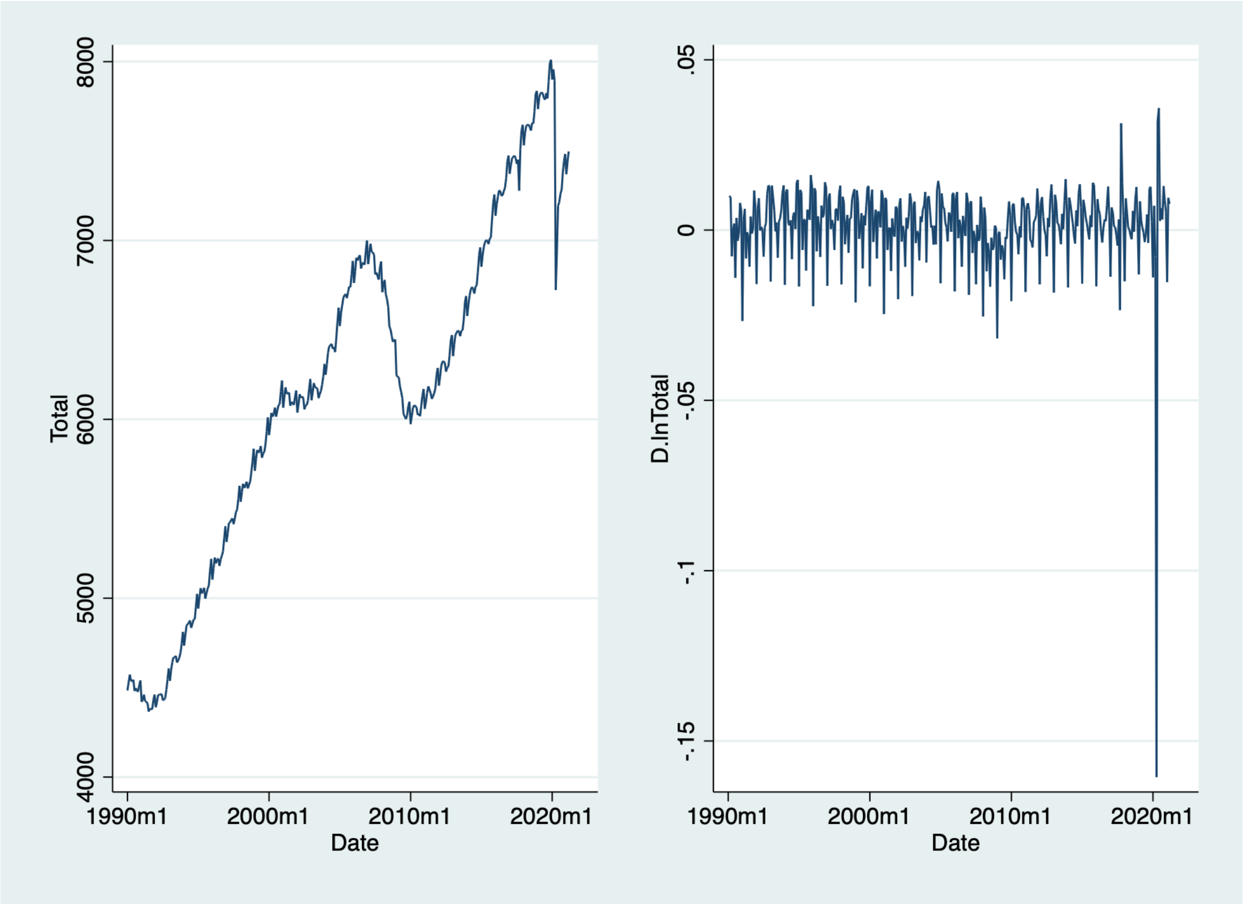
Part 1

Time Series Plots

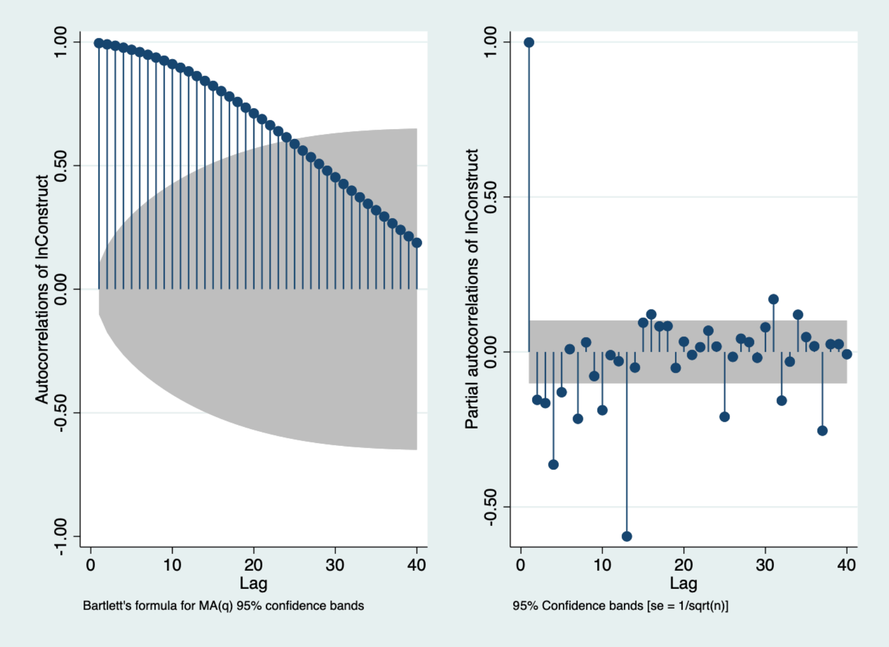


No Logs

You do not want to take logs because that adjusts the distance between values and normalizes the data. Without the distances between data being preserved, there is no good way to measure proportions unless the log transformed data has identical transforms performed (the data going in would have to be the same).

Persistent Data

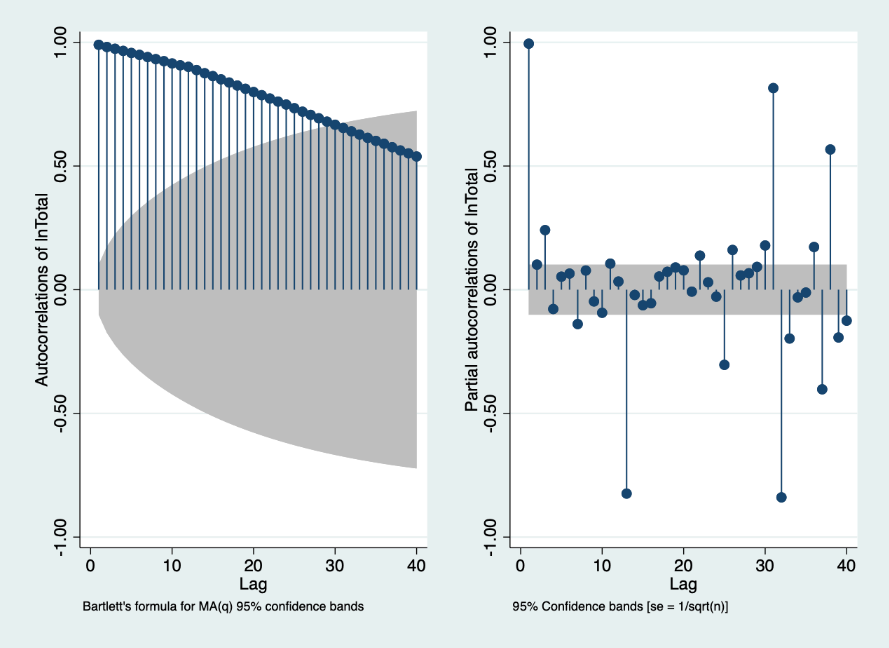
Construct



Both first lags are high which means we should difference.

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

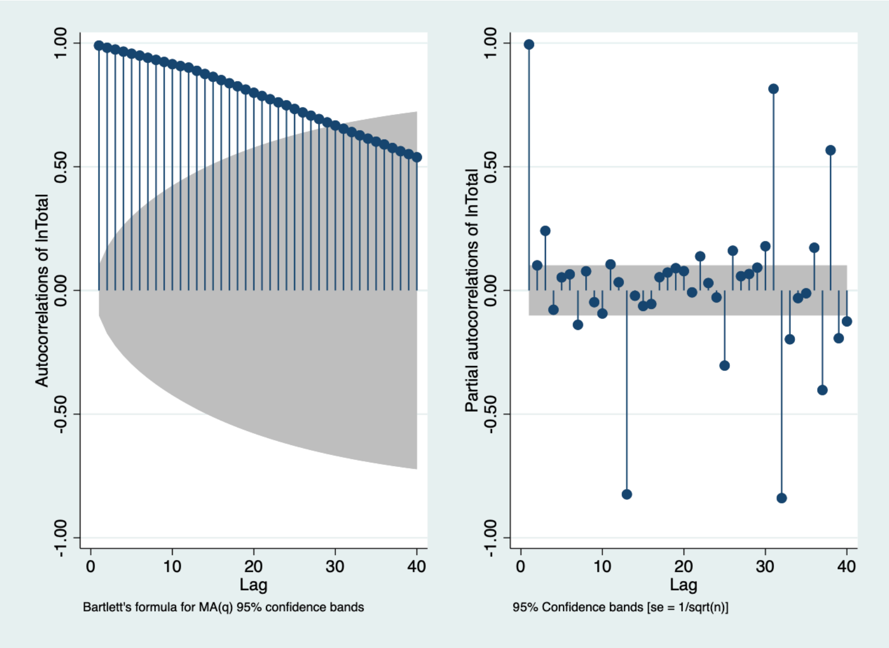
Manufacture



Both first lags are high which means we should difference.

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

Total

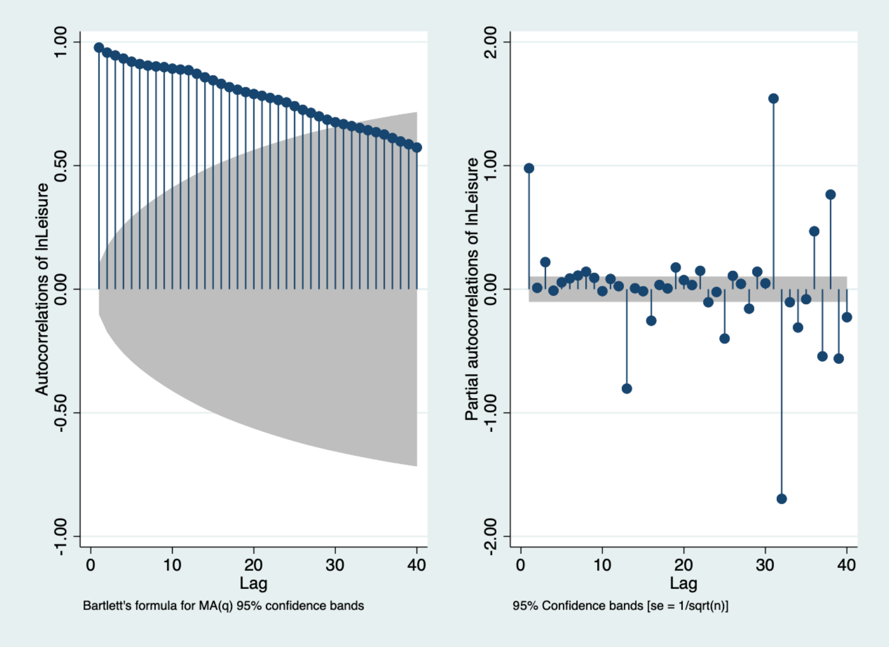


Both first lags are high which means we should difference.

