Dis 11: Big Data (Cont'd); Time Series

1 Shrinkage estimators (from last week)

1. (Adapted from Problem #2 of PS 10)

   You have a sample of size $n = 2$ with data $(x_1, y_1) = (1, 2)$ and $(x_2, y_2) = (3, 6)$. You are interested in the value of β in the regression model

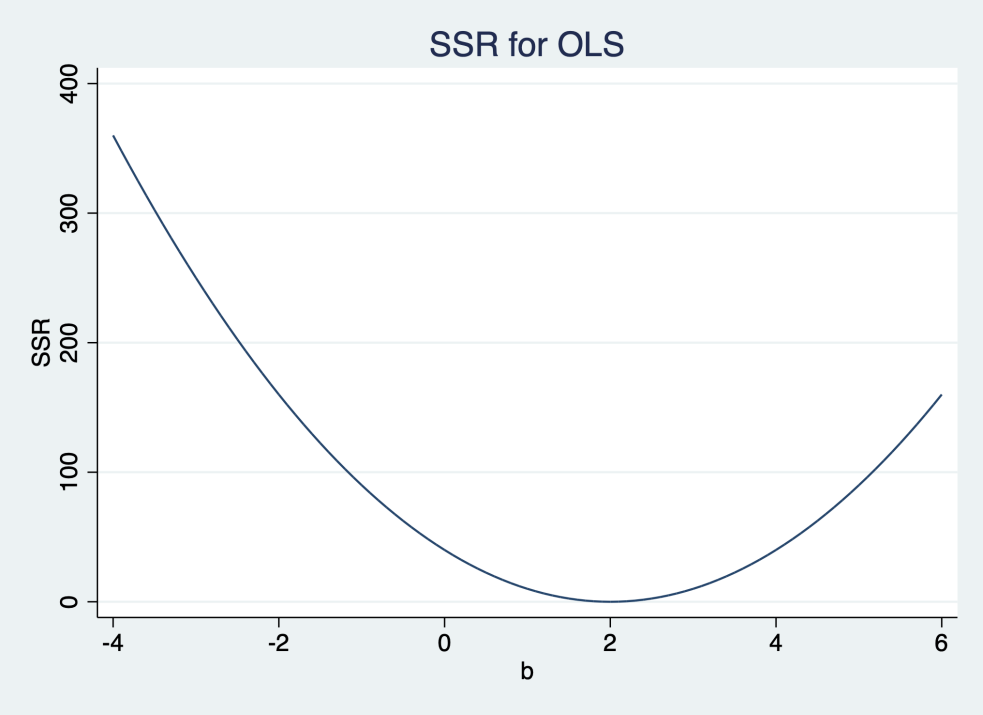
   $$Y = X\beta + u$$

   (notice that this model has no intercept term)

   (a) Plot sum of squared residuals $\sum_i (y_i - bx_i)^2$ as a function of b. Which level of b indicated on the plot is the solution to the OLS estimator?

   To plot the sum of squared residuals (SSR) as a function of b, we need to generate all different levels of b, and then evaluate SSR at these bs. Notice that based on the two data points, the β estimate is expected to be 2, so the b levels that we have in our dataset should include 2.

   Refer to the Do-file solution on how the bs are generated, and how the SSR function is evaluated at all these different bs. The plot of SSR should look like the following:



   (b) Find OLS estimate of β (i.e. $\hat{\beta}_{OLS}$)

   Recall that the OLS estimator minimizes the SSR. That is, whatever level of b solves the following minimization problem should be the OLS estimator:

   $$\min_b \sum_i (y_i - bx_i)^2 \quad \Leftrightarrow \quad \min_b \left[(2-b)^2 + (6-3b)^2\right]$$

Using first order condition, we have

$$\begin{aligned} -2(2-b) - 2 \times 3(6-3b) &= 0 \\ 2 - b + 18 - 9b &= 0 \\ 10b &= 20 \\ b &= 2 \end{aligned}$$

