

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

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

- 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 β 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: $\mathbf{y} = \mathbf{x}\beta + v$

- Keep track of the dimensions

First order condition of Applying RSS

- OLS estimators are residual sum squares (RSS)

$$\frac{\partial \text{RSS}}{\partial \beta_j} = 0 \Rightarrow \sum_{i=1}^n x_{ij} \hat{v}_i = 0, \quad (j = 0, 1, \dots, k)$$

↳ \hat{v}_i = residual

- System of $k+1$ equations written as $\mathbf{x}'\hat{v} = 0 \rightarrow (k+1) \times 1$ vector of \hat{v}
- ↳ transpose of \mathbf{x}

OLS Estimators in Matrix Form

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

$$\mathbf{x}'\mathbf{v} = 0$$

$$\mathbf{x}'(\mathbf{y} - \mathbf{x}\hat{\beta}) = 0$$

$$\mathbf{x}'\mathbf{y} = (\mathbf{x}'\mathbf{x})\hat{\beta}$$

$$\hat{\beta} = (\mathbf{x}'\mathbf{x})^{-1}(\mathbf{x}'\mathbf{y})$$

An important result

$$\hat{\beta} = (\mathbf{x}'\mathbf{x})^{-1}(\mathbf{x}'\mathbf{y}) = (\mathbf{x}'\mathbf{x})^{-1}(\mathbf{x}'(\mathbf{x}\beta + \mathbf{v})) = \beta + (\mathbf{x}'\mathbf{x})^{-1}(\mathbf{x}'\mathbf{v})$$

- $\hat{\beta}$ in general differs from β due to the error \mathbf{v}

- β is an unknown constant

- Distribution of $\hat{\beta}$ 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 $\hat{\beta}$ 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

$$E((\hat{\beta} - \beta)(\hat{\beta} - \beta)'|x) = \sigma^2(\mathbf{x}'\mathbf{x})^{-1}$$

Heteroskedasticity

$$E((\hat{\beta} - \beta)(\hat{\beta} - \beta)'|x) = (\mathbf{x}'\mathbf{x})^{-1}(\mathbf{x}'\Sigma x)(\mathbf{x}'\mathbf{x})^{-1}$$

Σ = diagonal matrix

White Sandwich Estimator

$$\mathbf{x}'\Sigma x = \sum_{i=1}^n \hat{v}_i^2 x_i' x_i$$

$$(\mathbf{x}'\mathbf{x})^{-1}(\mathbf{x}'\Sigma x)(\mathbf{x}'\mathbf{x})^{-1}$$

Predicted Values

$$\rho = \mathbf{x}(\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'$$

↳ Projection matrix

$$\rho = \rho' \quad \rho\rho = \rho \quad \rho x = x$$

Residuals

$$\hat{v} = y - \hat{y} = (\mathbf{I} - \rho)y = \mathbf{m}y$$

$$\mathbf{m} = \mathbf{I} - \rho$$

$$\mathbf{m}' = \mathbf{m} \quad \mathbf{m}\mathbf{m}' = \mathbf{m} \quad \mathbf{m}\mathbf{m}' = \mathbf{0}$$

Frisch-Waugh Theorem I

1.1 Population Regression Function

Wednesday, January 13, 2021 9:23 PM

PRF

Expected Value = $E(x) = \sum f_i x_i = \int_{-\infty}^{\infty} u f(u) du$
Random Variable?

↑
Probability density function



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

$E(Y|X)$ = linear function

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

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

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

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

SPECIFICATION ERROR: shape of approx \neq shape of PRF

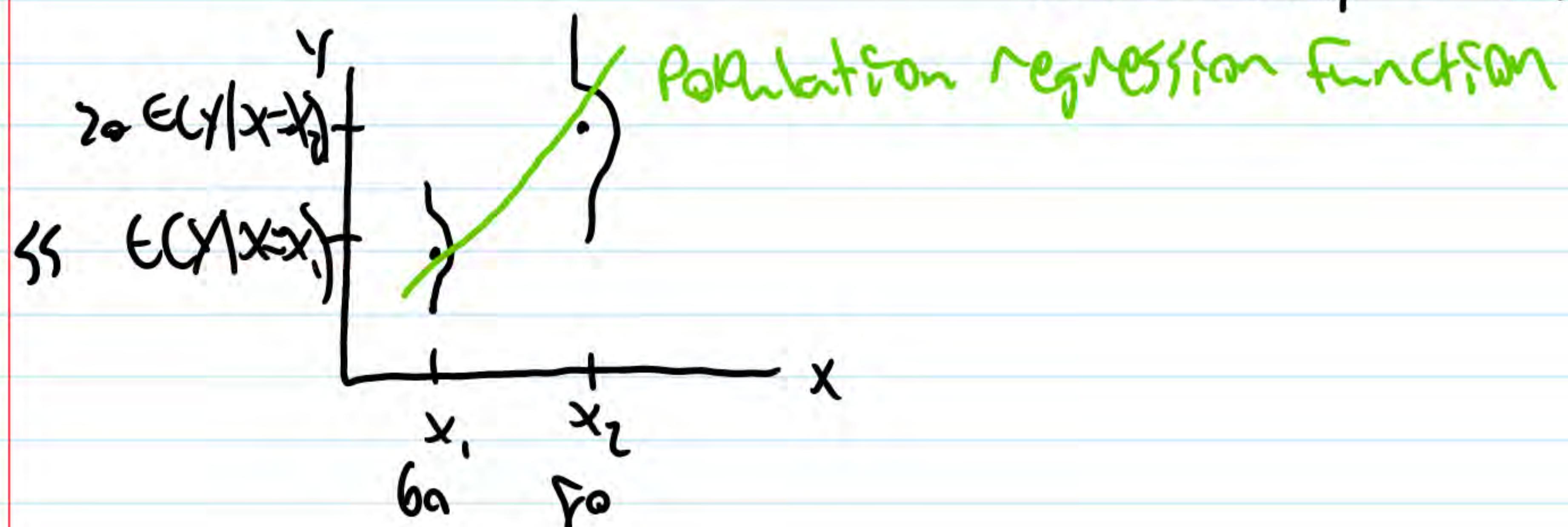
1.1 Population Regression Function

Wednesday, January 13, 2021 9:23 PM

PRF

Expected Value = $E(x) = \sum f_i x_i = \int_{-\infty}^{\infty} u f(u) du$
Random Variable?

↑
Probability density function



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

$E(Y|X)$ = linear function

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

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

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

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

SPECIFICATION ERROR: shape of approx \neq shape of PRF

1.2 Data in Matrix Form

Wednesday, January 13, 2021 9:52 PM

Data

y	x_1	x_2	\dots	x_k
70	60	50		
73	50	60		
75	75	\vdots		
70	90	\vdots		
\vdots	\vdots			
\vdots	\vdots			

$$x_0 = 1 \text{ for all}$$

$$Y \quad X$$

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

$$y_i = \mathbb{E}(y_i | x_i) + \epsilon_i$$

Predictable

$$\mathbb{E}(y | x) = x\beta$$

\hookrightarrow vector of unknowns
 \hookrightarrow x matrix

$$y = x\beta + \epsilon$$

\hookrightarrow residual matrix

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots \\ 1 & x_{21} & \vdots & \vdots \\ 1 & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$\vdots \epsilon_1, \epsilon_2, \dots, \epsilon_n$

$$y = x\beta + \epsilon$$

$$y_1 = 1 \cdot \beta_0 + x_{11} \beta_1 + x_{12} \beta_2 + \dots + x_{1k} \beta_k + \epsilon_1$$

$$\dots y_n = \dots + \epsilon_n$$

1.3 Estimating the Betas

Wednesday, January 13, 2021 10:03 PM

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

$$\hat{\eta}_i = y_i - \sum_j \beta_j x_{ij}$$

$$\hat{\eta}_i^2 = (y_i - \sum_j \beta_j x_{ij})^2$$

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

Sample

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

$\hat{\beta}$ is best estimate of β

\hat{y} is best prediction of y

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

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

Min Sample SSR OLS

1.4 Minimizing the SSR

Wednesday, January 13, 2021 10:13 PM

$$\frac{dSSR}{d\beta_0} = \left(2 \sum_i (y_i - \sum_j \beta_j x_{ij}) \right) (-1) = 0$$

$$\sum_i y_i = \sum_i \sum_j \beta_j x_{ij}$$

$$\frac{\sum_i y_i}{n} = \bar{y}$$

$$\sum_i y_i x_{i1} = \sum_i \bar{y} x_{i1}$$

1.5 The Normal Equations

Wednesday, January 13, 2021 10:19 PM

$$\frac{dSSR}{d\beta_j} = -2 \sum_i (\underbrace{y_i - \bar{y}}_{f_i}) x_{ij} = 0$$

$$\sum_i f_i \cdot x_{i0} = 0$$

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

$$\sum f_i = 0$$

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

$$\bar{Y}/n = (\sum \hat{\beta}_0 + \beta_1 x_1 + \hat{\beta}_2 x_2 + \dots) / n$$

$$\bar{Y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}_1 + \hat{\beta}_2 \bar{x}_2 + \dots$$

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

β = slope or population regression function

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

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

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

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

$$x' = x^{\text{Transpose}} = x^T$$

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

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

$$(x^T x)^{-1} x^T Y = (x^T x)^{-1} x^T x \hat{\beta}$$

$$\hat{\beta} = (x^T x)^{-1} x^T y$$

→ unknown pop. values

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

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

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

$\hookrightarrow \epsilon = 0$ in population

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

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

$$\frac{E((\hat{\beta} - \beta)^2)}{E((x^T x)^{-1} x^T \epsilon^2)} = \frac{E(\epsilon^2)}{(x^T x)^{-1} x^T \epsilon^2 x (x^T x)}$$

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

$$\epsilon \epsilon^T = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \begin{bmatrix} \epsilon_1 & \epsilon_2 & \dots & \epsilon_n \end{bmatrix} = n \times n \text{ matrix}$$

$$E(\epsilon \epsilon^T | x) = ?$$

Estimating the Sigma Matrix

$$E(\epsilon_i \epsilon_j | x) = 0$$

- 1) No spec error
- 2) Random sample

$$\Sigma = E[\text{diagonal matrix}]$$

$$\Sigma = [\text{diagonal hat matrix}] \rightarrow \text{Assume homoskedasticity}$$

$$E(\epsilon_i^2) = E(\epsilon_j^2)$$

$$\forall i = j$$

$$E(\epsilon_i^2) = \sigma^2 = \sigma_i^2 = \sigma^2$$

I = identity matrix

Homoskedasticity

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

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

→ default

= diagonal = variance → off diagonal = covariance

Sandwich estimator

Eicker-Huber-White estimator

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

hard to calculate

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

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

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

$$\left[\begin{array}{ccc} \widehat{\text{Var}(\hat{\beta}_1)} & \widehat{\text{Var}(\hat{\beta}_2)} & \widehat{\text{Var}(\hat{\beta}_n)} \end{array} \right]$$

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

$$\hookrightarrow s^2 = [\sum_i \hat{e}_i^2] / [N - k - 1]$$

$\hookrightarrow \hat{R}_j = \text{regress } x_j \text{ on all other } x_i (i \neq j) \text{ to } SSt_j$

$$\hookrightarrow SSt_j = \sum_i (x_{ij} - \bar{x}_{ij})^2 \rightarrow \text{total variability in } x_j$$

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

2.3 Understanding Coefficient Variance

Thursday, January 21, 2021 9:54 PM

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

$$\left[\begin{matrix} \widehat{\text{Var}(\hat{\beta}_1)} & \widehat{\text{Var}(\hat{\beta}_2)} & \widehat{\text{Var}(\hat{\beta}_n)} \end{matrix} \right]$$

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

$$\hookrightarrow s^2 = [\sum_i \hat{e}_i^2] / [N - k - 1]$$

$\hookrightarrow \hat{R}_j = \text{regress } x_j \text{ on all other } x_i (i \neq j) \text{ to } SST_j$

$$\hookrightarrow SST_j = \sum_i (x_{ij} - \bar{x}_j)^2 \rightarrow \text{total variability in } x_j$$

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

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 an 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

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

Precision vs accuracy/bias

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 Models10-2a 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

β_0 is the impact propensity or impact multiplier

long 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 Assumptions10-3a 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
 $\Rightarrow \text{Var}(u_t) = \sigma^2, t=1, 2, \dots, n$

No Serial Correlation:

Conditional on x , errors in two times are uncorrelated
 $\Rightarrow \text{Corr}(u_t, u_s | x) = 0, \text{ for all } t \neq s$

\hookrightarrow suffer from serial/autocorrelation when false

10-3c Inference Under the Classical Linear Model Assumptions

Normality:

errors u_t are independent of x and are independently and identically distributed as $\text{Normal}(0, \sigma^2)$

1 Introduction

Econometrics started with time series data

2 Overview

Core models to learn:

1) Autoregressive model (AR) $\Rightarrow Y_t = \alpha + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \epsilon_t$

2) Regression model $\Rightarrow Y_t = \alpha + \delta_0 X_t + \epsilon_t$

3) Distributed Lag (DL) model

$\hookrightarrow Y_t = \alpha + \delta_0 X_t + \delta_1 X_{t-1} + \dots + \delta_q X_{t-q} + \epsilon_t$

4) Autoregressive-Distributed Lag (ARDL) model

$\hookrightarrow Y_t = \alpha + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \delta_0 X_t + \delta_1 X_{t-1} + \dots + \delta_q X_{t-q} + \epsilon_t$

$p = \#$ lags of dependent var Y_t

$q = \#$ of lags of explanatory var X_t

ϵ_t = mean-zero shock

1M! Core insights!

3 Standard Errors and t statistics

Newey standard errors appropriate for simple (non-dynamic) regressions and DL models

\hookrightarrow distributed lag

base 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

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

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

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

$$\begin{matrix} \text{rev}_t = \beta_0 + \beta_1 \text{GDP}_t + \epsilon_t \\ \text{exp}_t = \beta_0 + \beta_1 \text{GDP}_t + \epsilon_t \end{matrix} \rightarrow \text{MGOPC} = \text{gdp} / \text{gdPd} \cdot \text{Pop}$$

↑
revenue/expenditure

expt : depend on what we want income ↑, more depends on available revenue

rev_t : avg tax rate · tax base ↑
income

10.3 Time Trends

Sunday, January 31, 2021 6:16 PM

Past affects future so std error isn't right

Serial correlations!

Linear trends by regressing time and a variable

Time trends

$$Y_t = \alpha_0 + \alpha_1 t + \epsilon_t$$

$$X_t = \gamma_0 + \gamma_1 t + \eta_t$$

$$\hat{Y}_t = \hat{\alpha}_0 + \hat{\alpha}_1 t$$

$$\hat{\epsilon}_t = (Y_t - \hat{\alpha}_0 - \hat{\alpha}_1 t)$$

Suppose $Y_t = \beta_0 + \beta_1 X_t + \delta t + \epsilon_t$

How is Y correlated with X ?

detrending = Put in t trend take residuals afterwards

differencing =

lags:

$$L.Y_t = Y_{t-1}$$

$$f.Y_t = Y_{t+1}$$

$$D.Y_t = Y_t - Y_{t-1}$$

\uparrow
differencing

$$Y_t = \beta_0 + \beta_1 X_t + \delta t + \epsilon_t$$

$$Y_{t-1} = \beta_0 + \beta_1 X_{t-1} + \delta(t-1) + \epsilon_{t-1}$$

$$D.Y_t = Y_t - Y_{t-1} = \delta + \beta_1(X_t - X_{t-1}) + (\epsilon_t - \epsilon_{t-1})$$

10.4 Seasonal Trends

Sunday, January 31, 2021 7:29 PM

add quarterly dummy variables to adjust for seasons

$$Y_t = \beta_0 + \beta_1 X_t + \delta_t + \gamma_{sp} + \gamma_{sum} + \gamma_{fall} + \gamma_t$$

↓
Seasons

Why does lag matter with seasons?

$$\hat{Y}_t = -0.46 - 0.22X_t + 0.65X_{t-1} - 0.003t + 0.04sp + 0.01sun + 0.02fall$$

↑
lag
↑
Spring
↑
Summer
↑
Fall

maybe lag matters more?

$$\hat{Y} = .15X_t + 1.56X_{t-1} + .69X_{t-2} - .63X_{t-3} - 1.5X_{t-4}$$

4 lags is one cycle

SPC install estout

lag: (X/Y) ,
 ↑
 ↓
 dot
 ← time X to Y

esttab → recommended for exporting

different # observations because data is binned to calculate lag

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

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

Intercept

$$= \alpha + \sum \beta_e X_{t-e}$$

$$Y_t = \text{Total } N$$

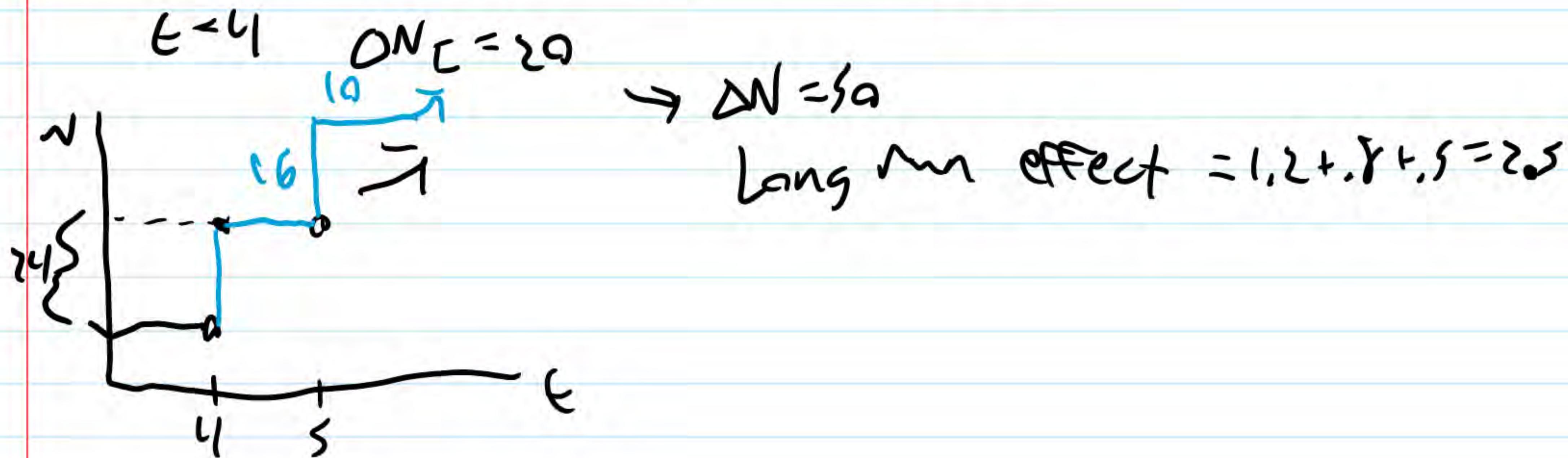
$$X_t = \text{Emp by industry of interest}$$

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

\uparrow takes 3 years

$$N_t = 100 + 1.2 N_{It} + 0.8 N_{It-1} + 0.5 N_{It-2}$$

$$\Delta N_t = 1.2 \Delta N_{It} + 0.8 \Delta N_{It-1} + 0.5 \Delta N_{It-2}$$



Advanced Time Series Topics

17.3 Spurious Regression

There's an extraneous reason for correlation

17.2 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 H_0 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

Section
<u>3 Static Model</u>
<u>3a</u>
<u>3b</u>
<u>3c</u>
<u>3d</u>
<u>3e</u>
<u>4 Finite Distributed Lag Model</u>
<u>4a</u>
<u>4b</u>
<u>4d</u>
<u>4e</u>
<u>Appendix A</u>
<u>Appendix B</u>

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_if	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_if	.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, is 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

```
1 clear
2 set more off
3
4 cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Problem
Set 1"
5
6 *2b Load the data
7 import delimited "Assignment_1_Monthly.txt"
8
9 rename lnu02300000 us_epr
10 rename flnan fl_nonfarm
11 rename fllfn fl_lf
12 rename flbpriv fl_bp
13 rename date datestring
14
15 *2c Turn on a log file
16 log using "Problem Set 1", replace
17
18 *2d Generate a monthly date variable (make its display format monthly time, %tm)
19 gen datec=date(datestring, "YMD")
20 gen date=mofd(datec)
21 format date %tm
22
23 *2e tsset your data
24 tsset date
25
26 *2f
27 gen ln_us_epr=log(us_epr)
28 gen ln_fl_nonfarm=log(fl_nonfarm)
29 gen ln_fl_lf=log(fl_lf)
30 gen ln_fl_bp=log(fl_bp)
31
32 *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.
33 regress ln_fl_nonfarm ln_fl_lf ln_us_epr ln_fl_bp
34
35 *3c Estimate the static model with month indicators and a time trend.
36 gen month=month(datec)
37 reg ln_fl_nonfarm ln_fl_lf ln_us_epr ln_fl_bp i.month date
38
39 *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.
40 regress ln_fl_nonfarm l(0/12).ln_fl_lf l(0/12).ln_us_epr l(0/12).ln_fl_bp
41
42 *4b Estimate the model in (a) but add month indicators and a time trend.
```

```

43 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

44
45 *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.
46 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
47 gen dateQ = qofd(datec)
48 format dateQ %tq
49 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
50
51 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
Total	10.7661015	395	.027255953	Adj R-squared	=	0.9784
				Root MSE	=	.02425

ln_fl_nonfarm	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
Total	10.7661015	395	.027255953	Adj R-squared	=	0.9911
				Root MSE	=	.01557

ln_fl_nonfarm	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	Number of obs	=	384
Model	9.45063897	39	.242324076	F(39, 344)	=	1506.36
Residual	.055338456	344	.000160868	Prob > F	=	0.0000
Total	9.50597742	383	.024819784	R-squared	=	0.9942
				Adj R-squared	=	0.9935
				Root MSE	=	.01268

ln_fl_nonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
ln_fl_lf					
---	-.3180953	.2192272	-1.45	0.148	-.7492898
L1.	-.4936055	.2780395	-1.78	0.077	-1.040477
L2.	.3085466	.27846	1.11	0.269	-.239152
L3.	1.173922	.2948363	3.98	0.000	.5940134
L4.	-.2346487	.2905929	-0.81	0.420	-.8062113
L5.	.2808166	.2958343	0.95	0.343	-.3010552
L6.	-.2076341	.3372426	-0.62	0.539	-.8709511
L7.	.428488	.3391507	1.26	0.207	-.2385821
L8.	.4803611	.3332665	1.44	0.150	-.1751354
L9.	.2977526	.3112925	0.96	0.339	-.3145235
L10.	-.00028	.3217814	-0.00	0.999	-.6331867
L11.	-.5860114	.3256137	-1.80	0.073	-1.226456
L12.	.0176351	.2499574	0.07	0.944	-.4740021
ln_us_epr					
---	1.180441	.1573579	7.50	0.000	.8709364
L1.	.2435207	.202013	1.21	0.229	-.1538155
L2.	-.1519264	.2015081	-0.75	0.451	-.5482695
L3.	-.719111	.2119425	-3.39	0.001	-1.135977
L4.	.1877102	.2014654	0.93	0.352	-.2085489
L5.	-.1596306	.206881	-0.77	0.441	-.5665414
L6.	.4937537	.2396216	2.06	0.040	.0224458
L7.	-.3031484	.236988	-1.28	0.202	-.7692764
L8.	-.2995254	.2312056	-1.30	0.196	-.7542801
L9.	.5953076	.2915942	2.04	0.042	.0217756
L10.	-.1656984	.352639	-0.47	0.639	-.8592984
L11.	.5326939	.3523697	1.51	0.132	-.1603764
L12.	-.4280274	.2543508	-1.68	0.093	-.928306
ln_fl_bp					
---	.0177815	.0051888	3.43	0.001	.0075758
L1.	.0056999	.0054688	1.04	0.298	-.0050566
L2.	.0123023	.0056879	2.16	0.031	.0011149
L3.	-.0005041	.0058381	-0.09	0.931	-.0119871
L4.	-.0040248	.0058282	-0.69	0.490	-.0154881
L5.	.0053648	.0058106	0.92	0.357	-.006064
L6.	.0122019	.0057914	2.11	0.036	.0008108
L7.	.0146252	.0057698	2.53	0.012	.0032766
L8.	.0114715	.0057663	1.99	0.047	.0001299
L9.	.0100892	.0057895	1.74	0.082	-.0012981
L10.	-.0077443	.0056515	-1.37	0.171	-.0188601
L11.	-.0129284	.0055227	-2.34	0.020	-.0237908
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
Model	9.47318331	51	.185748692	F(51, 332)	=	1880.48
Residual	.03279411	332	.000098777	Prob > F	=	0.0000
Total	9.50597742	383	.024819784	R-squared	=	0.9966
				Adj R-squared	=	0.9960
				Root MSE	=	.00994

ln_fl_nonfarm	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.	-.0015401	.0049271	-0.31	0.754	-.011261 .0081628

L8.	-0.001594	.00049974	-0.31	0.117	-0.011204	.00010200
L9.	.0010036	.0048898	0.21	0.838	-0.0086153	.0106225
L10.	-0.004773	.0047767	-0.10	0.920	-0.0098737	.008919
L11.	-0.0083937	.0046846	-1.79	0.074	-0.017609	.0008216
L12.	-0.0041455	.0044702	-0.93	0.354	-0.0129391	.004648
month						
2	.0077995	.0048077	1.62	0.106	-0.001658	.017257
3	.0052085	.0041637	1.25	0.212	-0.0029821	.0133991
4	-0.010198	.0053356	-0.19	0.849	-0.0115156	.009476
5	-0.0012298	.0047478	-0.26	0.796	-0.0105694	.0081098
6	-0.0122415	.0055844	-2.19	0.029	-0.0232267	-0.0012563
7	-0.0240128	.0047031	-5.11	0.000	-0.0332644	-0.0147612
8	-0.0152756	.0052483	-2.91	0.004	-0.0255997	-0.0049514
9	-0.0111308	.0045365	-2.45	0.015	-0.0200548	-0.0022068
10	-0.0046899	.006722	-0.70	0.486	-0.0179129	.0085332
11	.0076979	.0057763	1.33	0.184	-0.0036649	.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
Model	9.46703767	24	.394459903	F(24, 359)	=	3636.67
Residual	.038939751	359	.000108467	Prob > F	=	0.0000
Total	9.50597742	383	.024819784	R-squared	=	0.9959
				Adj R-squared	=	0.9956
				Root MSE	=	.01041

ln_fl_nonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
ln_fl_lf					
--	.2198644	.118892	1.85	0.065	-0.0139479 .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
Model	10.2740552	27	.380520563	F(27, 364)	=	2505.01
Residual	.055292923	364	.000151904	Prob > F	=	0.0000
Total	10.3293481	391	.02641777	R-squared	=	0.9946
				Adj R-squared	=	0.9942
				Root MSE	=	.01232

ln_fl_nonfarm	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

Section
<u>Part A</u>
<u>Part B</u>
<u>3 Autocorrelation and Weak Dependence</u>
<u>4 ARDL Model and Breusch-Godfrey Test</u>
<u>5 Dynamically Complete Models and Newey-West Standard Errors</u>
<u>Appendix A</u>
<u>Appendix B</u>

Part A

1. Write the model $y_t = \alpha + \delta t + \rho y_{t-1} + \beta x_{t-1} + r$ 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.)

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

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.

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

Part B

3. Autocorrelation and Weak Dependence

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

<u>Variable</u>	<u>Correlation with Lag</u>
<u>Inflnonfarm</u>	<u>.9981</u>
<u>Infllf</u>	<u>.9994</u>
<u>lnusepr</u>	<u>.9821</u>
<u>Inflbp</u>	<u>.9477</u>

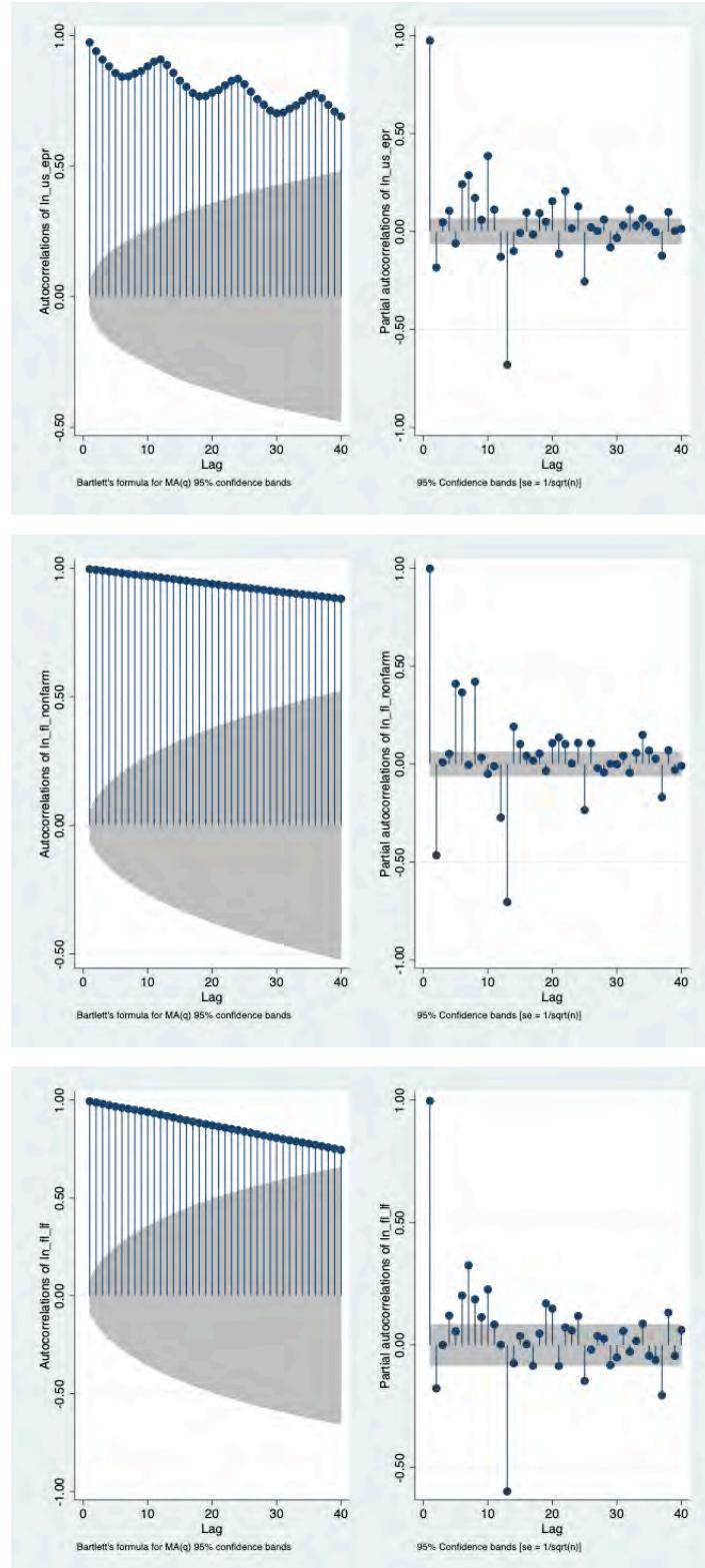
(obs=875)	corr ln_us_epr l1.ln_us_epr
	L.
	ln_us~r ln_us~r
ln_us_epr	
-.	1.0000
L1.	0.9758 1.0000

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

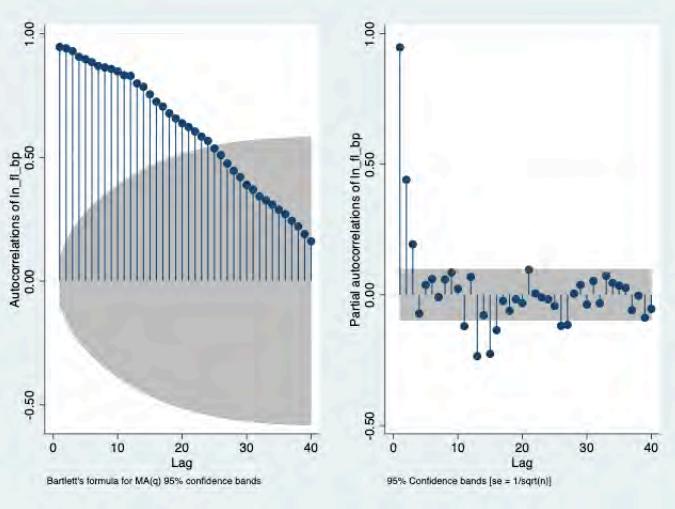
(obs=539)	corr ln_fl_if l1.ln_fl_if
	L.
	ln_fl_if ln_fl_if
ln_fl_if	
-.	1.0000
L1.	0.9997 1.0000

(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_if, ln_us_epr, and ln_fl_bp with .9997, .9758, and .9470 respectively.
2. Obtain the autocorrelogram and partial autocorrelogram 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 test for each variable.

<u>Variable</u>	<u>Dickey-Fuller p-value</u>
<u>Inflnonfarm</u>	<u>.0328</u>
<u>Infllf</u>	<u>.6285</u>
<u>lnusepr</u>	<u>.2246</u>
<u>Inflbp</u>	<u>.7774</u>

```
. dfuller ln_us_epr, trend regress
Dickey-Fuller test for unit root
Number of obs = 875
Test Statistic      Interpolated Dickey-Fuller
1% Critical Value      -3.960      5% Critical Value      -3.410      10% Critical Value      -3.120
Z(t)      -4.020
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

Test Statistic	Interpolated Dickey-Fuller		
	1% Critical Value	5% Critical Value	10% Critical Value
Z(t)	-0.653	-3.960	-3.410

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

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

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 below, illustrating the differences are clearly not I(1)."

4. ARDL Model and Breusch-Godfrey Test

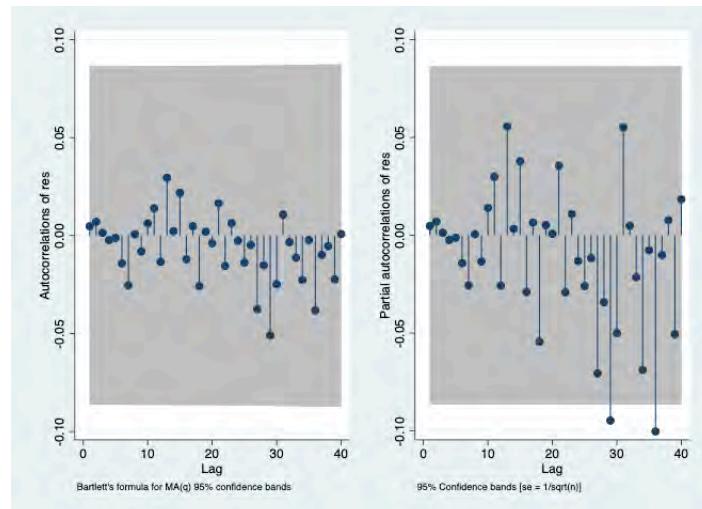
Given the results of the previous question, transform the data as needed and estimate a dynamically complete ARDL model for non-farm employment. Include at least one lag of the relevant dependent variable. How many additional lags of the dependent variable, and how many lags of which independent variables 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.

.	<code>regress d.In_fl_nonfarm l(1/48)d.In_fl_nonfarm</code>	<code>l(12/24)d.In_us_epr</code>	<code>l(1/18, 24)d.In_fl_if</code>	<code>date</code>
	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.				
In_fl_nonfarm	Coef.	Std. Err.	t	P>t
In_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

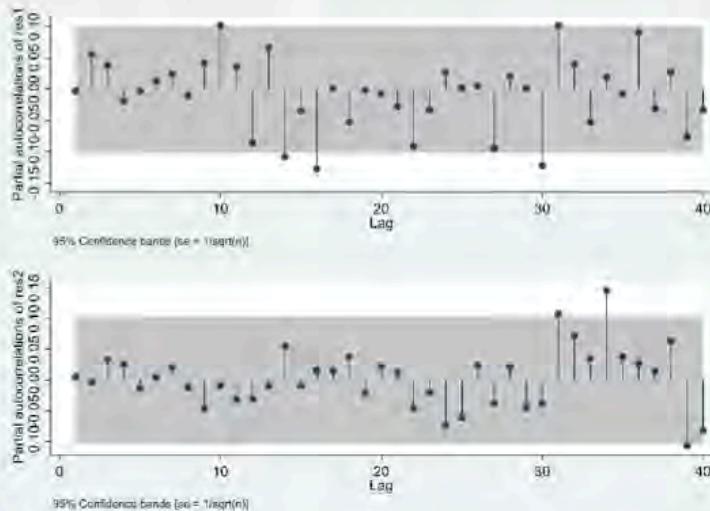
lags(p)	chi2	df	Prob > chi2
1	0.617	1	0.4321
2	1.630	2	0.4427

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

Lags	$p > \chi^2$	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
Model	.00146591	5	.000293182	F(5, 374)	=	2.97
Residual	.036972226	374	.000098856	Prob > F	=	0.0122
Total	.038438136	379	.00010142	R-squared	=	0.0381
				Adj R-squared	=	0.0253
				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
maximum lag: 4
Number of obs = 380
F(5, 374) = 4.01
Prob > F = 0.0015

D. ln_fl_nonf~m	Newey-West					
	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*** (0.00203)	0.00820** (0.00250)
LD.lnflbp	0.00793*** (0.00236)	0.00793** (0.00294)
L2D.lnflbp	0.00627* (0.00244)	0.00627 (0.00348)
L3D.lnflbp	0.00730** (0.00237)	0.00730* (0.00300)
L4D.lnflbp	0.00430* (0.00204)	0.00430* (0.00199)
<i>N</i>	379	379
<i>R</i> 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

```

1 clear
2 set more off
3
4 cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Problem
Set 2"
5
6 *2a
7 *Done
8
9 *2b Load the data
10 import delimited "Assignment_1_Monthly.txt"
11
12 rename lnu02300000 us_epr
13 rename flnan fl_nonfarm
14 rename fllfn fl_lf
15 rename flbpriv fl_bp
16 rename date datestring

```

```

17
18 *2c Turn on a log file
19 log using "Problem Set 1", replace
20
21 *2d Generate a monthly date variable (make its display format monthly time, %tm)
22 gen datec=date(datestring, "YMD")
23 gen date=mofd(datec)
24 format date %tm
25
26 *2e tsset your data
27 tsset date
28
29 *2f
30 gen ln_us_epr=log(us_epr)
31 gen ln_fl_nonfarm=log(f1_nonfarm)
32 gen ln_fl_lf=log(f1_lf)
33 gen ln_fl_bp=log(f1_bp)
34
35 *3a
36 corr ln_us_epr l1.ln_us_epr
37 corr ln_fl_nonfarm l1.ln_fl_nonfarm
38 corr ln_fl_lf l1.ln_fl_lf
39 corr ln_fl_bp l1.ln_fl_bp
40
41 *3b
42 ac ln_us_epr, saving(ac_ln_us_epr.gph, replace)
43 pac ln_us_epr, saving(pac_ln_us_epr.gph, replace)
44 graph combine ac_ln_us_epr.gph pac_ln_us_epr.gph, saving(combo_ln_us_epr.gph,
replace)
45
46 ac ln_fl_nonfarm, saving(ac_ln_fl_nonfarm.gph, replace)
47 pac ln_fl_nonfarm, saving(pac_ln_fl_nonfarm.gph, replace)
48 graph combine ac_ln_fl_nonfarm.gph pac_ln_fl_nonfarm.gph,
saving(combo_ln_fl_nonfarm.gph, replace)
49
50 ac ln_fl_lf, saving(ac_ln_fl_lf.gph, replace)
51 pac ln_fl_lf, saving(pac_ln_fl_lf.gph, replace)
52 graph combine ac_ln_fl_lf.gph pac_ln_fl_lf.gph, saving(combo_ln_fl_lf.gph, replace)
53
54 ac ln_fl_bp, saving(ac_ln_fl_bp.gph, replace)
55 pac ln_fl_bp, saving(pac_ln_fl_bp.gph, replace)
56 graph combine ac_ln_fl_bp.gph pac_ln_fl_bp.gph, saving(combo_ln_fl_bp.gph, replace)
57
58 *3c
59 dfuller ln_us_epr, trend regress
60 dfuller ln_fl_nonfarm, trend regress
61 dfuller ln_fl_lf, trend regress
62 dfuller ln_fl_bp, trend regress
63
64 *4

```

```

65 regress d.ln_fl_nonfarm l(1/48)d.ln_fl_nonfarm l(12/24)d.ln_us_epr l(1/18,
66 24)d.ln_fl_lf date
67 predict res, residual
68 ac res, saving(p4_ac.gph, replace)
69 pac res, saving(p4_pac.gph, replace)
70 graph combine p4_ac.gph p4_pac.gph, saving(p4_combo.gph, replace)
71 estat bgodfrey, lag(1/48)
72
73 *5
74 reg d.ln_fl_nonfarm l(0/4)d.ln_fl_bp if tin(1948m1,2020m1)
75 newey d.ln_fl_nonfarm l(0/4)d.ln_fl_bp if tin(1948m1,2020m1), lag(4)
76 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(f1_nonfarm)

. gen ln_fl_lf=log(f1_lf)
(444 missing values generated)

. gen ln_fl_bp=log(f1_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   |          |


```

```

--. | 1.0000
L1. | 0.9997  1.0000

. corr ln_fl_bp l1.ln_fl_bp
(obs=395)

. *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 Statistic          1% Critical Value          5% Critical Value          10% Critical 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   | -.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 Statistic          1% Critical Value          5% Critical Value          10% Critical 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					
	Interpolated Dickey-Fuller				
Test Statistic	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					
	Interpolated Dickey-Fuller				
Test Statistic	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

```
*  
. *  
. 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	Number of obs	=	515
Model	.050091055	81	.000618408	F(81, 433)	=	15.48
Residual	.01729335	433	.000039938	Prob > F	=	0.0000
Total	.067384405	514	.000131098	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	.000728	-2.19	0.029	-.2525689 -.0138524
L3D.	.0520745	.000831	0.86	0.392	-.0674863 .1716354
L4D.	.1139409	.0669067	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	.0904647	.489089
L25D.	.1178616	.098203	1.20	0.231	-.0751523	.3108754
L26D.	.0999885	.0980021	1.02	0.308	-.0926304	.2926674
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	.8242834
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	-.1058899	.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	-.2422684	.1449445
L39D.	.0871388	.0975069	0.89	0.372	-.194507	.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
<hr/>						
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	-.2466498	.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
<hr/>						
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.	-.2278642	.0723796	-3.14	0.002	-.3693234	-.0848851
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.	-.188737	.0797732	-2.27	0.024	-.3375278	-.0239463
L11D.	-.005726	.0880895	-0.07	0.944	-.1645537	.1531817
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.	.1252027	.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	0.5500

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
Model	.00146591	5	.000293182	F(5, 374)	=	2.97
Residual	.036972226	374	.000098856	Prob > F	=	0.0122
Total	.038438136	379	.00010142	R-squared	=	0.0381
				Adj R-squared	=	0.0253
				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. ln_fl_nonf~m	Newey-West				
	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?

- $List_t = \beta_0 + \beta_P \ln Permits_{t-1} + \beta_{Int} \ln Interest_{t-1} + \beta_{Inf} \ln Inflation_{t-1} + r_t$
- $\Delta List_t = \beta_0 + \beta_P \Delta \ln Permits_{t-1} + \beta_{Int} \Delta \ln Interest_{t-1} + \beta_{Inf} \Delta \ln Inflation_{t-1} + \Delta r_t$
- $\Delta List_t = \beta_0 + \beta_P \Delta \ln Permits_{t-1} + \beta_{Int} \Delta \ln Interest_{t-1} + \beta_{Inf} \Delta \ln Inflation_{t-1} + \Delta r_t$

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.

- $List_t = \beta_0 + \beta_P \ln Permits_{t-1} + \beta_{Int} \ln Interest_{t-1} + \beta_{Inf} \ln Inflation_{t-1} + r_t$
- $\Delta List_t = \underline{\beta_0} + \beta_P \ln Permits_{t-1} + \beta_{Int} \ln Interest_{t-1} + \beta_{Inf} \ln Inflation_{t-1} +$
 $\rho(List_{t-1} - \beta_0 - \beta_P \ln Permits_{t-1} - \beta_{Int} \ln Interest_{t-1} - \beta_{Inf} \ln Inflation_{t-1}) + \varepsilon_t$
- $\Delta List_t = \beta_0 + \beta_P \ln Permits_{t-1} + \beta_{Int} \ln Interest_{t-1} + \beta_{Inf} \ln Inflation_{t-1} +$
 $\rho(List_{t-1} - \beta_0 - \beta_P \ln Permits_{t-1} - \beta_{Int} \ln Interest_{t-1} - \beta_{Inf} \ln Inflation_{t-1}) + \varepsilon_t$
- 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 autoregressive 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 InList 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 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 their 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

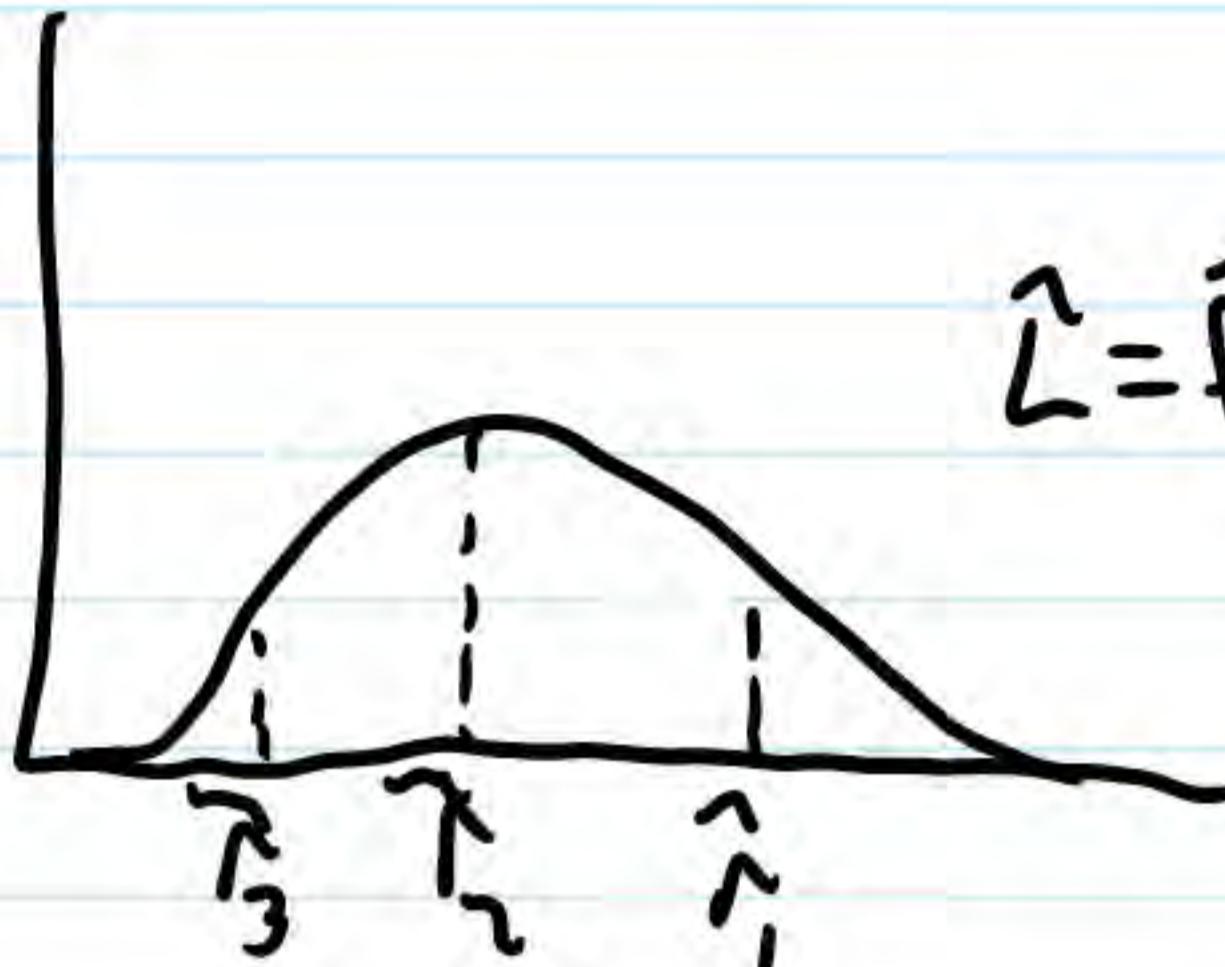
Wednesday, March 3, 2021 7:27 PM

1) domain knowledge
+ joint hypothesis tests

2) measures that guard against overfitting

↳ AIC = $2k - 2\ln \hat{L} \rightarrow$ good

↳ Bad \hat{L} = likelihood function
↳ smaller is better



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

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

↳ BIC = $\ln(T)k - 2\ln \hat{L}$

↳ smaller is better

↳ finite set of models

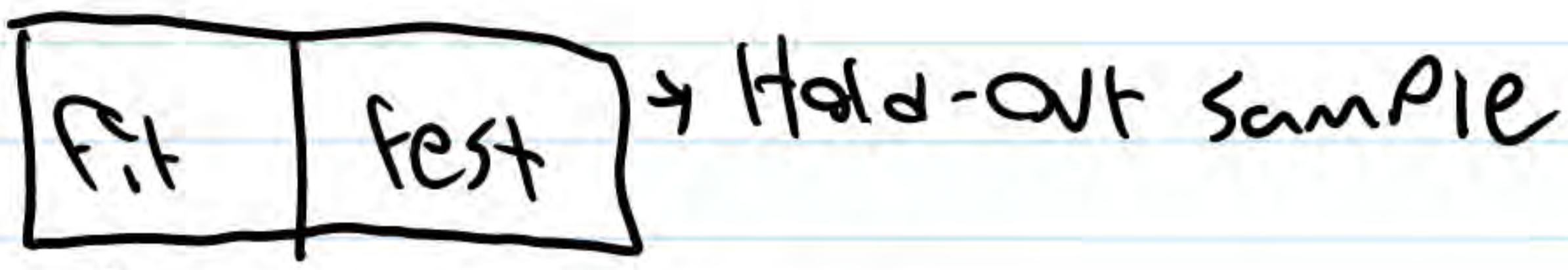
↳ tend to pick the right one at the limit

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



→ 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

OOS 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

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

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

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

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

$$\hat{N}_{f_i} = \frac{N}{k}$$

13.8 Forecasting 3

Wednesday, March 10, 2021 4:18 PM

Using differencing and log transforms

γ not $\delta\gamma$, $\ln\gamma$, or $\Delta\ln\gamma$

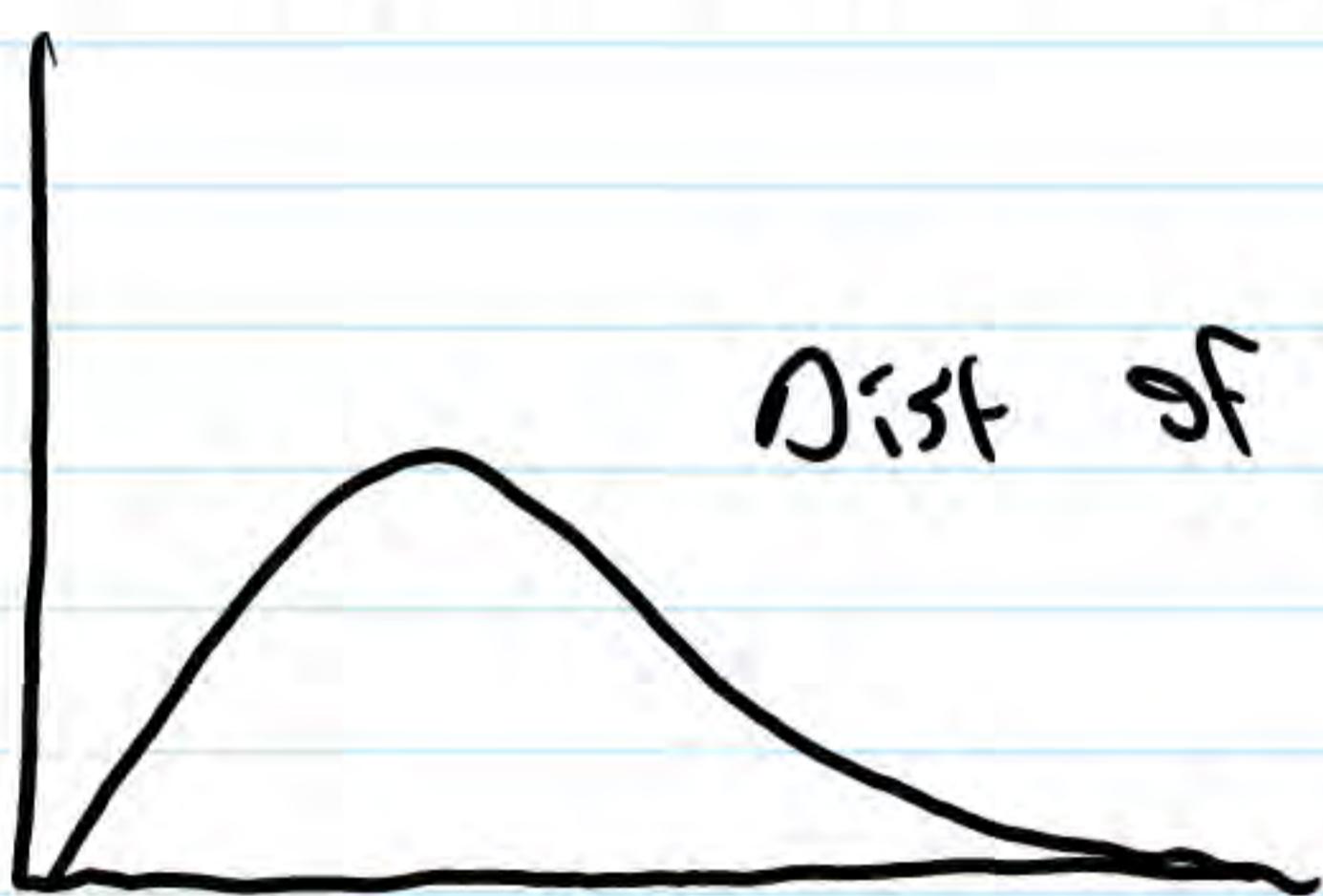
$\hat{f}_{t,h}$ $\hat{f}_{t+1} = \text{forecast at } t \text{ of } t+1$

$$\delta\gamma_{t+1} = \gamma_{t+1} - \gamma_t$$

$$\gamma_{t+1} = \gamma_t + \delta\gamma_{t+1}$$

$$\ln\gamma_{t+1} \Rightarrow \gamma_{t+1} = e^{\ln\gamma_{t+1}}$$

$$\Delta\ln\gamma_{t+1} = (\ln\gamma_{t+1} - \ln\gamma_t)$$



Dist of $y \sim \log \text{normal}$



Symmetric
↓
Dist of y if $\ln y$ is normal

$$C \widehat{\sum_{t=1}^{T-h} \epsilon_t^2} e^{\ln\gamma_t + \widehat{\Delta\ln\gamma_{t+1}}}$$

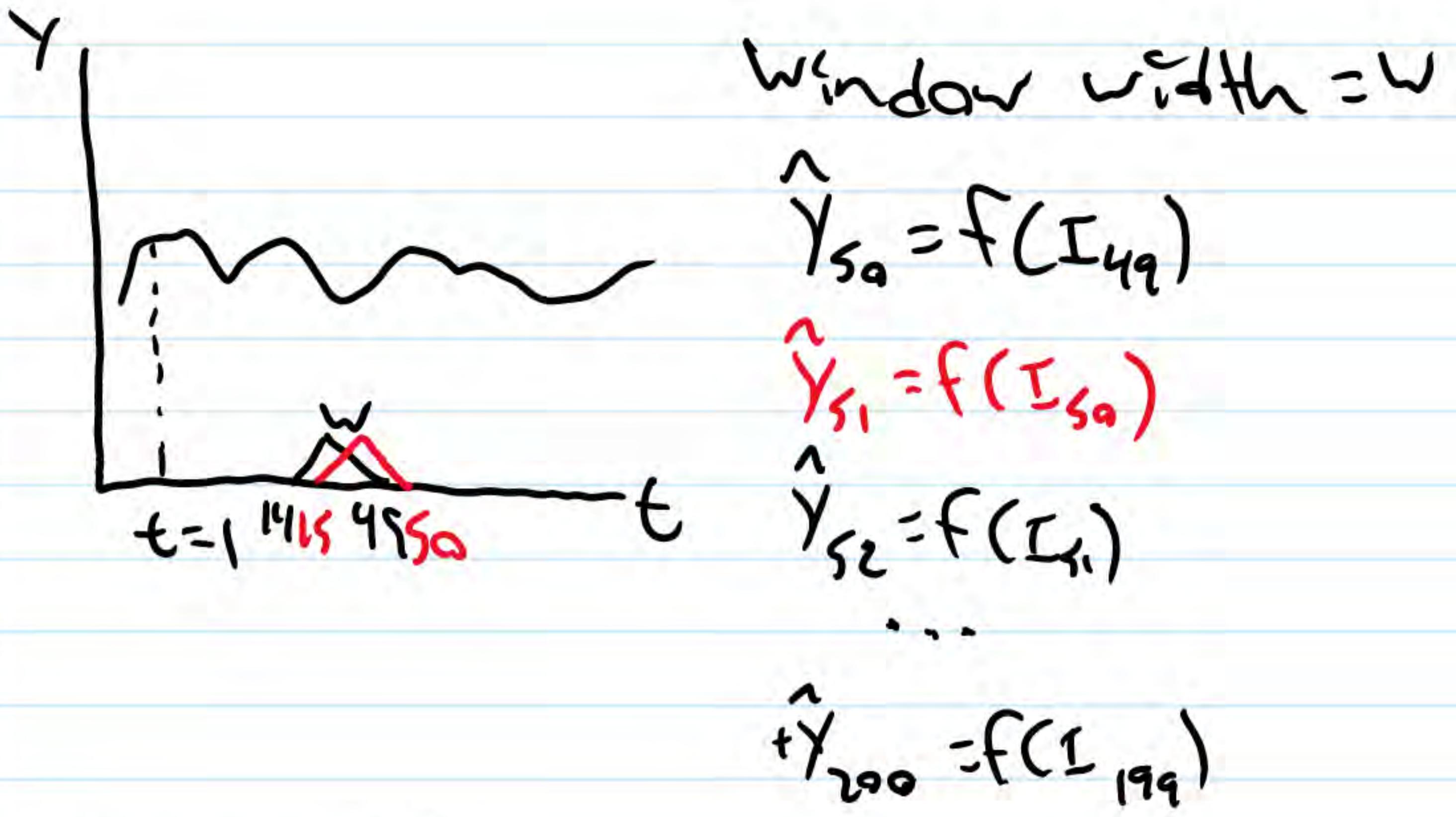
assumes Normality Empirical Version

$$\sum_i \hat{\epsilon}_i^2 / T$$

$$(e^{\ln\gamma_t + \widehat{\Delta\ln\gamma_{t+1}}}) \left(\sum_i \hat{\epsilon}_i^2 / T \right)$$

