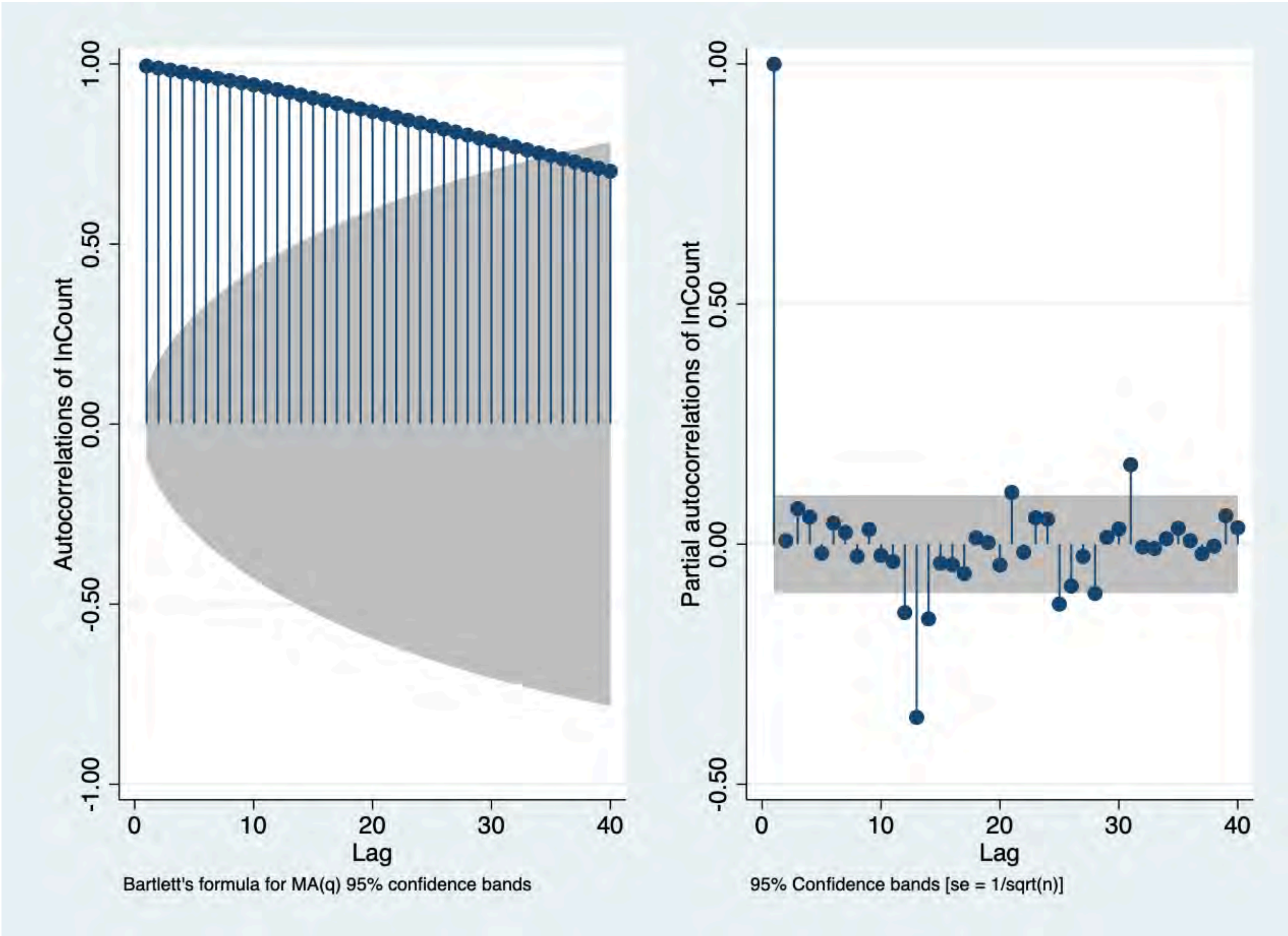
Final Project Deliverable 1

Summary statistics

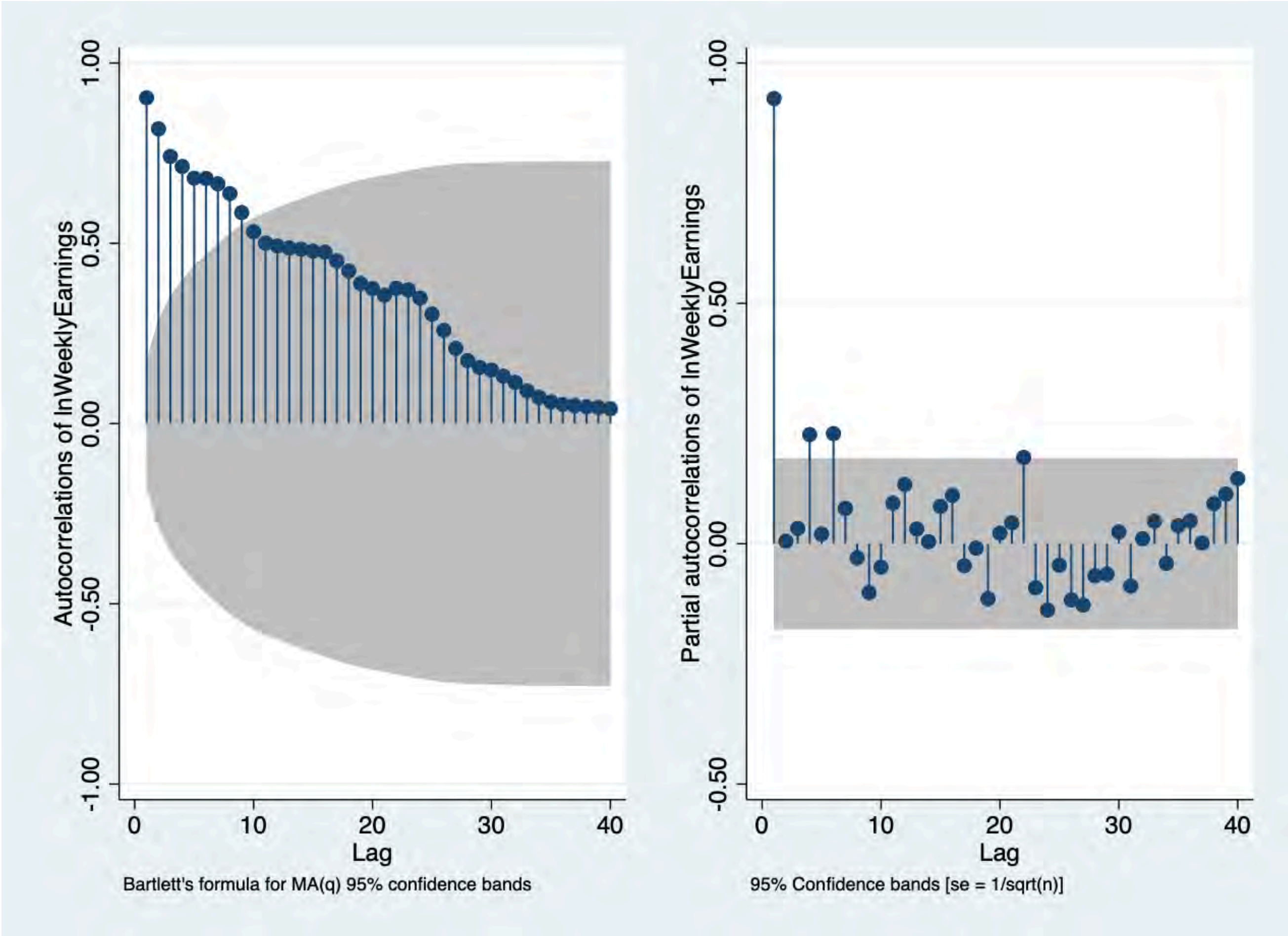
Variable	Obs	Mean	Std. Dev.	Min	Max
Count	374	14.18556	6.880684	5.3	28
WeekHours	122	36.86967	3.804193	28.3	45.8
HourlyEarn~s	122	19.70344	2.910126	15.01	24.6
WeeklyEarn~s	122	719.7972	84.82529	503.79	916.1
ServiceCount	374	10.40455	5.940013	3.9	22.8

Variable	Obs	Mean	Std. Dev.	Min	Max
lnCount	374	2.5174	.5398403	1.667707	3.332205
lnWeekHours	122	3.602049	.1041722	3.342862	3.824284
lnHourlyEa~s	122	2.969891	.148565	2.708717	3.202746
lnWeeklyEa~s	122	6.57194	.1198394	6.222159	6.820126
lnServiceC~t	374	2.16967	.5975865	1.360977	3.12676

AC and PAC



1.00
0.50
0.00
-0.50
-1.00
Autocorrelations of lnCount
0
10
20
30
40
Lag
Bartlett's formula for MA(q) 95% confidence bands
1.00
0.50
0.00
-0.50
Partial autocorrelations of lnCount
0
10
20
30
40
Lag
95% Confidence bands [se = 1/sqrt(n)]



> We should difference the data

Final Project Deliverable 2

Number of Employees

```
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(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
*/
```

I'm debating between the two. I'm also not sure if I should try and find another one that has more explanatory variables.

Weekly Earnings

```
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
*/
```

Code

```
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_txt2/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 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 differencen b

*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
*/

/*
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
*/

/*
gsreg dlnCount l1dlnCount l2dlnCount l3dlnCount l4dlnCount l5dlnCount l6dlnCount ///
  l7dlnCount l8dlnCount l9dlnCount l10dlnCount l11dlnCount l12dlnCount ///
  l24dlnCount l36dlnCount ///
  l1dlnWeekHours l2dlnWeekHours l3dlnWeekHours l4dlnWeekHours l5dlnWeekHours
l6dlnWeekHours ///
  l7dlnWeekHours l8dlnWeekHours l9dlnWeekHours l10dlnWeekHours l11dlnWeekHours
l12dlnWeekHours ///
  l24dlnWeekHours l36dlnWeekHours ///
  l1dlnHourlyEarnings l2dlnHourlyEarnings l3dlnHourlyEarnings l4dlnHourlyEarnings
///
  l5dlnHourlyEarnings l6dlnHourlyEarnings ///
  l7dlnHourlyEarnings l8dlnHourlyEarnings l9dlnHourlyEarnings l10dlnHourlyEarnings
///
  l11dlnHourlyEarnings l12dlnHourlyEarnings ///
  l24dlnHourlyEarnings l36dlnHourlyEarnings ///
  l1dlnWeeklyEarnings l2dlnWeeklyEarnings l3dlnWeeklyEarnings l4dlnWeeklyEarnings
///
  l5dlnWeeklyEarnings l6dlnWeeklyEarnings ///
  l7dlnWeeklyEarnings l8dlnWeeklyEarnings l9dlnWeeklyEarnings l10dlnWeeklyEarnings
///
  l11dlnWeeklyEarnings l12dlnWeeklyEarnings ///
  l24dlnWeeklyEarnings l36dlnWeeklyEarnings ///
  if tin(2011m1,2021m1), ///
  ncomb(1,4) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11) ///
  samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnCount_Full) replace
*/

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
*/


*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
*/

/*
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
*/

/*
gsreg dlnCount l1dlnCount l2dlnCount l3dlnCount l4dlnCount l5dlnCount l6dlnCount ///
  l7dlnCount l8dlnCount l9dlnCount l10dlnCount l11dlnCount l12dlnCount l24dlnCount ///
  l1dlnWeekHours l2dlnWeekHours l3dlnWeekHours l4dlnWeekHours l5dlnWeekHours l6dlnWeekHours ///
  l7dlnWeekHours l8dlnWeekHours l9dlnWeekHours l10dlnWeekHours l11dlnWeekHours l12dlnWeekHours l24dlnWeekHours ///
  l1dlnHourlyEarnings l2dlnHourlyEarnings l3dlnHourlyEarnings l4dlnHourlyEarnings ///
  l5dlnHourlyEarnings l6dlnHourlyEarnings ///
  l7dlnHourlyEarnings l8dlnHourlyEarnings l9dlnHourlyEarnings l10dlnHourlyEarnings ///
  l11dlnHourlyEarnings l12dlnHourlyEarnings l24dlnHourlyEarnings ///
  l1dlnWeeklyEarnings l2dlnWeeklyEarnings l3dlnWeeklyEarnings l4dlnWeeklyEarnings ///
  l5dlnWeeklyEarnings l6dlnWeeklyEarnings ///
  l7dlnWeeklyEarnings l8dlnWeeklyEarnings l9dlnWeeklyEarnings l10dlnWeeklyEarnings ///
  l11dlnWeeklyEarnings l12dlnWeeklyEarnings l24dlnWeeklyEarnings if tin(2011m1,2021m1), ///
  ncomb(1,4) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11) ///
  samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnWeeklyEarnings_Full) replace
*/

*log close
```