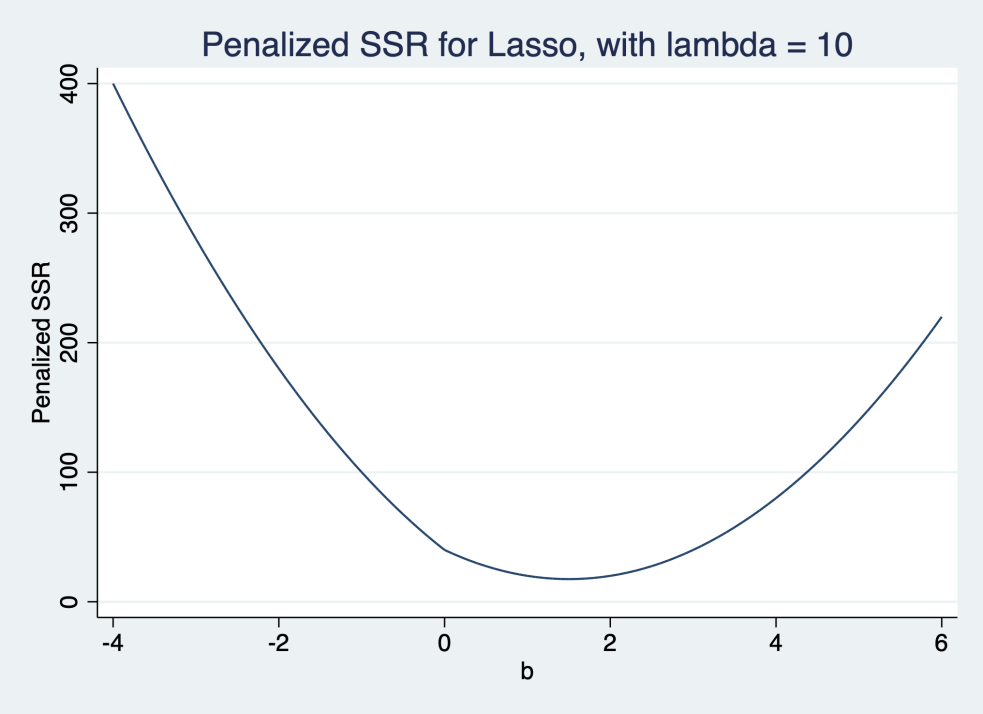
Thus, $\hat{\beta}_{OLS} = 2$, which also aligns with the minimum point at the graph produced in (a).

(c) Plot penalized sum of squared residuals for Lasso regression (which is sum of squared residuals + penalty term) as a function of b with $\lambda_{Lasso} = 10$. Which level of b indicated on the plot is the solution to the Lasso estimator?

Recall that the penalized SSR evaluated for Lasso regression is

$$SSR + \underbrace{\lambda_{Lasso}\,|b|}_{\text{penality term}} = \sum_i (y_i - bx_i)^2 + \lambda_{Lasso}\,|b|$$

Refer to the Do-file solution on how penalized SSR function is evaluated at all these different bs. The plot of penalized SSR for $\lambda_{Lasso} = 10$ should look like the following:



Lasso regressor still minimizes the penalized SSR, which means the level of b at the minimum of the penalized SSR plotted is the solution to the Lasso estimator. Notice that the Lasso estimator is a bit to the left of 2 (which is the OLS estimator). This is the idea of shrinkage: Lasso regressor (and also Ridge) shrinks the β estimate more towards 0.

(d) Find Lasso estimate of β (i.e. $\hat{\beta}_{Lasso}$) under $\lambda_{Lasso} = 10$

Lasso regressor is whatever level of b that solves the following minimization problem:

$$\min_b \sum_i (y_i - bx_i)^2 + \lambda_{Lasso} |b| \quad \Leftrightarrow \quad \min_b \left[(2-b)^2 + (6-3b)^2 + 10|b|\right]$$