19.1 Rolling Window Estimation

Friday, March 19, 2021 9:50 AM

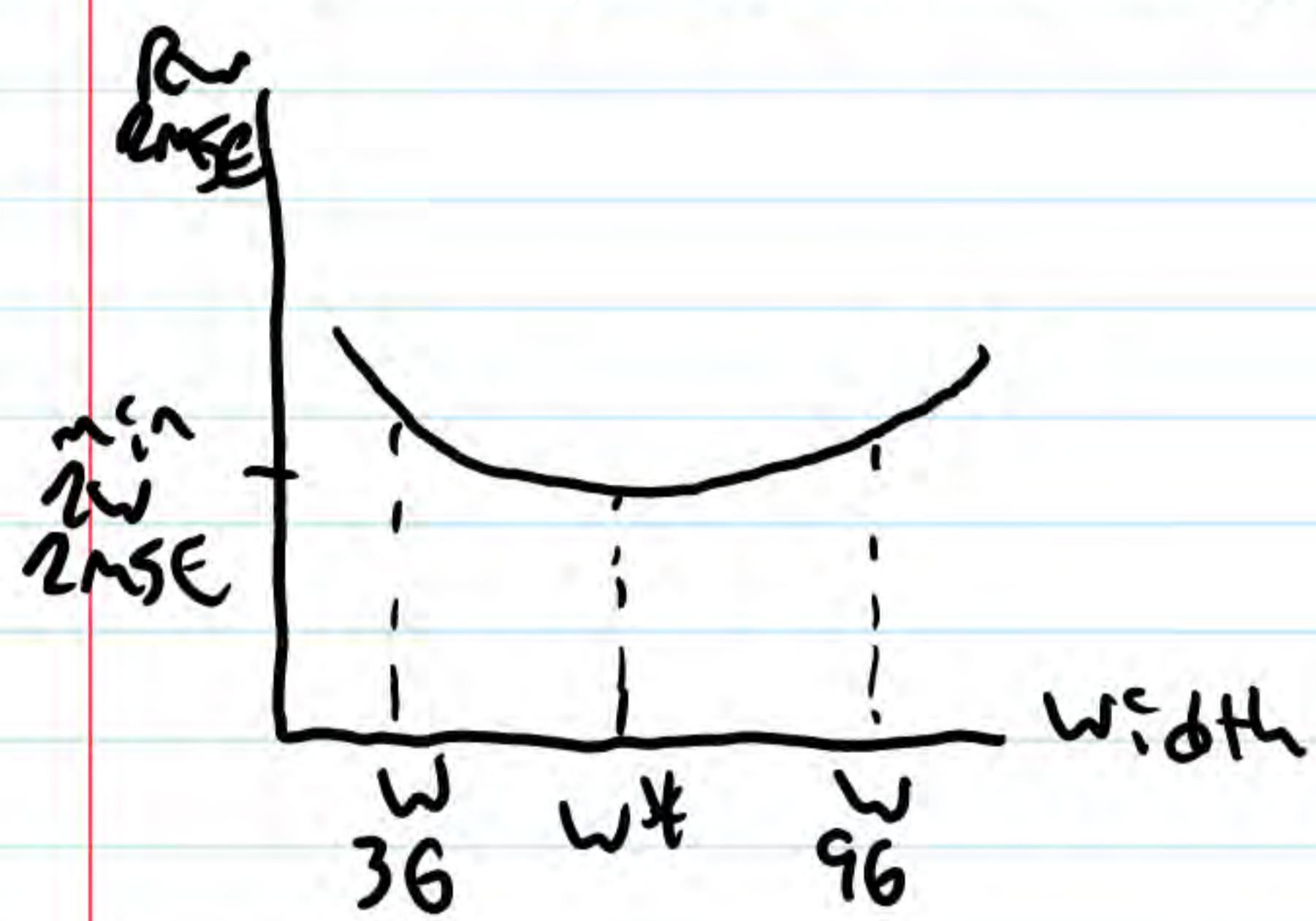


$$e_{s_0} = y_{s_0} - \hat{y}_{s_0}$$

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

$$\sqrt{\sum e^2 / n} = \text{Rolling Window RMSE}$$

$$\text{forecast interval} = \hat{y}_{\text{max}_H} \pm 1.96 \cdot \text{RWRMSE}$$



22.2 Seasonal Adjustments and Differencing

Monday, April 12, 2021 2:43 PM

Additive or multiplicative seasonal adjustments

$$Y_t = \rho_i Y_{t-1} + \beta_i X_{t-1} + \phi_{July} \cdot D_{July} + \phi_{Dec} \cdot D_{Dec}$$

\uparrow \uparrow
coefficient or a

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

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

```
1 log using "Problem Set 3", replace
2 *1a
3 clear
4 set obs 30
5 gen t=[_n]
6 tsset t
7 gen r=rnormal()
8 gen y=r if t<4
```

```

9  replace y=0.5+0.5*l.y-0.1*l2.y+0.25*l3.y+r if t>=4
10 drop r
11
12 *model
13 reg d.y l1d.y
14 *aic/bic
15 estat ic
16 scalar define df1=el(r(S),1,4)
17 scalar define aic1=el(r(S),1,5)
18 scalar define bic1=el(r(S),1,6)
19 *10 fold cv rmse
20 crossfold reg d.y l1d.y
21 scalar define k=10
22 matrix kMSE=r(est)'*r(est)
23 scalar krmse1=(el(kMSE,1,1)/k)^.5
24 scalar list krmse1
25 matrix drop kMSE
26 scalar drop k
27 *loocv
28 loocv reg d.y l1d.y
29 scalar define loormse1=r(rmse)
30
31 reg d.y l2d.y
32 estat ic
33 scalar define df2=el(r(S),1,4)
34 scalar define aic2=el(r(S),1,5)
35 scalar define bic2=el(r(S),1,6)
36 crossfold reg d.y l2d.y
37 scalar define k=10
38 matrix kMSE=r(est)'*r(est)
39 scalar krmse2=(el(kMSE,1,1)/k)^.5
40 scalar list krmse2
41 matrix drop kMSE
42 scalar drop k
43 loocv reg d.y l2d.y
44 scalar define loormse2=r(rmse)
45
46 reg d.y l3d.y
47 estat ic
48 scalar define df3=el(r(S),1,4)
49 scalar define aic3=el(r(S),1,5)
50 scalar define bic3=el(r(S),1,6)
51 crossfold reg d.y l3d.y
52 scalar define k=10
53 matrix kMSE=r(est)'*r(est)
54 scalar krmse3=(el(kMSE,1,1)/k)^.5
55 scalar list krmse3
56 matrix drop kMSE

```

```

57 scalar drop k
58 loocv reg d.y l3d.y
59 scalar define loormse3=r(rmse)
60
61 reg d.y l1d.y l2d.y
62 estat ic
63 scalar define df12=el(r(S),1,4)
64 scalar define aic12=el(r(S),1,5)
65 scalar define bic12=el(r(S),1,6)
66 crossfold reg d.y l1d.y l2d.y
67 scalar define k=10
68 matrix kMSE=r(est)/*r(est)
69 scalar krmse12=(el(kMSE,1,1)/k)^.5
70 scalar list krmse12
71 matrix drop kMSE
72 scalar drop k
73 loocv reg d.y l1d.y l2d.y
74 scalar define loormse12=r(rmse)
75
76 reg d.y l1d.y l3d.y
77 estat ic
78 scalar define df13=el(r(S),1,4)
79 scalar define aic13=el(r(S),1,5)
80 scalar define bic13=el(r(S),1,6)
81 crossfold reg d.y l1d.y l3d.y
82 scalar define k=10
83 matrix kMSE=r(est)/*r(est)
84 scalar krmse13=(el(kMSE,1,1)/k)^.5
85 scalar list krmse13
86 matrix drop kMSE
87 scalar drop k
88 loocv reg d.y l1d.y l3d.y
89 scalar define loormse13=r(rmse)
90
91 reg d.y      12d.y l3d.y
92 estat ic
93 scalar define df23=el(r(S),1,4)
94 scalar define aic23=el(r(S),1,5)
95 scalar define bic23=el(r(S),1,6)
96 crossfold reg d.y      12d.y l3d.y
97 scalar define k=10
98 matrix kMSE=r(est)/*r(est)
99 scalar krmse23=(el(kMSE,1,1)/k)^.5
100 scalar list krmse23
101 matrix drop kMSE
102 scalar drop k
103 loocv reg d.y      12d.y l3d.y
104 scalar define loormse23=r(rmse)

```

```

105
106 reg d.y l1d.y l2d.y l3d.y
107 estat ic
108 scalar define df123=el(r(S),1,4)
109 scalar define aic123=el(r(S),1,5)
110 scalar define bic123=el(r(S),1,6)
111 crossfold reg d.y l1d.y l2d.y l3d.y
112 scalar define k=10
113 matrix kMSE=r(est)/*r(est)
114 scalar krmse123=(el(kMSE,1,1)/k)^.5
115 scalar list krmse123
116 matrix drop kMSE
117 scalar drop k
118 loocv reg d.y l1d.y l2d.y l3d.y
119 scalar define loormse123=r(rmse)
120
121 matrix drop _all
122 matrix fit1=(df1,aic1,bic1,krmse1,loormse1)
123 matrix fit2=(df2,aic2,bic2,krmse2,loormse2)
124 matrix fit3=(df3,aic3,bic3,krmse3,loormse3)
125 matrix fit12=(df12,aic12,bic12,krmse12,loormse12)
126 matrix fit13=(df13,aic13,bic13,krmse13,loormse13)
127 matrix fit23=(df23,aic23,bic23,krmse23,loormse23)
128 matrix fit123=(df123,aic123,bic123,krmse123,loormse123)
129 matrix FIT=fit1\fit2\fit3\fit12\fit13\fit23\fit123
130 matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 12" "Model 13" "Model
23" "Model 123"
131 matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE
132 matrix list FIT
133
134 *1b
135 clear
136 set obs 300
137 gen t=[_n]
138 tsset t
139 gen r=rnormal()
140 gen y=r if t<4
141 replace y=0.5+0.5*l.y-0.1*l2.y+0.25*l3.y+r if t>=4
142 drop r
143
144 *model
145 reg d.y l1d.y
146 *aic/bic
147 estat ic
148 scalar define df1=el(r(S),1,4)
149 scalar define aic1=el(r(S),1,5)
150 scalar define bic1=el(r(S),1,6)
151 *10 fold cv rmse

```

```

152 crossfold reg d.y l1d.y
153 scalar define k=10
154 matrix kMSE=r(est)/*r(est)
155 scalar krmse1=(el(kMSE,1,1)/k)^.5
156 scalar list krmse1
157 matrix drop kMSE
158 scalar drop k
159 *loocv
160 loocv reg d.y l1d.y
161 scalar define loormse1=r(rmse)
162
163 reg d.y l2d.y
164 estat ic
165 scalar define df2=el(r(S),1,4)
166 scalar define aic2=el(r(S),1,5)
167 scalar define bic2=el(r(S),1,6)
168 crossfold reg d.y l2d.y
169 scalar define k=10
170 matrix kMSE=r(est)/*r(est)
171 scalar krmse2=(el(kMSE,1,1)/k)^.5
172 scalar list krmse2
173 matrix drop kMSE
174 scalar drop k
175 loocv reg d.y l2d.y
176 scalar define loormse2=r(rmse)
177
178 reg d.y l3d.y
179 estat ic
180 scalar define df3=el(r(S),1,4)
181 scalar define aic3=el(r(S),1,5)
182 scalar define bic3=el(r(S),1,6)
183 crossfold reg d.y l3d.y
184 scalar define k=10
185 matrix kMSE=r(est)/*r(est)
186 scalar krmse3=(el(kMSE,1,1)/k)^.5
187 scalar list krmse3
188 matrix drop kMSE
189 scalar drop k
190 loocv reg d.y l3d.y
191 scalar define loormse3=r(rmse)
192
193 reg d.y l1d.y l2d.y
194 estat ic
195 scalar define df12=el(r(S),1,4)
196 scalar define aic12=el(r(S),1,5)
197 scalar define bic12=el(r(S),1,6)
198 crossfold reg d.y l1d.y l2d.y
199 scalar define k=10

```

```

200 matrix kMSE=r(est)' $\times$ r(est)
201 scalar krmse12=(el(kMSE,1,1)/k) $^{.5}$ 
202 scalar list krmse12
203 matrix drop kMSE
204 scalar drop k
205 loocv reg d.y l1d.y l2d.y
206 scalar define loormse12=r(rmse)
207
208 reg d.y l1d.y l3d.y
209 estat ic
210 scalar define df13=el(r(S),1,4)
211 scalar define aic13=el(r(S),1,5)
212 scalar define bic13=el(r(S),1,6)
213 crossfold reg d.y l1d.y l3d.y
214 scalar define k=10
215 matrix kMSE=r(est)' $\times$ r(est)
216 scalar krmse13=(el(kMSE,1,1)/k) $^{.5}$ 
217 scalar list krmse13
218 matrix drop kMSE
219 scalar drop k
220 loocv reg d.y l1d.y l3d.y
221 scalar define loormse13=r(rmse)
222
223 reg d.y      12d.y l3d.y
224 estat ic
225 scalar define df23=el(r(S),1,4)
226 scalar define aic23=el(r(S),1,5)
227 scalar define bic23=el(r(S),1,6)
228 crossfold reg d.y      12d.y l3d.y
229 scalar define k=10
230 matrix kMSE=r(est)' $\times$ r(est)
231 scalar krmse23=(el(kMSE,1,1)/k) $^{.5}$ 
232 scalar list krmse23
233 matrix drop kMSE
234 scalar drop k
235 loocv reg d.y      12d.y l3d.y
236 scalar define loormse23=r(rmse)
237
238 reg d.y l1d.y l2d.y l3d.y
239 estat ic
240 scalar define df123=el(r(S),1,4)
241 scalar define aic123=el(r(S),1,5)
242 scalar define bic123=el(r(S),1,6)
243 crossfold reg d.y l1d.y l2d.y l3d.y
244 scalar define k=10
245 matrix kMSE=r(est)' $\times$ r(est)
246 scalar krmse123=(el(kMSE,1,1)/k) $^{.5}$ 
247 scalar list krmse123

```

```

248 matrix drop kMSE
249 scalar drop k
250 loocv reg d.y 11d.y 12d.y 13d.y
251 scalar define loormse123=r(rmse)
252
253 matrix drop _all
254 matrix fit1=(df1,aic1,bic1,krmse1,loormse1)
255 matrix fit2=(df2,aic2,bic2,krmse2,loormse2)
256 matrix fit3=(df3,aic3,bic3,krmse3,loormse3)
257 matrix fit12=(df12,aic12,bic12,krmse12,loormse12)
258 matrix fit13=(df13,aic13,bic13,krmse13,loormse13)
259 matrix fit23=(df23,aic23,bic23,krmse23,loormse23)
260 matrix fit123=(df123,aic123,bic123,krmse123,loormse123)
261 matrix FIT=fit1\fit2\fit3\fit12\fit13\fit23\fit123
262 matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 12" "Model 13" "Model
23" "Model 123"
263 matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE
264 matrix list FIT
265
266 *2
267 clear
268 cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets/Problem Set 3"
269 import delimited "Assignment_1_Monthly.txt"
270
271 rename lnu02300000 us_epr
272 rename flnan fl_nonfarm
273 rename fllfn fl_lf
274 rename flbppriv fl_bp
275 rename date datestring
276
277 *2d Generate a monthly date variable (make its display format monthly time, %tm)
278 gen datec=date(datestring, "YMD")
279 gen date=mofd(datec)
280 format date %tm
281
282 *2e tsset your data
283 tsset date
284 gen month=month(datec)
285
286 *2f
287 gen lnusepr=log(us_epr)
288 gen lnflnonfarm=log(fl_nonfarm)
289 gen lnfllf=log(fl_lf)
290 gen lnflbp=log(fl_bp)
291
292 *model

```

```

293 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf l(0/12)d.lnusepr
294 l(0/12)d.lnflbp i.month date
295 *aic/bic
296 estat ic
297 scalar define df1=el(r(S),1,4)
298 scalar define aic1=el(r(S),1,5)
299 scalar define bic1=el(r(S),1,6)
300 *10 fold cv rmse
301 crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf
302 l(0/12)d.lnusepr l(0/12)d.lnflbp i.month date
303 scalar define k=10
304 matrix kMSE=r(est)/*r(est)
305 scalar krmse1=(el(kMSE,1,1)/k)^.5
306 scalar list krmse1
307 matrix drop kMSE
308 scalar drop k
309 *loocv
310 loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnfllf l(0/12)d.lnusepr
311 l(0/12)d.lnflbp i.month date
312 scalar define loormse1=r(rmse)
313
314 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr
315 l(0/2)d.lnflbp i.month date
316 estat ic
317 scalar define df2=el(r(S),1,4)
318 scalar define aic2=el(r(S),1,5)
319 scalar define bic2=el(r(S),1,6)
320 crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr
321 l(0/2)d.lnflbp i.month date
322 scalar define k=10
323 matrix kMSE=r(est)/*r(est)
324 scalar krmse2=(el(kMSE,1,1)/k)^.5
325 scalar list krmse2
326 matrix drop kMSE
327 scalar drop k
328 *loocv
329 loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2)d.lnfllf l(0/2)d.lnusepr
330 l(0/2)d.lnflbp i.month date
331 scalar define loormse2=r(rmse)
332
333 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf l(0/2,12)d.lnflbp
334 i.month date
335 estat ic
336 scalar define df3=el(r(S),1,4)
337 scalar define aic3=el(r(S),1,5)
338 scalar define bic3=el(r(S),1,6)
339 crossfold reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf
340 l(0/2,12)d.lnflbp i.month date
341 scalar define k=10

```

```

333 matrix kMSE=r(est)' $\times$ r(est)
334 scalar krmse3=(el(kMSE,1,1)/k) $\cdot$ 5
335 scalar list krmse3
336 matrix drop kMSE
337 scalar drop k
338 loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/2,12)d.lnfllf l(0/2,12)d.lnflbp
i.month date
339 scalar define loormse3=r(rmse)
340
341 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
342 estat ic
343 scalar define df4=el(r(S),1,4)
344 scalar define aic4=el(r(S),1,5)
345 scalar define bic4=el(r(S),1,6)
346 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
347 scalar define k=10
348 matrix kMSE=r(est)' $\times$ r(est)
349 scalar krmse4=(el(kMSE,1,1)/k) $\cdot$ 5
350 scalar list krmse4
351 matrix drop kMSE
352 scalar drop k
353 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
354 scalar define loormse4=r(rmse)
355
356 matrix drop _all
357 matrix fit1=(df1,aic1,bic1,krmse1,loormse1)
358 matrix fit2=(df2,aic2,bic2,krmse2,loormse2)
359 matrix fit3=(df3,aic3,bic3,krmse3,loormse3)
360 matrix fit4=(df4,aic4,bic4,krmse4,loormse4)
361 matrix FIT=fit1\fit2\fit3\fit4
362 matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 4"
363 matrix colnames FIT=df AIC BIC K(10)RMSE LOORMSE
364 matrix list FIT
365
366 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

.*la
.*clear

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

```

```

* gen t=1/30
* tset 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*t.y-0.1*t2.y+0.25*t3.y+r if t>=4
(27 real changes made)

* drop r

* *model
reg d.y lid.y

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

D.y |      Coef.  Std. Err.      t  P>|t| [95% Conf. Interval]
y | -.4480631  .1659346  -2.70  0.012  -.7891465  -.1069798
_uons | .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 lid.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 lid.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
Model	.141091522	1	.141091522	F(1, 26)	=	0.14
Residual	25.8136069	25	1.03254427	Prob > F	=	0.7148
				R-squared	=	0.0054
Total	25.9546984	26	.99825763	Adj R-squared	=	-0.0343
				Root MSE	=	1.0161

D.y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
y	-.06877	.1860386	-0.37	0.715	-.4519236 .3143836
L2D.					
_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
```

```
. looey 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
Model	.100885437	1	.100885437	F(1, 24)	=	0.09
Residual	25.778986	24	1.07412442	Prob > F	=	0.7619
				R-squared	=	0.0039
Total	25.8798714	25	1.03519486	Adj R-squared	=	-0.0376
				Root MSE	=	1.0364

D.y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
y	.0592101	.1932007	0.31	0.762	-.3395366 .4579567
L3D.					
_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 13d.y



|      | RMSSE    |
|------|----------|
| 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
. looey reg d.y 13d.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
Model | 7.35816653 2 3.67908326 F(2, 24) = 4.75
Residual | 18.5965319 24 .774855494 Prob > F = 0.0183
Total | 25.9546984 26 .99825763 R-squared = 0.2835
                                         Adj R-squared = 0.2238
                                         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.u 11d.u 12d.u

```

```

. svy: regress d y lid.y l2d.y
      RMSE
est1  .2755585
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
. looey reg d.y lid.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 lid.y l3d.y

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

D.y |   Coef.   Std. Err.      t   P>|t| [95% Conf. Interval]
y | -.4285231  .1883531  -2.28  0.033  -.8181613  -.838885
LD. | .0320856  .1787074   0.18  0.859  -.3375988  .40177
L3D. | .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 lid.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      |
|----------|------------|----|------------|---------------|---|---------|
| Model    | .237741921 | 2  | .118870961 | F(2, 23)      | = | .0,11   |
| Residual | 25.6421295 | 23 | 1.1148752  | Prob > F      | = | .0.8993 |
| Total    | 25.8798714 | 25 | 1.03519486 | R-squared     | = | .0.0892 |



| D.y   | Coef.     | Std. Err. | t     | P> t   | [95% Conf. Interval] |
|-------|-----------|-----------|-------|--------|----------------------|
| y     | -.0805558 | .2299203  | -0.35 | .0.729 | -,5561822 .3950705   |
| L2D.  | .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 lid.y l2d.y l3d.y

Source |       SS          df        MS      Number of obs =      26
Model  | 8.04353652          3  2.68117851      F(3, 22)    =  3.31
Residual | 17.8363359         22  .81074254      Prob > F    = 0.0390
          |                                         R-squared = 0.3108
          |                                         Adj R-squared = 0.2168
Total   | 25.8798714         25  1.03519486      Root MSE   = .90041

```

D.y	Coeff.	Std. Err.	t	P> t	[95% Conf. Interval]
^x 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 lid.y l2d.y l3d.y

```

	RMSE
est1	1.277871
est2	.4126419
est3	1.187153
est4	.7160894
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
. looey reg d.y lid.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  74.288365  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 lid.y

Source |      SS        df       MS   Number of obs =     298
       | 40,2287435          1  40,2287435  F(1, 296)    =  30,29
       | 393.11806        296  1.32810155  Prob > F     =  0.0000
       |                                         R-squared    =  0.0928
       |                                         Adj R-squared =  0.0898
       |                                         Root MSE     =  1.1524

      D.y |      Coef.   Std. Err.      t    P>|t| [95% Conf. Interval]
       y |  -.3047742  .0553765  -5.50  0.000  -.4137557  -.1957926
       LD. |  .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 lid.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
        | 37.8014875     1  37.8014875   F(1, 295)    =  28.21
        | 395.272758    295  1.33990765   Prob > F    =  0.0000
        |                           R-squared =  0.0873
        |                           Adj R-squared =  0.0842
        |                           Root MSE   =  1.1575

D.y |      Coef.    Std. Err.      t   P>|t| [95% Conf. Interval]
y |  -.2956623  .0556646    -5.31  0.000  -.4052123  -.1861123
_l2d.y |  .0080967  .0671695     0.12  0.904  -.1240954  .1402888
_cons |
```

```
. 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	1.208812
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
Model  | 19.9598466     1 19.9598466   F(1, 294) =    14.20
Residual | 413.112855    294 1.40514577   Prob > F =  0.0002
          |                    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 | .214962   .0870353    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 | 1.156672

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
Model  | 112.32631     2 56.163155   F(2, 294) =    51.48
Residual | 320.747936    294 1.09097937   Prob > F =  0.0000
          |                    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]
```

	<i>y</i>	LD.	.4357556	.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 lid.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 lid.y l2d.y
```

Leave-One-Out Cross-Validation Results

Method	Value
Root Mean Squared Errors	1.0505398
Mean Absolute Errors	.8498225
Pseudo-R2	.2471428

. scalar define loormse12=r(rmse)

. reg d.y lid.y l3d.y

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

D.y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
<i>y</i>					
LD.	-.2642997	.0577048	-4.58	0.000	-.3778681 -.1507313
L3D.	.1369759	.0577577	2.37	0.018	.0233034 .2566484
_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
Model	45.215903	2	22.6079515	F(2, 293)	=	17.08
Residual	387.856799	293	1.32374334	Prob > F	=	0.0000
Total	433.072702	295	1.46804306	R-squared	=	0.1044
				Adj R-squared	=	0.0983
				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
1	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
. looov reg d.y      12d.y 13d.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 lid.y 12d.y 13d.y
```

Source	SS	df	MS	Number of obs	=	296
Model	113.494499	3	37.8314998	F(3, 292)	=	34.57
Residual	319.578203	292	1.0944459	Prob > F	=	0.0000
Total	433.072702	295	1.46804306	R-squared	=	0.2621
				Adj R-squared	=	0.2545
				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 lid.y 12d.y 13d.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
. looov reg d.y lid.y 12d.y 13d.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,krmsei,loormsei)

. matrix fit2=(df2,aic2,bic2,krmsei2,loormsei2)

. matrix fit3=(df3,aic3,bic3,krmsei3,loormsei3)

. matrix fit12=(df12,aic12,bic12,krmsei12,loormsei12)

. matrix fit13=(df13,aic13,bic13,krmsei13,loormsei13)

. matrix fit23=(df23,aic23,bic23,krmsei23,loormsei23)

. matrix fit123=(df123,aic123,bic123,krmsei123,loormsei123)

. 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  926.01363  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 lnu0230000 us_epr

. rename flnan fl_nonfarm

. rename fllfn fl_lf

. rename flbpriv 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 lnfibp=log(fl_bp)

```

(588 missing values generated)

```
* *model
reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(0/12)d.lnflif l(0/12)d.lnusepr l(0/12)d.inflbp i.month date
```

Source	SS	df	MS	Number of obs	=	383
Model	.050768432	63	.0000805848	F(63, 319)	=	53.24
Residual	.004828206	319	.000015135	Prob > F	=	0.0000
Total	.055596638	382	.000145541	R-squared	=	0.9132
				Adj R-squared	=	0.8960
				Root MSE	=	.00389
D_						
lnflnonfarm				Coeff.	Std. Err.	t
						P> t
						[95% Conf. Interval]
lnflnonfarm						
L1D,	-.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	.1688302
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,	-.1288027	.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
lnflif						
D1,	-.4678581	.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	.0998936	0.35	0.723	-.1597391	.2299535
L6D,	.0231131	.0990097	0.23	0.816	-.1716814	.2179875
L7D,	.1259544	.0974149	1.29	0.197	-.0657025	.3176112
L8D,	.024699	.0982434	0.25	0.882	-.1685878	.2179858
L9D,	.1150141	.1100678	1.04	0.297	-.1015364	.3315646
L10D,	.2080239	.1108839	1.88	0.062	-.0101322	.42618
L11D,	-.0752819	.1117782	-0.67	0.581	-.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	.0568895	.3977713
L3D,	-.1902081	.087627	-2.17	0.031	-.3626678	-.0178883
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	-.1589888	.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	-.0723888	.5022593
L12D,	.0161647	.1416189	0.11	0.909	-.2624602	.2947897
inflbp						
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	.002299	.0115047
L7D,	.007351	.0022427	3.27	0.001	.0029301	.0117719
L8D,	.0054368	.0023044	2.36	0.019	.0009931	.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	-.0018063	.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	.0036657	-4.02	0.000	-.0215823	-.0073942
7	-.0097851	.0032285	-3.03	0.003	-.016137	-.0034332
8	-.0006542	.0035161	-0.02	0.988	-.0069719	.0068635
9	.0051214	.0032399	1.58	0.115	-.0012529	.0114956
10	.0025368	.0044185	0.57	0.566	-.0061572	.0112288
11	-.00084647	.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 dfi=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 1(1/12)d.lnflnonfarm 1(0/12)d.lnflif 1(0/12)d.lnusepr 1(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 krmsei=(el(kMSE,1,1)/k)^.5
scalar list krmsei
krmsei = .00474646

matrix drop KMSE
scalar drop k

*loocv
loocv reg d.lnflnonfarm 1(1/12)d.lnflnonfarm 1(0/12)d.lnflif 1(0/12)d.lnusepr 1(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 loormsei=r(rmse)

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



| Source   | SS         | df  | MS         | Number of obs | = | 393    |
|----------|------------|-----|------------|---------------|---|--------|
| Model    | .050628703 | 33  | .001534203 | F(33, 359)    | = | 89.36  |
| Residual | .006163276 | 359 | .000017168 | Prob > F      | = | 0.0000 |
| Total    | .056791979 | 392 | .000144877 | R-squared     | = | 0.8915 |
|          |            |     |            | Adj R-squared | = | 0.8815 |
|          |            |     |            | 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 .0290935 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. .0569847 .0295656 1.92 0.055 -.001239 .1150483
L8D. .0333213 .0299318 1.11 0.266 -.0255424 .092185
L9D. .0904818 .0496354 1.82 0.059 -.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

lnflif
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	-.009851
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	-.0069822	.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
. matrix 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
Model	.045212283	32	.001412884	F(32, 350)	=	47.62
Residual	.010384356	350	.00002967	Prob > F	=	0.0000
Total	.055596638	382	.000145541	R-squared	=	0.8132
				Adj R-squared	=	0.7961
				Root MSE	=	.00545

D.

	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	0.595	-.0970668 .0479131
L7D,	.0020704	.0365831	0.06	0.955	-.0698861 .0740208
L8D,	-.0138142	.0369521	-0.37	0.709	-.0864904 .058862
L9D,	.0905825	.0658306	1.38	0.170	-.0388998 .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
inflif					
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	.0023819	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 -.002026
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.lnflif 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.lnflif 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.lnflif l(1/2,12,24)d.lnusepr i.month
```

Source	SS	df	MS	Number of obs	=	515
Model	.043561062	32	.001361283	F(32, 482)	=	27.54
Residual	.023823343	482	.000049426	Prob > F	=	0.0000
Total	.067384405	514	.000131098	R-squared	=	0.6665
				Adj R-squared	=	0.6230
				Root MSE	=	.00703

D. lnflnonfarm	Coef.	Std. Err.	T	P> T	[95% Conf. Interval]
lnflnonfarm					
LD.	-.0532095	.002857	-0.88	0.378	-.1716647 .0652456
L2D.	-.0454893	.0010899	-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
lnflif					
LD.	.0987675	.1177628	0.84	0.402	-.1326242 .3301593
L2D.	-.0771596	.1180294	-0.65	0.514	-.3990752 .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	0.518	-.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	-.0010544 .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.lnflif 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

. looov reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnflif 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.

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

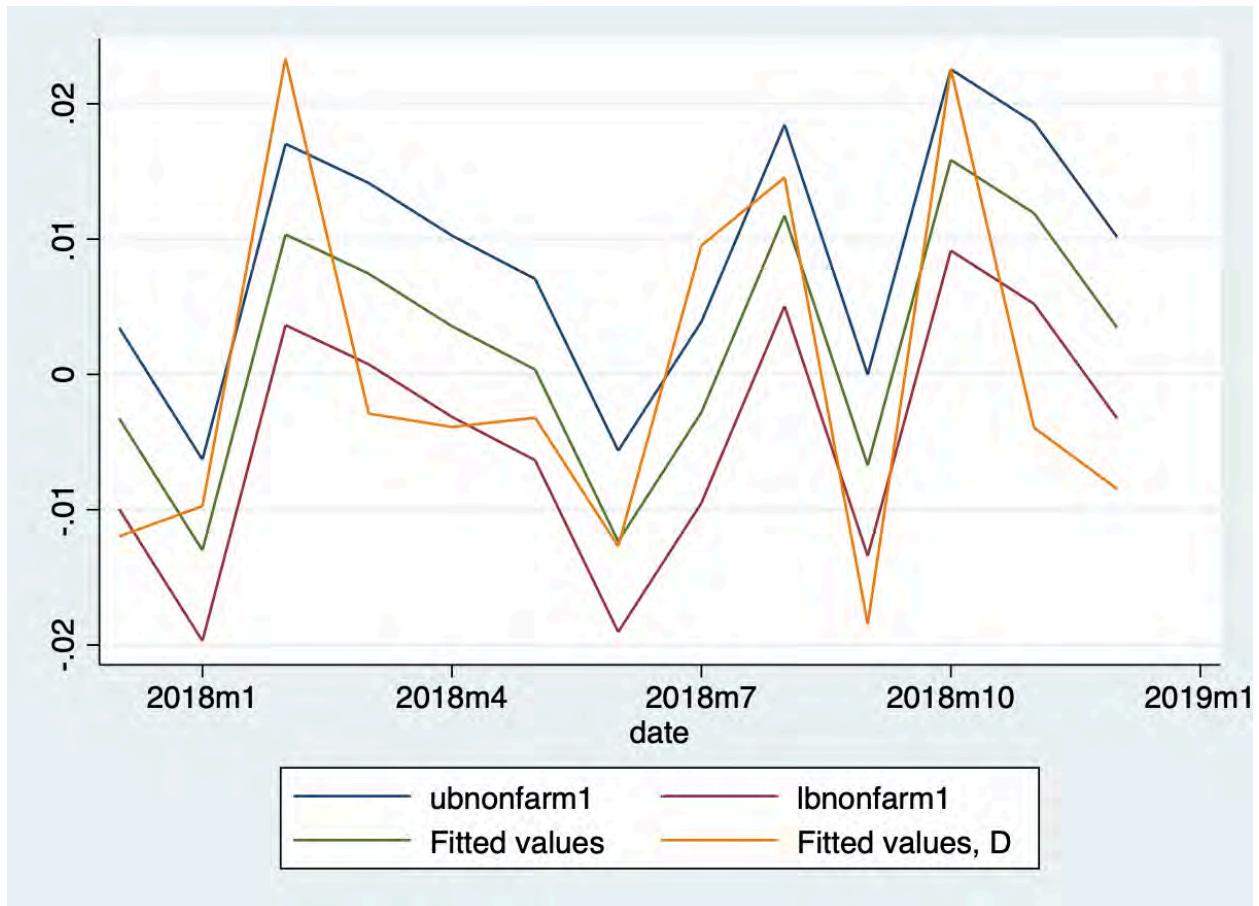
2. Refer to homework 3, question 2. Adapt the four models used there so they will be appropriate for making a one period ahead forecast.

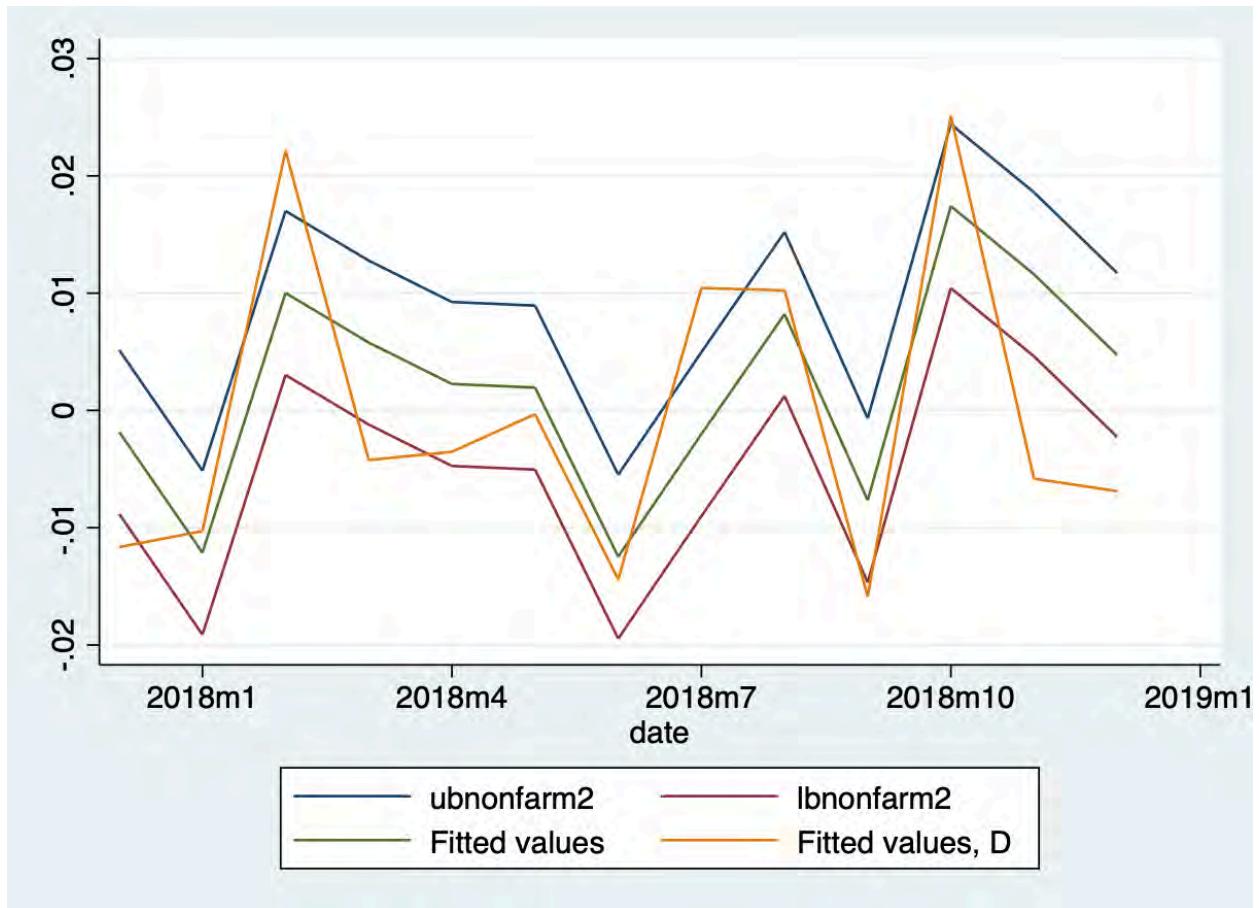
```
1 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr  
l(1/12)d.lnflbp i.month date  
2 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr  
l(1/2)d.lnflbp i.month date  
3 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp i.month  
date  
4 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
```

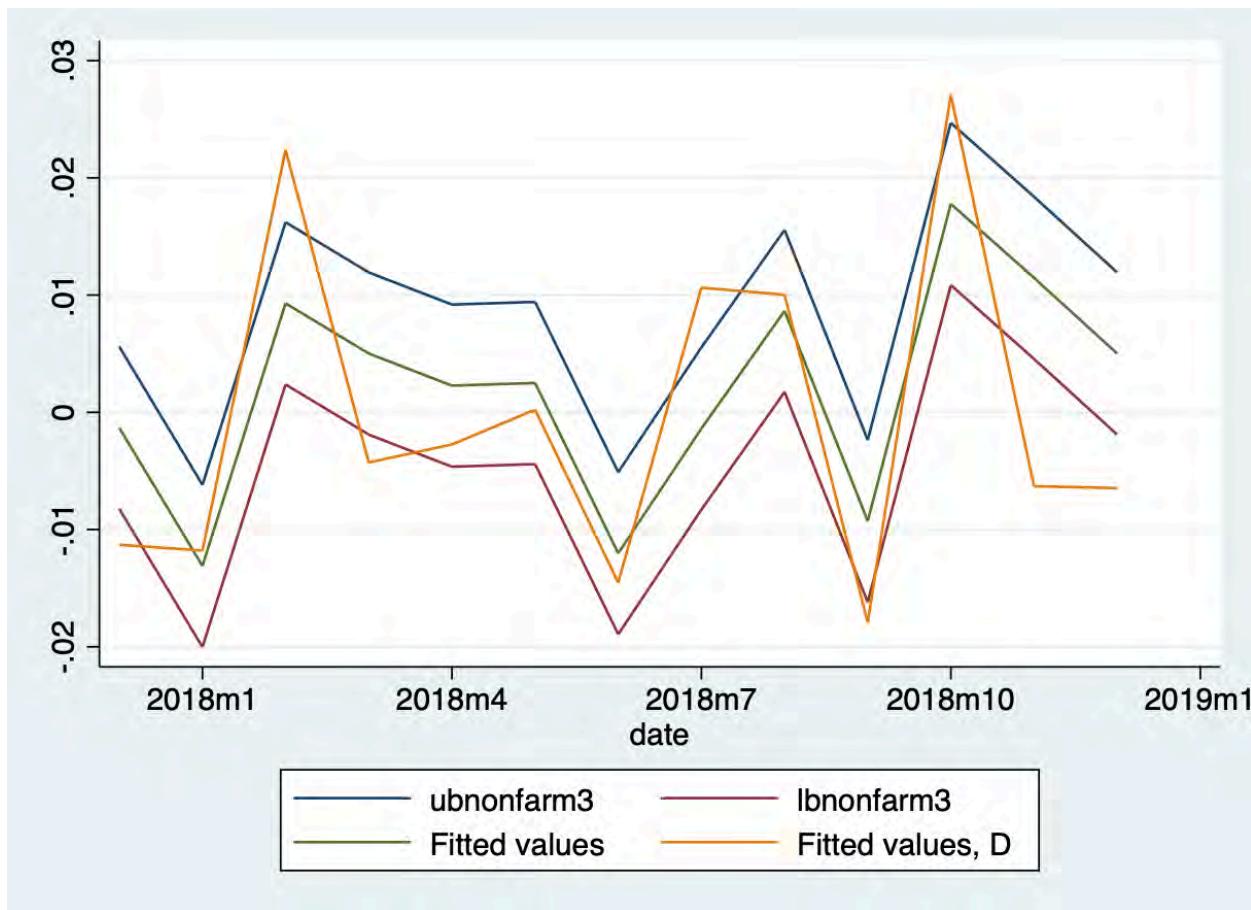
3. For each model, calculate the out of sample RMSE for the last year of observations (last 12 observations). To do this, you must not include these observations in the model estimation.

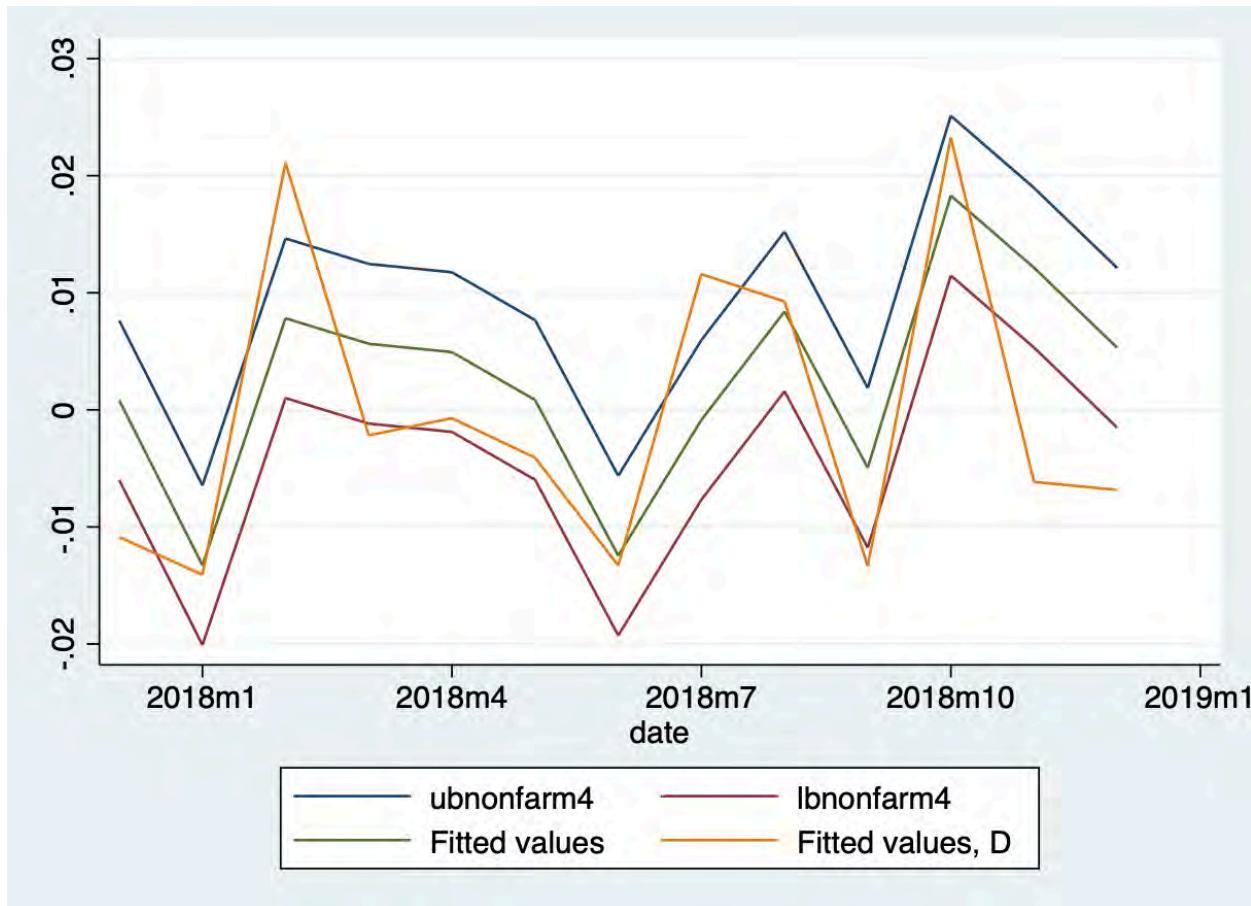
	c1	c2	c3	c4
r1	.00341856	.00356652	.00352901	.00347889

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.







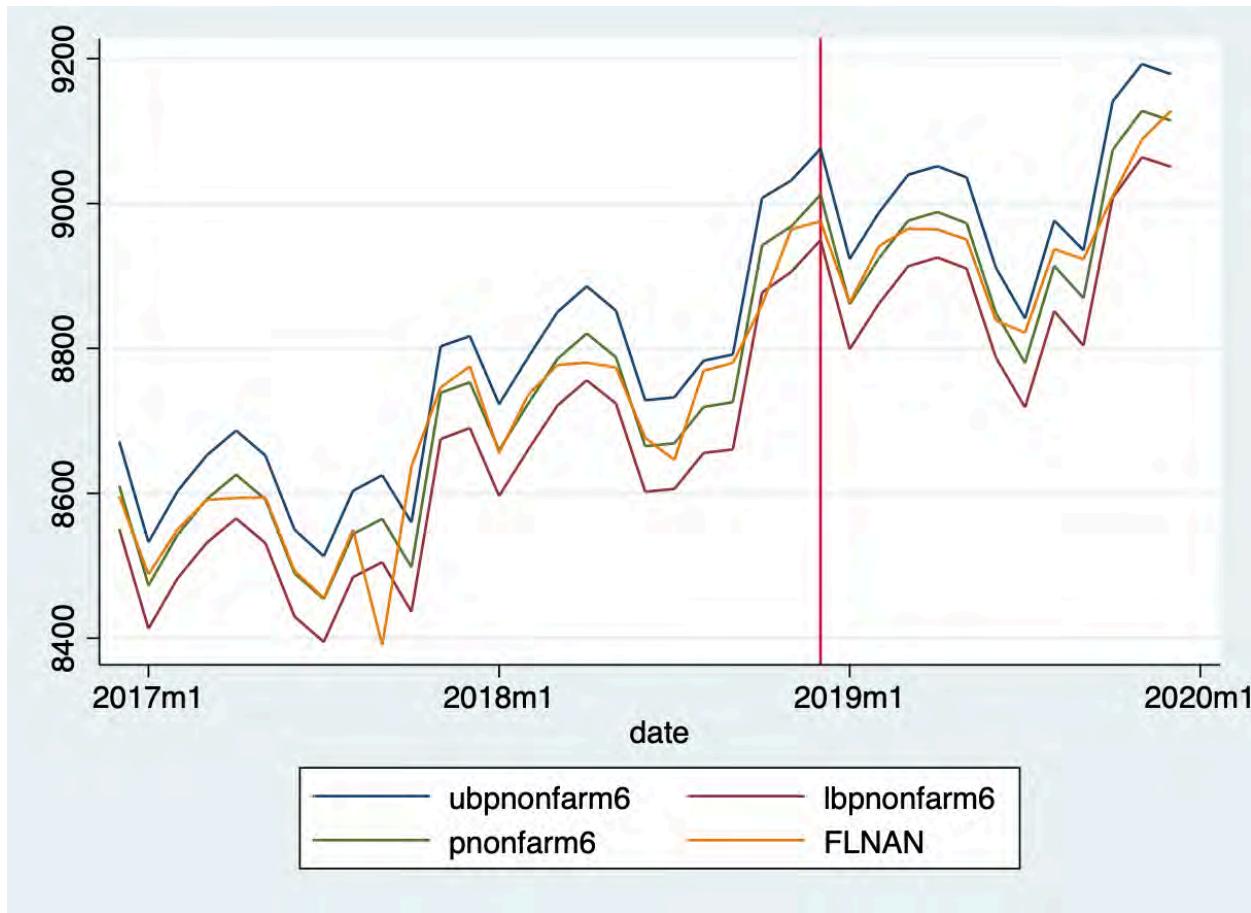


5. Select the best model for forecasting purposes based on AIC, BIC, LOOCV, and out of sample RMSE for the final year of data. Justify your choice.

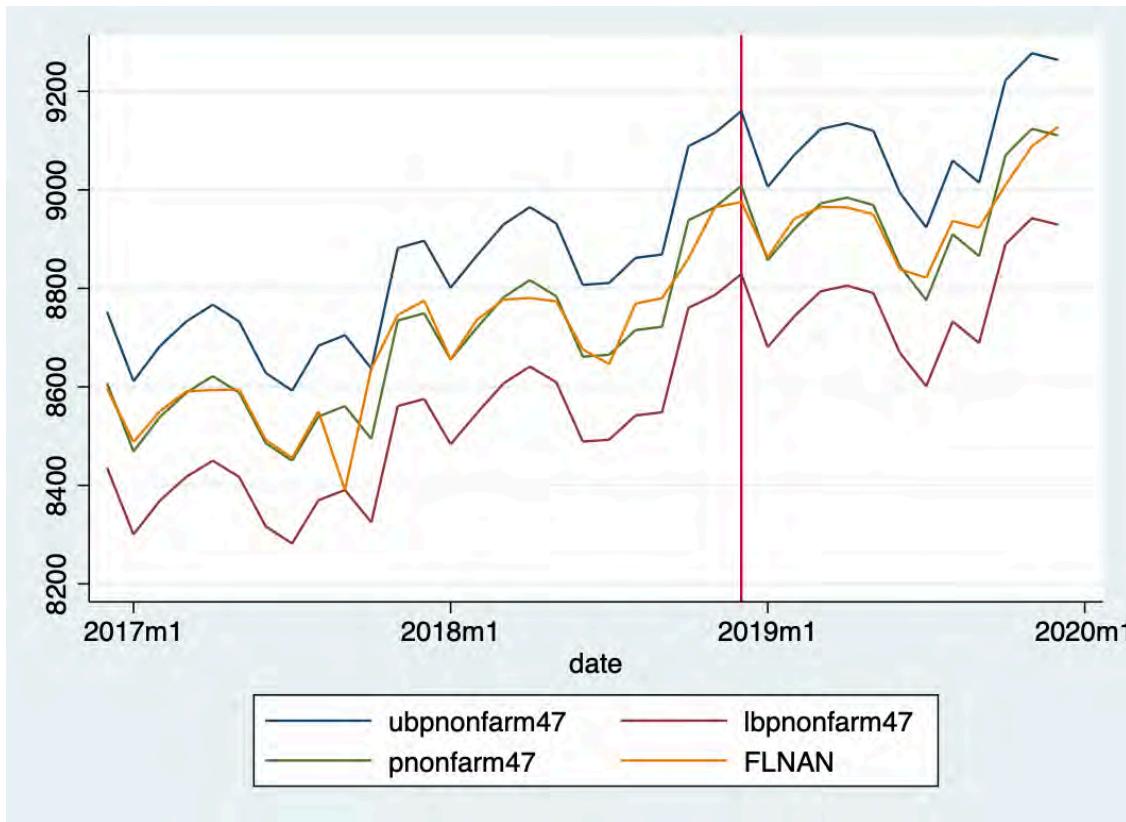
	df	AIC	BIC	RMSE	LOORMSE
Model 1	61	-3114.0527	-2875.1644	.00341856	.00380836
Model 2	31	-3190.9568	-3068.7301	.00356652	.00371849
Model 3	31	-3115.1451	-2993.7429	.00352901	.00375319
Model 4	33	-4236.6004	-4097.3209	.00347889	.00355785

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.



- 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.

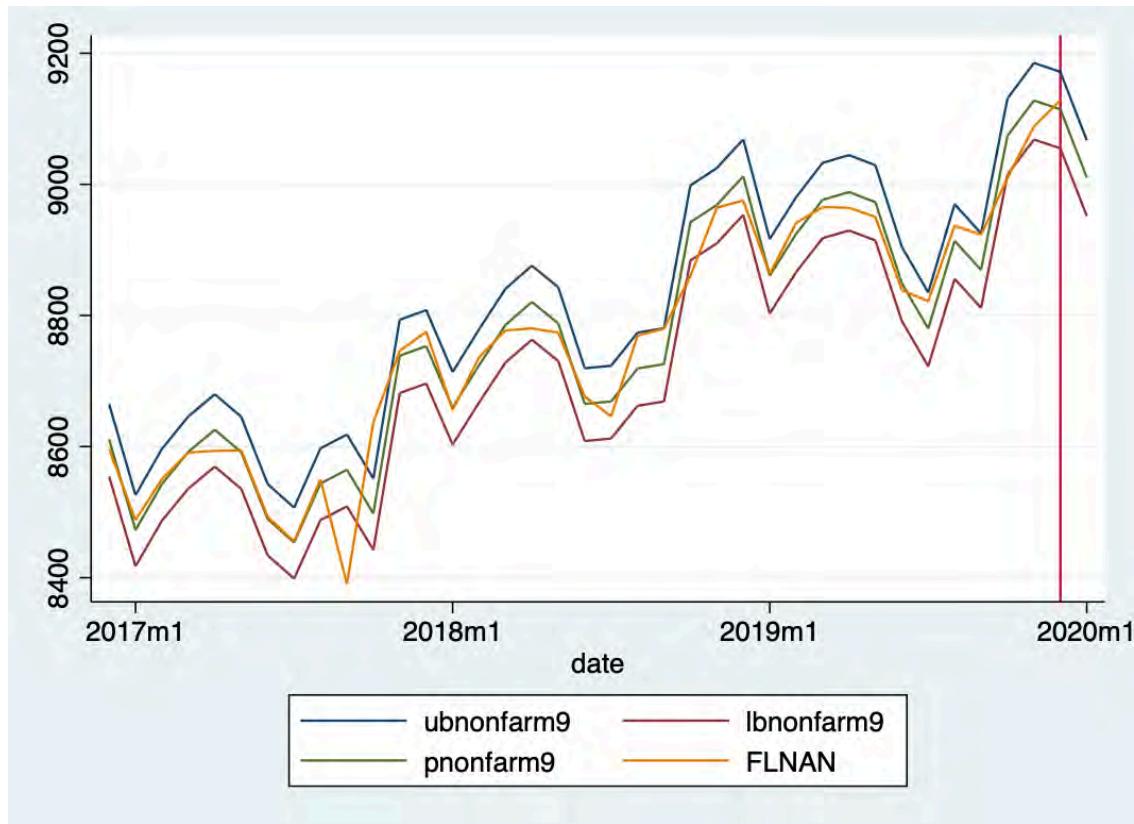


8. Run these commands to add January 2020 to the data (for which you will generate a forecast) and fill in the corresponding values for year and month:

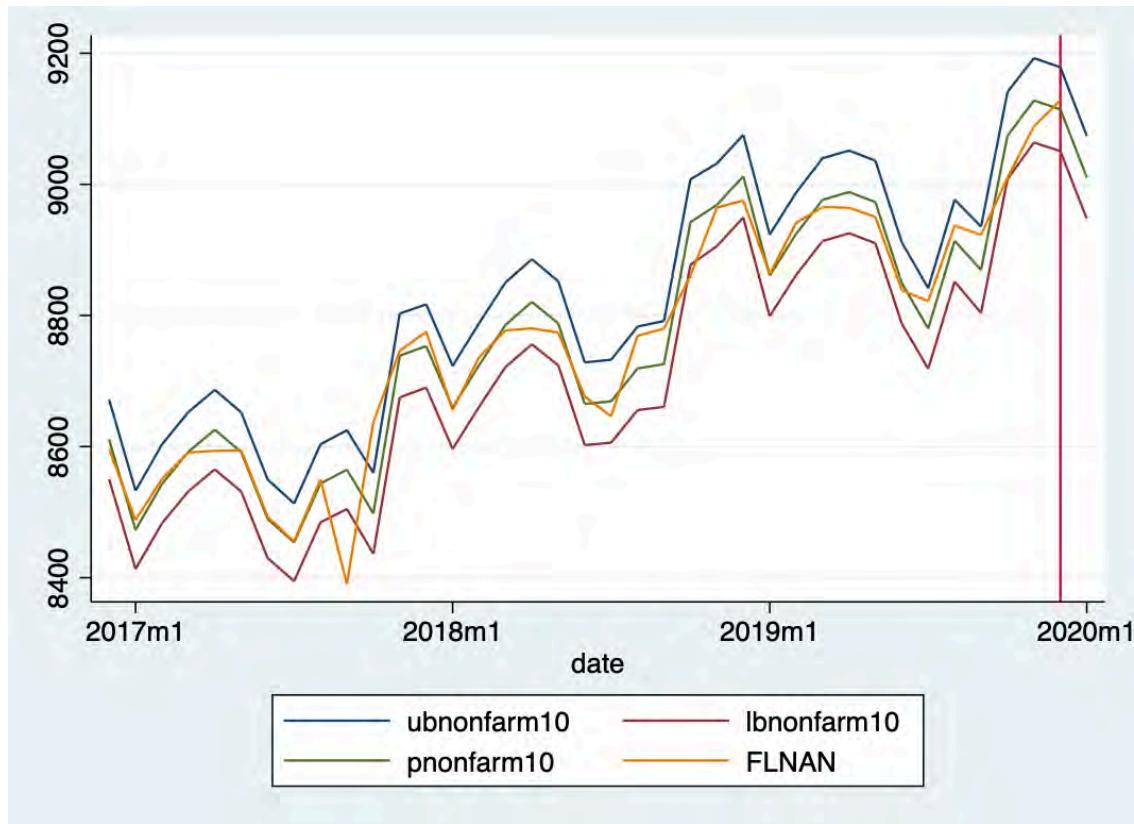
```
tsappend, add(1)
replace month=month(dofm(date)) if month==.
```

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

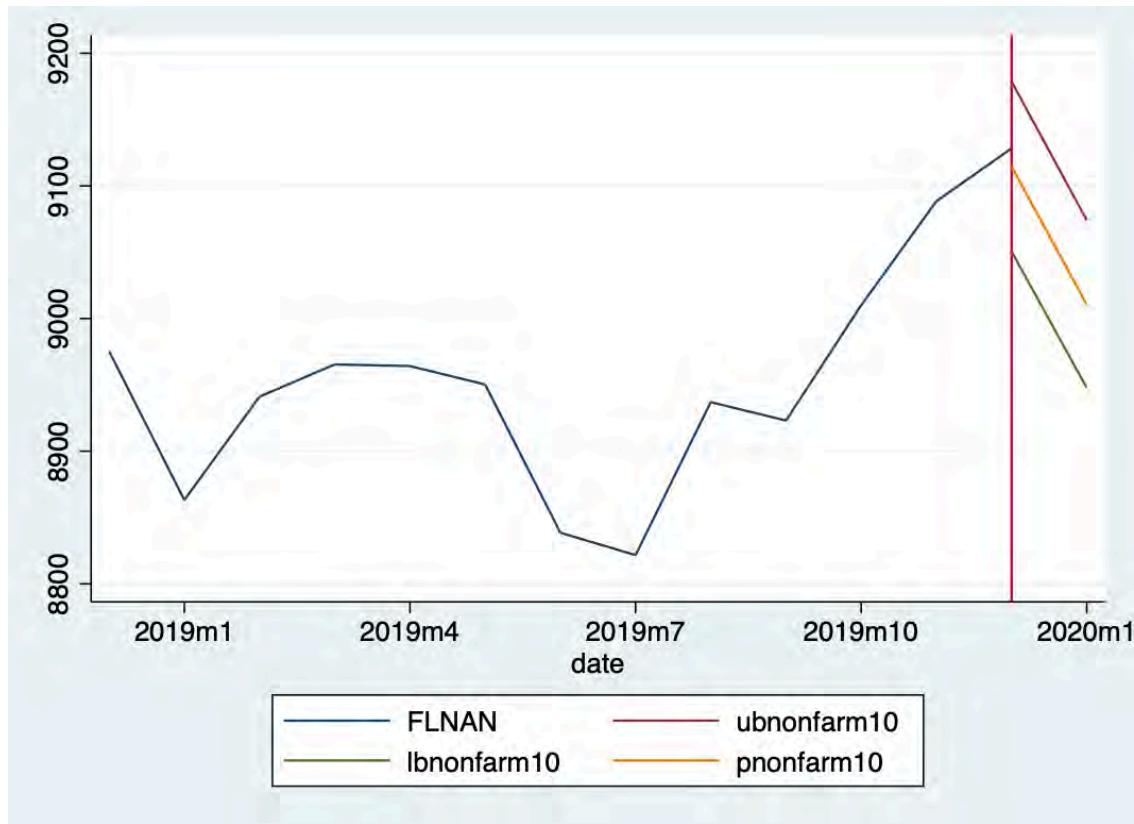
9. Run your selected model on the full sample and use it to forecast January 2020. Create point and interval forecasts for the change in the log of non-farm employment. Use the empirical approach.