Log File

```
                                                        ___  ____  ____  ____  ____(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:   8 Apr 2021, 17:47:08

     1 . import delimited "TS2020_Final_Project_txt2/TS2020_Final_Project_Monthly.txt"
      (6 vars, 374 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 .
    16 . gen datec=date(date, "YMD")

    17 . gen Date=mofd(datec)

    18 . gen month=month(datec)
```

```
19 . format Date %tm

20 . tsset Date
        time variable:  Date, 1990m1 to 2021m2
                delta:  1 month

21 .
22 . gen lnCount = ln(Count)

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

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

25 . gen lnWeeklyEarnings = ln(WeeklyEarnings)
  (252 missing values generated)

26 . gen lnServiceCount = ln(ServiceCount)

27 .
28 . gen m1=0

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

30 . gen m2=0

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

32 . gen m3=0

33 . replace m3=1 if month==3
  (31 real changes made)

34 . gen m4=0

35 . replace m4=1 if month==4
  (31 real changes made)

36 . gen m5=0

37 . replace m5=1 if month==5
  (31 real changes made)

38 . gen m6=0

39 . replace m6=1 if month==6
  (31 real changes made)
```

```
 40 . gen m7=0

 41 . replace m7=1 if month==7
   (31 real changes made)

 42 . gen m8=0

 43 . replace m8=1 if month==8
   (31 real changes made)

 44 . gen m9=0

 45 . replace m9=1 if month==9
   (31 real changes made)

 46 . gen m10=0

 47 . replace m10=1 if month==10
   (31 real changes made)

 48 . gen m11=0

 49 . replace m11=1 if month==11
   (31 real changes made)

 50 . gen m12=0

 51 . replace m12=1 if month==12
   (31 real changes made)

 52 .
 53 . gen dlnCount=d.lnCount
   (1 missing value generated)

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

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

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

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

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

 59 . gen l6dlnCount=l6d.lnCount
```

```
  (7 missing values generated)

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

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

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

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

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

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

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

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

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

69 .
70 . gen dlnWeekHours=d.lnWeekHours
  (253 missing values generated)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

 86 .
 87 . gen dlnHourlyEarnings=d.lnHourlyEarnings
   (253 missing values generated)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

103 .
104 . gen dlnWeeklyEarnings=d.lnWeeklyEarnings
   (253 missing values generated)

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

106 . gen l2dlnWeeklyEarnings=l2d.lnWeeklyEarnings
   (255 missing values generated)

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

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

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

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

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

112 . gen l8dlnWeeklyEarnings=l8d.lnWeeklyEarnings
```

```
  (261 missing values generated)

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

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

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

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

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

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

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

120 .
121 . gen dlnServiceCount=d.lnServiceCount
  (1 missing value generated)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

  137 .
  138 . /*
     > 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.
     > */
  139 . /* Count and WeeklyEarnings */
  140 .
  141 . summ Count WeekHours HourlyEarnings WeeklyEarnings ServiceCount

         Variable |        Obs        Mean    Std. Dev.       Min        Max
     -------------+---------------------------------------------------------
            Count |        374    14.18556    6.880684        5.3         28
        WeekHours |        122    36.86967    3.804193       28.3       45.8
     HourlyEarn~s |        122    19.70344    2.910126      15.01       24.6
     WeeklyEarn~s |        122    719.7972    84.82529     503.79      916.1
     ServiceCount |        374    10.40455    5.940013        3.9       22.8

  142 . summ lnCount lnWeekHours lnHourlyEarnings lnWeeklyEarnings lnServiceCount

         Variable |        Obs        Mean    Std. Dev.       Min        Max
     -------------+---------------------------------------------------------
          lnCount |        374      2.5174    .5398403   1.667707   3.332205
      lnWeekHours |        122    3.602049    .1041722   3.342862   3.824284
     lnHourlyEa~s |        122    2.969891     .148565   2.708717   3.202746
     lnWeeklyEa~s |        122     6.57194    .1198394   6.222159   6.820126
```

```
    lnServiceC~t |        374     2.16967    .5975865   1.360977    3.12676

   143 .
   144 . ac lnCount, saving(lnCount_ac, replace)
      (file lnCount_ac.gph saved)

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

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

   147 . 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)

   148 . ** Probably need to difference
   149 .
   150 . ac lnWeeklyEarnings, saving(lnWeeklyEarnings_ac, replace)
      (file lnWeeklyEarnings_ac.gph saved)

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

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

   153 . 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)

   154 . ** Probably need to differencen b
   155 .
   156 . *starter models for count
   157 . *I used a pair plot to examine the rise and fall of variables with respect to
      >  each other
   158 . 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
      ------------------------------------------------------------------------------

   159 . scalar drop _all

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

   161 . 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

   162 . /*
      > RWmaxobs132 =        132
      > RWminobs132 =         12
      > RWrmse132 =   .0172128
      > */
   163 .
   164 . 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
      ------------------------------------------------------------------------------
-

   165 . scalar drop _all

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

   167 . 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

   168 . /*
      > RWmaxobs84 =         84
      > RWminobs84 =         23
      > RWrmse84 =  .01950911
      > */
   169 .
```

```
  170 . /*
      > gsreg dlnCount l1dlnCount l2dlnCount l3dlnCount l4dlnCount l5dlnCount
l6dlnCo
      > unt ///
      >         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
      > */
  171 .
  172 . *gsreg suggestions
  173 . 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
      -------------------------------------------------------------------------------

   174 . scalar drop _all

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

   176 . 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

   177 . /*
      > RWmaxobs144 =        144
      > RWminobs144 =         12
      > RWrmse144 =  .01824906
      > */
   178 .
   179 . 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
      -----------------------------------------------------------------------------

   180 . scalar drop _all

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

   182 . 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

   183 . /*
      > RWmaxobs144 =        144
      > RWminobs144 =         12
      > RWrmse144 =  .01777071
      > */
   184 .
   185 . /*
      > gsreg dlnCount l1dlnCount l2dlnCount l3dlnCount l4dlnCount l5dlnCount l6dlnCo
      > unt ///
      >         l7dlnCount l8dlnCount l9dlnCount l10dlnCount l11dlnCount l12dlnCount
      > ///
      >         l24dlnCount l36dlnCount ///
      >         l1dlnWeekHours l2dlnWeekHours l3dlnWeekHours l4dlnWeekHours l5dlnWeek
      > Hours l6dlnWeekHours ///
      >         l7dlnWeekHours l8dlnWeekHours l9dlnWeekHours l10dlnWeekHours l11dlnWe
      > ekHours l12dlnWeekHours ///
      >         l24dlnWeekHours l36dlnWeekHours ///
      >         l1dlnHourlyEarnings l2dlnHourlyEarnings l3dlnHourlyEarnings l4dlnHour
      > lyEarnings ///
      >         l5dlnHourlyEarnings l6dlnHourlyEarnings ///
      >         l7dlnHourlyEarnings l8dlnHourlyEarnings l9dlnHourlyEarnings l10dlnHou
      > rlyEarnings ///
      >         l11dlnHourlyEarnings l12dlnHourlyEarnings ///
      >         l24dlnHourlyEarnings l36dlnHourlyEarnings ///
      >         l1dlnWeeklyEarnings l2dlnWeeklyEarnings l3dlnWeeklyEarnings l4dlnWeek
      > lyEarnings ///
      >         l5dlnWeeklyEarnings l6dlnWeeklyEarnings ///
      >         l7dlnWeeklyEarnings l8dlnWeeklyEarnings l9dlnWeeklyEarnings l10dlnWee
      > klyEarnings ///
      >         l11dlnWeeklyEarnings l12dlnWeeklyEarnings ///
      >         l24dlnWeeklyEarnings l36dlnWeeklyEarnings ///
      >         if tin(2011m1,2021m1), ///
```

```
      >         ncomb(1,4) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
m11)
      > ///
      >         samesample nindex( -1 aic -1 bic -1 rmse_out)
results(gsreg_dlnCount_
      > Full) replace
      > */
   186 .
   187 . 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.
Interval]
      -------------+----------------------------------------------------------------
-
       lnWeekHours |
              L4D. |  -.0013847   .0384012    -0.04   0.971    -.0776005
.074831
              L9D. |   .0397686   .0385964     1.03   0.305    -.0368346
.1163718
                   |
      lnHourlyEa~s |
              L8D. |   -.039029   .0414024    -0.94   0.348    -.1212014
.0431433
                   |
                m1 |  -.0097045   .0078517    -1.24   0.219    -.0252879
.0058789
                m2 |   .0000949   .0079445     0.01   0.990    -.0156727
.0158626
                m3 |   -.004712   .0083585    -0.56   0.574    -.0213013
.0118773
                m4 |  -.0273667   .0081729    -3.35   0.001    -.0435876
-.0111459
                m5 |  -.0076836   .0081259    -0.95   0.347    -.0238112
.008444
```

```
               m6 |   -.020254   .0081465    -2.49   0.015    -.0364227   -.0040854
               m7 |  -.0130812   .0081852    -1.60   0.113    -.0293265    .0031642
               m8 |   .0041701   .0081051     0.51   0.608    -.0119164    .0202565
               m9 |  -.0089171   .0082764    -1.08   0.284    -.0253435    .0075093
              m10 |   .0153608   .0081153     1.89   0.061    -.0007459    .0314674
              m11 |   .0040463   .0079619     0.51   0.612    -.0117559    .0198485
            _cons |   .0094122   .0056462     1.67   0.099    -.0017939    .0206183
      ------------------------------------------------------------------------------

   188 . scalar drop _all

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

   190 . 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

   191 . /*
      > RWmaxobs12 =         12
      > RWminobs12 =          2
      > RWrmse12 =   .0176238
      > */
   192 .
```

```
  193 . /*----------------------------------------------------------------------
---
      > -*/
  194 .
  195 . *starter models for weekly earnings
  196 . reg d.lnWeeklyEarnings l1d.lnWeekHours ld.lnHourlyEarnings

            Source |       SS           df       MS      Number of obs   =
120
      -------------+----------------------------------   F(2, 117)       =
1.48
             Model |   .00720643         2  .003603215   Prob > F        =
0.2317
          Residual |  .284697808       117  .002433315   R-squared       =
0.0247
      -------------+----------------------------------   Adj R-squared   =
0.0080
             Total |  .291904237       119  .002452977   Root MSE        =
.04933

      ------------------------------------------------------------------------------
-
      D.           |
      lnWeeklyEa~s |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
Interval]
      -------------+----------------------------------------------------------------
-
       lnWeekHours |
               LD. |  -.1344969    .105932    -1.27   0.207    -.3442897
.0752959
                   |
      lnHourlyEa~s |
               LD. |   .0732899   .1167768     0.63   0.531    -.1579805
.3045603
                   |
             _cons |   .0016809   .0045103     0.37   0.710    -.0072515
.0106133
      ------------------------------------------------------------------------------
-

  197 . scalar drop _all

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

  199 . scalar list
      RWmaxobs84 =         84
      RWminobs84 =          2
        RWrmse84 =  .06184586
      RWmaxobs72 =         72
      RWminobs72 =          2
        RWrmse72 =  .06163593
```

```
      RWmaxobs60 =         60
      RWminobs60 =          2
        RWrmse60 =  .06145693
      RWmaxobs48 =         48
      RWminobs48 =          2
        RWrmse48 =  .06185207
      RWmaxobs36 =         36
      RWminobs36 =          2
        RWrmse36 =  .06201342
      RWmaxobs24 =         24
      RWminobs24 =          2
        RWrmse24 =  .06223845
      RWmaxobs12 =         12
      RWminobs12 =          2
        RWrmse12 =  .06581989

   200 . /*
      > RWmaxobs60 =         60
      > RWminobs60 =          2
      > RWrmse60 =  .06145693
      > */
   201 .
   202 . /*
      > gsreg dlnWeeklyEarnings l1dlnWeeklyEarnings l2dlnWeeklyEarnings l3dlnWeeklyEa
      > rnings ///
      >         l4dlnWeeklyEarnings l5dlnWeeklyEarnings l6dlnWeeklyEarnings ///
      >         l7dlnWeeklyEarnings l8dlnWeeklyEarnings l9dlnWeeklyEarnings l10dlnWee
      > klyEarnings ///
      >         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_dlnWeekly
      > Earnings) replace
      > */
   203 .
   204 . 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)      =      2.22
             Model |  .062824046        13  .004832619   Prob > F        =    0.0134
          Residual |  .221893915       102  .002175431   R-squared       =    0.2207
```

```
    -------------+----------------------------------   Adj R-squared   =  0.1213
           Total |  .284717961       115  .002475808   Root MSE        =  .04664

    ------------------------------------------------------------------------------
    D.           |
    lnWeeklyEa~s |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
    -------------+----------------------------------------------------------------
    lnWeeklyEa~s |
            L3D. |   -.230666   .0948354    -2.43   0.017    -.4187716   -.0425604
            L5D. |  -.1701094   .0948714    -1.79   0.076    -.3582864    .0180676
                 |
              m1 |  -.0129295   .0212711    -0.61   0.545    -.0551207    .0292616
              m2 |   .0230655   .0211959     1.09   0.279    -.0189765    .0651074
              m3 |  -.0120234   .0220335    -0.55   0.586    -.0557268      .03168
              m4 |   .0106776   .0218502     0.49   0.626    -.0326621    .0540173
              m5 |   .0379684   .0215915     1.76   0.082    -.0048583    .0807952
              m6 |   .0272323   .0215653     1.26   0.210    -.0155424    .0700069
              m7 |  -.0216364   .0213979    -1.01   0.314     -.064079    .0208063
              m8 |   .0154331   .0210082     0.73   0.464    -.0262366    .0571028
              m9 |   .0206819    .021546     0.96   0.339    -.0220544    .0634182
             m10 |   .0210709   .0217906     0.97   0.336    -.0221507    .0642924
             m11 |   .0089656   .0216561     0.41   0.680    -.0339891    .0519203
           _cons |  -.0075658   .0151177    -0.50   0.618    -.0375517    .0224201
    ------------------------------------------------------------------------------

 205 . scalar drop _all

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

 207 . scalar list
    RWmaxobs84 =         84
```

```
      RWminobs84 =          2
        RWrmse84 =  .06004448
      RWmaxobs72 =         72
      RWminobs72 =          2
        RWrmse72 =   .0605325
      RWmaxobs60 =         60
      RWminobs60 =          2
        RWrmse60 =  .06068574
      RWmaxobs48 =         48
      RWminobs48 =          2
        RWrmse48 =  .06037733
      RWmaxobs36 =         36
      RWminobs36 =          2
        RWrmse36 =  .06130591
      RWmaxobs24 =         24
      RWminobs24 =          2
        RWrmse24 =  .06544875
      RWmaxobs12 =         12
      RWminobs12 =          2
        RWrmse12 =  .07012904

   208 . /*
      > RWmaxobs84 =         84
      > RWminobs84 =          2
      > RWrmse84 =  .06004448
      > */
   209 .
   210 . reg d.lnWeeklyEarnings l3d.lnWeeklyEarnings l5d.lnWeeklyEarnings l7d.lnWeekly
      > Earnings

            Source |       SS           df       MS      Number of obs   =       114
      -------------+----------------------------------   F(3, 110)       =      4.04
             Model |  .027592174         3  .009197391   Prob > F        =    0.0091
          Residual |  .250484433       110  .002277131   R-squared       =    0.0992
      -------------+----------------------------------   Adj R-squared   =    0.0747
             Total |  .278076607       113  .002460855   Root MSE        =    .04772

      ------------------------------------------------------------------------------
      D.           |
      lnWeeklyEa~s |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
```

```
      lnWeeklyEa~s |
               L3D. |  -.2446256   .0905231    -2.70   0.008    -.4240211    -.0652301
               L5D. |  -.1957955   .0901769    -2.17   0.032    -.3745049    -.0170861
               L7D. |   .0649411   .0912448     0.71   0.478    -.1158846    .2457668
                    |
              _cons |   .0027599   .0044776     0.62   0.539    -.0061138    .0116335
      ------------------------------------------------------------------------------

   211 . scalar drop _all

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

   213 . scalar list
      RWmaxobs84 =         84
      RWminobs84 =          2
        RWrmse84 =  .05250414
      RWmaxobs72 =         72
      RWminobs72 =          2
        RWrmse72 =  .05277312
      RWmaxobs60 =         60
      RWminobs60 =          2
        RWrmse60 =  .05308846
      RWmaxobs48 =         48
      RWminobs48 =          2
        RWrmse48 =  .05299891
      RWmaxobs36 =         36
      RWminobs36 =          2
        RWrmse36 =  .05349229
      RWmaxobs24 =         24
      RWminobs24 =          2
        RWrmse24 =  .05283688
      RWmaxobs12 =         12
      RWminobs12 =          2
        RWrmse12 =  .06056213

   214 . /*
      > RWmaxobs84 =         84
      > RWminobs84 =          2
      > RWrmse84 =  .05250414
      > */
   215 .
   216 . /*
      > gsreg dlnCount l1dlnCount l2dlnCount l3dlnCount l4dlnCount l5dlnCount l6dlnCo
      > unt ///
```

```
      >         l7dlnCount l8dlnCount l9dlnCount l10dlnCount l11dlnCount l12dlnCount
      > l24dlnCount ///
      >         l1dlnWeekHours l2dlnWeekHours l3dlnWeekHours l4dlnWeekHours l5dlnWeek
      > Hours l6dlnWeekHours ///
      >         l7dlnWeekHours l8dlnWeekHours l9dlnWeekHours l10dlnWeekHours l11dlnWe
      > ekHours l12dlnWeekHours l24dlnWeekHours ///
      >         l1dlnHourlyEarnings l2dlnHourlyEarnings l3dlnHourlyEarnings l4dlnHour
      > lyEarnings ///
      >         l5dlnHourlyEarnings l6dlnHourlyEarnings ///
      >         l7dlnHourlyEarnings l8dlnHourlyEarnings l9dlnHourlyEarnings l10dlnHou
      > rlyEarnings ///
      >         l11dlnHourlyEarnings l12dlnHourlyEarnings l24dlnHourlyEarnings ///
      >         l1dlnWeeklyEarnings l2dlnWeeklyEarnings l3dlnWeeklyEarnings l4dlnWeek
      > lyEarnings ///
      >         l5dlnWeeklyEarnings l6dlnWeeklyEarnings ///
      >         l7dlnWeeklyEarnings l8dlnWeeklyEarnings l9dlnWeeklyEarnings l10dlnWee
      > klyEarnings ///
      >         l11dlnWeeklyEarnings l12dlnWeeklyEarnings l24dlnWeeklyEarnings if tin
      > (2011m1,2021m1), ///
      >         ncomb(1,4) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11)
      > ///
      >         samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnWeekly
      > Earnings_Full) replace
      > */
   217 .
   218 . log close
            name:  <unnamed>
             log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Probl
      > em Sets/Final Project/Final Project.smcl
        log type:  smcl
       closed on:   8 Apr 2021, 17:48:56
      ------------------------------------------------------------------------------
```

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

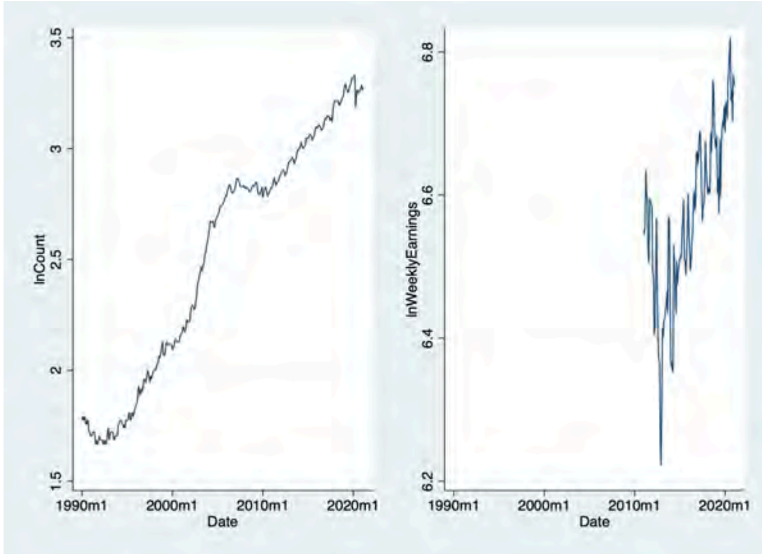
Of particular note in the non-transformed dat is that there is a wide variation in the number of private employment for The Villages has an ancredibly high range where the maximum value is more than five times greater than the minimum value. Meanwhile, hourly earnings have not increased by nearly the 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