From the graph generated in (c), we know that the solution should be $b > 0$. This means that our minimization problem equivalently becomes

$$\min_b \left[(2-b)^2 + (6-3b)^2 + 10b\right]$$

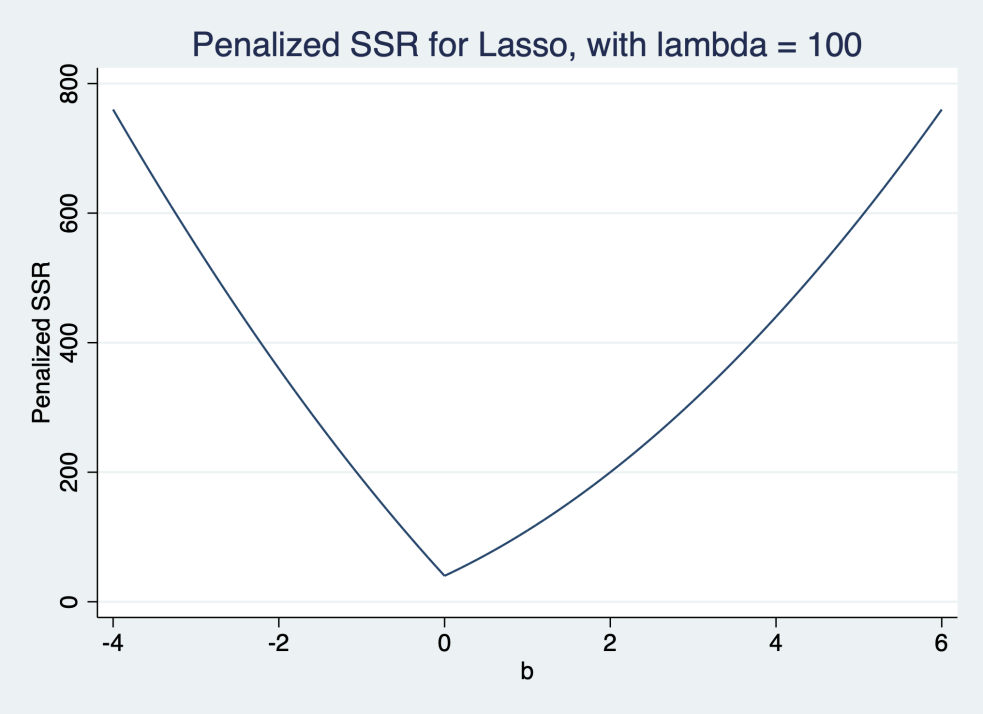
Also from the graph generated in (c), this function is continuous and smooth under $b > 0$, so we can take FOC to solve for the b that minimizes the function:

$$\begin{aligned} -2(2-b) - 2 \times 3(6-3b) + 10 &= 0 \\ 2(2 - b + 18 - 9b) &= 10 \\ 20 - 10b &= 5 \\ b &= 1.5 \end{aligned}$$

Thus, $\hat{\beta}_{Lasso} = 1.5$, which is indeed less than $\hat{\beta}_{OLS} = 2$.

(e) Repeat (c), but set $\lambda_{Lasso} = 100$

Now $\lambda_{Lasso} = 100$, so we are putting more weight on the penalizing term. We would expect the penalized SSR function to look a lot like the plot of $|b|$. The plot is the following:



(f) Find Lasso estimate of β (i.e. $\hat{\beta}_{Lasso}$) under $\lambda_{Lasso} = 100$

From (e), we see that the level of b that minimizes penalized SSR is $b = 0$. Thus, $\hat{\beta}_{Lasso} = 0$ under $\lambda_{Lasso} = 100$. This makes sense, as we put more weight on the penalizing term, we are more and more driving the β estimate towards 0.

2 Time Series

2.1 Terminology

- **Time series data**: Time ordered data with all observations associated with the same sampling unit

year	Y	X_1	X_2
2000	10	5	4
2001	21	3	10
2002	23	10	23
⋮	⋮	⋮	⋮
2019	124	40	83
2020	112	51	93

