Recitation 8: Time Series and Big Data

Matthew Davis

July 18, 2023

Practice Problem 1: The St. Louis Model

A model that attracted quite a bit of interest in macroeconomics in the 1970s was the St. Louis model. The underlying idea was to calculate fiscal and monetary policy impact and long-run *cumulative dynamic multipliers*, by relating real output (growth) to real government expenditure (growth) and real money supply (growth). The assumption was that both government expenditures and the money supply were exogenous.

In order to investigate the effect of a fiscal and monetary policies on output, you want to estimate a St. Louis type model using your quarterly data (i.e., make sure to use HAC standard errors) and report your results.

(a) and (b) Download FRED Data

Visit the Federal Reserve Bank of St. Louis where you have access to the FRED and download the data for the required three variables (i.e., real GDP (GDPC1), real money supply (M2REAL), and real government expenditure (GCEC1)). Don't need to learn this, but a convenient way of doing this is using the Quandl package to download the data directly

```
Quandl.api_key('xHu2y3xExQ6bGkGqcYEi') # This is my personal API key
gdp <- Quandl('FRED/GDPC1')
m2 <- Quandl('FRED/M2REAL')
gov <- Quandl('FRED/GCEC1')

head(gdp)
```

```
##         Date    Value
## 1 2021-10-01 19805.96
## 2 2021-07-01 19478.89
## 3 2021-04-01 19368.31
## 4 2021-01-01 19055.65
## 5 2020-10-01 18767.78
## 6 2020-07-01 18560.77
```

```
head(m2)
```

```
##         Date  Value
## 1 2021-12-01 7724.4
## 2 2021-11-01 7696.6
## 3 2021-10-01 7660.0
## 4 2021-09-01 7657.1
## 5 2021-08-01 7629.0
## 6 2021-07-01 7560.6
```

```
head(gov)
```

```
##         Date    Value
## 1 2021-10-01 3356.829
## 2 2021-07-01 3381.574
## 3 2021-04-01 3373.765
## 4 2021-01-01 3390.921
## 5 2020-10-01 3356.030
## 6 2020-07-01 3360.238
```

The real money supply *m2* is available on a monthly frequency basis: don't forget to convert it into a quarterly frequency variable to match it with the other two variables. Hint: Is *m2* a stock or flow variable? Although, you may take the last month of each quarter or the three-month average as your quarterly value, here use the first month of each quarter. Since M2 is a stock variable, you can take the value of this series corresponding to the first month of each quarter to get the quarterly values of this variable.

```
fred <- left_join(gdp, m2, by = 'Date') %>%
  left_join(gov, by = 'Date')
head(fred)
```

```
##         Date  Value.x Value.y    Value
## 1 2021-10-01 19805.96  7660.0 3356.829
## 2 2021-07-01 19478.89  7560.6 3381.574
## 3 2021-04-01 19368.31  7550.3 3373.765
## 4 2021-01-01 19055.65  7396.4 3390.921
## 5 2020-10-01 18767.78  7201.0 3356.030
## 6 2020-07-01 18560.77  7084.5 3360.238
```

```
fred %<>% rename(gdp = Value.x,
                 m2 = Value.y,
                 gov = Value)
head(fred)
```

```
##         Date      gdp     m2      gov
## 1 2021-10-01 19805.96 7660.0 3356.829
## 2 2021-07-01 19478.89 7560.6 3381.574
## 3 2021-04-01 19368.31 7550.3 3373.765
## 4 2021-01-01 19055.65 7396.4 3390.921
## 5 2020-10-01 18767.78 7201.0 3356.030
## 6 2020-07-01 18560.77 7084.5 3360.238
```

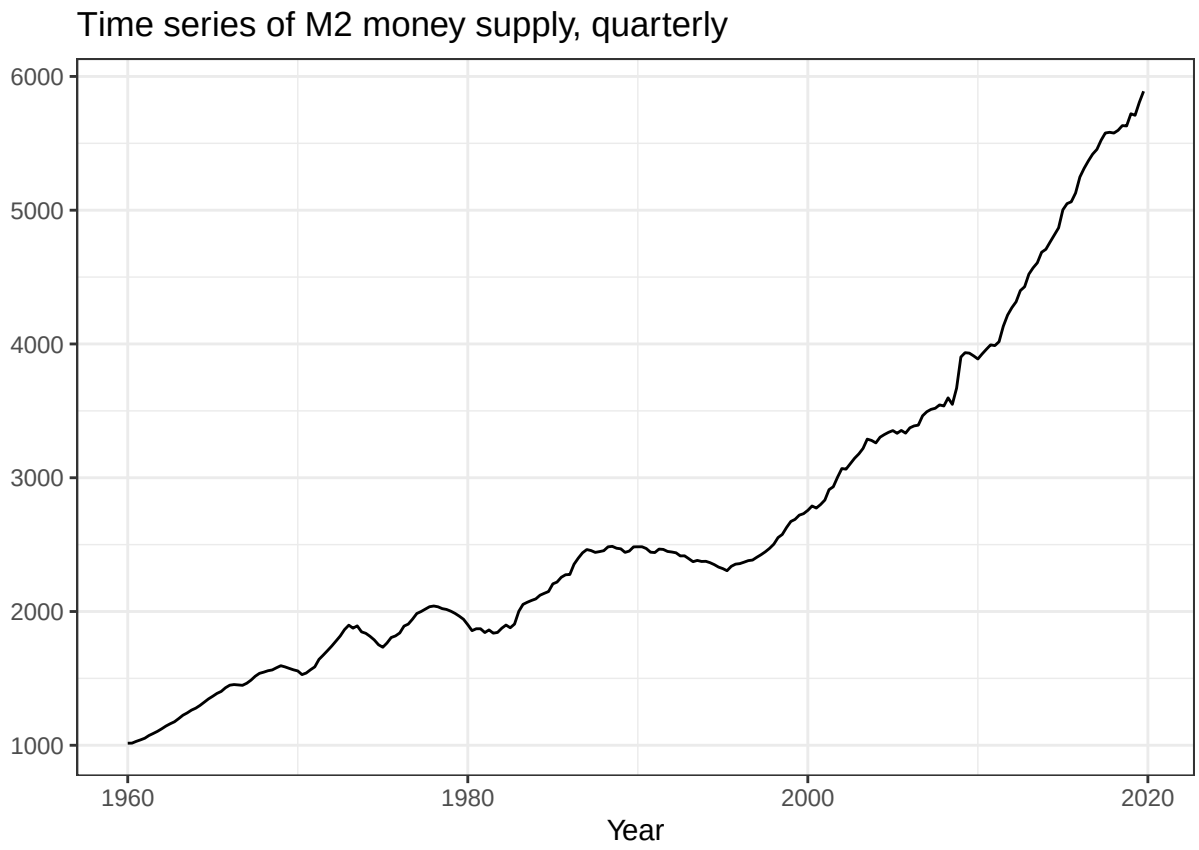
Notice that our time variable, Date, is not just a number variable but a 'Date' variable.