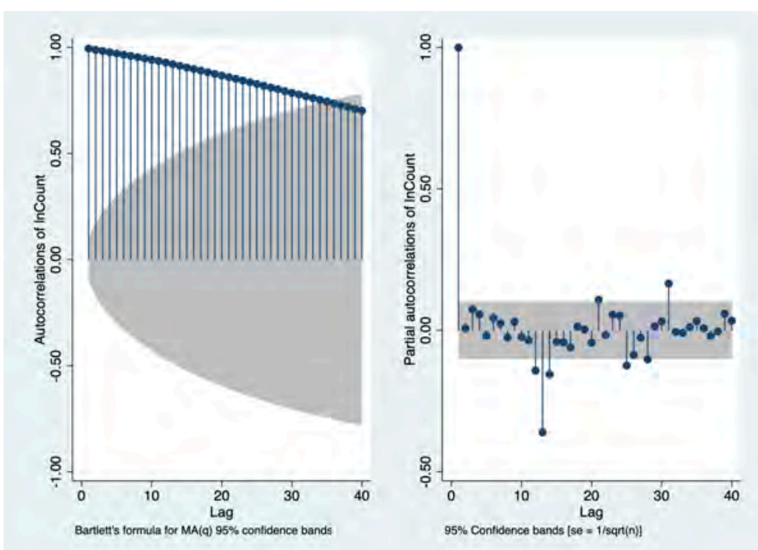
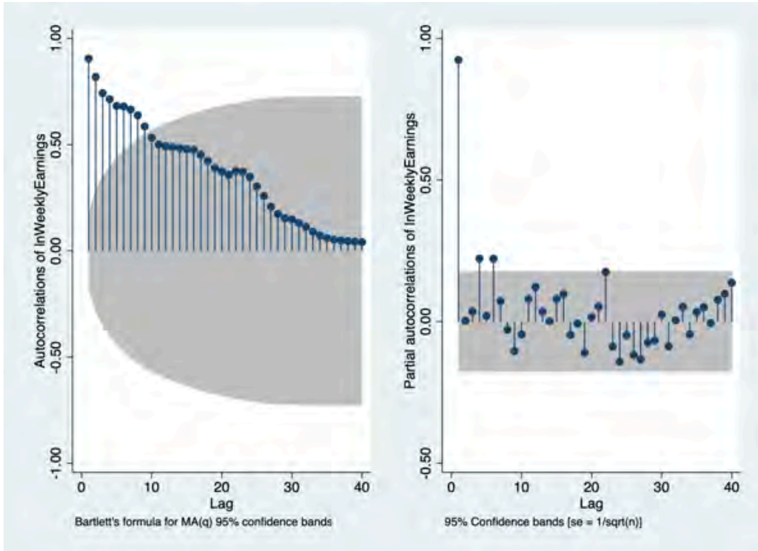


Figure 3 AC and PAC of lnWeeklyEarnings



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

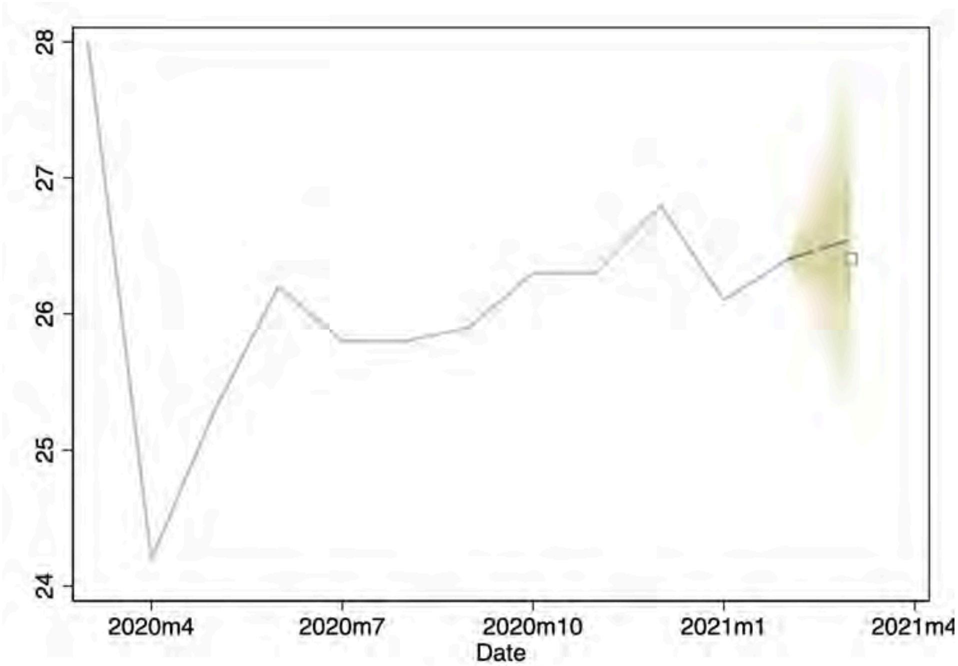
Final Results

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.

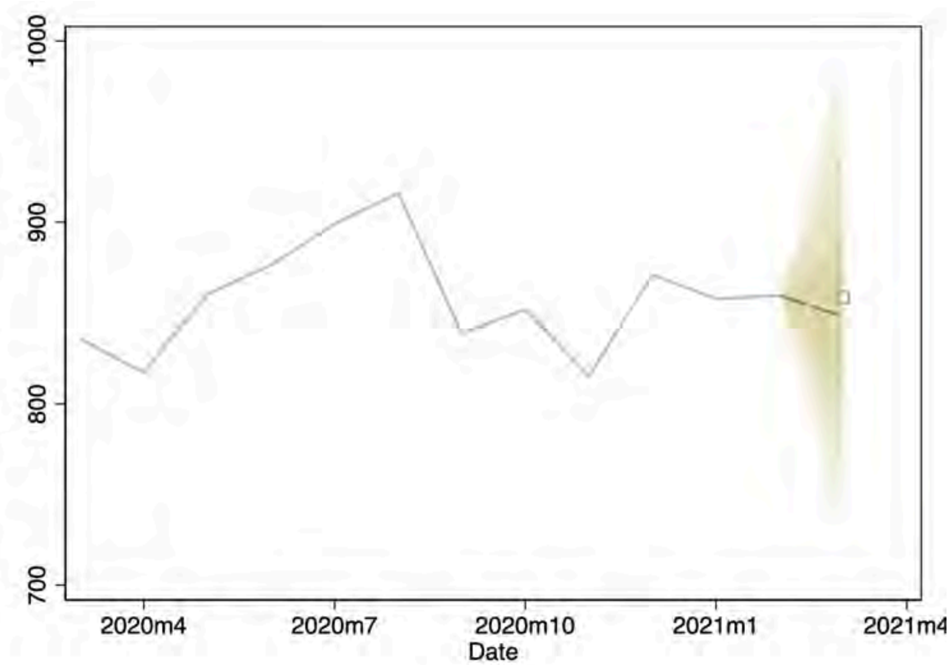
Figure 4 Fan Chart of Total Private Employment



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
```

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 .
```

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

    17 . gen Date=mofd(datec)

    18 . gen month=month(datec)

    19 . format Date %tm

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

    21 .
    22 . gen withMarchCount = Count

    23 . gen withMarchEarnings = WeeklyEarnings
      (252 missing values generated)

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

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

    26 .
    27 . gen lnCount = ln(Count)
      (1 missing value generated)

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

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

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

    31 . gen lnServiceCount = ln(ServiceCount)

    32 .
    33 . gen m1=0

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

    35 . gen m2=0

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

    37 . gen m3=0
```

```

  38 . replace m3=1 if month==3
    (32 real changes made)

  39 . gen m4=0

  40 . replace m4=1 if month==4
    (31 real changes made)

  41 . gen m5=0

  42 . replace m5=1 if month==5
    (31 real changes made)

  43 . gen m6=0

  44 . replace m6=1 if month==6
    (31 real changes made)

  45 . gen m7=0

  46 . replace m7=1 if month==7
    (31 real changes made)

  47 . gen m8=0

  48 . replace m8=1 if month==8
    (31 real changes made)

  49 . gen m9=0

  50 . replace m9=1 if month==9
    (31 real changes made)

  51 . gen m10=0

  52 . replace m10=1 if month==10
    (31 real changes made)

  53 . gen m11=0

  54 . replace m11=1 if month==11
    (31 real changes made)

  55 . gen m12=0

  56 . replace m12=1 if month==12
    (31 real changes made)

  57 .
  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)
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

 91 .
 92 . gen dlnHourlyEarnings=d.lnHourlyEarnings
   (253 missing values generated)

 93 . 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)

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

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

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

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

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

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

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

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

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

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

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

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

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

125 .
126 . gen dlnServiceCount=d.lnServiceCount
   (1 missing value generated)

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

128 . 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
      ------------------------------------------------------------------------------
--
      ---
```

```
  223 . scalar drop _all

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

  225 . scalar list
     RWmaxobs84 =         84
     RWminobs84 =          2
       RWrmse84 =  .06183191
     RWmaxobs72 =         72
     RWminobs72 =          2
       RWrmse72 =  .06162109
     RWmaxobs60 =         60
     RWminobs60 =          2
       RWrmse60 =  .06144232
     RWmaxobs48 =         48
     RWminobs48 =          2
       RWrmse48 =   .0618403
     RWmaxobs36 =         36
     RWminobs36 =          2
       RWrmse36 =  .06201409
     RWmaxobs24 =         24
     RWminobs24 =          2
       RWrmse24 =  .06224974
     RWmaxobs12 =         12
     RWminobs12 =          2
       RWrmse12 =  .06583082

  226 . /*
     > RWmaxobs60 =         60
     > RWminobs60 =          2
     > RWrmse60 =  .06145693
     > */
  227 .
  228 . quietly gsreg dlnWeeklyEarnings l1dlnWeeklyEarnings l2dlnWeeklyEarnings
l3dln
     > WeeklyEarnings ///
     >         l4dlnWeeklyEarnings l5dlnWeeklyEarnings l6dlnWeeklyEarnings ///
     >         l7dlnWeeklyEarnings l8dlnWeeklyEarnings l9dlnWeeklyEarnings
l10dlnWee
     > klyEarnings ///
     >         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_dlnWeekly
     > Earnings) replace

  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
```

```

  234 . /*
      > RWmaxobs84 =         84
      > RWminobs84 =          2
      > RWrmse84 =  .06004448
      > */
  235 .
  236 . reg d.lnWeeklyEarnings l3d.lnWeeklyEarnings l5d.lnWeeklyEarnings
l7d.lnWeekly
      > Earnings

            Source |       SS           df       MS      Number of obs   =
114
      -------------+----------------------------------   F(3, 110)       =
3.85
             Model |  .026396014         3  .008798671   Prob > F        =
0.0115
          Residual |  .251283445       110  .002284395   R-squared       =
0.0951
      -------------+----------------------------------   Adj R-squared   =
0.0704
             Total |  .277679459       113   .00245734   Root MSE        =
.0478

      ------------------------------------------------------------------------------
--
      ---
      D.                |
      lnWeeklyEarnings |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
Interv
      > al]
      ----------------+-------------------------------------------------------------
--
      ---
      lnWeeklyEarnings |
                  L3D. |  -.2408947   .0906673    -2.66   0.009    -.4205761
-.0612
      > 133
                  L5D. |  -.1892527   .0903206    -2.10   0.038    -.3682468
-.0102
      > 585
                  L7D. |   .0639647   .0913902     0.70   0.485    -.1171492
.2450
      > 786
                       |
                 _cons |   .0025724   .0044848     0.57   0.567    -.0063154
.0114
      > 602
      ------------------------------------------------------------------------------
--
      ---
```

```
  237 . scalar drop _all

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

  239 . scalar list
     RWmaxobs84 =         84
     RWminobs84 =          2
       RWrmse84 =  .05259823
     RWmaxobs72 =         72
     RWminobs72 =          2
       RWrmse72 =  .05283772
     RWmaxobs60 =         60
     RWminobs60 =          2
       RWrmse60 =  .05314168
     RWmaxobs48 =         48
     RWminobs48 =          2
       RWrmse48 =   .0530381
     RWmaxobs36 =         36
     RWminobs36 =          2
       RWrmse36 =  .05353125
     RWmaxobs24 =         24
     RWminobs24 =          2
       RWrmse24 =  .05282122
     RWmaxobs12 =         12
     RWminobs12 =          2
       RWrmse12 =  .06036464

  240 . /*
     > RWmaxobs84 =         84
     > RWminobs84 =          2
     > RWrmse84 =  .05250414
     > */
  241 .
  242 . drop pflcount ub1 ub2 ub3 lb1 lb2 lb3

  243 .
  244 . scalar rwrmse = .05250414

  245 . reg d.lnWeeklyEarnings l(3,5,7)d.lnWeeklyEarnings if tin(,2021m2)

        Source |       SS           df       MS      Number of obs   =
114
  -------------+----------------------------------   F(3, 110)       =
3.85
         Model |  .026396014         3  .008798671   Prob > F        =
0.0115
      Residual |  .251283445       110  .002284395   R-squared       =
0.0951
  -------------+----------------------------------   Adj R-squared   =
0.0704
```

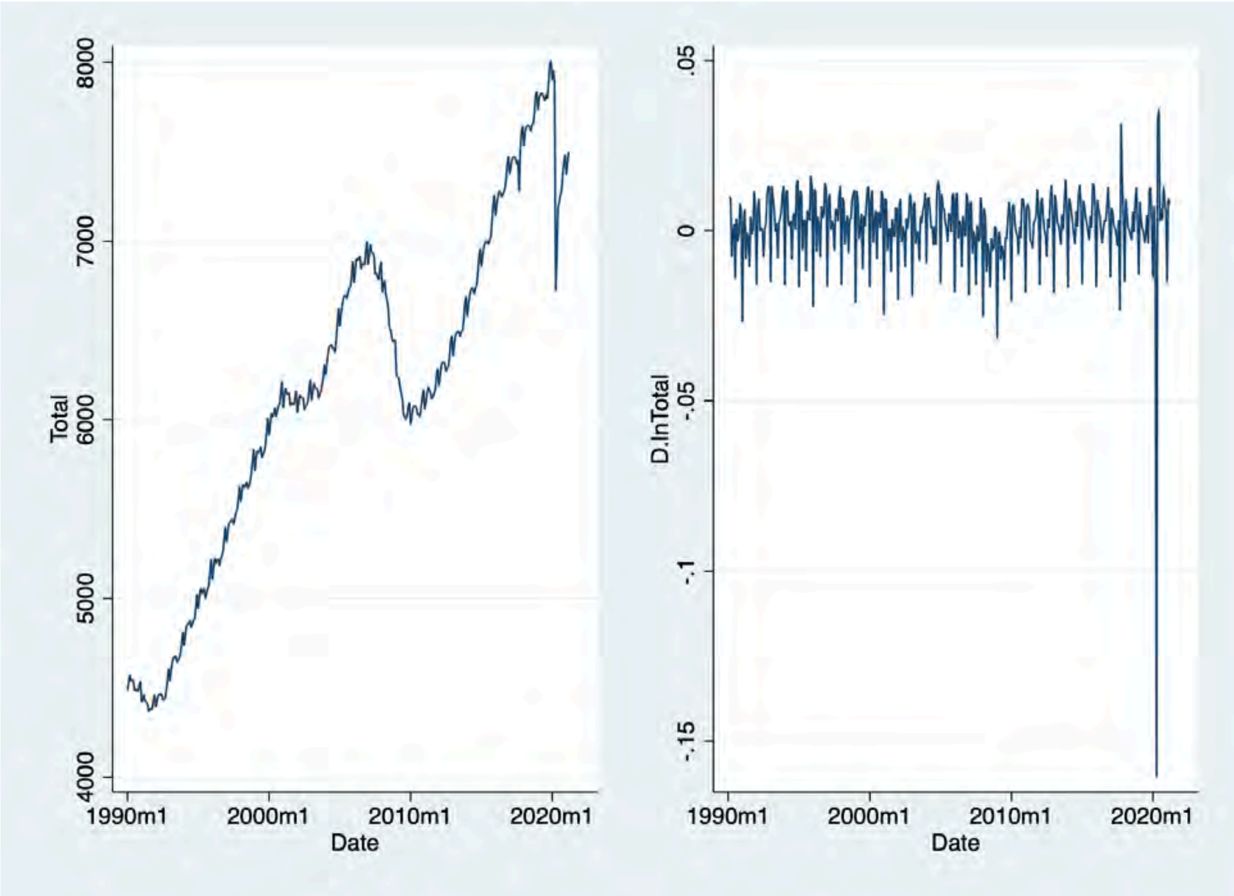
```
            Total |  .277679459       113   .00245734   Root MSE        =    .0478

      ---------------------------------------------------------------------------
      ---
      D.                |
      lnWeeklyEarnings |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interv
      > al]
      -----------------+---------------------------------------------------------
      ---
      lnWeeklyEarnings |
                  L3D. |  -.2408947   .0906673    -2.66   0.009    -.4205761    -.0612
      > 133
                  L5D. |  -.1892527   .0903206    -2.10   0.038    -.3682468    -.0102
      > 585
                  L7D. |   .0639647   .0913902     0.70   0.485    -.1171492     .2450
      > 786
                       |
                 _cons |   .0025724   .0044848     0.57   0.567    -.0063154     .0114
      > 602
      ---------------------------------------------------------------------------
      ---