Test	Result
MacKinnon approximate p-value for Z(t)	0.6570

Even though the dickey fuller test says we do not need to difference for construct, manufacture, and total, I'm still going to because ~~I already wrote everything with it differenced~~ the lags outside the confidence intervals in the PAC quite far apart and I can't imagine that there is really that much of an impact on current employment by employment from 30+ months ago. The only ones I can maybe understand are the 12 and 24 moth lags because employers will use historical data when hiring seasonal employees.

Leisure



Both first lags are high which means we should difference.

Test	Result
MacKinnon approximate p-value for Z(t)	0.0005

The p value for leisure is below .05 so I don't need an excuse to difference this time. However, it must be noted that lags 31 and 32 are extremely high.

Serial Correlation

Because not all lags are within the 95% confidence interval, we should worry at least a little bit about serial correlation. However, as I suggested earlier, the only lags that could have a true impact are the 12th and 24th lags and so I am not worried about serial correlation.

Dropping Lags

	Newey-West					
D.lnTotal	Coef.	Std. Err.	t	P>t	[95% Conf.	Interval]
Construct						
D1.	.0004252	.0000809	5.26	0.000	.0002661	.0005842
LD.	.0000247	.000044	0.56	0.574	-.0000617	.0001112
L2D.	.0000613	.0000403	1.52	0.129	-.0000179	.0001405
L3D.	-.0000674	.0000577	-1.17	0.244	-.000181	.0000462
L12D.	-.000035	.0000641	-0.55	0.585	-.000161	.000091
L24D.	.0000494	.0000684	0.72	0.470	-.0000851	.0001839
Manufacture						
D1.	.0011938	.0001924	6.21	0.000	.0008154	.0015723
LD.	-.0002374	.0001386	-1.71	0.088	-.00051	.0000353
L2D.	-.000336	.0000718	-4.68	0.000	-.0004773	-.0001948
L3D.	-.0002689	.0001059	-2.54	0.012	-.0004772	-.0000606
L12D.	.0003308	.0001381	2.39	0.017	.000059	.0006025
L24D.	.0002403	.0001413	1.70	0.090	-.0000376	.0005182
Leisure						
D1.	.0002134	8.38e-06	25.47	0.000	.0001969	.0002299
LD.	5.34e-06	9.19e-06	0.58	0.561	-.0000127	.0000234
L2D.	-2.42e-06	.0000172	-0.14	0.888	-.0000364	.0000315
L3D.	-8.76e-06	.0000174	-0.50	0.614	-.0000429	.0000254
L12D.	.0000629	.0000361	1.74	0.082	-8.04e-06	.0001339
L24D.	-.000016	.0000428	-0.37	0.708	-.0001002	.0000681
_cons	.0011957	.0003664	3.26	0.001	.000475	.0019165

The Newey-West test suggests we should drop lags 12 and 24.

Test Number	Variable		Test Value
(1)	L24D.Construct	=	0
(2)	L12D.Leisure	=	0
(3)	L24D.Leisure	=	0
(4)	L12D.Manufacture	=	0
(5)	L24D.Manufacture	=	0

`testparm` suggests that there is no interaction between the 12th and 24th lags of each construct, leisure, and manufacture.

Equal Effects

Interaction	Prob > F
Construct ~ Manufacture	0.0161
Construct ~ Leisure	0.0008
Manufacture ~ Leisure	0.0301

The p-value for all three is less than 0.05 which means we can reject the null hypothesis and accept the alternative that the impacts of each variable are not equal.

Part 2

Differencing, Log Transforms, and Month Dummies

Differencing

See above AC and PAC charts in Part 1.

Log Transforms

Log transforms make the data not have any values less than zero and forces the data into a normal distribution. It also transforms the data so it has proportional changes rather than absolute changes so that any changes over time can be reported as a percent change.

Month Dummies

There's not any reason to not include month dummies. If your data is monthly or any other form of seasonal, it will help your models because they're now identified to a particular season. If your data isn't seasonal, they won't have any effect.

Content Knowledge and Model Searches

Content Knowledge

Content knowledge can speed up the model selection process because you may already have an idea of what variables or lags have an effect on the dependent variable. For example, hourly wages and hours scheduled per week are probably a very good indication of monthly wages.

GSREG

Global search regression takes all the variables you feed it and runs a regression for any combination of the variables. This is a powerful tool to fine-tune your models, but without filtering the variables through content knowledge, it could take a very long time to run. Rather than just taking the highest scoring model, you should then examine common features of the highest scoring models on the basis of AIC, BIC, and out of sample root mean square error, and choose the most parsimonious one.

What's wrong with *stepwise* model selection?

It's prone to over fitting because it has bad predictive properties. Instead you should use out of sample fitting because it protects against over fitting. Over fitting is caused by dropping the most insignifcant each step which may include variables that should be included in the model but are not relevant on their own.

Choosing Models

Model Type	Model	AIC	BIC	Root Mean Squared Errors
AR only Lags 1-3 Month dummies	`reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	-2382.725	-2323.982	.0129605
AR only Lags 1-3,12,24 Month dummies	`reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	-2236.218	-2170.633	.01300797
ARDL Lags 1-3 Month dummies	`reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	-2393.008	-2299.019	.01611154
ARDL Lags 1-3,12,24 Month dummies	`reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11`	-2230.781	-2115.043	.01713897

Which are the best two and why?

Model 1 has the lowest root mean squared error and model 3 has the lowest AIC and BIC. I'm also inclined to believe these are the better ones because they don't include lags 12 and 24 which is a long time for subcomponents of the total employment variable to have an effect on the total employment variable.

Rolling Window

Model 1

```
reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
```

Value Type	Value
RWmaxobs12	12
RWminobs12	12
RWrmse12	.0132376

A window width of 12 had the lowest RWrmse. I thought that maybe a smaller window width would be better because the lags did not include lag 12 but I was wrong. Besides 12 months, 6 months had the second lowest.

Model 3

```
reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure
l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
```

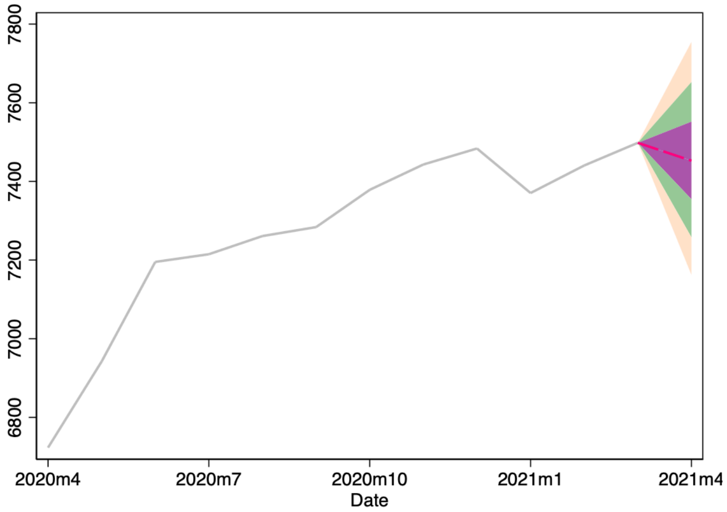
Value Type	Value
RWmaxobs12	12
RWminobs12	12
RWrmse12	.0132376

A window width of 12 had the lowest RWrmse. After my failure in model 1, I tried again hoping for better results. A window width of 12 is still the best.

Ultimately, I'm going to choose model 1 because it is autoregressive and that is what makes ARIMA work and without ARIMA I could not make my pretty fan charts. They have the same RWrmse anyways so I can't imagine the extra variables have too big a difference. And

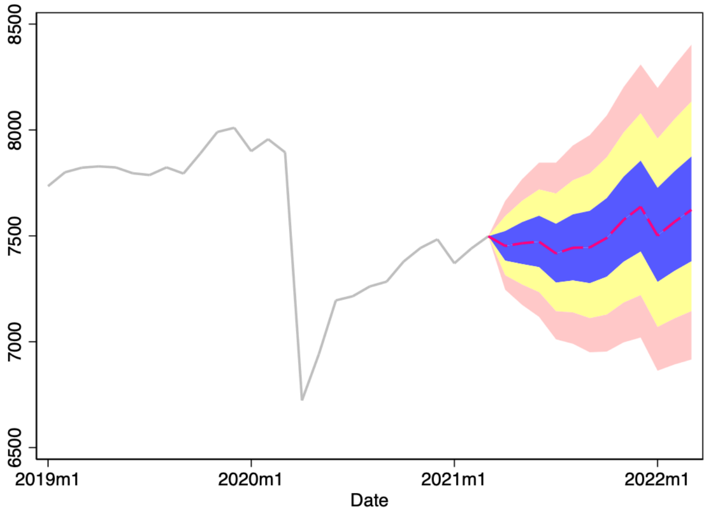
Forecasting

One month ahead



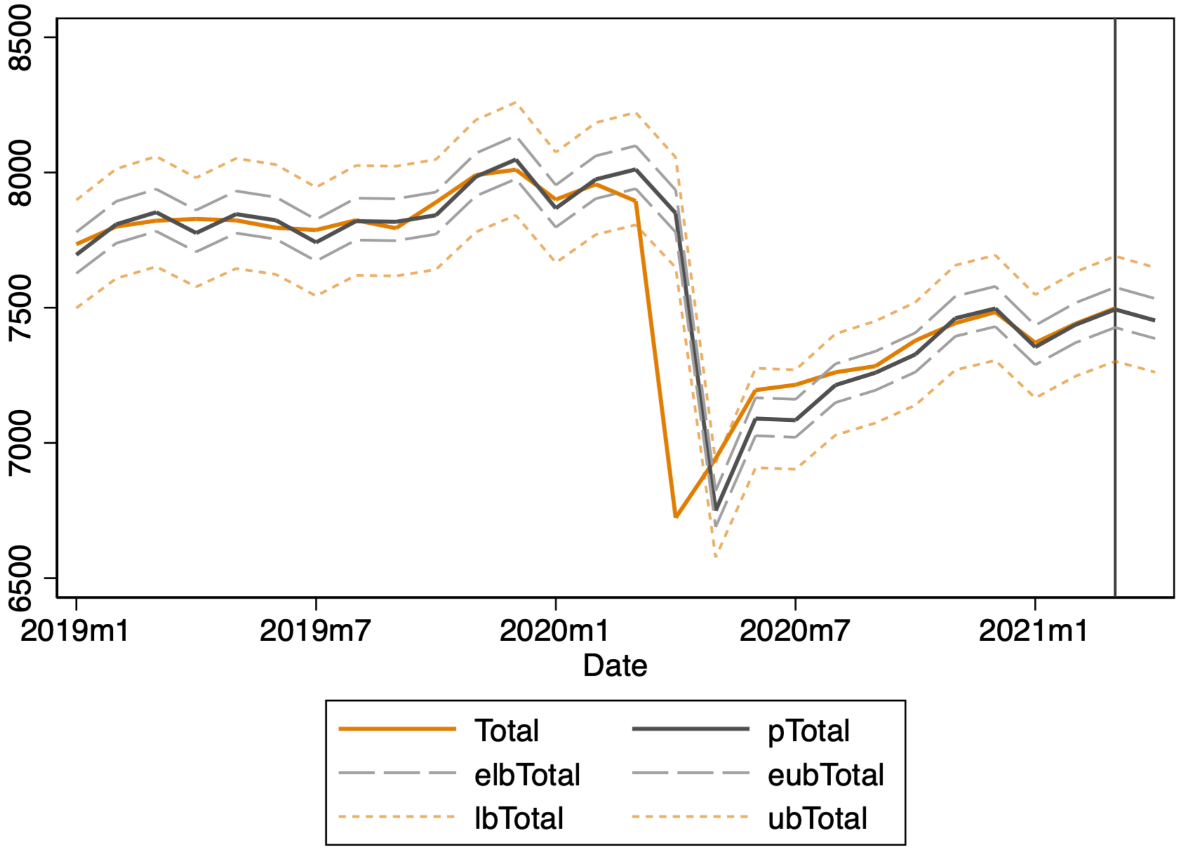
> Sorry for the bad colors. Hailey peer pressured me into it.