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

```

1 clear
2 set more off
3
4 cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets/Problem Set 4"
5
6 *2a
7 *Done
8
9 *2b Load the data
10 import delimited "Assignment_1_Monthly.txt"
11
12 rename lnu02300000 us_epr
13 rename flnan fl_nonfarm
14 rename fllfn fl_lf
15 rename flbpriv fl_bp
16 rename date datestring
17
18 *2c Turn on a log file
19 log using "Problem Set 4", replace

```

```

20
21 *2d Generate a monthly date variable (make its display format monthly time, %tm)
22 gen datec=date(datestring, "YMD")
23 gen date=mofd(datec)
24 gen month=month(datec)
25 format date %tm
26
27 *2e tsset your data
28 tsset date
29
30 *2f
31 gen lnusepr=log(us_epr)
32 gen lnflnonfarm=log(f1_nonfarm)
33 gen lnflif=log(f1_if)
34 gen lnflbp=log(f1_bp)
35
36 *1
37 drop if tin(2020m1,)
38
39 *2
40 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnflif l(1/12)d.lnusepr
   l(1/12)d.lnflbp i.month date
41 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnflif l(1/2)d.lnusepr
   l(1/2)d.lnflbp i.month date
42 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnflif l(1/2,12)d.lnflbp
   i.month date
43 reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnflif
   l(1/2,12,24)d.lnusepr i.month
44
45 *3
46 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnflif l(1/12)d.lnusepr
   l(1/12)d.lnflbp i.month date if tin(,2018m12)
47 scalar define rmse1=e(rmse)
48 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnflif l(1/2)d.lnusepr
   l(1/2)d.lnflbp i.month date if tin(,2018m12)
49 scalar define rmse2=e(rmse)
50 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnflif l(1/2,12)d.lnflbp
   i.month date if tin(,2018m12)
51 scalar define rmse3=e(rmse)
52 reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnflif
   l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
53 scalar define rmse4=e(rmse)
54
55 matrix drop _all
56 matrix row=(rmse1, rmse2, rmse3, rmse4)
57 matrix RMSE = row
58 matrix list RMSE
59

```

```

60 *4
61 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr
62 l(1/12)d.lnflbp i.month date if tin(,2018m12)
63 predict nonfarm1
64 gen ubnonfarm1=nonfarm1+1.96*e(rmse)
65 gen lbnonfarm1=nonfarm1-1.96*e(rmse)
66 tsline ubnonfarm1 lbnonfarm1 nonfarm1 d.nonfarm1 if tin(2017m12, 2018m12)
67 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr
68 l(1/2)d.lnflbp i.month date if tin(,2018m12)
69 predict nonfarm2
70 gen ubnonfarm2=nonfarm2+1.96*e(rmse)
71 gen lbnonfarm2=nonfarm2-1.96*e(rmse)
72 tsline ubnonfarm2 lbnonfarm2 nonfarm2 d.nonfarm2 if tin(2017m12, 2018m12)
73 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp
74 i.month date if tin(,2018m12)
75 predict nonfarm3
76 gen ubnonfarm3=nonfarm3+1.96*e(rmse)
77 gen lbnonfarm3=nonfarm3-1.96*e(rmse)
78 tsline ubnonfarm3 lbnonfarm3 nonfarm3 d.nonfarm3 if tin(2017m12, 2018m12)
79 reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
80 l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
81 predict nonfarm4
82 gen ubnonfarm4=nonfarm4+1.96*e(rmse)
83 gen lbnonfarm4=nonfarm4-1.96*e(rmse)
84 tsline ubnonfarm4 lbnonfarm4 nonfarm4 d.nonfarm4 if tin(2017m12, 2018m12)

85 *5
86 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr
87 l(1/12)d.lnflbp i.month date
88 estat ic
89 scalar define df1=el(r(S),1,4)
90 scalar define aic1=el(r(S),1,5)
91 scalar define bic1=el(r(S),1,6)
92 loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr
93 l(1/12)d.lnflbp i.month date
94 scalar define loormse1=r(rmse)

95 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr
96 l(1/2)d.lnflbp i.month date
97 estat ic
98 scalar define df2=el(r(S),1,4)
99 scalar define aic2=el(r(S),1,5)
100 scalar define bic2=el(r(S),1,6)
101 loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2)d.lnfllf l(1/2)d.lnusepr
102 l(1/2)d.lnflbp i.month date
103 scalar define loormse2=r(rmse)

```

```

99 reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp
100 i.month date
101 estat ic
102 scalar define df3=el(r(S),1,4)
103 scalar define aic3=el(r(S),1,5)
104 scalar define bic3=el(r(S),1,6)
105 loocv reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfllf l(1/2,12)d.lnflbp
106 i.month date
107 scalar define loormse3=r(rmse)
108
109 reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
110 l(1/2,12,24)d.lnusepr i.month
111 estat ic
112 scalar define df4=el(r(S),1,4)
113 scalar define aic4=el(r(S),1,5)
114 scalar define bic4=el(r(S),1,6)
115 loocv reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
116 l(1/2,12,24)d.lnusepr i.month
117 scalar define loormse4=r(rmse)
118
119 matrix drop _all
120 matrix fit1=(df1,aic1,bic1,rmse1,loormse1)
121 matrix fit2=(df2,aic2,bic2,rmse2,loormse2)
122 matrix fit3=(df3,aic3,bic3,rmse3,loormse3)
123 matrix fit4=(df4,aic4,bic4,rmse4,loormse4)
124 matrix FIT=fit1\fit2\fit3\fit4
125 matrix rownames FIT="Model 1" "Model 2" "Model 3" "Model 4"
126 matrix colnames FIT=df AIC BIC RMSE LOORMSE
127 matrix list FIT
128
129 *6
130 reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
131 l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
132 predict nonfarm6
133 predict stdfore6, stdf
134 gen pnonfarm6=exp(l.lnflnonfarm+nonfarm6)*exp(.5*e(rmse)^2)
135 gen ubpnonfarm6=exp(l.lnflnonfarm+nonfarm6+1.96*stdfore6)*exp(.5*e(rmse)^2)
136 gen lbpnonfarm6=exp(l.lnflnonfarm+nonfarm6-1.96*stdfore6)*exp(.5*e(rmse)^2)
137 tsline ubpnonfarm6 lbpnonfarm6 pnonfarm6 fl_nonfarm if tin(2016m12,2019m12),
138 tline(2018m12)
139
140 *7
141 reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfllf
142 l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
143 predict nonfarm47
144 predict pres47 if tin(2016m12,2018m12), residual
145 gen expres47=exp(pres47) if tin(2016m12,2018m12)
146 summ expres47

```

```

140 gen pnonfarm47=r(mean)*exp(l.lnflnonfarm+nonfarm47)
141 _pctile expres47, percentile(2.5,97.5)
142 gen lbpnonfarm47=r(r1)*exp(l.lnflnonfarm+nonfarm47)
143 gen ubpnonfarm47=r(r2)*exp(l.lnflnonfarm+nonfarm47)
144 tsline ubpnonfarm47 lbpnonfarm47 pnonfarm47 fl_nonfarm if tin(2016m12,2019m12),
tline(2018m12)

145
146 *8
147 tsappend, add(1)
148 replace month=month(dofm(date)) if month==.

149
150 *9
151 reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfl1f
l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
152 predict nonfarm9
153 predict pres9 if tin(,2019m12), residual
154 gen expres9=exp(pres9) if tin(,2019m12)
155 summ expres9
156 gen pnonfarm9=r(mean)*exp(l.lnflnonfarm+nonfarm9)
157 _pctile expres9, percentile(2.5,97.5)
158 gen lbnonfarm9=r(r1)*exp(l.lnflnonfarm+nonfarm9)
159 gen ubnonfarm9=r(r2)*exp(l.lnflnonfarm+nonfarm9)
160 tsline ubnonfarm9 lbnonfarm9 pnonfarm9 fl_nonfarm if tin(2016m12,2020m1),
tline(2019m12)

161
162 *10
163 reg d.lnflnonfarm l(1/12,24)d.lnflnonfarm l(1/2,12,24)d.lnfl1f
l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
164 predict nonfarm10
165 predict stdfore10, stdf
166 gen pnonfarm10=exp(l.lnflnonfarm+nonfarm10)*exp(.5*e(rmse)^2)
167 gen ubnonfarm10=exp(l.lnflnonfarm+nonfarm10+1.96*stdfore10)*exp(.5*e(rmse)^2)
168 gen lbnonfarm10=exp(l.lnflnonfarm+nonfarm10-1.96*stdfore10)*exp(.5*e(rmse)^2)
169 tsline ubnonfarm10 lbnonfarm10 pnonfarm10 fl_nonfarm if tin(2016m12,2020m1),
tline(2019m12)

170
171 *11
172 tsline fl_nonfarm if tin(2018m12,2020m1) || tsline ubnonfarm10 lbnonfarm10
pnonfarm10 if tin(2019m12,), tline(2019m12)

173
174 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

```

End of session - 2020-03-23 21:24:45

```

. *to generate a monthly wage variable where all variables become monthly same as my
. gen date=date(datestring, "YMD")
. gen date=mofd(datec)
. gen month=month(datec)
. format date %tm

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

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

. gen lnflnonfarm=log(f1_nonfarm)
. gen lnflf=log(f1_f1)
(444 missing values generated)

. gen lnflbp=log(f1_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.lnflf l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date

```

Source	SS	df	MS	Number of obs	=	371
Model	.033628663	60	.000560478	F(60, 310)	=	49.11
Residual	.003538017	310	.000011413	Prob > F	=	0.0000
Total	.037166668	370	.00010045	R-squared	=	0.9048
				Adj R-squared	=	0.8864
				Root MSE	=	.00338

D.	Inflnonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
lnflnonfarm						
L0D.	-.1709382	.0544484	-3.14	0.002	-.2780734	-.0638803
L2D.	-.1482615	.0563272	-2.63	0.009	-.2598935	-.0374296
L3D.	.162612	.057373	2.83	0.005	.0497221	.2758018
L4D.	.1338157	.0578492	2.31	0.021	.0199889	.2476424
L5D.	.0485865	.0586353	0.83	0.468	-.066787	.1639599
L6D.	.1034826	.0583736	1.77	0.077	-.0113761	.2183412
L7D.	-.0017194	.0587605	-0.83	0.977	-.1173393	.1139805
L8D.	-.0661616	.0589687	-1.12	0.263	-.1821911	.0498668
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
lnflf						
L0D.	-.1446453	.0983821	-1.47	0.143	-.3382263	.0489358
L2D.	-.1262282	.0987995	-1.28	0.292	-.3286385	.0681742
L3D.	-.0938871	.0998186	-0.94	0.348	-.2982948	.1025286
L4D.	-.0237969	.1021375	-0.23	0.816	-.2247673	.1771735
L5D.	.089078	.1014128	0.89	0.929	-.1984666	.2086226
L6D.	-.1064011	.1085346	-1.06	0.291	-.3042177	.0914155
L7D.	-.0363673	.1081261	-0.36	0.717	-.2333881	.1606454
L8D.	-.0227181	.0999956	-0.23	0.820	-.219289	.1738688
L9D.	.1419096	.0991963	1.43	0.154	-.0532736	.3370929
L10D.	.2356432	.0995816	2.37	0.819	-.0397018	.4315847
L11D.	-.0156215	.1085373	-0.16	0.877	-.2134433	.1822083
L12D.	-.1496896	.0991977	-1.51	0.132	-.3448754	.0454963
lnusepr						
L0D.	.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.	.148566	.1327682	1.86	0.291	-.1286748	.4018068
L5D.	-.0429764	.1313909	-0.83	0.744	-.3015071	.2155544
L6D.	.3174743	.1387471	2.43	0.816	-.0602104	.5747382
L7D.	.1643272	.1319288	1.25	0.214	-.0952619	.4239164
L8D.	-.0363099	.1388986	-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						
L0D.	.0014899	.0015781	0.94	0.346	-.0016152	.004595
L2D.	.0048244	.0018575	2.68	0.010	.0011694	.0084793
L3D.	.0005472	.0019601	3.34	0.001	.0026984	.010404
L4D.	.0049279	.001984	2.48	0.014	.0010242	.0088316
L5D.	.0006421	.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.	.0036467	.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	.0015780	1.07	0.050	.0001862	

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	.0042083	.0035921	1.17	0.243	-.0028678	.0112683
8	.0133789	.0037305	3.59	0.000	.0000387	.0207191
9	.0108298	.0031587	3.43	0.001	.0004146	.0178045
10	.0165185	.0040273	4.10	0.000	.0005942	.0244428
11	.0098475	.0037037	2.44	0.015	.0017599	.0163351
12	.0154964	.0034704	4.47	0.000	.0006679	.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.lnf1bp i.month date

Source	SS	df	MS	Number of obs	=	381
Model	.033990108	30	.001133004	F(30, 350)	=	90.75
Residual	.004369669	350	.000012485	Prob > F	=	0.0000
Total	.038359977	380	.000100947	R-squared	=	0.8861
				Adj R-squared	=	0.8763
				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.	-.0958662	.0518133	-1.85	0.065	-.1977648 .0668443
L3D.	.2398422	.0515342	4.65	0.000	.1384866 .3411978
L4D.	.1709118	.0489573	3.49	0.001	.0746243 .2671992
L5D.	.1515931	.0495961	3.86	0.002	.0540493 .249137
L6D.	.1371627	.0507474	2.70	0.007	.0373545 .2369708
L7D.	.0720218	.0527616	1.37	0.173	-.0317478 .1757914
L8D.	.0100887	.0513147	0.20	0.844	-.0980842 .1110807
L9D.	.0796876	.0488054	1.63	0.193	-.0163013 .1756764
L10D.	-.18866804	.047519	-3.97	0.000	-.2821391 -.0952217
L11D.	-.0844174	.0497603	-1.70	0.091	-.1822842 .0134493
L12D.	.388731	.0497483	7.81	0.000	.2988879 .4865741
lnfllf					
LD.	-.19426	.0941693	-2.86	0.040	-.3794688 -.0099513
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
lnf1bp					
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	0.886	-.0041884 .0053746
7	.0012385	.0021082	0.59	0.557	-.0029078 .0053847
8	.011499	.0024647	4.67	0.000	.0006615 .0163465
9	.0150889	.002545	5.98	0.000	.0100036 .02080143
10	.0207091	.0025776	8.03	0.000	.0156395 .0257787
11	.0124833	.0024281	5.14	0.000	.0077077 .0172588
12	.0125089	.0019735	6.34	0.000	.0086245 .0163873
date					
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.lnf1bp i.month date

Source	SS	df	MS	Number of obs	=	371
Model	.033019839	30	.001100661	F(30, 340)	=	90.24
Residual	.004146841	340	.000012197	Prob > F	=	0.0000
Total	.03716668	370	.00010045	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	.0664315 .2547854
L6D.	.1606861	.0504142	3.19	0.002	.0615231 .2598491
L7D.	.0928285	.0512561	1.88	0.073	-.0087985 .1928394
L8D.	.0257082	.0504855	0.51	0.611	-.073595 .1250115
L9D.	.0816557	.0485929	1.68	0.094	-.013925 .1772363
L10D.	-.1982283	.0474658	-4.18	0.000	-.2935889 -.1048616
L11D.	-.1022931	.0491268	-2.08	0.038	-.199005 -.0055813
L12D.	.3672239	.0495416	7.41	0.000	.2697773 .4646706
lnfllf					

	LD,	-.0415721	.0517745	-0.86	0.423	-.1434108	.0602666
	L2D,	-.1119071	.0515073	-2.17	0.030	-.2132203	-.0105939
	L12D,	-.1255582	.0492626	-2.55	0.011	-.2224562	-.0286603
	lnfibp						
	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	.0016548	.0023999	0.44	0.661	-.0036568	.0057754
	7	.0046191	.0020623	1.95	0.052	-.0000374	.0080755
	8	.0127434	.0023612	5.40	0.000	.008099	.0173878
	9	.0156048	.0025314	6.16	0.000	.0186256	.0205839
	10	.0204691	.0025295	8.09	0.000	.0154936	.0254446
	11	.0136586	.0022928	5.96	0.000	.0091487	.0181685
	12	.0133891	.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.lnflifl f(1/2,12,24)d.lnusepr i.month
```

Source	SS	df	MS	Number of obs	=	503
Model	.043188967	32	.001349655	F(32, 470)	=	111.72
Residual	.005677954	470	.000012081	Prob > F	=	0.0000
Total	.048866921	502	.000097344	R-squared	=	0.8838
				Adj R-squared	=	0.8759
				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	-0.86	0.953	-.0775989	.0731068
L3D,	.1749722	.0380384	4.60	0.000	.1002259	.2497186
L4D,	.1177758	.0368875	3.19	0.002	.045291	.1902667
L5D,	.1127275	.0389003	2.98	0.004	.0362875	.1891675
L6D,	.099976	.0406874	2.24	0.026	.0110242	.1709278
L7D,	.0099996	.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	-.1888982	-.0423337
L11D,	-.0153338	.0383706	-0.40	0.690	-.090737	.0600611
L12D,	.3692339	.0461555	8.00	0.000	.2785372	.4599307
L24D,	.19708316	.0414188	4.76	0.000	.1156426	.2784205
lnflifl						
LD,	-.1283352	.068539	-1.87	0.002	-.243016	.0063457
L2D,	-.2120246	.0677924	-3.13	0.002	-.3452383	-.0788199
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,	.2347888	.0984615	2.60	0.010	.0570268	.4125448
L12D,	-.0236969	.0938261	-0.25	0.801	-.2080674	.1606736
L24D,	-.4061036	.0863262	-4.70	0.000	-.5757366	-.2364786
month						
2	.0130388	.0021912	5.95	0.000	.0087331	.0173445
3	.0143267	.0024097	5.95	0.000	.005916	.0198618
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	.012699	.0019881	6.07	0.000	.0081632	.0159766
9	.0139913	.0019658	7.12	0.000	.0101286	.0178541
10	.0232689	.0024957	9.32	0.000	.0183568	.028165
11	.0157873	.0022757	6.94	0.000	.0113154	.0202591
12	.0128789	.0018317	7.83	0.000	.0092715	.0164792
cons	-.01163	.0016815	-6.92	0.000	-.0149341	-.0083258

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

Source	SS	df	MS	Number of obs	=	359
Model	.032942734	60	.000549046	F(60, 298)	=	46.98
Residual	.003482599	298	.000011687	Prob > F	=	0.0000
Total	.036425333	358	.000101747	R-squared	=	0.9044
				Adj R-squared	=	0.8851
				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	.0875841	-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	8.71	0.480	-.0756932	.1605066
L6D,	.0976656	.0600038	1.63	0.105	-.0204193	.2157504
L7D,	-.0007543	.0600507	-0.91	0.990	-.1198289	.1183294
L8D,	-.0676289	.0600776	-1.11	0.267	-.1872346	.0519769
L9D,	.0724574	.0600807	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
lnfl1f						
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	0.92	0.981	-.2045197	.2095296
L5D,	.0257598	.1045744	0.25	0.886	-.1800381	.2315577
L6D,	-.1051399	.1040557	-1.01	0.313	-.369917	.0996372
L7D,	-.0521678	.1035065	-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	.1929917
L12D,	-.1366552	.1019729	-1.34	0.181	-.3373335	.0648231
lnusepr						
LD,	.2199635	.1351476	1.63	0.105	-.0460011	.4859281
L2D,	.0824764	.1374525	0.68	0.549	-.1880241	.3529769
L3D,	.1238781	.1364468	0.91	0.365	-.1446433	.3923995
L4D,	.1231339	.135751	0.91	0.365	-.1440181	.398286
L5D,	-.0608881	.1344611	-0.45	0.651	-.3255017	.2037254
L6D,	.3128869	.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	-.3584779	.1842066
L10D,	.3110896	.1355495	-2.30	0.022	-.5778451	-.0443342
L11D,	.186931	.1358319	1.38	0.170	-.0883804	.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	.0098303
L3D,	.0065958	.0020468	3.22	0.001	.0025678	.0106239
L4D,	.0046992	.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	0.191	-.0006694	.0075488
L9D,	.0030855	.0020819	1.48	0.139	-.0010115	.0071825
L10D,	.0043761	.0020846	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	1.98	0.279	-.0028751	.0099346
6	-.0029861	.0038815	-0.77	0.442	-.0186247	.0046525
7	.0039106	.0036724	1.86	0.288	-.0033164	.0111377
8	.0138735	.0038896	3.57	0.000	.0002188	.0215281
9	.0114154	.0032689	3.49	0.001	.0049823	.0178485
10	.0173888	.004233	4.11	0.000	.0098577	.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	0.005	-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.lnfl1f l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date if tin(,2018m12)
```

Source	SS	df	MS	Number of obs =	369
Model	.033319208	30	.00111064	F(30, 338) =	87.31
Residual	.004293777	338	.00001272	Prob > F =	0.0000
Total	.037618585	368	.000102224	R-squared =	0.8857
				Adj R-squared =	0.8756
				Root MSE =	.003537

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	.0500658	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	.0508092	-1.84	0.067	-.1932573	.0065995
L12D,	.3768875	.0508592	7.41	0.000	.276847	.476928
lnfl1f						
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	-.1982458	.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	.0149314
5	.0028072	.0027842	0.75	0.454	-.0033893	.0075638
6	.0002948	.0025009	0.12	0.906	-.0046246	.0052141
7	.0008253	.0021584	0.38	0.702	-.0034284	.0059709
8	.0114443	.0025191	4.54	0.000	.0064878	.0163981
9	.0154884	.002613	5.93	0.000	.0103487	.0266282
10	.0212612	.0026566	8.01	0.000	.0109376	.0264847
11	.0130249	.0025092	5.19	0.000	.0088894	.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	-.00780518	.0019605	-3.68	0.000	-.010988	-.0031955

```
. scalar define rmse2=e(rmse)

. reg d.lnflnonfarm l(1/12)d.lnflnonfarm l(1/2,12)d.lnfilf l(1/2,12)d.lnflbp i.month date if tin(,2018m12)
```

Source	SS	df	MS	Number of obs	=	359
Model	.032340453	30	.001078015	F(30, 328)	=	86.56
Residual	.00408488	328	.000012454	Prob > F	=	0.0000
Total	.036425333	358	.000101747	R-squared	=	0.8879
				Adj R-squared	=	0.8776
				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	5.89	0.000	.1558465	.3524202
L4D,	.1689731	.0502832	3.36	0.001	.0700648	.2678914
L5D,	.1560811	.0507562	3.08	0.002	.0562324	.2559298
L6D,	.1559908	.0518295	3.01	0.003	.0540388	.2579509
L7D,	.094377	.0527031	1.79	0.074	-.0093018	.1988557
L8D,	.0330789	.0518367	0.64	0.524	-.0688954	.1358533
L9D,	.0893751	.0498407	1.79	0.074	-.0086727	.1874229
L10D,	-.1928458	.0486649	-3.97	0.000	-.2884625	-.0172291
L11D,	-.1106264	.0502767	-2.20	0.028	-.2095318	-.0117209
L12D,	.3558321	.050687	7.92	0.000	.2561195	.4555448
lnfilf						
LD,	-.0438007	.0531291	-0.82	0.410	-.1483174	.0607159
L2D,	-.1078659	.0528228	-2.04	0.042	-.2117983	-.0039414
L12D,	-.1212337	.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	-.004688	.0056794
7	.0035539	.0021131	1.67	0.095	-.0006179	.0076959
8	.0127989	.0024143	5.38	0.000	.0080414	.0175493
9	.0168119	.0026	6.16	0.000	.0108871	.0211268
10	.0209367	.0026979	8.83	0.000	.0158063	.026867
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.lnfilf l(1/2,12,24)d.lnusepr i.month date if tin(,2018m12)
```

Source	SS	df	MS	Number of obs	=	491
Model	.042575775	32	.001330493	F(32, 458)	=	109.93
Residual	.005543023	458	.0000012103	Prob > F	=	0.0000
Total	.048118799	490	.0000098202	R-squared	=	0.8848
				Adj R-squared	=	0.8768
				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,	.1810109	.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	7.89	0.000	.2440695	.4312969
L24D,	.2364096	.04478	5.28	0.000	.14841	.3244093
lnfl1f						
LD,	-.1308766	.0692757	-1.88	0.001	-.2662142	.0066699
L2D,	-.2188669	.0684865	-3.20	0.001	-.3534536	-.0842802
L12D,	-.0319786	.070685	-0.45	0.651	-.1788776	.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	.0949263	-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	.0156211	.0024611	6.10	0.000	.0101847	.0198576
4	.0146381	.0025223	5.57	0.000	.0090814	.0189494
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	.0042995	.0120353
8	.0119157	.0020208	5.98	0.000	.0079446	.0158868
9	.0141217	.0020005	7.06	0.000	.0101904	.0180853
10	.0242635	.0025397	9.55	0.000	.0192726	.0292543
11	.0162918	.0023257	7.01	0.000	.0117215	.0208621
12	.0127589	.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.lnfl1f l(1/12)d.lnusepr l(1/12)d.lnflbp i.month date if tin(,2018m12)

```

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

D. inflnonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
inflnonfarm						
LD,	-.1785286	.0557606	-3.20	0.002	-.2882631	-.0687941
L2D,	-.1466054	.0675841	-2.53	0.012	-.2589285	-.0322823
L3D,	.1670832	.0584604	2.86	0.005	.0520358	.2821307
L4D,	.135493	.0590676	2.29	0.023	.0191604	.2516456
L5D,	.0424067	.0600114	0.71	0.480	-.0756932	.1605066
L6D,	.0976656	.0600038	1.63	0.195	-.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	-.1017063
L11D,	-.0568776	.0572337	-0.98	0.328	-.168711	.0565558
L12D,	.3073789	.0660997	5.48	0.000	.1969771	.4177807
infl1f						
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	0.886	-.1800381	.2315577
L6D,	-.1051399	.1040557	-1.01	0.313	-.309917	.0996372
L7D,	-.0521678	.1035685	-0.50	0.615	-.2559861	.1516505
L8D,	-.0333629	.1034667	-0.34	0.733	-.238981	.1682551
L9D,	.1257419	.1027424	1.22	0.222	-.0704507	.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	.1381476	1.63	0.195	-.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	.392826
L5D,	-.0608881	.1344611	-0.45	0.651	-.3255017	.2037254
L6D,	.3128889	.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	.1842866
L10D.	-.3108896	.1355495	-2.38	0.022	-.5778451	-.0443342
L11D.	.186931	.1358319	1.38	0.170	-.0883804	.4542423
L12D.	.1507345	.1311892	1.15	0.251	-.1074402	.4089092
<i>Inflbp</i>						
LD.	.0018889	.0016344	1.16	0.249	-.0013275	.0051053
L2D.	.0052191	.0019412	2.68	0.008	.00139	.0098303
L3D.	.0065958	.0020468	3.22	0.001	.0025678	.0106239
L4D.	.00465902	.0020526	2.28	0.023	.0006598	.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	0.191	-.0006694	.0075488
L9D.	.0030855	.0020819	1.48	0.139	-.0010115	.0071825
L10D.	.0043761	.0020469	2.14	0.053	.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	1.98	0.279	-.0028751	.0099346
6	-.0029861	.0038815	-0.77	0.442	-.0106247	.0046525
7	.0039186	.0036724	1.06	0.288	-.0033164	.0111377
8	.0138735	.0038896	3.57	0.000	.0002188	.0215281
9	.0114154	.0032689	3.49	0.001	.0049233	.0178485
10	.0173888	.004233	4.11	0.000	.0098577	.0257183
11	.0094232	.0038365	2.46	0.015	.0018732	.0169733
12	.0153423	.0036493	4.20	0.000	.0001607	.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.lnlnonfarm l(1/12)d.lnlnonfarm l(1/2)d.lnflif l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date if tin(,2018m12)
```

Source	SS	df	MS	Number of obs	=	369
Model	.033319208	30	.00111064	F(30, 338)	=	87.31
Residual	.804299377	338	.00001272	Prob > F	=	0.0000
Total	.037618585	368	.000102224	R-squared	=	0.8857
				Adj R-squared	=	0.8756
				Root MSE	=	.008357
D. lnlnonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
lnlnonfarm						
LD.	-.133326	.0503841	-2.65	0.009	-.2324319	-.03422
L2D.	-.0952298	.0529421	-1.80	0.073	-.1993672	.0089876
L3D.	.2422019	.0525438	4.61	0.000	.1388477	.345556
L4D.	.1739116	.05080588	3.47	0.001	.0754456	.2723777
L5D.	.1496013	.05084662	2.94	0.004	.0493898	.2498129
L6D.	.1332644	.0521479	2.56	0.011	.0306891	.2358398
L7D.	.0730712	.05040717	1.35	0.177	-.0332882	.1794305
L8D.	.0161971	.0524574	0.31	0.758	-.086987	.1193812
L9D.	.0861744	.0499333	1.73	0.085	-.01020448	.1843937
L10D.	-.1843262	.048657	-3.79	0.000	-.2800348	-.0886175
L11D.	-.0933289	.05080022	-1.84	0.067	-.1932573	.0065995
L12D.	.3768875	.0508592	7.41	0.000	.276847	.476928
lnflif						
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	.1260638	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	.0148314
5	.0028872	.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	.0100376	.0264847
11	.0130249	.0025092	5.19	0.000	.0088894	.0179695
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	-0.000518	.0017000	-3.00	0.000	-0.010000	-0.0051200
-------	-----------	----------	-------	-------	-----------	------------

```
. 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.lnlnonfarm l(1/12)d.lnlnonfarm l(1/2,12)d.lnlllf l(1/2,12)d.lnfibp i.month date if tin(,2018m12)
```

Source	SS	df	MS	Number of obs	=	359
Model	.032340453	30	.001078015	F(30, 328)	=	86.56
Residual	.004084888	328	.000012454	Prob > F	=	0.0000
Total	.036425333	358	.000101747	R-squared	=	0.8879
				Adj R-squared	=	0.8776
				Root MSE	=	.00353

D. lnlnonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
lnlnonfarm					
LD,	-.1087415	.050957	-2.13	0.034	-.2089852 - .0084978
L2D,	-.0783122	.0511855	-1.53	0.127	-.1798055 .0223811
L3D,	.2541334	.0499623	5.09	0.000	.1558465 .3524202
L4D,	.1689731	.0502832	3.36	0.001	.0700548 .2678914
L5D,	.1568811	.0507562	3.08	0.002	.0562324 .2559298
L6D,	.1559908	.0518295	3.01	0.003	.0540308 .2579509
L7D,	.0943377	.0527831	1.79	0.074	-.0093018 .1988657
L8D,	.0330789	.0518367	0.64	0.524	-.0688954 .1350633
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
lnfifit					
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
lnfibp					
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	.0035359	.0021131	1.67	0.095	-.0006179 .0076959
8	.0127909	.0024143	5.30	0.000	.0058044 .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	-0.83	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.lnlnonfarm l(1/12,24)d.lnlnonfarm l(1/2,12,24)d.lnlllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)
```

Source	SS	df	MS	Number of obs	=	491
Model	.042575775	32	.001330493	F(32, 458)	=	109.93
Residual	.005543023	458	.000012103	Prob > F	=	0.0000
Total	.048118799	490	.0000098292	R-squared	=	0.8848
				Adj R-squared	=	0.8768
				Root MSE	=	.00348

D. lnlnonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
lnlnonfarm					
LD,	-.0311684	.0380987	-0.82	0.414	-.1060383 .0437014
L2D,	.0011509	.0387186	0.83	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	.0489493	0.11	0.910	-.0758401 .0851037
L8D	-.026522	.0392622	0.07	0.985	-.0456100 .051220

L00,	-.0345244	.0370040	-.0.07	0.000	-.1120107	.043402
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	7.89	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	-3.28	0.001	-.3534536	-.0842892
L12D,	-.0319796	.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	.0191847	.0198576
4	.0140381	.0025223	5.57	0.000	.0098814	.0189949
5	.0082786	.0023045	3.59	0.000	.0037499	.0128072
6	.0092759	.0023797	3.98	0.000	.0045993	.0139524
7	.0081629	.0019705	4.14	0.000	.0042905	.0120353
8	.0119157	.0020208	5.98	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)

*+
. reg d.lnfinonfarm l(1/12)d.lnfinonfarm l(1/12)d.lnfllf l(1/12)d.lnusepr l(1/12)d.lnfbp i.month date

```

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

D. lnfinonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
lnfinonfarm						
LD,	-.1709382	.0544484	-3.14	0.002	-.2780734	-.063803
L2D,	-.1482615	.0563272	-2.63	0.009	-.2599935	-.0374296
L3D,	.162612	.057373	2.83	0.005	.0497221	.2755018
L4D,	.1338157	.0578492	2.31	0.021	.0199889	.2476424
L5D,	.0485865	.0586353	0.83	0.468	-.066787	.1639599
L6D,	.1034826	.0583736	1.77	0.077	-.0113761	.2183412
L7D,	-.0017194	.0587665	-0.93	0.977	-.1173393	.1139005
L8D,	-.0661616	.0589687	-1.12	0.263	-.1821921	.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	-.2992948	.1025286
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	-.2333881	.1606454
L8D,	-.0227181	.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	.0985373	-0.16	0.877	-.2134433	.1822003
L12D,	-.1496896	.0991977	-1.51	0.132	-.3448754	.0454963
lnusepr						
LD,	.2160403	.1318859	1.64	0.102	-.0434644	.4765451
L2D,	.0596176	.1342172	0.44	0.657	-.2044743	.3237096
L3D,	.1365192	.1332945	1.02	0.307	-.1255752	.3987956
L4D,	.148566	.1327682	1.86	0.291	-.1266748	.4018068
L5D,	-.0429764	.1313909	-0.33	0.744	-.3015071	.2155544
L6D,	.3174743	.1387471	2.43	0.016	.0662104	.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,	.4505221	.127088	1.25	0.214	-.802322	.4412322

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	.0026984	.016404
L4D.	.0049279	.001984	2.48	0.014	.0010242	.0088316
L5D.	.0058421	.0019821	2.54	0.011	.001142	.0089422
L6D.	.0049581	.0020017	2.48	0.014	.0010194	.0088969
L7D.	.0043549	.0020066	2.17	0.031	.0004867	.0083031
L8D.	.0036497	.0020487	1.78	0.077	-.0003985	.0076719
L9D.	.0033308	.0020403	1.63	0.104	-.0006839	.0073454
L10D.	.004421	.0020819	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	.0061882
month						
2	.0096471	.0035242	2.74	0.007	.0027127	.0165816
3	.0085261	.003368	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	.0042083	.0035921	1.17	0.243	-.0028678	.012683
8	.0133789	.0037305	3.59	0.000	.0008387	.0207191
9	.0108298	.0031587	3.43	0.001	.0046146	.017045
10	.0165185	.0040273	4.10	0.000	.0088942	.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	-1.96	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.lnflif 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.lnflif l(1/2)d.lnusepr l(1/2)d.lnflbp i.month date
```

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

D. lnflnonfarm	Coeff.	Std. Err.	t	P> t	[95% Conf. Interval]	
<i>lnflnonfarm</i>						
LD.	-.1252994	.0491434	-2.55	0.011	-.2219529	-.0286459
L2D.	-.0958692	.0518133	-1.85	0.065	-.1977648	.0068443
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	.0504093	.249137
L6D.	.1371627	.0507474	2.70	0.007	.0373545	.2369708
L7D.	.0728218	.0527616	1.37	0.173	-.0317478	.1757914
L8D.	.0108807	.0513147	0.20	0.844	-.0908432	.1110047
L9D.	.0796876	.0488654	1.63	0.183	-.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
<i>lnflif</i>						
LD.	-.19426	.0941693	-2.06	0.040	-.3794688	-.0090513
L2D.	-.1402653	.0944467	-1.49	0.138	-.3260199	.0454893
<i>lnusepr</i>						
LD.	.2273847	.1224594	1.86	0.064	-.0134641	.4682334
L2D.	.0344122	.123157	0.28	0.780	-.2078886	.276633
<i>lnflbp</i>						
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	.0021982	0.59	0.557	-.0029078	.0053847
	8	.011499	.0024647	4.67	0.000	.0066515	.0163465
	9	.0150089	.002545	5.98	0.000	.0100036	.0208143
	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	Number of obs	=	371
Model	.033019839	30	.001100661	F(30, 340)	=	.90.24
Residual	.004146841	340	.000012197	Prob > F	=	0.0000
Total	.03716668	370	.000100845	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	.2624282	
L5D.	.1576085	.0494045	3.19	0.002	.0664315	.2547854	
L6D.	.1606861	.0504142	3.19	0.002	.0615231	.2598491	
L7D.	.0928205	.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.	-.1982533	.0474658	-4.18	0.000	-.2915889	-.1048616	
L11D.	-.1022931	.0491628	-2.08	0.038	-.199005	-.0055813	
L12D.	.3672239	.0495416	7.41	0.000	.2697773	.4646706	
infllf							
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	
infllp							
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	-.0036568	.0057754	
7	.0040191	.0020623	1.95	0.052	-.0000374	.0080755	
8	.0127434	.0023612	5.40	0.000	.0088099	.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	.0133891	.0019091	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)

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



| Source   | SS         | df  | MS         | Number of obs | = | 503    |
|----------|------------|-----|------------|---------------|---|--------|
| Model    | .043188967 | 32  | .001349655 | F(32, 470)    | = | 111.72 |
| Residual | .005677954 | 470 | .000012081 | Prob > F      | = | 0.0000 |
| Total    | .048866921 | 502 | .000097344 | R-squared     | = | 0.8838 |
|          |            |     |            | Adj R-squared | = | 0.8759 |
|          |            |     |            | 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 -.0775989 .0731086
L3D. .1749722 .0380384 4.68 0.000 .1002259 .2497186
L4D. .1177758 .0368875 3.19 0.002 .045291 .1902697
L5D. .1127275 .0389003 2.98 0.004 .0362875 .1891675
L6D. .090976 .0406874 2.24 0.026 .0110242 .1709278
L7D. .0090996 .0485277 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 -0.46 0.690 -.098737 .0608611
L12D. .3692339 .0461555 8.08 0.000 .2785372 .4599307
L24D. .1970816 .0414188 4.76 0.000 .1156426 .2784205

lnflif
LD. -.1283352 .068539 -1.87 0.062 -.263016 .0063457
L2D. -.2120246 .0677924 -3.13 0.002 -.3452383 -.0788169
L12D. -.0320849 .0699833 -0.46 0.647 -.1695679 .18547
L24D. .2125143 .0650669 3.27 0.001 .0846562 .3403724

lnusepr
LD. .1405225 .0097752 1.55 0.122 -.0378531 .318898
L2D. .2347858 .0904615 2.60 0.010 .0570268 .4125448
L12D. -.0236969 .0938261 -0.25 0.801 -.2088674 .1606736
L24D. -.4061036 .0863262 -4.70 0.000 -.5757366 -.2364706

month
2 .0130388 .0021912 5.95 0.000 .0087331 .0173445
3 .0142367 .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 6.87 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 7.83 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                | .86890652 |



. 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 .00388036
Model 2       31 -3190.9568 -3068.7301 .00356652 .00371849
Model 3       31 -3115.1451 -2993.7429 .00352981 .00375319
Model 4       33 -4236.6004 -4097.3289 .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, 468) = 109.93
Model | .042575775  32 .001330493 Prob > F = 0.0000
Residual | .005543023  458 .000012103 R-squared = 0.8848
Total | .048118799  490 .0000098202 Adj R-squared = 0.8768
                    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. .0011589 .0387186 0.83 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 .1907197
L6D. .0897111 .0411525 2.18 0.030 .0088401 .1705822
L7D. .0046318 .0489493 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 -.0369561
L11D. -.0221162 .038865 -0.57 0.567 -.0981196 .0537872
L12D. .3376832 .0476368 7.89 0.000 .2440695 .4312969
L24D. .2364096 .044478 5.28 0.000 .14841 .3244093

lnllf
LD. -.1300766 .0692757 -1.88 0.061 -.2662142 .0060609
L2D. -.2188669 .0684865 -3.20 0.001 -.3534536 -.0842802
L12D. -.0319786 .070685 -0.45 0.651 -.1788776 .1069365
L24D. .2248965 .065662 3.43 0.001 .0959428 .3538501

lnusepr
LD. .1537422 .092086 1.67 0.096 -.0272212 .3347886
L2D. .2542549 .0916164 2.78 0.006 .0742143 .4342955
L12D. -.0218538 .0949263 -0.23 0.818 -.2083793 .1646718
L24D. -.4432243 .0880674 -5.93 0.000 -.6162905 -.2701581

month
2 .0133765 .002229 6.08 0.000 .0089962 .0177569
3 .0156211 .0024611 6.10 0.000 .0101847 .0198576
4 .0140381 .0025223 5.57 0.000 .0090814 .0189494
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 .0042985 .0128353
8 .0119157 .0020028 5.98 0.000 .0079446 .0158868
9 .0141217 .0020005 7.96 0.000 .0101904 .018053
10 .0242635 .0025397 9.55 0.000 .0192726 .0292543
11 .0162918 .0023257 7.01 0.000 .0117215 .0208621
12 .0127859 .0018654 6.84 0.000 .0099932 .0164248

_cons -.0119962 .0017171 -6.59 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.lnlnonfarm+nonfarm6)*exp(.5*e(rmse)^2)
(469 missing values generated)

. gen ubnonfarm6=exp(l.lnlnonfarm+nonfarm6+1.96*stdfore6)*exp(-.5*e(rmse)^2)
(469 missing values generated)

. gen lbnonfarm6=exp(l.lnlnonfarm+nonfarm6-1.96*stdfore6)*exp(-.5*e(rmse)^2)
(469 missing values generated)

. tsline ubpnonfarm6 lpnonfarm6 pnonfarm6 fl_nonfarm if tin(2016m12,2019m12), tline(2018m12)

.

. *7
. reg d.lnlnonfarm l(1/12,24)d.lnlnonfarm l(1/2,12,24)d.lnllf l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)

```

Source	SS	df	MS	Number of obs	=	491
Model	.042575775	32	.001330493	F(32, 458)	=	109.93
Residual	.005543023	458	.000012193	Prob > F	=	0.0000
Total	.048118799	490	.0000098292	R-squared	=	0.8848
				Adj R-squared	=	0.8768
				Root MSE	=	.00348

D. lnlnonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
lnlnonfarm					
LD.	-.0311684	.0380987	-0.82	0.414	-.1060383 .0437014
L2D.	.0011589	.0387186	0.83	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	.0489493	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
lnllf					
LD.	-.1300766	.0692757	-1.88	0.061	-.2662142 .0060609
L2D.	-.2188669	.0684865	-3.20	0.001	-.3534536 -.0842892
L12D.	-.0319706	.0706085	-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	6.08	0.000	.0089962 .0177569
3	.0150211	.0024611	6.10	0.000	.0101847 .0198576
4	.0140381	.0025223	5.57	0.000	.0099814 .0189449
5	.0082786	.0023045	3.59	0.000	.0037499 .0128072
6	.0092759	.0023797	3.90	0.000	.0045993 .0137524
7	.0081629	.0019705	4.14	0.000	.0042985 .0128353
8	.019157	.0020208	5.90	0.000	.0079446 .0158868
9	.0141217	.0020005	7.06	0.000	.0101904 .0180653
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	.0099932 .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
(497 missing values generated)

. gen expres47=exp(pres47) if tin(2016m12,2018m12)
(497 missing values generated)

. summu expres47

Variable | Obs Mean Std. Dev. Min Max
expres47 | 25 .9995604 .0061667 .9796714 1.016379

. gen pnonfarm47=r(mean)*exp(l.lnlnonfarm+nonfarm47)
(469 missing values generated)

. _pctile expres47, percentile(2.5,97.5)

. gen lpnonfarm47=r(r1)*exp(l.lnlnonfarm+nonfarm47)
(469 missing values generated)

. gen ubpnonfarm47=r(r2)*exp(l.lnlnonfarm+nonfarm47)

```

```

(469 missing values generated)

. tsline ubpnnonfarm47 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
       Model | .04257575    32  .001330493   F(32, 48) =  109.93
       Residual | .005543023   458  .000012103   Prob > F =  0.0000
                    R-squared =  0.8848
                    Adj R-squared =  0.8768
                    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  0.83  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  .1907197
L6D. | .0897111  .0411525  2.18  0.030  .0088401  .1705822
L7D. | .0046318  .0489493  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

lnflif |
LD. | -.1300766  .0692757  -1.88  0.061  -.2662142  .0066699
L2D. | -.2188669  .0684865  -3.20  0.001  -.3534536  -.0842892
L12D. | -.0319706  .070685  -0.45  0.651  -.1708776  .1069365
L24D. | .2248965  .06562  3.43  0.001  .0989428  .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  -.6162985  -.2701581

month |
2 | .0133765  .002229  6.80  0.000  .0089962  .0177569
3 | .0150211  .0024611  6.10  0.000  .0101847  .0198576
4 | .0140381  .0025223  5.57  0.000  .0090814  .0189494
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  .0101984  .0180853
10 | .0242635  .0025397  9.55  0.000  .0192726  .0292543
11 | .0162918  .0023257  7.81  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 expres9=exp(pres9) if tin(,2019m12)
(470 missing values generated)

. summ expres9

Variable |     Obs      Mean    Std. Dev.      Min      Max
expres9 |     503  1.000002  .0033683  .9796714  1.016379

. gen pnonfarm9=r(mean)*exp(l.lnflnonfarm+nonfarm9)
(469 missing values generated)

. _pctile express9, 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.lnflif l(1/2,12,24)d.lnusepr i.month if tin(,2018m12)

```

Source	SS	DT	MS	NUMBER OF OBS	=	491
Model	.042575775	32	.001330493	F(32, 458)	=	109.93
Residual	.005543023	458	.000012103	Prob > F	=	0.0000
Total	.048118799	490	.0000098202	R-squared	=	0.8848
				Adj R-squared	=	0.8768
				Root MSE	=	.00348

D. lnlnonfarm	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
lnlnonfarm					
LD.	-.0311684	.0380987	-0.82	0.414	-.1060383 .0437014
L2D.	.0011569	.0387186	0.83	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	.0489493	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	-.1846961 -.0369951
L11D.	-.0221662	.03865	-0.57	0.567	-.0981196 .0537872
L12D.	.3376832	.0476368	7.09	0.000	.2440695 .4312969
L14D.	.2364096	.04478	5.28	0.000	.14841 .3244093
lnflif					
LD.	-.1300766	.0692757	-1.88	0.061	-.2662142 .0060609
L2D.	-.2188669	.0684865	-3.20	0.001	-.3534536 -.0842802
L12D.	-.0319706	.0706055	-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	6.18	0.000	.0191847 .0198576
4	.0140381	.0025223	5.57	0.000	.0090814 .0189949
5	.0082786	.0023045	3.59	0.000	.0037499 .0128072
6	.0092759	.0023797	3.98	0.000	.0045993 .0139524
7	.0081629	.0019705	4.14	0.000	.0042985 .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(1.lnlnonfarm+nonfarm10)*exp(.5*e(rmse)^2)
(469 missing values generated)

. gen ubnonfarm10=exp(1.lnlnonfarm+nonfarm10+1.96*stdfore10)*exp(.5*e(rmse)^2)
(469 missing values generated)

. gen lbnonfarm10=exp(1.lnlnonfarm+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

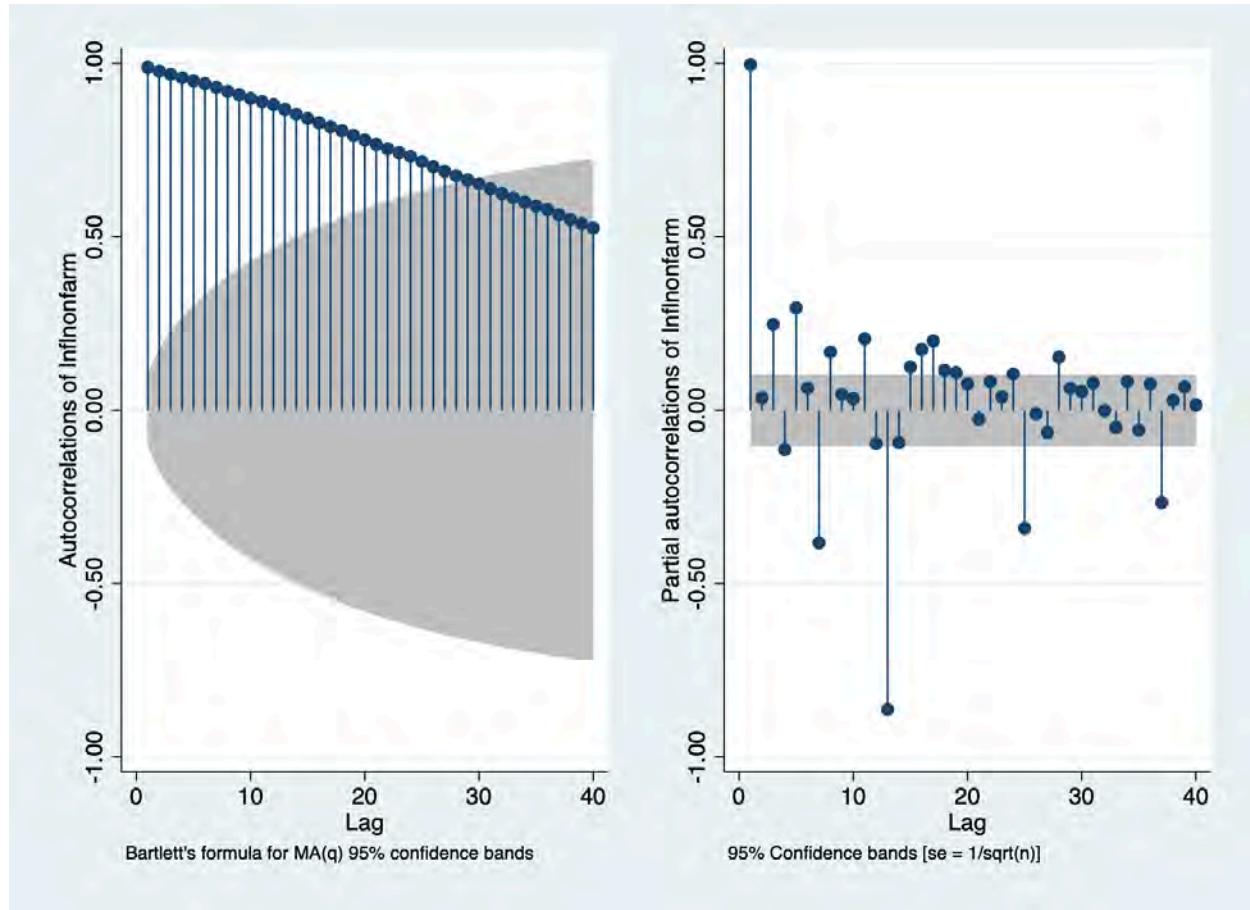
Table of Contents

Section
<u>Introduction</u>
<u>GSREG, Rolling Window, and Choosing Models</u>
<u>GSREG</u>
<u>Rolling Window</u>
<u>Using the Best Model to Forecast January 2020</u>
<u>Illustrations and Interpretations of the Models</u>
<u>Conclusion</u>
<u>Appendix A</u>
<u>Appendix B</u>

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`, `lnfl1f`, 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 13d.lnflnonfarm 16d.lnflnonfarm 112d.lnflnonfarm 124d.lnflnonfarm 124d.lnfl1f 16d.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 112d.lnflnonfarm m1 m3 m6 m9`. The third model I chose was of an average length at eight variables: `reg d.lnflnonfarm 13d.lnflnonfarm 112d.lnflnonfarm 124d.lnflnonfarm 124d.lnflnonfarm 16d.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

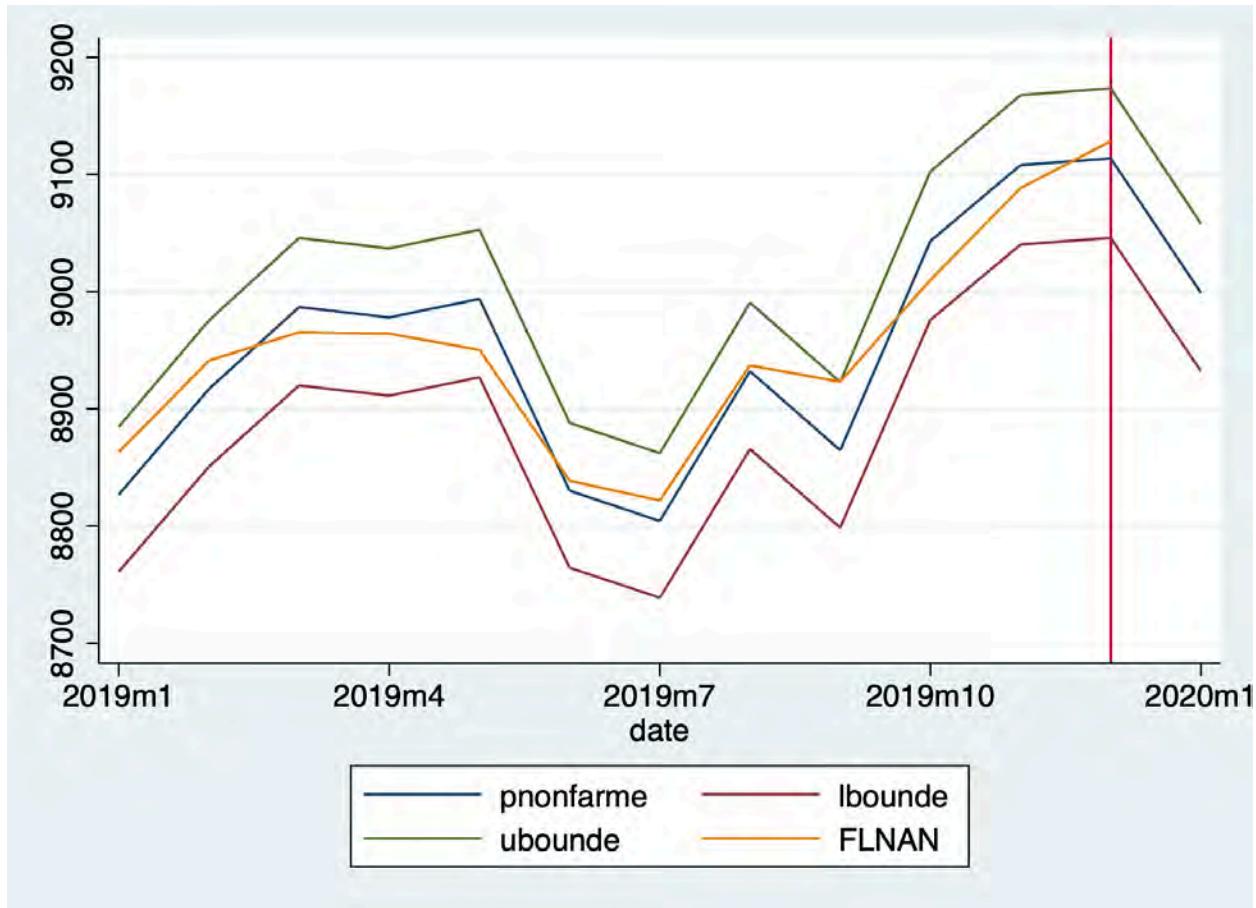
We can then use our best model and window width to forecast. If we limit our model data to several data points in the past, we can forecast the most recent past datapoints to check our model's accuracy.

month	pnonfarm	lbound	ubound
1	9001.077	8933.007	9069.667

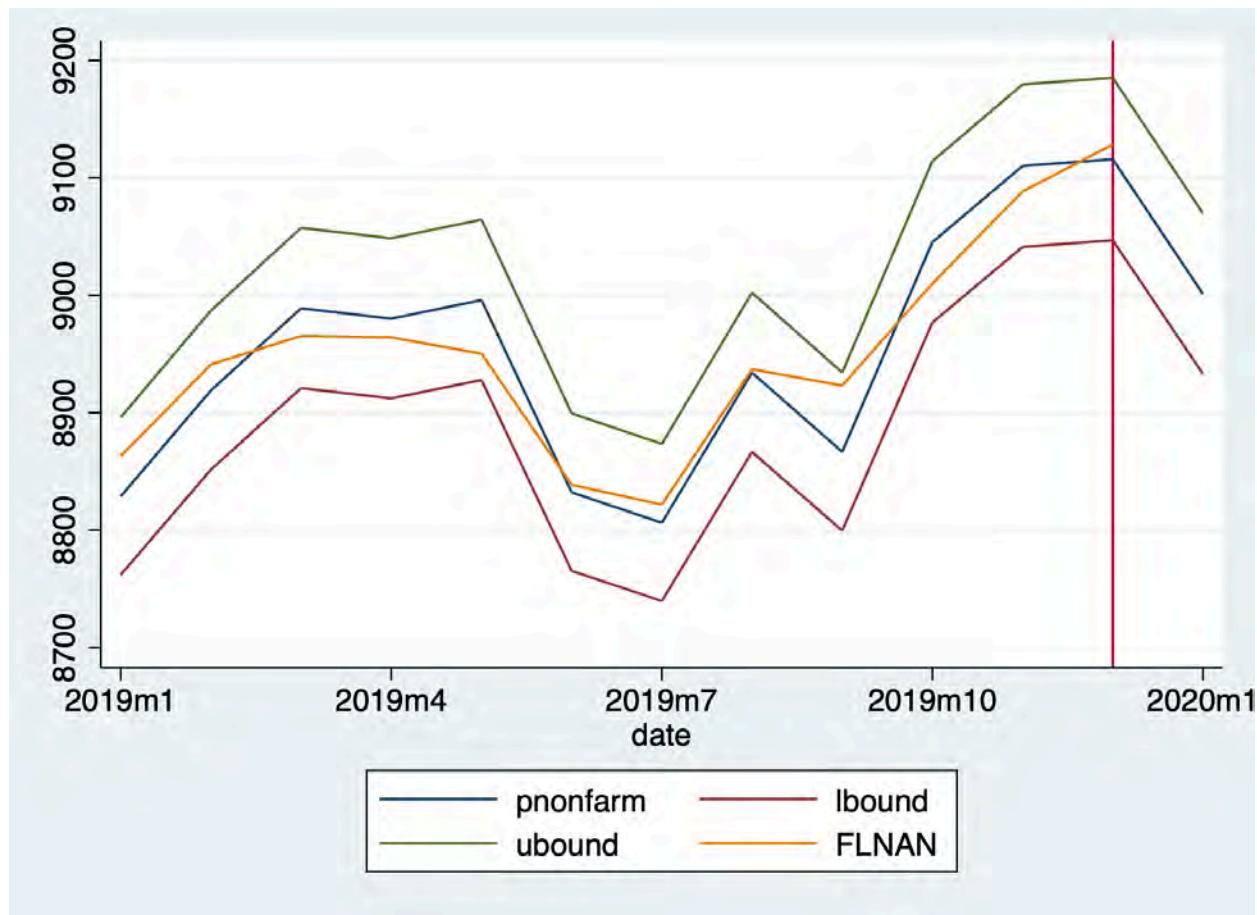
Table 1: One Month Ahead Forecast Predicting Non-farm Employment for January 2020

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.

