Statistical Inferences II (Confidence Interval)

Lucie Lu

May 13, 2018

Confidence Invertals

In the previous section, I have obtained the p-value by repeatedly calculating the test statistic from a large number of permutations of the data. On its surface, the permutation confidence interval in this section is simply the set of all values of the parameter for which the null hypothesis is not rejected. To obtain a confidence interval, I tried to use some p-values to test different hypotheses to create a collection of non-rejected hypotheses. This is the confidence interval from the permutation tests.

My null hypothesis are addictive treatment effects. I used two methods to calculate the confidence intervals: one with the uniroot function to find two tails of confidence limits where the p value from the *lm* or *lmrob* function of certain hypothesis is less than 5%, leading to the rejection of that hypothesis. This one is a CLT+IID based test.

The second test sets the significance level 0.05, and specify the confidence coefficient 0.95 to reflect the true coverage probability. I created null hypotheses as a constant addictive effect and compute an intervavl estimate where the right and left end points' p values are approximately equal to 0.05. Within the two-sided confidence intervals, the estimates' p-values are larger than 0.05. I show part of the procedure in this method in table "Searching for non-rejected intervals under permuation for lm and lmrob models."

From the table we can see this second method (see the results in the "permuted lm confidence interval" and "permuted lmrob confidence interval") produces confidence intervals that do not accord with the first method. This may be due to some coding issues that I cannot figure it out at this moment.

	1	2	3	4	5	6	7	8	9	10	11	12	13
additive treatment effect	-10.00	-9.00	-8.00	-7.00	-6.00	-5.00	-4.00	-3.00	-2.00	-1.00	0.00	1.00	2.00
p-values for lm	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.11	0.28	0.45	0.73
p-values for lmrob	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.06	0.17	0.50

	2.5 %	97.5 %
standard lm confidence interval	-8.07	2.40
uniroot lm confidence interval	-8.07	2.40
permuted lm confidence interval	-1.00	7.00
standard lmrob confidence interval	-6.50	-0.12
uniroot lmrob confidence interval	-6.50	-0.12
permuted lmrob confidence interval	0.00	6.00

	Variance–covariance matrix	Sandwich package
Standard OLS	2.68	2.68
HC0	2.53	2.53
HC1	2.57	2.57
HC2	2.58	2.58
HC3	2.62	2.62

	Robust Standard Errors
Clustered HC0	2.53
Clustered HC1	2.57
Clustered HC2	2.58
Clustered HC3	2.62
Clustered SEs from Variance-Covariance matrix	2.57
Wild Bootstrap	2.64
Clustered Wild Bootstrap	2.49

The standard errors drawn from the OLS canned function will be biased if there is a presence of heteroskedasticity of unknown form in my dataset. Note that the assumption the variance of the error term for each x is constant (Homoskedasticity) is not necessary to show that OLS estimators are unbiased. Heteroscedasticity occurs when the variance of the errors varies across observations (Long & Ervin, 2000, p.217). In simple words, when the variance of errors is not constant, one of the important assumptions of a linear model, homoscedasticity is violated. In the presence of heteroscedasticity, ordinary least squares estimates are still consistent, but the tests of significance are generally inappropriate because the standard errors obtained from the classic variance covariance matrix are no longer consistent. We will then perform an invalid statistical inference if we do not correct for the possible presence of heteroskedasticity. Figure 2 reveals that we might have a heteroskedasticity problem.

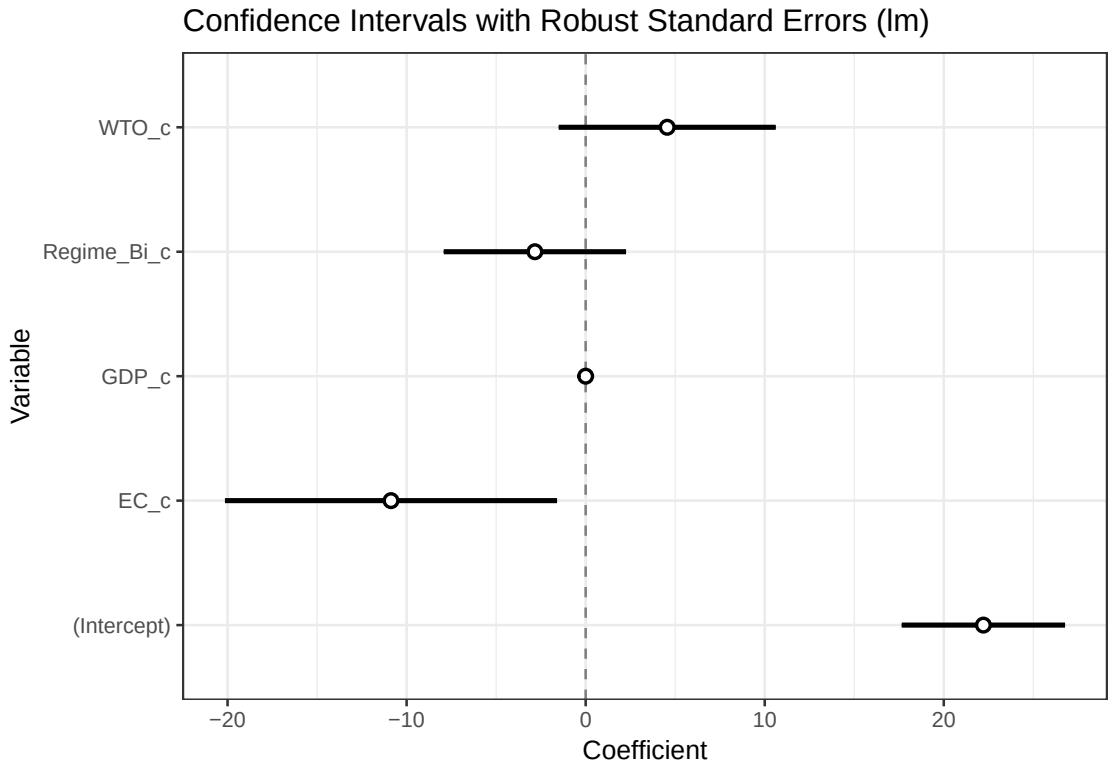
Here, I use a avariety of approaches to correct OLS standard errors, by hands and by using packages. The table shows that the results are similar.