The sample period should be from first quarter of 1960 to the fourth quarter of 2019 (i.e., 1960q1 to 2019q4). Quarters in this dataset are defined by the first day of the first month in that quarter, i.e. the 1st of January, April, July, and October

```
fred %<>% filter(Date >= '1960-01-01') %>%
  filter(Date <= '2019-12-31')
```

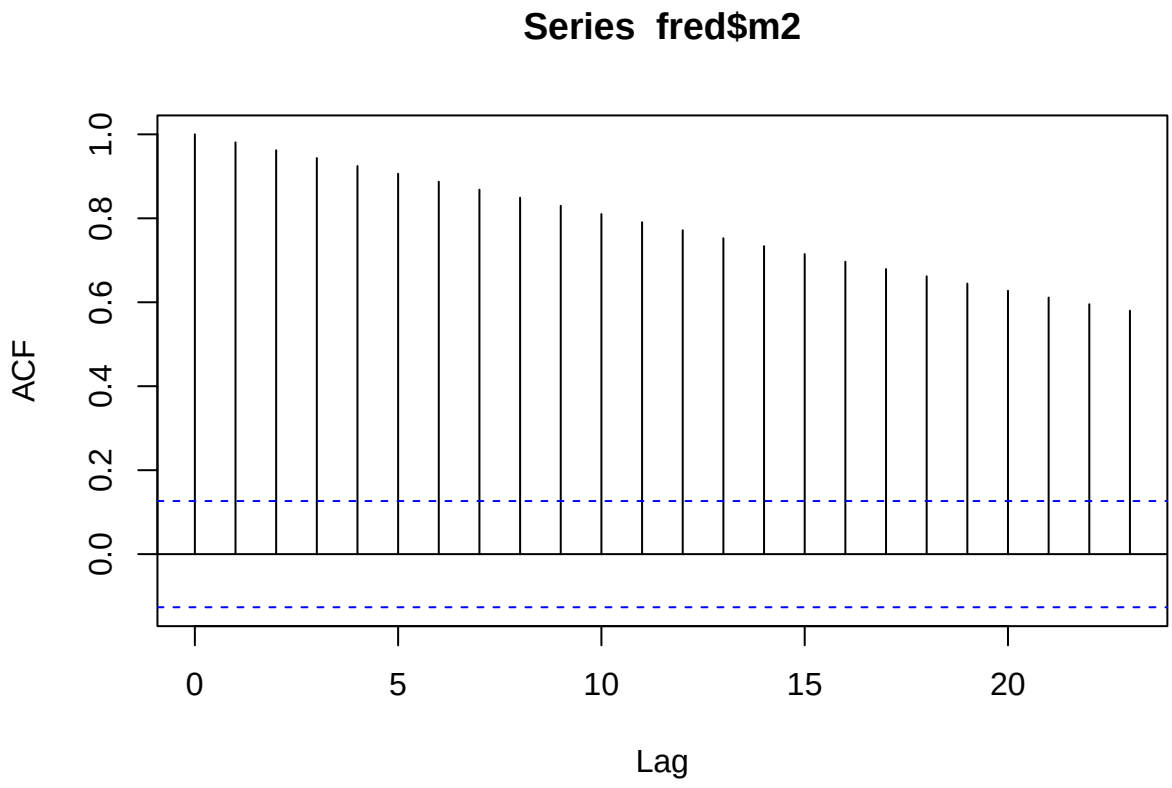
As a sanity check, let's plot one of these variables as a line graph:

```
ggplot(fred, aes(x = Date, y = m2)) +
  theme_bw() +
  geom_line() +
  xlab('Year') +
  ylab('') +
  ggtitle('Time series of M2 money supply, quarterly')
```



We can also derive the autocorrelations for this series either as a plot of a corellogram...

```
acf(fred$m2,
    na.action = na.pass) # ignore any missing data/gaps in the time series
```



Or by displaying the estimated autocorrelations, say up to 15 lags:

```
acf(fred$m2,
    na.action = na.exclude, # Ignore any missing data
    lag.max = 15, # Number of lags to include
    plot = FALSE)
```

```
##
## Autocorrelations of series 'fred$m2', by lag
##
##     0     1     2     3     4     5     6     7     8     9    10    11    12
## 1.000 0.981 0.962 0.944 0.925 0.906 0.887 0.868 0.849 0.830 0.810 0.791 0.771
##    13    14    15
## 0.752 0.734 0.715
```

Compute the growth rates of these three variables after you first transform them into natural logarithm and name/label them *ygrowth*, *mgrowth*, and *ggrowth* For a variable X_t, we'll compute the growth rate x_t using the following transformation:

$$x_t = \log(X_t) - \log(X_{t-1})$$

```
fred %<>% mutate(ygrowth = log(gdp)-lag(log(gdp)),
                 mgrowth = log(m2)-lag(log(m2)),
                 ggrowth = log(gov)-lag(log(gov)))
```

(c) Estimate a distributed lag model of *ygrowth* on current-period *mgrowth*, the effect of a monetary policy on current quarter's output growth

(d) Estimate a distributed lag model of *ygrowth* on current-period *ggrowth*, the effect of a fiscal policy on current quarter's output growth

We'll use our familiar feols package to estimate a series of distributed lag models. We'll find it convenient to use the same panel methods for time series, allowing us to use mostly the same commands we've used before. This will require us to define our unit and and time variables (since fixest is intended for panel datasets). Since time series are essentially just panel datasets with T time periods but only $N = 1$ units, let's just create a unit variable that takes on an arbitrary value for all observations:

```
fred %<>% mutate(unit = 'same')
```

This will also be necessary to make use of the other new component: new standard errors for time series and panel datasets which are robust to both heteroskedasticity and autocorrelation (HAC). We call these Newey-West (NW) standard errors and they are conveniently calculated using the $NW(m)$ function.