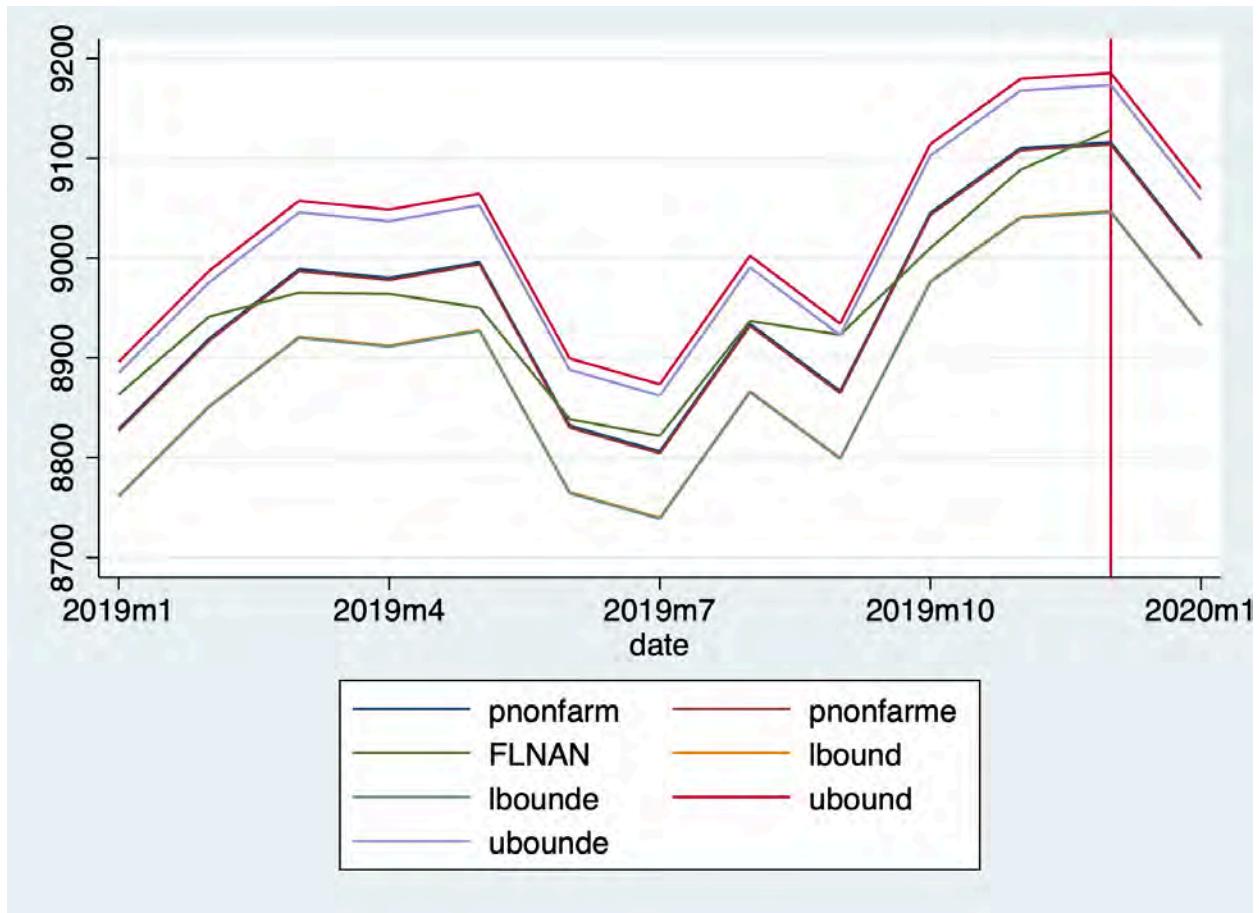
Illustrations and Interpretations of the Models



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)

```
1 clear
2 set more off
3
4 cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets/Problem Set 5"
5 log using "Problem Set 5", replace
6 import delimited "Assignment_1_Monthly.txt"
7
8 rename lnu02300000 us_epr
9 rename flnan fl_nonfarm
10 rename fllfn fl_lf
11 rename flbpriv fl_bp
12 rename date datestring
13
14 gen datec=date(datestring, "YMD")
15 gen date=mofd(datec)
16 gen month=month(datec)
17 format date %tm
18
19 tsset date
20
21 gen lnusepr=log(us_epr)
22 gen lnflnonfarm=log(fl_nonfarm)
23 gen lnfllf=log(fl_lf)
24 gen lnflbp=log(fl_bp)
25
26 *1
27 drop if !tin(1990m1,2019m12)
28
29 *2
30 tsset date
31 tsappend, add(1)
32 replace month=month(dofm(date)) if month==.
33
34 *interlude
35 ac lnflnonfarm, saving("ac_lnflnonfarm.gph", replace)
36 pac lnflnonfarm, saving("pac_lnflnonfarm.gph", replace)
37 graph combine ac_lnflnonfarm.gph pac_lnflnonfarm.gph
38 graph export ac_pac_lnflnonfarm.png, replace
39
40 *3
41 gen m1=0
42 replace m1=1 if month==1
43 gen m2=0
```

```
44 replace m2=1 if month==2
45 gen m3=0
46 replace m3=1 if month==3
47 gen m4=0
48 replace m4=1 if month==4
49 gen m5=0
50 replace m5=1 if month==5
51 gen m6=0
52 replace m6=1 if month==6
53 gen m7=0
54 replace m7=1 if month==7
55 gen m8=0
56 replace m8=1 if month==8
57 gen m9=0
58 replace m9=1 if month==9
59 gen m10=0
60 replace m10=1 if month==10
61 gen m11=0
62 replace m11=1 if month==11
63
64 gen dlnflnonfarm=d.lnflnonfarm
65 gen l1dlnflnonfarm=l1d.lnflnonfarm
66 gen l2dlnflnonfarm=l2d.lnflnonfarm
67 gen l3dlnflnonfarm=l3d.lnflnonfarm
68 gen l4dlnflnonfarm=l4d.lnflnonfarm
69 gen l5dlnflnonfarm=l5d.lnflnonfarm
70 gen l6dlnflnonfarm=l6d.lnflnonfarm
71 gen l7dlnflnonfarm=l7d.lnflnonfarm
72 gen l8dlnflnonfarm=l8d.lnflnonfarm
73 gen l9dlnflnonfarm=l9d.lnflnonfarm
74 gen l10dlnflnonfarm=l10d.lnflnonfarm
75 gen l11dlnflnonfarm=l11d.lnflnonfarm
76 gen l12dlnflnonfarm=l12d.lnflnonfarm
77 gen l24dlnflnonfarm=l24d.lnflnonfarm
78
79 gen dlnfllf=d.lnfllf
80 gen l1dlnfllf=l1d.lnfllf
81 gen l2dlnfllf=l2d.lnfllf
82 gen l3dlnfllf=l3d.lnfllf
83 gen l4dlnfllf=l4d.lnfllf
84 gen l5dlnfllf=l5d.lnfllf
85 gen l6dlnfllf=l6d.lnfllf
86 gen l7dlnfllf=l7d.lnfllf
87 gen l8dlnfllf=l8d.lnfllf
88 gen l9dlnfllf=l9d.lnfllf
89 gen l10dlnfllf=l10d.lnfllf
90 gen l11dlnfllf=l11d.lnfllf
91 gen l12dlnfllf=l12d.lnfllf
```

```

92 gen 124dlnflf=124d.lnflf
93
94 gen dlnusepr=d.lnusepr
95 gen 11dlnusepr=11d.lnusepr
96 gen 12dlnusepr=12d.lnusepr
97 gen 13dlnusepr=13d.lnusepr
98 gen 14dlnusepr=14d.lnusepr
99 gen 15dlnusepr=15d.lnusepr
100 gen 16dlnusepr=16d.lnusepr
101 gen 17dlnusepr=17d.lnusepr
102 gen 18dlnusepr=18d.lnusepr
103 gen 19dlnusepr=19d.lnusepr
104 gen 110dlnusepr=110d.lnusepr
105 gen 111dlnusepr=111d.lnusepr
106 gen 112dlnusepr=112d.lnusepr
107 gen 124dlnusepr=124d.lnusepr
108
109
110 gsreg dlnflnonfarm 11dlnflnonfarm 13dlnflnonfarm 16dlnflnonfarm 19dlnflnonfarm
/// 
111     112dlnflnonfarm 124dlnflnonfarm ///
112     11dlnflf 13dlnflf 16dlnflf 19dlnflf ///
113     112dlnflf 124dlnflf ///
114     11dlnusepr 13dlnusepr 16dlnusepr 19dlnusepr ///
115     112dlnusepr 124dlnusepr if tin(1990m1,2019m12), ///
116     ncomb(1,6) aic outsample(24) fix(m1 m3 m6 m9) ///
117     samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnrer) replace
118
119
120 *5
121 /*
122 Best model
123 reg dlnflnonfarm 13dlnflnonfarm 16dlnflnonfarm 112dlnflnonfarm 124dlnflnonfarm
124     124dlnflf 16dlnusepr m1 m3 m6 m9
125 */
126 scalar drop _all
127 quietly forval w=48(12)144 {
128     gen pred=.
129     gen nobs=.
130     forval t=529/720 {
131         gen wstart=`t'-'w'
132         gen wend=`t'-1
133         reg d.lnflnonfarm 13d.lnflnonfarm 16d.lnflnonfarm 112d.lnflnonfarm
134             124d.lnflnonfarm ///
135             124d.lnfllf 16d.lnusepr m1 m3 m6 m9 ///
136             if date>=wstart & date<=wend
137             replace nobs=e(N) if date==`t'
138             predict ptemp

```

```

138 replace pred=ptemp if date==`t'
139 drop ptemp wstart wend
140 }
141 gen errsq=(pred-d.lnflnonfarm)^2
142 summ errsq
143 scalar RWrmse`w'=r(mean)^.5
144 summ nobs
145 scalar RWminobs`w'=r(min)
146 scalar RWmaxobs`w'=r(max)
147 drop errsq pred nobs
148 }
149 scalar list
150 /*
151 RWmaxobs108 = 108
152 RWminobs108 = 108
153 RWrmse108 = .00388844
154 */
155 /*
156 Smallest / best model
157 reg dlnflnonfarm l12dlnflnonfarm m1 m3 m6 m9
158 */
159 scalar drop _all
160 quietly forval w=48(12)144 {
161 gen pred=.
162 gen nobs=.
163 forval t=529/720 {
164     gen wstart=`t'-`w'
165     gen wend=`t'-1
166     reg dlnflnonfarm l12dlnflnonfarm m1 m3 m6 m9 ///
167         if date>=wstart & date<=wend
168     replace nobs=e(N) if date==`t'
169     predict ptemp
170     replace pred=ptemp if date==`t'
171     drop ptemp wstart wend
172 }
173 gen errsq=(pred-d.lnflnonfarm)^2
174 summ errsq
175 scalar RWrmse`w'=r(mean)^.5
176 summ nobs
177 scalar RWminobs`w'=r(min)
178 scalar RWmaxobs`w'=r(max)
179 drop errsq pred nobs
180 }
181 scalar list
182 /*
183 RWmaxobs120 = 120
184 RWminobs120 = 120

```

```

186 RWrmse120 = .00423688
187 /*
188 */
189 /*
190 */
191 Best medium length model
192 reg dlnflnonfarm 13dlnflnonfarm 112dlnflnonfarm 124dlnflnonfarm 16dlnusepr
193   m1 m3 m6 m9
194 /*
195 scalar drop _all
196 quietly forval w=48(12)144 {
197 gen pred=.
198 gen nobs=.
199 forval t=529/720 {
200   gen wstart=`t'-'w'
201   gen wend=`t'-1
202   reg dlnflnonfarm 13dlnflnonfarm 112dlnflnonfarm 124dlnflnonfarm 16dlnusepr ///
203     m1 m3 m6 m9 ///
204     if date>=wstart & date<=wend
205   replace nobs=e(N) if date==`t'
206   predict ptemp
207   replace pred=ptemp if date==`t'
208   drop ptemp wstart wend
209 }
210 gen errsq=(pred-d.lnflnonfarm)^2
211 summ errsq
212 scalar RWrmse`w'=r(mean)^.5
213 summ nobs
214 scalar RWminobs`w'=r(min)
215 scalar RWmaxobs`w'=r(max)
216 drop errsq pred nobs
217 }
218 scalar list
219 /*
220 RWmaxobs108 = 108
221 RWminobs108 = 108
222 RWrmse108 = .00406403
223 */
224 /*
225 *6
226 /*
227 RWmaxobs108 = 108
228 RWminobs108 = 108
229 RWrmse108 = .00388844
230 */
231 scalar drop _all
232 quietly forval w=156(12)156 {
233   gen pred=.

```

```

234 gen nobs=.
235   forval t=432/720 {
236     gen wstart=`t'-'w'
237     gen wend=`t'-1
238     reg d.lnflnonfarm 13d.lnflnonfarm 16d.lnflnonfarm 112d.lnflnonfarm
239     124d.lnflnonfarm ///
240       124d.lnfllf 16d.lnusepr m1 m3 m6 m9 ///
241       if date>=wstart & date<=wend
242       replace nobs=e(N) if date==`t'
243       predict ptemp
244       replace pred=ptemp if date==`t'
245       drop ptemp wstart wend
246   }
247   gen errsq=(pred-d.lnflnonfarm)^2
248 }
249 summ nobs // checking all had a full window
250 *get error info for normal interval
251 summ errsq
252 scalar rwmse=r(mean)^0.5
253 scalar list rwmse
254 gen res=(d.lnflnonfarm-pred)
255 _pctile res, percentile(2.5,97.5)
256 return list
257
258 *7
259 predict temp if tin(2020m1,2020m1)
260 replace pred=temp if tin(2020m1,2020m1)
261 drop temp
262 gen pnonfarm=exp(l.lnflnonfarm+pred+(rwmse^2)/2)
263 gen ubound=exp(l.lnflnonfarm+pred+1.96*rwmse+(rwmse^2)/2)
264 gen lbound=exp(l.lnflnonfarm+pred-1.96*rwmse+(rwmse^2)/2)
265 list month pnonfarm lbound ubound if tin(2020m1,2020m1)
266 tsline pnonfarm lbound ubound fl_nonfarm if tin(2019m1,2020m1), tline(2019m12)
267 saving("Nonfarm_Normal", replace)
268 graph export "nonfarm_normal.png", replace
269
270 *8
271 *Empirical
272 drop res
273 gen res=(d.lnflnonfarm-pred)
274 gen expres=exp(res)
275 summ expres
276 scalar meanexpres=r(mean)
277 gen pnonfarme=exp(l.lnflnonfarm+pred)*meanexpres
278 _pctile res, percentile(2.5,97.5)
279 return list
280 gen lbounde=exp(l.lnflnonfarm+pred+r(r1))*meanexpres
281 gen ubounde=exp(l.lnflnonfarm+pred+r(r2))*meanexpres

```

```
280 list month pnonfarme lbounde ubounde if tin(2020m1,2020m1)
281 tsline pnonfarme lbounde ubounde fl_nonfarm if tin(2019m1,2020m1), ///
282     tline(2019m12) saving("Nonfarm_Epirical", replace)
283 graph export "nonfarm_empirical.png", replace
284
285 *9
286 tsline pnonfarm pnonfarme fl_nonfarm lbound lbounde ubound ubounde ///
287     if tin(2019m1,2020m1), tline(2019m12) saving("Normal_vs_Empirical", replace)
288 graph export "normal_vs_empirical.png", replace
289
290 translate "Problem Set 5.smcl" "Problem Set 5.txt", replace
291 log close
```

Appendix B (STATA Output)

```
1      _____(R) _____
2      _____/ _____/ _____/ _____/ _____
3      /____/ _____/ _____/ _____/ _____/ _____
4                                         Statistics/Data analysis
5
6  -----
7
8      name: <unnamed>
9      log: /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
10
11     Series/Probl
12         > em Sets/Problem Set 5/Problem Set 5.smcl
13         log type: smcl
14         opened on: 3 Apr 2021, 21:50:15
15
16         1 . import delimited "Assignment_1_Monthly.txt"
17             (5 vars, 984 obs)
18
19         2 .
20
21         3 . rename lnu02300000 us_epr
22
23         4 . rename flnan fl_nonfarm
24
25         5 . rename fllfn fl_lf
26
27         6 . rename flbpriv fl_bp
28
29         7 . rename date datestring
30
31         8 .
32         9 . gen datec=date(datestring, "YMD")
33
34         10 . gen date=mofd(datec)
35
36         11 . gen month=month(datec)
37
38         12 . format date %tm
39
40         13 .
41         14 . tsset date
42             time variable: date, 1939m1 to 2020m12
```

```

39          delta: 1 month
40
41      15 .
42 16 . gen lnusepr=log(us_epr)
43     (108 missing values generated)
44
45 17 . gen lnflnonfarm=log(fl_nonfarm)
46
47 18 . gen lnfllf=log(fl_lf)
48     (444 missing values generated)
49
50 19 . gen lnflbp=log(fl_bp)
51     (588 missing values generated)
52
53 20 .
54 21 . *1
55 22 . drop if !tin(1990m1,2019m12)
56     (624 observations deleted)
57
58 23 .
59 24 . *2
60 25 . tsset date
61     time variable: date, 1990m1 to 2019m12
62     delta: 1 month
63
64 26 . tsappend, add(1)
65
66 27 . replace month=month(dofm(date)) if month==.
67     (1 real change made)
68
69 28 .
70 29 . *interlude
71 30 . ac lnflnonfarm, saving("ac_lnflnonfarm.gph", replace)
72     (file ac_lnflnonfarm.gph saved)
73
74 31 . pac lnflnonfarm, saving("pac_lnflnonfarm.gph", replace)
75     (file pac_lnflnonfarm.gph saved)
76
77 32 . graph combine ac_lnflnonfarm.gph pac_lnflnonfarm.gph
78
79 33 . graph export ac_pac_lnflnonfarm.png, replace
80     (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets
81     > /Problem Set 5/ac_pac_lnflnonfarm.png written in PNG format)
82
83 34 .
84 35 . *3
85 36 . gen m1=0

```

```
86  
87     37 . replace m1=1 if month==1  
88         (31 real changes made)  
89  
90     38 . gen m2=0  
91  
92     39 . replace m2=1 if month==2  
93         (30 real changes made)  
94  
95     40 . gen m3=0  
96  
97     41 . replace m3=1 if month==3  
98         (30 real changes made)  
99  
100    42 . gen m4=0  
101  
102    43 . replace m4=1 if month==4  
103         (30 real changes made)  
104  
105    44 . gen m5=0  
106  
107    45 . replace m5=1 if month==5  
108         (30 real changes made)  
109  
110    46 . gen m6=0  
111  
112    47 . replace m6=1 if month==6  
113         (30 real changes made)  
114  
115    48 . gen m7=0  
116  
117    49 . replace m7=1 if month==7  
118         (30 real changes made)  
119  
120    50 . gen m8=0  
121  
122    51 . replace m8=1 if month==8  
123         (30 real changes made)  
124  
125    52 . gen m9=0  
126  
127    53 . replace m9=1 if month==9  
128         (30 real changes made)  
129  
130    54 . gen m10=0  
131  
132    55 . replace m10=1 if month==10  
133         (30 real changes made)
```

```
134
135      56 . gen m11=0
136
137      57 . replace m11=1 if month==11
138          (30 real changes made)
139
140      58 .
141      59 . gen dlnflnonfarm=d.lnflnonfarm
142          (2 missing values generated)
143
144      60 . gen l1dlnflnonfarm=l1d.lnflnonfarm
145          (2 missing values generated)
146
147      61 . gen l2dlnflnonfarm=l2d.lnflnonfarm
148          (3 missing values generated)
149
150      62 . gen l3dlnflnonfarm=l3d.lnflnonfarm
151          (4 missing values generated)
152
153      63 . gen l4dlnflnonfarm=l4d.lnflnonfarm
154          (5 missing values generated)
155
156      64 . gen l5dlnflnonfarm=l5d.lnflnonfarm
157          (6 missing values generated)
158
159      65 . gen l6dlnflnonfarm=l6d.lnflnonfarm
160          (7 missing values generated)
161
162      66 . gen l7dlnflnonfarm=l7d.lnflnonfarm
163          (8 missing values generated)
164
165      67 . gen l8dlnflnonfarm=l8d.lnflnonfarm
166          (9 missing values generated)
167
168      68 . gen l9dlnflnonfarm=l9d.lnflnonfarm
169          (10 missing values generated)
170
171      69 . gen l10dlnflnonfarm=l10d.lnflnonfarm
172          (11 missing values generated)
173
174      70 . gen l11dlnflnonfarm=l11d.lnflnonfarm
175          (12 missing values generated)
176
177      71 . gen l12dlnflnonfarm=l12d.lnflnonfarm
178          (13 missing values generated)
179
180      72 . gen l24dlnflnonfarm=l24d.lnflnonfarm
181          (25 missing values generated)
```

```
182
183    73 .
184    74 . gen dlnfllf=d.lnfllf
185        (2 missing values generated)
186
187    75 . gen l1dlnfllf=l1d.lnfllf
188        (2 missing values generated)
189
190    76 . gen l2dlnfllf=l2d.lnfllf
191        (3 missing values generated)
192
193    77 . gen l3dlnfllf=l3d.lnfllf
194        (4 missing values generated)
195
196    78 . gen l4dlnfllf=l4d.lnfllf
197        (5 missing values generated)
198
199    79 . gen l5dlnfllf=l5d.lnfllf
200        (6 missing values generated)
201
202    80 . gen l6dlnfllf=l6d.lnfllf
203        (7 missing values generated)
204
205    81 . gen l7dlnfllf=l7d.lnfllf
206        (8 missing values generated)
207
208    82 . gen l8dlnfllf=l8d.lnfllf
209        (9 missing values generated)
210
211    83 . gen l9dlnfllf=l9d.lnfllf
212        (10 missing values generated)
213
214    84 . gen l10dlnfllf=l10d.lnfllf
215        (11 missing values generated)
216
217    85 . gen l11dlnfllf=l11d.lnfllf
218        (12 missing values generated)
219
220    86 . gen l12dlnfllf=l12d.lnfllf
221        (13 missing values generated)
222
223    87 . gen l24dlnfllf=l24d.lnfllf
224        (25 missing values generated)
225
226    88 .
227    89 . gen dlnusepr=d.lnusepr
228        (2 missing values generated)
229
```

```

230 . gen 11dlnusepr=11d.lnusepr
      (2 missing values generated)
231
232
233 . gen 12dlnusepr=12d.lnusepr
      (3 missing values generated)
234
235
236 . gen 13dlnusepr=13d.lnusepr
      (4 missing values generated)
237
238
239 . gen 14dlnusepr=14d.lnusepr
      (5 missing values generated)
240
241
242 . gen 15dlnusepr=15d.lnusepr
      (6 missing values generated)
243
244
245 . gen 16dlnusepr=16d.lnusepr
      (7 missing values generated)
246
247
248 . gen 17dlnusepr=17d.lnusepr
      (8 missing values generated)
249
250
251 . gen 18dlnusepr=18d.lnusepr
      (9 missing values generated)
252
253
254 . gen 19dlnusepr=19d.lnusepr
      (10 missing values generated)
255
256
257 . gen 110dlnusepr=110d.lnusepr
      (11 missing values generated)
258
259
260 . gen 111dlnusepr=111d.lnusepr
      (12 missing values generated)
261
262
263 . gen 112dlnusepr=112d.lnusepr
      (13 missing values generated)
264
265
266 . gen 124dlnusepr=124d.lnusepr
      (25 missing values generated)
267
268
269 . 103 .
270 .
271 . gsreg dlnflnonfarm 11dlnflnonfarm 13dlnflnonfarm 16dlnflnonfarm
19dlnflnonfar
272     > m ///
273     >     112dlnflnonfarm 124dlnflnonfarm ///
274     >         11dlnfl1f 13dlnfl1f 16dlnfl1f 19dlnfl1f ///
275     >         112dlnfl1f 124dlnfl1f ///
276     >             11dlnusepr 13dlnusepr 16dlnusepr 19dlnusepr ///

```

```

277      >      112dlnusepr 124dlnusepr if tin(1990m1,2019m12), ///
278      >      ncomb(1,6) aic outsamp(24) fix(m1 m3 m6 m9) ///
279      >      samesample nindex( -1 aic -1 bic -1 rmse_out)
results(gsreg_dlnrer) r
280      > eplace
281 -----
282      Total Number of Estimations: 31179
283 -----
284 -----
285      Warning: Estimation could take about 14 minutes
286 -----
287
288      Computing combinations...
289      Preparing regression list...
290      Doing regressions...
291      Saving results...
292      file gsreg_dlnrer.dta saved
293 -----
294      Best estimation in terms of -1 aic -1 bic -1 rmse_out
295      Estimation number 19215
296 -----
297
298      Source |      SS          df          MS          Number of obs     =
312
299      -----+----- F(10, 301)     =
192.38
300      Model |  .027480005      10   .002748  Prob > F     =
0.0000
301      Residual |  .004299465      301  .000014284 R-squared     =
0.8647
302      -----+----- Adj R-squared     =
0.8602
303      Total |  .03177947      311  .000102185 Root MSE     =
.00378
304
305 -----
306      --
307      dlnflnonfarm |      Coef.    Std. Err.      t      P>|t|      [95% Conf.
Interv
308      > 1]
309      -----+
310      --
311      m1 |  -.0068526   .0013736    -4.99    0.000   -.0095558
-.00414
312      > 94

```

```

313          m3 |   .0009126   .0008591    1.06   0.289   -.0007781
314          .00260
315          > 33
316          m6 |  -.0044233   .0010173   -4.35   0.000   -.0064252
317          -.00242
318          > 14
319          m9 |  -.0008321   .0008569   -0.97   0.332   -.0025184
320          .00085
321          > 41
322          13dlnflnonfarm |   .0992292   .0282426    3.51   0.001   .0436512
323          .15480
324          > 71
325          16dlnflnonfarm |   .0741191   .0284776    2.60   0.010   .0180786
326          .13015
327          > 96
328          112dlnflnonfarm |   .5446389   .067997    8.01   0.000   .4108292
329          .67844
330          > 87
331          124dlnflnonfarm |   .1684682   .0598024    2.82   0.005   .0507844
332          .2861
333          > 52
334          124dlnfllf |  -.1553647   .0508596   -3.05   0.002   -.2554501
335          -.05527
336          > 93
337          16dlnusepr |   .1231808   .0528785    2.33   0.020   .0191224
338          .22723
339          > 92
340          _cons |   .0014172   .0003147    4.50   0.000   .0007979
341          .00203
342          > 65
343          -----
344          --
345          106 .
346          107 .
347          108 . *5
348          109 . /*
349          > Best model
350          > reg dlnflnonfarm 13dlnflnonfarm 16dlnflnonfarm 112dlnflnonfarm
351          124dlnflnonfar
352          > m
353          >      124dlnfllf 16dlnusepr m1 m3 m6 m9
354          > */
355          110 . scalar drop _all
356
357          111 . quietly forval w=48(12)144 {
358
359

```

```

349    112 . scalar list
350        RWmaxobs144 =      144
351        RWminobs144 =      144
352        RWrmse144 =  .00396645
353        RWmaxobs132 =      132
354        RWminobs132 =      132
355        RWrmse132 =  .00390407
356        RWmaxobs120 =      120
357        RWminobs120 =      120
358        RWrmse120 =  .00388926
359        RWmaxobs108 =      108
360        RWminobs108 =      108
361        RWrmse108 =  .00388844
362        RWmaxobs96 =       96
363        RWminobs96 =       96
364        RWrmse96 =  .00403691
365        RWmaxobs84 =       84
366        RWminobs84 =       84
367        RWrmse84 =  .00406426
368        RWmaxobs72 =       72
369        RWminobs72 =       72
370        RWrmse72 =  .00411873
371        RWmaxobs60 =       60
372        RWminobs60 =       60
373        RWrmse60 =  .00431692
374        RWmaxobs48 =       48
375        RWminobs48 =       48
376        RWrmse48 =  .00460352
377
378    113 . /*
379        > RWmaxobs108 = 108
380        > RWminobs108 = 108
381        > RWrmse108 = .00388844
382        > */
383    114 .
384    115 . /*
385        > Smallest / best model
386        > reg dlnflnonfarm l12dlnflnonfarm m1 m3 m6 m9
387        > */
388    116 . scalar drop _all
389
390    117 . quietly forval w=48(12)144 {
391
392    118 . scalar list
393        RWmaxobs144 =      144
394        RWminobs144 =      144
395        RWrmse144 =  .00431666
396        RWmaxobs132 =      132

```

```

397      RWminobs132 =      132
398      RWrmse132 = .00426742
399      RWmaxobs120 =      120
400      RWminobs120 =      120
401      RWrmse120 = .00423688
402      RWmaxobs108 =      108
403      RWminobs108 =      108
404      RWrmse108 = .00428159
405      RWmaxobs96 =      96
406      RWminobs96 =      96
407      RWrmse96 = .00436091
408      RWmaxobs84 =      84
409      RWminobs84 =      84
410      RWrmse84 = .00439555
411      RWmaxobs72 =      72
412      RWminobs72 =      72
413      RWrmse72 = .00443487
414      RWmaxobs60 =      60
415      RWminobs60 =      60
416      RWrmse60 = .00453048
417      RWmaxobs48 =      48
418      RWminobs48 =      48
419      RWrmse48 = .00458215
420
421 119 . /*
422      > RWmaxobs120 = 120
423      > RWminobs120 = 120
424      > RWrmse120 = .00423688
425      >
426      > */
427 120 .
428 121 . /*
429      > Best medium length model
430      > reg dlnflnonfarm 13dlnflnonfarm 112dlnflnonfarm 124dlnflnonfarm
16dlnusepr
431      > m1 m3 m6 m9
432      > */
433 122 . scalar drop _all
434
435 123 . quietly forval w=48(12)144 {
436
437 124 . scalar list
438      RWmaxobs144 =      144
439      RWminobs144 =      144
440      RWrmse144 = .00412303
441      RWmaxobs132 =      132
442      RWminobs132 =      132
443      RWrmse132 = .00407538

```

```

444      RWmaxobs120 =      120
445      RWminobs120 =      120
446      RWrmse120 = .00406735
447      RWmaxobs108 =      108
448      RWminobs108 =      108
449      RWrmse108 = .00406403
450      RWmaxobs96 =       96
451      RWminobs96 =       96
452      RWrmse96 = .00419684
453      RWmaxobs84 =       84
454      RWminobs84 =       84
455      RWrmse84 = .00423362
456      RWmaxobs72 =       72
457      RWminobs72 =       72
458      RWrmse72 = .00429113
459      RWmaxobs60 =       60
460      RWminobs60 =       60
461      RWrmse60 = .00448591
462      RWmaxobs48 =       48
463      RWminobs48 =       48
464      RWrmse48 = .00478837
465
466      125 . /*
467      > RWmaxobs108 = 108
468      > RWminobs108 = 108
469      > RWrmse108 = .00406403
470      > */
471      126 .
472      127 . *6
473      128 . /*
474      > RWmaxobs108 = 108
475      > RWminobs108 = 108
476      > RWrmse108 = .00388844
477      > */
478      129 . scalar drop _all
479
480      130 . quietly forval w=156(12)156 {
481
482      131 . summ nobs // checking all had a full window
483
484      Variable |      Obs        Mean      Std. Dev.      Min      Max
485      -----+-----+
486      nobs |      289     135.2561    32.98122      47      156
487
488      132 . *get error info for normal interval
489      133 . summ errsq
490
491      Variable |      Obs        Mean      Std. Dev.      Min      Max

```

```

492      -----
493      errsq |      288      .000015      .0000471    1.10e-12    .0005431
494
495 134 . scalar rwmse=r(mean)^0.5
496
497 135 . scalar list rwmse
498      rwmse =  .00387308
499
500 136 . gen res=(d.lnflnonfarm-pred)
501      (73 missing values generated)
502
503 137 . _pctile res, percentile(2.5,97.5)
504
505 138 . return list
506
507      scalars:
508          r(r1) =  -.0074653569608927
509          r(r2) =  .0065394379198551
510
511 139 .
512 140 . *7
513 141 . predict temp if tin(2020m1,2020m1)
514      (option xb assumed; fitted values)
515      (360 missing values generated)
516
517 142 . replace pred=temp if tin(2020m1,2020m1)
518      (0 real changes made)
519
520 143 . drop temp
521
522 144 . gen pnonfarm=exp(l.lnflnonfarm+pred+(rwmse^2)/2)
523      (72 missing values generated)
524
525 145 . gen ubound=exp(l.lnflnonfarm+pred+1.96*rwmse+(rwmse^2)/2)
526      (72 missing values generated)
527
528 146 . gen lbound=exp(l.lnflnonfarm+pred-1.96*rwmse+(rwmse^2)/2)
529      (72 missing values generated)
530
531 147 . list month pnonfarm lbound ubound if tin(2020m1,2020m1)
532
533      -----
534      | month    pnonfarm     lbound     ubound |
535      |-----|
536 361. |      1    9001.077    8933.007    9069.667 |
537      +-----+
538

```

```

539      148 . tsline pnonfarm lbound ubound fl_nonfarm if tin(2019m1,2020m1),
540      tline(2019m12
541          > ) saving("Nonfarm_Normal", replace)
542          (file Nonfarm_Normal.gph saved)
543
544      149 . graph export "nonfarm_normal.png", replace
545          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
546 Sets
547          > /Problem Set 5/nonfarm_normal.png written in PNG format)
548
549      150 .
550      151 . *8
551      152 . *Empirical
552      153 . drop res
553
554      154 . gen res=(d.lnflnonfarm-pred)
555          (73 missing values generated)
556
557      155 . gen expres=exp(res)
558          (73 missing values generated)
559
560      156 . summ expres
561
562      Variable |       Obs        Mean     Std. Dev.      Min      Max
563      -----+-----+
564      expres |       288     .9997746     .0038735    .9777296   1.023579
565
566      157 . scalar meanexpres=r(mean)
567
568      158 . gen pnonfarme=exp(l.lnflnonfarm+pred)*meanexpres
569          (72 missing values generated)
570
571      159 . _pctile res, percentile(2.5,97.5)
572
573      scalars:
574          r(r1) =  -.0074653569608927
575          r(r2) =   .0065394379198551
576
577      161 . gen lbounde=exp(l.lnflnonfarm+pred+r(r1))*meanexpres
578          (72 missing values generated)
579
580      162 . gen ubounde=exp(l.lnflnonfarm+pred+r(r2))*meanexpres
581          (72 missing values generated)
582
583      163 . list month pnonfarme lbounde ubounde if tin(2020m1,2020m1)
584

```

```

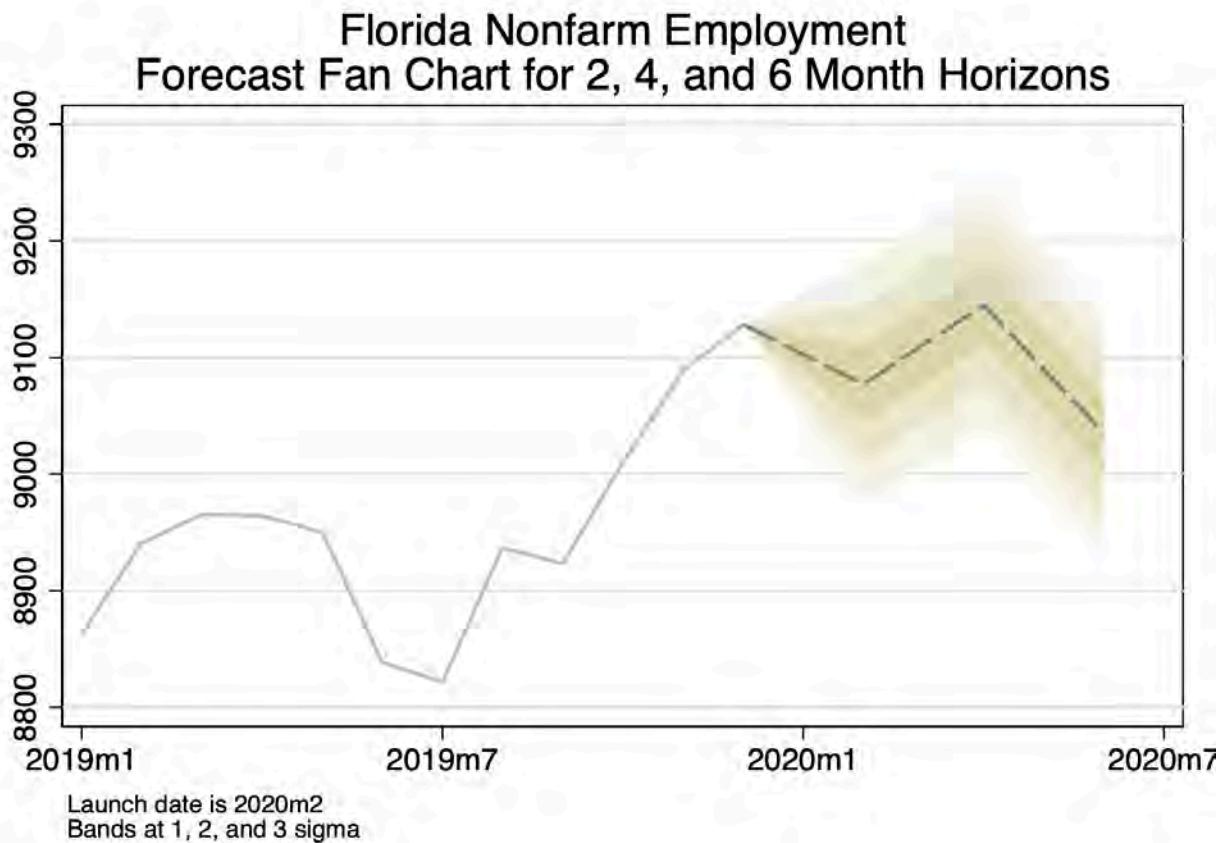
585      +-----+
586      | month    pnonfa~e      lbounde     ubounde |
587      |-----|
588      361. |       1   8998.981   8932.051   9058.022 |
589      +-----+
590
591      164 . tsline pnonfarme lbounde ubounde fl_nonfarm if tin(2019m1,2020m1), ///
592          >           tline(2019m12) saving("Nonfarm_Epirical", replace)
593          (file Nonfarm_Epirical.gph saved)
594
595      165 . graph export "nonfarm_empirical.png", replace
596          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets
597          > /Problem Set 5/nonfarm_empirical.png written in PNG format)
598
599      166 .
600      167 . *9
601      168 . tsline pnonfarm pnonfarme fl_nonfarm lbound lbounde ubound ubounde ///
602          > if tin(2019m1,2020m1), tline(2019m12) saving("Normal_vs_Empirical",
replace)
603          (file Normal_vs_Empirical.gph saved)
604
605      169 . graph export "normal_vs_empirical.png", replace
606          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets
607          > /Problem Set 5/normal_vs_empirical.png written in PNG format)
608
609      170 .

```

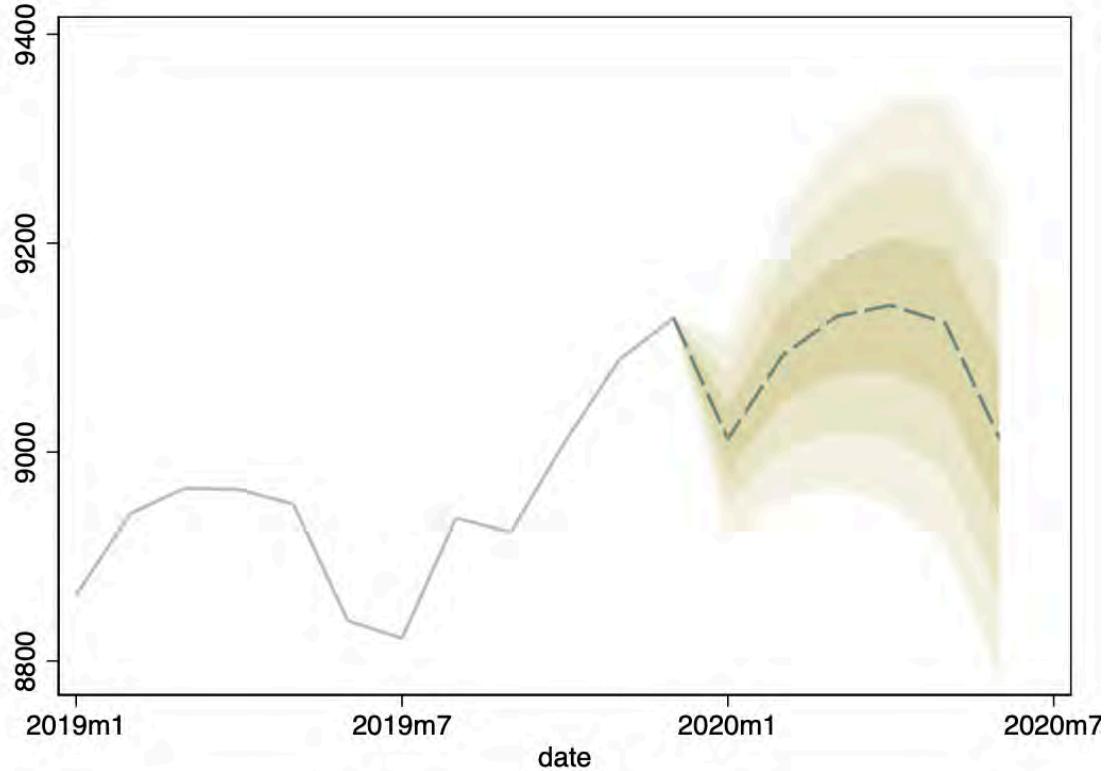
Problem Set 6

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 $0.0048 \times 6^{0.5} = 0.0118$. 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.

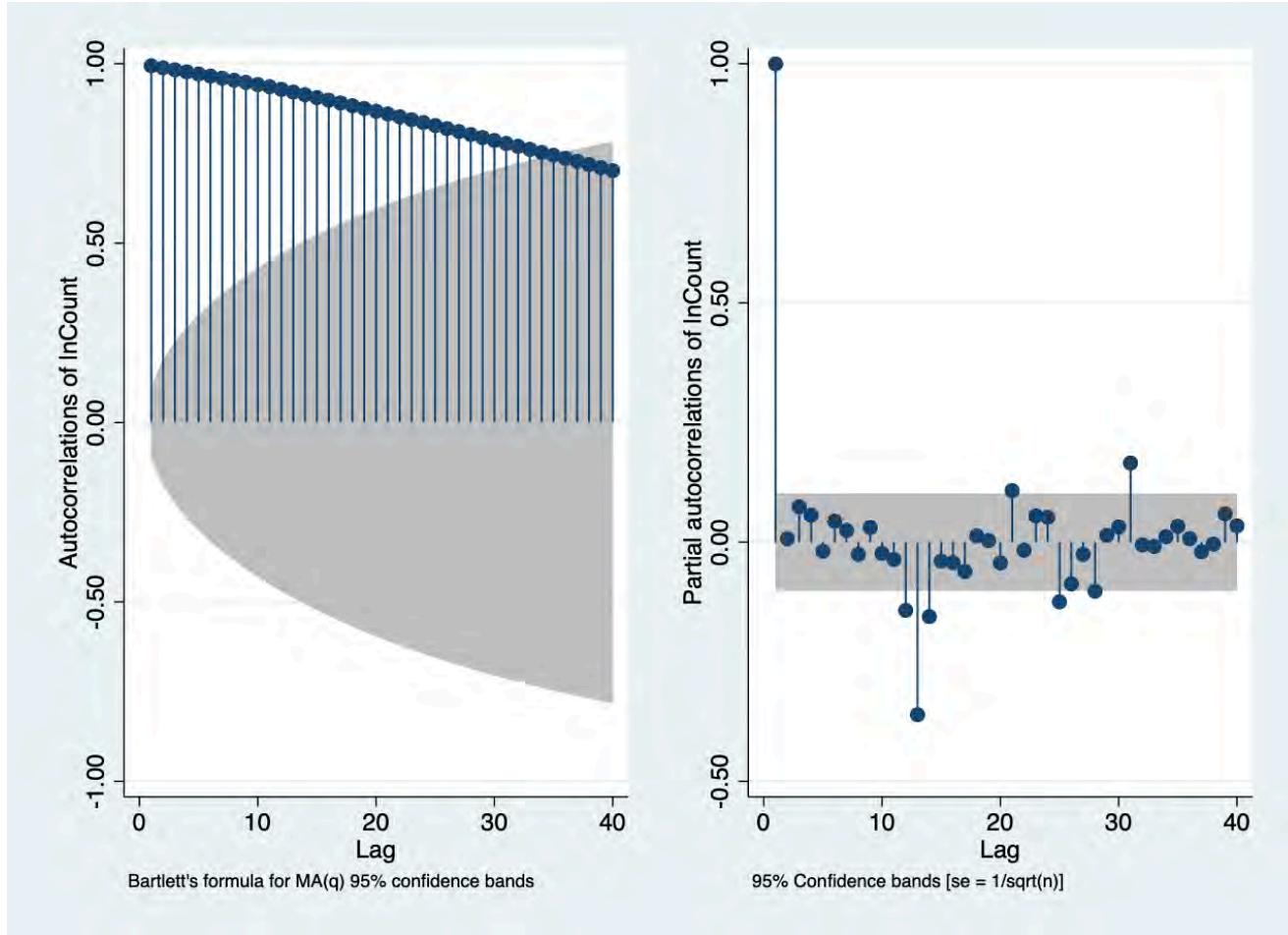
Final Project Deliverable 1

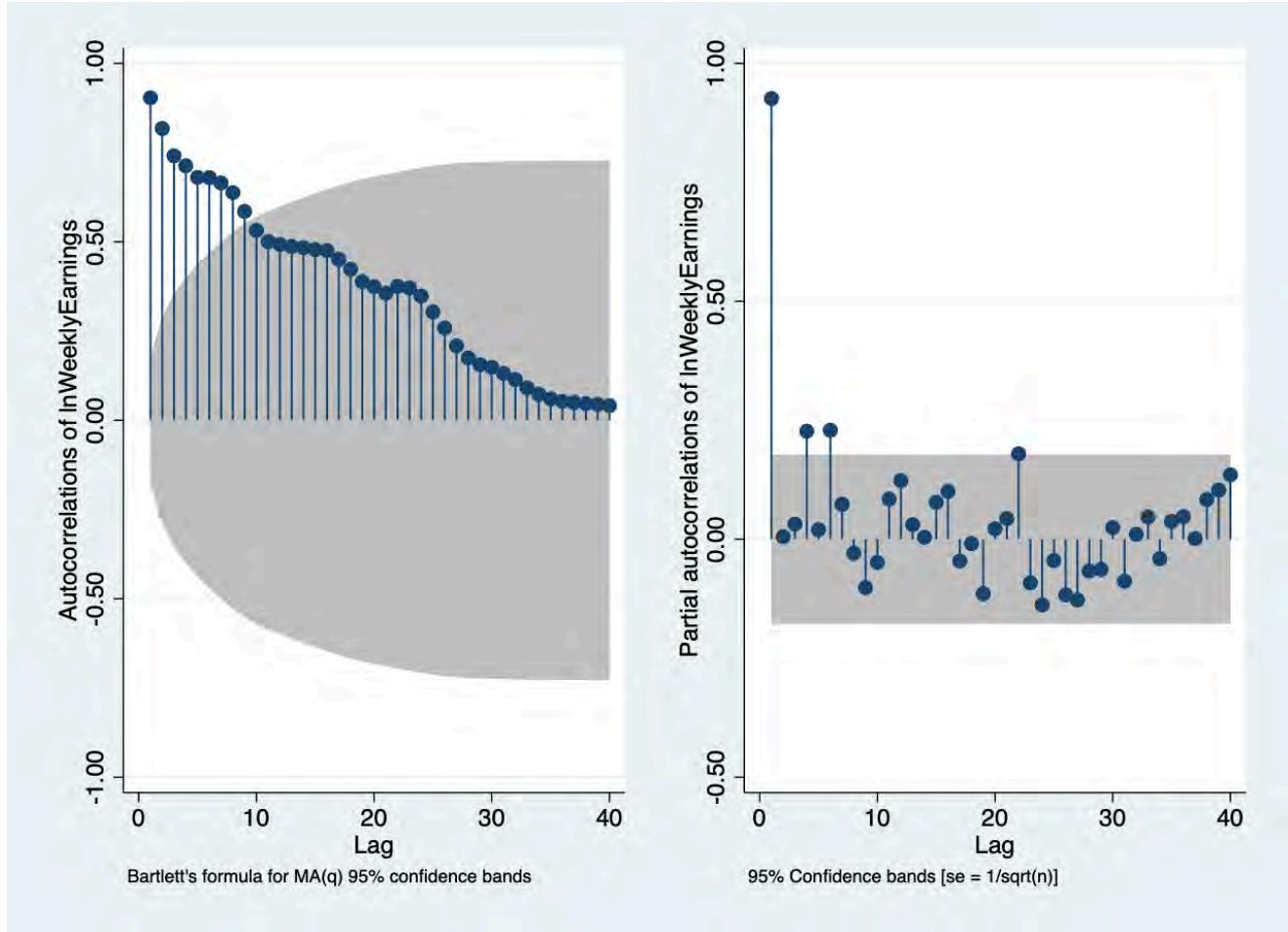
Summary statistics

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
InCount	374	2.5174	.5398403	1.667707	3.332205
InWeekHours	122	3.602049	.1041722	3.342862	3.824284
InHourlyEa~s	122	2.969891	.148565	2.708717	3.202746
InWeeklyEa~s	122	6.57194	.1198394	6.222159	6.820126
InServiceC~t	374	2.16967	.5975865	1.360977	3.12676

AC and PAC





We should difference the data

Final Project Deliverable 2

Number of Employees

```
1 reg d.lnCount 1(12,24,36,48)d.lnCount // .01637
2 scalar drop _all
3 quietly forval w=12(12)144 {
4 gen pred=.
5 gen nobs=.
6 forval t=421/733 {
7 gen wstart=`t'-'w'
8 gen wend=`t'-1
9 reg dlnCount 112dlnCount 124dlnCount 136dlnCount 148dlnCount ///
10 if Date>=wstart & Date<=wend
11 replace nobs=e(N) if Date==`t'
12 predict ptemp
13 replace pred=ptemp if Date==`t'
14 drop ptemp wstart wend
15 }
16 gen errsq=(pred-d.lnCount)^2
17 summ errsq
18 scalar RWrmse`w'=r(mean)^.5
19 summ nobs
20 scalar RWminobs`w'=r(min)
21 scalar RWmaxobs`w'=r(max)
22 drop errsq pred nobs
23 }
24 scalar list
25 /*
26 RWmaxobs132 = 132
27 RWminobs132 = 12
28 RWrmse132 = .0172128
29 */
```

```
1 reg d.lnCount 1(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
2 scalar drop _all
3 quietly forval w=12(12)144 {
4 gen pred=.
5 gen nobs=.
6 forval t=409/733 {
7 gen wstart=`t'-'w'
8 gen wend=`t'-1
9 reg dlnCount 112dlnCount 136dlnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
10 if Date>=wstart & Date<=wend
11 replace nobs=e(N) if Date==`t'
12 predict ptemp
```

```

13    replace pred=ptemp if Date==`t'
14    drop ptemp wstart wend
15  }
16 gen errsq=(pred-d.lnCount)^2
17 summ errsq
18 scalar RWrmse`w'=r(mean)^.5
19 summ nobs
20 scalar RWminobs`w'=r(min)
21 scalar RWmaxobs`w'=r(max)
22 drop errsq pred nobs
23 }
24 scalar list
25 /*
26 RWmaxobs144 =          144
27 RWminobs144 =          12
28 RWrmse144 =   .01777071
29 */

```

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

```

1 reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings 17d.lnWeeklyEarnings
2 scalar drop _all
3 quietly forval w=12(12)84 {
4 gen pred=.
5 gen nobs=.
6 forval t=622/733 {
7 gen wstart=`t'-'`w'
8 gen wend=`t'-1
9 reg dlnWeeklyEarnings 13dlnWeeklyEarnings 15dlnWeeklyEarnings 17dlnWeeklyEarnings
///
10 if Date>=wstart & Date<=wend
11 replace nobs=e(N) if Date==`t'
12 predict ptemp
13 replace pred=ptemp if Date==`t'
14 drop ptemp wstart wend
15 }
16 gen errsq=(pred-d.lnWeeklyEarnings)^2
17 summ errsq
18 scalar RWrmse`w'=r(mean)^.5
19 summ nobs
20 scalar RWminobs`w'=r(min)
21 scalar RWmaxobs`w'=r(max)
22 drop errsq pred nobs
23 }
24 scalar list
25 /*
26 RWmaxobs84 =          84