   246 . predict pd
      (option xb assumed; fitted values)
      (260 missing values generated)

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

   248 . gen ub1=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+1*rwrmse) if Date==tm(202
      > 1m3)
      (374 missing values generated)

   249 . gen lb1=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-1*rwrmse) if Date==tm(202
      > 1m3)
      (374 missing values generated)

   250 . gen ub2=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+2*rwrmse) if Date==tm(202
```

```
      > 1m3)
      (374 missing values generated)

   251 . gen lb2=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-2*rwrmse) if Date==tm(202
      > 1m3)
      (374 missing values generated)

   252 . gen ub3=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd+3*rwrmse) if Date==tm(202
      > 1m3)
      (374 missing values generated)

   253 . gen lb3=exp((rwrmse^2)/2)*exp(l.lnWeeklyEarnings+pd-3*rwrmse) if Date==tm(202
      > 1m3)
      (374 missing values generated)

   254 . drop pd

   255 .
   256 . replace pflcount=WeeklyEarnings if Date==tm(2021m2)
      (1 real change made)

   257 . replace ub1=WeeklyEarnings if Date==tm(2021m2)
      (1 real change made)

   258 . replace ub2=WeeklyEarnings if Date==tm(2021m2)
      (1 real change made)

   259 . replace ub3=WeeklyEarnings if Date==tm(2021m2)
      (1 real change made)

   260 . replace lb1=WeeklyEarnings if Date==tm(2021m2)
      (1 real change made)

   261 . replace lb2=WeeklyEarnings if Date==tm(2021m2)
      (1 real change made)

   262 . replace lb3=WeeklyEarnings if Date==tm(2021m2)
      (1 real change made)

   263 .
   264 . 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)
lege
    > nd(off)

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

 266 .
 267 . log close
          name:  <unnamed>
           log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Probl
    > em Sets/Final Project/Final Project.smcl
      log type:  smcl
     closed on:  19 Apr 2021, 21:08:41
    -----------------------------------------------------------------------------
--
```

Part 1

Time Series Plots

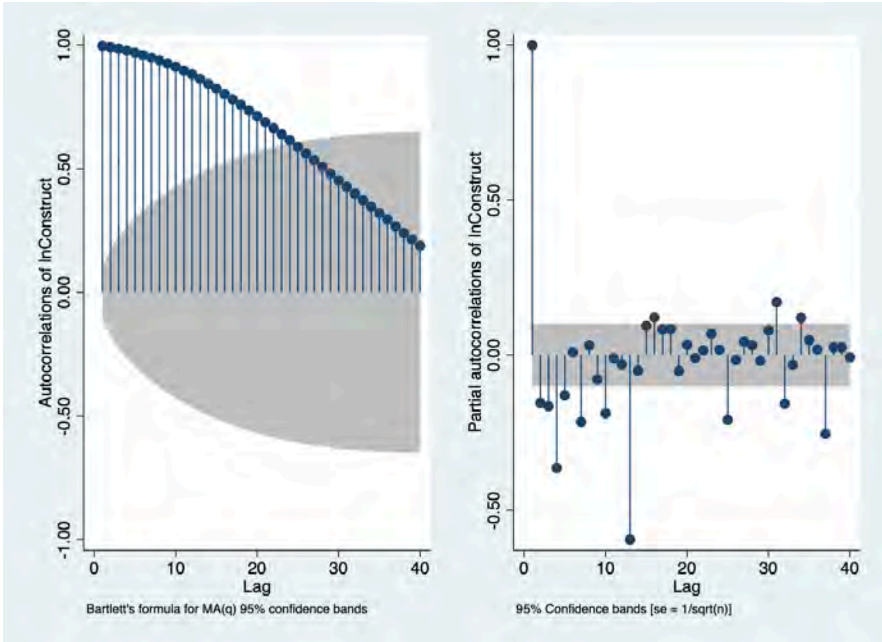


No Logs

You do not want to take logs because that adjusts the distance between values and normalizes the data. Without the distances between data being preserved, there is no good way to measure proportions unless the log transformed data has identical transforms performed (the data going in would have to be the same).

Persistent Data

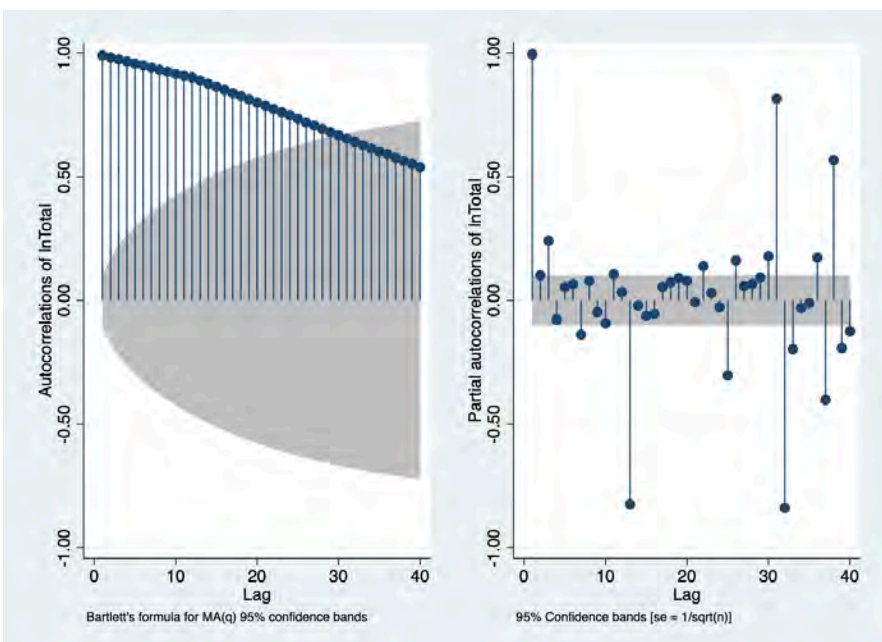
Construct



Both first lags are high which means we should difference.

Test	Result
MacKinnon approximate p-value for Z(t)	0.9796

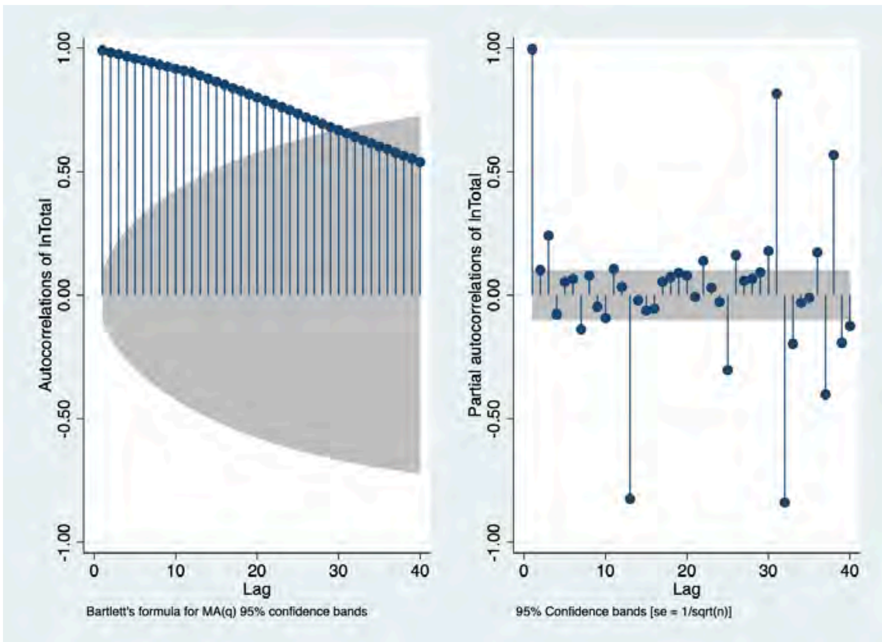
Manufacture



Both first lags are high which means we should difference.

Test	Result
MacKinnon approximate p-value for Z(t)	0.9963

Total

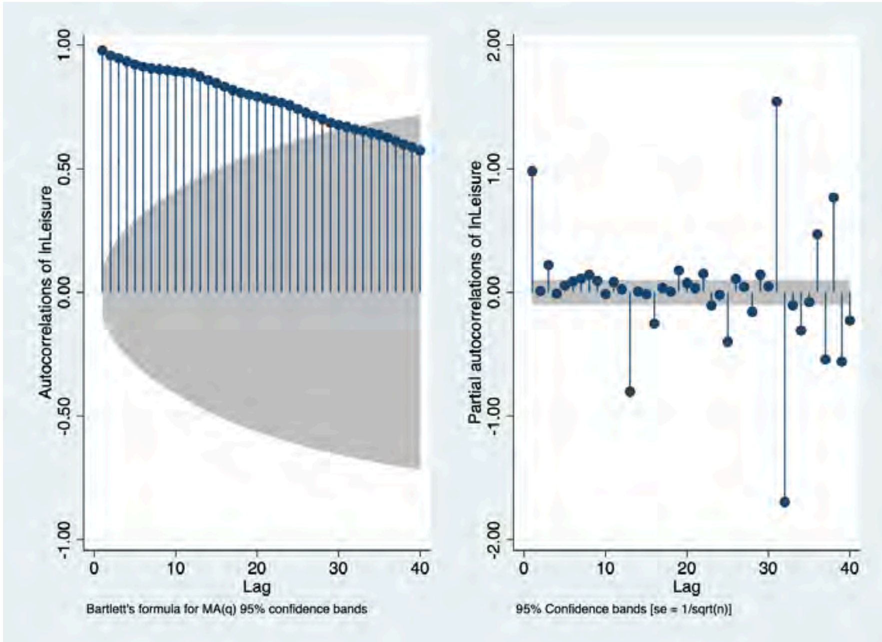


Both first lags are high which means we should difference.

Test	Result
MacKinnon approximate p-value for Z(t)	0.6570

Even though the dickey fuller test says we do not need to difference for construct, manufacture, and total, I'm still going to because ~~I already wrote everything with it differenced~~ the lags outside the confidence intervals in the PAC quite far apart and I can't imagine that there is really that much of an impact on current employment by employment from 30+ months ago. The only ones I can maybe understand are the 12 and 24 moth lags because employers will use historical data when hiring seasonal employees.

Leisure



Both first lags are high which means we should difference.

Test	Result
MacKinnon approximate p-value for Z(t)	0.0005

The p value for leisure is below .05 so I don't need an excuse to difference this time. However, it must be noted that lags 31 and 32 are extremely high.

Serial Correlation

Because not all lags are within the 95% confidence interval, we should worry at least a little bit about serial correlation. However, as I suggested earlier, the only lags that could have a true impact are the 12th and 24th lags and so I am not worried about serial correlation.

Dropping Lags

	Newey-West					
D.lnTotal	Coef.	Std. Err.	t	P>t	[95% Conf.	Interval]
Construct						
D1.	.0004252	.0000809	5.26	0.000	.0002661	.0005842
LD.	.0000247	.000044	0.56	0.574	-.0000617	.0001112
L2D.	.0000613	.0000403	1.52	0.129	-.0000179	.0001405
L3D.	-.0000674	.0000577	-1.17	0.244	-.000181	.0000462
L12D.	-.000035	.0000641	-0.55	0.585	-.000161	.000091
L24D.	.0000494	.0000684	0.72	0.470	-.0000851	.0001839
Manufacture						
D1.	.0011938	.0001924	6.21	0.000	.0008154	.0015723
LD.	-.0002374	.0001386	-1.71	0.088	-.00051	.0000353
L2D.	-.000336	.0000718	-4.68	0.000	-.0004773	-.0001948
L3D.	-.0002689	.0001059	-2.54	0.012	-.0004772	-.0000606
L12D.	.0003308	.0001381	2.39	0.017	.000059	.0006025
L24D.	.0002403	.0001413	1.70	0.090	-.0000376	.0005182
Leisure						
D1.	.0002134	8.38e-06	25.47	0.000	.0001969	.0002299
LD.	5.34e-06	9.19e-06	0.58	0.561	-.0000127	.0000234
L2D.	-2.42e-06	.0000172	-0.14	0.888	-.0000364	.0000315
L3D.	-8.76e-06	.0000174	-0.50	0.614	-.0000429	.0000254
L12D.	.0000629	.0000361	1.74	0.082	-8.04e-06	.0001339
L24D.	-.000016	.0000428	-0.37	0.708	-.0001002	.0000681
_cons	.0011957	.0003664	3.26	0.001	.000475	.0019165

The Newey-West test suggests we should drop lags 12 and 24.

Test Number	Variable		Test Value
(1)	L24D.Construct	=	0
(2)	L12D.Leisure	=	0
(3)	L24D.Leisure	=	0
(4)	L12D.Manufacture	=	0
(5)	L24D.Manufacture	=	0

`testparm` suggests that there is no interaction between the 12th and 24th lags of each construct, leisure, and manufacture.

Equal Effects

Interaction	Prob > F
Construct ~ Manufacture	0.0161
Construct ~ Leisure	0.0008
Manufacture ~ Leisure	0.0301

The p-value for all three is less than 0.05 which means we can reject the null hypothesis and accept the alternative that the impacts of each variable are not equal.

Part 2

Differencing, Log Transforms, and Month Dummies

Differencing

See above AC and PAC charts in Part 1.

Log Transforms

Log transforms make the data not have any values less than zero and forces the data into a normal distribution. It also transforms the data so it has proportional changes rather than absolute changes so that any changes over time can be reported as a percent change.

Month Dummies

There's not any reason to not include month dummies. If your data is monthly or any other form of seasonal, it will help your models because they're now identified to a particular season. If your data isn't seasonal, they won't have any effect.

Content Knowledge and Model Searches

Content Knowledge

Content knowledge can speed up the model selection process because you may already have an idea of what variables or lags have an effect on the dependent variable. For example, hourly wages and hours scheduled per week are probably a very good indication of monthly wages.

GSREG

Global search regression takes all the variables you feed it and runs a regression for any combination of the variables. This is a powerful tool to fine-tune your models, but without filtering the variables through content knowledge, it could take a very long time to run. Rather than just taking the highest scoring model, you should then examine common features of the highest scoring models on the basis of AIC, BIC, and out of sample root mean square error, and choose the most parsimonious one.

What's wrong with *stepwise* model selection?

It's prone to over fitting because it has bad predictive properties. Instead you should use out of sample fitting because it protects against over fitting. Over fitting is caused by dropping the most insignifcant each step which may include variables that should be included in the model but are not relevant on their own.

Choosing Models

Model Type	Model	AIC	BIC	Root Mean Squared Errors
AR only Lags 1-3 Month dummies	`reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	-2382.725	-2323.982	.0129605
AR only Lags 1-3,12,24 Month dummies	`reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	-2236.218	-2170.633	.01300797
ARDL Lags 1-3 Month dummies	`reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	-2393.008	-2299.019	.01611154
ARDL Lags 1-3,12,24 Month dummies	`reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	-2230.781	-2115.043	.01713897

Which are the best two and why?

Model 1 has the lowest root mean squared error and model 3 has the lowest AIC and BIC. I'm also inclined to believe these are the better ones because they don't include lags 12 and 24 which is a long time for subcomponents of the total employment variable to have an effect on the total employment variable.

Rolling Window

Model 1

`reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`

Value Type	Value
RWmaxobs12	12
RWminobs12	12
RWrmse12	.0132376

A window width of 12 had the lowest RWrmse. I thought that maybe a smaller window width would be better because the lags did not include lag 12 but I was wrong. Besides 12 months, 6 months had the second lowest.

Model 3