These standard errors require us to determine the appropriate lag truncation parameter m. This is a matter of applying the rule-of-thumb formula from the textbook and rounding up:

```
m <- 0.75*nrow(fred)^(1/3)
m
```

```
## [1] 4.660849
```

```
# Round up
m <- ceiling(m)
m
```

```
## [1] 5
```

Last note: we can of course use the same functions to estimate autoregressive models AR(p). In this case, we do not want to use HAC standard errors and in addition, we'll want to specify iid standard errors so we don't cluster standard errors

Now we can estimate the desired models:

```
mod.c <- feols(ygrowth ~ mgrowth, fred,
               panel.id = c('unit', 'Date'),
               vcov = NW(5))
```

```
## NOTE: 1 observation removed because of NA values (LHS: 1, RHS: 1).
```

```
mod.d <- feols(ygrowth ~ ggrowth, fred,
               panel.id = c('unit', 'Date'),
               vcov = NW(5))
```

```
## NOTE: 1 observation removed because of NA values (LHS: 1, RHS: 1).
```

```
etable(mod.c, mod.d)
```

```
##                                mod.c               mod.d
## Dependent Var.:              ygrowth             ygrowth
##
## Constant        -0.0066*** (0.0008) -0.0064*** (0.0007)
## mgrowth             0.1135 (0.0781)
## ggrowth                                0.2148*** (0.0520)
## ---------------- -------------------- --------------------
## S.E. type          Newey-West (L=5)    Newey-West (L=5)
## Observations                   239                 239
## R2                         0.02722             0.06508
## Adj. R2                    0.02312             0.06114
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(e) Estimate a distributed lag model of *ygrowth* on current and next quarter's monetary policy on output growth

(f) Estimate a distributed lag model of *ygrowth* on current and next quarter's fiscal policy

To clarify the ambiguous language of the question, we are being asked to regress GDP growth between period t and t+1 to changes to monetary/fiscal policy between periods t and t+1 and changes to monetary/fiscal policy between periods t-1 and t.

Other than the standard errors, the other new component here is that our models are "dynamic", meaning they relate observations from different time periods to one another. Here, we'll find it very convenient to make use of the fixest package, which has functions l(), d(), and f(), which allow us to refer to lagged, differenced, and lead terms in a regression formula without having to manually create new variables.

Estimating the above models:

```
mod.e <- feols(ygrowth ~ l(mgrowth, 0:1), fred,
               panel.id = c('unit', 'Date'),
               vcov = NW(5))
```

```
## NOTE: 2 observations removed because of NA values (LHS: 1, RHS: 2).
```

```
mod.f <- feols(ygrowth ~ l(ggrowth, 0:1),
               fred,panel.id = c('unit', 'Date'),
               vcov = NW(5))
```

```
## NOTE: 2 observations removed because of NA values (LHS: 1, RHS: 2).
```

```
# Or combining them into one command
mods.ef <- feols(ygrowth ~ sw(l(mgrowth, 0:1),
                              l(ggrowth, 0:1)),
                 fred,panel.id = c('unit', 'Date'),
                 vcov = NW(5))
```

```
## NOTE: 1 observation removed because of NA values (LHS: 1).
##       |-> this msg only concerns the variables common to all estimations
```

```
etable(mods.ef)
```

```
##                           mods.ef.1           mods.ef.2
## Dependent Var.:             ygrowth             ygrowth
##
## Constant        -0.0058*** (0.0008) -0.0067*** (0.0006)
## mgrowth             0.0165 (0.0629)
## l(mgrowth,1)      0.2069** (0.0650)
## ggrowth                              0.2313*** (0.0512)
## l(ggrowth,1)                           -0.0622 (0.0454)
## _______________ ___________________ ___________________
## S.E. type          Newey-West (L=5)    Newey-West (L=5)
## Observations                   238                 238
## R2                         0.09842             0.07352
## Adj. R2                    0.09075             0.06563
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice the second argument in the lag function l() can take on a vector of numbers if we want to include multiple lags (including lag 0 here).

What is the impact multiplier? Explain the meaning.

What is the cumulative multiplier? Explain the meaning.

(g) Estimate a distributed lag model of *ygrowth* on the change (i.e., the *first difference*) in current and four lags of *mgrowth* and *ggrowth* (to mimic the original St. Louis Equation)

```
# Create first-differenced versions of our regressors
fred %<>% mutate(diff.mgrowth = mgrowth-lag(mgrowth),
                 diff.ggrowth = ggrowth-lag(ggrowth))

mod.g <- feols(ygrowth ~ l(diff.mgrowth, 0:3) + l(mgrowth, 4) + l(diff.ggrowth, 0:3) + l(ggrowth, 4),
               fred, panel.id = c('unit', 'Date'),
               vcov = NW(5))
```

```
## NOTE: 6 observations removed because of NA values (LHS: 1, RHS: 6).
```

```
etable(mod.g)
```

```
##                                mod.g
## Dependent Var.:              ygrowth
##
## Constant          -0.0066*** (0.0008)
```

```
## diff.mgrowth       0.1687*** (0.0485)
## l(diff.mgrowth,1)    0.1505* (0.0591)
## l(diff.mgrowth,2)     0.0415 (0.0744)
## l(diff.mgrowth,3)    -0.0163 (0.0627)
## l(mgrowth,4)         0.1120* (0.0461)
## diff.ggrowth         -0.0158 (0.0524)
## l(diff.ggrowth,1) -0.2164*** (0.0640)
## l(diff.ggrowth,2)   -0.1462* (0.0657)
## l(diff.ggrowth,3)  -0.1472** (0.0488)
## l(ggrowth,4)          0.0346 (0.0498)
## ----------------- -------------------
## S.E. type            Newey-West (L=5)
## Observations                  234
## R2                        0.18118
## Adj. R2                   0.14446
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(h) Assuming that money and government expenditures are exogenous, what do the coefficients represent? Calculate the h-period cumulative dynamic multipliers from these.