I simplify my regression without controlling for covariates as the following at this moment: $tradepolicy_{i,t} = \beta_0 + \beta_1 * DEMOCRACY_{i,t-1} + \varepsilon_i$.

I calculated the unclusterd HC0, HC1, HC2, and HC3 robust standard errors by heteroskedasticity consistent covariance matrix (HCCME) (also known as White robust errors) and by using a "sandwich" R-statistic package. The results are the same, so I know the results from the package are not misleading. I also use the "wild bootstrap" simulation method to cross-check the results. The "HC" standard errors taking into account the Heteroscedasticity in fact smaller than the default standard errors from the OLS.

The clustered versions of standard errors from different methods are also very close to the unclustered versions. After controlling for clustering, I note that the using different methods including covariance matrix and wild bootstrap, the clustered standard errors and unclustered standard errors turn out to be very similar. It is possibly due to the fact that I have collapsed the data in different periods for the given countries, so that the correlated model errors in different time periods within the country level have been removed. In other words, the variances over time within-country are no longer present in the dataset. The number of clustering at the country level may still exist, but the standard errors controlling for the unknown form of cluster error correlation do not deviate much from the unclusterd robust standard errors. The previous tables show the comparions of the standard errors calculted by different methods.

Confidence Interval

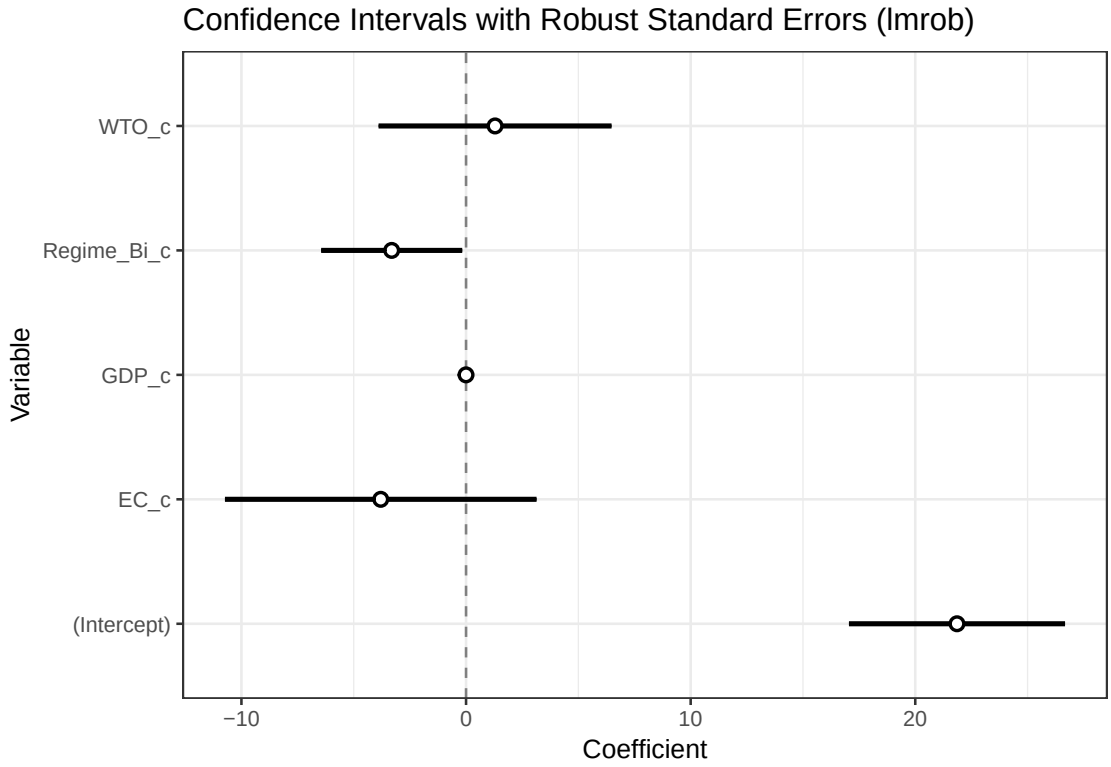


Call: lm-rob(formula = Tariff_c ~ Regime_Bi_c + GDP_c + EC_c + WTO_c, data = mydata_5) --> method = "MM" Residuals: Min 1Q Median 3Q Max -15.47 -4.57 -0.56 5.70 41.22