```
reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure
l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
```

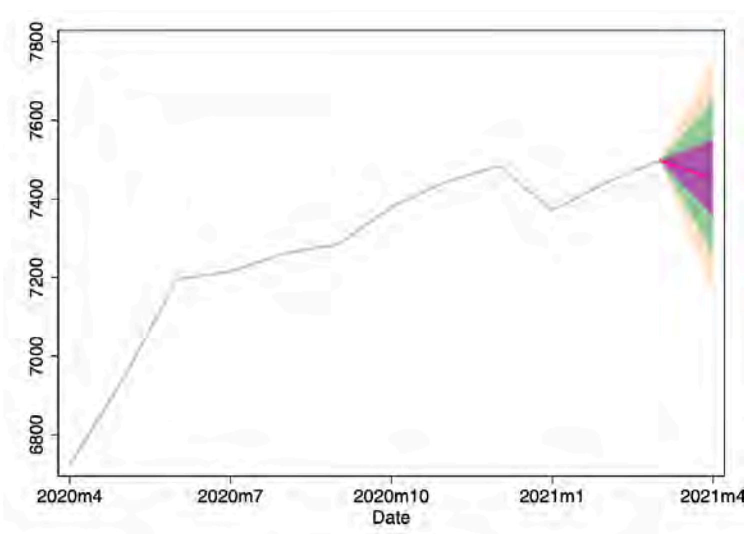
Value Type	Value
RWmaxobs12	12
RWminobs12	12
RWrmse12	.0132376

A window width of 12 had the lowest RWrmse. After my failure in model 1, I tried again hoping for better results. A window width of 12 is still the best.

Ultimately, I'm going to choose model 1 because it is autoregressive and that is what makes ARIMA work and without ARIMA I could not make my pretty fan charts. They have the same RWrmse anyways so I can't imagine the extra variables have too big a difference. And

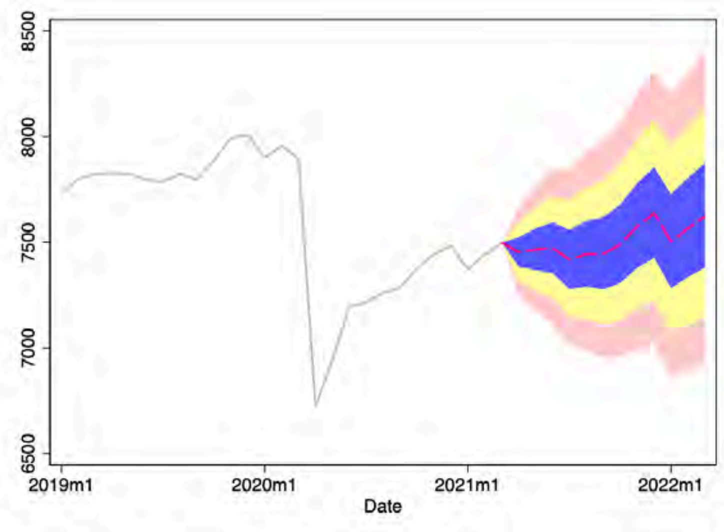
Forecasting

One month ahead



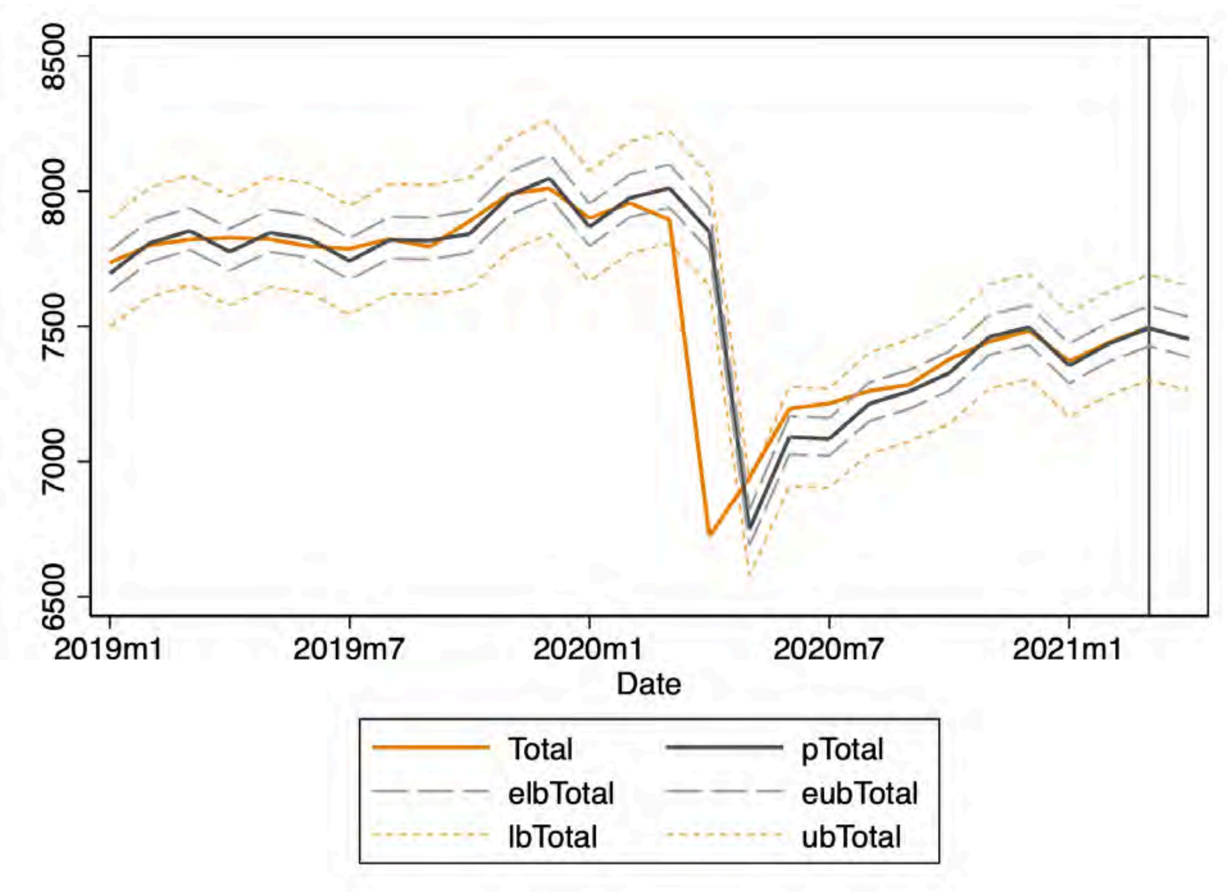
> Sorry for the bad colors. Hailey peer pressured me into it.

One year out

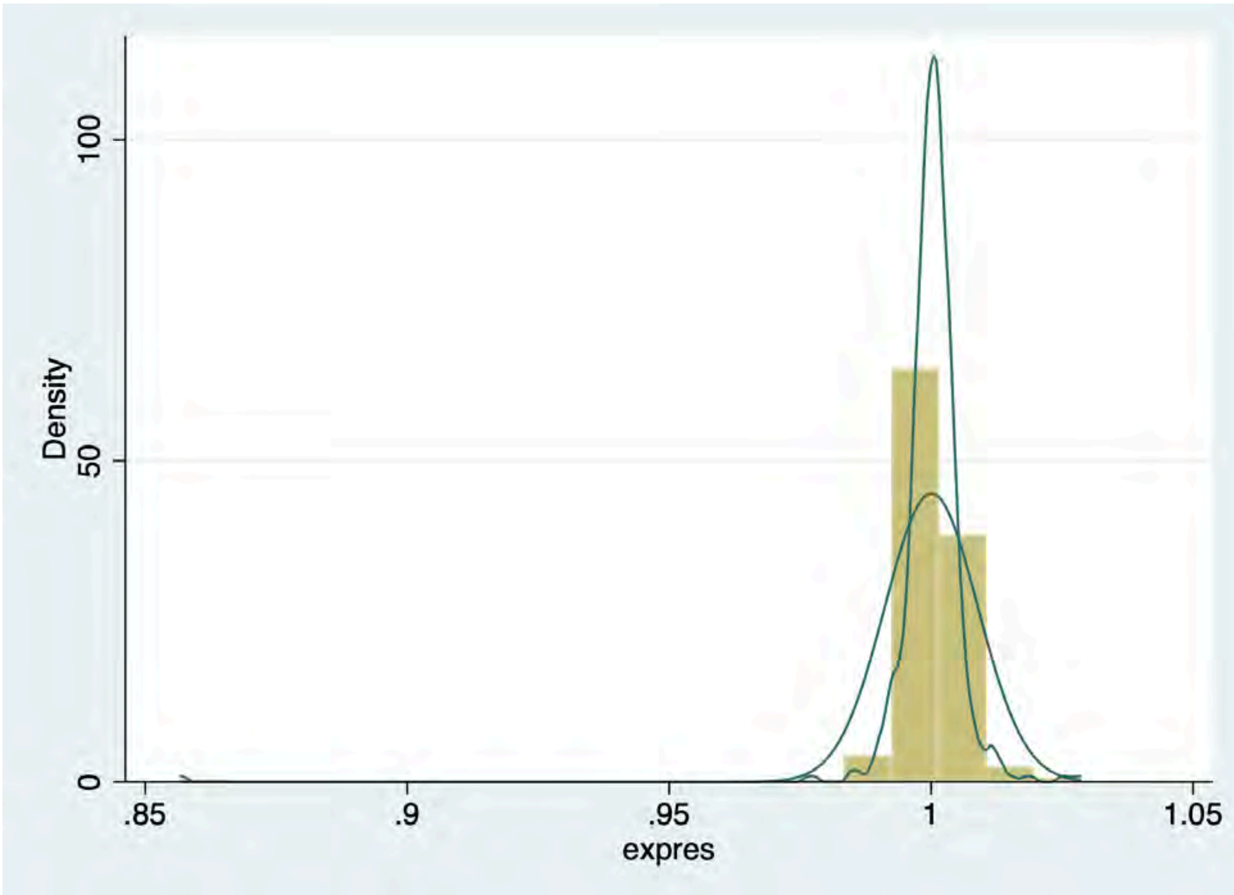


> Ditto my earlier comment on the colors :)

Forecast Evaluation



The prediction is pretty good and stays well within bounds (as it should). The only real mis-step is around the pandemic becuse the model could not have forseen such an event. Even so, once it did hit, the model stuck the landing and continued to forecast quite will to the present.



Residuals are good. Nice and normal.

Chebyshev

I think I have a basic understanding of Chebyshev. It's just a super general way of estimating what X portion of your population can be X distance away from the mean. In theory, this should hold even if it's not normal. That second part's key because it means we can use it on non-log transformed data.

Do File

```
clear
set more off

cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Final Exam"
log using "Final Exam.smcl", replace
import delimited "SP21Final.csv"
```

```

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

gen lnConstruct = ln(construct)
gen lnLeisure = ln(leisure)
gen lnManufacture = ln(manufacture)
gen lnTotal = ln(total)

gen Total = total
gen Construct = construct
gen Leisure = leisure
gen Manufacture = manufacture

/*
gen withMarchTotal = Total
replace Total=. if tin(2021m3,)
*/

tsset Date
tsappend, add(12)
replace month=month(dofm(Date))

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
```

```
summ construct leisure manufacture total
summ lnConstruct lnLeisure lnManufacture lnTotal

tsline lnConstruct lnLeisure, saving(lnConstructLeisure_tsline.gph, replace)
tsline lnManufacture, saving(lnManufacture_tsline.gph, replace)
graph combine lnConstructLeisure_tsline.gph lnManufacture_tsline.gph, ///
  saving(lnConstructLeisure, replace)
graph export "lnConstructLeisure-Manufacture_tsline.png", replace

tsline lnTotal, saving(lnTotal_tsline.gph, replace)
tsline d.lnTotal, saving(dlnTotal_tsline.gph, replace)
graph combine Total_tsline.gph dlnTotal_tsline.gph, saving(lnTotal-Total, replace)
graph export "lnTotal-dlnTotal_tsline.png", replace

ac lnTotal, saving(lnTotal_ac, replace)
pac lnTotal, saving(lnTotal_pac, replace)
graph combine lnTotal_ac.gph lnTotal_pac.gph, saving(lnTotal_ac_pac, replace)
graph export "lnTotal_ac_pac.png", replace
dfuller lnTotal, trend regress

ac lnConstruct, saving(lnConstruct_ac, replace)
pac lnConstruct, saving(lnConstruct_pac, replace)
graph combine lnConstruct_ac.gph lnConstruct_pac.gph, saving(lnConstruct_ac_pac, replace)
graph export "lnConstruct_ac_pac.png", replace
dfuller lnConstruct, trend regress

ac lnLeisure, saving(lnLeisure_ac, replace)
pac lnLeisure, saving(lnLeisure_pac, replace)
graph combine lnLeisure_ac.gph lnLeisure_pac.gph, saving(lnLeisure_ac_pac, replace)
graph export "lnLeisure_ac_pac.png", replace
dfuller lnLeisure, trend regress

ac lnManufacture, saving(lnManufacture_ac, replace)
pac lnManufacture, saving(lnManufacture_pac, replace)
graph combine lnManufacture_ac.gph lnManufacture_pac.gph, saving(lnManufacture_ac_pac, replace)
graph export "lnManufacture_ac_pac.png", replace
dfuller lnManufacture, trend regress

quietly reg l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture
testparm l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture
```

```
newey d.lnTotal l(0/3,12,24)d.Construct l(0/3,12,24)d.Manufacture l(0/3,12,24)d.Leisure, lag(24)
test ld.Construct + ld.Construct + l2d.Construct + l3d.Construct + l12d.Construct + l24d.Construct ///
  == d.Manufacture + ld.Manufacture + l2d.Manufacture + l3d.Manufacture + ///
    l12d.Manufacture + l24d.Manufacture
test d.Construct + ld.Construct + l2d.Construct + l3d.Construct + l12d.Construct + l24d.Construct ///
  == d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure + l12d.Leisure + l24d.Leisure
test d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure + l12d.Leisure + l24d.Leisure ///
  == d.Manufacture + ld.Manufacture + l2d.Manufacture + l3d.Manufacture + ///
    l12d.Manufacture + l24d.Manufacture

*------------------------------------------------------------------------------*
gen dlnConstruct=d.lnConstruct
gen l1dlnConstruct=l1d.lnConstruct
gen l2dlnConstruct=l2d.lnConstruct
gen l3dlnConstruct=l3d.lnConstruct
gen l12dlnConstruct=l12d.lnConstruct
gen l24dlnConstruct=l24d.lnConstruct

gen dlnLeisure=d.lnLeisure
gen l1dlnLeisure=l1d.lnLeisure
gen l2dlnLeisure=l2d.lnLeisure
gen l3dlnLeisure=l3d.lnLeisure
gen l12dlnLeisure=l12d.lnLeisure
gen l24dlnLeisure=l24d.lnLeisure

gen dlnManufacture=d.lnManufacture
gen l1dlnManufacture=l1d.lnManufacture
gen l2dlnManufacture=l2d.lnManufacture
gen l3dlnManufacture=l3d.lnManufacture
gen l12dlnManufacture=l12d.lnManufacture
gen l24dlnManufacture=l24d.lnManufacture

gen dlnTotal=d.lnTotal
gen l1dlnTotal=l1d.lnTotal
gen l2dlnTotal=l2d.lnTotal
gen l3dlnTotal=l3d.lnTotal
gen l12dlnTotal=l12d.lnTotal
gen l24dlnTotal=l24d.lnTotal

/*
gsreg dlnTotal dlnConstruct l1dlnConstruct l2dlnConstruct l3dlnConstruct ///
  l12dlnConstruct l24dlnConstruct ///
```

```
  dlnLeisure l1dlnLeisure l2dlnLeisure l3dlnLeisure l12dlnLeisure l24dlnLeisure ///
  dlnManufacture l1dlnManufacture l2dlnManufacture l3dlnManufacture ///
  l12dlnManufacture l24dlnManufacture ///
  if tin(1990m1,2021m3), ///
  ncomb(1,6) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11) ///
  samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnTtoal) replace
*/


loocv reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
quietly reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
estat ic

loocv reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
quietly reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
estat ic

loocv reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
  l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
quietly reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
  l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
estat ic

loocv reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct ///
  l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
quietly reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct ///
  l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
estat ic

*Lowest rmse (1)
reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
scalar drop _all
quietly forval w=12(12)180 {
gen pred=.
gen nobs=.
  forval t=544/734 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg d.lnTotal l(1/3)d.lnTotal 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.lnTotal)^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 =         12
RWrmse12 =   .0132376
*/

*lowest AIC and BIC (3)
reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
  l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
scalar drop _all
quietly forval w=3(3)180 {
gen pred=.
gen nobs=.
  forval t=544/734 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
    l(1/3)d.lnManufacture 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.lnTotal)^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 =         12
RWrmse12 =   .0132376
*/
```

```
* Going with model 1 because average RWrmse is lower across window sizes
scalar rwrmse = .0132376
reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(,2021m3)
predict pd
gen pflcount=exp((rwrmse^2)/2)*exp(l.lnTotal+pd) if Date==tm(2021m4)
gen ub1=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+1*rwrmse) if Date==tm(2021m4)
gen lb1=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-1*rwrmse) if Date==tm(2021m4)
gen ub2=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+2*rwrmse) if Date==tm(2021m4)
gen lb2=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-2*rwrmse) if Date==tm(2021m4)
gen ub3=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+3*rwrmse) if Date==tm(2021m4)
gen lb3=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-3*rwrmse) if Date==tm(2021m4)
drop pd

replace pflcount=Total if Date==tm(2021m3)
replace ub1=Total if Date==tm(2021m3)
replace ub2=Total if Date==tm(2021m3)
replace ub3=Total if Date==tm(2021m3)
replace lb1=Total if Date==tm(2021m3)
replace lb2=Total if Date==tm(2021m3)
replace lb3=Total if Date==tm(2021m3)

twoway (tsrline ub3 ub2 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
  (tsrline ub2 ub1 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
  (tsrline ub1 pflcount if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
  (tsrline pflcount lb1 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
  (tsrline lb1 lb2 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
  (tsrline lb2 lb3 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
  (tsline Total pflcount if tin(2020m4,2021m4) , ///
  lcolor(gs12 pink) lwidth(medthick medthick) ///
  lpattern(solid longdash)), scheme(s1mono) legend(off)
graph export "TotalFan1.png", replace