```
stl.est <- coeftable(mod.g)
stl.est
```

```
##                        Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept)        -0.006592834 0.0007511044 -8.7775207 3.636806e-16
## diff.mgrowth        0.168727833 0.0484537481  3.4822452 5.936453e-04
## l(diff.mgrowth, 1)  0.150496282 0.0591134932  2.5458871 1.154554e-02
## l(diff.mgrowth, 2)  0.041516018 0.0744361091  0.5577403 5.775572e-01
## l(diff.mgrowth, 3) -0.016273219 0.0626708169 -0.2596618 7.953539e-01
## l(mgrowth, 4)       0.112017586 0.0461457772  2.4274721 1.596348e-02
## diff.ggrowth       -0.015797496 0.0524098815 -0.3014221 7.633615e-01
## l(diff.ggrowth, 1) -0.216419889 0.0639485787 -3.3842799 8.372077e-04
## l(diff.ggrowth, 2) -0.146205740 0.0657339718 -2.2242037 2.709386e-02
## l(diff.ggrowth, 3) -0.147186933 0.0488436594 -3.0134297 2.868565e-03
## l(ggrowth, 4)       0.034571318 0.0497538378  0.6948473 4.878433e-01
## attr(,"type")
## [1] "Newey-West (L=5)"
```

Assuming money and government expenditures are exogenous, the coefficients estimated represent the (monetary/fiscal) cumulative multipliers. If you differenced them out, they would represent the dynamic multipliers. The coefficients are easy to plot when estimated through fixest:

```
# Collect the cumulative multipliers
cumu <- data.frame(Estimate = stl.est[-1,'Estimate']) %>%
  mutate(Lag = rep(0:4, 2),
         Policy = rep(c('Monetary', 'Fiscal'), each = 5))
dyna <- group_by(cumu, Policy) %>%
  mutate(Estimate = ifelse(Lag == 0, Estimate, Estimate-lag(Estimate)))
```

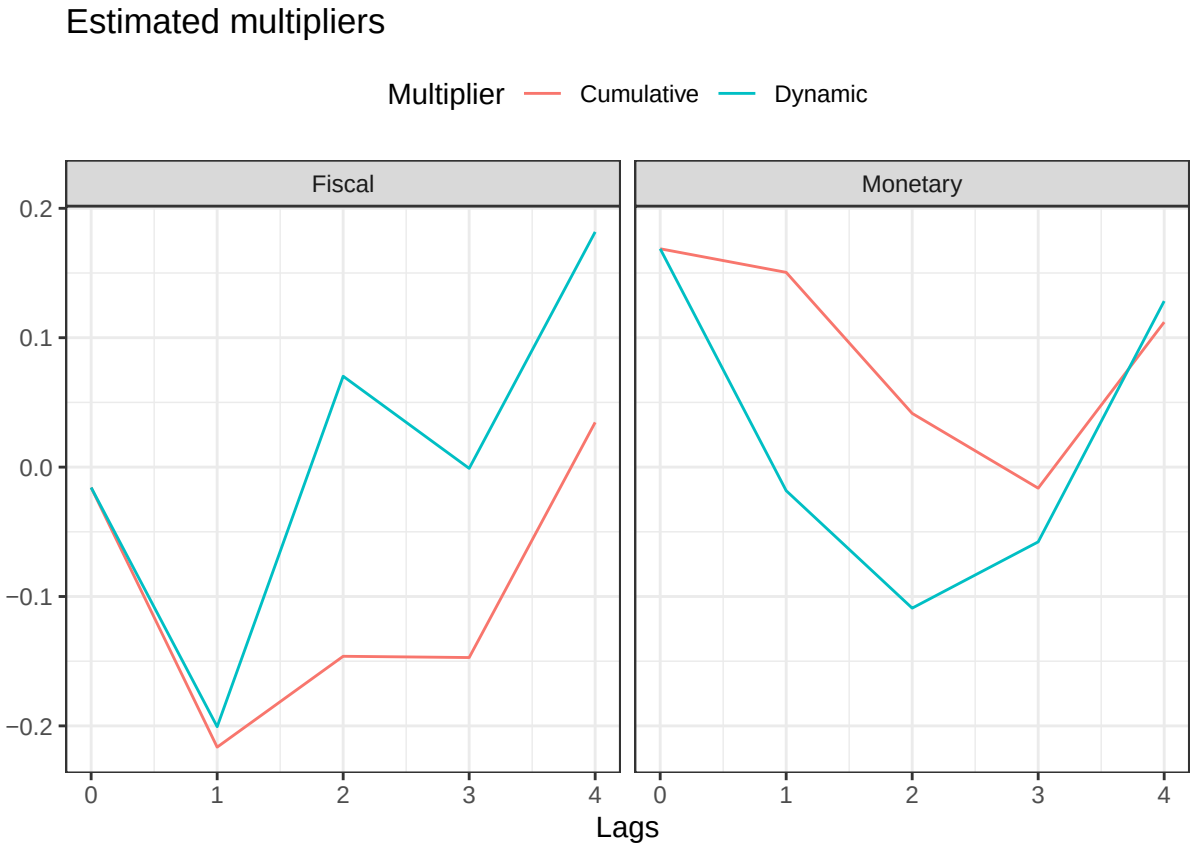
```r
multipliers <- rbind(cumu, dyna) %>%
  mutate(Multiplier = rep(c('Cumulative', 'Dynamic'), each = 10))

multi.table <- pivot_wider(multipliers,
                           names_from = c(Policy, Multiplier),
                           values_from = Estimate)
print(multi.table)
```

```
## # A tibble: 5 x 5
##     Lag Monetary_Cumulative Fiscal_Cumulative Monetary_Dynamic Fiscal_Dynamic
##   <int>               <dbl>             <dbl>            <dbl>          <dbl>
## 1     0               0.169           -0.0158            0.169       -0.0158
## 2     1               0.150            -0.216           -0.0182       -0.201
## 3     2              0.0415            -0.146            -0.109        0.0702
## 4     3             -0.0163            -0.147            -0.0578      -0.000981
## 5     4               0.112            0.0346             0.128        0.182
```

How can you test for the statistical significance of the cumulative dynamic multipliers and the long-run cumulative dynamic multiplier?

(i) Sketch the estimated dynamic and cumulative dynamic fiscal and monetary multipliers (similar to Fig 16.2 in the textbook)