- How frequent does the time period needs to be?
  - Depends on your analysis.
    Could be hourly, daily, weekly, monthly, quarterly, yearly, etc.
  - But time frequency must be consistent throughout the dataset (for example, you can't have the first ten rows of data in quarterly frequency, and then the rest in yearly frequency)
- Time period is usually denoted as t, and value in that time is subscripted with t (Y_t, X_{1t}, X_{2t}, etc.)
- Lags and differences:
  - The **first lag of** Y_t is Y_{t-1}. The **j-th lag of** Y_t is Y_{t-j}.
  - The **first difference of** Y_t is $\Delta Y_t = Y_t - Y_{t-1}$
  - The **first difference of the logarithm of** Y_t is $\Delta \ln(Y_t) = \ln(Y_t) - \ln(Y_{t-1})$
  - The **percentage change of** Y_t **between** t **and** $t-1$ is approximately $100\Delta \ln(Y_t)$

2.2 What do we do with time series data?

- Time series data, unlike cross sectional or panel data, is usually used for **prediction**.
  - So, like what we talked about last week regarding Big Data, we're interested in the prediction aspect of econometric analysis when dealing with time series data
  - However, unlike last week regarding X and Y variables have to be standardized / demeaned in big data prediction model, we usually **don't** do that for small scale time series prediction.
    (The reason why we standardize / demean variables in big data models is to make all βs to be in terms of increase one standard deviation of X, since we have so many X variables to include in a big data model, and the units of X become hard to keep track of.)
- So how to conduct prediction for time series data?
  - Time series data are often **autocorrelated** (also known as **serially correlated**): Past value(s) of Y should be correlated with current or future value of Y.

    - j-th autocovariance $= Cov(Y_t, Y_{t-j})$
    - j-th autocorrelation $= Corr(Y_t, Y_{t-j}) = \frac{Cov(Y_t, Y_{t-j})}{\sqrt{Var(Y_t)Var(Y_{t-j})}}$
  - **Stationarity**: A time series Y_t is stationary if its probability distribution does NOT change over time. In other words, the distribution of the time series today is the same as its distribution in the past.

    Variables can also be jointly stationary. For example, (X_t, Y_t) can jointly have probability distribution that does NOT change over time.
  - Therefore,
    - If the variable to forecast is stationary, then try to include lagged value of Y_t (and potentially lagged value of a different variable X_t) to predict future Y.
      $\Rightarrow$ Autoregression & Autoregressive distributed lag model
    - If the forecast variable is NOT stationary, then **trend** and **breaks** need to be accounted for.

- **Autoregression**
  - A p-th order autoregressive model, or $AR(p)$, looks like the following:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \ldots + \beta_p Y_{t-p} + u_t$$

  - What are these βs?

    Recall that β_1 is calculated as

$$\beta_1 = \frac{Cov(Y_t, Y_{t-1})}{Var(Y_{t-1})}$$

    this looks awful a lot like the first autocorrelation formula.

    In fact, since we assume that Y_t is stationary,

$$Var(Y_t) = Var(Y_{t-1})$$

    given that distribution of Y at time t is exactly the same as the distribution of Y at time $t-1$. This gives us

$$\text{first autocorrelation} = \frac{Cov(Y_t, Y_{t-1})}{\sqrt{Var(Y_t)Var(Y_{t-1})}} = \frac{Cov(Y_t, Y_{t-1})}{Var(Y_{t-1})} = \beta_1$$

    Thus, under stationarity, β_1 records the first order autocorrelation.

    By extension, under stationarity, β_p records the p-th order autocorrelation.

- **Autoregressive distributed lag model**
  - A autoregressive distributed lag model with p lags of Y_t and q lags of X_t, or $ADL(p, q)$, looks like the following:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \ldots + \beta_p Y_{t-p} + \delta_1 X_{t-1} + \ldots + \delta_q X_{t-q} + u_t$$

- Both $AR(p)$ and $ADL(p, q)$ are estimated using OLS. But let's see how OLS assumptions under a multivariate environment hold up under time series environment:

	OLS Assumptions	Time Series Assumptions
1	**Zero conditional mean**: $E[u_i \mid X_{1i}, X_{2i}, \ldots, X_{Ki}] = 0$	**Zero conditional mean**: $E[u_t \mid Y_{t-1}, \ldots, Y_{t-p}, X_{1,t-1}, \ldots, X_{1,t-q}, \ldots, X_{K,t-1}, \ldots, X_{K,t-q}] = 0$
2	**I.I.D. Data**: $(x_{1i}, x_{2i}, \ldots, x_{ki}, y_i)$ are i.i.d. (independent and identically distributed)	**a. Stationarity**: The distribution of the time series today is the same as its distribution in the past. ($\Rightarrow$ replace "identically distributed") **b. Independence for large time gap**: $(Y_t, X_{1t}, \ldots, X_{Kt})$ and $(Y_{t-j}, X_{1,t-j}, \ldots, X_{K,t-j})$ becomes independent as j gets large. ($\Rightarrow$ replace "independent")
3	**Large outliers are unlikely**: There doesn't exist $(X_{1i}, X_{2i}, \ldots, X_{Ki}, Y_i)$ that live in a dramatically different region.	**Large outliers are unlikely**: There doesn't exist $(X_{1t}, X_{2t}, \ldots, X_{Kt}, Y_t)$ that live in a dramatically different region.
4	**No perfect multicollinearity**: One of the regressors cannot be a perfect linear function of the other regressors.	**No perfect multicollinearity**: One of the regressors cannot be a perfect linear function of the other regressors.

- How to determine how many lags to include in an $AR(p)$ or $ADL(p,q)$ model?

  Idea: The marginal benefits of including one more lag outweighs the marginal costs of doing so.
- **Marginal benefits of one more lag term**: More information, reduces omitted variable bias.
- **Marginal costs of one more lag term**: Introduces additional estimation error.
  - In time series context, the predicted future Y is refered to as **forecast**, and the prediction error is referred to as **forecast error**, where

$$\text{Forecast Error} = Y_{T+1} - \hat{Y}_{T+1|T}$$

    * Y_{T+1} represents the true value of Y at time $T+1$
    * $\hat{Y}_{T+1|T}$ represents the forecast of Y at time $T+1$, using all data up to time period T to construct the forecast
  - Similar to what we saw from Big Data, we often are interested in the average level of forecast error, which we call **mean squared forecast error (MSFE)**:

$$MSFE = E\left[\left(Y_{T+1} - \hat{Y}_{T+1|T}\right)^2\right]$$

  - The tricky bit is, how do we estimate MSFE? Let's try some transformation on it first:

$$\begin{aligned} MSFE &= E\left[\left(Y_{T+1} - \mu_Y - (\hat{Y}_{T+1|T} - \mu_Y)\right)^2\right] \\ &= E\left[\left(Y_{T+1} - \mu_Y\right)^2 + \left(\hat{Y}_{T+1|T} - \mu_Y\right)^2 - 2\left(Y_{T+1} - \mu_Y\right)\left(\hat{Y}_{T+1|T} - \mu_Y\right)\right] \end{aligned}$$

$$= \underbrace{E\left[(Y_{T+1} - \mu_Y)^2\right]}_{\text{variance from randomness}} + \underbrace{E\left[\left(\hat{Y}_{T+1|T} - \mu_Y\right)^2\right]}_{\text{variance from estimation}} - 2 \underbrace{E\left[(Y_{T+1} - \mu_Y)\left(\hat{Y}_{T+1|T} - \mu_Y\right)\right]}_{=0 \text{ since prediction arises independently of the truth}}$$
$$= E\left[(Y_{T+1} - \mu_Y)^2\right] + E\left[\left(\hat{Y}_{T+1|T} - \mu_Y\right)^2\right]$$

For example, for an $AR(p)$ model,

$$\begin{aligned} MSFE &= \text{variance from randomness} + \text{variance from estimation} \\ &= \sigma_u^2 + Var(\hat{\beta}_0 + \hat{\beta}_1 Y_T + \ldots \hat{\beta}_p Y_{T-p+1}) \end{aligned}$$

– For an $AR(p)$ model, there are 3 methods to estimate MSFE:

1. **Using standard error of the regression**:
   * Only use when T is large relative to p
   * Idea: When T is large enough, the variance from estimation will become negligible. This makes $MSFE \approx \sigma_u^2$, which means the estimate of MSFE becomes

$$\widehat{MSFE}_{SER} = s_{\hat{u}}^2 = \frac{SSR}{T-p-1} = \frac{\sum_t (Y_t - \hat{Y}_t)^2}{T-p-1}$$

2. **Using the final prediction error**:
   * Use when T is NOT large relative to p & error is homoskedastic
   * Idea: Under homoskedastic error, we can simplify the variance from estimation as the following:

$$Var(\hat{\beta}_0 + \hat{\beta}_1 Y_T + \ldots \hat{\beta}_p Y_{T-p+1}) \approx \sigma_u^2 \left(\frac{p+1}{T}\right)$$

   which means that MSFE now becomes

$$MSFE \approx \sigma_u^2 + \sigma_u^2 \left(\frac{p+1}{T}\right) = \sigma_u^2 \left(\frac{T+p+1}{T}\right)$$

   Thus, to estimate MSFE, we use

$$\widehat{MSFE}_{FPE} = s_{\hat{u}}^2 \left(\frac{T+p+1}{T}\right) = \frac{SSR}{T-p-1}\left(\frac{T+p+1}{T}\right) = \frac{T+p+1}{T-p-1} \cdot \frac{\sum_t (Y_t - \hat{Y}_t)^2}{T}$$

3. **Using pseudo out-of-sample forecasting**:
   * Use whenever (though T cannot be too small); similar idea to big data prediction model
   * Procedure:
   (a) For a sample with T time periods, choose W observations to set aside for prediction.
   (b) The size of sample used for model estimation is $S = T - W$. Use $t = 1, 2, \ldots, S$ to estimate the model.
   (c) Compute forecast of $S+1$ time period. Obtain forecast error: $e_{S+1} = Y_{S+1} - \hat{Y}_{S+1|S}$.
   (d) Now use $W-1$ period to set aside for prediction, meaning that the size of sample to estimate the model is now $S+1$, with time period $t = 1, 2, \ldots, S, S+1$. Forecast outcome at time $S+2$ and obtain forecast error.

(e) Repeat, and stop after the sample set aside for prediction reduces its size to 1.

The estimated MSFE is

$$\widehat{MSFE}_{POOS} = \frac{1}{W}\sum_{j=S+1}^{T} e_j^2$$

– Side note: How to obtain forecast interval?

We usually assume error in the forecast model is normally distributed. Thus, the way to construct confidence interval for a forecast is very similar to the confidence interval we have been constructing for coefficient estimate $\hat{\beta}$:

$$\text{point forecast} \pm z_{1-\frac{\alpha}{2}} \times se(\text{point estimator}) \quad \Leftrightarrow \quad \hat{Y}_{T+1|T} \pm z_{1-\frac{\alpha}{2}} \times \sqrt{\widehat{MSFE}}$$

where α = significance level = 1 − confidence level.

- Going back to what we were talking about: How do we balance the marginal benefits and marginal costs of introducing an additional lag term in our forecast models?
  - **F-statistic approach**: Start with some number of lags, and keep testing whether the largest lag is statistically significant. If yes, keep it. If no, drop it and re-estimate the model.
  - **Use information criterion**: An information criterion is a function of p (or the number of lags in general). It's essentially a penalty function, so the **smaller** the information criterion is at p, the better.

    Under an $AR(p)$ model:

    * AIC (Akaike information criterion):

$$AIC(p) = \ln\left[\frac{SSR(p)}{T}\right] + (p+1)\frac{2}{T}$$

    * BIC (Bayes information criterion):

$$BIC(p) = \ln\left[\frac{SSR(p)}{T}\right] + (p+1)\frac{\ln(T)}{T}$$

    Under an $ADL(p,q)$ model, the total number of predictors included is $K = p+q+1$, so that

$$AIC(K) = \ln\left[\frac{SSR(K)}{T}\right] + K\frac{2}{T} \qquad BIC(K) = \ln\left[\frac{SSR(K)}{T}\right] + K\frac{\ln(T)}{T}$$

    To obtain AIC and BIC in Stata, run the regression model first, then store the estimates, and finally use `estimate stats` to produce AIC and BIC:

```
reg Y l.Y, robust // first model
estimates store model1
reg Y l.Y l(1/2).Y, robust // second model
estimates store model2
estimate stats model1 model2 // obtain AIC and BIC
```

    We'll see an example in the Problems section.