* More than 1 step
arima d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
tin(1990m1,2021m3)
predict pnonfarm, dynamic(tm(2021m3))
predict mse, mse dynamic(mofd(tm(2021m4)))
gen totmse = mse if Date==tm(2021m4)
replace totmse = l.totmse+mse if Date>tm(2021m4)
gen pnonfarma = Total if Date==tm(2021m3)
```

```
replace pnonfarma = l.pnonfarma*exp(pnonfarm+mse/2) if Date>tm(2021m3)

gen ub1a = pnonfarma*exp(totmse^.5)
gen ub2a = pnonfarma*exp(2*totmse^.5)
gen ub3a = pnonfarma*exp(3*totmse^.5)
gen lb1a = pnonfarma/exp(totmse^.5)
gen lb2a = pnonfarma/exp(2*totmse^.5)
gen lb3a = pnonfarma/exp(3*totmse^.5)

replace ub1a=Total if Date == tm(2021m3)
replace ub2a=Total if Date == tm(2021m3)
replace ub3a=Total if Date == tm(2021m3)
replace lb1a=Total if Date == tm(2021m3)
replace lb2a=Total if Date == tm(2021m3)
replace lb3a=Total if Date == tm(2021m3)

twoway (tsrline ub3a ub2a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
  (tsrline ub2a ub1a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
  (tsrline ub1a pnonfarma if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
  (tsrline pnonfarma lb1a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
  (tsrline lb1a lb2a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
  (tsrline lb2a lb3a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
  (tsline Total pnonfarma if tin(2019m1,2022m3) , ///
  lcolor(gs12 pink) lwidth(medthick medthick) ///
  lpattern(solid longdash)) , scheme(s1mono) legend(off)
graph export "TotalFan12.png", replace

scalar rmse_mod1 = .0132376
reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
tin(1990m1,2021m3)
predict plTotal
predict temp if tin(2021m3,2021m3)
replace plTotal=temp if tin(2021m3,2021m3)
drop temp
gen pTotal=exp(l.lnTotal+plTotal+(rmse_mod1^2)/2)
gen lbTotal=exp(l.lnTotal+plTotal-1.96*rmse_mod1+(rmse_mod1^2)/2)
gen ubTotal=exp(l.lnTotal+plTotal+1.96*rmse_mod1+(rmse_mod1^2)/2)

gen res=(d.lnTotal-plTotal)
gen expres=exp(res)
summ expres
scalar meanexpres=r(mean)
```

```
gen epTotal=exp(l.lnTotal+plTotal)*meanexpres
_pctile res, percentile(2.5,97.5)
return list
gen elbTotal=exp(l.lnTotal+plTotal+r(r1))*meanexpres
gen eubTotal=exp(l.lnTotal+plTotal+r(r2))*meanexpres

tsline Total pTotal elbTotal eubTotal lbTotal ubTotal if tin(2019m1,2021m4), ///
  scheme(s1mono) tline(2021m3, lcolor(gs4)) ///
  lpattern(solid solid longdash longdash shortdash shortdash) ///
  lcolor(dkorange gs5 gs10 gs10 dkorange%60 dkorange%60) ///
  lwidth(medthick medthick medium medium)
graph export "interval_tsline.png", replace

histogram expres, normal kdensity saving(residuals.gph, replace)
graph export "residuals.png", replace

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

Log File

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

      -------------------------------------------------------------------------------
            name:  <unnamed>
             log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Probl
      > em Sets/Final Exam/Final Exam.smcl
        log type:  smcl
       opened on:  29 Apr 2021, 12:07:09

     1 . import delimited "SP21Final.csv"
      (5 vars, 375 obs)

     2 .
     3 . gen datec=date(date, "YMD")

     4 . gen Date=mofd(datec)
```

```

  5 . gen month=month(datec)

  6 . format Date %tm

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

  8 .
  9 . gen lnConstruct = ln(construct)

 10 . gen lnLeisure = ln(leisure)

 11 . gen lnManufacture = ln(manufacture)

 12 . gen lnTotal = ln(total)

 13 .
 14 . gen Total = total

 15 . gen Construct = construct

 16 . gen Leisure = leisure

 17 . gen Manufacture = manufacture

 18 .
 19 . /*
   > gen withMarchTotal = Total
   > replace Total=. if tin(2021m3,)
   > */
 20 .
 21 . tsset Date
          time variable:  Date, 1990m1 to 2021m3
                  delta:  1 month

 22 . tsappend, add(12)

 23 . replace month=month(dofm(Date))
   (12 real changes made)

 24 .
 25 . gen m1=0

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

```

```
27 . gen m2=0

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

29 . gen m3=0

30 . replace m3=1 if month==3
  (33 real changes made)

31 . gen m4=0

32 . replace m4=1 if month==4
  (32 real changes made)

33 . gen m5=0

34 . replace m5=1 if month==5
  (32 real changes made)

35 . gen m6=0

36 . replace m6=1 if month==6
  (32 real changes made)

37 . gen m7=0

38 . replace m7=1 if month==7
  (32 real changes made)

39 . gen m8=0

40 . replace m8=1 if month==8
  (32 real changes made)

41 . gen m9=0

42 . replace m9=1 if month==9
  (32 real changes made)

43 . gen m10=0

44 . replace m10=1 if month==10
  (32 real changes made)

45 . gen m11=0

46 . replace m11=1 if month==11
```

```
      (32 real changes made)

    47 .
    48 .
    49 . summ construct leisure manufacture total

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
         construct |        375    461.0043     95.8947      323.9      696.1
           leisure |        375    930.2083    159.6216      660.6     1287.5
       manufacture |        375    410.0496    63.14375      307.9      518.2
             total |        375    6161.164    958.2068     4366.1     8010.4

    50 . summ lnConstruct lnLeisure lnManufacture lnTotal

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
       lnConstruct |        375    6.112377    .2043883   5.780435   6.545493
         lnLeisure |        375    6.820994    .1695282   6.493148   7.160458
      lnManufact~e |        375    6.004071    .1577555   5.729775   6.250361
           lnTotal |        375     8.71332    .1619029   8.381625   8.988496

    51 .
    52 . tsline lnConstruct lnLeisure, saving(lnConstructLeisure_tsline.gph, replace)
      (file lnConstructLeisure_tsline.gph saved)

    53 . tsline lnManufacture, saving(lnManufacture_tsline.gph, replace)
      (file lnManufacture_tsline.gph saved)

    54 . graph combine lnConstructLeisure_tsline.gph lnManufacture_tsline.gph, ///
      >         saving(lnConstructLeisure, replace)
      (file lnConstructLeisure.gph saved)

    55 . graph export "lnConstructLeisure-Manufacture_tsline.png", replace
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
      > /Final Exam/lnConstructLeisure-Manufacture_tsline.png written in PNG format)

    56 .
    57 . tsline lnTotal, saving(lnTotal_tsline.gph, replace)
      (file lnTotal_tsline.gph saved)

    58 . tsline d.lnTotal, saving(dlnTotal_tsline.gph, replace)
      (file dlnTotal_tsline.gph saved)

```

```
  59 . graph combine Total_tsline.gph dlnTotal_tsline.gph, saving(lnTotal-Total, rep
    > lace)
    (file lnTotal-Total.gph saved)

  60 . graph export "lnTotal-dlnTotal_tsline.png", replace
    (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
    > /Final Exam/lnTotal-dlnTotal_tsline.png written in PNG format)

  61 .
  62 . ac lnTotal, saving(lnTotal_ac, replace)
    (file lnTotal_ac.gph saved)

  63 . pac lnTotal, saving(lnTotal_pac, replace)
    (file lnTotal_pac.gph saved)

  64 . graph combine lnTotal_ac.gph lnTotal_pac.gph, saving(lnTotal_ac_pac, replace)
    (file lnTotal_ac_pac.gph saved)

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

  66 . dfuller lnTotal, trend regress

    Dickey-Fuller test for unit root                   Number of obs   =   374

                               ---------- Interpolated Dickey-Fuller ---------
                      Test         1% Critical       5% Critical      10% Critical
                   Statistic           Value             Value             Value
    ------------------------------------------------------------------------------
     Z(t)             -1.895            -3.985            -3.425            -3.130
    ------------------------------------------------------------------------------
    MacKinnon approximate p-value for Z(t) = 0.6570

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

```
        D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
          lnTotal |
              L1. |  -.0191916   .0101251    -1.90   0.059    -.0391013 .0007181
                  |
           _trend |   .0000223   .0000151     1.47   0.142    -7.48e-06 .0000521
            _cons |   .1644033    .085612     1.92   0.056    -.0039423 .3327489
      ------------------------------------------------------------------------------

    67 .
    68 . ac lnConstruct, saving(lnConstruct_ac, replace)
      (file lnConstruct_ac.gph saved)

    69 . pac lnConstruct, saving(lnConstruct_pac, replace)
      (file lnConstruct_pac.gph saved)

    70 . graph combine lnConstruct_ac.gph lnConstruct_pac.gph, saving(lnConstruct_ac_p
      > ac, replace)
      (file lnConstruct_ac_pac.gph saved)

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

    72 . dfuller lnConstruct, trend regress

      Dickey-Fuller test for unit root                   Number of obs   =  374

                                    ---------- Interpolated Dickey-Fuller ---------
                        Test         1% Critical       5% Critical      10% Critical
                     Statistic           Value             Value             Value
      ------------------------------------------------------------------------------
       Z(t)             -0.586            -3.985            -3.425            -3.130
```

```
      ------------------------------------------------------------------------
-----
      MacKinnon approximate p-value for Z(t) = 0.9796

      ------------------------------------------------------------------------
-----
      D.           |
       lnConstruct |      Coef.   Std. Err.      t    P>|t|     [95% Conf.
Interval]
      -------------+----------------------------------------------------------
-----
       lnConstruct |
               L1. |  -.0021673   .0036962    -0.59   0.558    -.0094355
.0051009
                   |
            _trend |   5.05e-06   6.99e-06     0.72   0.470    -8.68e-06
.0000188
             _cons |   .0132628   .0221901     0.60   0.550    -.0303713
.0568969
      ------------------------------------------------------------------------
-----

    73 .
    74 . ac lnLeisure, saving(lnLeisure_ac, replace)
      (file lnLeisure_ac.gph saved)

    75 . pac lnLeisure, saving(lnLeisure_pac, replace)
      (file lnLeisure_pac.gph saved)

    76 . graph combine lnLeisure_ac.gph lnLeisure_pac.gph,
saving(lnLeisure_ac_pac, re
      > place)
      (file lnLeisure_ac_pac.gph saved)

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

    78 . dfuller lnLeisure, trend regress

      Dickey-Fuller test for unit root                   Number of obs   =
  374

                                     ---------- Interpolated Dickey-Fuller ----
-----
                        Test         1% Critical       5% Critical      10%
Critical
```

```
                     Statistic           Value             Value             Value
      ------------------------------------------------------------------------------
       Z(t)             -4.787            -3.985            -3.425            -3.130
      ------------------------------------------------------------------------------
      MacKinnon approximate p-value for Z(t) = 0.0005

      ------------------------------------------------------------------------------
       D.lnLeisure |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
         lnLeisure |
               L1. |  -.1242368   .0259543    -4.79   0.000    -.1752728   -.0732007
                   |
            _trend |   .0001744   .0000407     4.28   0.000     .0000943    .0002545
             _cons |   .8156996   .1699707     4.80   0.000     .4814728    1.149926
      ------------------------------------------------------------------------------

    79 .
    80 . ac lnManufacture, saving(lnManufacture_ac, replace)
      (file lnManufacture_ac.gph saved)

    81 . pac lnManufacture, saving(lnManufacture_pac, replace)
      (file lnManufacture_pac.gph saved)

    82 . graph combine lnManufacture_ac.gph lnManufacture_pac.gph, saving(lnManufactur
      > e_ac_pac, replace)
      (file lnManufacture_ac_pac.gph saved)

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

    84 . dfuller lnManufacture, trend regress

      Dickey-Fuller test for unit root                   Number of obs   =       374
```

```

                                 ---------- Interpolated Dickey-Fuller ---------
                        Test         1% Critical       5% Critical      10% Critical
                     Statistic           Value             Value             Value
      ------------------------------------------------------------------------------
       Z(t)              0.313            -3.985            -3.425            -3.130
      ------------------------------------------------------------------------------
      MacKinnon approximate p-value for Z(t) = 0.9963

      -------------------------------------------------------------------------------
      D.             |
      lnManufacture |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      --------------+----------------------------------------------------------------
      lnManufacture |
                L1. |   .0013361   .0042655     0.31   0.754    -.0070515    .0097238
                    |
             _trend |   9.58e-06   6.23e-06     1.54   0.125    -2.67e-06    .0000218
              _cons |  -.0106361   .0265921    -0.40   0.689    -.0629262    .0416539
      -------------------------------------------------------------------------------

    85 .
    86 . quietly reg l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture

    87 . testparm l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture

       ( 1)  L24D.Construct = 0
       ( 2)  L12D.Leisure = 0
       ( 3)  L24D.Leisure = 0
       ( 4)  L12D.Manufacture = 0
       ( 5)  L24D.Manufacture = 0

             F(  5,   356) =  146.28
                  Prob > F =    0.0000