```r
ggplot(multipliers, aes(x = Lag, y = Estimate, color = Multiplier)) +
  theme_bw() +
  geom_line() +
  theme(legend.position = 'top') +
  ggtitle('Estimated multipliers') +
  xlab('Lags') + ylab('') + facet_grid(~ Policy)
```



(j) For these coefficients to represent dynamic ultipliers, the money supply and government expenditures must be exogenous variables. Explain why this is unlikely to be the case. As a result, what importance should you attach to the above results?

There is little reason to believe that these government instruments are exogenous. Even if the monetary base and those components of government expenditures which do not respond to business cycle fluctuations had been chosen rather than the above regressors, these instruments respond to changes in the growth rate of GDP.

In fact, government reaction functions were also estimated at the time to capture how government instruments respond to changes in target variables. As a result, the regressors will be correlated with the error term, OLS estimation is inconsistent, and inference not dependable. It is hard to imagine how useable information can be retrieved from these numbers.

Practice Problem 2: Stock-Watson Empirical Exercise 16.1

Note that the textbook leads you to the data hosted on the official textbook website. This data does not match what's in the textbook or the solutions! It doesn't even have the variables you need to answer any of these subquestions. I had to find the original dataset on an unofficial website so download it from my recitation folder instead:

```
macro <- read_excel('data/UsMacro_Monthly.xlsx')
head(macro)
```

```
## # A tibble: 6 x 6
##    Year Month    IP   Oil   CPI  PCED
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1947     1  13.6    NA  21.5    NA
## 2  1947     2  13.7    NA  21.6    NA
## 3  1947     3  13.7    NA  22      NA
## 4  1947     4  13.6    NA  22      NA
## 5  1947     5  13.7    NA  22.0    NA
## 6  1947     6  13.7    NA  22.1    NA
```

Part a: Compute the monthly growth rate in IP, expressed in percentage points. What are the mean and standard deviation of IP growth over the 1960:M1–2017:M12 sample period? What are the units for IP growth (percent, percent per annum, percent per month, or something else)?

```
macro %<>% mutate(ip.growth = 100*(log(IP)-log(lag(IP))))

macro.sample <- filter(macro, Year >= 1960 & Year <= 2017)
mean(macro.sample$ip.growth)
```

```
## [1] 0.2235496
```

```
sd(macro.sample$ip.growth)
```

```
## [1] 0.7781159
```

So we have a mean of about 0.22 and a standard deviation of about 0.75. This is slightly different from the provided solutions, which seem to be using a sample from the period 1952-2009 for some reason (different textbook?) rather than 1960-2017.

Part b: Plot the value of O_t. Why are so many values of O_t equal to 0? Why aren't some values of O_t negative?

We want to plot the value of oil. Let's create a Date variable that combines the info in Year and Month first:

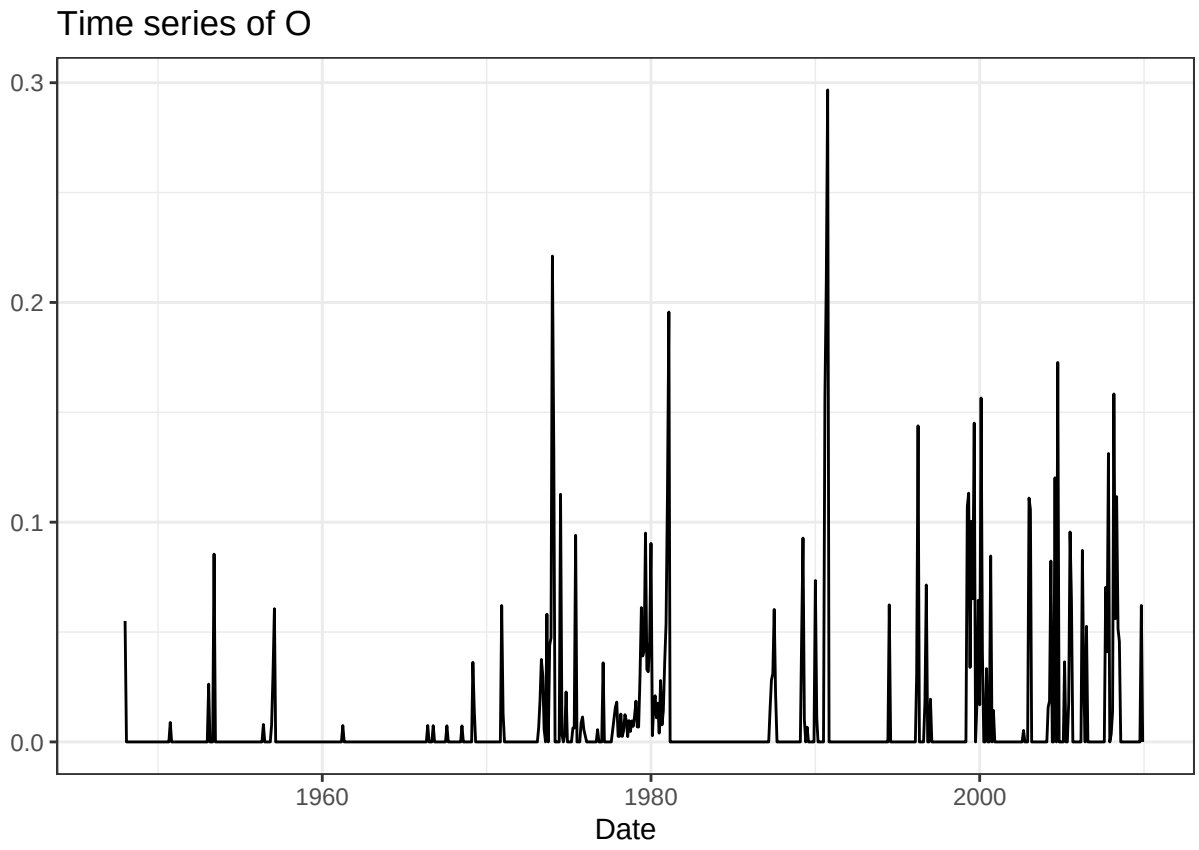
```
macro %<>% mutate(Date = as.Date(ISOdate(Year, Month, 1)))
```

Here, the 1 just tells R to assign it the first day of the month.

Now let's plot the oil time series:

```
ggplot(macro, aes(x = Date, y = Oil)) +
  theme_bw() + # A black-and-white theme I like over the gray default
  geom_line() + # Draw a line graph using Date on the x and Oil on the y
  xlab('Date') + ggtitle('Time series of O') + ylab('')
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```



From Exercise 16.1, O here is defined as "the greater of 0 or the percentage point difference between oil prices at date t and their maximum value during the past three years." Thus it equals 0 whenever the price of oil is less than the maximum during the previous three years.

Part c: Estimate a distributed lag model regressing IP growth against the current value and 18 lagged values of O, including an intercept. What value of the HAC standard error truncation parameter did you choose? Why?

We are estimating a distributed lag model of IP growth onto the current and 18 lagged values of the variable O_t.