One year out

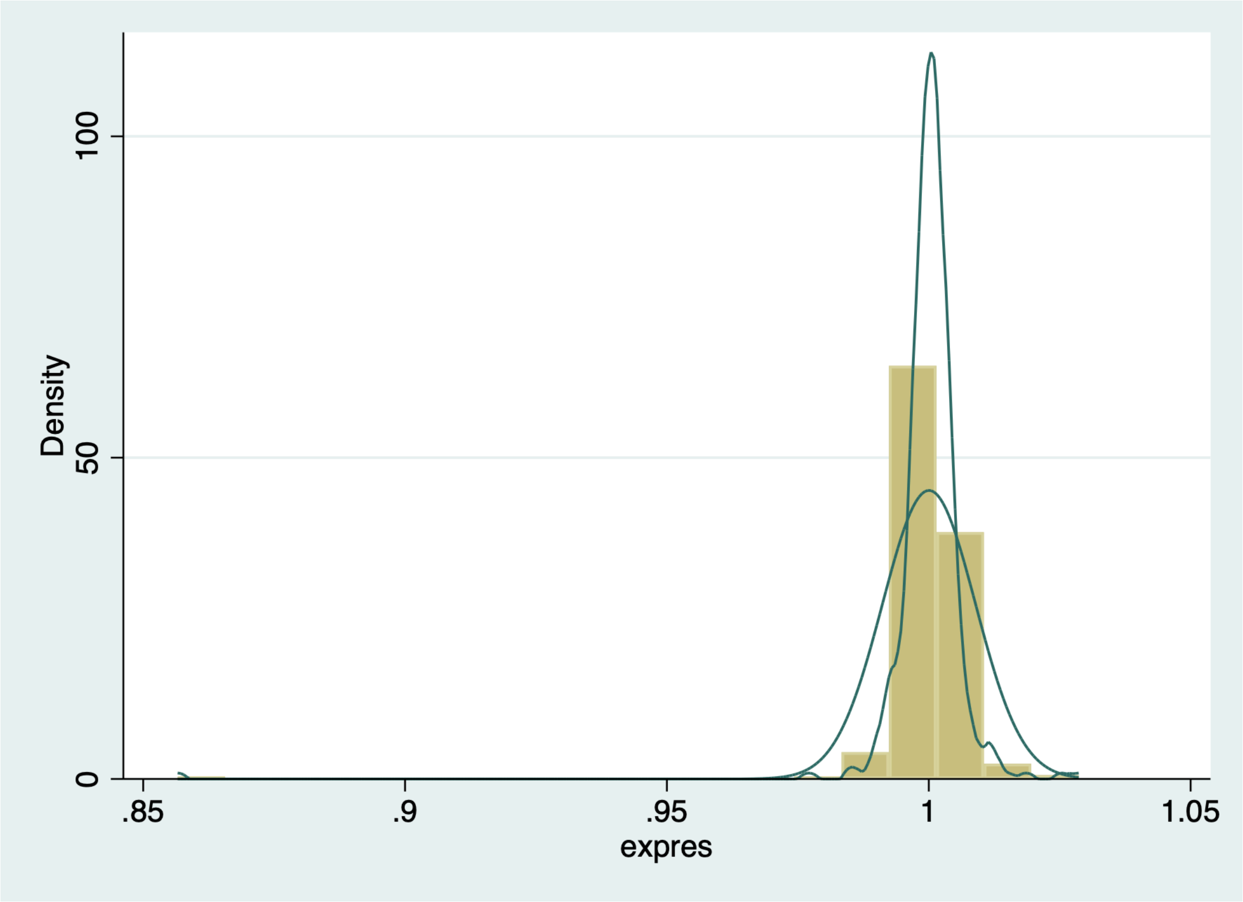


> Ditto my earlier comment on the colors :)

Forecast Evaluation



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



Residuals are good. Nice and normal.

Chebyshev

I think I have a basic understanding of Chebyshev. It's just a super general way of estimating what X portion of your population can be X distance away from the mean. In theory, this should hold even if it's not normal. That second part's key because it means we can use it on non-log transformed data.

Do File

```
clear
set more off

cd "/Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets/Final Exam"
log using "Final Exam.smcl", replace
import delimited "SP21Final.csv"
```

```

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

gen lnConstruct = ln(construct)
gen lnLeisure = ln(leisure)
gen lnManufacture = ln(manufacture)
gen lnTotal = ln(total)

gen Total = total
gen Construct = construct
gen Leisure = leisure
gen Manufacture = manufacture

/*
gen withMarchTotal = Total
replace Total=. if tin(2021m3,)
*/

tsset Date
tsappend, add(12)
replace month=month(dofm(Date))

gen m1=0
replace m1=1 if month==1
gen m2=0
replace m2=1 if month==2
gen m3=0
replace m3=1 if month==3
gen m4=0
replace m4=1 if month==4
gen m5=0
replace m5=1 if month==5
gen m6=0
replace m6=1 if month==6
gen m7=0
replace m7=1 if month==7
gen m8=0
replace m8=1 if month==8
gen m9=0
replace m9=1 if month==9
gen m10=0
replace m10=1 if month==10
gen m11=0
replace m11=1 if month==11
```

```


summ construct leisure manufacture total
summ lnConstruct lnLeisure lnManufacture lnTotal

tsline lnConstruct lnLeisure, saving(lnConstructLeisure_tsline.gph, replace)
tsline lnManufacture, saving(lnManufacture_tsline.gph, replace)
graph combine lnConstructLeisure_tsline.gph lnManufacture_tsline.gph, ///
  saving(lnConstructLeisure, replace)
graph export "lnConstructLeisure-Manufacture_tsline.png", replace

tsline lnTotal, saving(lnTotal_tsline.gph, replace)
tsline d.lnTotal, saving(dlnTotal_tsline.gph, replace)
graph combine Total_tsline.gph dlnTotal_tsline.gph, saving(lnTotal-Total, replace)
graph export "lnTotal-dlnTotal_tsline.png", replace

ac lnTotal, saving(lnTotal_ac, replace)
pac lnTotal, saving(lnTotal_pac, replace)
graph combine lnTotal_ac.gph lnTotal_pac.gph, saving(lnTotal_ac_pac, replace)
graph export "lnTotal_ac_pac.png", replace
dfuller lnTotal, trend regress

ac lnConstruct, saving(lnConstruct_ac, replace)
pac lnConstruct, saving(lnConstruct_pac, replace)
graph combine lnConstruct_ac.gph lnConstruct_pac.gph, saving(lnConstruct_ac_pac, replace)
graph export "lnConstruct_ac_pac.png", replace
dfuller lnConstruct, trend regress

ac lnLeisure, saving(lnLeisure_ac, replace)
pac lnLeisure, saving(lnLeisure_pac, replace)
graph combine lnLeisure_ac.gph lnLeisure_pac.gph, saving(lnLeisure_ac_pac, replace)
graph export "lnLeisure_ac_pac.png", replace
dfuller lnLeisure, trend regress

ac lnManufacture, saving(lnManufacture_ac, replace)
pac lnManufacture, saving(lnManufacture_pac, replace)
graph combine lnManufacture_ac.gph lnManufacture_pac.gph, saving(lnManufacture_ac_pac, replace)
graph export "lnManufacture_ac_pac.png", replace
dfuller lnManufacture, trend regress

quietly reg l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture
testparm l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture

```

```
newey d.lnTotal l(0/3,12,24)d.Construct l(0/3,12,24)d.Manufacture l(0/3,12,24)d.Leisure, lag(24)
test ld.Construct + ld.Construct + l2d.Construct + l3d.Construct + l12d.Construct + l24d.Construct ///
  == d.Manufacture + ld.Manufacture + l2d.Manufacture + l3d.Manufacture + ///
    l12d.Manufacture + l24d.Manufacture
test d.Construct + ld.Construct + l2d.Construct + l3d.Construct + l12d.Construct + l24d.Construct ///
  == d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure + l12d.Leisure + l24d.Leisure
test d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure + l12d.Leisure + l24d.Leisure ///
  == d.Manufacture + ld.Manufacture + l2d.Manufacture + l3d.Manufacture + ///
    l12d.Manufacture + l24d.Manufacture

*------------------------------------------------------------------------------*
gen dlnConstruct=d.lnConstruct
gen l1dlnConstruct=l1d.lnConstruct
gen l2dlnConstruct=l2d.lnConstruct
gen l3dlnConstruct=l3d.lnConstruct
gen l12dlnConstruct=l12d.lnConstruct
gen l24dlnConstruct=l24d.lnConstruct

gen dlnLeisure=d.lnLeisure
gen l1dlnLeisure=l1d.lnLeisure
gen l2dlnLeisure=l2d.lnLeisure
gen l3dlnLeisure=l3d.lnLeisure
gen l12dlnLeisure=l12d.lnLeisure
gen l24dlnLeisure=l24d.lnLeisure

gen dlnManufacture=d.lnManufacture
gen l1dlnManufacture=l1d.lnManufacture
gen l2dlnManufacture=l2d.lnManufacture
gen l3dlnManufacture=l3d.lnManufacture
gen l12dlnManufacture=l12d.lnManufacture
gen l24dlnManufacture=l24d.lnManufacture

gen dlnTotal=d.lnTotal
gen l1dlnTotal=l1d.lnTotal
gen l2dlnTotal=l2d.lnTotal
gen l3dlnTotal=l3d.lnTotal
gen l12dlnTotal=l12d.lnTotal
gen l24dlnTotal=l24d.lnTotal

/*
gsreg dlnTotal dlnConstruct l1dlnConstruct l2dlnConstruct l3dlnConstruct ///
  l12dlnConstruct l24dlnConstruct ///
  dlnLeisure l1dlnLeisure l2dlnLeisure l3dlnLeisure l12dlnLeisure l24dlnLeisure ///
  dlnManufacture l1dlnManufacture l2dlnManufacture l3dlnManufacture ///
  l12dlnManufacture l24dlnManufacture ///
  if tin(1990m1,2021m3), ///
  ncomb(1,6) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11) ///
  samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnTtoal) replace
*/


loocv reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
quietly reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
estat ic

loocv reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
quietly reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
estat ic

loocv reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
  l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
quietly reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
  l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
estat ic

loocv reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct ///
  l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
quietly reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct ///
  l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
estat ic

*Lowest rmse (1)
reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
scalar drop _all
quietly forval w=12(12)180 {
gen pred=.
gen nobs=.
  forval t=544/734 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
    if Date>=wstart & Date<=wend
  replace nobs=e(N) if Date==`t'
  predict ptemp
  replace pred=ptemp if Date==`t'
  drop ptemp wstart wend
  }
gen errsq=(pred-d.lnTotal)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list

/*
RWmaxobs12 =         12
RWminobs12 =         12
RWrmse12 =   .0132376
*/

*lowest AIC and BIC (3)
reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
  l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11
scalar drop _all
quietly forval w=3(3)180 {
gen pred=.
gen nobs=.
  forval t=544/734 {
  gen wstart=`t'-`w'
  gen wend=`t'-1
  reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
    l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 ///
    if Date>=wstart & Date<=wend
  replace nobs=e(N) if Date==`t'
  predict ptemp
  replace pred=ptemp if Date==`t'
  drop ptemp wstart wend
  }
gen errsq=(pred-d.lnTotal)^2
summ errsq
scalar RWrmse`w'=r(mean)^.5
summ nobs
scalar RWminobs`w'=r(min)
scalar RWmaxobs`w'=r(max)
drop errsq pred nobs
}
scalar list
/*
RWmaxobs12 =         12
RWminobs12 =         12
RWrmse12 =   .0132376
*/
```

```
* Going with model 1 because average RWrmse is lower across window sizes
scalar rwrmse = .0132376
reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(,2021m3)
predict pd
gen pflcount=exp((rwrmse^2)/2)*exp(l.lnTotal+pd) if Date==tm(2021m4)
gen ub1=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+1*rwrmse) if Date==tm(2021m4)
gen lb1=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-1*rwrmse) if Date==tm(2021m4)
gen ub2=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+2*rwrmse) if Date==tm(2021m4)
gen lb2=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-2*rwrmse) if Date==tm(2021m4)
gen ub3=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+3*rwrmse) if Date==tm(2021m4)
gen lb3=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-3*rwrmse) if Date==tm(2021m4)
drop pd

replace pflcount=Total if Date==tm(2021m3)
replace ub1=Total if Date==tm(2021m3)
replace ub2=Total if Date==tm(2021m3)
replace ub3=Total if Date==tm(2021m3)
replace lb1=Total if Date==tm(2021m3)
replace lb2=Total if Date==tm(2021m3)
replace lb3=Total if Date==tm(2021m3)

twoway (tsrline ub3 ub2 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
  (tsrline ub2 ub1 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
  (tsrline ub1 pflcount if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
  (tsrline pflcount lb1 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
  (tsrline lb1 lb2 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
  (tsrline lb2 lb3 if tin(2020m4,2021m4), ///
  recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
  (tsline Total pflcount if tin(2020m4,2021m4) , ///
  lcolor(gs12 pink) lwidth(medthick medthick) ///
  lpattern(solid longdash)), scheme(s1mono) legend(off)
graph export "TotalFan1.png", replace

* More than 1 step
arima d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(1990m1,2021m3)
predict pnonfarm, dynamic(tm(2021m3))
predict mse, mse dynamic(mofd(tm(2021m4)))
gen totmse = mse if Date==tm(2021m4)
replace totmse = l.totmse+mse if Date>tm(2021m4)
gen pnonfarma = Total if Date==tm(2021m3)
```

```
replace pnonfarma = l.pnonfarma*exp(pnonfarm+mse/2) if Date>tm(2021m3)

gen ub1a = pnonfarma*exp(totmse^.5)
gen ub2a = pnonfarma*exp(2*totmse^.5)
gen ub3a = pnonfarma*exp(3*totmse^.5)
gen lb1a = pnonfarma/exp(totmse^.5)
gen lb2a = pnonfarma/exp(2*totmse^.5)
gen lb3a = pnonfarma/exp(3*totmse^.5)

replace ub1a=Total if Date == tm(2021m3)
replace ub2a=Total if Date == tm(2021m3)
replace ub3a=Total if Date == tm(2021m3)
replace lb1a=Total if Date == tm(2021m3)
replace lb2a=Total if Date == tm(2021m3)
replace lb3a=Total if Date == tm(2021m3)

twoway (tsrline ub3a ub2a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
  (tsrline ub2a ub1a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
  (tsrline ub1a pnonfarma if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
  (tsrline pnonfarma lb1a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
  (tsrline lb1a lb2a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
  (tsrline lb2a lb3a if tin(2019m1,2022m3), ///
  recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
  (tsline Total pnonfarma if tin(2019m1,2022m3) , ///
  lcolor(gs12 pink) lwidth(medthick medthick) ///
  lpattern(solid longdash)) , scheme(s1mono) legend(off)
graph export "TotalFan12.png", replace

scalar rmse_mod1 = .0132376
reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(1990m1,2021m3)
predict plTotal
predict temp if tin(2021m3,2021m3)
replace plTotal=temp if tin(2021m3,2021m3)
drop temp
gen pTotal=exp(l.lnTotal+plTotal+(rmse_mod1^2)/2)
gen lbTotal=exp(l.lnTotal+plTotal-1.96*rmse_mod1+(rmse_mod1^2)/2)
gen ubTotal=exp(l.lnTotal+plTotal+1.96*rmse_mod1+(rmse_mod1^2)/2)

gen res=(d.lnTotal-plTotal)
gen expres=exp(res)
summ expres
scalar meanexpres=r(mean)
```

```
gen epTotal=exp(l.lnTotal+plTotal)*meanexpres
_pctile res, percentile(2.5,97.5)
return list
gen elbTotal=exp(l.lnTotal+plTotal+r(r1))*meanexpres
gen eubTotal=exp(l.lnTotal+plTotal+r(r2))*meanexpres

tsline Total pTotal elbTotal eubTotal lbTotal ubTotal if tin(2019m1,2021m4), ///
  scheme(s1mono) tline(2021m3, lcolor(gs4)) ///
  lpattern(solid solid longdash longdash shortdash shortdash) ///
  lcolor(dkorange gs5 gs10 gs10 dkorange%60 dkorange%60) ///
  lwidth(medthick medthick medium medium)
graph export "interval_tsline.png", replace

histogram expres, normal kdensity saving(residuals.gph, replace)
graph export "residuals.png", replace

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

Log File

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


      -------------------------------------------------------------------------------
            name:  <unnamed>
             log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Probl
      > em Sets/Final Exam/Final Exam.smcl
        log type:  smcl
       opened on:  29 Apr 2021, 12:07:09

     1 . import delimited "SP21Final.csv"
      (5 vars, 375 obs)

     2 .
     3 . gen datec=date(date, "YMD")

     4 . gen Date=mofd(datec)
```

```

  5 . gen month=month(datec)

  6 . format Date %tm

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

  8 .
  9 . gen lnConstruct = ln(construct)

 10 . gen lnLeisure = ln(leisure)

 11 . gen lnManufacture = ln(manufacture)

 12 . gen lnTotal = ln(total)

 13 .
 14 . gen Total = total

 15 . gen Construct = construct

 16 . gen Leisure = leisure

 17 . gen Manufacture = manufacture

 18 .
 19 . /*
   > gen withMarchTotal = Total
   > replace Total=. if tin(2021m3,)
   > */
 20 .
 21 . tsset Date
          time variable:  Date, 1990m1 to 2021m3
                  delta:  1 month

 22 . tsappend, add(12)

 23 . replace month=month(dofm(Date))
   (12 real changes made)

 24 .
 25 . gen m1=0

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

```

```
    27 . gen m2=0

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

    29 . gen m3=0

    30 . replace m3=1 if month==3
      (33 real changes made)

    31 . gen m4=0

    32 . replace m4=1 if month==4
      (32 real changes made)

    33 . gen m5=0

    34 . replace m5=1 if month==5
      (32 real changes made)

    35 . gen m6=0

    36 . replace m6=1 if month==6
      (32 real changes made)

    37 . gen m7=0

    38 . replace m7=1 if month==7
      (32 real changes made)

    39 . gen m8=0

    40 . replace m8=1 if month==8
      (32 real changes made)

    41 . gen m9=0

    42 . replace m9=1 if month==9
      (32 real changes made)

    43 . gen m10=0

    44 . replace m10=1 if month==10
      (32 real changes made)

    45 . gen m11=0

    46 . replace m11=1 if month==11
```

```
      (32 real changes made)

    47 .
    48 .
    49 . summm construct leisure manufacture total

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
         construct |        375    461.0043     95.8947      323.9      696.1
           leisure |        375    930.2083    159.6216      660.6     1287.5
       manufacture |        375    410.0496    63.14375      307.9      518.2
             total |        375    6161.164    958.2068     4366.1     8010.4

    50 . summ lnConstruct lnLeisure lnManufacture lnTotal

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
       lnConstruct |        375    6.112377    .2043883   5.780435   6.545493
         lnLeisure |        375    6.820994    .1695282   6.493148   7.160458
      lnManufact~e |        375    6.004071    .1577555   5.729775   6.250361
           lnTotal |        375     8.71332    .1619029   8.381625   8.988496

    51 .
    52 . tsline lnConstruct lnLeisure, saving(lnConstructLeisure_tsline.gph, replace)
      (file lnConstructLeisure_tsline.gph saved)

    53 . tsline lnManufacture, saving(lnManufacture_tsline.gph, replace)
      (file lnManufacture_tsline.gph saved)

    54 . graph combine lnConstructLeisure_tsline.gph lnManufacture_tsline.gph, ///
      >         saving(lnConstructLeisure, replace)
      (file lnConstructLeisure.gph saved)

    55 . graph export "lnConstructLeisure-Manufacture_tsline.png", replace
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
      > /Final Exam/lnConstructLeisure-Manufacture_tsline.png written in PNG format)

    56 .
    57 . tsline lnTotal, saving(lnTotal_tsline.gph, replace)
      (file lnTotal_tsline.gph saved)

    58 . tsline d.lnTotal, saving(dlnTotal_tsline.gph, replace)
      (file dlnTotal_tsline.gph saved)

    59 . graph combine Total_tsline.gph dlnTotal_tsline.gph, saving(lnTotal-Total, rep
      > lace)
      (file lnTotal-Total.gph saved)

    60 . graph export "lnTotal-dlnTotal_tsline.png", replace
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
      > /Final Exam/lnTotal-dlnTotal_tsline.png written in PNG format)

    61 .
    62 . ac lnTotal, saving(lnTotal_ac, replace)
      (file lnTotal_ac.gph saved)

    63 . pac lnTotal, saving(lnTotal_pac, replace)
      (file lnTotal_pac.gph saved)

    64 . graph combine lnTotal_ac.gph lnTotal_pac.gph, saving(lnTotal_ac_pac, replace)
      (file lnTotal_ac_pac.gph saved)

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

    66 . dfuller lnTotal, trend regress

      Dickey-Fuller test for unit root                   Number of obs   =     374

                                     ---------- Interpolated Dickey-Fuller ---------
                        Test         1% Critical       5% Critical      10% Critical
                     Statistic           Value             Value             Value
      ------------------------------------------------------------------------------
       Z(t)             -1.895            -3.985            -3.425            -3.130
      ------------------------------------------------------------------------------
      MacKinnon approximate p-value for Z(t) = 0.6570

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