3 Problems

2. (Time Series)

   Load this week's dataset* into into Stata (don't forget to change your working directory).

   This dataset records real GDP growth rate from the second quarter of 1947 to the second quarter of 2018. Variable that records such growth rate is `gdp`.

   We will assume throughout this question that real GDP growth is stationary, and attempt to use both $AR(p)$ and $ADL(p,q)$ models to forecast real GDP growth. Assume error is heteroskedastic for all parts of analysis.

   (a) Start by running an $AR(1)$ model to forecast real GDP growth.
   An $AR(1)$ model should look like the following:

   $$Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t$$

   Here, $Y_t = \texttt{gdp}_t$. In Stata, this means that we are regressing `gdp` on the first lag of `gdp`. See Do-file solution for code.

   (b) Run an $AR(2)$ model. Is there any concern about including the second lag? Explain.
   An $AR(2)$ model should look like the following:

   $$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + u_t$$

   which means that we should include both the first and the second lag of Y in our regression.
   Notice that this dataset records quarterly observation. When evaluating whether a second lag should be included, we can think both qualitatively and quantatively:

   - Qualitatively: Do we think the second lag of Y makes sense in predicting Y? If Y has a strong second order serial correlation, then maybe this is valid. In the context of GDP, it's pretty reasonable to expect that GDP growth from the last two quarters are related to the GDP growth of the next quarter, but as the number of lags increase, this connection becomes harder and harder to establish.
   - Quantatively: The regression output table has the p-value on the second lag's coefficient to be 0.108, which is not statistically significant at 5% size. One might thus worry whether it's worthwhile to include the second lag. However, since our goal now is prediction / forecast, statistical significance might not be all that we care about. We'll see in a bit that information criterion might be better at evaluating what to include in a forecast model.

   (c) Proceed to run $AR(p)$ models up to $p = 4$. Use AIC, decide how many lags should be included in an $AR(p)$ model. Repeat the exercise for using BIC.
   Refer to Do-file solution for Stata code needed for this part. The resulting table that allows us to use AIC and BIC to evaluate model is the following:

*from Econ 460 (Economic Forecasting) taught by Prof. Bruce Hansen in Fall 2018

```
. estimate stats `ar_models'

Akaike's information criterion and Bayesian information criterion
```

Model	N	ll(null)	ll(model)	df	AIC	BIC
ar1	284	-786.8058	-767.0295	2	1538.059	1545.357
ar2	283	-783.9882	-762.6945	3	1531.389	1542.325
ar3	282	-781.3838	-757.6582	4	1523.316	1537.884
ar4	281	-778.8179	-754.5937	5	1519.187	1537.379

Note: BIC uses N = number of observations. See [R] BIC note.

The model with the lowest AIC / BIC is the best. Hence, if we use AIC, then $AR(4)$ would be selected. Using BIC, $AR(4)$ would also be selected.

(d) Now consider an $ADL(1,1)$ model, with the X variable to include here being the real growth rate of private domestic investment (pdi in the dataset).

An $ADL(1,1)$ model should look like the following:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \delta_1 X_{t-1} + u_t$$

Here, $Y_t = \text{gdp}_t$, and $X_t = \text{pdi}_t$. In Stata, this means that we are regressing gdp on the first lag of gdp, and the first lag of pdi. See Do-file solution for code.

(e) Proceed to run $ADL(p,q)$ models with all possible combinations of p and q, such that $p = 1,2,3,4$ and $q = 1,2,3,4$. Use AIC, how many lags should be included? What about using BIC?

Refer to Do-file solution for Stata code needed for this part. For all possible combinations, this means we're evaluating all possible ways that (p,q) can be. For example, when $p = 1$, q can be either 1, 2, 3, or 4. When $p = 2$, q can still be either 1, 2, 3, or 4. Repeat this for also $p = 3$ and $p = 4$ will give us all possible combinations of p and q.