First, let's calculate the HAC truncation parameter m:

```
m <-  0.75*nrow(macro)^(1/3)
m <- ceiling(m)
m
```

```
## [1] 7
```

Estimating the requested distributed lag model:

```r
# Create an id variable like before
macro %<>% mutate(id = 999)
oil.dyn <- feols(ip.growth ~ l(Oil, 0:18), macro,
                 panel.id = ~ id + Date,
                 vcov = NW(m))
```

```
## NOTE: 30 observations removed because of NA values (LHS: 1, RHS: 30).
```

Part d: Taken as a group, are the coefficients on O_t statistically significantly different from 0?

Printing the regression output, including an F-test on all these regressors:

```r
etable(oil.dyn, fitstat = ~ wald + wald.p)
```

```
##                                      oil.dyn
## Dependent Var.:                    ip.growth
##
## Constant                    0.4275*** (0.0675)
## Oil                            0.1553 (0.8532)
## l(Oil,1)                      -0.9760 (0.9559)
## l(Oil,2)                      -1.403. (0.7918)
## l(Oil,3)                      -0.8315 (0.9295)
## l(Oil,4)                      -0.4064 (0.8824)
## l(Oil,5)                      -0.4202 (0.7929)
## l(Oil,6)                       -2.555. (1.472)
## l(Oil,7)                      -0.2286 (0.9797)
## l(Oil,8)                       0.8800 (1.008)
## l(Oil,9)                       -1.639 (1.028)
## l(Oil,10)                     -3.923* (1.872)
## l(Oil,11)                      -2.607 (1.954)
## l(Oil,12)                     -0.2247 (1.292)
## l(Oil,13)                      -1.555 (1.131)
## l(Oil,14)                      -1.483 (0.9120)
## l(Oil,15)                     -1.479. (0.8443)
## l(Oil,16)                     -0.1002 (0.9037)
## l(Oil,17)                      0.4981 (0.7493)
## l(Oil,18)                      0.0096 (0.9699)
## ____________________________ __________________
## S.E. type                     Newey-West (L=7)
## Wald (joint nullity)                   1.7399
## Wald (joint nullity), p-value         0.02607
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Or alternatively:

```r
wald(oil.dyn)
```

```
## Wald test, H0: joint nullity of Oil, l(Oil, 1), l(Oil, 2), l(Oil, 3), l(Oil, 4), l(Oil, 5) and 13 ot
##  stat = 1.73988, p-value = 0.02607, on 19 and 706 DoF, VCOV: Newey-West (L=7).
```

With a p-value of 0.026, we can reject the hypothesis of nullity in the Oil regressors at a 5% significance level

Part e: Construct graphs like those in Figure 16.2, showing the estimated dynamic multipliers, cumulative multipliers, and 95% confidence intervals. Comment on the real-world size of the multipliers.

The above model captured dynamic (non-cumulative) effects. To capture cumulative effects, we would run a related model of up to *seventeen* lags of *differenced* Oil values and then include the 18th lag of *non-differenced* Oil. This means we'll need to create a new variable that is the first difference of the Oil variable:

```
macro %<>% mutate(diff.Oil = Oil-lag(Oil))
oil.cumdyn <- feols(ip.growth ~ l(diff.Oil, 0:17) + l(Oil, 18),
                        macro,
                        panel.id = ~ id + Date,
                        vcov = NW(m))
```

```
## NOTE: 30 observations removed because of NA values (LHS: 1, RHS: 30).
```

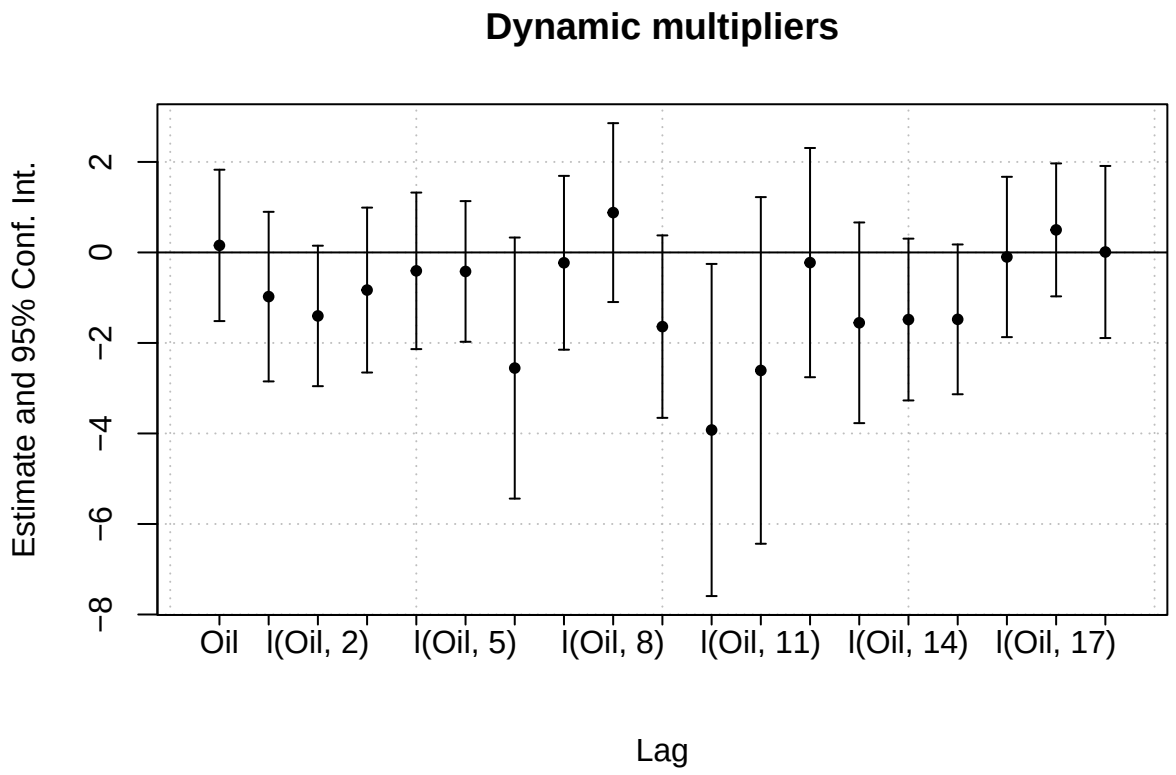
```
etable(oil.dyn, oil.cumdyn)
```