```
         D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
           lnTotal |
               L1. |  -.0191916   .0101251    -1.90   0.059    -.0391013    .0007181
                   |
            _trend |   .0000223   .0000151     1.47   0.142    -7.48e-06    .0000521
             _cons |   .1644033    .085612     1.92   0.056    -.0039423    .3327489
      ------------------------------------------------------------------------------

    67 .
    68 . ac lnConstruct, saving(lnConstruct_ac, replace)
      (file lnConstruct_ac.gph saved)

    69 . pac lnConstruct, saving(lnConstruct_pac, replace)
      (file lnConstruct_pac.gph saved)

    70 . graph combine lnConstruct_ac.gph lnConstruct_pac.gph, saving(lnConstruct_ac_p
      > ac, replace)
      (file lnConstruct_ac_pac.gph saved)

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

    72 . dfuller lnConstruct, trend regress

      Dickey-Fuller test for unit root                   Number of obs   =       374

                                     ---------- Interpolated Dickey-Fuller ---------
                        Test         1% Critical       5% Critical      10% Critical
                     Statistic           Value             Value             Value
      ------------------------------------------------------------------------------
       Z(t)             -0.586            -3.985            -3.425            -3.130
      ------------------------------------------------------------------------------
      MacKinnon approximate p-value for Z(t) = 0.9796

      ------------------------------------------------------------------------------
      D.           |
       lnConstruct |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
       lnConstruct |
               L1. |  -.0021673   .0036962    -0.59   0.558    -.0094355    .0051009
                   |
            _trend |   5.05e-06   6.99e-06     0.72   0.470    -8.68e-06    .0000188
             _cons |   .0132628   .0221901     0.60   0.550    -.0303713    .0568969
      ------------------------------------------------------------------------------

    73 .
    74 . ac lnLeisure, saving(lnLeisure_ac, replace)
      (file lnLeisure_ac.gph saved)

    75 . pac lnLeisure, saving(lnLeisure_pac, replace)
      (file lnLeisure_pac.gph saved)

    76 . graph combine lnLeisure_ac.gph lnLeisure_pac.gph, saving(lnLeisure_ac_pac, re
      > place)
      (file lnLeisure_ac_pac.gph saved)

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

    78 . dfuller lnLeisure, trend regress

      Dickey-Fuller test for unit root                   Number of obs   =       374

                                     ---------- Interpolated Dickey-Fuller ---------
                        Test         1% Critical       5% Critical      10% Critical
```

```
                     Statistic           Value             Value             Value
      ------------------------------------------------------------------------------
       Z(t)             -4.787            -3.985            -3.425            -3.130
      ------------------------------------------------------------------------------
      MacKinnon approximate p-value for Z(t) = 0.0005

      ------------------------------------------------------------------------------
       D.lnLeisure |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
         lnLeisure |
               L1. |  -.1242368   .0259543    -4.79   0.000    -.1752728   -.0732007
                   |
            _trend |   .0001744   .0000407     4.28   0.000     .0000943    .0002545
             _cons |   .8156996   .1699707     4.80   0.000     .4814728    1.149926
      ------------------------------------------------------------------------------

    79 .
    80 . ac lnManufacture, saving(lnManufacture_ac, replace)
      (file lnManufacture_ac.gph saved)

    81 . pac lnManufacture, saving(lnManufacture_pac, replace)
      (file lnManufacture_pac.gph saved)

    82 . graph combine lnManufacture_ac.gph lnManufacture_pac.gph, saving(lnManufactur
      > e_ac_pac, replace)
      (file lnManufacture_ac_pac.gph saved)

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

    84 . dfuller lnManufacture, trend regress

      Dickey-Fuller test for unit root                   Number of obs   =       374
```

```

                                     ---------- Interpolated Dickey-Fuller ---------
                        Test         1% Critical       5% Critical      10% Critical
                     Statistic           Value             Value             Value
      ------------------------------------------------------------------------------
       Z(t)              0.313            -3.985            -3.425            -3.130
      ------------------------------------------------------------------------------
      MacKinnon approximate p-value for Z(t) = 0.9963

      -------------------------------------------------------------------------------
      D.            |
      lnManufacture |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      --------------+----------------------------------------------------------------
      lnManufacture |
                L1. |   .0013361   .0042655     0.31   0.754    -.0070515    .0097238
                    |
             _trend |   9.58e-06   6.23e-06     1.54   0.125    -2.67e-06    .0000218
              _cons |  -.0106361   .0265921    -0.40   0.689    -.0629262    .0416539
      -------------------------------------------------------------------------------

    85 .
    86 . quietly reg l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture

    87 . testparm l(12,24)d.Construct l(12,24)d.Leisure l(12,24)d.Manufacture

       ( 1)  L24D.Construct = 0
       ( 2)  L12D.Leisure = 0
       ( 3)  L24D.Leisure = 0
       ( 4)  L12D.Manufacture = 0
       ( 5)  L24D.Manufacture = 0

             F(  5,   356) =  146.28
                  Prob > F =    0.0000

    88 .
    89 . newey d.lnTotal l(0/3,12,24)d.Construct l(0/3,12,24)d.Manufacture l(0/3,12,24
      > )d.Leisure, lag(24)

      Regression with Newey-West standard errors      Number of obs     = 350
      maximum lag: 24                                 F( 18,       331) = 4461.04
                                                      Prob > F          = 0.0000

      ------------------------------------------------------------------------------
                         |             Newey-West
               D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
               Construct |
                     D1. |   .0004252   .0000809     5.26   0.000     .0002661 .0005842
                     LD. |   .0000247    .000044     0.56   0.574    -.0000617 .0001112
                    L2D. |   .0000613   .0000403     1.52   0.129    -.0000179 .0001405
                    L3D. |  -.0000674   .0000577    -1.17   0.244     -.000181 .0000462
                   L12D. |   -.000035   .0000641    -0.55   0.585     -.000161 .000091
                   L24D. |   .0000494   .0000684     0.72   0.470    -.0000851 .0001839
                         |
             Manufacture |
                     D1. |   .0011938   .0001924     6.21   0.000     .0008154 .0015723
                     LD. |  -.0002374   .0001386    -1.71   0.088      -.00051 .0000353
                    L2D. |   -.000336   .0000718    -4.68   0.000    -.0004773 -.0001948
                    L3D. |  -.0002689   .0001059    -2.54   0.012    -.0004772 -.0000606
                   L12D. |   .0003308   .0001381     2.39   0.017      .000059 .0006025
                   L24D. |   .0002403   .0001413     1.70   0.090    -.0000376 .0005182
                         |
                 Leisure |
                     D1. |   .0002134   8.38e-06    25.47   0.000     .0001969    .0002299
                     LD. |   5.34e-06   9.19e-06     0.58   0.561    -.0000127    .0000234
                    L2D. |  -2.42e-06   .0000172    -0.14   0.888    -.0000364    .0000315
                    L3D. |  -8.76e-06   .0000174    -0.50   0.614    -.0000429    .0000254
                   L12D. |   .0000629   .0000361     1.74   0.082    -8.04e-06    .0001339
                   L24D. |   -.000016   .0000428    -0.37   0.708    -.0001002    .0000681
                         |
                   _cons |   .0011957   .0003664     3.26   0.001      .000475    .0019165
            ------------------------------------------------------------------------------

          90 . test ld.Construct + ld.Construct + l2d.Construct + l3d.Construct + l12d.Const
            > ruct + l24d.Construct ///
            >         == d.Manufacture + ld.Manufacture + l2d.Manufacture + l3d.Manufacture
            >  + ///
            >                 l12d.Manufacture + l24d.Manufacture

             ( 1)  2*LD.Construct + L2D.Construct + L3D.Construct + L12D.Construct +
                   L24D.Construct - D.Manufacture - LD.Manufacture - L2D.Manufacture -
                   L3D.Manufacture - L12D.Manufacture - L24D.Manufacture = 0

                   F(  1,   331) =    5.85
                        Prob > F =    0.0161

          91 . test d.Construct + ld.Construct + l2d.Construct + l3d.Construct + l12d.Constr
            > uct + l24d.Construct ///
            >         == d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure + l12d.Leisure
            > + l24d.Leisure

             ( 1)  D.Construct + LD.Construct + L2D.Construct + L3D.Construct +
                   L12D.Construct + L24D.Construct - D.Leisure - LD.Leisure - L2D.Leisure -
                   L3D.Leisure - L12D.Leisure - L24D.Leisure = 0

                   F(  1,   331) =   11.36
                        Prob > F =    0.0008

          92 . test d.Leisure + ld.Leisure + l2d.Leisure + l3d.Leisure + l12d.Leisure + l24d
            > .Leisure ///
            >         == d.Manufacture + ld.Manufacture + l2d.Manufacture + l3d.Manufacture
            >  + ///
            >                 l12d.Manufacture + l24d.Manufacture

             ( 1)  - D.Manufacture - LD.Manufacture - L2D.Manufacture - L3D.Manufacture -
                   L12D.Manufacture - L24D.Manufacture + D.Leisure + LD.Leisure +
                   L2D.Leisure + L3D.Leisure + L12D.Leisure + L24D.Leisure = 0

                   F(  1,   331) =    4.74
                        Prob > F =    0.0301

          93 .
          94 . *-----------------------------------------------------------------------------
            > --*
          95 . gen dlnConstruct=d.lnConstruct
            (13 missing values generated)

          96 . gen l1dlnConstruct=l1d.lnConstruct
            (13 missing values generated)

          97 . gen l2dlnConstruct=l2d.lnConstruct
            (13 missing values generated)

          98 . gen l3dlnConstruct=l3d.lnConstruct
            (13 missing values generated)

          99 . gen l12dlnConstruct=l12d.lnConstruct
            (13 missing values generated)

         100 . gen l24dlnConstruct=l24d.lnConstruct
            (25 missing values generated)

         101 .
         102 . gen dlnLeisure=d.lnLeisure
            (13 missing values generated)

         103 . gen l1dlnLeisure=l1d.lnLeisure
            (13 missing values generated)