```

```
    88 .
    89 . newey d.lnTotal l(0/3,12,24)d.Construct l(0/3,12,24)d.Manufacture l(0/3,12,24
      > )d.Leisure, lag(24)

      Regression with Newey-West standard errors      Number of obs     =       350
      maximum lag: 24                                 F( 18,       331) =   4461.04
                                                      Prob > F          =    0.0000

      ------------------------------------------------------------------------------
                   |             Newey-West
         D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
         Construct |
               D1. |   .0004252   .0000809     5.26   0.000     .0002661    .0005842
               LD. |   .0000247    .000044     0.56   0.574    -.0000617    .0001112
              L2D. |   .0000613   .0000403     1.52   0.129    -.0000179    .0001405
              L3D. |  -.0000674   .0000577    -1.17   0.244     -.000181    .0000462
             L12D. |   -.000035   .0000641    -0.55   0.585     -.000161     .000091
             L24D. |   .0000494   .0000684     0.72   0.470    -.0000851    .0001839
                   |
       Manufacture |
               D1. |   .0011938   .0001924     6.21   0.000     .0008154    .0015723
               LD. |  -.0002374   .0001386    -1.71   0.088      -.00051    .0000353
              L2D. |   -.000336   .0000718    -4.68   0.000    -.0004773   -.0001948
              L3D. |  -.0002689   .0001059    -2.54   0.012    -.0004772   -.0000606
             L12D. |   .0003308   .0001381     2.39   0.017      .000059    .0006025
             L24D. |   .0002403   .0001413     1.70   0.090    -.0000376    .0005182
                   |
           Leisure |
```

```
                D1. |   .0002134   8.38e-06    25.47   0.000     .0001969
.0002299
                LD. |   5.34e-06   9.19e-06     0.58   0.561    -.0000127
.0000234
               L2D. |  -2.42e-06   .0000172    -0.14   0.888    -.0000364
.0000315
               L3D. |  -8.76e-06   .0000174    -0.50   0.614    -.0000429
.0000254
              L12D. |   .0000629   .0000361     1.74   0.082    -8.04e-06
.0001339
              L24D. |   -.000016   .0000428    -0.37   0.708    -.0001002
.0000681
                    |
              _cons |   .0011957   .0003664     3.26   0.001      .000475
.0019165
      ------------------------------------------------------------------------
-----

    90 . test ld.Construct + ld.Construct + l2d.Construct + l3d.Construct +
l12d.Const
      > ruct + l24d.Construct ///
      >         == d.Manufacture + ld.Manufacture + l2d.Manufacture +
l3d.Manufacture
      >  + ///
      >                 l12d.Manufacture + l24d.Manufacture

       ( 1)  2*LD.Construct + L2D.Construct + L3D.Construct + L12D.Construct +
             L24D.Construct - D.Manufacture - LD.Manufacture - L2D.Manufacture
-
             L3D.Manufacture - L12D.Manufacture - L24D.Manufacture = 0

             F(  1,   331) =    5.85
                  Prob > F =    0.0161

    91 . test d.Construct + ld.Construct + l2d.Construct + l3d.Construct +
l12d.Constr
      > uct + l24d.Construct ///
      >         == d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure +
l12d.Leisure
      > + l24d.Leisure

       ( 1)  D.Construct + LD.Construct + L2D.Construct + L3D.Construct +
             L12D.Construct + L24D.Construct - D.Leisure - LD.Leisure -
L2D.Leisure -
             L3D.Leisure - L12D.Leisure - L24D.Leisure = 0

             F(  1,   331) =   11.36
                  Prob > F =    0.0008
```

```

   92 . test d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure + l12d.Leisure + l24d
      > .Leisure ///
      >         == d.Manufacture + ld.Manufacture + l2d.Manufacture + l3d.Manufacture
      >  + ///
      >                 l12d.Manufacture + l24d.Manufacture

       ( 1)  - D.Manufacture - LD.Manufacture - L2D.Manufacture - L3D.Manufacture -
             L12D.Manufacture - L24D.Manufacture + D.Leisure + LD.Leisure +
             L2D.Leisure + L3D.Leisure + L12D.Leisure + L24D.Leisure = 0

             F(  1,   331) =    4.74
                  Prob > F =    0.0301

   93 .
   94 . *--------------------------------------------------------------------------
      > --*
   95 . gen dlnConstruct=d.lnConstruct
      (13 missing values generated)

   96 . gen l1dlnConstruct=l1d.lnConstruct
      (13 missing values generated)

   97 . gen l2dlnConstruct=l2d.lnConstruct
      (13 missing values generated)

   98 . gen l3dlnConstruct=l3d.lnConstruct
      (13 missing values generated)

   99 . gen l12dlnConstruct=l12d.lnConstruct
      (13 missing values generated)

  100 . gen l24dlnConstruct=l24d.lnConstruct
      (25 missing values generated)

  101 .
  102 . gen dlnLeisure=d.lnLeisure
      (13 missing values generated)

  103 . gen l1dlnLeisure=l1d.lnLeisure
      (13 missing values generated)

  104 . gen l2dlnLeisure=l2d.lnLeisure
      (13 missing values generated)
```

```
105 . gen l3dlnLeisure=l3d.lnLeisure
   (13 missing values generated)

106 . gen l12dlnLeisure=l12d.lnLeisure
   (13 missing values generated)

107 . gen l24dlnLeisure=l24d.lnLeisure
   (25 missing values generated)

108 .
109 . gen dlnManufacture=d.lnManufacture
   (13 missing values generated)

110 . gen l1dlnManufacture=l1d.lnManufacture
   (13 missing values generated)

111 . gen l2dlnManufacture=l2d.lnManufacture
   (13 missing values generated)

112 . gen l3dlnManufacture=l3d.lnManufacture
   (13 missing values generated)

113 . gen l12dlnManufacture=l12d.lnManufacture
   (13 missing values generated)

114 . gen l24dlnManufacture=l24d.lnManufacture
   (25 missing values generated)

115 .
116 . gen dlnTotal=d.lnTotal
   (13 missing values generated)

117 . gen l1dlnTotal=l1d.lnTotal
   (13 missing values generated)

118 . gen l2dlnTotal=l2d.lnTotal
   (13 missing values generated)

119 . gen l3dlnTotal=l3d.lnTotal
   (13 missing values generated)

120 . gen l12dlnTotal=l12d.lnTotal
   (13 missing values generated)

121 . gen l24dlnTotal=l24d.lnTotal
   (25 missing values generated)
```

```
  122 .
  123 . /*
      > gsreg dlnTotal dlnConstruct l1dlnConstruct l2dlnConstruct l3dlnConstruct ///
      >         l12dlnConstruct l24dlnConstruct ///
      >         dlnLeisure l1dlnLeisure l2dlnLeisure l3dlnLeisure l12dlnLeisure l24dl
      > nLeisure ///
      >         dlnManufacture l1dlnManufacture l2dlnManufacture l3dlnManufacture ///
      >         l12dlnManufacture l24dlnManufacture ///
      >         if tin(1990m1,2021m3), ///
      >         ncomb(1,6) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11)
      > ///
      >         samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnTtoal)
      >  replace
      > */
  124 .
  125 .
  126 . loocv reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11


       Leave-One-Out Cross-Validation Results
      -----------------------------------------
               Method          |    Value
      -------------------------+---------------
      Root Mean Squared Errors |   .0129605
      Mean Absolute Errors     |   .00443719
      Pseudo-R2                |   .14054754
      -----------------------------------------

  127 . quietly reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

  128 . estat ic

      Akaike's information criterion and Bayesian information criterion

      -----------------------------------------------------------------------------
             Model |          N   ll(null)  ll(model)      df        AIC         BIC
      -------------+---------------------------------------------------------------
                 . |        371   1102.298   1206.362      15  -2382.725   -2323.982
```

```
      --------------------------------------------------------------------------------
      Note: BIC uses N = number of observations. See [R] BIC note.

   129 .
   130 . loocv reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11


       Leave-One-Out Cross-Validation Results
      -----------------------------------------
               Method          |    Value
      -------------------------+---------------
      Root Mean Squared Errors |   .01300797
      Mean Absolute Errors     |   .00427199
      Pseudo-R2                |   .14852182
      -----------------------------------------

   131 . quietly reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m1
      > 1

   132 . estat ic

      Akaike's information criterion and Bayesian information criterion

      --------------------------------------------------------------------------------
             Model |          N   ll(null)  ll(model)      df         AIC        BIC
      -------------+------------------------------------------------------------------
                 . |        350    1035.49   1135.109      17   -2236.218   -2170.633
      --------------------------------------------------------------------------------
      Note: BIC uses N = number of observations. See [R] BIC note.

   133 .
   134 . loocv reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
      >         l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11


       Leave-One-Out Cross-Validation Results
      -----------------------------------------
               Method          |    Value
      -------------------------+---------------
```

```
      Root Mean Squared Errors |   .01611154
      Mean Absolute Errors     |   .00423375
      Pseudo-R2                |   .06722376
      -----------------------------------------

   135 . quietly reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure /
      > //
      >         l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

   136 . estat ic

      Akaike's information criterion and Bayesian information criterion

      -----------------------------------------------------------------------------
             Model |          N   ll(null)  ll(model)      df        AIC         BIC
      -------------+---------------------------------------------------------------
                 . |        371   1102.298   1220.504      24  -2393.008   -2299.019
      -----------------------------------------------------------------------------
      Note: BIC uses N = number of observations. See [R] BIC note.

   137 .
   138 . loocv reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct ///
      >         l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8
      >  m9 m10 m11


       Leave-One-Out Cross-Validation Results
      -----------------------------------------
               Method          |    Value
      -------------------------+---------------
      Root Mean Squared Errors |   .01713897
      Mean Absolute Errors     |   .004436
      Pseudo-R2                |   .05184396
      -----------------------------------------

   139 . quietly reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct ///
      >         l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8
      >  m9 m10 m11
```

```

  140 . estat ic

     Akaike's information criterion and Bayesian information criterion

     -----------------------------------------------------------------------------
            Model |          N   ll(null)  ll(model)      df        AIC        BIC
     -------------+---------------------------------------------------------------
                . |        350    1035.49   1145.391      30  -2230.781  -2115.043
     -----------------------------------------------------------------------------
     Note: BIC uses N = number of observations. See [R] BIC note.

  141 .
  142 . *Lowest rmse (1)
  143 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

           Source |       SS           df       MS      Number of obs   =       371
     -------------+----------------------------------   F(14, 356)      =     19.13
            Model |  .024491123        14  .001749366   Prob > F        =    0.0000
         Residual |  .032550038       356  .000091433   R-squared       =    0.4294
     -------------+----------------------------------   Adj R-squared   =    0.4069
            Total |  .057041161       370  .000154165   Root MSE        =    .00956

     ------------------------------------------------------------------------------
        D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
     -------------+----------------------------------------------------------------
          lnTotal |
              LD. |  -.0016173   .0529806    -0.03   0.976    -.1058116    .1025769
             L2D. |  -.1325018   .0525096    -2.52   0.012    -.2357698   -.0292338
             L3D. |    .026173   .0529824     0.49   0.622    -.0780247    .1303708
                  |
```

```
                  m1 |  -.0254857   .0024679   -10.33   0.000    -.0303391    -.0206322
                  m2 |   .0004869   .0029045     0.17   0.867    -.0052252     .006199
                  m3 |   -.003913   .0027686    -1.41   0.158    -.0093579     .0015318
                  m4 |   -.013317   .0026543    -5.02   0.000    -.0185371    -.0080969
                  m5 |  -.0064632   .0026182    -2.47   0.014    -.0116124    -.0013141
                  m6 |  -.0088465   .0025835    -3.42   0.001    -.0139273    -.0037657
                  m7 |  -.0159741   .0025331    -6.31   0.000    -.0209559    -.0109924
                  m8 |  -.0051433   .0026374    -1.95   0.052    -.0103302     .0000435
                  m9 |  -.0096793   .0025635    -3.78   0.000    -.0147208    -.0046378
                 m10 |  -.0021472   .0025489    -0.84   0.400      -.00716     .0028657
                 m11 |   .0024587   .0024684     1.00   0.320    -.0023958     .0073131
               _cons |   .0088125   .0018473     4.77   0.000     .0051795     .0124454
      ------------------------------------------------------------------------------

   144 . scalar drop _all

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

   146 . scalar list
      RWmaxobs180 =        180
      RWminobs180 =        180
       RWrmse180 =  .01789015
      RWmaxobs168 =        168
      RWminobs168 =        168
       RWrmse168 =  .01812779
      RWmaxobs156 =        156
      RWminobs156 =        156
       RWrmse156 =   .0186461
      RWmaxobs144 =        144
      RWminobs144 =        144
       RWrmse144 =  .01935361
      RWmaxobs132 =        132
      RWminobs132 =        132
       RWrmse132 =  .02010893
      RWmaxobs120 =        120
```

```
      RWminobs120 =        120
        RWrmse120 =  .02273114
      RWmaxobs108 =        108
      RWminobs108 =        108
        RWrmse108 =   .0233275
       RWmaxobs96 =         96
       RWminobs96 =         96
         RWrmse96 =   .0244855
       RWmaxobs84 =         84
       RWminobs84 =         84
         RWrmse84 =  .02493232
       RWmaxobs72 =         72
       RWminobs72 =         72
         RWrmse72 =  .02471091
       RWmaxobs60 =         60
       RWminobs60 =         60
         RWrmse60 =  .02489376
       RWmaxobs48 =         48
       RWminobs48 =         48
         RWrmse48 =  .02475145
       RWmaxobs36 =         36
       RWminobs36 =         36
         RWrmse36 =  .02791922
       RWmaxobs24 =         24
       RWminobs24 =         24
         RWrmse24 =  .16402211
       RWmaxobs12 =         12
       RWminobs12 =         12
         RWrmse12 =   .0132376

  147 .
  148 . /*
      > RWmaxobs12 =         12
      > RWminobs12 =         12
      > RWrmse12 =   .0132376
      > */
  149 .
  150 . *lowest AIC and BIC (3)
  151 . reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
      >         l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

            Source |       SS           df       MS      Number of obs   =   371
      -------------+----------------------------------   F(23, 347)      = 13.45
             Model |  .026880308        23  .001168709   Prob > F        = 0.0000
```

```
          Residual |  .030160853       347  .000086919   R-squared       = 0.4712
      -------------+----------------------------------   Adj R-squared   = 0.4362
             Total |  .057041161       370  .000154165   Root MSE        = .00932

      ------------------------------------------------------------------------------
          D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      --------------+---------------------------------------------------------------
            lnTotal |
                LD. |   .2862672   .2667427     1.07   0.284    -.2383687    .8109032
               L2D. |   .2454317   .2661885     0.92   0.357    -.2781141    .7689776
               L3D. |   .4266043   .2702181     1.58   0.115    -.1048673    .9580758
                    |
        lnConstruct |
                LD. |   6.41e-06    .081453     0.00   1.000    -.1601974    .1602102
               L2D. |   .0327201   .0818081     0.40   0.689     -.128182    .1936223
               L3D. |  -.0371503   .0817044    -0.45   0.650    -.1978486     .123548
                    |
          lnLeisure |
                LD. |  -.1208348   .0688988    -1.75   0.080    -.2563467    .0146771
               L2D. |  -.1169829   .0684363    -1.71   0.088     -.251585    .0176192
               L3D. |  -.1609605    .069157    -2.33   0.021    -.2969801    -.024941
                    |
      lnManufacture |
                LD. |   .0485695   .1821043     0.27   0.790    -.3095975    .4067365
               L2D. |  -.1583242   .1862613    -0.85   0.396    -.5246673     .208019
               L3D. |   .1372412   .1883006     0.73   0.467    -.2331129    .5075952
                    |
                 m1 |  -.0259546   .0026844    -9.67   0.000    -.0312344   -.0206748
```

```
                m2 |   .0061874   .0054433     1.14   0.256    -.0045187
.0168935
                m3 |   .0041315   .0052298     0.79   0.430    -.0061545
.0144176
                m4 |  -.0019669   .0049557    -0.40   0.692     -.011714
.0077802
                m5 |  -.0024728   .0032223    -0.77   0.443    -.0088106
.0038649
                m6 |  -.0057242   .0032307    -1.77   0.077    -.0120785
.00063
                m7 |  -.0128758   .0035363    -3.64   0.000    -.0198311
-.0059206
                m8 |  -.0017029    .003742    -0.46   0.649    -.0090627
.005657
                m9 |   -.006856   .0032793    -2.09   0.037    -.0133058
-.0004062
               m10 |   .0015872   .0030906     0.51   0.608    -.0044915
.007666
               m11 |   .0030825   .0026162     1.18   0.240    -.0020632
.0082282
             _cons |   .0039798   .0029489     1.35   0.178    -.0018201
.0097797
      ------------------------------------------------------------------------
------

   152 . scalar drop _all

   153 . quietly forval w=3(3)180 {

   154 . scalar list
      RWmaxobs180 =        180
      RWminobs180 =        180
       RWrmse180 =  .06231312
      RWmaxobs177 =        177
      RWminobs177 =        177
       RWrmse177 =  .06310936
      RWmaxobs174 =        174
      RWminobs174 =        174
       RWrmse174 =  .06557282
      RWmaxobs171 =        171
      RWminobs171 =        171
       RWrmse171 =  .06644967
      RWmaxobs168 =        168
      RWminobs168 =        168
       RWrmse168 =   .0679586
      RWmaxobs165 =        165
      RWminobs165 =        165
       RWrmse165 =   .0691683
```

```
      RWmaxobs162 =        162
      RWminobs162 =        162
       RWrmse162 =  .06977207
      RWmaxobs159 =        159
      RWminobs159 =        159
       RWrmse159 =  .07051975
      RWmaxobs156 =        156
      RWminobs156 =        156
       RWrmse156 =   .0708996
      RWmaxobs153 =        153
      RWminobs153 =        153
       RWrmse153 =  .07104253
      RWmaxobs150 =        150
      RWminobs150 =        150
       RWrmse150 =  .07151194
      RWmaxobs147 =        147
      RWminobs147 =        147
       RWrmse147 =   .0724523
      RWmaxobs144 =        144
      RWminobs144 =        144
       RWrmse144 =  .07292486
      RWmaxobs141 =        141
      RWminobs141 =        141
       RWrmse141 =  .07359791
      RWmaxobs138 =        138
      RWminobs138 =        138
       RWrmse138 =  .07415885
      RWmaxobs135 =        135
      RWminobs135 =        135
       RWrmse135 =  .07551713
      RWmaxobs132 =        132
      RWminobs132 =        132
       RWrmse132 =  .07895271
      RWmaxobs129 =        129
      RWminobs129 =        129
       RWrmse129 =  .08680818
      RWmaxobs126 =        126
      RWminobs126 =        126
       RWrmse126 =  .08791729
      RWmaxobs123 =        123
      RWminobs123 =        123
       RWrmse123 =   .0885621
      RWmaxobs120 =        120
      RWminobs120 =        120
       RWrmse120 =   .0893681
      RWmaxobs117 =        117
      RWminobs117 =        117
       RWrmse117 =  .09222918
```

```
       RWmaxobs114 =        114
       RWminobs114 =        114
        RWrmse114 =  .09918976
       RWmaxobs111 =        111
       RWminobs111 =        111
        RWrmse111 =  .10159669
       RWmaxobs108 =        108
       RWminobs108 =        108
        RWrmse108 =   .1074872
       RWmaxobs105 =        105
       RWminobs105 =        105
        RWrmse105 =  .11024241
       RWmaxobs102 =        102
       RWminobs102 =        102
        RWrmse102 =  .11448013
       RWmaxobs99 =         99
       RWminobs99 =         99
         RWrmse99 =  .11681123
       RWmaxobs96 =         96
       RWminobs96 =         96
         RWrmse96 =  .12412703
       RWmaxobs93 =         93
       RWminobs93 =         93
         RWrmse93 =  .13010923
       RWmaxobs90 =         90
       RWminobs90 =         90
         RWrmse90 =  .12958828
       RWmaxobs87 =         87
       RWminobs87 =         87
         RWrmse87 =  .13728567
       RWmaxobs84 =         84
       RWminobs84 =         84
         RWrmse84 =  .14028654
       RWmaxobs81 =         81
       RWminobs81 =         81
         RWrmse81 =  .14073066
       RWmaxobs78 =         78
       RWminobs78 =         78
         RWrmse78 =  .14462066
       RWmaxobs75 =         75
       RWminobs75 =         75
         RWrmse75 =  .14520888
       RWmaxobs72 =         72
       RWminobs72 =         72
         RWrmse72 =  .14882576
       RWmaxobs69 =         69
       RWminobs69 =         69
         RWrmse69 =  .16426363
```

```
     RWmaxobs66 =         66
     RWminobs66 =         66
       RWrmse66 =  .16732114
     RWmaxobs63 =         63
     RWminobs63 =         63
       RWrmse63 =   .1679914
     RWmaxobs60 =         60
     RWminobs60 =         60
       RWrmse60 =  .18224837
     RWmaxobs57 =         57
     RWminobs57 =         57
       RWrmse57 =  .19923544
     RWmaxobs54 =         54
     RWminobs54 =         54
       RWrmse54 =   .2008839
     RWmaxobs51 =         51
     RWminobs51 =         51
       RWrmse51 =   .2096647
     RWmaxobs48 =         48
     RWminobs48 =         48
       RWrmse48 =  .20847129
     RWmaxobs45 =         45
     RWminobs45 =         45
       RWrmse45 =   .2244767
     RWmaxobs42 =         42
     RWminobs42 =         42
       RWrmse42 =  .25720484
     RWmaxobs39 =         39
     RWminobs39 =         39
       RWrmse39 =  .24835914
     RWmaxobs36 =         36
     RWminobs36 =         36
       RWrmse36 =  .24045465
     RWmaxobs33 =         33
     RWminobs33 =         33
       RWrmse33 =  .26919986
     RWmaxobs30 =         30
     RWminobs30 =         30
       RWrmse30 =  .49888354
     RWmaxobs27 =         27
     RWminobs27 =         27
       RWrmse27 =  .32666868
     RWmaxobs24 =         24
     RWminobs24 =         24
       RWrmse24 =  .66642366
     RWmaxobs21 =         21
     RWminobs21 =         21
       RWrmse21 =  .34614878
```

```
      RWmaxobs18 =          18
      RWminobs18 =          18
        RWrmse18 =     .757383
      RWmaxobs15 =          15
      RWminobs15 =          15
        RWrmse15 =   .21744051
      RWmaxobs12 =          12
      RWminobs12 =          12
        RWrmse12 =    .0132376
       RWmaxobs9 =           9
       RWminobs9 =           9
         RWrmse9 =   .01629782
       RWmaxobs6 =           6
       RWminobs6 =           6
         RWrmse6 =    .0162575
       RWmaxobs3 =           3
       RWminobs3 =           3
         RWrmse3 =    .0206163

   155 . /*
      > RWmaxobs12 =          12
      > RWminobs12 =          12
      > RWrmse12 =    .0132376
      > */
   156 .
   157 .
   158 . * Going with model 1 because average RWrmse is lower across window sizes
   159 . scalar rwrmse = .0132376

   160 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(,2021
      > m3)

            Source |       SS           df       MS      Number of obs   =   371
      -------------+----------------------------------   F(14, 356)      = 19.13
             Model |  .024491123        14  .001749366   Prob > F        = 0.0000
          Residual |  .032550038       356  .000091433   R-squared       = 0.4294
      -------------+----------------------------------   Adj R-squared   = 0.4069
             Total |  .057041161       370  .000154165   Root MSE        = .00956

