Assignment 3

Basic Econometrics Spring 2023

Andreas Dzemski

Due **Feb 9, 18:00h**.
Only hand-ins following the submission guidelines will be accepted.
Starred (*) problems will not be graded.

Problem 8

This problem uses methods that require Stata version 16 or higher.

For this problem, we will use the data set "cps_big_data.dta". I generated this data set from a subsample of the data set "cps.dta" by creating new variables (feature extraction). First, I copied two continuous variables and nine dummies from "cps.dta" and renamed them `cont1`, `cont2`, `dummy1`, ..., `dummy9`. I will refer to these 11 predictors as the predictors that correspond to levels of the economic variables. Then I created new variables by

- adding squares and cubes of the continuous variables (4 new variables),
- interacting all dummies with all continuous variables (18 new variables),
- interacting all continuous variables with each other (1 new variable),
- interacting all dummies with all squares of continuous variables (18 new variables),
- interacting all squares of continuous variables with levels of continuous variables (exclude own interaction, 1 new variable),
- interacting squares of continuous variables with squares of continuous variables (exclude own interaction, 1 new variable).

The interaction variables are called `interact1`, ..., `interact40`. The names of the features have a standardized format. This will make it easy for us to specify predictive models that use a lot of features without having to manually type the names of all the features. In addition to these features the data set contains the outcome variable log wage `lwage` = log(`wage`).

I have already pre-processed the data set and standardized all features (i.e. subtracted the mean and divided by the standard deviation). Such pre-processing is recommended

when working with predictive models (and in particular penalized regression models). Note that after the pre-process step the dummy variables are no longer technically dummy variables (after standardization they typically do not take the values zero or one).

1. We start by dividing the sample into a training subsample (70% of sample) and a testing subsample (30% of sample) by using the following Stata commands.

```
splitsample , generate(sample) split(.7 .3) rseed(1223)
label define slabel 1 "Training" 2 "Test"
label values sample slabel
```

   A new variable has been created. What does this variable indicate? Convince yourself that we have indeed sorted (at least approximately) 70% of the observations into the training subsample and 30% into the test subsample (*Hint:* Use `tabulate`).

2. We first run an OLS regression on the levels of the continuous and the dummy variables using only the training data. We use the `estimates store` command to remember fitted regression models (Don't worry about the details here).

```
reg lwage cont1-cont2 dummy1-dummy9 if sample==1
estimates store ols_levels
reg lwage cont* dummy* interact* if sample==1
estimates store ols_all
```

   How do we tell Stata to use only the training data when fitting the model? Explain what the wildcard operator (*) does. What does `dummy1-dummy9` mean?

3. We will now compute the *training error* of the two regressions.

```
lassogof ols_levels ols_all if sample==1
```

   Which regression has the smaller training error (MSE = mean squared error)? Explain why we could have answered this question even without looking at the data.

4. We will now compute the *test error* of the two regressions.

```
lassogof ols_levels ols_all if sample==2
```

   Which regression has the smaller test error? Why can a regression have small training error but large test error?

5. We now run a Ridge regression and compare it to the OLS regressions.

```
elasticnet linear lwage cont* dummy* interact* if sample==1,
    alpha(0) nolog rseed(1223)
estimates store ridge_all
lassogof ols_all ols_levels ridge_all if sample==2
```

   Ridge regression uses as many features as the OLS regression that regresses on all features. Why does Ridge regression still perform well on the test sample? (*Hint:* Shrinkage.)

6. In addition to the Ridge regression above, run a second Ridge regression using only the levels of the predictors. For the two Ridge regressions, compare the lambda values selected by Stata. For which regression do we apply more shrinkage? Is this choice intuitive?