         104 . gen l2dlnLeisure=l2d.lnLeisure
            (13 missing values generated)
```

```

105 . gen l3dlnLeisure=l3d.lnLeisure
   (13 missing values generated)

106 . gen l12dlnLeisure=l12d.lnLeisure
   (13 missing values generated)

107 . gen l24dlnLeisure=l24d.lnLeisure
   (25 missing values generated)

108 .
109 . gen dlnManufacture=d.lnManufacture
   (13 missing values generated)

110 . gen l1dlnManufacture=l1d.lnManufacture
   (13 missing values generated)

111 . gen l2dlnManufacture=l2d.lnManufacture
   (13 missing values generated)

112 . gen l3dlnManufacture=l3d.lnManufacture
   (13 missing values generated)

113 . gen l12dlnManufacture=l12d.lnManufacture
   (13 missing values generated)

114 . gen l24dlnManufacture=l24d.lnManufacture
   (25 missing values generated)

115 .
116 . gen dlnTotal=d.lnTotal
   (13 missing values generated)

117 . gen l1dlnTotal=l1d.lnTotal
   (13 missing values generated)

118 . gen l2dlnTotal=l2d.lnTotal
   (13 missing values generated)

119 . gen l3dlnTotal=l3d.lnTotal
   (13 missing values generated)

120 . gen l12dlnTotal=l12d.lnTotal
   (13 missing values generated)

121 . gen l24dlnTotal=l24d.lnTotal
   (25 missing values generated)

122 .
123 . /*
   > gsreg dlnTotal dlnConstruct l1dlnConstruct l2dlnConstruct l3dlnConstruct ///
   >         l12dlnConstruct l24dlnConstruct ///
   >         dlnLeisure l1dlnLeisure l2dlnLeisure l3dlnLeisure l12dlnLeisure l24dl
   > nLeisure ///
   >         dlnManufacture l1dlnManufacture l2dlnManufacture l3dlnManufacture ///
   >         l12dlnManufacture l24dlnManufacture ///
   >         if tin(1990m1,2021m3), ///
   >         ncomb(1,6) aic outsample(24) fix(m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11)
   > ///
   >         samesample nindex( -1 aic -1 bic -1 rmse_out) results(gsreg_dlnTtoal)
   >  replace
   > */
124 .
125 .
126 . loocv reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11


    Leave-One-Out Cross-Validation Results
   -----------------------------------------
            Method          |    Value
   -------------------------+---------------
   Root Mean Squared Errors |   .0129605
   Mean Absolute Errors     |   .00443719
   Pseudo-R2                |   .14054754
   -----------------------------------------

127 . quietly reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

128 . estat ic

   Akaike's information criterion and Bayesian information criterion

   -----------------------------------------------------------------------------
          Model |          N   ll(null)  ll(model)      df         AIC        BIC
   -------------+---------------------------------------------------------------
              . |        371   1102.298   1206.362      15   -2382.725  -2323.982
   ------------------------------------------------------------------------------
   Note: BIC uses N = number of observations. See [R] BIC note.

129 .
130 . loocv reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11


    Leave-One-Out Cross-Validation Results
   -----------------------------------------
            Method          |    Value
   -------------------------+---------------
   Root Mean Squared Errors |   .01300797
   Mean Absolute Errors     |   .00427199
   Pseudo-R2                |   .14852182
   -----------------------------------------

131 . quietly reg d.lnTotal l(1/3,12,24)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m1
   > 1

132 . estat ic

   Akaike's information criterion and Bayesian information criterion

   ------------------------------------------------------------------------------
          Model |          N   ll(null)  ll(model)      df        AIC        BIC
   -------------+----------------------------------------------------------------
              . |        350    1035.49   1135.109      17  -2236.218  -2170.633
   ------------------------------------------------------------------------------
   Note: BIC uses N = number of observations. See [R] BIC note.

133 .
134 . loocv reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
   >         l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11


    Leave-One-Out Cross-Validation Results
   -----------------------------------------
            Method          |    Value
   -------------------------+---------------
   Root Mean Squared Errors |   .01611154
   Mean Absolute Errors     |   .00423375
   Pseudo-R2                |   .06722376
   -----------------------------------------

135 . quietly reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure /
   > //
   >         l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

136 . estat ic

   Akaike's information criterion and Bayesian information criterion

   -----------------------------------------------------------------------------
          Model |          N   ll(null)  ll(model)      df         AIC        BIC
   -------------+---------------------------------------------------------------
              . |        371   1102.298   1220.504      24   -2393.008  -2299.019
   -----------------------------------------------------------------------------
   Note: BIC uses N = number of observations. See [R] BIC note.

137 .
138 . loocv reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct ///
   >         l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8
   >  m9 m10 m11


    Leave-One-Out Cross-Validation Results
   -----------------------------------------
            Method          |    Value
   -------------------------+---------------
   Root Mean Squared Errors |   .01713897
   Mean Absolute Errors     |   .004436
   Pseudo-R2                |   .05184396
   -----------------------------------------

139 . quietly reg d.lnTotal l(1/3,12,24)d.lnTotal l(1/3,12,24)d.lnConstruct ///
   >         l(1/3,12,24)d.lnLeisure l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8
   >  m9 m10 m11

140 . estat ic

   Akaike's information criterion and Bayesian information criterion

   -------------------------------------------------------------------------------
          Model |          N   ll(null)  ll(model)      df        AIC        BIC
   -------------+-----------------------------------------------------------------
              . |        350    1035.49   1145.391      30  -2230.781  -2115.043
   -------------------------------------------------------------------------------
   Note: BIC uses N = number of observations. See [R] BIC note.

141 .
142 . *Lowest rmse (1)
143 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

         Source |       SS           df       MS      Number of obs   =       371
   -------------+----------------------------------   F(14, 356)      =     19.13
          Model |  .024491123        14  .001749366   Prob > F        =    0.0000
       Residual |  .032550038       356  .000091433   R-squared       =    0.4294
   -------------+----------------------------------   Adj R-squared   =    0.4069
          Total |  .057041161       370  .000154165   Root MSE        =    .00956

   --------------------------------------------------------------------------------
      D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
   -------------+------------------------------------------------------------------
        lnTotal |
            LD. |  -.0016173   .0529806    -0.03   0.976    -.1058116    .1025769
           L2D. |  -.1325018   .0525096    -2.52   0.012    -.2357698   -.0292338
           L3D. |    .026173   .0529824     0.49   0.622    -.0780247    .1303708
                |
             m1 |  -.0254857   .0024679   -10.33   0.000    -.0303391   -.0206322
             m2 |   .0004869   .0029045     0.17   0.867    -.0052252     .006199
             m3 |   -.003913   .0027686    -1.41   0.158    -.0093579    .0015318
             m4 |   -.013317   .0026543    -5.02   0.000    -.0185371   -.0080969
             m5 |  -.0064632   .0026182    -2.47   0.014    -.0116124   -.0013141
             m6 |  -.0088465   .0025835    -3.42   0.001    -.0139273   -.0037657
             m7 |  -.0159741   .0025331    -6.31   0.000    -.0209559   -.0109924
             m8 |  -.0051433   .0026374    -1.95   0.052    -.0103302    .0000435
             m9 |  -.0096793   .0025635    -3.78   0.000    -.0147208   -.0046378
            m10 |  -.0021472   .0025489    -0.84   0.400      -.00716    .0028657
            m11 |   .0024587   .0024684     1.00   0.320    -.0023958    .0073131
          _cons |   .0088125   .0018473     4.77   0.000     .0051795    .0124454
   ------------------------------------------------------------------------------

144 . scalar drop _all

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

146 . scalar list
   RWmaxobs180 =        180
   RWminobs180 =        180
    RWrmse180 =  .01789015
   RWmaxobs168 =        168
   RWminobs168 =        168
    RWrmse168 =  .01812779
   RWmaxobs156 =        156
   RWminobs156 =        156
    RWrmse156 =   .0186461
   RWmaxobs144 =        144
   RWminobs144 =        144
    RWrmse144 =  .01935361
   RWmaxobs132 =        132
   RWminobs132 =        132
    RWrmse132 =  .02010893
   RWmaxobs120 =        120
   RWminobs120 =        120
    RWrmse120 =  .02273114
   RWmaxobs108 =        108
   RWminobs108 =        108
    RWrmse108 =   .0233275
   RWmaxobs96 =         96
   RWminobs96 =         96
     RWrmse96 =   .0244855
   RWmaxobs84 =         84
   RWminobs84 =         84
     RWrmse84 =  .02493232
   RWmaxobs72 =         72
   RWminobs72 =         72
     RWrmse72 =  .02471091
   RWmaxobs60 =         60
   RWminobs60 =         60
     RWrmse60 =  .02489376
   RWmaxobs48 =         48
   RWminobs48 =         48
     RWrmse48 =  .02475145
   RWmaxobs36 =         36
   RWminobs36 =         36
     RWrmse36 =  .02791922
   RWmaxobs24 =         24
   RWminobs24 =         24
     RWrmse24 =  .16402211
   RWmaxobs12 =         12
   RWminobs12 =         12
     RWrmse12 =   .0132376

147 .
148 . /*
   > RWmaxobs12 =         12
   > RWminobs12 =         12
   > RWrmse12 =   .0132376
   > */
149 .
150 . *lowest AIC and BIC (3)
151 . reg d.lnTotal l(1/3)d.lnTotal l(1/3)d.lnConstruct l(1/3)d.lnLeisure ///
   >         l(1/3)d.lnManufacture m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11

         Source |       SS           df       MS      Number of obs   =      371
   -------------+----------------------------------   F(23, 347)      =    13.45
          Model |  .026880308        23  .001168709   Prob > F        =   0.0000
```

```
          Residual |  .030160853       347  .000086919   R-squared       = 0.4712
      -------------+----------------------------------   Adj R-squared   = 0.4362
             Total |  .057041161       370  .000154165   Root MSE        = .00932

      ------------------------------------------------------------------------------
          D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      --------------+---------------------------------------------------------------
            lnTotal |
                LD. |   .2862672   .2667427     1.07   0.284    -.2383687    .8109032
               L2D. |   .2454317   .2661885     0.92   0.357    -.2781141    .7689776
               L3D. |   .4266043   .2702181     1.58   0.115    -.1048673    .9580758
                    |
        lnConstruct |
                LD. |   6.41e-06    .081453     0.00   1.000    -.1601974    .1602102
               L2D. |   .0327201   .0818081     0.40   0.689     -.128182    .1936223
               L3D. |  -.0371503   .0817044    -0.45   0.650    -.1978486     .123548
                    |
          lnLeisure |
                LD. |  -.1208348   .0688988    -1.75   0.080    -.2563467    .0146771
               L2D. |  -.1169829   .0684363    -1.71   0.088     -.251585    .0176192
               L3D. |  -.1609605    .069157    -2.33   0.021    -.2969801    -.024941
                    |
      lnManufacture |
                LD. |   .0485695   .1821043     0.27   0.790    -.3095975    .4067365
               L2D. |  -.1583242   .1862613    -0.85   0.396    -.5246673     .208019
               L3D. |   .1372412   .1883006     0.73   0.467    -.2331129    .5075952
                    |
                 m1 |  -.0259546   .0026844    -9.67   0.000    -.0312344   -.0206748
                 m2 |   .0061874   .0054433     1.14   0.256    -.0045187    .0168935
                 m3 |   .0041315   .0052298     0.79   0.430    -.0061545    .0144176
                 m4 |  -.0019669   .0049557    -0.40   0.692     -.011714    .0077802
                 m5 |  -.0024728   .0032223    -0.77   0.443    -.0088106    .0038649
                 m6 |  -.0057242   .0032307    -1.77   0.077    -.0120785      .00063
                 m7 |  -.0128758   .0035363    -3.64   0.000    -.0198311   -.0059206
                 m8 |  -.0017029    .003742    -0.46   0.649    -.0090627     .005657
                 m9 |   -.006856   .0032793    -2.09   0.037    -.0133058   -.0004062
                m10 |   .0015872   .0030906     0.51   0.608    -.0044915     .007666
                m11 |   .0030825   .0026162     1.18   0.240    -.0020632    .0082282
              _cons |   .0039798   .0029489     1.35   0.178    -.0018201    .0097797
      ------------------------------------------------------------------------------