```

```
27 | RWminobs84 =          2
28 | RWrmse84 =   .05250414
29 | */
```

Code

```
1  clear
2  set more off
3
4  cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Final
Project"
5  *log using "Final Project.smcl", replace
6  import delimited "TS2020_Final_Project_txt2/TS2020_Final_Project_Monthly.txt"
7  rename smu12455400500000001 Count
8  rename smu12455400500000002 WeekHours
9  rename smu12455400500000003 HourlyEarnings
10 rename smu12455400500000011 WeeklyEarnings
11 rename smu12455400800000001 ServiceCount
12
13
14 label variable Count "Count"
15 label variable WeekHours "WeekHours"
16 label variable HourlyEarnings "HourlyEarnings"
17 label variable WeeklyEarnings "WeeklyEarnings"
18 label variable ServiceCount "ServiceCount"
19
20
21 gen datec=date(date, "YMD")
22 gen Date=mofd(datec)
23 gen month=month(datec)
24 format Date %tm
25 tsset Date
26
27 gen lnCount = ln(Count)
28 gen lnWeekHours = ln(WeekHours)
29 gen lnHourlyEarnings = ln(HourlyEarnings)
30 gen lnWeeklyEarnings = ln(WeeklyEarnings)
31 gen lnServiceCount = ln(ServiceCount)
32
33 gen m1=0
34 replace m1=1 if month==1
35 gen m2=0
36 replace m2=1 if month==2
37 gen m3=0
38 replace m3=1 if month==3
39 gen m4=0
40 replace m4=1 if month==4
41 gen m5=0
```

```
42 replace m5=1 if month==5
43 gen m6=0
44 replace m6=1 if month==6
45 gen m7=0
46 replace m7=1 if month==7
47 gen m8=0
48 replace m8=1 if month==8
49 gen m9=0
50 replace m9=1 if month==9
51 gen m10=0
52 replace m10=1 if month==10
53 gen m11=0
54 replace m11=1 if month==11
55 gen m12=0
56 replace m12=1 if month==12
57
58 gen dlnCount=d.lnCount
59 gen 11dlnCount=11d.lnCount
60 gen 12dlnCount=12d.lnCount
61 gen 13dlnCount=13d.lnCount
62 gen 14dlnCount=14d.lnCount
63 gen 15dlnCount=15d.lnCount
64 gen 16dlnCount=16d.lnCount
65 gen 17dlnCount=17d.lnCount
66 gen 18dlnCount=18d.lnCount
67 gen 19dlnCount=19d.lnCount
68 gen 110dlnCount=110d.lnCount
69 gen 111dlnCount=111d.lnCount
70 gen 112dlnCount=112d.lnCount
71 gen 124dlnCount=124d.lnCount
72 gen 136dlnCount=136d.lnCount
73 gen 148dlnCount=148d.lnCount
74
75 gen dlnWeekHours=d.lnWeekHours
76 gen 11dlnWeekHours=11d.lnWeekHours
77 gen 12dlnWeekHours=12d.lnWeekHours
78 gen 13dlnWeekHours=13d.lnWeekHours
79 gen 14dlnWeekHours=14d.lnWeekHours
80 gen 15dlnWeekHours=15d.lnWeekHours
81 gen 16dlnWeekHours=16d.lnWeekHours
82 gen 17dlnWeekHours=17d.lnWeekHours
83 gen 18dlnWeekHours=18d.lnWeekHours
84 gen 19dlnWeekHours=19d.lnWeekHours
85 gen 110dlnWeekHours=110d.lnWeekHours
86 gen 111dlnWeekHours=111d.lnWeekHours
87 gen 112dlnWeekHours=112d.lnWeekHours
88 gen 124dlnWeekHours=124d.lnWeekHours
89 gen 136dlnWeekHours=136d.lnWeekHours
90 gen 148dlnWeekHours=148d.lnWeekHours
91
92 gen dlnHourlyEarnings=d.lnHourlyEarnings
```

```
93 gen 11dlnHourlyEarnings=11d.lnHourlyEarnings
94 gen 12dlnHourlyEarnings=12d.lnHourlyEarnings
95 gen 13dlnHourlyEarnings=13d.lnHourlyEarnings
96 gen 14dlnHourlyEarnings=14d.lnHourlyEarnings
97 gen 15dlnHourlyEarnings=15d.lnHourlyEarnings
98 gen 16dlnHourlyEarnings=16d.lnHourlyEarnings
99 gen 17dlnHourlyEarnings=17d.lnHourlyEarnings
100 gen 18dlnHourlyEarnings=18d.lnHourlyEarnings
101 gen 19dlnHourlyEarnings=19d.lnHourlyEarnings
102 gen 110dlnHourlyEarnings=110d.lnHourlyEarnings
103 gen 111dlnHourlyEarnings=111d.lnHourlyEarnings
104 gen 112dlnHourlyEarnings=112d.lnHourlyEarnings
105 gen 124dlnHourlyEarnings=124d.lnHourlyEarnings
106 gen 136dlnHourlyEarnings=136d.lnHourlyEarnings
107 gen 148dlnHourlyEarnings=148d.lnHourlyEarnings
108
109 gen dlnWeeklyEarnings=d.lnWeeklyEarnings
110 gen 11dlnWeeklyEarnings=11d.lnWeeklyEarnings
111 gen 12dlnWeeklyEarnings=12d.lnWeeklyEarnings
112 gen 13dlnWeeklyEarnings=13d.lnWeeklyEarnings
113 gen 14dlnWeeklyEarnings=14d.lnWeeklyEarnings
114 gen 15dlnWeeklyEarnings=15d.lnWeeklyEarnings
115 gen 16dlnWeeklyEarnings=16d.lnWeeklyEarnings
116 gen 17dlnWeeklyEarnings=17d.lnWeeklyEarnings
117 gen 18dlnWeeklyEarnings=18d.lnWeeklyEarnings
118 gen 19dlnWeeklyEarnings=19d.lnWeeklyEarnings
119 gen 110dlnWeeklyEarnings=110d.lnWeeklyEarnings
120 gen 111dlnWeeklyEarnings=111d.lnWeeklyEarnings
121 gen 112dlnWeeklyEarnings=112d.lnWeeklyEarnings
122 gen 124dlnWeeklyEarnings=124d.lnWeeklyEarnings
123 gen 136dlnWeeklyEarnings=136d.lnWeeklyEarnings
124 gen 148dlnWeeklyEarnings=148d.lnWeeklyEarnings
125
126 gen dlnServiceCount=d.lnServiceCount
127 gen 11dlnServiceCount=11d.lnServiceCount
128 gen 12dlnServiceCount=12d.lnServiceCount
129 gen 13dlnServiceCount=13d.lnServiceCount
130 gen 14dlnServiceCount=14d.lnServiceCount
131 gen 15dlnServiceCount=15d.lnServiceCount
132 gen 16dlnServiceCount=16d.lnServiceCount
133 gen 17dlnServiceCount=17d.lnServiceCount
134 gen 18dlnServiceCount=18d.lnServiceCount
135 gen 19dlnServiceCount=19d.lnServiceCount
136 gen 110dlnServiceCount=110d.lnServiceCount
137 gen 111dlnServiceCount=111d.lnServiceCount
138 gen 112dlnServiceCount=112d.lnServiceCount
139 gen 124dlnServiceCount=124d.lnServiceCount
140 gen 136dlnServiceCount=136d.lnServiceCount
141 gen 148dlnServiceCount=148d.lnServiceCount
142
143 /*
```

```

144 The project is to forecast the March non-seasonally adjusted estimates of average
145 weekly earnings and total employment for private employers (total private) for a
146 Florida MSA of your choice and write up a professional report on your forecast.
147 */
148 /* Count and WeeklyEarnings */
149
150 summ Count WeekHours HourlyEarnings WeeklyEarnings ServiceCount
151 summ lnCount lnWeekHours lnHourlyEarnings lnWeeklyEarnings lnServiceCount
152
153 ac lnCount, saving(lnCount_ac, replace)
154 pac lnCount, saving(lnCount_pac, replace)
155 graph combine lnCount_ac.gph lnCount_pac.gph, saving(lnCount_ac_pac, replace)
156 graph export "lnCount_ac_pac.png", replace
157 ** Probably need to difference
158
159 ac lnWeeklyEarnings, saving(lnWeeklyEarnings_ac, replace)
160 pac lnWeeklyEarnings, saving(lnWeeklyEarnings_pac, replace)
161 graph combine lnWeeklyEarnings_ac.gph lnWeeklyEarnings_pac.gph,
162 saving(lnWeeklyEarnings_ac_pac, replace)
163 graph export "lnWeeklyEarnings_ac_pac.png", replace
164 ** Probably need to differencen b
165
166 *starter models for count
167 *I used a pair plot to examine the rise and fall of variables with respect to each
168 other
169 reg d.lnCount l(12,24,36,48)d.lnCount // .01637
170 scalar drop _all
171 quietly forval w=12(12)144 {
172 gen pred=.
173 gen nobs=.
174 forval t=421/733 {
175 gen wstart=`t'-`w'
176 gen wend=`t'-1
177 reg dlnCount 112dlnCount 124dlnCount 136dlnCount 148dlnCount ///
178 if Date>=wstart & Date<=wend
179 replace nobs=e(N) if Date==`t'
180 predict ptemp
181 replace pred=ptemp if Date==`t'
182 drop ptemp wstart wend
183 }
184 gen errsq=(pred-d.lnCount)^2
185 summ errsq
186 scalar RWrmse`w'=r(mean)^.5
187 scalar nobs
188 scalar RWminobs`w'=r(min)
189 scalar RWmaxobs`w'=r(max)
190 drop errsq pred nobs
191 }
192 scalar list
193 /*
194 RWmaxobs132 =          132

```

```

191 RWminobs132 =          12
192 RWrmse132 =    .0172128
193 */
194
195 reg d.lnCount 1(5,12,24,36,48)d.lnCount 1(5)d.lnWeekHours m5 // .01711
196 scalar drop _all
197 quietly forval w=12(12)84 {
198     gen pred=.
199     gen nobs=.
200     forval t=641/733 {
201         gen wstart=`t'-'`w'
202         gen wend=`t'-1
203         reg dlnCount 15dlnCount 112dlnCount 124dlnCount 136dlnCount 148dlnCount
204             15dlnWeekHours m5 ///
205             if Date>=wstart & Date<=wend
206             replace nobs=e(N) if Date==`t'
207             predict ptemp
208             replace pred=ptemp if Date==`t'
209             drop ptemp wstart wend
210     }
211     gen errsq=(pred-d.lnCount)^2
212     summ errsq
213     scalar RWrmse`w'=r(mean)^.5
214     summ nobs
215     scalar RWminobs`w'=r(min)
216     scalar RWmaxobs`w'=r(max)
217     drop errsq pred nobs
218 }
219 scalar list
220 /*
221 RWmaxobs84 =          84
222 RWminobs84 =          23
223 RWrmse84 =    .01950911
224 */
225 /*
226 gsreg dlnCount 11dlnCount 12dlnCount 13dlnCount 14dlnCount 15dlnCount 16dlnCount ///
227     17dlnCount 18dlnCount 19dlnCount 110dlnCount 111dlnCount 112dlnCount ///
228     124dlnCount 136dlnCount 148dlnCount ///
229     if tin(1990m1,2021m1), ///
230     ncomb(1,12) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12) ///
231     samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnCount) replace
232 */
233
234 *gsreg suggestions
235 reg d.lnCount 112d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
236 scalar drop _all
237 quietly forval w=12(12)144 {
238     gen pred=.
239     gen nobs=.
240     forval t=385/733 {

```

```

241 gen wstart=`t'-'w'
242 gen wend=`t'-1
243 reg dlnCount 112dlnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
244     if Date>=wstart & Date<=wend
245 replace nobs=e(N) if Date==`t'
246 predict ptemp
247 replace pred=ptemp if Date==`t'
248 drop ptemp wstart wend
249 }
250 gen errsq=(pred-d.lnCount)^2
251 summ errsq
252 scalar RWrmse`w'=r(mean)^.5
253 summ nobs
254 scalar RWminobs`w'=r(min)
255 scalar RWmaxobs`w'=r(max)
256 drop errsq pred nobs
257 }
258 scalar list
259 /*
260 RWmaxobs144 =      144
261 RWminobs144 =      12
262 RWrmse144 = .01824906
263 */
264
265 reg d.lnCount 1(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
266 scalar drop _all
267 quietly forval w=12(12)144 {
268     gen pred=.
269     gen nobs=.
270     forval t=409/733 {
271         gen wstart=`t'-'w'
272         gen wend=`t'-1
273         reg dlnCount 112dlnCount 136dlnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
274             if Date>=wstart & Date<=wend
275         replace nobs=e(N) if Date==`t'
276         predict ptemp
277         replace pred=ptemp if Date==`t'
278         drop ptemp wstart wend
279     }
280     gen errsq=(pred-d.lnCount)^2
281     summ errsq
282     scalar RWrmse`w'=r(mean)^.5
283     summ nobs
284     scalar RWminobs`w'=r(min)
285     scalar RWmaxobs`w'=r(max)
286     drop errsq pred nobs
287 }
288 scalar list
289 /*
290 RWmaxobs144 =      144
291 RWminobs144 =      12

```

```

292 RWrmse144 = .01777071
293 /*
294 /*
295 gsreg dlnCount 11dlnCount 12dlnCount 13dlnCount 14dlnCount 15dlnCount 16dlnCount ///
296 17dlnCount 18dlnCount 19dlnCount 110dlnCount 111dlnCount 112dlnCount ///
297 124dlnCount 136dlnCount ///
298 11dlnWeekHours 12dlnWeekHours 13dlnWeekHours 14dlnWeekHours 15dlnWeekHours
299 16dlnWeekHours ///
300 17dlnWeekHours 18dlnWeekHours 19dlnWeekHours 110dlnWeekHours 111dlnWeekHours
301 112dlnWeekHours ///
302 124dlnWeekHours 136dlnWeekHours ///
303 11dlnHourlyEarnings 12dlnHourlyEarnings 13dlnHourlyEarnings 14dlnHourlyEarnings
304 ///
305 15dlnHourlyEarnings 16dlnHourlyEarnings ///
306 17dlnHourlyEarnings 18dlnHourlyEarnings 19dlnHourlyEarnings 110dlnHourlyEarnings
307 ///
308 111dlnHourlyEarnings 112dlnHourlyEarnings ///
309 124dlnHourlyEarnings 136dlnHourlyEarnings ///
310 11dlnWeeklyEarnings 12dlnWeeklyEarnings 13dlnWeeklyEarnings 14dlnWeeklyEarnings
311 ///
312 15dlnWeeklyEarnings 16dlnWeeklyEarnings ///
313 17dlnWeeklyEarnings 18dlnWeeklyEarnings 19dlnWeeklyEarnings 110dlnWeeklyEarnings
314 ///
315 */
316
317 reg d.lnCount 14d.lnWeekHours 19d.lnWeekHours 18d.lnHourlyEarnings m1 m2 m3 m4 m5 m6
m7 m8 m9 m10 m11
318 scalar drop _all
319 quietly forval w=12(12)84 {
320 gen pred=.
321 gen nobs=.
322 forval t=624/733 {
323 gen wstart=`t'-`w'
324 gen wend=`t'-1
325 reg dlnCount 14dlnWeekHours 19dlnWeekHours 18dlnHourlyEarnings m1 m2 m3 m4 m5 m6
m7 m8 m9 m10 m11 ///
326 if Date>=wstart & Date<=wend
327 replace nobs=e(N) if Date==`t'
328 predict ptemp
329 replace pred=ptemp if Date==`t'
330 drop ptemp wstart wend
331 }
332 gen errsq=(pred-d.lnCount)^2
333 summ errsq
334 scalar RWrmse`w'=r(mean)^.5

```

```

335 summ nobs
336 scalar RWminobs`w'=r(min)
337 scalar RWmaxobs`w'=r(max)
338 drop errsq pred nobs
339 }
340 scalar list
341 /*
342 RWmaxobs12 = 12
343 RWminobs12 = 2
344 RWrmse12 = .0176238
345 */
346
347
348 *starter models for weekly earnings
349 reg d.lnWeeklyEarnings l1d.lnWeekHours l1d.lnHourlyEarnings
350 scalar drop _all
351 quietly forval w=12(12)84 {
352 gen pred=.
353 gen nobs=.
354 forval t=616/733 {
355 gen wstart=`t'-'`w'
356 gen wend=`t'-1
357 reg dlnWeeklyEarnings l1dlnWeekHours l1dlnHourlyEarnings ///
358 if Date>=wstart & Date<=wend
359 replace nobs=e(N) if Date==`t'
360 predict ptemp
361 replace pred=ptemp if Date==`t'
362 drop ptemp wstart wend
363 }
364 gen errsq=(pred-d.lnWeeklyEarnings)^2
365 summ errsq
366 scalar RWrmse`w'=r(mean)^.5
367 summ nobs
368 scalar RWminobs`w'=r(min)
369 scalar RWmaxobs`w'=r(max)
370 drop errsq pred nobs
371 }
372 scalar list
373 /*
374 RWmaxobs60 = 60
375 RWminobs60 = 2
376 RWrmse60 = .06145693
377 */
378
379 /*
380 gsreg dlnWeeklyEarnings l1dlnWeeklyEarnings l2dlnWeeklyEarnings l3dlnWeeklyEarnings
381 ///
382 14dlnWeeklyEarnings 15dlnWeeklyEarnings 16dlnWeeklyEarnings ///
383 17dlnWeeklyEarnings 18dlnWeeklyEarnings 19dlnWeeklyEarnings 110dlnWeeklyEarnings
384 ///
385 111dlnWeeklyEarnings 112dlnWeeklyEarnings ///

```

```

384 124dlnWeeklyEarnings 136dlnWeeklyEarnings ///
385 if tin(2011m1,2021m1), ///
386 ncomb(1,12) aic outsamp(24) ///
387 samesamp nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnWeeklyEarnings)
388 replace
389 */
390
390 reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings m1 m2 m3 m4 m5 m6
m7 m8 m9 m10 m11
391 scalar drop _all
392 quietly forval w=12(12)84 {
393 gen pred=.
394 gen nobs=.
395 forval t=620/733 {
396 gen wstart=`t'-`w'
397 gen wend=`t'-1
398 reg dlnWeeklyEarnings 13dlnWeeklyEarnings 15dlnWeeklyEarnings m1 m2 m3 m4 m5 m6 m7
m8 m9 m10 m11 ///
399 if Date>=wstart & Date<=wend
400 replace nobs=e(N) if Date==`t'
401 predict ptemp
402 replace pred=ptemp if Date==`t'
403 drop ptemp wstart wend
404 }
405 gen errsq=(pred-d.lnWeeklyEarnings)^2
406 summ errsq
407 scalar RWrmse`w'=r(mean)^.5
408 summ nobs
409 scalar RWminobs`w'=r(min)
410 scalar RWmaxobs`w'=r(max)
411 drop errsq pred nobs
412 }
413 scalar list
414 /*
415 RWmaxobs84 = 84
416 RWminobs84 = 2
417 RWrmse84 = .06004448
418 */
419
420 reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings
17d.lnWeeklyEarnings
421 scalar drop _all
422 quietly forval w=12(12)84 {
423 gen pred=.
424 gen nobs=.
425 forval t=622/733 {
426 gen wstart=`t'-`w'
427 gen wend=`t'-1
428 reg dlnWeeklyEarnings 13dlnWeeklyEarnings 15dlnWeeklyEarnings 17dlnWeeklyEarnings
///
429 if Date>=wstart & Date<=wend

```

```

430 replace nobs=e(N) if Date==`t'
431 predict ptemp
432 replace pred=ptemp if Date==`t'
433 drop ptemp wstart wend
434 }
435 gen errsq=(pred-d.lnWeeklyEarnings)^2
436 summ errsq
437 scalar RWrmse`w'=r(mean)^.5
438 summ nobs
439 scalar RWminobs`w'=r(min)
440 scalar RWmaxobs`w'=r(max)
441 drop errsq pred nobs
442 }
443 scalar list
444 /*
445 RWmaxobs84 = 84
446 RWminobs84 = 2
447 RWrmse84 = .05250414
448 */
449 /*
450 gsreg dlnCount 11dlnCount 12dlnCount 13dlnCount 14dlnCount 15dlnCount 16dlnCount ///
451 17dlnCount 18dlnCount 19dlnCount 110dlnCount 111dlnCount 112dlnCount 124dlnCount
///
453 11dlnWeekHours 12dlnWeekHours 13dlnWeekHours 14dlnWeekHours 15dlnWeekHours
16dlnWeekHours ///
454 17dlnWeekHours 18dlnWeekHours 19dlnWeekHours 110dlnWeekHours 111dlnWeekHours
112dlnWeekHours 124dlnWeekHours ///
455 11dlnHourlyEarnings 12dlnHourlyEarnings 13dlnHourlyEarnings 14dlnHourlyEarnings
///
456 15dlnHourlyEarnings 16dlnHourlyEarnings ///
457 17dlnHourlyEarnings 18dlnHourlyEarnings 19dlnHourlyEarnings 110dlnHourlyEarnings
///
458 111dlnHourlyEarnings 112dlnHourlyEarnings 124dlnHourlyEarnings ///
459 11dlnWeeklyEarnings 12dlnWeeklyEarnings 13dlnWeeklyEarnings 14dlnWeeklyEarnings
///
460 15dlnWeeklyEarnings 16dlnWeeklyEarnings ///
461 17dlnWeeklyEarnings 18dlnWeeklyEarnings 19dlnWeeklyEarnings 110dlnWeeklyEarnings
///
462 111dlnWeeklyEarnings 112dlnWeeklyEarnings 124dlnWeeklyEarnings if
tin(2011m1,2021m1), ///
463 ncomb(1,4) aic outsamp(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11) ///
464 samesample nindex( -1 aic -1 bic -1 rmse_out)
results(gsreg_dlnWeeklyEarnings_Full) replace
465 */
466
467 *log close

```

Log File

```
1 _____(R) _____
2 _____/____/____/____/____/
3 ____/____/____/____/____/
4 Statistics/Data analysis
5
6 -----
7 --
8         name: <unnamed>
9         log: /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
10 Series/Probl
11     > em Sets/Final Project/Final Project.smcl
12         log type: smcl
13         opened on: 8 Apr 2021, 17:47:08
14
15
16     1 . import delimited
17         "TS2020_Final_Project_txt2/TS2020_Final_Project_Monthly.txt"
18             (6 vars, 374 obs)
19
20     2 . rename smu12455400500000001 Count
21
22     3 . rename smu12455400500000002 WeekHours
23
24     4 . rename smu12455400500000003 HourlyEarnings
25
26     5 . rename smu12455400500000011 WeeklyEarnings
27
28     6 . rename smu12455400800000001 ServiceCount
29
30     7 .
31     8 .
32     9 . label variable Count "Count"
33
34     10 . label variable WeekHours "WeekHours"
35
36     11 . label variable HourlyEarnings "HourlyEarnings"
37
38     12 . label variable WeeklyEarnings "WeeklyEarnings"
39
40     13 . label variable ServiceCount "ServiceCount"
41
42     14 .
43     15 .
44     16 . gen datec=date(date, "YMD")
45
46     17 . gen Date=mofd(datec)
47
48     18 . gen month=month(datec)
```

```
45  
46     19 . format Date %tm  
47  
48     20 . tsset Date  
        time variable: Date, 1990m1 to 2021m2  
        delta: 1 month  
51  
52     21 .  
53     22 . gen lnCount = ln(Count)  
54  
55     23 . gen lnWeekHours = ln(WeekHours)  
        (252 missing values generated)  
57  
58     24 . gen lnHourlyEarnings = ln(HourlyEarnings)  
        (252 missing values generated)  
60  
61     25 . gen lnWeeklyEarnings = ln(WeeklyEarnings)  
        (252 missing values generated)  
63  
64     26 . gen lnServiceCount = ln(ServiceCount)  
65  
66     27 .  
67     28 . gen m1=0  
68  
69     29 . replace m1=1 if month==1  
        (32 real changes made)  
71  
72     30 . gen m2=0  
73  
74     31 . replace m2=1 if month==2  
        (32 real changes made)  
76  
77     32 . gen m3=0  
78  
79     33 . replace m3=1 if month==3  
        (31 real changes made)  
81  
82     34 . gen m4=0  
83  
84     35 . replace m4=1 if month==4  
        (31 real changes made)  
86  
87     36 . gen m5=0  
88  
89     37 . replace m5=1 if month==5  
        (31 real changes made)  
91  
92     38 . gen m6=0  
93  
94     39 . replace m6=1 if month==6  
        (31 real changes made)
```

```
96  
97     40 . gen m7=0  
98  
99     41 . replace m7=1 if month==7  
100      (31 real changes made)  
101  
102     42 . gen m8=0  
103  
104     43 . replace m8=1 if month==8  
105      (31 real changes made)  
106  
107     44 . gen m9=0  
108  
109     45 . replace m9=1 if month==9  
110      (31 real changes made)  
111  
112     46 . gen m10=0  
113  
114     47 . replace m10=1 if month==10  
115      (31 real changes made)  
116  
117     48 . gen m11=0  
118  
119     49 . replace m11=1 if month==11  
120      (31 real changes made)  
121  
122     50 . gen m12=0  
123  
124     51 . replace m12=1 if month==12  
125      (31 real changes made)  
126  
127     52 .  
128     53 . gen dlnCount=d.lnCount  
129      (1 missing value generated)  
130  
131     54 . gen l1dlnCount=l1d.lnCount  
132      (2 missing values generated)  
133  
134     55 . gen l2dlnCount=l2d.lnCount  
135      (3 missing values generated)  
136  
137     56 . gen l3dlnCount=l3d.lnCount  
138      (4 missing values generated)  
139  
140     57 . gen l4dlnCount=l4d.lnCount  
141      (5 missing values generated)  
142  
143     58 . gen l5dlnCount=l5d.lnCount  
144      (6 missing values generated)  
145  
146     59 . gen l6dlnCount=l6d.lnCount
```

```
147      (7 missing values generated)
148
149      60 . gen 17dlnCount=17d.lnCount
150      (8 missing values generated)
151
152      61 . gen 18dlnCount=18d.lnCount
153      (9 missing values generated)
154
155      62 . gen 19dlnCount=19d.lnCount
156      (10 missing values generated)
157
158      63 . gen 110dlnCount=110d.lnCount
159      (11 missing values generated)
160
161      64 . gen 111dlnCount=111d.lnCount
162      (12 missing values generated)
163
164      65 . gen 112dlnCount=112d.lnCount
165      (13 missing values generated)
166
167      66 . gen 124dlnCount=124d.lnCount
168      (25 missing values generated)
169
170      67 . gen 136dlnCount=136d.lnCount
171      (37 missing values generated)
172
173      68 . gen 148dlnCount=148d.lnCount
174      (49 missing values generated)
175
176      69 .
177      70 . gen dlnWeekHours=d.lnWeekHours
178      (253 missing values generated)
179
180      71 . gen 11dlnWeekHours=11d.lnWeekHours
181      (254 missing values generated)
182
183      72 . gen 12dlnWeekHours=12d.lnWeekHours
184      (255 missing values generated)
185
186      73 . gen 13dlnWeekHours=13d.lnWeekHours
187      (256 missing values generated)
188
189      74 . gen 14dlnWeekHours=14d.lnWeekHours
190      (257 missing values generated)
191
192      75 . gen 15dlnWeekHours=15d.lnWeekHours
193      (258 missing values generated)
194
195      76 . gen 16dlnWeekHours=16d.lnWeekHours
196      (259 missing values generated)
197
```

```
198    77 . gen 17dlnWeekHours=17d.lnWeekHours  
199      (260 missing values generated)  
200  
201    78 . gen 18dlnWeekHours=18d.lnWeekHours  
202      (261 missing values generated)  
203  
204    79 . gen 19dlnWeekHours=19d.lnWeekHours  
205      (262 missing values generated)  
206  
207    80 . gen 110dlnWeekHours=110d.lnWeekHours  
208      (263 missing values generated)  
209  
210    81 . gen 111dlnWeekHours=111d.lnWeekHours  
211      (264 missing values generated)  
212  
213    82 . gen 112dlnWeekHours=112d.lnWeekHours  
214      (265 missing values generated)  
215  
216    83 . gen 124dlnWeekHours=124d.lnWeekHours  
217      (277 missing values generated)  
218  
219    84 . gen 136dlnWeekHours=136d.lnWeekHours  
220      (289 missing values generated)  
221  
222    85 . gen 148dlnWeekHours=148d.lnWeekHours  
223      (301 missing values generated)  
224  
225    86 .  
226    87 . gen dlnHourlyEarnings=d.lnHourlyEarnings  
227      (253 missing values generated)  
228  
229    88 . gen 11dlnHourlyEarnings=11d.lnHourlyEarnings  
230      (254 missing values generated)  
231  
232    89 . gen 12dlnHourlyEarnings=12d.lnHourlyEarnings  
233      (255 missing values generated)  
234  
235    90 . gen 13dlnHourlyEarnings=13d.lnHourlyEarnings  
236      (256 missing values generated)  
237  
238    91 . gen 14dlnHourlyEarnings=14d.lnHourlyEarnings  
239      (257 missing values generated)  
240  
241    92 . gen 15dlnHourlyEarnings=15d.lnHourlyEarnings  
242      (258 missing values generated)  
243  
244    93 . gen 16dlnHourlyEarnings=16d.lnHourlyEarnings  
245      (259 missing values generated)  
246  
247    94 . gen 17dlnHourlyEarnings=17d.lnHourlyEarnings  
248      (260 missing values generated)
```

```
249  
250     95 . gen 18dlnHourlyEarnings=18d.lnHourlyEarnings  
251         (261 missing values generated)  
252  
253     96 . gen 19dlnHourlyEarnings=19d.lnHourlyEarnings  
254         (262 missing values generated)  
255  
256     97 . gen 110dlnHourlyEarnings=110d.lnHourlyEarnings  
257         (263 missing values generated)  
258  
259     98 . gen 111dlnHourlyEarnings=111d.lnHourlyEarnings  
260         (264 missing values generated)  
261  
262     99 . gen 112dlnHourlyEarnings=112d.lnHourlyEarnings  
263         (265 missing values generated)  
264  
265     100 . gen 124dlnHourlyEarnings=124d.lnHourlyEarnings  
266         (277 missing values generated)  
267  
268     101 . gen 136dlnHourlyEarnings=136d.lnHourlyEarnings  
269         (289 missing values generated)  
270  
271     102 . gen 148dlnHourlyEarnings=148d.lnHourlyEarnings  
272         (301 missing values generated)  
273  
274     103 .  
275     104 . gen dlnWeeklyEarnings=d.lnWeeklyEarnings  
276         (253 missing values generated)  
277  
278     105 . gen 11dlnWeeklyEarnings=11d.lnWeeklyEarnings  
279         (254 missing values generated)  
280  
281     106 . gen 12dlnWeeklyEarnings=12d.lnWeeklyEarnings  
282         (255 missing values generated)  
283  
284     107 . gen 13dlnWeeklyEarnings=13d.lnWeeklyEarnings  
285         (256 missing values generated)  
286  
287     108 . gen 14dlnWeeklyEarnings=14d.lnWeeklyEarnings  
288         (257 missing values generated)  
289  
290     109 . gen 15dlnWeeklyEarnings=15d.lnWeeklyEarnings  
291         (258 missing values generated)  
292  
293     110 . gen 16dlnWeeklyEarnings=16d.lnWeeklyEarnings  
294         (259 missing values generated)  
295  
296     111 . gen 17dlnWeeklyEarnings=17d.lnWeeklyEarnings  
297         (260 missing values generated)  
298  
299     112 . gen 18dlnWeeklyEarnings=18d.lnWeeklyEarnings
```

```
300      (261 missing values generated)
301
302 113 . gen 19dlnWeeklyEarnings=19d.lnWeeklyEarnings
303      (262 missing values generated)
304
305 114 . gen 110dlnWeeklyEarnings=110d.lnWeeklyEarnings
306      (263 missing values generated)
307
308 115 . gen 111dlnWeeklyEarnings=111d.lnWeeklyEarnings
309      (264 missing values generated)
310
311 116 . gen 112dlnWeeklyEarnings=112d.lnWeeklyEarnings
312      (265 missing values generated)
313
314 117 . gen 124dlnWeeklyEarnings=124d.lnWeeklyEarnings
315      (277 missing values generated)
316
317 118 . gen 136dlnWeeklyEarnings=136d.lnWeeklyEarnings
318      (289 missing values generated)
319
320 119 . gen 148dlnWeeklyEarnings=148d.lnWeeklyEarnings
321      (301 missing values generated)
322
323 120 .
324 121 . gen dlnServiceCount=d.lnServiceCount
325      (1 missing value generated)
326
327 122 . gen 11dlnServiceCount=11d.lnServiceCount
328      (2 missing values generated)
329
330 123 . gen 12dlnServiceCount=12d.lnServiceCount
331      (3 missing values generated)
332
333 124 . gen 13dlnServiceCount=13d.lnServiceCount
334      (4 missing values generated)
335
336 125 . gen 14dlnServiceCount=14d.lnServiceCount
337      (5 missing values generated)
338
339 126 . gen 15dlnServiceCount=15d.lnServiceCount
340      (6 missing values generated)
341
342 127 . gen 16dlnServiceCount=16d.lnServiceCount
343      (7 missing values generated)
344
345 128 . gen 17dlnServiceCount=17d.lnServiceCount
346      (8 missing values generated)
347
348 129 . gen 18dlnServiceCount=18d.lnServiceCount
349      (9 missing values generated)
350
```

```

351 130 . gen 19dlnServiceCount=19d.lnServiceCount
352     (10 missing values generated)
353
354 131 . gen 110dlnServiceCount=110d.lnServiceCount
355     (11 missing values generated)
356
357 132 . gen 111dlnServiceCount=111d.lnServiceCount
358     (12 missing values generated)
359
360 133 . gen 112dlnServiceCount=112d.lnServiceCount
361     (13 missing values generated)
362
363 134 . gen 124dlnServiceCount=124d.lnServiceCount
364     (25 missing values generated)
365
366 135 . gen 136dlnServiceCount=136d.lnServiceCount
367     (37 missing values generated)
368
369 136 . gen 148dlnServiceCount=148d.lnServiceCount
370     (49 missing values generated)
371
372 137 .
373 138 . /*
374      > The project is to forecast the March non-seasonally adjusted estimates of
375      ave
376      > rage weekly earnings and total employment for private employers (total
377      privat
378      > e) for a Florida MSA of your choice and write up a professional report on
379      you
380      > r forecast.
381      > */
382 139 . /* Count and WeeklyEarnings */
383 140 .
384 141 . summ Count WeekHours HourlyEarnings WeeklyEarnings ServiceCount
385
386
387
388
389
390
391
392
393
394
395
396
397
398
```

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
HourlyEarnings	122	19.70344	2.910126	15.01	24.6
WeeklyEarnings	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
lnHourlyEarnings	122	2.969891	.148565	2.708717	3.202746
lnWeeklyEarnings	122	6.57194	.1198394	6.222159	6.820126

```

399      lnServiceC~t |      374      2.16967      .5975865      1.360977      3.12676
400
401      143 .
402      144 . ac lnCount, saving(lnCount_ac, replace)
403          (file lnCount_ac.gph saved)
404
405      145 . pac lnCount, saving(lnCount_pac, replace)
406          (file lnCount_pac.gph saved)
407
408      146 . graph combine lnCount_ac.gph lnCount_pac.gph, saving(lnCount_ac_pac,
409      replace)
410          (file lnCount_ac_pac.gph saved)
411
412      147 . graph export "lnCount_ac_pac.png", replace
413          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
414      Sets
415          > /Final Project/lnCount_ac_pac.png written in PNG format)
416
417      148 . ** Probably need to difference
418
419      149 .
420      150 . ac lnWeeklyEarnings, saving(lnWeeklyEarnings_ac, replace)
421          (file lnWeeklyEarnings_ac.gph saved)
422
423      151 . pac lnWeeklyEarnings, saving(lnWeeklyEarnings_pac, replace)
424          (file lnWeeklyEarnings_pac.gph saved)
425
426      152 . graph combine lnWeeklyEarnings_ac.gph lnWeeklyEarnings_pac.gph,
427      saving(lnWeek
428          > lyEarnings_ac_pac, replace)
429          (file lnWeeklyEarnings_ac_pac.gph saved)
430
431      153 . graph export "lnWeeklyEarnings_ac_pac.png", replace
432          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
433      Sets
434          > /Final Project/lnWeeklyEarnings_ac_pac.png written in PNG format)
435
436      154 . ** Probably need to differencen b
437
438      155 .
439      156 . *starter models for count
440      157 . *I used a pair plot to examine the rise and fall of variables with respect
441      to
442          > each other
443      158 . reg d.lnCount l(12,24,36,48)d.lnCount // .01637
444
445          Source |      SS           df           MS       Number of obs     =
446          325
447          -----+----- F(4, 320)      =
448          16.54
449          Model |   .017539188      4   .004384797   Prob > F      =
450          0.0000

```

```

441      Residual | .084856979      320  .000265178  R-squared      =
442      0.1713
443      -----
444      Total | .102396167      324  .000316038  Root MSE       =
445      .01628
446      -----
447      D.lnCount |      Coef.    Std. Err.      t     P>|t|      [95% Conf.
448      Interval]
449      -----
450      lnCount |
451      L12D. |   .3609966   .0621085    5.81  0.000   .238804
452      .4831893
453      L24D. |   .137848   .0617615    2.23  0.026   .016338
454      .259358
455      L36D. |  -.0160136   .0614584   -0.26  0.795  -.1369272
456      .1049
457      L48D. |   .1265117   .0585322    2.16  0.031   .0113551
458      .2416683
459      |
460      _cons |   .0017116   .0009853    1.74  0.083  -.0002269
461      .0036502
462      -----
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479      159 . scalar drop _all
480
481      160 . quietly forval w=12(12)144 {
482
483      161 . scalar list
484      RWmaxobs144 =      144
485      RWminobs144 =      12
486      RWrmse144 =   .0172276
487      RWmaxobs132 =      132
488      RWminobs132 =      12
489      RWrmse132 =   .0172128
490      RWmaxobs120 =      120
491      RWminobs120 =      12
492      RWrmse120 =   .01721825
493      RWmaxobs108 =      108
494      RWminobs108 =      12
495      RWrmse108 =   .01723674
496      RWmaxobs96 =       96
497      RWminobs96 =       12
498      RWrmse96 =   .01722006
499      RWmaxobs84 =       84
500      RWminobs84 =       12
501      RWrmse84 =   .01726063

```

```

480      RWmaxobs72 =      72
481      RWminobs72 =      12
482      RWrmse72 = .01722377
483      RWmaxobs60 =      60
484      RWminobs60 =      12
485      RWrmse60 = .0173443
486      RWmaxobs48 =      48
487      RWminobs48 =      12
488      RWrmse48 = .01755803
489      RWmaxobs36 =      36
490      RWminobs36 =      12
491      RWrmse36 = .01805924
492      RWmaxobs24 =      24
493      RWminobs24 =      12
494      RWrmse24 = .0185871
495      RWmaxobs12 =      12
496      RWminobs12 =      12
497      RWrmse12 = .02320505
498
499      162 . /*
500      > RWmaxobs132 =      132
501      > RWminobs132 =      12
502      > RWrmse132 = .0172128
503      > */
504      163 .
505      164 . reg d.lnCount 1(5,12,24,36,48)d.lnCount 1(5)d.lnWeekHours m5 // .01711
506
507      Source |      SS          df          MS          Number of obs      =
508      -----+----- F(7, 108)      =
5.94
509      Model | .012171566          7  .001738795  Prob > F      =
0.0000
510      Residual | .03162877          108  .000292859  R-squared      =
0.2779
511      -----+----- Adj R-squared      =
0.2311
512      Total | .043800336          115  .000380872  Root MSE      =
.01711
513
514      -----
515      D.lnCount |      Coef.      Std. Err.          t      P>|t|      [95% Conf.
516      Interval]
517      -----
518      lnCount |
519      L5D. | -.1231921  .0845717  -1.46  0.148  -.290828
.0444438
519      L12D. | .5811114  .1685831  3.45  0.001  .2469504
.9152724

```

```

520          L24D. | -.1196017   .1627467   -0.73   0.464   -.4421938
521          .2029904
521          L36D. | .2532303   .1742525   1.45   0.149   -.0921684
521          .5986291
522          L48D. | .1341638   .1858633   0.72   0.472   -.2342495
522          .5025771
523          |
524          lnWeekHours |
525          L5D. | .0170123   .0364906   0.47   0.642   -.0553184
525          .089343
526          |
527          m5 | .0067588   .0061605   1.10   0.275   -.0054524
527          .0189699
528          _cons | .0004279   .0018229   0.23   0.815   -.0031854
528          .0040412
529          -----
530
531      165 . scalar drop _all
532
533      166 . quietly forval w=12(12)84 {
534
535      167 . scalar list
536          RWmaxobs84 =      84
537          RWminobs84 =      23
538          RWrmse84 = .01950911
539          RWmaxobs72 =      72
540          RWminobs72 =      23
541          RWrmse72 = .01949719
542          RWmaxobs60 =      60
543          RWminobs60 =      23
544          RWrmse60 = .0199438
545          RWmaxobs48 =      48
546          RWminobs48 =      23
547          RWrmse48 = .02035982
548          RWmaxobs36 =      36
549          RWminobs36 =      23
550          RWrmse36 = .02138785
551          RWmaxobs24 =      24
552          RWminobs24 =      23
553          RWrmse24 = .02268585
554          RWmaxobs12 =      12
555          RWminobs12 =      12
556          RWrmse12 = .05004898
557
558      168 . /*
559          > RWmaxobs84 =      84
560          > RWminobs84 =      23
561          > RWrmse84 = .01950911
562          > */
563      169 .

```

```

564      170 . /*
565      > gsreg dlnCount l1dlnCount l2dlnCount l3dlnCount l4dlnCount l5dlnCount
566      > l6dlnCo
567      > unt ///
568      >      17dlnCount 18dlnCount 19dlnCount 110dlnCount 111dlnCount
569      >      112dlnCount
570      >      /**
571      >      124dlnCount 136dlnCount 148dlnCount /**
572      >      if tin(1990m1,2021m1), /**
573      >      ncomb(1,12) aic outsamp(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
574      m11
575      >      m12) /**
576      >      samesample nindex( -1 aic -1 bic -1 rmse_out)
577      results(gsreg_dlnCount)
578      >      replace
579      > */
580      171 .
581      172 . *gsreg suggestions
582      173 . reg d.lnCount l12d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
583
584      Source |      SS          df          MS      Number of obs     =
585      361      -----
586      Model |   .022616974          12   .001884748      Prob > F      =
587      6.91      0.0000
588      Residual |   .094871179          348   .000272618      R-squared      =
589      0.1925
590      -----
591      Total |   .117488153          360   .000326356      Root MSE      =
592      .01651
593
594      -----
595      D.lnCount |      Coef.      Std. Err.          t      P>|t|      [95% Conf.
596      Interval]
597      -----
598
599      lnCount |
600      L12D. |   .1748571   .0594184      2.94      0.003      .0579928
601      .2917215
602
603      |
604      m1 |   -.0099477   .0043004     -2.31      0.021     -.0184056
605      -.0014897
606
607      m2 |   .0009939   .0042297      0.23      0.814     -.0073251
608      .0093129
609
610      m3 |   .0030247   .0042759      0.71      0.480     -.0053851
611      .0114345
612
613      m4 |   -.0071933   .0042648     -1.69      0.093     -.0155814
614      .0011948

```

597		m5 -.0098194	.0043178	-2.27	0.024	-.0183118
598		-.0013271				
599		m6 -.0133285	.0043874	-3.04	0.003	-.0219576
		-.0046994				
600		m7 -.0091828	.0042967	-2.14	0.033	-.0176336
		-.000732				
601		m8 -.0017998	.0042632	-0.42	0.673	-.0101846
		.006585				
602		m9 -.006737	.0042824	-1.57	0.117	-.0151597
		.0016858				
603		m10 .0062149	.0042795	1.45	0.147	-.0022021
		.0146319				
604		m11 .0042124	.0042811	0.98	0.326	-.0042078
		.0126325				
605		_cons .0072199	.0030452	2.37	0.018	.0012306
		.0132093				

606		-				
607	174	. scalar drop _all				
608						
609	175	. quietly forval w=12(12)144 {				
610						
611	176	. scalar list				
612		RWmaxobs144 = 144				
613		RWminobs144 = 12				
614		RWrmse144 = .01824906				
615		RWmaxobs132 = 132				
616		RWminobs132 = 12				
617		RWrmse132 = .01832173				
618		RWmaxobs120 = 120				
619		RWminobs120 = 12				
620		RWrmse120 = .01833557				
621		RWmaxobs108 = 108				
622		RWminobs108 = 12				
623		RWrmse108 = .01841089				
624		RWmaxobs96 = 96				
625		RWminobs96 = 12				
626		RWrmse96 = .01836974				
627		RWmaxobs84 = 84				
628		RWminobs84 = 12				
629		RWrmse84 = .01849267				
630		RWmaxobs72 = 72				
631		RWminobs72 = 12				
632		RWrmse72 = .01861349				
633		RWmaxobs60 = 60				
634		RWminobs60 = 12				
635		RWrmse60 = .01911515				
636		RWmaxobs48 = 48				
637		RWminobs48 = 12				
638		RWrmse48 = .01922268				

```

639      RWmaxobs36 =          36
640      RWminobs36 =          12
641      RWrmse36 = .01991683
642      RWmaxobs24 =          24
643      RWminobs24 =          12
644      RWrmse24 = .02022186
645      RWmaxobs12 =          12
646      RWminobs12 =          12
647      RWrmse12 = .02009249
648
649 177 . /*
650      > RWmaxobs144 =          144
651      > RWminobs144 =          12
652      > RWrmse144 = .01824906
653      > */
654 178 .
655 179 . reg d.lnCount l(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
656
657      Source |       SS           df          MS      Number of obs     =
658      337
659      -----+----- F(13, 323)     =
660      5.68
661      Model | .019946057      13   .001534312  Prob > F     =
662      0.0000
663      Residual | .087185203    323   .000269923 R-squared     =
664      0.1862
665      -----+----- Adj R-squared     =
666      0.1534
667      Total | .107131259    336   .000318843 Root MSE      =
668      .01643
669
670      -----
671      D.lnCount |      Coef.    Std. Err.      t     P>|t| [95% Conf.
672      Interval]
673
674      -----
675
676      lnCount |
677      L12D. | .1849403   .0636401     2.91   0.004   .059739
678      .3101417
679      L36D. | -.049332   .0606582    -0.81   0.417  -.1686671
680      .0700031
681
682      |
683      m1 | -.0073418   .0044769    -1.64   0.102  -.0161493
684      .0014658
685      m2 | .0022711   .0043559     0.52   0.602  -.0062984
686      .0108407
687      m3 | .0043593   .0044160     0.99   0.324  -.0043285
688      .0130471
689      m4 | -.0065438   .0043922    -1.49   0.137  -.0151847
690      .002097

```

```

675          m5 | -.0089215 .0045194 -1.97 0.049 -.0178126
676          -.0000304
677          m6 | -.0133453 .0046241 -2.89 0.004 -.0224425
678          -.004248
679          m7 | -.0085154 .004457 -1.91 0.057 -.0172839
680          .0002531
681          m8 | -.0004554 .004392 -0.10 0.917 -.0090959
682          .0081852
683          m9 | -.0056625 .0044299 -1.28 0.202 -.0143775
684          .0030526
685          m10 | .0071688 .0044386 1.62 0.107 -.0015635
686          .0159011
687          m11 | .0042074 .0044259 0.95 0.343 -.0044998
688          .0129146
689          _cons | .0067355 .0031722 2.12 0.034 .0004948
690          .0129762
691          -----
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
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

```

```

717      RWmaxobs36 =          36
718      RWminobs36 =          12
719      RWrmse36 = .01863303
720      RWmaxobs24 =          24
721      RWminobs24 =          12
722      RWrmse24 = .0196745
723      RWmaxobs12 =          12
724      RWminobs12 =          12
725      RWrmse12 = .01880291
726
727 183 . /*
728 > RWmaxobs144 =          144
729 > RWminobs144 =          12
730 > RWrmse144 = .01777071
731 > */
732 184 .
733 185 . /*
734 > gsreg dlnCount 11dlnCount 12dlnCount 13dlnCount 14dlnCount 15dlnCount
16dlnCo
735 > unt /**
736 >      17dlnCount 18dlnCount 19dlnCount 110dlnCount 111dlnCount
112dlnCount
737 > /**
738 >      124dlnCount 136dlnCount /**
739 >      11dlnWeekHours 12dlnWeekHours 13dlnWeekHours 14dlnWeekHours
15dlnWeek
740 > Hours 16dlnWeekHours /**
741 >      17dlnWeekHours 18dlnWeekHours 19dlnWeekHours 110dlnWeekHours
111dlnWe
742 > ekHours 112dlnWeekHours /**
743 >      124dlnWeekHours 136dlnWeekHours /**
744 >      11dlnHourlyEarnings 12dlnHourlyEarnings 13dlnHourlyEarnings
14dlnHour
745 > lyEarnings /**
746 >      15dlnHourlyEarnings 16dlnHourlyEarnings /**
747 >      17dlnHourlyEarnings 18dlnHourlyEarnings 19dlnHourlyEarnings
110dlnHou
748 > rlyEarnings /**
749 >      111dlnHourlyEarnings 112dlnHourlyEarnings /**
750 >      124dlnHourlyEarnings 136dlnHourlyEarnings /**
751 >      11dlnWeeklyEarnings 12dlnWeeklyEarnings 13dlnWeeklyEarnings
14dlnWeek
752 > lyEarnings /**
753 >      15dlnWeeklyEarnings 16dlnWeeklyEarnings /**
754 >      17dlnWeeklyEarnings 18dlnWeeklyEarnings 19dlnWeeklyEarnings
110dlnWee
755 > klyEarnings /**
756 >      111dlnWeeklyEarnings 112dlnWeeklyEarnings /**
757 >      124dlnWeeklyEarnings 136dlnWeeklyEarnings /**
758 >      if tin(2011m1,2021m1), /**

```

```

759      >          ncomb(1,4) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
760      m11)
761      > /**
762      >          samesample nindex( -1 aic -1 bic -1 rmse_out)
763      results(gsreg_dlnCount_
764      > Full) replace
765      > */
766      186 .
767      187 . reg d.lnCount 14d.lnWeekHours 19d.lnWeekHours 18d.lnHourlyEarnings m1 m2
768      m3 m
769      > 4 m5 m6 m7 m8 m9 m10 m11
770
771      Source |      SS           df           MS           Number of obs   =
772      112
773      -----+----- F(14, 97)   =
774      3.24
775      Model |  .013917393      14   .0009941  Prob > F   =
776      0.0003
777      Residual |  .029798432      97   .0003072  R-squared   =
778      0.3184
779      -----+----- Adj R-squared   =
780      0.2200
781      Total |  .043715825     111   .000393836 Root MSE   =
782      .01753
783
784      -----
785      D.lnCount |      Coef.       Std. Err.        t      P>|t|      [95% Conf.
786      Interval]
787      -----
788      lnWeekHours |
789      L4D. |  -.0013847   .0384012    -0.04   0.971   -.0776005
790      .074831
791      L9D. |   .0397686   .0385964     1.03   0.305   -.0368346
792      .1163718
793      |
794      lnHourlyEa~s |
795      L8D. |  -.039029   .0414024    -0.94   0.348   -.1212014
796      .0431433
797      |
798      m1 |  -.0097045   .0078517    -1.24   0.219   -.0252879
799      .0058789
800      m2 |   .0000949   .0079445     0.01   0.990   -.0156727
801      .0158626
802      m3 |  -.004712   .0083585    -0.56   0.574   -.0213013
803      .0118773
804      m4 |  -.0273667   .0081729    -3.35   0.001   -.0435876
805      -.0111459
806      m5 |  -.0076836   .0081259    -0.95   0.347   -.0238112
807      .008444

```

```

790          m6 | -.020254 .0081465 -2.49 0.015 -.0364227
791          -.0040854
792          m7 | -.0130812 .0081852 -1.60 0.113 -.0293265
793          .0031642
794          m8 | .0041701 .0081051 0.51 0.608 -.0119164
795          .0202565
796          m9 | -.0089171 .0082764 -1.08 0.284 -.0253435
797          .0075093
798          m10 | .0153608 .0081153 1.89 0.061 -.0007459
799          .0314674
800          m11 | .0040463 .0079619 0.51 0.612 -.0117559
801          .0198485
802          _cons | .0094122 .0056462 1.67 0.099 -.0017939
803          .0206183
804          -----
805          -
806
807          188 . scalar drop _all
808
809          189 . quietly forval w=12(12)84 {
810
811          190 . scalar list
812          RWmaxobs84 = 84
813          RWminobs84 = 2
814          RWrmse84 = .01847546
815          RWmaxobs72 = 72
816          RWminobs72 = 2
817          RWrmse72 = .01855448
818          RWmaxobs60 = 60
819          RWminobs60 = 2
820          RWrmse60 = .01850723
821          RWmaxobs48 = 48
822          RWminobs48 = 2
823          RWrmse48 = .01850217
824          RWmaxobs36 = 36
825          RWminobs36 = 2
826          RWrmse36 = .01942535
827          RWmaxobs24 = 24
828          RWminobs24 = 2
829          RWrmse24 = .02208272
830          RWmaxobs12 = 12
831          RWminobs12 = 2
832          RWrmse12 = .0176238
833
834
835          191 . /*
836          > RWmaxobs12 = 12
837          > RWminobs12 = 2
838          > RWrmse12 = .0176238
839          > */
840
841          192 .