We now want to look at another penalized regression/shrinkage method, the LASSO (least absolute shrinkage and selection operator). The LASSO solves the following minimization problem

$$\begin{aligned}&\left(\hat{\beta}_0^{\text{lasso}}, \hat{\beta}_1^{\text{lasso}}, \ldots, \hat{\beta}_k^{\text{lasso}}\right)\\ &= \arg\min_{b_0,b_1,\ldots,b_k} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (Y_i - b_0 - b_1 X_{1,i} - \cdots - b_k X_{k,i})^2 + \lambda \sum_{\ell=1}^{k} |b_\ell| \right\},\end{aligned}$$

where λ is regularization parameter. This is very similar to Ridge regression but uses a different penalty function that measures how big the coefficient vector is by looking at the sum of the absolute values of the components (rather than the sum of the squares of the components). The absolute value function has a kink at zero, whereas the operation of taking a square is smooth. A result of this is that in cases where Ridge regression estimates a $\hat{\beta}_j^{\text{ridge}}$ very close to zero, LASSO often estimates a corner solution and sets $\hat{\beta}_j^{\text{lasso}}$ *exactly* equal to zero. If the LASSO estimates $\hat{\beta}_j^{\text{lasso}} \neq 0$ then we say that X_j is *selected* by the LASSO, if the LASSO estimates $\hat{\beta}_j^{\text{lasso}} = 0$ then we say that X_j is *not selected* by the LASSO. This is why the LASSO is called a selection operator.

7. We fit a LASSO regression by using the following commands.

```
elasticnet linear lwage cont* dummy* interact* if sample==1,
    alpha(1) nolog rseed(1223)
estimates store lasso_all
```

   Compare the (estimated) *test error* of Ridge, LASSO and OLS using all features as well as OLS using only the features corresponding to levels.

8. Find out which variables were selected by the LASSO by using the following command.

```
lassocoef, display(coef)
```

   Consider the following statement:

   > "The variables that are not selected by the LASSO (e.g. `cont2` = `age` and `cont2_square` = `age`^{2}) probably do not have an economic effect on (log) wages.

   Explain why this statement is wrong.

Problem 9

For this problem we will be using the data set "ts_minwage.dta". The data set contains the following variables:

variable	description
`emp_foot`	employment in footwear manufacturing
`year`	year
`month`	month (1 = January, 2 = February, ...)
`minwage`	federal hourly minimum wage in US$

1. Open the data in Stata. How is time encoded in the dataset?

2. We now encode time in one variable using the following commands.

```
gen date = ym(year, month)
label variable date "date in year-month format"
```

   Open the data editor and check how the date is encoded. How are the numbers in `date` supposed to be interpreted? (*Hint:* `help datetime`). How would you advance all dates by one year? For the convenience of humans who look at the data, prettify how the date is displayed.

```
format date %tm
```

   Look again at `date`. What has changed?

3. Define new variables `lminwage` and `lemp_foot` corresponding to logged values of `minwage` and `emp_foot`, respectively. Let's plot the two time series `lemp_foot` and `lminwage`. We put them together into the same figure.

```
twoway line lemp_foot date, saving(emp, replace)
twoway line lminwage date, saving(minwage, replace)

gr combine emp.gph minwage.gph, col(1) iscale(1)
```

   Do the time series exhibit (stochastic) trends?

4. We now use

```
tsset date
```

   to tell Stata that we are using a time series and that `date` is the time index. This will allow us to use special commands such as the `D.` operator to refer to first differences and the `L.` operator to refer to lagged values. For example we can write `L.lminwage` to refer to the lagged values (i.e. the previous period's value) of `lminwage`.

5. Compute the correlation between `lemp_foot` and the lagged value of `lminwage` (*Hint:* use the command `cor` and the `L.` operator). If we are interested in predicting the level of the minimum wage based on employment, why would we look at the correlation of employment with the *lagged* minimum wage rather than the *contemporaneous* (i.e. current period) minimum wage? (*Hint:* monthly data).

6. Explain the notion of a "spurious correlation" in the presence of trending time series.

7. From now on we consider first differences. This gets rid of trends and hopefully is a first step towards making our time series stationary. Plot the first difference `D.lemp_foot` against time.