The resulting table that allows us to use AIC and BIC to evaluate model is the following:

```
. estimate stats `adl_models'

Akaike's information criterion and Bayesian information criterion
```

Model	N	ll(null)	ll(model)	df	AIC	BIC
adl11	284	-786.8058	-766.9127	3	1539.825	1550.772
adl12	283	-783.9882	-762.3332	4	1532.666	1547.248
adl13	282	-781.3838	-758.1746	5	1526.349	1544.559
adl14	281	-778.8179	-754.0937	6	1520.187	1542.017
adl21	283	-783.9882	-762.4956	4	1532.991	1547.573
adl22	283	-783.9882	-762.1032	5	1534.206	1552.434
adl23	282	-781.3838	-757.7559	6	1527.512	1549.363
adl24	281	-778.8179	-753.3444	7	1520.689	1546.157
adl31	282	-781.3838	-757.5823	5	1525.165	1543.374
adl32	282	-781.3838	-757.2478	6	1526.496	1548.347
adl33	282	-781.3838	-757.2221	7	1528.444	1553.937
adl34	281	-778.8179	-753.0901	8	1522.18	1551.287
adl41	281	-778.8179	-754.5934	6	1521.187	1543.017
adl42	281	-778.8179	-754.3545	7	1522.709	1548.178
adl43	281	-778.8179	-754.3241	8	1524.648	1553.755
adl44	281	-778.8179	-752.9723	9	1523.945	1556.69

Note: BIC uses N = number of observations. See [R] BIC note.

The model with the lowest AIC / BIC is the best. Hence, if we use AIC, then $ADL(1,4)$ – that is, one lag on Y, and four lags on X – would be selected. Using BIC, $ADL(1,4)$ would also be selected.

(f) Consider all the $AR(p)$ models and $ADL(p,q)$ models that we run. Using AIC, which model should we select? What about using BIC?

We can compare AIC and BIC across all models that we run. The resulting table that allows us to use AIC and BIC to evaluate model is the following:

```
. estimate stats `ar_models' `adl_models'

Akaike's information criterion and Bayesian information criterion
```

Model	N	ll(null)	ll(model)	df	AIC	BIC
ar1	284	-786.8058	-767.0295	2	1538.059	1545.357
ar2	283	-783.9882	-762.6945	3	1531.389	1542.325
ar3	282	-781.3838	-757.6582	4	1523.316	1537.884
ar4	281	-778.8179	-754.5937	5	1519.187	1537.379
adl11	284	-786.8058	-766.9127	3	1539.825	1550.772
adl12	283	-783.9882	-762.3332	4	1532.666	1547.248
adl13	282	-781.3838	-758.1746	5	1526.349	1544.559
adl14	281	-778.8179	-754.0937	6	1520.187	1542.017
adl21	283	-783.9882	-762.4956	4	1532.991	1547.573
adl22	283	-783.9882	-762.1032	5	1534.206	1552.434
adl23	282	-781.3838	-757.7559	6	1527.512	1549.363
adl24	281	-778.8179	-753.3444	7	1520.689	1546.157
adl31	282	-781.3838	-757.5823	5	1525.165	1543.374
adl32	282	-781.3838	-757.2478	6	1526.496	1548.347
adl33	282	-781.3838	-757.2221	7	1528.444	1553.937
adl34	281	-778.8179	-753.0901	8	1522.18	1551.287
adl41	281	-778.8179	-754.5934	6	1521.187	1543.017
adl42	281	-778.8179	-754.3545	7	1522.709	1548.178
adl43	281	-778.8179	-754.3241	8	1524.648	1553.755
adl44	281	-778.8179	-752.9723	9	1523.945	1556.69

Note: BIC uses N = number of observations. See **[R] BIC note**.

Choosing the model with the lowest AIC and BIC, we find that both AIC and BIC selects the $AR(4)$ model. This tells us that including lag of other variables isn't necessarily better than just including the lag of the variable to forecast: sometimes including other variables can come with too much costs on increasing the variance of prediction.