```
##                             oil.dyn         oil.cumdyn
## Dependent Var.:           ip.growth          ip.growth
##
## Constant         0.4275*** (0.0675) 0.4275*** (0.0675)
## Oil                 0.1553 (0.8532)
## l(Oil,1)           -0.9760 (0.9559)
## l(Oil,2)           -1.403. (0.7918)
## l(Oil,3)           -0.8315 (0.9295)
## l(Oil,4)           -0.4064 (0.8824)
## l(Oil,5)           -0.4202 (0.7929)
## l(Oil,6)            -2.555. (1.472)
## l(Oil,7)           -0.2286 (0.9797)
## l(Oil,8)            0.8800 (1.008)
## l(Oil,9)            -1.639 (1.028)
## l(Oil,10)          -3.923* (1.872)
## l(Oil,11)           -2.607 (1.954)
## l(Oil,12)          -0.2247 (1.292)
## l(Oil,13)           -1.555 (1.131)
## l(Oil,14)          -1.483 (0.9120)
## l(Oil,15)          -1.479. (0.8443)
## l(Oil,16)          -0.1002 (0.9037)
## l(Oil,17)           0.4981 (0.7493)
## l(Oil,18)           0.0096 (0.9699)  -18.29*** (4.478)
## diff.Oil                              0.1553 (0.8532)
## l(diff.Oil,1)                        -0.8207 (1.150)
## l(diff.Oil,2)                         -2.224 (1.391)
## l(diff.Oil,3)                        -3.055. (1.786)
## l(diff.Oil,4)                        -3.462. (1.964)
## l(diff.Oil,5)                        -3.882. (2.298)
## l(diff.Oil,6)                        -6.437* (2.884)
## l(diff.Oil,7)                        -6.666* (2.813)
```

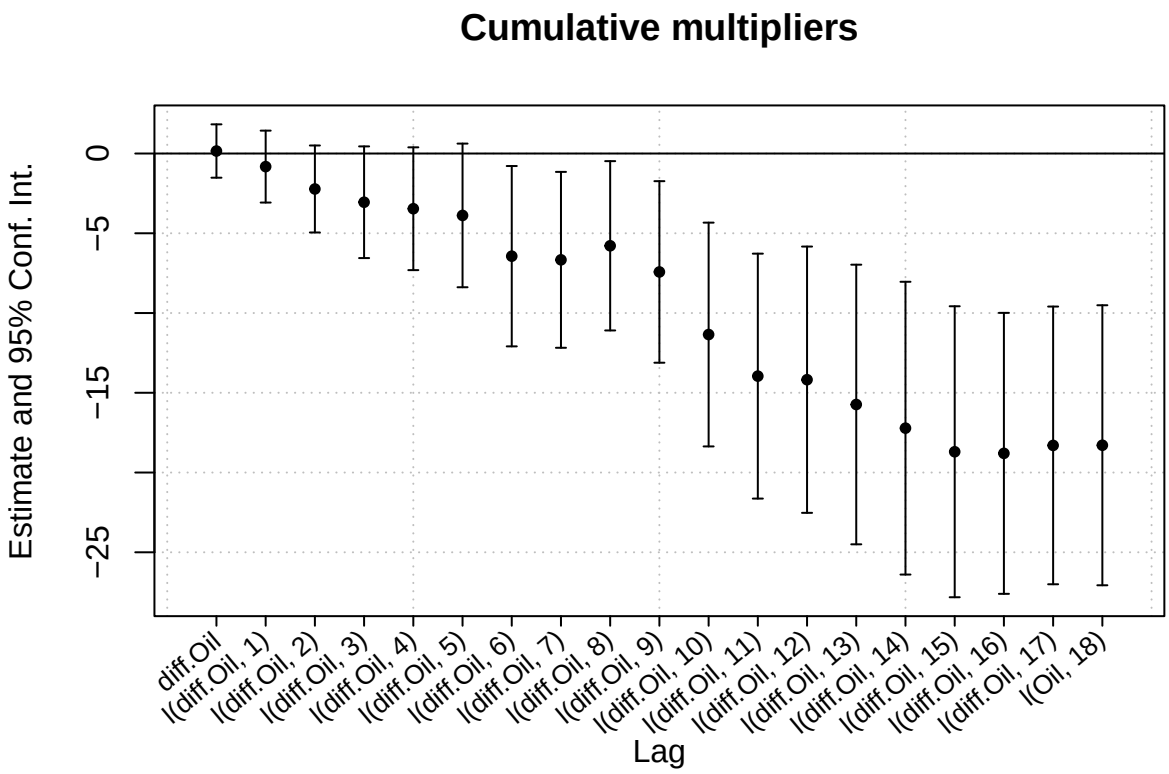
```
## l(diff.Oil,8)                          -5.786* (2.708)
## l(diff.Oil,9)                          -7.425* (2.903)
## l(diff.Oil,10)                        -11.35** (3.579)
## l(diff.Oil,11)                        -13.96*** (3.917)
## l(diff.Oil,12)                        -14.18*** (4.259)
## l(diff.Oil,13)                        -15.74*** (4.472)
## l(diff.Oil,14)                        -17.22*** (4.682)
## l(diff.Oil,15)                        -18.70*** (4.653)
## l(diff.Oil,16)                        -18.80*** (4.493)
## l(diff.Oil,17)                        -18.30*** (4.441)
## --------------- ------------------ ------------------
## S.E. type          Newey-West (L=7)   Newey-West (L=7)
## Observations                   726                726
## R2                         0.07860            0.07860
## Adj. R2                    0.05380            0.05380
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Then it is easy to draw the desired graphs using fixest's coefplot function:

```
coefplot(oil.dyn, drop = 'Constant',
         main = 'Dynamic multipliers', xlab = 'Lag')
```