```

```

832    193 . /*-----
833    ---> -*/
834    194 .
835    195 . *starter models for weekly earnings
836    196 . reg d.lnWeeklyEarnings l1d.lnWeekHours l1d.lnHourlyEarnings
837
838      Source |       SS          df          MS      Number of obs   =
839      120
840      -----+----- F(2, 117)   =
841      1.48
840      Model |   .00720643          2   .003603215  Prob > F   =
841      0.2317
841      Residual |   .284697808        117   .002433315  R-squared   =
841      0.0247
842      -----+----- Adj R-squared   =
842      0.0080
843      Total |   .291904237        119   .002452977  Root MSE   =
843      .04933
844
845      -----
846      D.          |
847      lnWeeklyEa~s |     Coef.    Std. Err.      t    P>|t|    [95% Conf.
848      Interval]
848      -----
849      lnWeekHours |
850      LD. |   -.1344969   .105932   -1.27   0.207   -.3442897
850      .0752959
851      |
852      lnHourlyEa~s |
853      LD. |   .0732899   .1167768   0.63   0.531   -.1579805
853      .3045603
854      |
855      _cons |   .0016809   .0045103   0.37   0.710   -.0072515
855      .0106133
856      -----
857
858      197 . scalar drop _all
859
860      198 . quietly forval w=12(12)84 {
861
862      199 . scalar list
863      RWmaxobs84 =      84
864      RWminobs84 =      2
865      RWrmse84 =   .06184586
866      RWmaxobs72 =      72
867      RWminobs72 =      2
868      RWrmse72 =   .06163593

```

```

869      RWmaxobs60 =       60
870      RWminobs60 =        2
871      RWrmse60 = .06145693
872      RWmaxobs48 =       48
873      RWminobs48 =        2
874      RWrmse48 = .06185207
875      RWmaxobs36 =       36
876      RWminobs36 =        2
877      RWrmse36 = .06201342
878      RWmaxobs24 =       24
879      RWminobs24 =        2
880      RWrmse24 = .06223845
881      RWmaxobs12 =       12
882      RWminobs12 =        2
883      RWrmse12 = .06581989
884
885 200 . /*
886      > RWmaxobs60 =       60
887      > RWminobs60 =        2
888      > RWrmse60 = .06145693
889      > */
890 201 .
891 202 . /*
892      > gsreg dlnWeeklyEarnings 11dlnWeeklyEarnings 12dlnWeeklyEarnings
     13dlnWeeklyEa
893      > rnings ///
894      >      14dlnWeeklyEarnings 15dlnWeeklyEarnings 16dlnWeeklyEarnings ///
895      >      17dlnWeeklyEarnings 18dlnWeeklyEarnings 19dlnWeeklyEarnings
     110dlnWee
896      > klyEarnings ///
897      >      111dlnWeeklyEarnings 112dlnWeeklyEarnings ///
898      >      124dlnWeeklyEarnings 136dlnWeeklyEarnings ///
899      >      if tin(2011m1,2021m1), ///
900      >      ncomb(1,12) aic outsamp(24) ///
901      >      samesample nindex( -1 aic -1 bic -1 rmse_out)
     results(gsreg_dlnWeekly
902      > Earnings) replace
903      > */
904 203 .
905 204 . reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings m1 m2 m3
     m4
906      > m5 m6 m7 m8 m9 m10 m11
907
908      Source |      SS          df          MS      Number of obs      =
909      116
910      -----+----- F(13, 102)      =
911      2.22
910      Model | .062824046      13  .004832619  Prob > F      =
911      0.0134
911      Residual | .221893915      102  .002175431  R-squared      =
911      0.2207

```

```

912      -----+-----+-----+-----+-----+-----+-----+
913      0.1213          Adj R-squared      =
914      Total | .284717961      115 .002475808 Root MSE      =
915      .04664
916      -----
917      D.          |           Coef.   Std. Err.      t      P>|t|      [95% Conf.
918      lnWeeklyEa~s |           Interval]
919      -----
920      L3D. | -.230666  .0948354 -2.43  0.017 -.4187716
921      -.0425604
922      L5D. | -.1701094  .0948714 -1.79  0.076 -.3582864
923      .0180676
924      m1 | -.0129295  .0212711 -0.61  0.545 -.0551207
925      .0292616
926      m2 | .0230655  .0211959  1.09  0.279 -.0189765
927      .0651074
928      m3 | -.0120234  .0220335 -0.55  0.586 -.0557268
929      .03168
930      m4 | .0106776  .0218502  0.49  0.626 -.0326621
931      .0540173
932      m5 | .0379684  .0215915  1.76  0.082 -.0048583
933      .0807952
934      m6 | .0272323  .0215653  1.26  0.210 -.0155424
935      .0700069
936      m7 | -.0216364  .0213979 -1.01  0.314 -.064079
937      .0208063
938      m8 | .0154331  .0210082  0.73  0.464 -.0262366
939      .0571028
940      m9 | .0206819  .021546  0.96  0.339 -.0220544
941      .0634182
942      m10 | .0210709  .0217906  0.97  0.336 -.0221507
943      .0642924
944      m11 | .0089656  .0216561  0.41  0.680 -.0339891
945      .0519203
946      _cons | -.0075658  .0151177 -0.50  0.618 -.0375517
947      .0224201
948      -----
949      -
950      205 . scalar drop _all
951
952      206 . quietly forval w=12(12)84 {
953
954      207 . scalar list
955      RWmaxobs84 =
```

```

943      RWminobs84 =          2
944      RWrmse84 = .06004448
945      RWmaxobs72 =         72
946      RWminobs72 =          2
947      RWrmse72 = .0605325
948      RWmaxobs60 =         60
949      RWminobs60 =          2
950      RWrmse60 = .06068574
951      RWmaxobs48 =         48
952      RWminobs48 =          2
953      RWrmse48 = .06037733
954      RWmaxobs36 =         36
955      RWminobs36 =          2
956      RWrmse36 = .06130591
957      RWmaxobs24 =         24
958      RWminobs24 =          2
959      RWrmse24 = .06544875
960      RWmaxobs12 =         12
961      RWminobs12 =          2
962      RWrmse12 = .07012904
963
964 208 . /*
965      > RWmaxobs84 =         84
966      > RWminobs84 =          2
967      > RWrmse84 = .06004448
968      > */
969 209 .
970 210 . reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings
   17d.lnWeekly
971      > Earnings
972
973      Source |      SS          df          MS      Number of obs     =
974      114
975      -----+----- F(3, 110)      =
976      4.04
977      Model | .027592174      3  .009197391  Prob > F      =
978      0.0091
979      Residual | .250484433     110  .002277131  R-squared      =
980      0.0992
981      -----+----- Adj R-squared      =
982      0.0747
983      Total | .278076607     113  .002460855  Root MSE      =
984      .04772
985
986      -----
987      D.          |
988      lnWeeklyEa~s |      Coef.    Std. Err.      t      P>|t|      [95% Conf.
989      Interval]
990      -----

```

```

984      lnWeeklyEa~s |
985          L3D. | -.2446256 .0905231 -2.70 0.008 -.4240211
986          -.0652301
987          L5D. | -.1957955 .0901769 -2.17 0.032 -.3745049
988          -.0170861
989          L7D. | .0649411 .0912448 0.71 0.478 -.1158846
990          .2457668
991          |
992          _cons | .0027599 .0044776 0.62 0.539 -.0061138
993          .0116335
994          -----
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027

```

-

```

211 . scalar drop _all
212 . quietly forval w=12(12)84 {
213 . scalar list
214 . /*
215 > RWmaxobs84 = 84
216 > RWminobs84 = 2
217 > RWrmse84 = .05250414
218 > RWmaxobs72 = 72
219 > RWminobs72 = 2
220 > RWrmse72 = .05277312
221 > RWmaxobs60 = 60
222 > RWminobs60 = 2
223 > RWrmse60 = .05308846
224 > RWmaxobs48 = 48
225 > RWminobs48 = 2
226 > RWrmse48 = .05299891
227 > RWmaxobs36 = 36
228 > RWminobs36 = 2
229 > RWrmse36 = .05349229
230 > RWmaxobs24 = 24
231 > RWminobs24 = 2
232 > RWrmse24 = .05283688
233 > RWmaxobs12 = 12
234 > RWminobs12 = 2
235 > RWrmse12 = .06056213
236 . /*
237 > RWmaxobs84 = 84
238 > RWminobs84 = 2
239 > RWrmse84 = .05250414
240 > */
241 . gsreg dlnCount 11dlnCount 12dlnCount 13dlnCount 14dlnCount 15dlnCount
242 16dlnCo
243 > unt ///

```

```

1028      >          17dlnCount 18dlnCount 19dlnCount 110dlnCount 111dlnCount
1029      112dlnCount
1030      >          124dlnCount ///
1030      >          11dlnWeekHours 12dlnWeekHours 13dlnWeekHours 14dlnWeekHours
1031      15dlnWeek
1031      >          Hours 16dlnWeekHours ///
1032      >          17dlnWeekHours 18dlnWeekHours 19dlnWeekHours 110dlnWeekHours
1032      111dlnWe
1033      >          ekHours 112dlnWeekHours 124dlnWeekHours ///
1034      >          11dlnHourlyEarnings 12dlnHourlyEarnings 13dlnHourlyEarnings
1034      14dlnHour
1035      >          lyEarnings ///
1036      >          15dlnHourlyEarnings 16dlnHourlyEarnings ///
1037      >          17dlnHourlyEarnings 18dlnHourlyEarnings 19dlnHourlyEarnings
1037      110dlnHou
1038      >          rlyEarnings ///
1039      >          111dlnHourlyEarnings 112dlnHourlyEarnings 124dlnHourlyEarnings ///
1040      >          11dlnWeeklyEarnings 12dlnWeeklyEarnings 13dlnWeeklyEarnings
1040      14dlnWeek
1041      >          lyEarnings ///
1042      >          15dlnWeeklyEarnings 16dlnWeeklyEarnings ///
1043      >          17dlnWeeklyEarnings 18dlnWeeklyEarnings 19dlnWeeklyEarnings
1043      110dlnWee
1044      >          klyEarnings ///
1045      >          111dlnWeeklyEarnings 112dlnWeeklyEarnings 124dlnWeeklyEarnings if
1045      tin
1046      >          (2011m1,2021m1), ///
1047      >          ncomb(1,4) aic outsamp(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
1047      m11)
1048      >          ///
1049      >          samesample nindex( -1 aic -1 bic -1 rmse_out)
1049      results(gsreg_dlnWeekly
1050      >          Earnings_Full) replace
1051      >          */
1052      217 .
1053      218 . log close
1054      name: <unnamed>
1055      log: /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Probl
1056      > em Sets/Final Project/Final Project.smcl
1057      log type: smcl
1058      closed on: 8 Apr 2021, 17:48:56
1059
-----
```

--

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

Section
<u>Abstract</u>
<u>Table of Contents</u>
<u>Introduction</u>
<u>Data</u>
<u>Model Estimation and Selection</u>
<u>Final Results</u>
<u>Conclusion</u>
<u>Appendix A</u>
<u>Appendix B</u>

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 available 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 `tset` 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 the 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

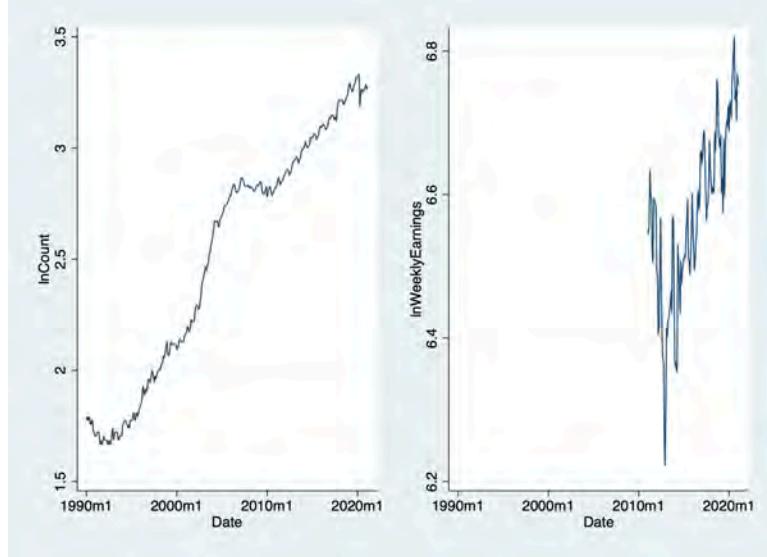
Of particular note in the non-transformed data is that there is a wide variation in the number of private employment for The Villages. The Villages has an incredibly high range where the maximum value is more than five times greater than the minimum value. Meanwhile, hourly earnings have not increased 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
InCount	374	2.5174	.5398403	1.667707	3.332205
InWeekHours	123	3.602488	.10385	3.342862	3.824284
InHourlyEarnings	123	2.970779	.1482819	2.708717	3.202746
InWeeklyEarnings	122	6.571775	.1195694	6.222159	6.820126
InServiceCount	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 InCount and InWeeklyEarnings



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 InWeeklyEarnings on the right, the data does not start until 2011 but the graph has been adjusted to show the same time period as InCount. 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 InCount and InWeekly Earnings.

Figure 2 AC and PAC of InCount

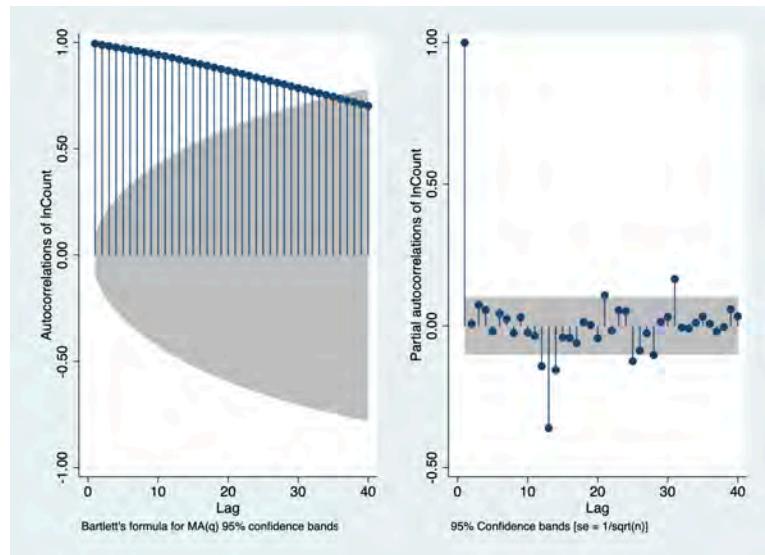
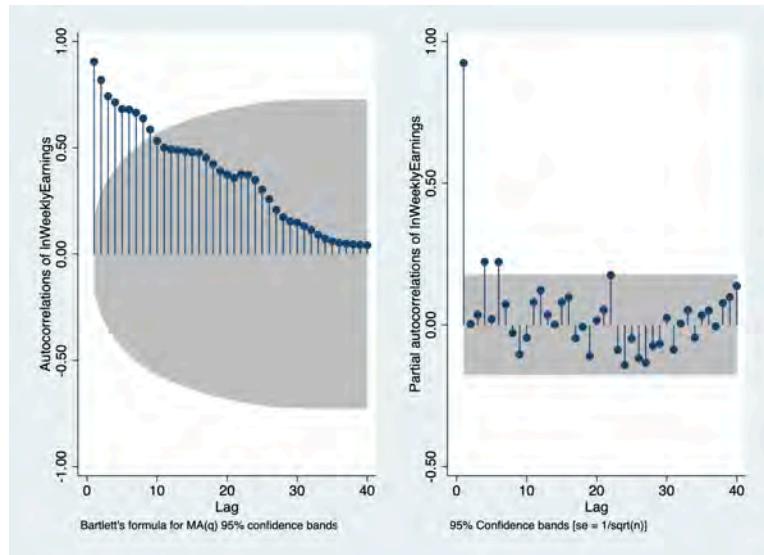


Figure 3 AC and PAC of InWeeklyEarnings



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-eighth 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 1(12,24,36,48)d.lnCount`. When run through a rolling window forecast to find the optimal window size, I found that this initial 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 1(5,12,24,36,48)d.lnCount 1(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
<code>reg d.lnCount 112d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11</code>	144	0.01824906
<code>reg d.lnCount 1(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11</code>	144	0.01777071
<code>reg d.lnCount 14d.lnWeekHours 19d.lnWeekHours 18d.lnHourlyEarnings m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11</code>	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 ran 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
<code>reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11</code>	84	0.06004448
<code>reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings 17d.lnWeeklyEarnings</code>	84	0.05250414

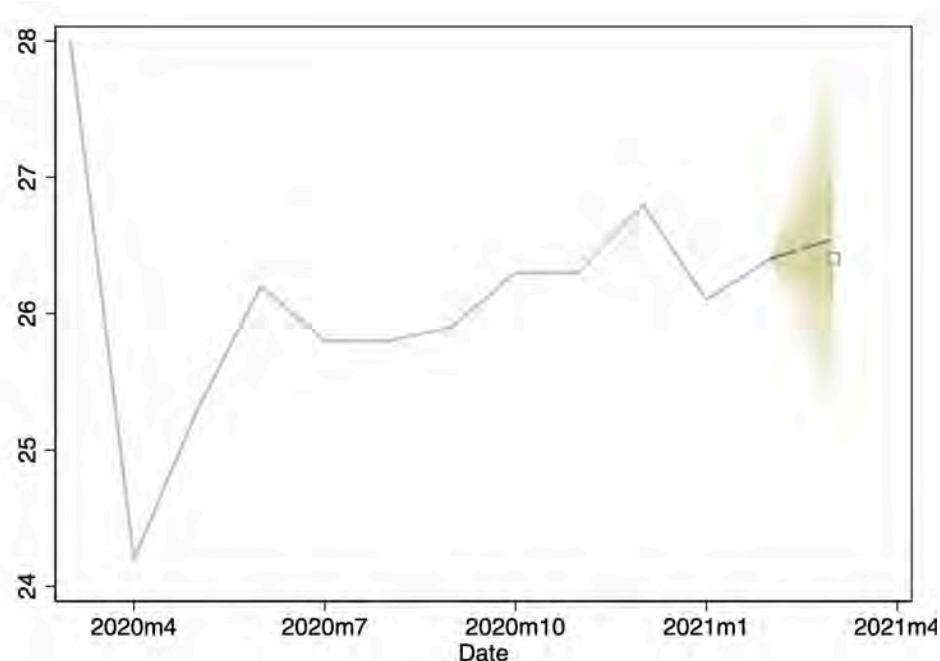
Final Results

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 1(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.

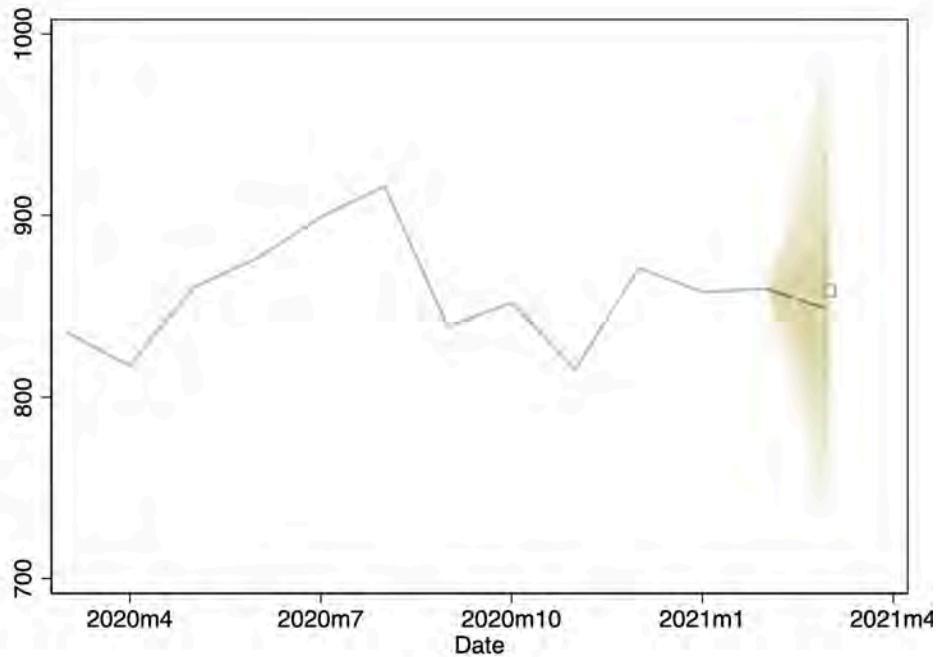
Figure 4 Fan Chart of Total Private Employment



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 1(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

```

1  clear
2  set more off
3
4  cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Final
Project"
5  log using "Final Project.smcl", replace
6  import delimited "TS2020_Final_Project_Monthly.txt"
7  rename smu12455400500000001 Count
8  rename smu12455400500000002 WeekHours
9  rename smu12455400500000003 HourlyEarnings
10 rename smu12455400500000011 WeeklyEarnings
11 rename smu12455400800000001 ServiceCount
12
13
14 label variable Count "Count"
```

```
15 label variable WeekHours "WeekHours"
16 label variable HourlyEarnings "HourlyEarnings"
17 label variable WeeklyEarnings "WeeklyEarnings"
18 label variable ServiceCount "ServiceCount"
19
20
21 gen datec=date(date, "YMD")
22 gen Date=mofd(datec)
23 gen month=month(datec)
24 format Date %tm
25 tsset Date
26
27 gen withMarchCount = Count
28 gen withMarchEarnings = WeeklyEarnings
29 replace Count=. if tin(2021m3,)
30 replace WeeklyEarnings=. if tin(2021m3,)
31
32 gen lnCount = ln(Count)
33 gen lnWeekHours = ln(WeekHours)
34 gen lnHourlyEarnings = ln(HourlyEarnings)
35 gen lnWeeklyEarnings = ln(WeeklyEarnings)
36 gen lnServiceCount = ln(ServiceCount)
37
38 gen m1=0
39 replace m1=1 if month==1
40 gen m2=0
41 replace m2=1 if month==2
42 gen m3=0
43 replace m3=1 if month==3
44 gen m4=0
45 replace m4=1 if month==4
46 gen m5=0
47 replace m5=1 if month==5
48 gen m6=0
49 replace m6=1 if month==6
50 gen m7=0
51 replace m7=1 if month==7
52 gen m8=0
53 replace m8=1 if month==8
54 gen m9=0
55 replace m9=1 if month==9
56 gen m10=0
57 replace m10=1 if month==10
58 gen m11=0
59 replace m11=1 if month==11
60 gen m12=0
61 replace m12=1 if month==12
62
63 gen dlnCount=d.lnCount
64 gen l1dlnCount=l1d.lnCount
65 gen l2dlnCount=l2d.lnCount
```

```
66 gen 13dlnCount=13d.lnCount
67 gen 14dlnCount=14d.lnCount
68 gen 15dlnCount=15d.lnCount
69 gen 16dlnCount=16d.lnCount
70 gen 17dlnCount=17d.lnCount
71 gen 18dlnCount=18d.lnCount
72 gen 19dlnCount=19d.lnCount
73 gen 110dlnCount=110d.lnCount
74 gen 111dlnCount=111d.lnCount
75 gen 112dlnCount=112d.lnCount
76 gen 124dlnCount=124d.lnCount
77 gen 136dlnCount=136d.lnCount
78 gen 148dlnCount=148d.lnCount
79
80 gen dlnWeekHours=d.lnWeekHours
81 gen 11dlnWeekHours=11d.lnWeekHours
82 gen 12dlnWeekHours=12d.lnWeekHours
83 gen 13dlnWeekHours=13d.lnWeekHours
84 gen 14dlnWeekHours=14d.lnWeekHours
85 gen 15dlnWeekHours=15d.lnWeekHours
86 gen 16dlnWeekHours=16d.lnWeekHours
87 gen 17dlnWeekHours=17d.lnWeekHours
88 gen 18dlnWeekHours=18d.lnWeekHours
89 gen 19dlnWeekHours=19d.lnWeekHours
90 gen 110dlnWeekHours=110d.lnWeekHours
91 gen 111dlnWeekHours=111d.lnWeekHours
92 gen 112dlnWeekHours=112d.lnWeekHours
93 gen 124dlnWeekHours=124d.lnWeekHours
94 gen 136dlnWeekHours=136d.lnWeekHours
95 gen 148dlnWeekHours=148d.lnWeekHours
96
97 gen dlnHourlyEarnings=d.lnHourlyEarnings
98 gen 11dlnHourlyEarnings=11d.lnHourlyEarnings
99 gen 12dlnHourlyEarnings=12d.lnHourlyEarnings
100 gen 13dlnHourlyEarnings=13d.lnHourlyEarnings
101 gen 14dlnHourlyEarnings=14d.lnHourlyEarnings
102 gen 15dlnHourlyEarnings=15d.lnHourlyEarnings
103 gen 16dlnHourlyEarnings=16d.lnHourlyEarnings
104 gen 17dlnHourlyEarnings=17d.lnHourlyEarnings
105 gen 18dlnHourlyEarnings=18d.lnHourlyEarnings
106 gen 19dlnHourlyEarnings=19d.lnHourlyEarnings
107 gen 110dlnHourlyEarnings=110d.lnHourlyEarnings
108 gen 111dlnHourlyEarnings=111d.lnHourlyEarnings
109 gen 112dlnHourlyEarnings=112d.lnHourlyEarnings
110 gen 124dlnHourlyEarnings=124d.lnHourlyEarnings
111 gen 136dlnHourlyEarnings=136d.lnHourlyEarnings
112 gen 148dlnHourlyEarnings=148d.lnHourlyEarnings
113
114 gen dlnWeeklyEarnings=d.lnWeeklyEarnings
115 gen 11dlnWeeklyEarnings=11d.lnWeeklyEarnings
116 gen 12dlnWeeklyEarnings=12d.lnWeeklyEarnings
```

```

117 gen 13dlnWeeklyEarnings=13d.lnWeeklyEarnings
118 gen 14dlnWeeklyEarnings=14d.lnWeeklyEarnings
119 gen 15dlnWeeklyEarnings=15d.lnWeeklyEarnings
120 gen 16dlnWeeklyEarnings=16d.lnWeeklyEarnings
121 gen 17dlnWeeklyEarnings=17d.lnWeeklyEarnings
122 gen 18dlnWeeklyEarnings=18d.lnWeeklyEarnings
123 gen 19dlnWeeklyEarnings=19d.lnWeeklyEarnings
124 gen 110dlnWeeklyEarnings=110d.lnWeeklyEarnings
125 gen 111dlnWeeklyEarnings=111d.lnWeeklyEarnings
126 gen 112dlnWeeklyEarnings=112d.lnWeeklyEarnings
127 gen 124dlnWeeklyEarnings=124d.lnWeeklyEarnings
128 gen 136dlnWeeklyEarnings=136d.lnWeeklyEarnings
129 gen 148dlnWeeklyEarnings=148d.lnWeeklyEarnings
130
131 gen dlnServiceCount=d.lnServiceCount
132 gen 11dlnServiceCount=11d.lnServiceCount
133 gen 12dlnServiceCount=12d.lnServiceCount
134 gen 13dlnServiceCount=13d.lnServiceCount
135 gen 14dlnServiceCount=14d.lnServiceCount
136 gen 15dlnServiceCount=15d.lnServiceCount
137 gen 16dlnServiceCount=16d.lnServiceCount
138 gen 17dlnServiceCount=17d.lnServiceCount
139 gen 18dlnServiceCount=18d.lnServiceCount
140 gen 19dlnServiceCount=19d.lnServiceCount
141 gen 110dlnServiceCount=110d.lnServiceCount
142 gen 111dlnServiceCount=111d.lnServiceCount
143 gen 112dlnServiceCount=112d.lnServiceCount
144 gen 124dlnServiceCount=124d.lnServiceCount
145 gen 136dlnServiceCount=136d.lnServiceCount
146 gen 148dlnServiceCount=148d.lnServiceCount
147
148 /*
149 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.
150 */
151 /* Count and WeeklyEarnings */
152
153 summ Count WeekHours HourlyEarnings WeeklyEarnings ServiceCount
154 summ lnCount lnWeekHours lnHourlyEarnings lnWeeklyEarnings lnServiceCount
155
156 ac lnCount, saving(lnCount_ac, replace)
157 pac lnCount, saving(lnCount_pac, replace)
158 graph combine lnCount_ac.gph lnCount_pac.gph, saving(lnCount_ac_pac, replace)
159 graph export "lnCount_ac_pac.png", replace
160 ** Probably need to difference
161
162 ac lnWeeklyEarnings, saving(lnWeeklyEarnings_ac, replace)
163 pac lnWeeklyEarnings, saving(lnWeeklyEarnings_pac, replace)
164 graph combine lnWeeklyEarnings_ac.gph lnWeeklyEarnings_pac.gph,
saving(lnWeeklyEarnings_ac_pac, replace)

```

```

165 graph export "lnWeeklyEarnings_ac_pac.png", replace
166 ** Probably need to difference
167
168 *starter models for count
169 *I used a pair plot to examine the rise and fall of variables with respect to each
other
170 reg d.lnCount l(12,24,36,48)d.lnCount // .01637
171 scalar drop _all
172 quietly forval w=12(12)144 {
173 gen pred=.
174 gen nobs=.
175 forval t=421/733 {
176 gen wstart=`t'-'`w'
177 gen wend=`t'-1
178 reg dlnCount 112dlnCount 124dlnCount 136dlnCount 148dlnCount ///
179 if Date>=wstart & Date<=wend
180 replace nobs=e(N) if Date==`t'
181 predict ptemp
182 replace pred=ptemp if Date==`t'
183 drop ptemp wstart wend
184 }
185 gen errsq=(pred-d.lnCount)^2
186 summ errsq
187 scalar RWrmse`w'=r(mean)^.5
188 summ nobs
189 scalar RWminobs`w'=r(min)
190 scalar RWmaxobs`w'=r(max)
191 drop errsq pred nobs
192 }
193 scalar list
194 /*
195 RWmaxobs132 = 132
196 RWminobs132 = 12
197 RWrmse132 = .0172128
198 */
199
200 reg d.lnCount l(5,12,24,36,48)d.lnCount l(5)d.lnWeekHours m5 // .01711
201 scalar drop _all
202 quietly forval w=12(12)84 {
203 gen pred=.
204 gen nobs=.
205 forval t=641/733 {
206 gen wstart=`t'-'`w'
207 gen wend=`t'-1
208 reg dlnCount 15dlnCount 112dlnCount 124dlnCount 136dlnCount 148dlnCount
15dlnWeekHours m5 ///
209 if Date>=wstart & Date<=wend
210 replace nobs=e(N) if Date==`t'
211 predict ptemp
212 replace pred=ptemp if Date==`t'
213 drop ptemp wstart wend

```

```

214      }
215      gen errsq=(pred-d.lnCount)^2
216      summ errsq
217      scalar RWrmse`w'=r(mean)^.5
218      summ nobs
219      scalar RWminobs`w'=r(min)
220      scalar RWmaxobs`w'=r(max)
221      drop errsq pred nobs
222    }
223    scalar list
224  /*
225  RWmaxobs84 =          84
226  RWminobs84 =          23
227  RWrmse84 = .01950911
228 */
229
230 quietly gsreg dlnCount 11dlnCount 12dlnCount 13dlnCount 14dlnCount 15dlnCount
231 16dlnCount ///
232 17dlnCount 18dlnCount 19dlnCount 110dlnCount 111dlnCount 112dlnCount ///
233 124dlnCount 136dlnCount 148dlnCount ///
234 if tin(1990m1,2021m1), ///
235 ncomb(1,12) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12) ///
236 samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnCount) replace
237 *gsreg suggestions
238 reg d.lnCount 112d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
239 scalar drop _all
240 quietly forval w=12(12)144 {
241 gen pred=.
242 gen nobs=.
243 forval t=385/733 {
244 gen wstart=`t'-`w'
245 gen wend=`t'-1
246 reg dlnCount 112dlnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
247 if Date>=wstart & Date<=wend
248 replace nobs=e(N) if Date==`t'
249 predict ptemp
250 replace pred=ptemp if Date==`t'
251 drop ptemp wstart wend
252 }
253 gen errsq=(pred-d.lnCount)^2
254 summ errsq
255 scalar RWrmse`w'=r(mean)^.5
256 summ nobs
257 scalar RWminobs`w'=r(min)
258 scalar RWmaxobs`w'=r(max)
259 drop errsq pred nobs
260 }
261 scalar list
262 /*
263 RWmaxobs144 =          144

```

```

264 RWminobs144 =          12
265 RWrmse144 =   .01824906
266 */
267
268 reg d.lnCount 1(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
269 scalar drop _all
270 quietly forval w=12(12)144 {
271 gen pred=.
272 gen nobs=.
273     forval t=409/733 {
274         gen wstart=`t'-'`w'
275         gen wend=`t'-1
276         reg dlnCount 112dlnCount 136dlnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
277             if Date>=wstart & Date<=wend
278         replace nobs=e(N) if Date==`t'
279         predict ptemp
280         replace pred=ptemp if Date==`t'
281         drop ptemp wstart wend
282     }
283     gen errsq=(pred-d.lnCount)^2
284     summ errsq
285     scalar RWrmse`w'=r(mean)^.5
286     summ nobs
287     scalar RWminobs`w'=r(min)
288     scalar RWmaxobs`w'=r(max)
289     drop errsq pred nobs
290 }
291 scalar list
292 /*
293 RWmaxobs144 =          144
294 RWminobs144 =          12
295 RWrmse144 =   .01777071
296 */
297
298 reg d.lnCount 14d.lnWeekHours 19d.lnWeekHours 18d.lnHourlyEarnings m1 m2 m3 m4 m5 m6
m7 m8 m9 m10 m11
299 scalar drop _all
300 quietly forval w=12(12)84 {
301 gen pred=.
302 gen nobs=.
303     forval t=624/733 {
304         gen wstart=`t'-'`w'
305         gen wend=`t'-1
306         reg dlnCount 14dlnWeekHours 19dlnWeekHours 18dlnHourlyEarnings m1 m2 m3 m4 m5 m6
m7 m8 m9 m10 m11 ///
307             if Date>=wstart & Date<=wend
308         replace nobs=e(N) if Date==`t'
309         predict ptemp
310         replace pred=ptemp if Date==`t'
311         drop ptemp wstart wend
312     }

```

```

313 gen errsq=(pred-d.lnCount)^2
314 summ errsq
315 scalar RWrmse`w'=r(mean)^.5
316 summ nobs
317 scalar RWminobs`w'=r(min)
318 scalar RWmaxobs`w'=r(max)
319 drop errsq pred nobs
320 }
321 scalar list
322 /*
323 RWmaxobs12 = 12
324 RWminobs12 = 2
325 RWrmse12 = .0176238
326 */
327
328 scalar rwmse = .0172128
329 reg d.lnCount 1(12,24,36,48)d.lnCount if tin(,2021m2)
330 predict pd
331 gen pflcount=exp((rwmse^2)/2)*exp(l.lnCount+pd) if Date==tm(2021m3)
332 gen ubl=exp((rwmse^2)/2)*exp(l.lnCount+pd+1*rwmse) if Date==tm(2021m3)
333 gen lb1=exp((rwmse^2)/2)*exp(l.lnCount+pd-1*rwmse) if Date==tm(2021m3)
334 gen ub2=exp((rwmse^2)/2)*exp(l.lnCount+pd+2*rwmse) if Date==tm(2021m3)
335 gen lb2=exp((rwmse^2)/2)*exp(l.lnCount+pd-2*rwmse) if Date==tm(2021m3)
336 gen ub3=exp((rwmse^2)/2)*exp(l.lnCount+pd+3*rwmse) if Date==tm(2021m3)
337 gen lb3=exp((rwmse^2)/2)*exp(l.lnCount+pd-3*rwmse) if Date==tm(2021m3)
338 drop pd
339
340 replace pflcount=Count if Date==tm(2021m2)
341 replace ub1=Count if Date==tm(2021m2)
342 replace ub2=Count if Date==tm(2021m2)
343 replace ub3=Count if Date==tm(2021m2)
344 replace lb1=Count if Date==tm(2021m2)
345 replace lb2=Count if Date==tm(2021m2)
346 replace lb3=Count if Date==tm(2021m2)
347
348 twoway (tsrline ub3 ub2 if tin(2020m3,2021m3), ///
349   recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
350   (tsrline ub2 ub1 if tin(2020m3,2021m3), ///
351   recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
352   (tsrline ub1 pflcount if tin(2020m3,2021m3), ///
353   recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
354   (tsrline pflcount lb1 if tin(2020m3,2021m3), ///
355   recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
356   (tsrline lb1 lb2 if tin(2020m3,2021m3), ///
357   recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
358   (tsrline lb2 lb3 if tin(2020m3,2021m3), ///
359   recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
360   (tsline Count pflcount if tin(2020m3,2021m3) , ///
361   lcolor(gs12 teal) lwidth(medthick medthick) ///
362   lpattern(solid longdash)) ///
363   (scatter withMarchCount Date if tin(2021m3,)), scheme(slmono) legend(off)

```

```

364 graph export "CountFan.png", replace
365
366 /*-----*/
367
368 *starter models for weekly earnings
369 reg d.lnWeeklyEarnings l1d.lnWeekHours ld.lnHourlyEarnings
370 scalar drop _all
371 quietly forval w=12(12)84 {
372 gen pred=.
373 gen nobs=.
374 forval t=616/733 {
375 gen wstart=`t'-'w'
376 gen wend=`t'-1
377 reg dlnWeeklyEarnings l1dlnWeekHours l1dlnHourlyEarnings ///
378 if Date>=wstart & Date<=wend
379 replace nobs=e(N) if Date==`t'
380 predict ptemp
381 replace pred=ptemp if Date==`t'
382 drop ptemp wstart wend
383 }
384 gen errsq=(pred-d.lnWeeklyEarnings)^2
385 summ errsq
386 scalar RWrmse`w'=r(mean)^.5
387 summ nobs
388 scalar RWminobs`w'=r(min)
389 scalar RWmaxobs`w'=r(max)
390 drop errsq pred nobs
391 }
392 scalar list
393 /*
394 RWmaxobs60 = 60
395 RWminobs60 = 2
396 RWrmse60 = .06145693
397 */
398
399 quietly gsreg dlnWeeklyEarnings l1dlnWeeklyEarnings l2dlnWeeklyEarnings
400 13dlnWeeklyEarnings ///
401 14dlnWeeklyEarnings 15dlnWeeklyEarnings 16dlnWeeklyEarnings ///
402 17dlnWeeklyEarnings 18dlnWeeklyEarnings 19dlnWeeklyEarnings 110dlnWeeklyEarnings
403 /**
404 111dlnWeeklyEarnings 112dlnWeeklyEarnings ///
405 124dlnWeeklyEarnings 136dlnWeeklyEarnings ///
406 if tin(2011m1,2021m1), ///
407 ncomb(1,12) aic outsample(24) ///
408 samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnWeeklyEarnings)
409 replace
410
411 reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings m1 m2 m3 m4 m5 m6
412 m7 m8 m9 m10 m11
413 scalar drop _all
414 quietly forval w=12(12)84 {

```

```

411 gen pred=.
412 gen nobs=.
413   forval t=620/733 {
414     gen wstart=`t'-'`w'
415     gen wend=`t'-1
416     reg dlnWeeklyEarnings 13dlnWeeklyEarnings 15dlnWeeklyEarnings m1 m2 m3 m4 m5 m6 m7
417       m8 m9 m10 m11 ///
418         if Date>=wstart & Date<=wend
419         replace nobs=e(N) if Date==`t'
420         predict ptemp
421         replace pred=ptemp if Date==`t'
422         drop ptemp wstart wend
423     }
424     gen errsq=(pred-d.lnWeeklyEarnings)^2
425     summ errsq
426     scalar RWrmse`w'=r(mean)^.5
427     summ nobs
428     scalar RWminobs`w'=r(min)
429     scalar RWmaxobs`w'=r(max)
430     drop errsq pred nobs
431   }
432   scalar list
433 /* 
434 RWmaxobs84 =          84
435 RWminobs84 =          2
436 RWrmse84 = .06004448
437 */
438 reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings
439 17d.lnWeeklyEarnings
440 scalar drop _all
441 quietly forval w=12(12)84 {
442   gen pred=.
443   gen nobs=.
444   forval t=622/733 {
445     gen wstart=`t'-'`w'
446     gen wend=`t'-1
447     reg dlnWeeklyEarnings 13dlnWeeklyEarnings 15dlnWeeklyEarnings 17dlnWeeklyEarnings
448     ///
449       if Date>=wstart & Date<=wend
450       replace nobs=e(N) if Date==`t'
451       predict ptemp
452       replace pred=ptemp if Date==`t'
453       drop ptemp wstart wend
454     }
455     gen errsq=(pred-d.lnWeeklyEarnings)^2
456     summ errsq
457     scalar RWrmse`w'=r(mean)^.5
458     scalar RWminobs`w'=r(min)
459     scalar RWmaxobs`w'=r(max)

```

```

459 drop errsq pred nobs
460 }
461 scalar list
462 /*
463 RWmaxobs84 = 84
464 RWminobs84 = 2
465 RWrmse84 = .05250414
466 */
467
468 drop pflcount ub1 ub2 ub3 lb1 lb2 lb3
469
470 scalar rwmse = .05250414
471 reg d.lnWeeklyEarnings l(3,5,7)d.lnWeeklyEarnings if tin(,2021m2)
472 predict pd
473 gen pflcount=exp((rwmse^2)/2)*exp(1.lnWeeklyEarnings+pd) if Date==tm(2021m3)
474 gen ub1=exp((rwmse^2)/2)*exp(1.lnWeeklyEarnings+pd+1*rwmse) if Date==tm(2021m3)
475 gen lb1=exp((rwmse^2)/2)*exp(1.lnWeeklyEarnings+pd-1*rwmse) if Date==tm(2021m3)
476 gen ub2=exp((rwmse^2)/2)*exp(1.lnWeeklyEarnings+pd+2*rwmse) if Date==tm(2021m3)
477 gen lb2=exp((rwmse^2)/2)*exp(1.lnWeeklyEarnings+pd-2*rwmse) if Date==tm(2021m3)
478 gen ub3=exp((rwmse^2)/2)*exp(1.lnWeeklyEarnings+pd+3*rwmse) if Date==tm(2021m3)
479 gen lb3=exp((rwmse^2)/2)*exp(1.lnWeeklyEarnings+pd-3*rwmse) if Date==tm(2021m3)
480 drop pd
481
482 replace pflcount=WeeklyEarnings if Date==tm(2021m2)
483 replace ub1=WeeklyEarnings if Date==tm(2021m2)
484 replace ub2=WeeklyEarnings if Date==tm(2021m2)
485 replace ub3=WeeklyEarnings if Date==tm(2021m2)
486 replace lb1=WeeklyEarnings if Date==tm(2021m2)
487 replace lb2=WeeklyEarnings if Date==tm(2021m2)
488 replace lb3=WeeklyEarnings if Date==tm(2021m2)
489
490 twoway (tsrline ub3 ub2 if tin(2020m3,2021m3), ///
491     recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
492     (tsrline ub2 ub1 if tin(2020m3,2021m3), ///
493     recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
494     (tsrline ub1 pflcount if tin(2020m3,2021m3), ///
495     recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
496     (tsrline pflcount lb1 if tin(2020m3,2021m3), ///
497     recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
498     (tsrline lb1 lb2 if tin(2020m3,2021m3), ///
499     recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
500     (tsrline lb2 lb3 if tin(2020m3,2021m3), ///
501     recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
502     (tsline WeeklyEarnings pflcount if tin(2020m3,2021m3) , ///
503     lcolor(gs12 teal) lwidth(medthick medthick) ///
504     lpattern(solid longdash)) ///
505     (scatter withMarchEarnings Date if tin(2021m3,,), scheme(slmono) legend(off)
506 graph export "WeeklyFan.png", replace
507
508 log close
509 translate "Final Project.smcl" "Final Project.txt", replace

```

Appendix B

```

1      ____(R)          _____
2                               /__   /____/   /____/
3                               __/   /   /__/   /   /__/
4                                         Statistics/Data analysis
5
6 -----
7 --
8           name: <unnamed>
9           log: /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
10          Series/Probl
11          > em Sets/Final Project/Final Project.smcl
12          log type: smcl
13          opened on: 19 Apr 2021, 20:43:14
14
15
16          1 . import delimited "TS2020_Final_Project_Monthly.txt"
17          (6 vars, 375 obs)
18
19          2 . rename smu12455400500000001 Count
20
21          3 . rename smu12455400500000002 WeekHours
22
23          4 . rename smu12455400500000003 HourlyEarnings
24
25          5 . rename smu12455400500000011 WeeklyEarnings
26
27          6 . rename smu12455400800000001 ServiceCount
28
29          7 .
30          8 .
31          9 . label variable Count "Count"
32
33          10 . label variable WeekHours "WeekHours"
34
35          11 . label variable HourlyEarnings "HourlyEarnings"
36
37          12 . label variable WeeklyEarnings "WeeklyEarnings"
38
39          13 . label variable ServiceCount "ServiceCount"
40
41          14 .
42          15 .

```

```
40 . gen datec=date(date, "YMD")
41
42 . gen Date=mofd(datec)
43
44 . gen month=month(datec)
45
46 . format Date %tm
47
48 . tsset Date
        time variable: Date, 1990m1 to 2021m3
        delta: 1 month
49
50
51 .
52 . gen withMarchCount = Count
53
54
55 . gen withMarchEarnings = WeeklyEarnings
56     (252 missing values generated)
57
58 . replace Count=. if tin(2021m3,)
59     (1 real change made, 1 to missing)
60
61 . replace WeeklyEarnings=. if tin(2021m3,)
62     (1 real change made, 1 to missing)
63
64 .
65 . gen lnCount = ln(Count)
66     (1 missing value generated)
67
68 . gen lnWeekHours = ln(WeekHours)
69     (252 missing values generated)
70
71 . gen lnHourlyEarnings = ln(HourlyEarnings)
72     (252 missing values generated)
73
74 . gen lnWeeklyEarnings = ln(WeeklyEarnings)
75     (253 missing values generated)
76
77 . gen lnServiceCount = ln(ServiceCount)
78
79 .
80 . gen m1=0
81
82 . replace m1=1 if month==1
83     (32 real changes made)
84
85 . gen m2=0
86
87 . replace m2=1 if month==2
88     (32 real changes made)
89
90 . gen m3=0
```

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

```
142      (2 missing values generated)
143
144 59 . gen l1dlnCount=l1d.lnCount
145      (2 missing values generated)
146
147 60 . gen l2dlnCount=l2d.lnCount
148      (3 missing values generated)
149
150 61 . gen l3dlnCount=l3d.lnCount
151      (4 missing values generated)
152
153 62 . gen l4dlnCount=l4d.lnCount
154      (5 missing values generated)
155
156 63 . gen l5dlnCount=l5d.lnCount
157      (6 missing values generated)
158
159 64 . gen l6dlnCount=l6d.lnCount
160      (7 missing values generated)
161
162 65 . gen l7dlnCount=l7d.lnCount
163      (8 missing values generated)
164
165 66 . gen l8dlnCount=l8d.lnCount
166      (9 missing values generated)
167
168 67 . gen l9dlnCount=l9d.lnCount
169      (10 missing values generated)
170
171 68 . gen l10dlnCount=l10d.lnCount
172      (11 missing values generated)
173
174 69 . gen l11dlnCount=l11d.lnCount
175      (12 missing values generated)
176
177 70 . gen l12dlnCount=l12d.lnCount
178      (13 missing values generated)
179
180 71 . gen l24dlnCount=l24d.lnCount
181      (25 missing values generated)
182
183 72 . gen l36dlnCount=l36d.lnCount
184      (37 missing values generated)
185
186 73 . gen l48dlnCount=l48d.lnCount
187      (49 missing values generated)
188
189 74 .
190 75 . gen dlnWeekHours=d.lnWeekHours
191      (253 missing values generated)
192
```

```
193 . gen l1dlnWeekHours=l1d.lnWeekHours  
194     (254 missing values generated)  
195  
196 . gen l2dlnWeekHours=l2d.lnWeekHours  
197     (255 missing values generated)  
198  
199 . gen l3dlnWeekHours=l3d.lnWeekHours  
200     (256 missing values generated)  
201  
202 . gen l4dlnWeekHours=l4d.lnWeekHours  
203     (257 missing values generated)  
204  
205 . gen l5dlnWeekHours=l5d.lnWeekHours  
206     (258 missing values generated)  
207  
208 . gen l6dlnWeekHours=l6d.lnWeekHours  
209     (259 missing values generated)  
210  
211 . gen l7dlnWeekHours=l7d.lnWeekHours  
212     (260 missing values generated)  
213  
214 . gen l8dlnWeekHours=l8d.lnWeekHours  
215     (261 missing values generated)  
216  
217 . gen l9dlnWeekHours=l9d.lnWeekHours  
218     (262 missing values generated)  
219  
220 . gen l10dlnWeekHours=l10d.lnWeekHours  
221     (263 missing values generated)  
222  
223 . gen l11dlnWeekHours=l11d.lnWeekHours  
224     (264 missing values generated)  
225  
226 . gen l12dlnWeekHours=l12d.lnWeekHours  
227     (265 missing values generated)  
228  
229 . gen l124dlnWeekHours=l124d.lnWeekHours  
230     (277 missing values generated)  
231  
232 . gen l136dlnWeekHours=l136d.lnWeekHours  
233     (289 missing values generated)  
234  
235 . gen l148dlnWeekHours=l148d.lnWeekHours  
236     (301 missing values generated)  
237  
238 .  
239 . gen dlnHourlyEarnings=d.lnHourlyEarnings  
240     (253 missing values generated)  
241  
242 . gen l1dlnHourlyEarnings=l1d.lnHourlyEarnings  
243     (254 missing values generated)
```

```
244  
245     94 . gen 12dlnHourlyEarnings=12d.lnHourlyEarnings  
246         (255 missing values generated)  
247  
248     95 . gen 13dlnHourlyEarnings=13d.lnHourlyEarnings  
249         (256 missing values generated)  
250  
251     96 . gen 14dlnHourlyEarnings=14d.lnHourlyEarnings  
252         (257 missing values generated)  
253  
254     97 . gen 15dlnHourlyEarnings=15d.lnHourlyEarnings  
255         (258 missing values generated)  
256  
257     98 . gen 16dlnHourlyEarnings=16d.lnHourlyEarnings  
258         (259 missing values generated)  
259  
260     99 . gen 17dlnHourlyEarnings=17d.lnHourlyEarnings  
261         (260 missing values generated)  
262  
263     100 . gen 18dlnHourlyEarnings=18d.lnHourlyEarnings  
264         (261 missing values generated)  
265  
266     101 . gen 19dlnHourlyEarnings=19d.lnHourlyEarnings  
267         (262 missing values generated)  
268  
269     102 . gen 110dlnHourlyEarnings=110d.lnHourlyEarnings  
270         (263 missing values generated)  
271  
272     103 . gen 111dlnHourlyEarnings=111d.lnHourlyEarnings  
273         (264 missing values generated)  
274  
275     104 . gen 112dlnHourlyEarnings=112d.lnHourlyEarnings  
276         (265 missing values generated)  
277  
278     105 . gen 124dlnHourlyEarnings=124d.lnHourlyEarnings  
279         (277 missing values generated)  
280  
281     106 . gen 136dlnHourlyEarnings=136d.lnHourlyEarnings  
282         (289 missing values generated)  
283  
284     107 . gen 148dlnHourlyEarnings=148d.lnHourlyEarnings  
285         (301 missing values generated)  
286  
287     108 .  
288     109 . gen dlnWeeklyEarnings=d.lnWeeklyEarnings  
289         (254 missing values generated)  
290  
291     110 . gen 11dlnWeeklyEarnings=11d.lnWeeklyEarnings  
292         (254 missing values generated)  
293  
294     111 . gen 12dlnWeeklyEarnings=12d.lnWeeklyEarnings
```

```
295      (255 missing values generated)
296
297 112 . gen 13dlnWeeklyEarnings=13d.lnWeeklyEarnings
298      (256 missing values generated)
299
300 113 . gen 14dlnWeeklyEarnings=14d.lnWeeklyEarnings
301      (257 missing values generated)
302
303 114 . gen 15dlnWeeklyEarnings=15d.lnWeeklyEarnings
304      (258 missing values generated)
305
306 115 . gen 16dlnWeeklyEarnings=16d.lnWeeklyEarnings
307      (259 missing values generated)
308
309 116 . gen 17dlnWeeklyEarnings=17d.lnWeeklyEarnings
310      (260 missing values generated)
311
312 117 . gen 18dlnWeeklyEarnings=18d.lnWeeklyEarnings
313      (261 missing values generated)
314
315 118 . gen 19dlnWeeklyEarnings=19d.lnWeeklyEarnings
316      (262 missing values generated)
317
318 119 . gen 110dlnWeeklyEarnings=110d.lnWeeklyEarnings
319      (263 missing values generated)
320
321 120 . gen 111dlnWeeklyEarnings=111d.lnWeeklyEarnings
322      (264 missing values generated)
323
324 121 . gen 112dlnWeeklyEarnings=112d.lnWeeklyEarnings
325      (265 missing values generated)
326
327 122 . gen 124dlnWeeklyEarnings=124d.lnWeeklyEarnings
328      (277 missing values generated)
329
330 123 . gen 136dlnWeeklyEarnings=136d.lnWeeklyEarnings
331      (289 missing values generated)
332
333 124 . gen 148dlnWeeklyEarnings=148d.lnWeeklyEarnings
334      (301 missing values generated)
335
336 125 .
337 126 . gen dlnServiceCount=d.lnServiceCount
338      (1 missing value generated)
339
340 127 . gen 11dlnServiceCount=11d.lnServiceCount
341      (2 missing values generated)
342
343 128 . gen 12dlnServiceCount=12d.lnServiceCount
344      (3 missing values generated)
345
```

```
346 129 . gen 13dlnServiceCount=13d.lnServiceCount
347     (4 missing values generated)
348
349 130 . gen 14dlnServiceCount=14d.lnServiceCount
350     (5 missing values generated)
351
352 131 . gen 15dlnServiceCount=15d.lnServiceCount
353     (6 missing values generated)
354
355 132 . gen 16dlnServiceCount=16d.lnServiceCount
356     (7 missing values generated)
357
358 133 . gen 17dlnServiceCount=17d.lnServiceCount
359     (8 missing values generated)
360
361 134 . gen 18dlnServiceCount=18d.lnServiceCount
362     (9 missing values generated)
363
364 135 . gen 19dlnServiceCount=19d.lnServiceCount
365     (10 missing values generated)
366
367 136 . gen 110dlnServiceCount=110d.lnServiceCount
368     (11 missing values generated)
369
370 137 . gen 111dlnServiceCount=111d.lnServiceCount
371     (12 missing values generated)
372
373 138 . gen 112dlnServiceCount=112d.lnServiceCount
374     (13 missing values generated)
375
376 139 . gen 124dlnServiceCount=124d.lnServiceCount
377     (25 missing values generated)
378
379 140 . gen 136dlnServiceCount=136d.lnServiceCount
380     (37 missing values generated)
381
382 141 . gen 148dlnServiceCount=148d.lnServiceCount
383     (49 missing values generated)
384
385 142 .
386 143 . /*
387      > The project is to forecast the March non-seasonally adjusted estimates of
ave
388      > rage weekly earnings and total employment for private employers (total
privat
389      > e) for a Florida MSA of your choice and write up a professional report on
you
390      > r forecast.
391      > */
392 144 . /* Count and WeeklyEarnings */
393 145 .
```

```

394 146 . summ Count WeekHours HourlyEarnings WeeklyEarnings ServiceCount
395
396      Variable |       Obs        Mean     Std. Dev.      Min      Max
397      -----+
398          Count |      374    14.18556    6.880684      5.3      28
399          WeekHours |      123   36.88455    3.791817    28.3    45.8
400          HourlyEarn~s |      123      19.72    2.903968    15.01    24.6
401          WeeklyEarn~s |      122   719.6542   84.57241    503.79   916.1
402          ServiceCount |      375   10.43387   5.959179      3.9    22.8
403
404 147 . summ lnCount lnWeekHours lnHourlyEarnings lnWeeklyEarnings lnServiceCount
405
406      Variable |       Obs        Mean     Std. Dev.      Min      Max
407      -----+
408          lnCount |      374     2.5174    .5398403    1.667707  3.332205
409          lnWeekHours |      123   3.602488    .10385    3.342862  3.824284
410          lnHourlyEa~s |      123   2.970779    .1482819    2.708717  3.202746
411          lnWeeklyEa~s |      122   6.571775    .1195694    6.222159  6.820126
412          lnServiceC~t |      375   2.172053    .5985689    1.360977  3.12676
413
414 148 .
415 149 . ac lnCount, saving(lnCount_ac, replace)
416     (file lnCount_ac.gph saved)
417
418 150 . pac lnCount, saving(lnCount_pac, replace)
419     (file lnCount_pac.gph saved)
420
421 151 . graph combine lnCount_ac.gph lnCount_pac.gph, saving(lnCount_ac_pac,
422 replace)
423     (file lnCount_ac_pac.gph saved)
424
425 152 . graph export "lnCount_ac_pac.png", replace
426     (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
427 Sets
428     > /Final Project/lnCount_ac_pac.png written in PNG format)
429
430 153 . ** Probably need to difference
431
432 154 .
433 155 . ac lnWeeklyEarnings, saving(lnWeeklyEarnings_ac, replace)
434     (file lnWeeklyEarnings_ac.gph saved)
435
436 156 . pac lnWeeklyEarnings, saving(lnWeeklyEarnings_pac, replace)
437     (file lnWeeklyEarnings_pac.gph saved)
438
439 157 . graph combine lnWeeklyEarnings_ac.gph lnWeeklyEarnings_pac.gph,
440 saving(lnWeek
441     > lyEarnings_ac_pac, replace)
442     (file lnWeeklyEarnings_ac_pac.gph saved)
443
444 158 . graph export "lnWeeklyEarnings_ac_pac.png", replace

```

```

441      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
442 Sets
443
444      159 . ** Probably need to difference
445      160 .
446      161 . *starter models for count
447      162 . *I used a pair plot to examine the rise and fall of variables with respect
448 to
449      > each other
450
451      163 . reg d.lnCount l(12,24,36,48)d.lnCount // .01637
452
453      Source |      SS          df          MS      Number of obs   =
454      325
455      -----+----- F(4, 320)   =
456      16.54
457      Model |  .017539188      4  .004384797  Prob > F   =
458      0.0000
459      Residual |  .084856979    320  .000265178  R-squared   =
460      0.1713
461      -----+----- Adj R-squared   =
462      0.1609
463      Total |  .102396167    324  .000316038  Root MSE    =
464      .01628
465
466      -----
467      D.lnCount |      Coef.      Std. Err.        t      P>|t|      [95% Conf.
468 Interval]
469      -----
470      lnCount |
471      L12D. |  .3609966  .0621085      5.81  0.000  .238804
472      .4831893
473      L24D. |  .137848  .0617615      2.23  0.026  .016338
474      .259358
475      L36D. |  -.0160136  .0614584     -0.26  0.795  -.1369272
476      .1049
477      L48D. |  .1265117  .0585322      2.16  0.031  .0113551
478      .2416683
479      |
480      _cons |  .0017116  .0009853      1.74  0.083  -.0002269
481      .0036502
482
483      -----
484
485      164 . scalar drop _all
486
487      165 . quietly forval w=12(12)144 {
488
489      166 . scalar list

```

```

475      RWmaxobs144 =      144
476      RWminobs144 =       12
477      RWrmse144 =   .0172276
478      RWmaxobs132 =      132
479      RWminobs132 =       12
480      RWrmse132 =   .0172128
481      RWmaxobs120 =      120
482      RWminobs120 =       12
483      RWrmse120 =   .01721825
484      RWmaxobs108 =      108
485      RWminobs108 =       12
486      RWrmse108 =   .01723674
487      RWmaxobs96 =       96
488      RWminobs96 =       12
489      RWrmse96 =   .01722006
490      RWmaxobs84 =      84
491      RWminobs84 =       12
492      RWrmse84 =   .01726063
493      RWmaxobs72 =      72
494      RWminobs72 =       12
495      RWrmse72 =   .01722377
496      RWmaxobs60 =      60
497      RWminobs60 =       12
498      RWrmse60 =   .01734443
499      RWmaxobs48 =      48
500      RWminobs48 =       12
501      RWrmse48 =   .01755803
502      RWmaxobs36 =      36
503      RWminobs36 =       12
504      RWrmse36 =   .01805924
505      RWmaxobs24 =      24
506      RWminobs24 =       12
507      RWrmse24 =   .0185871
508      RWmaxobs12 =      12
509      RWminobs12 =       12
510      RWrmse12 =   .02320505
511
512 167 . /*
513 > RWmaxobs132 =      132
514 > RWminobs132 =       12
515 > RWrmse132 =   .0172128
516 > */
517 168 .
518 . reg d.lnCount 1(5,12,24,36,48)d.lnCount 1(5)d.lnWeekHours m5 // .01711
519
520      Source |      SS          df          MS      Number of obs     =
521      -----+----- F(7, 108)      =
5.94
522      Model |  .012171566      7  .001738795  Prob > F      =
0.0000

```

```

523      Residual |   .03162877      108  .000292859  R-squared      =
524      0.2779
525      -----
526      Total |   .043800336      115  .000380872  Root MSE       =
527      .01711
528      -----
529      D.lnCount |     Coef.    Std. Err.      t    P>|t|    [95% Conf.
530      Interval]
531      -----
532      lnCount |
533      L5D. |  -.1231921  .0845717  -1.46  0.148  -.290828
534      .0444438
535      L12D. |   .5811114  .1685831   3.45  0.001  .2469504
536      .9152724
537      L24D. |  -.1196017  .1627467  -0.73  0.464  -.4421938
538      .2029904
539      L36D. |   .2532303  .1742525   1.45  0.149  -.0921684
540      .5986291
541      L48D. |   .1341638  .1858633   0.72  0.472  -.2342495
542      .5025771
543      |
544      lnWeekHours |
545      L5D. |   .0170123  .0364906   0.47  0.642  -.0553184
546      .089343
547      |
548      m5 |   .0067588  .0061605   1.10  0.275  -.0054524
549      .0189699
550      _cons |   .0004279  .0018229   0.23  0.815  -.0031854
551      .0040412
552      -----
553
554      170 . scalar drop _all
555
556      171 . quietly forval w=12(12)84 {
557
558      172 . scalar list
559      RWmaxobs84 =      84
560      RWminobs84 =      23
561      RWrmse84 =   .01950911
562      RWmaxobs72 =      72
563      RWminobs72 =      23
564      RWrmse72 =   .01949719
565      RWmaxobs60 =      60
566      RWminobs60 =      23
567      RWrmse60 =   .0199438
568      RWmaxobs48 =      48

```

```

559      RWminobs48 =      23
560      RWrmse48 = .02035982
561      RWmaxobs36 =      36
562      RWminobs36 =      23
563      RWrmse36 = .02138785
564      RWmaxobs24 =      24
565      RWminobs24 =      23
566      RWrmse24 = .02268585
567      RWmaxobs12 =      12
568      RWminobs12 =      12
569      RWrmse12 = .05004898
570
571 173 . /*
572 > RWmaxobs84 =      84
573 > RWminobs84 =      23
574 > RWrmse84 = .01950911
575 > */
576 174 .
577 175 . quietly gsreg dlnCount 11dlnCount 12dlnCount 13dlnCount 14dlnCount
15dlnCount
578      > 16dlnCount ///
579      >      17dlnCount 18dlnCount 19dlnCount 110dlnCount 111dlnCount
112dlnCount
580      > ///
581      >      124dlnCount 136dlnCount 148dlnCount ///
582      >      if tin(1990m1,2021m1), ///
583      >      ncomb(1,12) aic outsamp(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
m11
584      > m12) ///
585      >      samesample nindex( -1 aic -1 bic -1 rmse_out)
results(gsreg_dlnCount)
586      > replace
587
588 176 .
589 177 . *gsreg suggestions
590 178 . reg d.lnCount 112d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
591
592      Source |      SS          df          MS      Number of obs     =
361      -----+----- F(12, 348)     =
593      Model | .022616974      12  .001884748  Prob > F     =
0.0000
594      Residual | .094871179      348  .000272618  R-squared     =
0.1925
595      -----+----- Adj R-squared     =
0.1647
596      Total | .117488153      360  .000326356  Root MSE     =
.01651
597
598