   152 . scalar drop _all

   153 . quietly forval w=3(3)180 {

   154 . scalar list
      RWmaxobs180 =        180
      RWminobs180 =        180
       RWrmse180 =  .06231312
      RWmaxobs177 =        177
      RWminobs177 =        177
       RWrmse177 =  .06310936
      RWmaxobs174 =        174
      RWminobs174 =        174
       RWrmse174 =  .06557282
      RWmaxobs171 =        171
      RWminobs171 =        171
       RWrmse171 =  .06644967
      RWmaxobs168 =        168
      RWminobs168 =        168
       RWrmse168 =   .0679586
      RWmaxobs165 =        165
      RWminobs165 =        165
       RWrmse165 =   .0691683
      RWmaxobs162 =        162
      RWminobs162 =        162
       RWrmse162 =  .06977207
      RWmaxobs159 =        159
      RWminobs159 =        159
       RWrmse159 =  .07051975
      RWmaxobs156 =        156
      RWminobs156 =        156
       RWrmse156 =   .0708996
      RWmaxobs153 =        153
      RWminobs153 =        153
       RWrmse153 =  .07104253
      RWmaxobs150 =        150
      RWminobs150 =        150
       RWrmse150 =  .07151194
      RWmaxobs147 =        147
      RWminobs147 =        147
       RWrmse147 =   .0724523
      RWmaxobs144 =        144
      RWminobs144 =        144
       RWrmse144 =  .07292486
      RWmaxobs141 =        141
      RWminobs141 =        141
       RWrmse141 =  .07359791
      RWmaxobs138 =        138
      RWminobs138 =        138
       RWrmse138 =  .07415885
      RWmaxobs135 =        135
      RWminobs135 =        135
       RWrmse135 =  .07551713
      RWmaxobs132 =        132
      RWminobs132 =        132
       RWrmse132 =  .07895271
      RWmaxobs129 =        129
      RWminobs129 =        129
       RWrmse129 =  .08680818
      RWmaxobs126 =        126
      RWminobs126 =        126
       RWrmse126 =  .08791729
      RWmaxobs123 =        123
      RWminobs123 =        123
       RWrmse123 =   .0885621
      RWmaxobs120 =        120
      RWminobs120 =        120
       RWrmse120 =   .0893681
      RWmaxobs117 =        117
      RWminobs117 =        117
       RWrmse117 =  .09222918
      RWmaxobs114 =        114
      RWminobs114 =        114
       RWrmse114 =  .09918976
      RWmaxobs111 =        111
      RWminobs111 =        111
       RWrmse111 =  .10159669
      RWmaxobs108 =        108
      RWminobs108 =        108
       RWrmse108 =   .1074872
      RWmaxobs105 =        105
      RWminobs105 =        105
       RWrmse105 =  .11024241
      RWmaxobs102 =        102
      RWminobs102 =        102
       RWrmse102 =  .11448013
      RWmaxobs99 =         99
      RWminobs99 =         99
        RWrmse99 =  .11681123
      RWmaxobs96 =         96
      RWminobs96 =         96
        RWrmse96 =  .12412703
      RWmaxobs93 =         93
      RWminobs93 =         93
        RWrmse93 =  .13010923
      RWmaxobs90 =         90
      RWminobs90 =         90
        RWrmse90 =  .12958828
      RWmaxobs87 =         87
      RWminobs87 =         87
        RWrmse87 =  .13728567
      RWmaxobs84 =         84
      RWminobs84 =         84
        RWrmse84 =  .14028654
      RWmaxobs81 =         81
      RWminobs81 =         81
        RWrmse81 =  .14073066
      RWmaxobs78 =         78
      RWminobs78 =         78
        RWrmse78 =  .14462066
      RWmaxobs75 =         75
      RWminobs75 =         75
        RWrmse75 =  .14520888
      RWmaxobs72 =         72
      RWminobs72 =         72
        RWrmse72 =  .14882576
      RWmaxobs69 =         69
      RWminobs69 =         69
        RWrmse69 =  .16426363
      RWmaxobs66 =          66
      RWminobs66 =          66
        RWrmse66 =  .16732114
      RWmaxobs63 =          63
      RWminobs63 =          63
        RWrmse63 =   .1679914
      RWmaxobs60 =          60
      RWminobs60 =          60
        RWrmse60 =  .18224837
      RWmaxobs57 =          57
      RWminobs57 =          57
        RWrmse57 =  .19923544
      RWmaxobs54 =          54
      RWminobs54 =          54
        RWrmse54 =   .2008839
      RWmaxobs51 =          51
      RWminobs51 =          51
        RWrmse51 =   .2096647
      RWmaxobs48 =          48
      RWminobs48 =          48
        RWrmse48 =  .20847129
      RWmaxobs45 =          45
      RWminobs45 =          45
        RWrmse45 =   .2244767
      RWmaxobs42 =          42
      RWminobs42 =          42
        RWrmse42 =  .25720484
      RWmaxobs39 =          39
      RWminobs39 =          39
        RWrmse39 =  .24835914
      RWmaxobs36 =          36
      RWminobs36 =          36
        RWrmse36 =  .24045465
      RWmaxobs33 =          33
      RWminobs33 =          33
        RWrmse33 =  .26919986
      RWmaxobs30 =          30
      RWminobs30 =          30
        RWrmse30 =  .49888354
      RWmaxobs27 =          27
      RWminobs27 =          27
        RWrmse27 =  .32666868
      RWmaxobs24 =          24
      RWminobs24 =          24
        RWrmse24 =  .66642366
      RWmaxobs21 =          21
      RWminobs21 =          21
        RWrmse21 =  .34614878
```

```
      RWmaxobs18 =         18
      RWminobs18 =         18
        RWrmse18 =    .757383
      RWmaxobs15 =         15
      RWminobs15 =         15
        RWrmse15 =  .21744051
      RWmaxobs12 =         12
      RWminobs12 =         12
        RWrmse12 =   .0132376
       RWmaxobs9 =          9
       RWminobs9 =          9
         RWrmse9 =  .01629782
       RWmaxobs6 =          6
       RWminobs6 =          6
         RWrmse6 =   .0162575
       RWmaxobs3 =          3
       RWminobs3 =          3
         RWrmse3 =   .0206163

   155 . /*
      > RWmaxobs12 =         12
      > RWminobs12 =         12
      > RWrmse12 =   .0132376
      > */
   156 .
   157 .
   158 . * Going with model 1 because average RWrmse is lower across window sizes
   159 . scalar rwrmse = .0132376

   160 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(,2021
      > m3)

            Source |       SS           df       MS      Number of obs   =       371
      -------------+----------------------------------   F(14, 356)      =     19.13
             Model |  .024491123        14  .001749366   Prob > F        =    0.0000
          Residual |  .032550038       356  .000091433   R-squared       =    0.4294
      -------------+----------------------------------   Adj R-squared   =    0.4069
             Total |  .057041161       370  .000154165   Root MSE        =    .00956

      ------------------------------------------------------------------------------
           D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
             lnTotal |
                 LD. |  -.0016173   .0529806    -0.03   0.976    -.1058116    .1025769
                L2D. |  -.1325018   .0525096    -2.52   0.012    -.2357698   -.0292338
                L3D. |    .026173   .0529824     0.49   0.622    -.0780247    .1303708
                     |
                  m1 |  -.0254857   .0024679   -10.33   0.000    -.0303391   -.0206322
                  m2 |   .0004869   .0029045     0.17   0.867    -.0052252     .006199
                  m3 |   -.003913   .0027686    -1.41   0.158    -.0093579    .0015318
                  m4 |   -.013317   .0026543    -5.02   0.000    -.0185371   -.0080969
                  m5 |  -.0064632   .0026182    -2.47   0.014    -.0116124   -.0013141
                  m6 |  -.0088465   .0025835    -3.42   0.001    -.0139273   -.0037657
                  m7 |  -.0159741   .0025331    -6.31   0.000    -.0209559   -.0109924
                  m8 |  -.0051433   .0026374    -1.95   0.052    -.0103302    .0000435
                  m9 |  -.0096793   .0025635    -3.78   0.000    -.0147208   -.0046378
                 m10 |  -.0021472   .0025489    -0.84   0.400      -.00716    .0028657
                 m11 |   .0024587   .0024684     1.00   0.320    -.0023958    .0073131
               _cons |   .0088125   .0018473     4.77   0.000     .0051795    .0124454
      ------------------------------------------------------------------------------

   161 . predict pd
      (option xb assumed; fitted values)
      (15 missing values generated)

   162 . gen pflcount=exp((rwrmse^2)/2)*exp(l.lnTotal+pd) if Date==tm(2021m4)
      (386 missing values generated)

   163 . gen ub1=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+1*rwrmse) if Date==tm(2021m4)
      (386 missing values generated)

   164 . gen lb1=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-1*rwrmse) if Date==tm(2021m4)
      (386 missing values generated)

   165 . gen ub2=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+2*rwrmse) if Date==tm(2021m4)
      (386 missing values generated)

   166 . gen lb2=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-2*rwrmse) if Date==tm(2021m4)
      (386 missing values generated)

   167 . gen ub3=exp((rwrmse^2)/2)*exp(l.lnTotal+pd+3*rwrmse) if Date==tm(2021m4)
      (386 missing values generated)

   168 . gen lb3=exp((rwrmse^2)/2)*exp(l.lnTotal+pd-3*rwrmse) if Date==tm(2021m4)
      (386 missing values generated)

   169 . drop pd

   170 .
   171 . replace pflcount=Total if Date==tm(2021m3)
      (1 real change made)

   172 . replace ub1=Total if Date==tm(2021m3)
      (1 real change made)

   173 . replace ub2=Total if Date==tm(2021m3)
      (1 real change made)

   174 . replace ub3=Total if Date==tm(2021m3)
      (1 real change made)

   175 . replace lb1=Total if Date==tm(2021m3)
      (1 real change made)