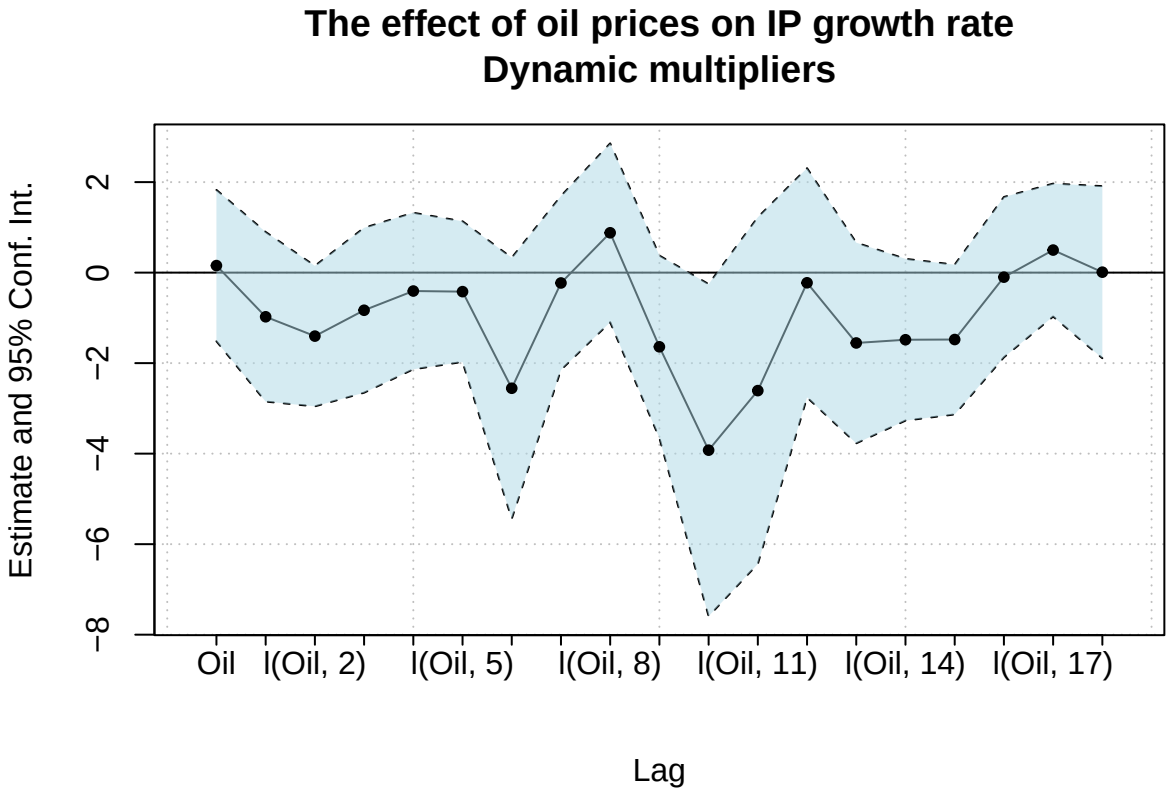



```
coefplot(oil.cumdyn, drop = c('Constant', 'lag(Oil, 18)'),
         main = 'Cumulative multipliers', xlab = 'Lag')
```

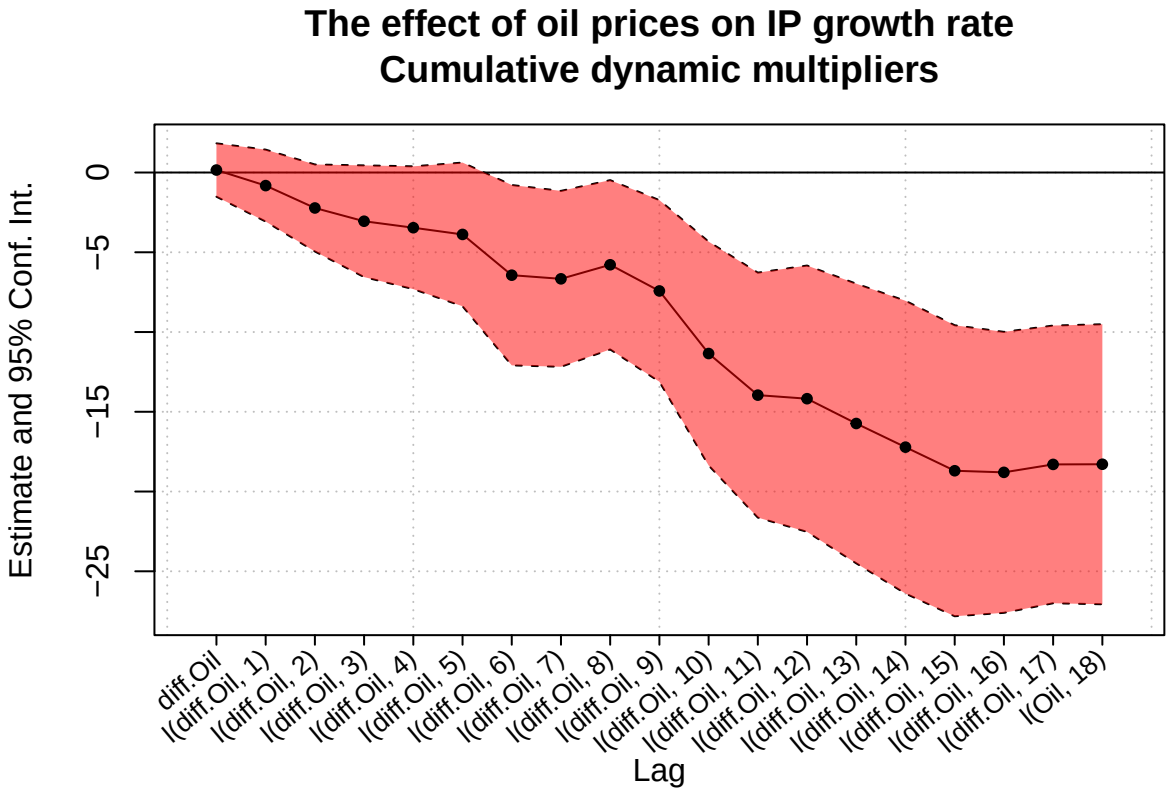


Or with some customization:

```
dyn <- coefplot(oil.dyn, drop = 'Constant',
         main = 'The effect of oil prices on IP growth rate\nDynamic multipliers',
         xlab = 'Lag',
         ci.join = T, ci.lty = 0, ci.fill = T, ci.fill.par = list(col = 'lightblue'),
         pt.join = T)
```




```
cumdyn <- coefplot(oil.cumdyn, drop = 'Constant', ci.join = T,
         main = 'The effect of oil prices on IP growth rate\nCumulative dynamic multipliers',
         xlab = 'Lag',
         ci.lty = 0, ci.fill = T, ci.fill.par = list(col = 'red'),
         pt.join = T)
```



Practice Problems 3-5: Stock-Watson non-empirical exercises 14.1, 14.2, 14.5 on iPad