8. Labor markets often exhibit systematic fluctuations over the course of a year. This is called seasonality. We remove the seasonal component from the employment time series by using the following code.

```
tab month , gen(m)
reg D.lemp_foot m2-m12
predict d_lemp_foot_adj , residuals
```

   What does the first line of this piece of code do? The time series `d_lemp_foot_adj` is called *seasonally adjusted*. Explain why the seasonally adjusted time series has no seasonal component (i.e. it will not exhibit a predictable pattern of variation over the course of a year). From now on, we will use the seasonally adjusted time series. Explain why a time series with a seasonal component cannot be stationary.

9. Regress `d_lemp_foot_adj` on the first difference of the lagged minimum wage. Does the minimum wage today predict employment in footwear manufacturing tomorrow ($\alpha = 0.05$, robust standard errors)?

10. Explain why, in a time series context, even (heteroscedasticity) robust standard errors may be incorrect. Compute auto-correlation robust standard errors (Newey-West) by using the following code.

```
newey d_lemp_foot_adj D.L.lminwage , lag(12)
```

Problem 10

For this problem we use the data set "fatality_long_allyears.dta". This is the same data that we used in the lecture. The data set contains the following variables:

variable	description
`fr`	road accident fatality rate per 10,000 population
`tax`	tax in US$ on a case of beer
`year`	year
`state`	state identifier

1. We set up the data set as a panel data set and replicate the panel regression from the lecture.

```
xtset state year
xtreg fr tax, fe
```

   What is the numerical value of the t-statistic for the null hypothesis that tests the coefficient on `fr` against zero? Add the option `vce(cluster state)` to the fixed effect regression. What does it do? After adding the option you'll see a decrease in the realized absolute value of the t-statistic for the null hypothesis that tests the coefficient on `fr` against zero. Explain intuitively why this is expected.

2. Suppose that due to safer cars the fatality rate of road accidents decreases over time. In addition, suppose that state beer taxes tend to increase over time for unrelated reasons. Intuitively argue why we may overestimate the causal effect of the beer tax.

3. We now want to account explicitly for a time trend by including time dummies.

```
tab year, generate(dummy_y)
xtreg fr tax dummy_y2-dummy_y7, fe vce(cluster state)
```

   Is it true that there is a time trend of decreasing number of road accidents?

   Consider the following statement:

   > "If we observe that there is no decreasing time trend then we can just as well estimate the model without the time dummies. This will allow us to use the specification where `tax` is significant, rather than the specification where it is insignificant."

   Explain why this statement is incorrect (*Hint:* p-hacking).

   There is an easier way to add the time dummies to the regression.

```
xtreg fr tax i.year, fe vce(cluster state)
```

   Verify that this command yields the same estimation results as our previous approach.

The time dummies that we included in the previous regression are called *time fixed effects*. Just like an *individual* or *unit fixed effect* has a constant effect (over all time periods) on a unit, a time fixed effect has a constant effect (for all units) on a time period. To write down a panel model with time and unit fixed effects we can write down for example

$$Y_t = \alpha_t + \beta_1 X_{1,t} + \ldots + \beta_k X_{k,t} + A + U_t \qquad \text{for } t = 1, \ldots, T,$$

where α_t is the time fixed effect for time period t and A is the individual or unit fixed effect.

What have you learned?

Econometric skills: After working through this problem set, you can ✎ ✗

- ☐ use a test sample to evaluate a predictive model
- ☐ interpret a ridge regression
- ☐ conduct variable selection using the LASSO
- ☐ understand how to interpret the variables selected by the LASSO
- ☐ seasonally adjust a time series
- ☐ compute standard errors that are robust to serial correlation
- ☐ estimate and interpret a panel regression with unit and time fixed effects.

Programming skills: After working through this problem set, you can ✎ ✗

- ☐ divide the sample into a training and a test subsample
- ☐ compute the MSE on the training and test samples.
- ☐ run a Ridge regression
- ☐ run a LASSO regression and display the selected variables
- ☐ work with time and date formats
- ☐ define a time series using `tsset`
- ☐ convert a categorical variable into dummy variables
- ☐ use the `D.` operator to compute first differences
- ☐ fit a time series regression and compute Newey-West standard errors
- ☐ add levels of a numerical variable as dummy variables to a regression using the `i.` operator.