```

```

599      -----
600      D.lnCount |      Coef.    Std. Err.      t     P>|t|      [95% Conf.
601      Interval]
602      -----
603      lnCount |      L12D. | .1748571   .0594184   2.94   0.003   .0579928
604      |      m1 | -.0099477   .0043004  -2.31   0.021  -.0184056
605      |      m2 | .0009939   .0042297   0.23   0.814  -.0073251
606      |      m3 | .0030247   .0042759   0.71   0.480  -.0053851
607      |      m4 | -.0071933   .0042648  -1.69   0.093  -.0155814
608      |      m5 | -.0098194   .0043178  -2.27   0.024  -.0183118
609      |      m6 | -.0133285   .0043874  -3.04   0.003  -.0219576
610      |      m7 | -.0091828   .0042967  -2.14   0.033  -.0176336
611      |      m8 | -.0017998   .0042632  -0.42   0.673  -.0101846
612      |      m9 | -.006737   .0042824  -1.57   0.117  -.0151597
613      |      m10 | .0062149   .0042795   1.45   0.147  -.0022021
614      |      m11 | .0042124   .0042811   0.98   0.326  -.0042078
615      |      _cons | .0072199   .0030452   2.37   0.018   .0012306
616      -----
617      -----
618      -
619      179 . scalar drop _all
620
621      180 . quietly forval w=12(12)144 {
622
623      181 . scalar list
624          RWmaxobs144 =      144
625          RWminobs144 =      12
626          RWrmse144 = .01824906
627          RWmaxobs132 =      132
628          RWminobs132 =      12
629          RWrmse132 = .01832173
630          RWmaxobs120 =      120
631          RWminobs120 =      12
632          RWrmse120 = .01833557

```

```

633      RWmaxobs108 =      108
634      RWminobs108 =       12
635      RWrmse108 = .01841089
636      RWmaxobs96 =       96
637      RWminobs96 =       12
638      RWrmse96 = .01836974
639      RWmaxobs84 =       84
640      RWminobs84 =       12
641      RWrmse84 = .01849267
642      RWmaxobs72 =       72
643      RWminobs72 =       12
644      RWrmse72 = .01861349
645      RWmaxobs60 =       60
646      RWminobs60 =       12
647      RWrmse60 = .01911515
648      RWmaxobs48 =       48
649      RWminobs48 =       12
650      RWrmse48 = .01922268
651      RWmaxobs36 =       36
652      RWminobs36 =       12
653      RWrmse36 = .01991683
654      RWmaxobs24 =       24
655      RWminobs24 =       12
656      RWrmse24 = .02022186
657      RWmaxobs12 =       12
658      RWminobs12 =       12
659      RWrmse12 = .02009249
660
661 182 . /*
662 > RWmaxobs144 =      144
663 > RWminobs144 =       12
664 > RWrmse144 = .01824906
665 > */
666 183 .
667 184 . reg d.lnCount l(12,36)d.lnCount m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
668
669      Source |      SS          df          MS      Number of obs     =
337
670      -----+----- F(13, 323)     =
5.68
671      Model | .019946057      13  .001534312  Prob > F     =
0.0000
672      Residual | .087185203     323  .000269923  R-squared     =
0.1862
673      -----+----- Adj R-squared     =
0.1534
674      Total | .107131259     336  .000318843  Root MSE     =
.01643
675
676      -----

```

	D.lnCount		Coef.	Std. Err.	t	P> t	[95% Conf.
677	Interval]						
678		-----+-----					
679		-					
680	lnCount						
	L12D.		.1849403	.0636401	2.91	0.004	.059739
681	.3101417						
	L36D.		-.049332	.0606582	-0.81	0.417	-.1686671
682	.0700031						
683							
	m1		-.0073418	.0044769	-1.64	0.102	-.0161493
684	.0014658						
	m2		.0022711	.0043559	0.52	0.602	-.0062984
685	.0108407						
	m3		.0043593	.004416	0.99	0.324	-.0043285
686	.0130471						
	m4		-.0065438	.0043922	-1.49	0.137	-.0151847
687	.002097						
	m5		-.0089215	.0045194	-1.97	0.049	-.0178126
688	-.0000304						
	m6		-.0133453	.0046241	-2.89	0.004	-.0224425
689	-.004248						
	m7		-.0085154	.004457	-1.91	0.057	-.0172839
690	.0002531						
	m8		-.0004554	.004392	-0.10	0.917	-.0090959
691	.0081852						
	m9		-.0056625	.0044299	-1.28	0.202	-.0143775
692	.0030526						
	m10		.0071688	.0044386	1.62	0.107	-.0015635
693	.0159011						
	m11		.0042074	.0044259	0.95	0.343	-.0044998
694	.0129146						
	_cons		.0067355	.0031722	2.12	0.034	.0004948
695	.0129762						
696	-----+-----						
697	185 . scalar drop _all						
698							
699	186 . quietly forval w=12(12)144 {						
700							
701	187 . scalar list						
702	RWmaxobs144 =	144					
703	RWminobs144 =	12					
704	RWrmse144 =	.01777071					
705	RWmaxobs132 =	132					
706	RWminobs132 =	12					
707	RWrmse132 =	.01782557					
708	RWmaxobs120 =	120					
709	RWminobs120 =	12					
710	RWrmse120 =	.01785253					

```

711      RWmaxobs108 =      108
712      RWminobs108 =      12
713      RWrmse108 = .01794692
714      RWmaxobs96 =      96
715      RWminobs96 =      12
716      RWrmse96 = .01793358
717      RWmaxobs84 =      84
718      RWminobs84 =      12
719      RWrmse84 = .01803355
720      RWmaxobs72 =      72
721      RWminobs72 =      12
722      RWrmse72 = .01807408
723      RWmaxobs60 =      60
724      RWminobs60 =      12
725      RWrmse60 = .01843535
726      RWmaxobs48 =      48
727      RWminobs48 =      12
728      RWrmse48 = .01835092
729      RWmaxobs36 =      36
730      RWminobs36 =      12
731      RWrmse36 = .01863303
732      RWmaxobs24 =      24
733      RWminobs24 =      12
734      RWrmse24 = .0196745
735      RWmaxobs12 =      12
736      RWminobs12 =      12
737      RWrmse12 = .01880291
738
739 188 . /*
740 > RWmaxobs144 =      144
741 > RWminobs144 =      12
742 > RWrmse144 = .01777071
743 > */
744 189 .
745 190 . reg d.lnCount 14d.lnWeekHours 19d.lnWeekHours 18d.lnHourlyEarnings m1 m2
m3 m
746 > 4 m5 m6 m7 m8 m9 m10 m11
747
748      Source |      SS          df          MS      Number of obs      =
749 112      -----+----- F(14, 97)      =
750 3.24      Model | .013917393      14      .0009941      Prob > F      =
751 0.0003      Residual | .029798432      97      .0003072      R-squared      =
752 0.3184      -----+----- Adj R-squared      =
753 0.2200      Total | .043715825      111      .000393836      Root MSE      =
754 .01753

```

	D.lnCount	Coef.	Std. Err.	t	P> t	[95% Conf.
755	--					
756	---					
757	Interv					
758	> al]					
759	-----+-----					
760	--					
761	lnWeekHours					
762	L4D. -.0013847	.0384012	-0.04	0.971	-.0776005	
763	.074					
764	> 831					
765	L9D. .0397686	.0385964	1.03	0.305	-.0368346	
766	.1163					
767	> 718					
768	lnHourlyEarnings					
769	L8D. -.039029	.0414024	-0.94	0.348	-.1212014	
770	.0431					
771	> 433					
772	m1 -.0097045	.0078517	-1.24	0.219	-.0252879	
773	.0058					
774	> 789					
775	m2 .0000949	.0079445	0.01	0.990	-.0156727	
776	.0158					
777	> 626					
778	m3 -.004712	.0083585	-0.56	0.574	-.0213013	
779	.0118					
780	> 773					
781	m4 -.0273667	.0081729	-3.35	0.001	-.0435876	
782	-.0111					
783	> 459					
784	m5 -.0076836	.0081259	-0.95	0.347	-.0238112	
785	.008					
786	> 444					
787	m6 -.020254	.0081465	-2.49	0.015	-.0364227	
788	-.0040					
789	> 854					
790	m7 -.0130812	.0081852	-1.60	0.113	-.0293265	
791	.0031					
792	> 642					
793	m8 .0041701	.0081051	0.51	0.608	-.0119164	
794	.0202					
795	> 565					
796	m9 -.0089171	.0082764	-1.08	0.284	-.0253435	
797	.0075					
798	> 093					
799	m10 .0153608	.0081153	1.89	0.061	-.0007459	
800	.0314					

```

790      > 674
791          m11 |   .0040463   .0079619    0.51   0.612   -.0117559
792      .0198
793      > 485
794          _cons |   .0094122   .0056462    1.67   0.099   -.0017939
795      .0206
796      > 183
797      -----
798
799      191 . scalar drop _all
800
801      192 . quietly forval w=12(12)84 {
802
803      193 . scalar list
804          RWmaxobs84 =           84
805          RWminobs84 =           2
806          RWrmse84 =   .01847546
807          RWmaxobs72 =           72
808          RWminobs72 =           2
809          RWrmse72 =   .01855448
810          RWmaxobs60 =           60
811          RWminobs60 =           2
812          RWrmse60 =   .01850723
813          RWmaxobs48 =           48
814          RWminobs48 =           2
815          RWrmse48 =   .01850217
816          RWmaxobs36 =           36
817          RWminobs36 =           2
818          RWrmse36 =   .01942535
819          RWmaxobs24 =           24
820          RWminobs24 =           2
821          RWrmse24 =   .02208272
822          RWmaxobs12 =           12
823          RWminobs12 =           2
824          RWrmse12 =   .0176238
825
826      194 . /*
827          > RWmaxobs12 =           12
828          > RWminobs12 =           2
829          > RWrmse12 =   .0176238
830          > */
831
832      195 .
833      196 . scalar rwmse = .0172128
834
835          Source |       SS           df           MS       Number of obs   =

```

325

```

836      -----+----- F(4, 320)      =
837      16.54
838      Model | .017539188      4 .004384797 Prob > F      =
839      0.0000
840      Residual | .084856979      320 .000265178 R-squared      =
841      0.1713
842      -----+----- Adj R-squared      =
843      0.1609
844      Total | .102396167      324 .000316038 Root MSE      =
845      .01628
846
847      -----
848      D.lnCount | Coef. Std. Err. t P>|t| [95% Conf.
849      Interval]
850      -----
851      lnCount |
852      L12D. | .3609966 .0621085 5.81 0.000 .238804
853      .4831893
854      L24D. | .137848 .0617615 2.23 0.026 .016338
855      .259358
856      L36D. | -.0160136 .0614584 -0.26 0.795 -.1369272
857      .1049
858      L48D. | .1265117 .0585322 2.16 0.031 .0113551
859      .2416683
860      |
861      _cons | .0017116 .0009853 1.74 0.083 -.0002269
862      .0036502
863
864      198 . predict pd
865      (option xb assumed; fitted values)
866      (49 missing values generated)
867
868      199 . gen pflcount=exp((rwmse^2)/2)*exp(l.lnCount+pd) if Date==tm(2021m3)
869      (374 missing values generated)
870
871      200 . gen ub1=exp((rwmse^2)/2)*exp(l.lnCount+pd+1*rwmse) if Date==tm(2021m3)
872      (374 missing values generated)
873
874      201 . gen lb1=exp((rwmse^2)/2)*exp(l.lnCount+pd-1*rwmse) if Date==tm(2021m3)
875      (374 missing values generated)
876
877      202 . gen ub2=exp((rwmse^2)/2)*exp(l.lnCount+pd+2*rwmse) if Date==tm(2021m3)
878      (374 missing values generated)
879
880      203 . gen lb2=exp((rwmse^2)/2)*exp(l.lnCount+pd-2*rwmse) if Date==tm(2021m3)
881      (374 missing values generated)
882

```

```

873 204 . gen ub3=exp((rwrms^2)/2)*exp(l.lnCount+pd+3*rwrms) if Date==tm(2021m3)
874     (374 missing values generated)
875
876 205 . gen lb3=exp((rwrms^2)/2)*exp(l.lnCount+pd-3*rwrms) if Date==tm(2021m3)
877     (374 missing values generated)
878
879 206 . drop pd
880
881 207 .
882 208 . replace pflcount=Count if Date==tm(2021m2)
883     (1 real change made)
884
885 209 . replace ub1=Count if Date==tm(2021m2)
886     (1 real change made)
887
888 210 . replace ub2=Count if Date==tm(2021m2)
889     (1 real change made)
890
891 211 . replace ub3=Count if Date==tm(2021m2)
892     (1 real change made)
893
894 212 . replace lb1=Count if Date==tm(2021m2)
895     (1 real change made)
896
897 213 . replace lb2=Count if Date==tm(2021m2)
898     (1 real change made)
899
900 214 . replace lb3=Count if Date==tm(2021m2)
901     (1 real change made)
902
903 215 .
904 216 . twoway (tsrline ub3 ub2 if tin(2020m3,2021m3), ///
905     >      recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
906     >      (tsrline ub2 ub1 if tin(2020m3,2021m3), ///
907     >      recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
908     >      (tsrline ub1 pflcount if tin(2020m3,2021m3), ///
909     >      recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
910     >      (tsrline pflcount lb1 if tin(2020m3,2021m3), ///
911     >      recast(rarea) fcolor(khaki) fintensity(65) lwidth(none) ) ///
912     >      (tsrline lb1 lb2 if tin(2020m3,2021m3), ///
913     >      recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
914     >      (tsrline lb2 lb3 if tin(2020m3,2021m3), ///
915     >      recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
916     >      (tsline Count pflcount if tin(2020m3,2021m3) , ///
917     >      lcolor(gs12 teal) lwidth(medthick medthick) ///
918     >      lpattern(solid longdash)) ///
919     >      (scatter withMarchCount Date if tin(2021m3,)), scheme(slmono)
920 legend(
921     > off)
922 217 . graph export "CountFan.png", replace

```

```

923      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
924 Sets
925
926      218 .
927      219 . /*-----
928 -----
929      > _*/
930      220 .
931      221 . *starter models for weekly earnings
932      222 . reg d.lnWeeklyEarnings l1d.lnWeekHours ld.lnHourlyEarnings
933
934      Source |      SS          df          MS      Number of obs   =
935      -----+----- F(2, 117)   =
936      1.48
937      Model |    .0071986      2     .0035993  Prob > F   =
938      0.2316
939      Residual |   .284290844     117   .002429836 R-squared   =
940      0.0247
941      -----+----- Adj R-squared   =
942      0.0080
943      Total |   .291489443     119   .002449491 Root MSE   =
944      .04929
945
946 -----
947      D.
948      |      Coef.  Std. Err.      t  P>|t|  [95% Conf.
949      lnWeeklyEarnings |      Interv
950      > al]
951
952      ---+
953      lnWeekHours |
954      LD. |  -.1334776   .1058563   -1.26   0.210   -.3431204
955      .0761
956      > 651
957      |
958      lnHourlyEarnings |
959      LD. |   .0746807   .1166933    0.64   0.523   -.1564244
960      .3057
961      > 857
962      |
963      _cons |   .0015108   .0045071    0.34   0.738   -.0074152
964      .0104
965      > 368
966
967 -----
968      ---+

```

```

959
960      223 . scalar drop _all
961
962      224 . quietly forval w=12(12)84 {
963
964      225 . scalar list
965          RWmaxobs84 =          84
966          RWminobs84 =          2
967          RWrmse84 =   .06183191
968          RWmaxobs72 =          72
969          RWminobs72 =          2
970          RWrmse72 =   .06162109
971          RWmaxobs60 =          60
972          RWminobs60 =          2
973          RWrmse60 =   .06144232
974          RWmaxobs48 =          48
975          RWminobs48 =          2
976          RWrmse48 =   .0618403
977          RWmaxobs36 =          36
978          RWminobs36 =          2
979          RWrmse36 =   .06201409
980          RWmaxobs24 =          24
981          RWminobs24 =          2
982          RWrmse24 =   .06224974
983          RWmaxobs12 =          12
984          RWminobs12 =          2
985          RWrmse12 =   .06583082
986
987      226 . /*
988          > RWmaxobs60 =          60
989          > RWminobs60 =          2
990          > RWrmse60 =   .06145693
991          > */
992      227 .
993      228 . quietly gsreg dlnWeeklyEarnings 11dlnWeeklyEarnings 12dlnWeeklyEarnings
13dln
994          > WeeklyEarnings ///
995          >     14dlnWeeklyEarnings 15dlnWeeklyEarnings 16dlnWeeklyEarnings ///
996          >     17dlnWeeklyEarnings 18dlnWeeklyEarnings 19dlnWeeklyEarnings
110dlnWee
997          > klyEarnings ///
998          >     111dlnWeeklyEarnings 112dlnWeeklyEarnings ///
999          >     124dlnWeeklyEarnings 136dlnWeeklyEarnings ///
1000         >     if tin(2011m1,2021m1), ///
1001         >     ncomb(1,12) aic outsample(24) ///
1002         >     samesample nindex( -1 aic -1 bic -1 rmse_out)
results(gsreg_dlnWeekly
1003         > Earnings) replace
1004
1005      229 .

```

```

1006      230 . reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings m1 m2 m3
1007      m4
1008
1009      Source |       SS          df          MS      Number of obs   =
1010      116
1011      -----+----- F(13, 102)   =
2.16
1012      Model |  .061304493      13  .00471573  Prob > F   =
0.0166
1013      Residual |  .222983103     102  .002186109 R-squared   =
0.2156
1014      -----+----- Adj R-squared   =
0.1157
1015      Total |  .284287596     115  .002472066 Root MSE    =
.04676
1016
1017      -----
1018      D.
1019      lnWeeklyEarnings |      Coef.  Std. Err.      t  P>|t|  [95% Conf.
1020      Interv
1021      > al]
1022
1023      -----
1024      lnWeeklyEarnings |
1025      L3D. |  -.2267216  .0950679  -2.38  0.019  -.4152883
-.0381
1026      > 549
1027      L5D. |  -.1621197  .095104  -1.70  0.091  -.350758
.0265
1028      > 186
1029      |
1030      m1 |  -.013308  .0213233  -0.62  0.534  -.0556027
.0289
1031      > 866
1032      m2 |  .020775  .0212478  0.98  0.331  -.02137
.06
1033      > 292
1034      m3 |  -.0123903  .0220875  -0.56  0.576  -.0562008
.0314
1035      > 201
1036      m4 |  .0105198  .0219037  0.48  0.632  -.0329261
.0539
1037      > 657
1038      m5 |  .0377285  .0216445  1.74  0.084  -.0052032
.0806
1039      > 602

```

```

1039          m6 |   .0272631   .0216181    1.26   0.210   -.0156164
1040      .0701
1040      > 426
1041          m7 |  -.0220653   .0214504   -1.03   0.306   -.064612
1041      .0204
1042      > 813
1043          m8 |   .0152172   .0210597    0.72   0.472   -.0265547
1043      .0569
1044      > 891
1045          m9 |   .0201901   .0215988    0.93   0.352   -.0226509
1045      .0630
1046      > 312
1047          m10 |   .0207722   .021844     0.95   0.344   -.0225553
1047      .0640
1048      > 997
1049          m11 |   .0084712   .0217091    0.39   0.697   -.0345888
1049      .0515
1050      > 312
1051          _cons |  -.0073031   .0151547   -0.48   0.631   -.0373625
1051      .0227
1052      > 563
1053  -----
1054  --
1054  ---
1055
1056  231 . scalar drop _all
1057
1058  232 . quietly forval w=12(12)84 {
1059
1060  233 . scalar list
1061      RWmaxobs84 =        84
1062      RWminobs84 =        2
1063      RWrmse84 =   .06011868
1064      RWmaxobs72 =        72
1065      RWminobs72 =        2
1066      RWrmse72 =   .06057642
1067      RWmaxobs60 =        60
1068      RWminobs60 =        2
1069      RWrmse60 =   .06071208
1070      RWmaxobs48 =        48
1071      RWminobs48 =        2
1072      RWrmse48 =   .06042055
1073      RWmaxobs36 =        36
1074      RWminobs36 =        2
1075      RWrmse36 =   .06125152
1076      RWmaxobs24 =        24
1077      RWminobs24 =        2
1078      RWrmse24 =   .06537943
1079      RWmaxobs12 =        12
1080      RWminobs12 =        2
1081      RWrmse12 =   .0702019

```

```

1082
1083      234 . /*
1084          > RWmaxobs84 =           84
1085          > RWminobs84 =           2
1086          > RWrmsse84 =   .06004448
1087          > */
1088      235 .
1089      236 . reg d.lnWeeklyEarnings 13d.lnWeeklyEarnings 15d.lnWeeklyEarnings
1090          17d.lnWeekly
1091          > Earnings
1092
1093      Source |      SS          df          MS      Number of obs     =
1094      114
1095      -----+----- F(3, 110)      =
1096      3.85
1097      Model |   .026396014      3   .008798671  Prob > F      =
1098      0.0115
1099      Residual |   .251283445     110   .002284395  R-squared      =
1100      0.0951
1101      -----+----- Adj R-squared      =
1102      0.0704
1103      Total |   .277679459     113   .00245734  Root MSE      =
1104      .0478
1105
1106      -----
1107      D.
1108      lnWeeklyEarnings |      Coef.    Std. Err.      t      P>|t|      [95% Conf.
1109      Interv
1110      > al]
1111      -----
1112      --
1113      lnWeeklyEarnings |
1114      L3D. |   -.2408947   .0906673   -2.66   0.009   -.4205761
1115      -.0612
1116      > 133
1117      L5D. |   -.1892527   .0903206   -2.10   0.038   -.3682468
1118      -.0102
1119      > 585
1120      L7D. |   .0639647   .0913902   0.70   0.485   -.1171492
1121      .2450
1122      > 786
1123      |
1124      _cons |   .0025724   .0044848   0.57   0.567   -.0063154
1125      .0114
1126      > 602
1127      -----
1128      --
1129      ---

```

```

1118
1119      237 . scalar drop _all
1120
1121      238 . quietly forval w=12(12)84 {
1122
1123      239 . scalar list
1124          RWmaxobs84 =          84
1125          RWminobs84 =          2
1126          RWrmse84 =   .05259823
1127          RWmaxobs72 =          72
1128          RWminobs72 =          2
1129          RWrmse72 =   .05283772
1130          RWmaxobs60 =          60
1131          RWminobs60 =          2
1132          RWrmse60 =   .05314168
1133          RWmaxobs48 =          48
1134          RWminobs48 =          2
1135          RWrmse48 =   .0530381
1136          RWmaxobs36 =          36
1137          RWminobs36 =          2
1138          RWrmse36 =   .05353125
1139          RWmaxobs24 =          24
1140          RWminobs24 =          2
1141          RWrmse24 =   .05282122
1142          RWmaxobs12 =          12
1143          RWminobs12 =          2
1144          RWrmse12 =   .06036464
1145
1146      240 . /*
1147          > RWmaxobs84 =          84
1148          > RWminobs84 =          2
1149          > RWrmse84 =   .05250414
1150          > */
1151
1152      241 .
1153
1154      242 . drop pflcount ub1 ub2 ub3 lb1 lb2 lb3
1155
1156
1157      243 .
1158
1159      244 . scalar rwmse = .05250414
1160
1161
1162      245 . reg d.lnWeeklyEarnings l(3,5,7)d.lnWeeklyEarnings if tin(,2021m2)
1163
1164
1165      Source |      SS          df          MS      Number of obs     =
1166      -----+----- F(3, 110)      =
1167      3.85
1168
1169      Model |   .026396014          3   .008798671  Prob > F      =
1170      0.0115
1171
1172      Residual |   .251283445        110   .002284395  R-squared      =
1173      0.0951
1174
1175      -----+----- Adj R-squared      =
1176      0.0704

```

```

1164          Total | .277679459      113   .00245734   Root MSE      =
1165          .0478
1166          -----
1167          --
1168          D.          |
1169          lnWeeklyEarnings |      Coef.    Std. Err.      t     P>|t|      [95% Conf.
1170          Interv
1171          > al]
1172          --
1173          lnWeeklyEarnings |
1174          L3D. | -.2408947   .0906673   -2.66   0.009   -.4205761
1175          -.0612
1176          > 133
1177          L5D. | -.1892527   .0903206   -2.10   0.038   -.3682468
1178          -.0102
1179          > 585
1180          L7D. | .0639647   .0913902   0.70    0.485   -.1171492
1181          .2450
1182          > 786
1183          -----
1184          --
1185          ---
1186          246 . predict pd
1187          (option xb assumed; fitted values)
1188          (260 missing values generated)
1189
1190          247 . gen pflcount=exp((rwrms^2)/2)*exp(l.lnWeeklyEarnings+pd) if
1191          Date==tm(2021m3)
1192          (374 missing values generated)
1193
1194          248 . gen ub1=exp((rwrms^2)/2)*exp(l.lnWeeklyEarnings+pd+1*rwrms) if
1195          Date==tm(202
1196          > 1m3)
1197          (374 missing values generated)
1198
1199          249 . gen lb1=exp((rwrms^2)/2)*exp(l.lnWeeklyEarnings+pd-1*rwrms) if
1200          Date==tm(202
1201          > 1m3)
1202          (374 missing values generated)
1203
1204          250 . gen ub2=exp((rwrms^2)/2)*exp(l.lnWeeklyEarnings+pd+2*rwrms) if
1205          Date==tm(202

```

```

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

```

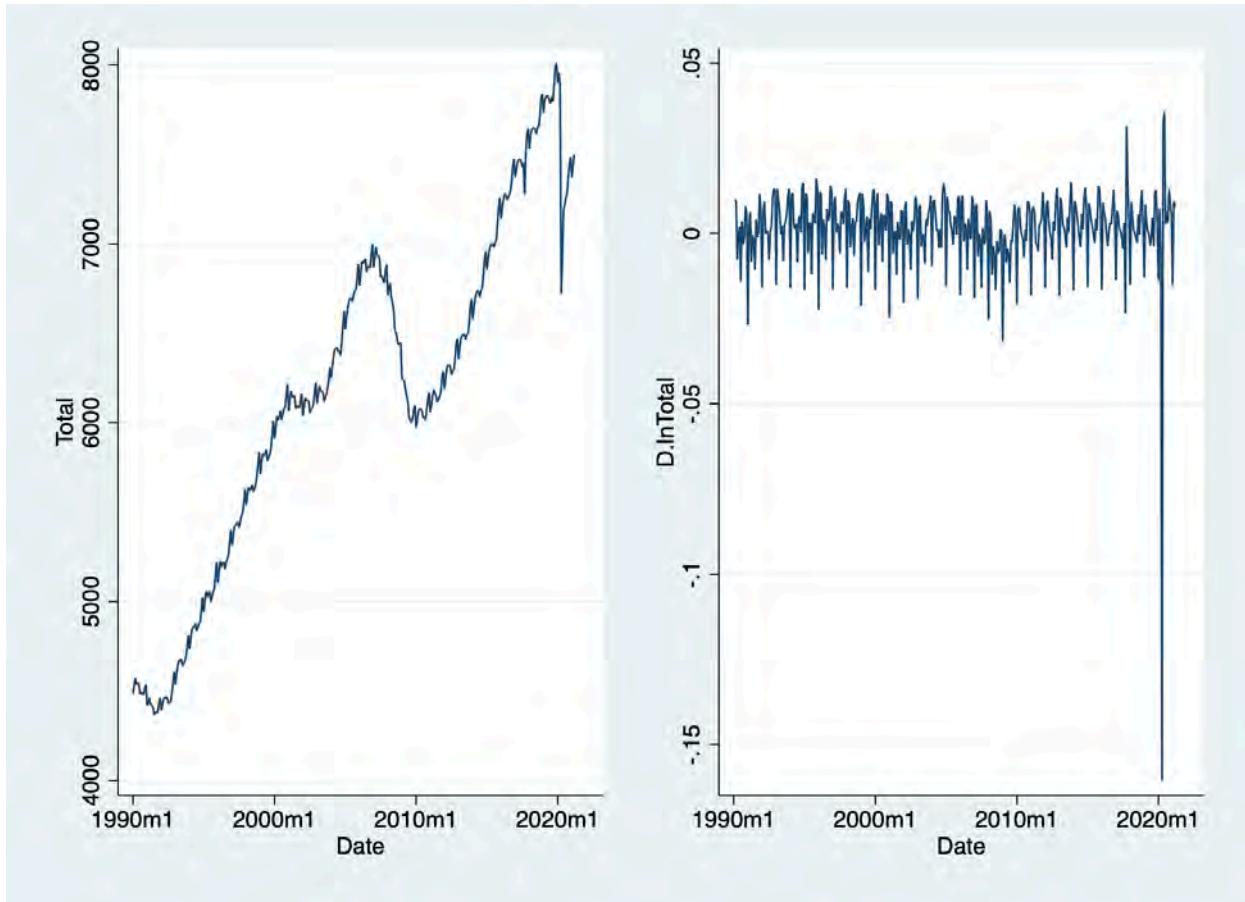
```
1250      >          (tsrline lb1 lb2 if tin(2020m3,2021m3), ///
1251      >          recast(rarea) fcolor(khaki) fintensity(40) lwidth(none) ) ///
1252      >          (tsrline lb2 lb3 if tin(2020m3,2021m3), ///
1253      >          recast(rarea) fcolor(khaki) fintensity(20) lwidth(none) ) ///
1254      >          (tsline WeeklyEarnings pflcount if tin(2020m3,2021m3) , ///
1255      >          lcolor(gs12 teal) lwidth(medthick medthick) ///
1256      >          lpattern(solid longdash)) ///
1257      >          (scatter withMarchEarnings Date if tin(2021m3,)), scheme(s1mono)

  lege
1258      > nd(off)
1259
1260      265 . graph export "WeeklyFan.png", replace
1261          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
  Sets
1262          > /Final Project/WeeklyFan.png written in PNG format)
1263
1264      266 .
1265      267 . log close
1266          name: <unnamed>
1267          log: /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
  Series/Probl
1268      > em Sets/Final Project/Final Project.smcl
1269          log type: smcl
1270          closed on: 19 Apr 2021, 21:08:41
1271
-----
```

--

Part 1

Time Series Plots

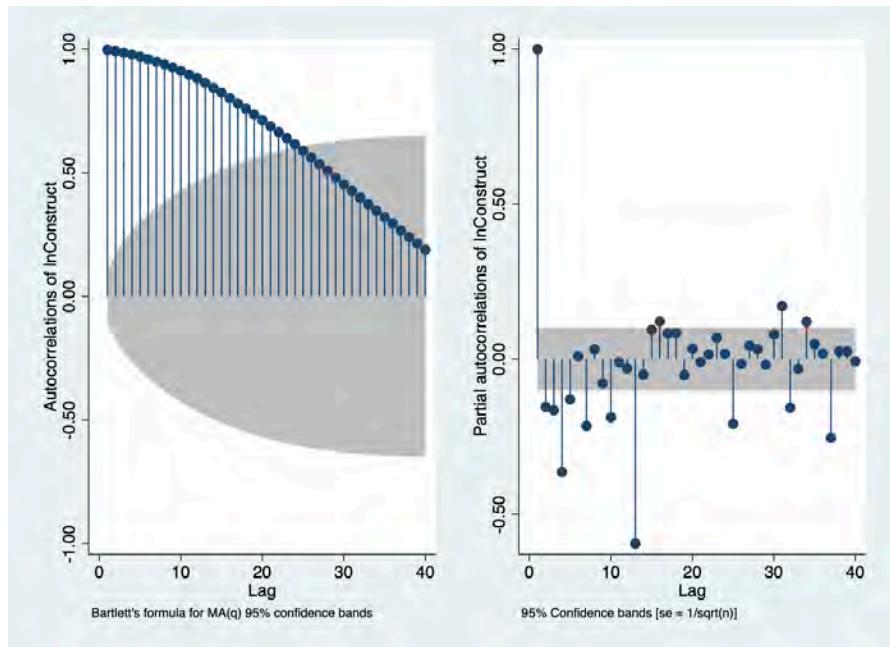


No Logs

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

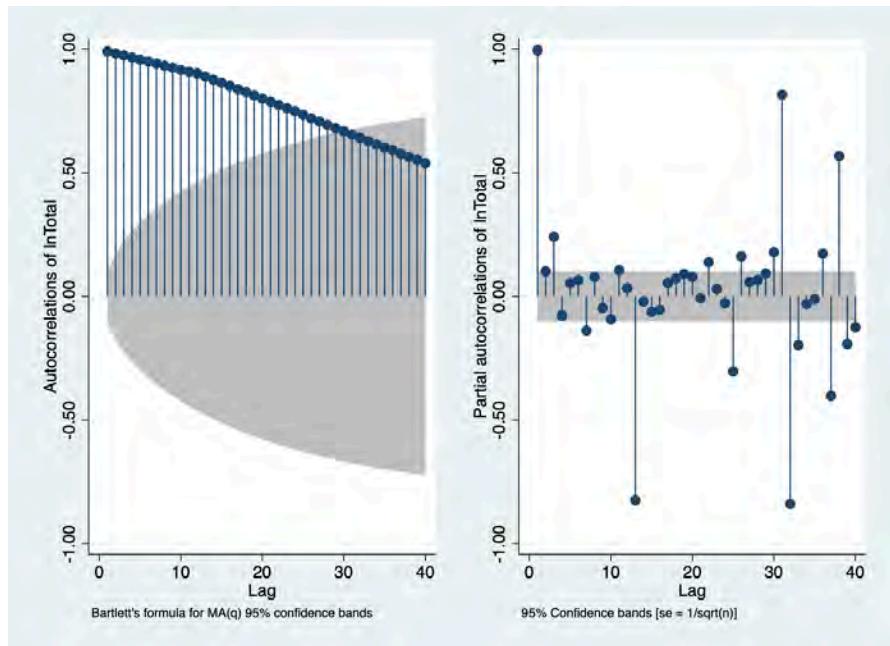
Construct



Both first lags are high which means we should difference.

Test	Result
MacKinnon approximate p-value for Z(t)	0.9796

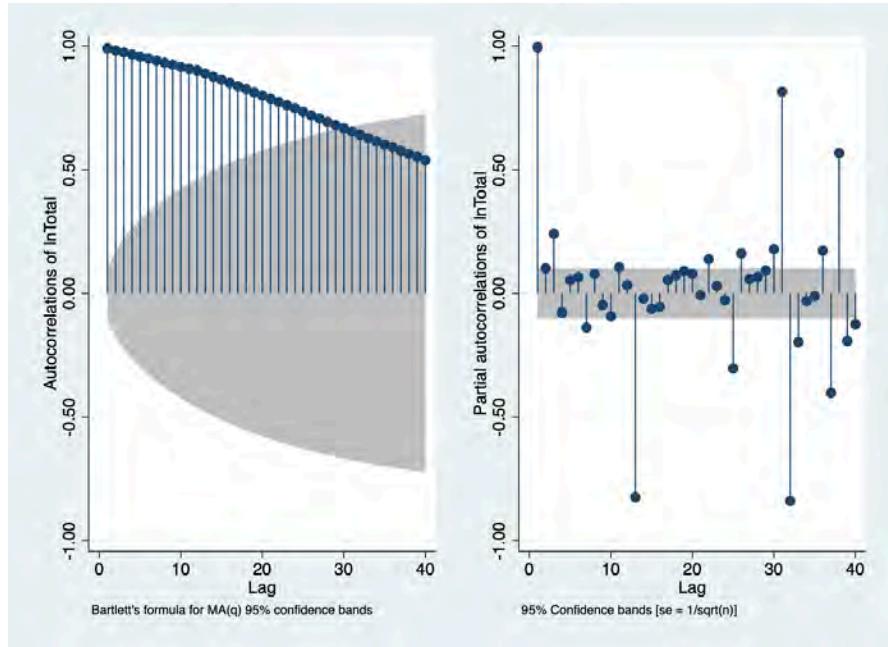
Manufacture



Both first lags are high which means we should difference.

Test	Result
MacKinnon approximate p-value for Z(t)	0.9963

Total

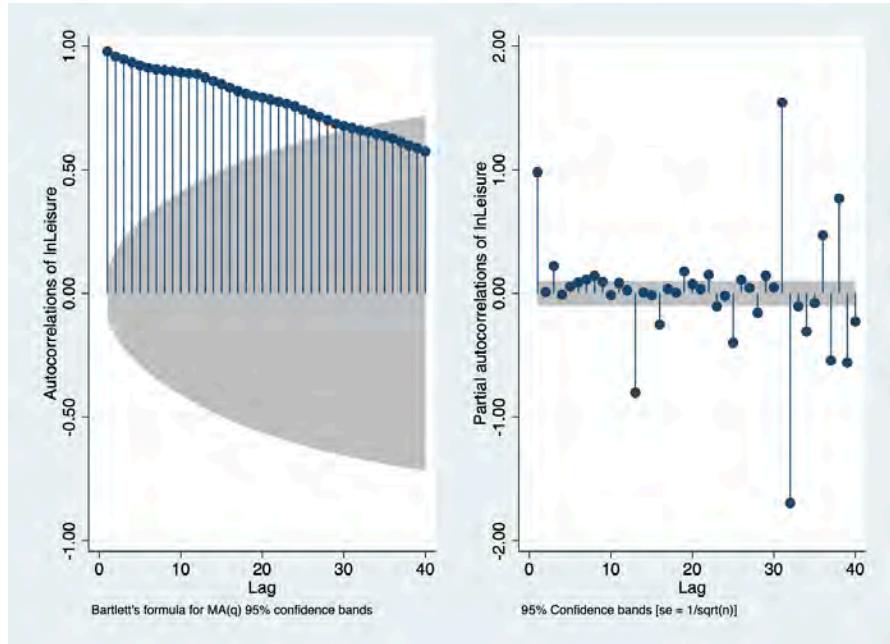


Both first lags are high which means we should difference.

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.InTotal	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 insignificant 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	<pre>reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11</pre>	-2382.725	-2323.982	.0129605
AR only Lags 1-3,12,24 Month dummies	<pre>reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11</pre>	-2236.218	-2170.633	.01300797
ARDL Lags 1-3 Month dummies	<pre>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</pre>	-2393.008	-2299.019	.01611154
ARDL Lags 1-3,12,24 Month dummies	<pre>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</pre>	-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
```

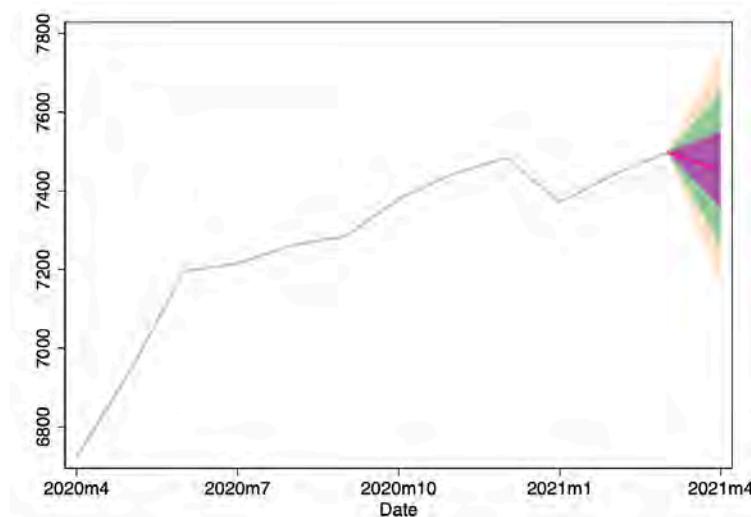
Value Type	Value
RWmaxobs12	12
RWminobs12	12
RWrmse12	.0132376

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

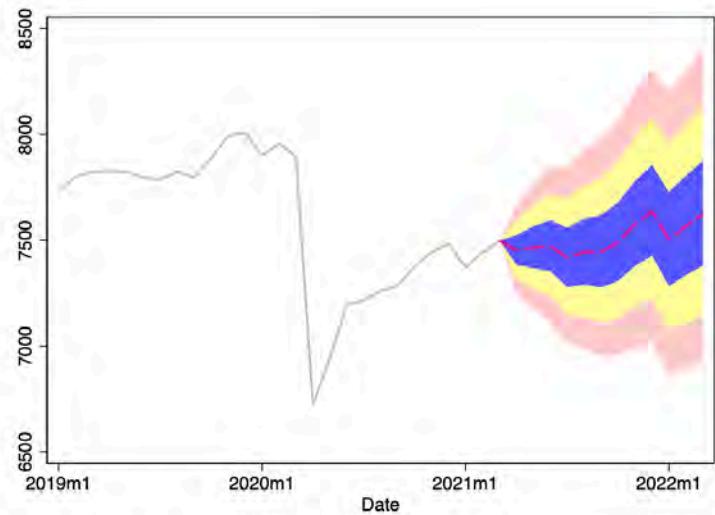
Forecasting

One month ahead



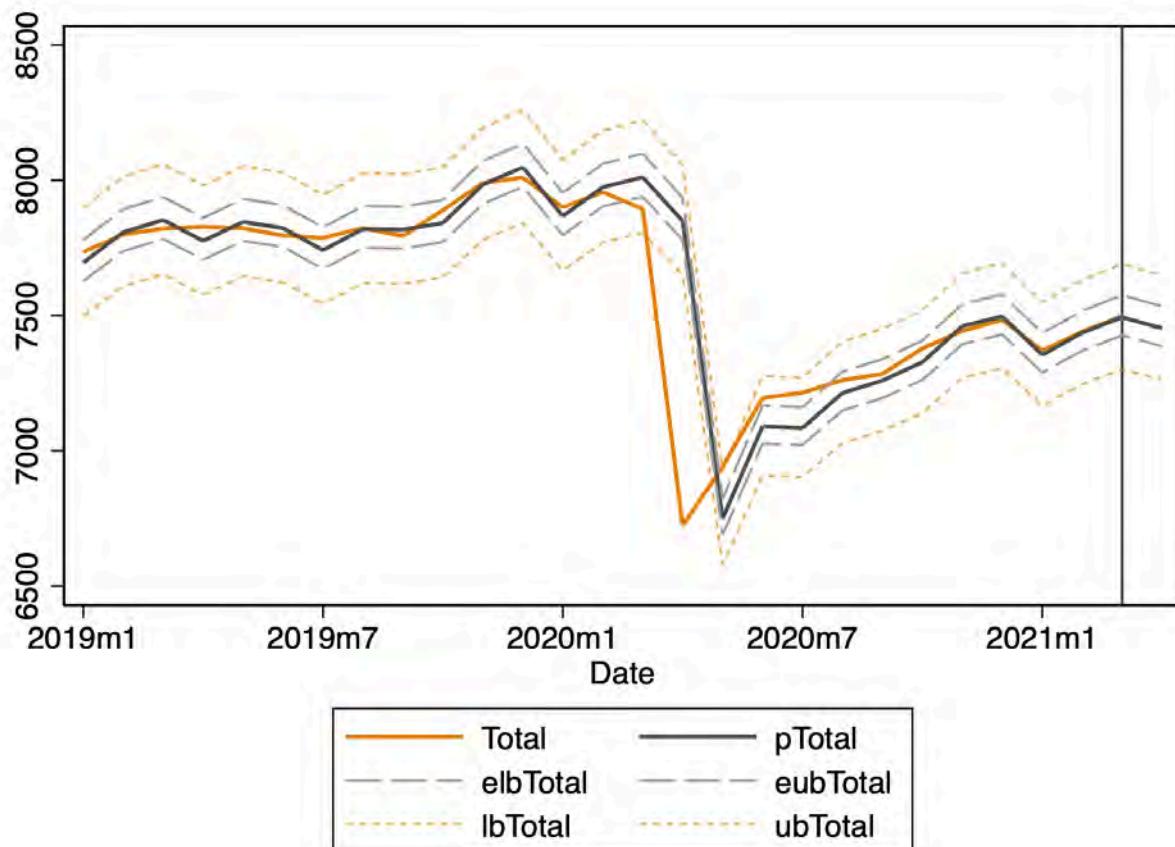
Sorry for the bad colors. Hailey peer pressured me into it.

One year out

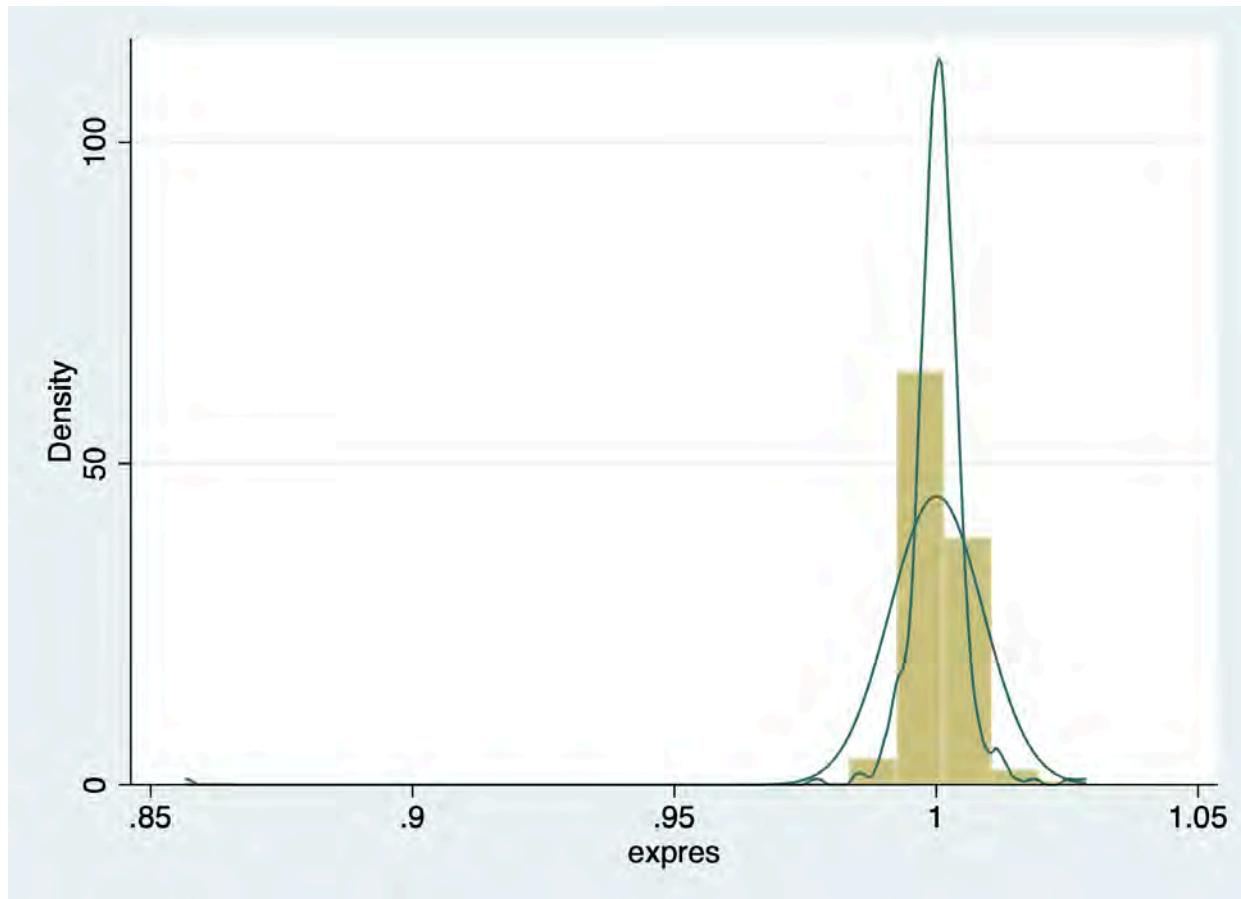


Ditto my earlier comment on the colors :)

Forecast Evaluation



The prediction is pretty good and stays well within bounds (as it should). The only real mis-step is around the pandemic because the model could not have foreseen such an event. Even so, once it did hit, the model stuck the landing and continued to forecast quite well 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

```
1 clear
2 set more off
3
4 cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem
Sets/Final Exam"
5 log using "Final Exam.smcl", replace
6 import delimited "SP21Final.csv"
```

```
7
8 gen datec=date(date, "YMD")
9 gen Date=mofd(datec)
10 gen month=month(datec)
11 format Date %tm
12 tsset Date
13
14 gen lnConstruct = ln(construct)
15 gen lnLeisure = ln(leisure)
16 gen lnManufacture = ln(manufacture)
17 gen lnTotal = ln(total)
18
19 gen Total = total
20 gen Construct = construct
21 gen Leisure = leisure
22 gen Manufacture = manufacture
23
24 /*
25 gen withMarchTotal = Total
26 replace Total=.` if tin(2021m3,`.)
27 */
28
29 tsset Date
30 tsappend, add(12)
31 replace month=month(dofm(Date))
32
33 gen m1=0
34 replace m1=1 if month==1
35 gen m2=0
36 replace m2=1 if month==2
37 gen m3=0
38 replace m3=1 if month==3
39 gen m4=0
40 replace m4=1 if month==4
41 gen m5=0
42 replace m5=1 if month==5
43 gen m6=0
44 replace m6=1 if month==6
45 gen m7=0
46 replace m7=1 if month==7
47 gen m8=0
48 replace m8=1 if month==8
49 gen m9=0
50 replace m9=1 if month==9
51 gen m10=0
52 replace m10=1 if month==10
53 gen m11=0
54 replace m11=1 if month==11
```

```

55
56
57 summ construct leisure manufacture total
58 summ lnConstruct lnLeisure lnManufacture lnTotal
59
60 tsline lnConstruct lnLeisure, saving(lnConstructLeisure_tsline.gph, replace)
61 tsline lnManufacture, saving(lnManufacture_tsline.gph, replace)
62 graph combine lnConstructLeisure_tsline.gph lnManufacture_tsline.gph, ///
   saving(lnConstructLeisure, replace)
63 graph export "lnConstructLeisure-Manufacture_tsline.png", replace
64
65
66 tsline lnTotal, saving(lnTotal_tsline.gph, replace)
67 tsline d.lnTotal, saving(dlnTotal_tsline.gph, replace)
68 graph combine Total_tsline.gph dlnTotal_tsline.gph, saving(lnTotal-Total,
   replace)
69 graph export "lnTotal-dlnTotal_tsline.png", replace
70
71 ac lnTotal, saving(lnTotal_ac, replace)
72 pac lnTotal, saving(lnTotal_pac, replace)
73 graph combine lnTotal_ac.gph lnTotal_pac.gph, saving(lnTotal_ac_pac, replace)
74 graph export "lnTotal_ac_pac.png", replace
75 dfuller lnTotal, trend regress
76
77 ac lnConstruct, saving(lnConstruct_ac, replace)
78 pac lnConstruct, saving(lnConstruct_pac, replace)
79 graph combine lnConstruct_ac.gph lnConstruct_pac.gph, saving(lnConstruct_ac_pac,
   replace)
80 graph export "lnConstruct_ac_pac.png", replace
81 dfuller lnConstruct, trend regress
82
83 ac lnLeisure, saving(lnLeisure_ac, replace)
84 pac lnLeisure, saving(lnLeisure_pac, replace)
85 graph combine lnLeisure_ac.gph lnLeisure_pac.gph, saving(lnLeisure_ac_pac,
   replace)
86 graph export "lnLeisure_ac_pac.png", replace
87 dfuller lnLeisure, trend regress
88
89 ac lnManufacture, saving(lnManufacture_ac, replace)
90 pac lnManufacture, saving(lnManufacture_pac, replace)
91 graph combine lnManufacture_ac.gph lnManufacture_pac.gph,
   saving(lnManufacture_ac_pac, replace)
92 graph export "lnManufacture_ac_pac.png", replace
93 dfuller lnManufacture, trend regress
94
95 quietly reg l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture
96 testparm l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture
97

```

```

98 newey d.lnTotal l(0/3,12,24)d.Construct l(0/3,12,24)d.Manufacture
100 l(0/3,12,24)d.Leisure, lag(24)
101 test ld.Construct + ld.Construct + 12d.Construct + 13d.Construct +
102 112d.Construct + 124d.Construct ///
103 == d.Manufacture + ld.Manufacture + 12d.Manufacture + 13d.Manufacture + ///
104 112d.Manufacture + 124d.Manufacture
105 test d.Construct + ld.Construct + 12d.Construct + 13d.Construct + 112d.Construct
106 + 124d.Construct ///
107 == d.Leisure + ld.Leisure + 12d.Leisure + 13d.Leisure + 112d.Leisure +
108 124d.Leisure
109 test d.Leisure + ld.Leisure + 12d.Leisure + 13d.Leisure + 112d.Leisure +
110 124d.Leisure ///
111 == d.Manufacture + ld.Manufacture + 12d.Manufacture + 13d.Manufacture + ///
112 112d.Manufacture + 124d.Manufacture
113
114 *-----
115 gen dlnConstruct=d.lnConstruct
116 gen l1dlnConstruct=l1d.lnConstruct
117 gen l2dlnConstruct=l2d.lnConstruct
118 gen l3dlnConstruct=l3d.lnConstruct
119 gen l12dlnConstruct=l12d.lnConstruct
120 gen l24dlnConstruct=l24d.lnConstruct
121
122 gen dlnLeisure=d.lnLeisure
123 gen l1dlnLeisure=l1d.lnLeisure
124 gen l2dlnLeisure=l2d.lnLeisure
125 gen l3dlnLeisure=l3d.lnLeisure
126 gen l12dlnLeisure=l12d.lnLeisure
127 gen l24dlnLeisure=l24d.lnLeisure
128
129 gen dlnManufacture=d.lnManufacture
130 gen l1dlnManufacture=l1d.lnManufacture
131 gen l2dlnManufacture=l2d.lnManufacture
132 gen l3dlnManufacture=l3d.lnManufacture
133 gen l12dlnManufacture=l12d.lnManufacture
134 gen l24dlnManufacture=l24d.lnManufacture
135
136
137 /*
138 gsreg dlnTotal dlnConstruct l1dlnConstruct l2dlnConstruct l3dlnConstruct ///
139 112dlnConstruct 124dlnConstruct ///

```

```

140 dlnLeisure l1dlnLeisure 12dlnLeisure 13dlnLeisure 112dlnLeisure 124dlnLeisure
141 /**
142     dlnManufacture l1dlnManufacture 12dlnManufacture 13dlnManufacture /**
143     l12dlnManufacture 124dlnManufacture /**
144     if tin(1990m1,2021m3), /**
145     ncomb(1,6) aic outsamp(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11) /**
146     samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnTtoal) replace
147 */
148
149 loocv reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
150 quietly reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
151 estat ic
152
153 loocv reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
154 quietly reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
155 estat ic
156
157 loocv reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure /**
158     l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
159 quietly reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure /**
160     l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
161 estat ic
162
163 loocv reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct /**
164     l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
165     m11
166 quietly reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct /**
167     l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
168     m11
169 estat ic
170
171 *Lowest rmse (1)
172 reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
173 scalar drop _all
174 quietly forval w=12(12)180 {
175     gen pred=.
176     gen nobs=.
177     forval t=544/734 {
178         gen wstart=`t'-'`w'
179         gen wend=`t'-1
180         reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 /**
181             if Date>=wstart & Date<=wend
182             replace nobs=e(N) if Date==`t'
183             predict ptemp
184             replace pred=ptemp if Date==`t'
185             drop ptemp wstart wend
186     }

```

```

185 gen errsq=(pred-d.lnTotal)^2
186 summ errsq
187 scalar RWrmse`w'=r(mean)^.5
188 summ nobs
189 scalar RWminobs`w'=r(min)
190 scalar RWmaxobs`w'=r(max)
191 drop errsq pred nobs
192 }
193 scalar list
194 /*
195 RWmaxobs12 = 12
196 RWminobs12 = 12
197 RWrmse12 = .0132376
198 */
199
200
201 *lowest AIC and BIC (3)
202 reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
203 l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
204 scalar drop _all
205 quietly forval w=3(3)180 {
206 gen pred=.
207 gen nobs=.
208 forval t=544/734 {
209 gen wstart=`t'-'`w'
210 gen wend=`t'-1
211 reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
212 l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
213 if Date>=wstart & Date<=wend
214 replace nobs=e(N) if Date==`t'
215 predict ptemp
216 replace pred=ptemp if Date==`t'
217 drop ptemp wstart wend
218 }
219 gen errsq=(pred-d.lnTotal)^2
220 summ errsq
221 scalar RWrmse`w'=r(mean)^.5
222 summ nobs
223 scalar RWminobs`w'=r(min)
224 scalar RWmaxobs`w'=r(max)
225 drop errsq pred nobs
226 }
227 scalar list
228 /*
229 RWmaxobs12 = 12
230 RWminobs12 = 12
231 RWrmse12 = .0132376
232 */

```

```

233
234
235 * Going with model 1 because average RWrmse is lower across window sizes
236 scalar rwmse = .0132376
237 reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(,2021m3)
238 predict pd
239 gen pflcount=exp((rwmse^2)/2)*exp(l.lnTotal+pd) if Date==tm(2021m4)
240 gen ub1=exp((rwmse^2)/2)*exp(l.lnTotal+pd+1*rwmse) if Date==tm(2021m4)
241 gen lb1=exp((rwmse^2)/2)*exp(l.lnTotal+pd-1*rwmse) if Date==tm(2021m4)
242 gen ub2=exp((rwmse^2)/2)*exp(l.lnTotal+pd+2*rwmse) if Date==tm(2021m4)
243 gen lb2=exp((rwmse^2)/2)*exp(l.lnTotal+pd-2*rwmse) if Date==tm(2021m4)
244 gen ub3=exp((rwmse^2)/2)*exp(l.lnTotal+pd+3*rwmse) if Date==tm(2021m4)
245 gen lb3=exp((rwmse^2)/2)*exp(l.lnTotal+pd-3*rwmse) if Date==tm(2021m4)
246 drop pd
247
248 replace pflcount=Total if Date==tm(2021m3)
249 replace ub1=Total if Date==tm(2021m3)
250 replace ub2=Total if Date==tm(2021m3)
251 replace ub3=Total if Date==tm(2021m3)
252 replace lb1=Total if Date==tm(2021m3)
253 replace lb2=Total if Date==tm(2021m3)
254 replace lb3=Total if Date==tm(2021m3)
255
256 twoway (tsrline ub3 ub2 if tin(2020m4,2021m4), ///
257     recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
258     (tsrline ub2 ub1 if tin(2020m4,2021m4), ///
259     recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
260     (tsrline ub1 pflcount if tin(2020m4,2021m4), ///
261     recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
262     (tsrline pflcount lb1 if tin(2020m4,2021m4), ///
263     recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
264     (tsrline lb1 lb2 if tin(2020m4,2021m4), ///
265     recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
266     (tsrline lb2 lb3 if tin(2020m4,2021m4), ///
267     recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
268     (tsline Total pflcount if tin(2020m4,2021m4) , ///
269     lcolor(gs12 pink) lwidth(medthick medthick) ///
270     lpattern(solid longdash)), scheme(s1mono) legend(off)
271 graph export "TotalFan1.png", replace
272
273 * More than 1 step
274 arima d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
275 tin(1990m1,2021m3)
276 predict pnonfarm, dynamic(tm(2021m3))
277 predict mse, mse dynamic(mofd(tm(2021m4)))
278 gen totmse = mse if Date==tm(2021m4)
279 replace totmse = l.totmse+mse if Date>tm(2021m4)
280 gen pnonfarma = Total if Date==tm(2021m3)

```