   176 . replace lb2=Total if Date==tm(2021m3)
      (1 real change made)

   177 . replace lb3=Total if Date==tm(2021m3)
      (1 real change made)

   178 .
   179 . twoway (tsrline ub3 ub2 if tin(2020m4,2021m4), ///
      >         recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
      >         (tsrline ub2 ub1 if tin(2020m4,2021m4), ///
      >         recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
      >         (tsrline ub1 pflcount if tin(2020m4,2021m4), ///
      >         recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
      >         (tsrline pflcount lb1 if tin(2020m4,2021m4), ///
      >         recast(rarea) fcolor(purple) fintensity(65) lwidth(none) ) ///
      >         (tsrline lb1 lb2 if tin(2020m4,2021m4), ///
      >         recast(rarea) fcolor(green) fintensity(40) lwidth(none) ) ///
      >         (tsrline lb2 lb3 if tin(2020m4,2021m4), ///
      >         recast(rarea) fcolor(orange) fintensity(20) lwidth(none) ) ///
      >         (tsline Total pflcount if tin(2020m4,2021m4) , ///
      >         lcolor(gs12 pink) lwidth(medthick medthick) ///
      >         lpattern(solid longdash)), scheme(s1mono) legend(off)

   180 . graph export "TotalFan1.png", replace
      (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time
Series/Problem Sets
      > /Final Exam/TotalFan1.png written in PNG format)

   181 .
   182 . * More than 1 step
   183 . arima d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if
tin(199
      > 0m1,2021m3)

      (setting optimization to BHHH)
      Iteration 0:   log likelihood =  1206.3625
      Iteration 1:   log likelihood =  1206.3625

      ARIMA regression

      Sample:  1990m5 - 2021m3                        Number of obs     =
371
                                                      Wald chi2(14)     =
83.94
      Log likelihood =  1206.362                      Prob > chi2       =
0.0000

      ------------------------------------------------------------------------
-----
                   |                 OPG
         D.lnTotal |      Coef.   Std. Err.      z    P>|z|     [95% Conf.
Interval]
      -------------+----------------------------------------------------------
-----
```

```
      lnTotal      |
           lnTotal |
               LD. |  -.0016173   .0448009    -0.04   0.971    -.0894254    .0861907
              L2D. |  -.1325018   .0486537    -2.72   0.006    -.2278614   -.0371422
              L3D. |    .026173   .0400229     0.65   0.513    -.0522705    .1046166
                   |
                m1 |  -.0254857   .0054401    -4.68   0.000    -.0361481   -.0148232
                m2 |   .0004869   .0068588     0.07   0.943    -.0129561    .0139299
                m3 |   -.003913   .0058695    -0.67   0.505    -.0154171     .007591
                m4 |   -.013317   .0047531    -2.80   0.005    -.0226329   -.0040012
                m5 |  -.0064632   .0082351    -0.78   0.433    -.0226038    .0096773
                m6 |  -.0088465   .0069016    -1.28   0.200    -.0223735    .0046804
                m7 |  -.0159741   .0056734    -2.82   0.005    -.0270938   -.0048545
                m8 |  -.0051433   .0074271    -0.69   0.489    -.0197002    .0094135
                m9 |  -.0096793   .0051505    -1.88   0.060    -.0197741    .0004155
               m10 |  -.0021472   .0048943    -0.44   0.661    -.0117399    .0074456
               m11 |   .0024587   .0066745     0.37   0.713    -.0106231    .0155405
             _cons |   .0088125   .0042033     2.10   0.036     .0005741    .0170508
      -------------+-----------------------------------------------------------------
            /sigma |   .0093667   .0001378    67.96   0.000     .0090966    .0096369
      -------------------------------------------------------------------------------
      Note: The test of the variance against zero is one sided, and the two-sided
            confidence interval is truncated at zero.

   184 . predict pnonfarm, dynamic(tm(2021m3))
      (option xb assumed; predicted values)
      (4 missing values generated)

   185 . predict mse, mse dynamic(mofd(tm(2021m4)))
```

```
186 . gen totmse = mse if Date==tm(2021m4)
   (386 missing values generated)

187 . replace totmse = l.totmse+mse if Date>tm(2021m4)
   (11 real changes made)

188 . gen pnonfarma = Total if Date==tm(2021m3)
   (386 missing values generated)

189 . replace pnonfarma = l.pnonfarma*exp(pnonfarm+mse/2) if Date>tm(2021m3)
   (12 real changes made)

190 .
191 . gen ub1a = pnonfarma*exp(totmse^.5)
   (375 missing values generated)

192 . gen ub2a = pnonfarma*exp(2*totmse^.5)
   (375 missing values generated)

193 . gen ub3a = pnonfarma*exp(3*totmse^.5)
   (375 missing values generated)

194 . gen lb1a = pnonfarma/exp(totmse^.5)
   (375 missing values generated)

195 . gen lb2a = pnonfarma/exp(2*totmse^.5)
   (375 missing values generated)

196 . gen lb3a = pnonfarma/exp(3*totmse^.5)
   (375 missing values generated)

197 .
198 . replace ub1a=Total if Date == tm(2021m3)
   (1 real change made)

199 . replace ub2a=Total if Date == tm(2021m3)
   (1 real change made)

200 . replace ub3a=Total if Date == tm(2021m3)
   (1 real change made)

201 . replace lb1a=Total if Date == tm(2021m3)
   (1 real change made)

202 . replace lb2a=Total if Date == tm(2021m3)
   (1 real change made)
```

```
  203 . replace lb3a=Total if Date == tm(2021m3)
     (1 real change made)

  204 .
  205 . twoway (tsrline ub3a ub2a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
     >         (tsrline ub2a ub1a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
     >         (tsrline ub1a pnonfarma if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
     >         (tsrline pnonfarma lb1a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(blue) fintensity(65) lwidth(none) ) ///
     >         (tsrline lb1a lb2a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(yellow) fintensity(40) lwidth(none) ) ///
     >         (tsrline lb2a lb3a if tin(2019m1,2022m3), ///
     >         recast(rarea) fcolor(red) fintensity(20) lwidth(none) ) ///
     >         (tsline Total pnonfarma if tin(2019m1,2022m3) , ///
     >         lcolor(gs12 pink) lwidth(medthick medthick) ///
     >         lpattern(solid longdash)) , scheme(s1mono) legend(off)

  206 . graph export "TotalFan12.png", replace
     (file /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Problem Sets
     > /Final Exam/TotalFan12.png written in PNG format)

  207 .
  208 . scalar rmse_mod1 = .0132376

  209 . reg d.lnTotal l(1/3)d.lnTotal m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 if tin(1990m
     > 1,2021m3)

           Source |       SS           df       MS      Number of obs   =       371
     -------------+----------------------------------   F(14, 356)      =     19.13
            Model |  .024491123        14  .001749366   Prob > F        =    0.0000
         Residual |  .032550038       356  .000091433   R-squared       =    0.4294
     -------------+----------------------------------   Adj R-squared   =    0.4069
            Total |  .057041161       370  .000154165   Root MSE        =    .00956

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

```
         D.lnTotal |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
      -------------+----------------------------------------------------------------
           lnTotal |
               LD. |  -.0016173   .0529806    -0.03   0.976    -.1058116    .1025769
              L2D. |  -.1325018   .0525096    -2.52   0.012    -.2357698   -.0292338
              L3D. |    .026173   .0529824     0.49   0.622    -.0780247    .1303708
                   |
                m1 |  -.0254857   .0024679   -10.33   0.000    -.0303391   -.0206322
                m2 |   .0004869   .0029045     0.17   0.867    -.0052252     .006199
                m3 |   -.003913   .0027686    -1.41   0.158    -.0093579    .0015318
                m4 |   -.013317   .0026543    -5.02   0.000    -.0185371   -.0080969
                m5 |  -.0064632   .0026182    -2.47   0.014    -.0116124   -.0013141
                m6 |  -.0088465   .0025835    -3.42   0.001    -.0139273   -.0037657
                m7 |  -.0159741   .0025331    -6.31   0.000    -.0209559   -.0109924
                m8 |  -.0051433   .0026374    -1.95   0.052    -.0103302    .0000435
                m9 |  -.0096793   .0025635    -3.78   0.000    -.0147208   -.0046378
               m10 |  -.0021472   .0025489    -0.84   0.400      -.00716    .0028657
               m11 |   .0024587   .0024684     1.00   0.320    -.0023958    .0073131
             _cons |   .0088125   .0018473     4.77   0.000     .0051795    .0124454
      ------------------------------------------------------------------------------

   210 . predict plTotal
      (option xb assumed; fitted values)
      (15 missing values generated)

   211 . predict temp if tin(2021m3,2021m3)
      (option xb assumed; fitted values)
      (386 missing values generated)

   212 . replace plTotal=temp if tin(2021m3,2021m3)
```

```
      (0 real changes made)

   213 . drop temp

   214 . gen pTotal=exp(l.lnTotal+plTotal+(rmse_mod1^2)/2)
      (15 missing values generated)

   215 . gen lbTotal=exp(l.lnTotal+plTotal-1.96*rmse_mod1+(rmse_mod1^2)/2)
      (15 missing values generated)

   216 . gen ubTotal=exp(l.lnTotal+plTotal+1.96*rmse_mod1+(rmse_mod1^2)/2)
      (15 missing values generated)

   217 .
   218 . gen res=(d.lnTotal-plTotal)
      (16 missing values generated)

   219 . gen expres=exp(res)
      (16 missing values generated)

   220 . summ expres

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
            expres |        371    1.000042    .0088821   .8565838   1.028623

   221 . scalar meanexpres=r(mean)

   222 . gen epTotal=exp(l.lnTotal+plTotal)*meanexpres
      (15 missing values generated)

   223 . _pctile res, percentile(2.5,97.5)

   224 . return list

      scalars:
                       r(r1) =  -.0089262239634991
                       r(r2) =  .0108835604041815

   225 . gen elbTotal=exp(l.lnTotal+plTotal+r(r1))*meanexpres
      (15 missing values generated)

   226 . gen eubTotal=exp(l.lnTotal+plTotal+r(r2))*meanexpres
      (15 missing values generated)

   227 .
   228 . tsline Total pTotal elbTotal eubTotal lbTotal ubTotal if tin(2019m1,2021m4),
      > ///
      >         scheme(s1mono) tline(2021m3, lcolor(gs4)) ///
      >         lpattern(solid solid longdash longdash shortdash shortdash) ///
      >         lcolor(dkorange gs5 gs10 gs10 dkorange%60 dkorange%60) ///
      >         lwidth(medthick medthick medium medium)

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

   230 .
   231 . histogram expres, normal kdensity saving(residuals.gph, replace)
      (bin=19, start=.85658383, width=.00905468)
      (file residuals.gph saved)

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

   233 .
   234 . log close
            name:  <unnamed>
             log:  /Users/guslipkin/Documents/Spring2020/CAP 4763 ~ Time Series/Probl
      > em Sets/Final Exam/Final Exam.smcl
        log type:  smcl
       closed on:  29 Apr 2021, 12:09:27
      -------------------------------------------------------------------------------
```