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 21.857413 2.454387 8.91 2e-13 ***Regime_Bi_c -3.310348 1.602314 -2.07 0.04224***
GDP_c -0.000890 0.000241 -3.69 0.00042 * EC_c -3.796942 3.538998 -1.07 0.28672
WTO_c 1.291864 2.645846 0.49 0.62677
— Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Robust residual standard error: 7.31 Multiple R-squared: 0.171, Adjusted R-squared: 0.127 Convergence in 12 IRWLS iterations

Robustness weights: 3 observations c(3,33,55) are outliers with |weight| = 0 (< 0.0012); 11 weights are ~= 1. The remaining 67 ones are summarized as Min. 1st Qu. Median Mean 3rd Qu. Max. 0.634 0.870 0.945 0.909 0.980 0.999 Algorithmic parameters: tuning.chi bb tuning.psi refine.tol rel.tol solve.tol eps.outlier 1.5476400000 0.5000000000 4.6850610000 0.0000001000 0.0000001000 0.0000001000 0.0012345679 eps.x warn.limit.reject warn.limit.meanrw 0.0000000337 0.5000000000 0.5000000000 nResample max.it best.r.s k.fast.s k.max maxit.scale trace.lev mts 500 50 2 1 200 200 0 1000 compute.rd fast.s.large.n 0 2000 psi subsampling cov compute.outlier.stats "bisquare" "nonsingular" ".vcov.avar1" "SM" seed : int(0)



I plotted the confidence intervals computed by the clustered robust standard errors. I assume the standard errors obtained from the clustered function I wrote are accurate because they have corrected for the possible problem of clustering on the state level and the heteroskedastyin the data structure. I also compute the confidence interval with the White HC1 and the clustered standard errors. They are the same (results are not shown).

In this plot (on the left), we can see the confidence interval for the targeted estimator includes zero. It means that we cannot reject the null hypothesis that the treatment effect equals to zero.

On the right hand side, I ploted the confidence intervals calculated from the default *lmrobust* function. For the targeted estimator, the confidence interval is very close to zero. I cannot reject the possibility that there is no treatment effect. The democratic developing countries on average over time may not have lower tariff rates than non-democratic developing countries.

I also use a simulation method for estimating the coverage probability of the confidence interval calculated from the robust standard error I calculated. Because I have a relatively large sample, I compute the 95% confidence interval as an asymptotic one:

$$\hat{\beta}$$

+/- 1.96*se. After I obtained the confidence interval, I need to check its coverage probability.

Here I explain how I proceed to calculate the coverage probability of the confidence interval. Under simulation, I had 1000 samples drawn from the same population (of sample size from original dataset without replacement). I then use the standard error to compute the confidence interval for each sample. I then computed the proportion of samples for which the *true* mean of the 'population' fall into the confidence interval in each of the 1000 iterations. In such a large repetaed samples, I want to check if the confidence intervals include the true population paramter 95 precent of the time.

Unfortunately, the confidence interval fails to meet its desired object. 100% of the estimated confidence intervals contain the true population mean. This confidence interval is too wide, including the True paramter more than it should. I will risk accepting a null hypothesis when it is false. In other words, my hypothesis testing is underpower: I may fail to reject a null hypothesis when it is false.

It suggests my standard errors are not accurate, or my hypothesis testing fails to achieve its purpose. Or, this result may suggest that my method of calculating the coverage probability is not correct.

```
CP <- NULL
res_t <- matrix (, nrow = 1000, ncol = 100)
for (j in 1:100){
resample_original<-data.frame(t
                              (replicate(1000,
      coef(lm(Tariff_c ~ Regime_Bi_c + mydata_5$GDP_c + mydata_5$EC_c + mydata_5$WTO_c,
              data=sample_n(mydata_5[,c("Regime_Bi_c","Tariff_c")],size=81,replace=F))))))

res_t[,j] <- resample_original$Regime_Bi_c

t <- mean(res_t[,j])
## Coverage probabilities.
n  <- 81
SE <- SEOLS_m5[1]
## Simulate a data set.
#X <- resample_original[,2]

## Construct the CI.
M <- apply(res_t,2,mean)
C <- 1.96*SE

## Check for coverage.
ci <- ((M-C < 0) & ( M+C > 0))
ci

# or that it is
#ci <- (M-C)<=0 # if the coefficient is positive
#ci <- (M+C)>=0 # if the coefficient is negative
# Then do
CP[j] <- mean(ci)
}

CP[j]
```

```
[1] 1
```