```

280 replace pnonfarma = 1.pnonfarma*exp(pnonfarm+rmse/2) if Date>tm(2021m3)
281
282 gen ub1a = pnonfarma*exp(totrmse^.5)
283 gen ub2a = pnonfarma*exp(2*totrmse^.5)
284 gen ub3a = pnonfarma*exp(3*totrmse^.5)
285 gen lb1a = pnonfarma/exp(totrmse^.5)
286 gen lb2a = pnonfarma/exp(2*totrmse^.5)
287 gen lb3a = pnonfarma/exp(3*totrmse^.5)
288
289 replace ub1a=Total if Date == tm(2021m3)
290 replace ub2a=Total if Date == tm(2021m3)
291 replace ub3a=Total if Date == tm(2021m3)
292 replace lb1a=Total if Date == tm(2021m3)
293 replace lb2a=Total if Date == tm(2021m3)
294 replace lb3a=Total if Date == tm(2021m3)
295
296 twoway (tsrline ub3a ub2a if tin(2019m1,2022m3), ///
297     recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
298     (tsrline ub2a ub1a if tin(2019m1,2022m3), ///
299     recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
300     (tsrline ub1a pnonfarma if tin(2019m1,2022m3), ///
301     recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
302     (tsrline pnonfarma lb1a if tin(2019m1,2022m3), ///
303     recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
304     (tsrline lb1a lb2a if tin(2019m1,2022m3), ///
305     recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
306     (tsrline lb2a lb3a if tin(2019m1,2022m3), ///
307     recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
308     (tsline Total pnonfarma if tin(2019m1,2022m3) , ///
309     lcolor(gs12 pink) lwidth(medthick medthick) ///
310     lpattern(solid longdash)) , scheme(slmono) legend(off)
311 graph export "TotalFan12.png", replace
312
313 scalar rmse_mod1 = .0132376
314 reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
tin(1990m1,2021m3)
315 predict plTotal
316 predict temp if tin(2021m3,2021m3)
317 replace plTotal=temp if tin(2021m3,2021m3)
318 drop temp
319 gen pTotal=exp(l.lnTotal+plTotal+(rmse_mod1^2)/2)
320 gen lbTotal=exp(l.lnTotal+plTotal-1.96*rmse_mod1+(rmse_mod1^2)/2)
321 gen ubTotal=exp(l.lnTotal+plTotal+1.96*rmse_mod1+(rmse_mod1^2)/2)
322
323 gen res=(d.lnTotal-plTotal)
324 gen expres=exp(res)
325 summ expres
326 scalar meanexpres=r(mean)

```

```

327  gen epTotal=exp(l.lnTotal+plTotal)*meanexpres
328  _pctile res, percentile(2.5,97.5)
329  return list
330  gen elbTotal=exp(l.lnTotal+plTotal+r(r1))*meanexpres
331  gen eubTotal=exp(l.lnTotal+plTotal+r(r2))*meanexpres
332
333  tsline Total pTotal elbTotal eubTotal lbTotal ubTotal if tin(2019m1,2021m4), ///
334    scheme(s1mono) tline(2021m3, lcolor(gs4)) ///
335    lpattern(solid solid longdash longdash shortdash shortdash) ///
336    lcolor(dkorange gs5 gs10 gs10 dkorange%60 dkorange%60) ///
337    lwidth(medthick medthick medium medium)
338  graph export "interval_tsline.png", replace
339
340  histogram expres, normal kdensity saving(residuals.gph, replace)
341  graph export "residuals.png", replace
342
343  log close
344  translate "Final Exam.smcl" "Final Project.txt", replace

```

Log File

```

1      ____(R)          _____
2      ____/          /____ /____/
3      /____/          ____/  /  /____/  /
4                                         Statistics/Data analysis
5
6  -----
7
8      name: <unnamed>
9      log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
10     Series/Probl
11       > em Sets/Final Exam/Final Exam.smcl
12       log type: smcl
13       opened on: 29 Apr 2021, 12:07:09
14
15
16       2 .
17       3 . gen datec=date(date, "YMD")
18
19       4 . gen Date=mofd(datec)

```

```
20
21      5 . gen month=month(datec)
22
23      6 . format Date %tm
24
25      7 . tsset Date
26          time variable: Date, 1990m1 to 2021m3
27          delta: 1 month
28
29      8 .
30      9 . gen lnConstruct = ln(construct)
31
32      10 . gen lnLeisure = ln(leisure)
33
34      11 . gen lnManufacture = ln(manufacture)
35
36      12 . gen lnTotal = ln(total)
37
38      13 .
39      14 . gen Total = total
40
41      15 . gen Construct = construct
42
43      16 . gen Leisure = leisure
44
45      17 . gen Manufacture = manufacture
46
47      18 .
48      19 . /*
49          > gen withMarchTotal = Total
50          > replace Total=. if tin(2021m3,)
51          > */
52
53      21 . tsset Date
54          time variable: Date, 1990m1 to 2021m3
55          delta: 1 month
56
57      22 . tsappend, add(12)
58
59      23 . replace month=month(dofm(Date))
60          (12 real changes made)
61
62      24 .
63      25 . gen m1=0
64
65      26 . replace m1=1 if month==1
66          (33 real changes made)
67
```

```
68      27 . gen m2=0
69
70      28 . replace m2=1 if month==2
71          (33 real changes made)
72
73      29 . gen m3=0
74
75      30 . replace m3=1 if month==3
76          (33 real changes made)
77
78      31 . gen m4=0
79
80      32 . replace m4=1 if month==4
81          (32 real changes made)
82
83      33 . gen m5=0
84
85      34 . replace m5=1 if month==5
86          (32 real changes made)
87
88      35 . gen m6=0
89
90      36 . replace m6=1 if month==6
91          (32 real changes made)
92
93      37 . gen m7=0
94
95      38 . replace m7=1 if month==7
96          (32 real changes made)
97
98      39 . gen m8=0
99
100     40 . replace m8=1 if month==8
101         (32 real changes made)
102
103     41 . gen m9=0
104
105     42 . replace m9=1 if month==9
106         (32 real changes made)
107
108     43 . gen m10=0
109
110     44 . replace m10=1 if month==10
111         (32 real changes made)
112
113     45 . gen m11=0
114
115     46 . replace m11=1 if month==11
```

```

116      (32 real changes made)
117
118      47 .
119      48 .
120      49 . summ construct leisure manufacture total
121
122      Variable |       Obs        Mean     Std. Dev.      Min      Max
123      -----
124      construct |      375    461.0043    95.8947    323.9    696.1
125      leisure |      375   930.2083   159.6216    660.6   1287.5
126      manufacture |      375   410.0496   63.14375   307.9    518.2
127      total |      375  6161.164   958.2068  4366.1   8010.4
128
129      50 . summ lnConstruct lnLeisure lnManufacture lnTotal
130
131      Variable |       Obs        Mean     Std. Dev.      Min      Max
132      -----
133      lnConstruct |      375    6.112377    .2043883   5.780435   6.545493
134      lnLeisure |      375    6.820994    .1695282   6.493148   7.160458
135      lnManufacture |      375    6.004071    .1577555   5.729775   6.250361
136      lnTotal |      375    8.71332     .1619029   8.381625   8.988496
137
138      51 .
139      52 . tsline lnConstruct lnLeisure, saving(lnConstructLeisure_tsline.gph,
replace)
140          (file lnConstructLeisure_tsline.gph saved)
141
142      53 . tsline lnManufacture, saving(lnManufacture_tsline.gph, replace)
143          (file lnManufacture_tsline.gph saved)
144
145      54 . graph combine lnConstructLeisure_tsline.gph lnManufacture_tsline.gph,
///
146          > saving(lnConstructLeisure, replace)
147          (file lnConstructLeisure.gph saved)
148
149      55 . graph export "lnConstructLeisure-Manufacture_tsline.png", replace
150          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Problem Sets
151          > /Final Exam/lnConstructLeisure-Manufacture_tsline.png written in PNG
format)
152
153      56 .
154      57 . tsline lnTotal, saving(lnTotal_tsline.gph, replace)
155          (file lnTotal_tsline.gph saved)
156
157      58 . tsline d.lnTotal, saving(dlnTotal_tsline.gph, replace)
158          (file dlnTotal_tsline.gph saved)
159

```

```

160      59 . graph combine Total_tsline.gph dlnTotal_tsline.gph, saving(lnTotal-
161          Total, rep
162              > lace)
163          (file lnTotal-Total.gph saved)
164
165      60 . graph export "lnTotal-dlnTotal_tsline.png", replace
166          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
167          Series/Problem Sets
168              > /Final Exam/lnTotal-dlnTotal_tsline.png written in PNG format)
169
170      61 .
171
172      62 . ac lnTotal, saving(lnTotal_ac, replace)
173          (file lnTotal_ac.gph saved)
174
175      63 . pac lnTotal, saving(lnTotal_pac, replace)
176          (file lnTotal_pac.gph saved)
177
178      64 . graph combine lnTotal_ac.gph lnTotal_pac.gph, saving(lnTotal_ac_pac,
179          replace)
180          (file lnTotal_ac_pac.gph saved)
181
182      65 . graph export "lnTotal_ac_pac.png", replace
183          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
184          Series/Problem Sets
185              > /Final Exam/lnTotal_ac_pac.png written in PNG format)
186
187      66 . dfuller lnTotal, trend regress
188
189          Dickey-Fuller test for unit root                      Number of obs     =
190          374
191
192          -----
193          ----- Interpolated Dickey-Fuller -----
194          -----
195          Critical
196          Test          1% Critical        5% Critical        10%
197          Critical
198          Statistic       Value           Value
199          Value
200          -----
201          -----
202          Z(t)          -1.895          -3.985          -3.425
203          -3.130
204          -----
205          -----
206          MacKinnon approximate p-value for Z(t) = 0.6570
207
208          -----
209          -----

```

```

195      D.lnTotal |      Coef.    Std. Err.      t      P>|t|      [95% Conf.
196      Interval]
197      -----
198      lnTotal |
199      L1. |  -.0191916   .0101251     -1.90   0.059   -.0391013
200      .0007181
201      |_
202      _trend |   .0000223   .0000151     1.47   0.142   -7.48e-06
203      .0000521
204      _cons |   .1644033   .085612     1.92   0.056   -.0039423
205      .3327489
206      -----
207      |
208      67 .
209      68 . ac lnConstruct, saving(lnConstruct_ac, replace)
210          (file lnConstruct_ac.gph saved)
211
212      69 . pac lnConstruct, saving(lnConstruct_pac, replace)
213          (file lnConstruct_pac.gph saved)
214
215      70 . graph combine lnConstruct_ac.gph lnConstruct_pac.gph,
216          saving(lnConstruct_ac_p
217          > ac, replace)
218          (file lnConstruct_ac_pac.gph saved)
219
220      71 . graph export "lnConstruct_ac_pac.png", replace
221          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
222          Series/Problem Sets
223          > /Final Exam/lnConstruct_ac_pac.png written in PNG format)
224
225      72 . dfuller lnConstruct, trend regress
226
227      Dickey-Fuller test for unit root                      Number of obs = 374
228
229      ----- Interpolated Dickey-Fuller -----
230
231      Critical
232      Test      1% Critical      5% Critical      10%
233      Statistic
234      Value
235      Value
236      -----
237      Z(t)      -0.586      -3.985      -3.425
238      -3.130

```

```

228      -----
229      MacKinnon approximate p-value for Z(t) = 0.9796
230
231      -----
232      D.          |
233      lnConstruct |      Coef.    Std. Err.      t      P>|t|      [95% Conf.
234      Interval]
235      -----
236      lnConstruct |
237      L1. | -.0021673   .0036962     -0.59   0.558   -.0094355
238      .0051009
239      |
240      _trend |  5.05e-06   6.99e-06     0.72   0.470   -8.68e-06
241      .0000188
242      _cons |   .0132628   .0221901     0.60   0.550   -.0303713
243      .0568969
244      -----
245
246      73 .
247      74 . ac lnLeisure, saving(lnLeisure_ac, replace)
248      (file lnLeisure_ac.gph saved)
249
250      75 . pac lnLeisure, saving(lnLeisure_pac, replace)
251      (file lnLeisure_pac.gph saved)
252
253      76 . graph combine lnLeisure_ac.gph lnLeisure_pac.gph,
254      saving(lnLeisure_ac_pac, re
255      > place)
256      (file lnLeisure_ac_pac.gph saved)
257
258      77 . graph export "lnLeisure_ac_pac.png", replace
259      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
260      Series/Problem Sets
261      > /Final Exam/lnLeisure_ac_pac.png written in PNG format)
262
263      78 . dfuller lnLeisure, trend regress
264
265      Dickey-Fuller test for unit root                      Number of obs   =
266      374
267
268      -----
269      ----- Interpolated Dickey-Fuller -----
270
271      -----
272      Test          1% Critical       5% Critical       10%
273      Critical

```

```

263          Statistic      Value      Value
264          Value
265          -----
266          Z(t)        -4.787      -3.985      -3.425
267          -3.130
268          -----
269          MacKinnon approximate p-value for Z(t) = 0.0005
270          -----
271          D.lnLeisure | Coef.   Std. Err.      t      P>|t|      [95% Conf.
272          Interval]
273          -----
274          lnLeisure |
275          L1. | -.1242368 .0259543     -4.79    0.000    -.1752728
276          -.0732007
277          |
278          _trend | .0001744 .0000407      4.28    0.000    .0000943
279          .0002545
280          _cons | .8156996 .1699707      4.80    0.000    .4814728
281          1.149926
282          -----
283          79 .
284          80 . ac lnManufacture, saving(lnManufacture_ac, replace)
285          (file lnManufacture_ac.gph saved)
286
287          81 . pac lnManufacture, saving(lnManufacture_pac, replace)
288          (file lnManufacture_pac.gph saved)
289
290          82 . graph combine lnManufacture_ac.gph lnManufacture_pac.gph,
291          saving(lnManufactur
292          > e_ac_pac, replace)
293          (file lnManufacture_ac_pac.gph saved)
294
295          83 . graph export "lnManufacture_ac_pac.png", replace
296          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Problem Sets
297          > /Final Exam/lnManufacture_ac_pac.png written in PNG format)
298
299          84 . dfuller lnManufacture, trend regress
300
301          Dickey-Fuller test for unit root
302          Number of obs      =
303          374

```

```

297
298                               ----- Interpolated Dickey-Fuller -----
299
300
301
302      Test          1% Critical      5% Critical      10%
303      Critical
304      Statistic       Value        Value
305      Value
306
307      -----
308      Z(t)           0.313        -3.985        -3.425
309      -3.130
310
311
312
313
314
315
316
317      85 .
318      86 . quietly reg l(12,24)d.Construct l(12,24)d.Leisure
319      l(12,24)d.Manufacture
320
321
322      87 . testparm l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture
323
324      ( 1) L24D.Construct = 0
325      ( 2) L12D.Leisure = 0
326      ( 3) L24D.Leisure = 0
327      ( 4) L12D.Manufacture = 0
328      ( 5) L24D.Manufacture = 0
329
330      F( 5,    356) = 146.28
331                  Prob > F = 0.0000

```

```

331     88 .
332     89 . newey d.lnTotal l(0/3,12,24)d.Construct l(0/3,12,24)d.Manufacture
l(0/3,12,24
333         > )d.Leisure, lag(24)
334
335     Regression with Newey-West standard errors      Number of obs      =
350
336             maximum lag: 24                           F( 18,      331) =
4461.04
337
338             Prob > F      =
0.0000
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359

```

		Newey-West				
	D.lnTotal	Coef.	Std. Err.	t	P> t	[95% Conf.
Interval]						
.0005842	Construct D1.	.0004252	.0000809	5.26	0.000	.0002661
.0001112	LD.	.0000247	.000044	0.56	0.574	-.0000617
.0001405	L2D.	.0000613	.0000403	1.52	0.129	-.0000179
.0000462	L3D.	-.0000674	.0000577	-1.17	0.244	-.000181
.000091	L12D.	-.000035	.0000641	-0.55	0.585	-.000161
.0001839	L24D.	.0000494	.0000684	0.72	0.470	-.0000851
	Manufacture					
.0015723	D1.	.0011938	.0001924	6.21	0.000	.0008154
.0000353	LD.	-.0002374	.0001386	-1.71	0.088	-.00051
-.0001948	L2D.	-.000336	.0000718	-4.68	0.000	-.0004773
-.0000606	L3D.	-.0002689	.0001059	-2.54	0.012	-.0004772
.0006025	L12D.	.0003308	.0001381	2.39	0.017	.000059
.0005182	L24D.	.0002403	.0001413	1.70	0.090	-.0000376
	Leisure					

360	D1.		.0002134	8.38e-06	25.47	0.000	.0001969
361	.0002299						
362	LD.		5.34e-06	9.19e-06	0.58	0.561	-.0000127
363	.0000234						
364	L2D.		-2.42e-06	.0000172	-0.14	0.888	-.0000364
365	.0000315						
366	L3D.		-8.76e-06	.0000174	-0.50	0.614	-.0000429
367	.0000254						
368	L12D.		.0000629	.0000361	1.74	0.082	-8.04e-06
369	.0001339						
370	L24D.		-.000016	.0000428	-0.37	0.708	-.0001002
371	.0000681						
372							
373	_cons		.0011957	.0003664	3.26	0.001	.000475
374	.0019165						
375	-----						
376	-----						
377	90 . test ld.Construct + ld.Construct + l2d.Construct + l3d.Construct +						
378	l12d.Const						
379	> ruct + l24d.Construct ///						
380	> == d.Manufacture + ld.Manufacture + l2d.Manufacture +						
381	l3d.Manufacture						
382	> + ///						
383	>						
384	l12d.Manufacture + l24d.Manufacture						
385	-----						
386	(1) 2*LD.Construct + L2D.Construct + L3D.Construct + L12D.Construct +						
387	L24D.Construct - D.Manufacture - LD.Manufacture - L2D.Manufacture						
388	-						
389	L3D.Manufacture - L12D.Manufacture - L24D.Manufacture = 0						
390	-----						
391	-----						
392	91 . test d.Construct + ld.Construct + l2d.Construct + l3d.Construct +						
393	l12d.Constr						
394	> uct + l24d.Construct ///						
395	> == d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure +						
396	l12d.Leisure						
397	> + l24d.Leisure						
398	-----						
399	(1) D.Construct + LD.Construct + L2D.Construct + L3D.Construct +						
400	L12D.Construct + L24D.Construct - D.Leisure - LD.Leisure -						
401	L2D.Leisure -						
402	L3D.Leisure - L12D.Leisure - L24D.Leisure = 0						
403	-----						
404	-----						
405	F(1, 331) = 5.85						
406	Prob > F = 0.0161						
407	-----						
408	-----						
409	-----						
410	-----						
411	-----						
412	-----						
413	-----						
414	-----						
415	-----						
416	-----						
417	-----						
418	-----						
419	-----						
420	-----						
421	-----						
422	-----						
423	-----						
424	-----						
425	-----						
426	-----						
427	-----						
428	-----						
429	-----						
430	-----						
431	-----						
432	-----						
433	-----						
434	-----						
435	-----						
436	-----						
437	-----						
438	-----						
439	-----						
440	-----						
441	-----						
442	-----						
443	-----						
444	-----						
445	-----						
446	-----						
447	-----						
448	-----						
449	-----						
450	-----						
451	-----						
452	-----						
453	-----						
454	-----						
455	-----						
456	-----						
457	-----						
458	-----						
459	-----						
460	-----						
461	-----						
462	-----						
463	-----						
464	-----						
465	-----						
466	-----						
467	-----						
468	-----						
469	-----						
470	-----						
471	-----						
472	-----						
473	-----						
474	-----						
475	-----						
476	-----						
477	-----						
478	-----						
479	-----						
480	-----						
481	-----						
482	-----						
483	-----						
484	-----						
485	-----						
486	-----						
487	-----						
488	-----						
489	-----						
490	-----						
491	-----						
492	-----						
493	-----						
494	-----						
495	-----						
496	-----						
497	-----						
498	-----						
499	-----						
500	-----						
501	-----						
502	-----						
503	-----						
504	-----						
505	-----						
506	-----						
507	-----						
508	-----						
509	-----						
510	-----						
511	-----						
512	-----						
513	-----						
514	-----						
515	-----						
516	-----						
517	-----						
518	-----						
519	-----						
520	-----						
521	-----						
522	-----						
523	-----						
524	-----						
525	-----						
526	-----						
527	-----						
528	-----						
529	-----						
530	-----						
531	-----						
532	-----						
533	-----						
534	-----						
535	-----						
536	-----						
537	-----						
538	-----						
539	-----						
540	-----						
541	-----						
542	-----						
543	-----						
544	-----						
545	-----						
546	-----						
547	-----						
548	-----						
549	-----						
550	-----						
551	-----						
552	-----						
553	-----						
554	-----						
555	-----						
556	-----						
557	-----						
558	-----						
559	-----						
560	-----						
561	-----						
562	-----						
563	-----						
564	-----						
565	-----						
566	-----						
567	-----						
568	-----						
569	-----						
570	-----						
571	-----						
572	-----						
573	-----						
574	-----						
575	-----						
576	-----						
577	-----						
578	-----						
579	-----						
580	-----						
581	-----						
582	-----						
583	-----						
584	-----						
585	-----						
586	-----						
587	-----						
588	-----						
589	-----						
590	-----						
591	-----						
592	-----						
593	-----						
594	-----						
595	-----						
596	-----						
597	-----						
598	-----						
599	-----						
600	-----						
601	-----						
602	-----						
603	-----						
604	-----						
605	-----						
606	-----						
607	-----						
608	-----						
609	-----				</		

```

394
395      92 . test d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure + l12d.Leisure
+ l24d
396      > .Leisure ///
397      > == d.Manufacture + ld.Manufacture + l2d.Manufacture +
398      l3d.Manufacture
399      > + ///
400      >           l12d.Manufacture + l24d.Manufacture
401
402      ( 1) - D.Manufacture - LD.Manufacture - L2D.Manufacture -
403      L3D.Manufacture -
404
405      L12D.Manufacture - L24D.Manufacture + D.Leisure + LD.Leisure +
406      L2D.Leisure + L3D.Leisure + L12D.Leisure + L24D.Leisure = 0
407
408      F( 1,    331) =     4.74
409      Prob > F =     0.0301
410
411      93 .
412
413
414      94 . *-----
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
-----
```

> --*

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)

```
438  
439      105 . gen l3dlnLeisure=l3d.lnLeisure  
440          (13 missing values generated)  
441  
442      106 . gen l12dlnLeisure=l12d.lnLeisure  
443          (13 missing values generated)  
444  
445      107 . gen l24dlnLeisure=l24d.lnLeisure  
446          (25 missing values generated)  
447  
448      108 .  
449      109 . gen dlnManufacture=d.lnManufacture  
450          (13 missing values generated)  
451  
452      110 . gen l1dlnManufacture=l1d.lnManufacture  
453          (13 missing values generated)  
454  
455      111 . gen l2dlnManufacture=l2d.lnManufacture  
456          (13 missing values generated)  
457  
458      112 . gen l3dlnManufacture=l3d.lnManufacture  
459          (13 missing values generated)  
460  
461      113 . gen l12dlnManufacture=l12d.lnManufacture  
462          (13 missing values generated)  
463  
464      114 . gen l24dlnManufacture=l24d.lnManufacture  
465          (25 missing values generated)  
466  
467      115 .  
468      116 . gen dlnTotal=d.lnTotal  
469          (13 missing values generated)  
470  
471      117 . gen l1dlnTotal=l1d.lnTotal  
472          (13 missing values generated)  
473  
474      118 . gen l2dlnTotal=l2d.lnTotal  
475          (13 missing values generated)  
476  
477      119 . gen l3dlnTotal=l3d.lnTotal  
478          (13 missing values generated)  
479  
480      120 . gen l12dlnTotal=l12d.lnTotal  
481          (13 missing values generated)  
482  
483      121 . gen l24dlnTotal=l24d.lnTotal  
484          (25 missing values generated)  
485
```

```

486    122 .
487    123 . /*
488        > gsreg dlnTotal dlnConstruct l1dlnConstruct l2dlnConstruct
489        l3dlnConstruct ///
490        >           l12dlnConstruct l24dlnConstruct ///
491        >           dlnLeisure l1dlnLeisure l2dlnLeisure l3dlnLeisure l12dlnLeisure
492        l24dl
493        >           nLeisure ///
494        >           dlnManufacture l1dlnManufacture l2dlnManufacture
495        l3dlnManufacture ///
496        >           l12dlnManufacture l24dlnManufacture ///
497        >           if tin(1990m1,2021m3), ///
498        >           ncomb(1,6) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
499        m11)
500        > /**
501        >           samesample nindex( -1 aic -1 bic -1 rmse_out)
502        results(gsreg_dlnTtoal)
503        > replace
504        > */
505    124 .
506    125 .
507    126 . loocv reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
508
509    127 . quietly reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
510    m11
511
512    128 . estat ic
513
514    Akaike's information criterion and Bayesian information criterion
515
516    -----
517
518    Model | N ll(null) ll(model) df AIC
519    BIC
520    -----
521    . | 371 1102.298 1206.362 15 -2382.725
522
523    -2323.982

```

```

524      -----
525      Note: BIC uses N = number of observations. See [R] BIC note.
526
527      129 .
528      130 . loocv reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9
m10 m11
529
530
531      Leave-One-Out Cross-Validation Results
532      -----
533      Method | Value
534      +-
535      Root Mean Squared Errors | .01300797
536      Mean Absolute Errors | .00427199
537      Pseudo-R2 | .14852182
538
539
540      131 . quietly reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9
m10 m1
541      > 1
542
543      132 . estat ic
544
545      Akaike's information criterion and Bayesian information criterion
546
547      -----
548      Model | N ll(null) ll(model) df AIC
549      BIC
550      +-
551      . | 350 1035.49 1135.109 17 -2236.218
-2170.633
552
553
554      Note: BIC uses N = number of observations. See [R] BIC note.
555
556      133 .
557      134 . loocv reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct
l(1/3)d.lnLeisure ///
558      > l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
559
560
561      Leave-One-Out Cross-Validation Results
562      -----
563      Method | Value
564      +-

```

```

563      Root Mean Squared Errors |   .01611154
564      Mean Absolute Errors   |   .00423375
565      Pseudo-R2              |   .06722376
566 -----
567
568      135 . quietly reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct
569          l(1/3)d.lnLeisure /
570          > //
571          >           l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
572
573      136 . estat ic
574
575      Akaike's information criterion and Bayesian information criterion
576 -----
577
578      -----+-----+-----+-----+
579      Model |       N    ll(null)    ll(model)      df      AIC
580      BIC
581      . |     371    1102.298    1220.504      24   -2393.008
582      -2299.019
583
584      Note: BIC uses N = number of observations. See [R] BIC note.
585
586      137 .
587      138 . loocv reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct
588      ///
589          >           l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6
590          m7 m8
591          >   m9 m10 m11
592
593      Leave-One-Out Cross-Validation Results
594
595      -----+-----+
596      Method        |   Value
597
598      Root Mean Squared Errors |   .01713897
599      Mean Absolute Errors   |   .004436
600      Pseudo-R2              |   .05184396
601
602
603      139 . quietly reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct
604      ///
605          >           l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6
606          m7 m8
607          >   m9 m10 m11

```

```

601
602     140 . estat ic
603
604         Akaike's information criterion and Bayesian information criterion
605
606     -----
607
608     ----- Model |      N    ll(null)    ll(model)      df      AIC
BIC
609     -----+-----+
610
611     . |      350    1035.49    1145.391      30   -2230.781
-2115.043
612
613
614     141 .
615     142 . *Lowest rmse (1)
616     143 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
617
618     ----- Source |      SS          df          MS      Number of obs      =
371
619     -----+----- F(14, 356)      =
19.13
620     Model |  .024491123      14  .001749366  Prob > F      =
0.0000
621     Residual |  .032550038      356  .000091433  R-squared      =
0.4294
622     -----+----- Adj R-squared      =
0.4069
623     Total |  .057041161      370  .000154165  Root MSE      =
.00956
624
625
626     ----- D.lnTotal |      Coef.    Std. Err.        t    P>|t|    [95% Conf.
Interval]
627
628     -----+-----+
629
630     lnTotal |
631     LD. |  -.0016173  .0529806  -0.03  0.976  -.1058116
.1025769
632     L2D. |  -.1325018  .0525096  -2.52  0.012  -.2357698
-.0292338
633     L3D. |  .026173  .0529824  0.49  0.622  -.0780247
.1303708
634
635

```

632		m1	-.0254857	.0024679	-10.33	0.000	-.0303391
633		-.0206322					
		m2	.0004869	.0029045	0.17	0.867	-.0052252
634		.006199					
		m3	-.003913	.0027686	-1.41	0.158	-.0093579
635		.0015318					
		m4	-.013317	.0026543	-5.02	0.000	-.0185371
636		-.0080969					
		m5	-.0064632	.0026182	-2.47	0.014	-.0116124
637		-.0013141					
		m6	-.0088465	.0025835	-3.42	0.001	-.0139273
638		-.0037657					
		m7	-.0159741	.0025331	-6.31	0.000	-.0209559
639		-.0109924					
		m8	-.0051433	.0026374	-1.95	0.052	-.0103302
640		.0000435					
		m9	-.0096793	.0025635	-3.78	0.000	-.0147208
641		-.0046378					
		m10	-.0021472	.0025489	-0.84	0.400	-.00716
642		.0028657					
		m11	.0024587	.0024684	1.00	0.320	-.0023958
643		.0073131					
		_cons	.0088125	.0018473	4.77	0.000	.0051795
644		.0124454					

645		-----					
646		144 . scalar drop _all					
647							
648		145 . quietly forval w=12(12)180 {					
649							
650		146 . scalar list					
651		RWmaxobs180 = 180					
652		RWminobs180 = 180					
653		RWrmse180 = .01789015					
654		RWmaxobs168 = 168					
655		RWminobs168 = 168					
656		RWrmse168 = .01812779					
657		RWmaxobs156 = 156					
658		RWminobs156 = 156					
659		RWrmse156 = .0186461					
660		RWmaxobs144 = 144					
661		RWminobs144 = 144					
662		RWrmse144 = .01935361					
663		RWmaxobs132 = 132					
664		RWminobs132 = 132					
665		RWrmse132 = .02010893					
666		RWmaxobs120 = 120					

```

667      RWminobs120 =      120
668      RWrmse120 = .02273114
669      RWmaxobs108 =      108
670      RWminobs108 =      108
671      RWrmse108 = .0233275
672      RWmaxobs96 =      96
673      RWminobs96 =      96
674      RWrmse96 = .0244855
675      RWmaxobs84 =      84
676      RWminobs84 =      84
677      RWrmse84 = .02493232
678      RWmaxobs72 =      72
679      RWminobs72 =      72
680      RWrmse72 = .02471091
681      RWmaxobs60 =      60
682      RWminobs60 =      60
683      RWrmse60 = .02489376
684      RWmaxobs48 =      48
685      RWminobs48 =      48
686      RWrmse48 = .02475145
687      RWmaxobs36 =      36
688      RWminobs36 =      36
689      RWrmse36 = .02791922
690      RWmaxobs24 =      24
691      RWminobs24 =      24
692      RWrmse24 = .16402211
693      RWmaxobs12 =      12
694      RWminobs12 =      12
695      RWrmse12 = .0132376
696
697      147 .
698      148 . /*
699      > RWmaxobs12 =      12
700      > RWminobs12 =      12
701      > RWrmse12 = .0132376
702      > */
703      149 .
704      150 . *lowest AIC and BIC (3)
705      151 . reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure
    ///
706      >          l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
707
708      Source |      SS           df           MS       Number of obs   =
709      371
710      -----+----- F(23, 347)   =
13.45
710      Model | .026880308      23  .001168709  Prob > F      =
0.0000

```

711	Residual .030160853	347	.000086919	R-squared	=
712	0.4712			Adj R-squared	=
713	0.4362				
714	Total .057041161	370	.000154165	Root MSE	=
715	.00932				
716	D.lnTotal Coef. Std. Err. t P> t [95% Conf.				
717	Interval]				
718	lnTotal				
719	LD. .2862672 .2667427 1.07 0.284 -.2383687				
720	.8109032				
721	L2D. .2454317 .2661885 0.92 0.357 -.2781141				
722	.7689776				
723	L3D. .4266043 .2702181 1.58 0.115 -.1048673				
724	.9580758				
725					
726	lnConstruct				
727	LD. 6.41e-06 .081453 0.00 1.000 -.1601974				
728	.1602102				
729	L2D. .0327201 .0818081 0.40 0.689 -.128182				
730	.1936223				
731	L3D. -.0371503 .0817044 -0.45 0.650 -.1978486				
732	.123548				
733					
734	lnLeisure				
735	LD. -.1208348 .0688988 -1.75 0.080 -.2563467				
736	.0146771				
737	L2D. -.1169829 .0684363 -1.71 0.088 -.251585				
738	.0176192				
739	L3D. -.1609605 .069157 -2.33 0.021 -.2969801				
740	-.0249411				
741					
742	lnManufacture				
743	LD. .0485695 .1821043 0.27 0.790 -.3095975				
744	.4067365				
745	L2D. -.1583242 .1862613 -0.85 0.396 -.5246673				
746	.208019				
747	L3D. .1372412 .1883006 0.73 0.467 -.2331129				
748	.5075952				
749					
750	m1 -.0259546 .0026844 -9.67 0.000 -.0312344				
751	-.0206748				

739		m2	.0061874	.0054433	1.14	0.256	-.0045187
	.0168935						
740		m3	.0041315	.0052298	0.79	0.430	-.0061545
	.0144176						
741		m4	-.0019669	.0049557	-0.40	0.692	-.011714
	.0077802						
742		m5	-.0024728	.0032223	-0.77	0.443	-.0088106
	.0038649						
743		m6	-.0057242	.0032307	-1.77	0.077	-.0120785
	.00063						
744		m7	-.0128758	.0035363	-3.64	0.000	-.0198311
	-.0059206						
745		m8	-.0017029	.003742	-0.46	0.649	-.0090627
	.005657						
746		m9	-.006856	.0032793	-2.09	0.037	-.0133058
	-.0004062						
747		m10	.0015872	.0030906	0.51	0.608	-.0044915
	.007666						
748		m11	.0030825	.0026162	1.18	0.240	-.0020632
	.0082282						
749		_cons	.0039798	.0029489	1.35	0.178	-.0018201
	.0097797						
750		<hr/>					
		<hr/>					
751							
752	152	. scalar drop _all					
753							
754	153	. quietly forval w=3(3)180 {					
755							
756	154	. scalar list					
757	RWmaxobs180 =	180					
758	RWminobs180 =	180					
759	RWrmse180 =	.06231312					
760	RWmaxobs177 =	177					
761	RWminobs177 =	177					
762	RWrmse177 =	.06310936					
763	RWmaxobs174 =	174					
764	RWminobs174 =	174					
765	RWrmse174 =	.06557282					
766	RWmaxobs171 =	171					
767	RWminobs171 =	171					
768	RWrmse171 =	.06644967					
769	RWmaxobs168 =	168					
770	RWminobs168 =	168					
771	RWrmse168 =	.0679586					
772	RWmaxobs165 =	165					
773	RWminobs165 =	165					
774	RWrmse165 =	.0691683					

```
775      RWmaxobs162 =      162
776      RWminobs162 =      162
777      RWrmse162 = .06977207
778      RWmaxobs159 =      159
779      RWminobs159 =      159
780      RWrmse159 = .07051975
781      RWmaxobs156 =      156
782      RWminobs156 =      156
783      RWrmse156 = .0708996
784      RWmaxobs153 =      153
785      RWminobs153 =      153
786      RWrmse153 = .07104253
787      RWmaxobs150 =      150
788      RWminobs150 =      150
789      RWrmse150 = .07151194
790      RWmaxobs147 =      147
791      RWminobs147 =      147
792      RWrmse147 = .0724523
793      RWmaxobs144 =      144
794      RWminobs144 =      144
795      RWrmse144 = .07292486
796      RWmaxobs141 =      141
797      RWminobs141 =      141
798      RWrmse141 = .07359791
799      RWmaxobs138 =      138
800      RWminobs138 =      138
801      RWrmse138 = .07415885
802      RWmaxobs135 =      135
803      RWminobs135 =      135
804      RWrmse135 = .07551713
805      RWmaxobs132 =      132
806      RWminobs132 =      132
807      RWrmse132 = .07895271
808      RWmaxobs129 =      129
809      RWminobs129 =      129
810      RWrmse129 = .08680818
811      RWmaxobs126 =      126
812      RWminobs126 =      126
813      RWrmse126 = .08791729
814      RWmaxobs123 =      123
815      RWminobs123 =      123
816      RWrmse123 = .0885621
817      RWmaxobs120 =      120
818      RWminobs120 =      120
819      RWrmse120 = .0893681
820      RWmaxobs117 =      117
821      RWminobs117 =      117
822      RWrmse117 = .09222918
```

```
823      RWmaxobs114 =      114
824      RWminobs114 =      114
825      RWrmse114 = .09918976
826      RWmaxobs111 =      111
827      RWminobs111 =      111
828      RWrmse111 = .10159669
829      RWmaxobs108 =      108
830      RWminobs108 =      108
831      RWrmse108 = .1074872
832      RWmaxobs105 =      105
833      RWminobs105 =      105
834      RWrmse105 = .11024241
835      RWmaxobs102 =      102
836      RWminobs102 =      102
837      RWrmse102 = .11448013
838      RWmaxobs99 =      99
839      RWminobs99 =      99
840      RWrmse99 = .11681123
841      RWmaxobs96 =      96
842      RWminobs96 =      96
843      RWrmse96 = .12412703
844      RWmaxobs93 =      93
845      RWminobs93 =      93
846      RWrmse93 = .13010923
847      RWmaxobs90 =      90
848      RWminobs90 =      90
849      RWrmse90 = .12958828
850      RWmaxobs87 =      87
851      RWminobs87 =      87
852      RWrmse87 = .13728567
853      RWmaxobs84 =      84
854      RWminobs84 =      84
855      RWrmse84 = .14028654
856      RWmaxobs81 =      81
857      RWminobs81 =      81
858      RWrmse81 = .14073066
859      RWmaxobs78 =      78
860      RWminobs78 =      78
861      RWrmse78 = .14462066
862      RWmaxobs75 =      75
863      RWminobs75 =      75
864      RWrmse75 = .14520888
865      RWmaxobs72 =      72
866      RWminobs72 =      72
867      RWrmse72 = .14882576
868      RWmaxobs69 =      69
869      RWminobs69 =      69
870      RWrmse69 = .16426363
```

```

871      RWmaxobs66 =       66
872      RWminobs66 =       66
873          RWrmse66 =   .16732114
874      RWmaxobs63 =       63
875      RWminobs63 =       63
876          RWrmse63 =   .1679914
877      RWmaxobs60 =       60
878      RWminobs60 =       60
879          RWrmse60 =   .18224837
880      RWmaxobs57 =       57
881      RWminobs57 =       57
882          RWrmse57 =   .19923544
883      RWmaxobs54 =       54
884      RWminobs54 =       54
885          RWrmse54 =   .2008839
886      RWmaxobs51 =       51
887      RWminobs51 =       51
888          RWrmse51 =   .2096647
889      RWmaxobs48 =       48
890      RWminobs48 =       48
891          RWrmse48 =   .20847129
892      RWmaxobs45 =       45
893      RWminobs45 =       45
894          RWrmse45 =   .2244767
895      RWmaxobs42 =       42
896      RWminobs42 =       42
897          RWrmse42 =   .25720484
898      RWmaxobs39 =       39
899      RWminobs39 =       39
900          RWrmse39 =   .24835914
901      RWmaxobs36 =       36
902      RWminobs36 =       36
903          RWrmse36 =   .24045465
904      RWmaxobs33 =       33
905      RWminobs33 =       33
906          RWrmse33 =   .26919986
907      RWmaxobs30 =       30
908      RWminobs30 =       30
909          RWrmse30 =   .49888354
910      RWmaxobs27 =       27
911      RWminobs27 =       27
912          RWrmse27 =   .32666868
913      RWmaxobs24 =       24
914      RWminobs24 =       24
915          RWrmse24 =   .66642366
916      RWmaxobs21 =       21
917      RWminobs21 =       21
918          RWrmse21 =   .34614878

```

```

919      RWmaxobs18 =          18
920      RWminobs18 =          18
921          RWrmse18 =     .757383
922      RWmaxobs15 =          15
923      RWminobs15 =          15
924          RWrmse15 =   .21744051
925      RWmaxobs12 =          12
926      RWminobs12 =          12
927          RWrmse12 =   .0132376
928      RWmaxobs9 =           9
929      RWminobs9 =           9
930          RWrmse9 =   .01629782
931      RWmaxobs6 =           6
932      RWminobs6 =           6
933          RWrmse6 =   .0162575
934      RWmaxobs3 =           3
935      RWminobs3 =           3
936          RWrmse3 =   .0206163
937
938 155 . /*
939      > RWmaxobs12 =          12
940      > RWminobs12 =          12
941      > RWrmse12 =   .0132376
942      > */
943 156 .
944 157 .
945 158 . * Going with model 1 because average RWrmse is lower across window
946 sizes
947
948 160 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
949 tin(,2021
950      > m3)
951
952      Source |       SS          df          MS      Number of obs     =
953      371
954      -----+----- F(14, 356)     =
955 19.13
956      Model |   .024491123        14   .001749366  Prob > F     =
957 0.0000
958      Residual |   .032550038       356   .000091433 R-squared     =
959 0.4294
960      -----+----- Adj R-squared     =
961 0.4069
962      Total |   .057041161       370   .000154165 Root MSE     =
963 .00956
964
965

```

```

958      -----
959      D.lnTotal |      Coef.    Std. Err.      t      P>|t|      [95% Conf.
960      Interval]
961      -----
962      lnTotal |
963      LD. |   -.0016173   .0529806   -0.03   0.976   -.1058116
964      .1025769
965      L2D. |   -.1325018   .0525096   -2.52   0.012   -.2357698
966      -.0292338
967      L3D. |    .026173   .0529824    0.49   0.622   -.0780247
968      .1303708
969      |
970      m1 |   -.0254857   .0024679   -10.33  0.000   -.0303391
971      -.0206322
972      m2 |    .0004869   .0029045    0.17   0.867   -.0052252
973      .006199
974      m3 |   -.003913   .0027686   -1.41   0.158   -.0093579
975      .0015318
976      m4 |   -.013317   .0026543   -5.02   0.000   -.0185371
977      -.0080969
978      m5 |   -.0064632   .0026182   -2.47   0.014   -.0116124
979      -.0013141
980      m6 |   -.0088465   .0025835   -3.42   0.001   -.0139273
981      -.0037657
982      m7 |   -.0159741   .0025331   -6.31   0.000   -.0209559
983      -.0109924
984      m8 |   -.0051433   .0026374   -1.95   0.052   -.0103302
985      .0000435
986      m9 |   -.0096793   .0025635   -3.78   0.000   -.0147208
987      -.0046378
988      m10 |   -.0021472   .0025489   -0.84   0.400   -.00716
989      .0028657
990      m11 |    .0024587   .0024684    1.00   0.320   -.0023958
991      .0073131
992      _cons |    .0088125   .0018473    4.77   0.000   .0051795
993      .0124454
994      -----
995
996      161 . predict pd
997          (option xb assumed; fitted values)
998          (15 missing values generated)
999
1000     162 . gen pflcount=exp((rwrms^2)/2)*exp(l.lnTotal+pd) if Date==tm(2021m4)
1001          (386 missing values generated)
1002
1003

```

```

987      163 . gen ub1=exp((rwmse^2)/2)*exp(l.lnTotal+pd+1*rwmse) if
Date==tm(2021m4)
988          (386 missing values generated)
989
990      164 . gen lb1=exp((rwmse^2)/2)*exp(l.lnTotal+pd-1*rwmse) if
Date==tm(2021m4)
991          (386 missing values generated)
992
993      165 . gen ub2=exp((rwmse^2)/2)*exp(l.lnTotal+pd+2*rwmse) if
Date==tm(2021m4)
994          (386 missing values generated)
995
996      166 . gen lb2=exp((rwmse^2)/2)*exp(l.lnTotal+pd-2*rwmse) if
Date==tm(2021m4)
997          (386 missing values generated)
998
999      167 . gen ub3=exp((rwmse^2)/2)*exp(l.lnTotal+pd+3*rwmse) if
Date==tm(2021m4)
1000          (386 missing values generated)
1001
1002      168 . gen lb3=exp((rwmse^2)/2)*exp(l.lnTotal+pd-3*rwmse) if
Date==tm(2021m4)
1003          (386 missing values generated)
1004
1005      169 . drop pd
1006
1007      170 .
1008      171 . replace pflcount=Total if Date==tm(2021m3)
1009          (1 real change made)
1010
1011      172 . replace ub1=Total if Date==tm(2021m3)
1012          (1 real change made)
1013
1014      173 . replace ub2=Total if Date==tm(2021m3)
1015          (1 real change made)
1016
1017      174 . replace ub3=Total if Date==tm(2021m3)
1018          (1 real change made)
1019
1020      175 . replace lb1=Total if Date==tm(2021m3)
1021          (1 real change made)
1022
1023      176 . replace lb2=Total if Date==tm(2021m3)
1024          (1 real change made)
1025
1026      177 . replace lb3=Total if Date==tm(2021m3)
1027          (1 real change made)
1028

```

```

1029      178 .
1030      179 . twoway (tsrline ub3 ub2 if tin(2020m4,2021m4), ///
1031          >         recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
1032          >         (tsrline ub2 ub1 if tin(2020m4,2021m4), ///
1033          >         recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
1034          >         (tsrline ub1 pflcount if tin(2020m4,2021m4), ///
1035          >         recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
1036          >         (tsrline pflcount lb1 if tin(2020m4,2021m4), ///
1037          >         recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
1038          >         (tsrline lb1 lb2 if tin(2020m4,2021m4), ///
1039          >         recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
1040          >         (tsrline lb2 lb3 if tin(2020m4,2021m4), ///
1041          >         recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
1042          >         (tsline Total pflcount if tin(2020m4,2021m4) , ///
1043          >         lcolor(gs12 pink) lwidth(medthick medthick) ///
1044          >         lpattern(solid longdash)), scheme(s1mono) legend(off)
1045
1046      180 . graph export "TotalFan1.png", replace
1047          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Problem Sets
1048          > /Final Exam/TotalFan1.png written in PNG format)
1049
1050      181 .
1051      182 . * More than 1 step
1052      183 . arima d.lnTotal 1(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
tin(199
1053          > 0m1,2021m3)
1054
1055          (setting optimization to BHHH)
1056          Iteration 0:    log likelihood =  1206.3625
1057          Iteration 1:    log likelihood =  1206.3625
1058
1059          ARIMA regression
1060
1061          Sample:  1990m5 - 2021m3                               Number of obs      =
1062          371                                              Wald chi2(14)      =
1063          83.94                                         Prob > chi2      =
1064          Log likelihood =  1206.362                         0.0000
1065          -----
1066          -----
1067          D.lnTotal |      Coef.     Std. Err.          z      P>|z|      [95% Conf.
1068          Interval]
1069          -----
1070          -----

```

1069	lnTotal						
1070	lnTotal						
1071	LD.		-.0016173	.0448009	-0.04	0.971	-.0894254
	.0861907						
1072	L2D.		-.1325018	.0486537	-2.72	0.006	-.2278614
	-.0371422						
1073	L3D.		.026173	.0400229	0.65	0.513	-.0522705
	.1046166						
1074							
1075	m1		-.0254857	.0054401	-4.68	0.000	-.0361481
	-.0148232						
1076	m2		.0004869	.0068588	0.07	0.943	-.0129561
	.0139299						
1077	m3		-.003913	.0058695	-0.67	0.505	-.0154171
	.007591						
1078	m4		-.013317	.0047531	-2.80	0.005	-.0226329
	-.0040012						
1079	m5		-.0064632	.0082351	-0.78	0.433	-.0226038
	.0096773						
1080	m6		-.0088465	.0069016	-1.28	0.200	-.0223735
	.0046804						
1081	m7		-.0159741	.0056734	-2.82	0.005	-.0270938
	-.0048545						
1082	m8		-.0051433	.0074271	-0.69	0.489	-.0197002
	.0094135						
1083	m9		-.0096793	.0051505	-1.88	0.060	-.0197741
	.0004155						
1084	m10		-.0021472	.0048943	-0.44	0.661	-.0117399
	.0074456						
1085	m11		.0024587	.0066745	0.37	0.713	-.0106231
	.0155405						
1086	_cons		.0088125	.0042033	2.10	0.036	.0005741
	.0170508						
1087		-----+-----					

1088	/sigma		.0093667	.0001378	67.96	0.000	.0090966
	.0096369						
1089		-----					

1090	Note:	The test of the variance against zero is one sided, and the two-sided					
1091		confidence interval is truncated at zero.					
1092							
1093	184	. predict pnonfarm, dynamic(tm(2021m3))					
1094		(option xb assumed; predicted values)					
1095		(4 missing values generated)					
1096							
1097	185	. predict mse, mse dynamic(mofd(tm(2021m4)))					

```
1098  
1099     186 . gen totmse = mse if Date==tm(2021m4)  
1100         (386 missing values generated)  
1101  
1102     187 . replace totmse = l.totmse+mse if Date>tm(2021m4)  
1103         (11 real changes made)  
1104  
1105     188 . gen pnonfarma = Total if Date==tm(2021m3)  
1106         (386 missing values generated)  
1107  
1108     189 . replace pnonfarma = l.pnonfarma*exp(pnonfarm+mse/2) if Date>tm(2021m3)  
1109         (12 real changes made)  
1110  
1111     190 .  
1112     191 . gen ub1a = pnonfarma*exp(totmse^.5)  
1113         (375 missing values generated)  
1114  
1115     192 . gen ub2a = pnonfarma*exp(2*totmse^.5)  
1116         (375 missing values generated)  
1117  
1118     193 . gen ub3a = pnonfarma*exp(3*totmse^.5)  
1119         (375 missing values generated)  
1120  
1121     194 . gen lb1a = pnonfarma/exp(totmse^.5)  
1122         (375 missing values generated)  
1123  
1124     195 . gen lb2a = pnonfarma/exp(2*totmse^.5)  
1125         (375 missing values generated)  
1126  
1127     196 . gen lb3a = pnonfarma/exp(3*totmse^.5)  
1128         (375 missing values generated)  
1129  
1130     197 .  
1131     198 . replace ub1a=Total if Date == tm(2021m3)  
1132         (1 real change made)  
1133  
1134     199 . replace ub2a=Total if Date == tm(2021m3)  
1135         (1 real change made)  
1136  
1137     200 . replace ub3a=Total if Date == tm(2021m3)  
1138         (1 real change made)  
1139  
1140     201 . replace lb1a=Total if Date == tm(2021m3)  
1141         (1 real change made)  
1142  
1143     202 . replace lb2a=Total if Date == tm(2021m3)  
1144         (1 real change made)  
1145
```

```

1146    203 . replace lb3a=Total if Date == tm(2021m3)
1147          (1 real change made)
1148
1149    204 .
1150    205 . twoway (tsrline ub3a ub2a if tin(2019m1,2022m3), ///
1151        >      recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
1152        >      (tsrline ub2a ub1a if tin(2019m1,2022m3), ///
1153        >      recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
1154        >      (tsrline ub1a pnonfarma if tin(2019m1,2022m3), ///
1155        >      recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
1156        >      (tsrline pnonfarma lb1a if tin(2019m1,2022m3), ///
1157        >      recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
1158        >      (tsrline lb1a lb2a if tin(2019m1,2022m3), ///
1159        >      recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
1160        >      (tsrline lb2a lb3a if tin(2019m1,2022m3), ///
1161        >      recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
1162        >      (tsline Total pnonfarma if tin(2019m1,2022m3) , ///
1163        >      lcolor(gs12 pink) lwidth(medthick medthick) ///
1164        >      lpattern(solid longdash)) , scheme(slmono) legend(off)
1165
1166    206 . graph export "TotalFan12.png", replace
1167          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Problem Sets
1168          > /Final Exam/TotalFan12.png written in PNG format)
1169
1170    207 .
1171    208 . scalar rmse_mod1 = .0132376
1172
1173    209 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
tin(1990m
1174          > 1,2021m3)
1175
1176          Source |      SS           df          MS       Number of obs     =
1177          371
1178          -----+----- F(14, 356)     =
1179          19.13
1180          Model |   .024491123      14   .001749366  Prob > F     =
1181          0.0000
1182          Residual |   .032550038      356   .000091433  R-squared     =
1183          0.4294
1184          -----+----- Adj R-squared     =
1185          0.4069
1186          Total |   .057041161      370   .000154165  Root MSE     =
1187          .00956
1188
1189          -----

```

	D.lnTotal		Coef.	Std. Err.	t	P> t	[95% Conf.
Interval]							
1185	-----+-----	-----					

1186	lnTotal						
1187	LD.		-.0016173	.0529806	-0.03	0.976	-.1058116
	.1025769						
1188	L2D.		-.1325018	.0525096	-2.52	0.012	-.2357698
	-.0292338						
1189	L3D.		.026173	.0529824	0.49	0.622	-.0780247
	.1303708						
1190							
1191	m1		-.0254857	.0024679	-10.33	0.000	-.0303391
	-.0206322						
1192	m2		.0004869	.0029045	0.17	0.867	-.0052252
	.006199						
1193	m3		-.003913	.0027686	-1.41	0.158	-.0093579
	.0015318						
1194	m4		-.013317	.0026543	-5.02	0.000	-.0185371
	-.0080969						
1195	m5		-.0064632	.0026182	-2.47	0.014	-.0116124
	-.0013141						
1196	m6		-.0088465	.0025835	-3.42	0.001	-.0139273
	-.0037657						
1197	m7		-.0159741	.0025331	-6.31	0.000	-.0209559
	-.0109924						
1198	m8		-.0051433	.0026374	-1.95	0.052	-.0103302
	.0000435						
1199	m9		-.0096793	.0025635	-3.78	0.000	-.0147208
	-.0046378						
1200	m10		-.0021472	.0025489	-0.84	0.400	-.00716
	.0028657						
1201	m11		.0024587	.0024684	1.00	0.320	-.0023958
	.0073131						
1202	_cons		.0088125	.0018473	4.77	0.000	.0051795
	.0124454						
1203	-----+-----	-----					

1204							
1205	210	.	predict plTotal				
1206		(option xb assumed; fitted values)					
1207		(15 missing values generated)					
1208							
1209	211	.	predict temp if tin(2021m3,2021m3)				
1210		(option xb assumed; fitted values)					
1211		(386 missing values generated)					
1212							
1213	212	.	replace plTotal=temp if tin(2021m3,2021m3)				

```

1214      (0 real changes made)
1215
1216 213 . drop temp
1217
1218 214 . gen pTotal=exp(l.lnTotal+plTotal+(rmse_mod1^2)/2)
1219      (15 missing values generated)
1220
1221 215 . gen lbTotal=exp(l.lnTotal+plTotal-1.96*rmse_mod1+(rmse_mod1^2)/2)
1222      (15 missing values generated)
1223
1224 216 . gen ubTotal=exp(l.lnTotal+plTotal+1.96*rmse_mod1+(rmse_mod1^2)/2)
1225      (15 missing values generated)
1226
1227 217 .
1228 218 . gen res=(d.lnTotal-plTotal)
1229      (16 missing values generated)
1230
1231 219 . gen expres=exp(res)
1232      (16 missing values generated)
1233
1234 220 . summ expres
1235
1236      Variable |       Obs        Mean      Std. Dev.       Min       Max
1237      -----+
1238      expres |       371     1.000042     .0088821     .8565838    1.028623
1239
1240 221 . scalar meanexpres=r(mean)
1241
1242 222 . gen epTotal=exp(l.lnTotal+plTotal)*meanexpres
1243      (15 missing values generated)
1244
1245 223 . _pctile res, percentile(2.5,97.5)
1246
1247 224 . return list
1248
1249      scalars:
1250          r(r1) =  -.0089262239634991
1251          r(r2) =   .0108835604041815
1252
1253 225 . gen elbTotal=exp(l.lnTotal+plTotal+r(r1))*meanexpres
1254      (15 missing values generated)
1255
1256 226 . gen eubTotal=exp(l.lnTotal+plTotal+r(r2))*meanexpres
1257      (15 missing values generated)
1258
1259 227 .
1260 228 . tsline Total pTotal elbTotal eubTotal lbTotal ubTotal if
tin(2019m1,2021m4),

```

```
1261      > ///
1262      >     scheme(slmono) tline(2021m3, lcolor(gs4)) ///
1263      >     lpattern(solid solid longdash longdash shortdash shortdash) ///
1264      >     lcolor(dkorange gs5 gs10 gs10 dkorange%60 dkorange%60) ///
1265      >     linewidth(medthick medthick medium medium)
1266
1267      229 . graph export "interval_tsline.png", replace
1268          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
1269          Series/Problem Sets
1270              > /Final Exam/interval_tsline.png written in PNG format)
1271
1272      230 .
1273      231 . histogram expres, normal kdensity saving(residuals.gph, replace)
1274          (bin=19, start=.85658383, width=.00905468)
1275          (file residuals.gph saved)
1276
1277      232 . graph export "residuals.png", replace
1278          (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
1279          Series/Problem Sets
1280              > /Final Exam/residuals.png written in PNG format)
1281
1282      233 .
1283      234 . log close
1284          name: <unnamed>
1285          log: /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
1286          Series/Probl
1287              > em Sets/Final Exam/Final Exam.smcl
1288                  log type: smcl
1289                  closed on: 29 Apr 2021, 12:09:27
1290
-----  
-----
```