```

```
      ------------------------------------------------------------------------------
         D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
           lnTotal |
               LD. |  -.0016173   .0529806    -0.03   0.976    -.1058116    .1025769
              L2D. |  -.1325018   .0525096    -2.52   0.012    -.2357698   -.0292338
              L3D. |    .026173   .0529824     0.49   0.622    -.0780247    .1303708
                   |
                m1 |  -.0254857   .0024679   -10.33   0.000    -.0303391   -.0206322
                m2 |   .0004869   .0029045     0.17   0.867    -.0052252     .006199
                m3 |   -.003913   .0027686    -1.41   0.158    -.0093579    .0015318
                m4 |   -.013317   .0026543    -5.02   0.000    -.0185371   -.0080969
                m5 |  -.0064632   .0026182    -2.47   0.014    -.0116124   -.0013141
                m6 |  -.0088465   .0025835    -3.42   0.001    -.0139273   -.0037657
                m7 |  -.0159741   .0025331    -6.31   0.000    -.0209559   -.0109924
                m8 |  -.0051433   .0026374    -1.95   0.052    -.0103302    .0000435
                m9 |  -.0096793   .0025635    -3.78   0.000    -.0147208   -.0046378
               m10 |  -.0021472   .0025489    -0.84   0.400      -.00716    .0028657
               m11 |   .0024587   .0024684     1.00   0.320    -.0023958    .0073131
             _cons |   .0088125   .0018473     4.77   0.000     .0051795    .0124454
      ------------------------------------------------------------------------------

   161 . predict pd
      (option xb assumed; fitted values)
      (15 missing values generated)

   162 . gen pflcount=exp((rwrmse^2)/2)*exp(l.lnTotal+pd) if Date==tm(2021m4)
      (386 missing values generated)

```

```
  163 . gen ub1=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+1*rwrmse) if
Date==tm(2021m4)
      (386 missing values generated)

  164 . gen lb1=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-1*rwrmse) if
Date==tm(2021m4)
      (386 missing values generated)

  165 . gen ub2=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+2*rwrmse) if
Date==tm(2021m4)
      (386 missing values generated)

  166 . gen lb2=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-2*rwrmse) if
Date==tm(2021m4)
      (386 missing values generated)

  167 . gen ub3=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+3*rwrmse) if
Date==tm(2021m4)
      (386 missing values generated)

  168 . gen lb3=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-3*rwrmse) if
Date==tm(2021m4)
      (386 missing values generated)

  169 . drop pd

  170 .
  171 . replace pflcount=Total if Date==tm(2021m3)
      (1 real change made)

  172 . replace ub1=Total if Date==tm(2021m3)
      (1 real change made)

  173 . replace ub2=Total if Date==tm(2021m3)
      (1 real change made)

  174 . replace ub3=Total if Date==tm(2021m3)
      (1 real change made)

  175 . replace lb1=Total if Date==tm(2021m3)
      (1 real change made)

  176 . replace lb2=Total if Date==tm(2021m3)
      (1 real change made)

  177 . replace lb3=Total if Date==tm(2021m3)
      (1 real change made)
```

```
  178 .
  179 . twoway (tsrline ub3 ub2 if tin(2020m4,2021m4), ///
     >         recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
     >         (tsrline ub2 ub1 if tin(2020m4,2021m4), ///
     >         recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
     >         (tsrline ub1 pflcount if tin(2020m4,2021m4), ///
     >         recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
     >         (tsrline pflcount lb1 if tin(2020m4,2021m4), ///
     >         recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
     >         (tsrline lb1 lb2 if tin(2020m4,2021m4), ///
     >         recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
     >         (tsrline lb2 lb3 if tin(2020m4,2021m4), ///
     >         recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
     >         (tsline Total pflcount if tin(2020m4,2021m4) , ///
     >         lcolor(gs12 pink) lwidth(medthick medthick) ///
     >         lpattern(solid longdash)), scheme(s1mono) legend(off)

  180 . graph export "TotalFan1.png", replace
     (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
     > /Final Exam/TotalFan1.png written in PNG format)

  181 .
  182 . * More than 1 step
  183 . arima d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(199
     > 0m1,2021m3)

     (setting optimization to BHHH)
     Iteration 0:   log likelihood =  1206.3625
     Iteration 1:   log likelihood =  1206.3625

     ARIMA regression

     Sample:  1990m5 - 2021m3                        Number of obs     =        371
                                                     Wald chi2(14)     =      83.94
     Log likelihood =  1206.362                      Prob > chi2       =     0.0000

     ------------------------------------------------------------------------------
                  |                 OPG
        D.lnTotal |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
     -------------+----------------------------------------------------------------
```

```
      lnTotal      |
           lnTotal |
               LD. |  -.0016173   .0448009    -0.04   0.971    -.0894254    .0861907
              L2D. |  -.1325018   .0486537    -2.72   0.006    -.2278614   -.0371422
              L3D. |    .026173   .0400229     0.65   0.513    -.0522705    .1046166
                   |
                m1 |  -.0254857   .0054401    -4.68   0.000    -.0361481   -.0148232
                m2 |   .0004869   .0068588     0.07   0.943    -.0129561    .0139299
                m3 |   -.003913   .0058695    -0.67   0.505    -.0154171     .007591
                m4 |   -.013317   .0047531    -2.80   0.005    -.0226329   -.0040012
                m5 |  -.0064632   .0082351    -0.78   0.433    -.0226038    .0096773
                m6 |  -.0088465   .0069016    -1.28   0.200    -.0223735    .0046804
                m7 |  -.0159741   .0056734    -2.82   0.005    -.0270938   -.0048545
                m8 |  -.0051433   .0074271    -0.69   0.489    -.0197002    .0094135
                m9 |  -.0096793   .0051505    -1.88   0.060    -.0197741    .0004155
               m10 |  -.0021472   .0048943    -0.44   0.661    -.0117399    .0074456
               m11 |   .0024587   .0066745     0.37   0.713    -.0106231    .0155405
             _cons |   .0088125   .0042033     2.10   0.036     .0005741    .0170508
      -------------+----------------------------------------------------------------
            /sigma |   .0093667   .0001378    67.96   0.000     .0090966    .0096369
      ------------------------------------------------------------------------------
      Note: The test of the variance against zero is one sided, and the two-sided
            confidence interval is truncated at zero.

   184 . predict pnonfarm, dynamic(tm(2021m3))
      (option xb assumed; predicted values)
      (4 missing values generated)

   185 . predict mse, mse dynamic(mofd(tm(2021m4)))
```

```
186 . gen totmse = mse if Date==tm(2021m4)
   (386 missing values generated)

187 . replace totmse = l.totmse+mse if Date>tm(2021m4)
   (11 real changes made)

188 . gen pnonfarma = Total if Date==tm(2021m3)
   (386 missing values generated)

189 . replace pnonfarma = l.pnonfarma*exp(pnonfarm+mse/2) if Date>tm(2021m3)
   (12 real changes made)

190 .
191 . gen ub1a = pnonfarma*exp(totmse^.5)
   (375 missing values generated)

192 . gen ub2a = pnonfarma*exp(2*totmse^.5)
   (375 missing values generated)

193 . gen ub3a = pnonfarma*exp(3*totmse^.5)
   (375 missing values generated)

194 . gen lb1a = pnonfarma/exp(totmse^.5)
   (375 missing values generated)

195 . gen lb2a = pnonfarma/exp(2*totmse^.5)
   (375 missing values generated)

196 . gen lb3a = pnonfarma/exp(3*totmse^.5)
   (375 missing values generated)

197 .
198 . replace ub1a=Total if Date == tm(2021m3)
   (1 real change made)

199 . replace ub2a=Total if Date == tm(2021m3)
   (1 real change made)

200 . replace ub3a=Total if Date == tm(2021m3)
   (1 real change made)

201 . replace lb1a=Total if Date == tm(2021m3)
   (1 real change made)

202 . replace lb2a=Total if Date == tm(2021m3)
   (1 real change made)
```

```
  203 . replace lb3a=Total if Date == tm(2021m3)
     (1 real change made)

  204 .
  205 . twoway (tsrline ub3a ub2a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
     >         (tsrline ub2a ub1a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
     >         (tsrline ub1a pnonfarma if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
     >         (tsrline pnonfarma lb1a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
     >         (tsrline lb1a lb2a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
     >         (tsrline lb2a lb3a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
     >         (tsline Total pnonfarma if tin(2019m1,2022m3) , ///
     >         lcolor(gs12 pink) lwidth(medthick medthick) ///
     >         lpattern(solid longdash)) , scheme(s1mono) legend(off)

  206 . graph export "TotalFan12.png", replace
     (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Problem Sets
     > /Final Exam/TotalFan12.png written in PNG format)

  207 .
  208 . scalar rmse_mod1 = .0132376

  209 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
tin(1990m
     > 1,2021m3)

           Source |       SS           df       MS      Number of obs   =
  371
     -------------+----------------------------------   F(14, 356)      =
19.13
            Model |  .024491123        14  .001749366   Prob > F        =
0.0000
         Residual |  .032550038       356  .000091433   R-squared       =
0.4294
     -------------+----------------------------------   Adj R-squared   =
0.4069
            Total |  .057041161       370  .000154165   Root MSE        =
.00956

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

```
        D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+------------------------------------------------------------------
          lnTotal |
              LD. |  -.0016173   .0529806    -0.03   0.976    -.1058116    .1025769
             L2D. |  -.1325018   .0525096    -2.52   0.012    -.2357698   -.0292338
             L3D. |    .026173   .0529824     0.49   0.622    -.0780247    .1303708
                  |
               m1 |  -.0254857   .0024679   -10.33   0.000    -.0303391   -.0206322
               m2 |   .0004869   .0029045     0.17   0.867    -.0052252     .006199
               m3 |   -.003913   .0027686    -1.41   0.158    -.0093579    .0015318
               m4 |   -.013317   .0026543    -5.02   0.000    -.0185371   -.0080969
               m5 |  -.0064632   .0026182    -2.47   0.014    -.0116124   -.0013141
               m6 |  -.0088465   .0025835    -3.42   0.001    -.0139273   -.0037657
               m7 |  -.0159741   .0025331    -6.31   0.000    -.0209559   -.0109924
               m8 |  -.0051433   .0026374    -1.95   0.052    -.0103302    .0000435
               m9 |  -.0096793   .0025635    -3.78   0.000    -.0147208   -.0046378
              m10 |  -.0021472   .0025489    -0.84   0.400      -.00716    .0028657
              m11 |   .0024587   .0024684     1.00   0.320    -.0023958    .0073131
            _cons |   .0088125   .0018473     4.77   0.000     .0051795    .0124454
      -----------------------------------------------------------------------------------

   210 . predict plTotal
      (option xb assumed; fitted values)
      (15 missing values generated)

   211 . predict temp if tin(2021m3,2021m3)
      (option xb assumed; fitted values)
      (386 missing values generated)

   212 . replace plTotal=temp if tin(2021m3,2021m3)
```

```
      (0 real changes made)

   213 . drop temp

   214 . gen pTotal=exp(l.lnTotal+plTotal+(rmse_mod1^2)/2)
      (15 missing values generated)

   215 . gen lbTotal=exp(l.lnTotal+plTotal-1.96*rmse_mod1+(rmse_mod1^2)/2)
      (15 missing values generated)

   216 . gen ubTotal=exp(l.lnTotal+plTotal+1.96*rmse_mod1+(rmse_mod1^2)/2)
      (15 missing values generated)

   217 .
   218 . gen res=(d.lnTotal-plTotal)
      (16 missing values generated)

   219 . gen expres=exp(res)
      (16 missing values generated)

   220 . summ expres

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
            expres |        371    1.000042    .0088821   .8565838   1.028623

   221 . scalar meanexpres=r(mean)

   222 . gen epTotal=exp(l.lnTotal+plTotal)*meanexpres
      (15 missing values generated)

   223 . _pctile res, percentile(2.5,97.5)

   224 . return list

      scalars:
                       r(r1) =  -.0089262239634991
                       r(r2) =  .0108835604041815

   225 . gen elbTotal=exp(l.lnTotal+plTotal+r(r1))*meanexpres
      (15 missing values generated)

   226 . gen eubTotal=exp(l.lnTotal+plTotal+r(r2))*meanexpres
      (15 missing values generated)

   227 .
   228 . tsline Total pTotal elbTotal eubTotal lbTotal ubTotal if
 tin(2019m1,2021m4),
```

```
      > ///
      >         scheme(s1mono) tline(2021m3, lcolor(gs4)) ///
      >         lpattern(solid solid longdash longdash shortdash shortdash) ///
      >         lcolor(dkorange gs5 gs10 gs10 dkorange%60 dkorange%60) ///
      >         lwidth(medthick medthick medium medium)

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

   230 .
   231 . histogram expres, normal kdensity saving(residuals.gph, replace)
      (bin=19, start=.85658383, width=.00905468)
      (file residuals.gph saved)

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

   233 .
   234 . log close
            name:  <unnamed>
             log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Probl
      > em Sets/Final Exam/Final Exam.smcl
        log type:  smcl
       closed on:  29 Apr 2021, 12:09:27
      ------------------------------------------------------------------------
------
```