NAME: Geonho Marco ID NUMBER: 122200
SURNAME: You

ECONOMETRICS AND DATA ANALYSIS

Home Assessment N.3 - Solution

Deadline: 03 January 2021, 11h59pm [Paris time]

Contents

Problem 1 – Classical Measurement Error

Set-up

Consider the following population regression model

$$Y_i = \beta_0 + \beta_1 X_i + u_i$$

Suppose that instead of properly measuring X_i, we incorrectly measure it with a measurement error: we observe $\widetilde{X}_i$. As a consequence, you end up estimating

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 \widetilde{X}_i + \underbrace{\beta_1 (X_i - \widetilde{X}_i) + u_i}_{v_i} \\ &= \beta_0 + \beta_1 \widetilde{X}_i + v_i \end{aligned}$$

where $\widetilde{X}_i$ and the error term v_i might be correlated.

Assume that the measurement error $w_i = \widetilde{X}_i - X_i$ has zero mean, is uncorrelated with the variable X_i and with the error term of the population regression model u_i:

$$\begin{aligned} \widetilde{X}_i &= X_i + w_i \\ \rho_{w,u} &= 0 \\ \rho_{w,X} &= 0 \end{aligned}$$

and as a consequence, that

$$\hat{\beta}_1 \xrightarrow{p} \frac{\sigma_X^2}{\sigma_X^2 + \sigma_w^2} \beta_1$$

Assume that (X, Y) are jointly normally distributed according with

$$(X, Y) \sim N\left[\begin{pmatrix} 50 \\ 100 \end{pmatrix}, \begin{pmatrix} 10 & 5 \\ 5 & 10 \end{pmatrix} \right]$$

and that you do not observe X_i but only $\widetilde{X}_i = X_i + w_i$ where w_i are i.i.d and normally distributed random variable drawn from $N(0, 10)$.

Question 1

Set seed to 123, draw an i.i.d sample of (X_i, Y_i), generate $\widetilde{X}_i$ and write the R code to estimate β_1 in

$$Y_i = \beta_0 + \beta_1 X_i + u_i$$
$$Y_i = \beta_0 + \beta_1 \widetilde{X}_i + v_i$$

Briefly discuss your results plotting in a graph the population and the two sample regression functions.

Solution 1

The Stargazer library imports stargazer function which is similar to summary function but better in visualisation.

```
library(stargazer)
```

Referring to the bivariate normal distribution of X and Y, X has mean of 50 and variance of 10, Y has mean of 100 and variance of 10. X and Y have non null covariance equal to 5. Given covariance of X and Y, variance of X, and variance of Y, the correlation coefficient of X and Y can easily be computed. All the distribution parameters are set as below:

```
# 1. Generate artificial data
## set seed
set.seed(123)
## normal distribution parameters
mu1  <- 50 ; mu2  <- 100
sd1  <- sqrt(10) ; sd2 <- sqrt(10)
cov12 <- 5 ; cor12 <- (cov12/(sd1*sd2))
```

Now that all the distribution parameters proper to each random variable and their correlation coefficient are provided, the distribution parameter matrices for bivariate normal distribution can be constructed:

```
## bivariate normal distribution parameters
mu   <- c(50,100)
sigma <- matrix(c(sd1^2, sd1*sd2*cor12, sd2*sd1*cor12, sd2^2),2)
```

From the distribution parameter matrices, mvrnorm function can generate random samples of X and Y. The sample size is arbitrary (here the sample size is set at 1000).

```
## random sample generation
N  <- 1000
XY <- mvrnorm(N,mu,sigma)
X  <- XY[, 1] ; Y <- XY[, 2]
```

Lastly, measurement error that follows a normal distribution of mean of 0 and variance of 10 is generated and added to the random sample X

```
## measurement error
w  <- rnorm(N, mean = 0, sd = 10)
Xt <- X + w
```

Remind that there are 3 datasets Y, X, and $\widetilde{X}$.

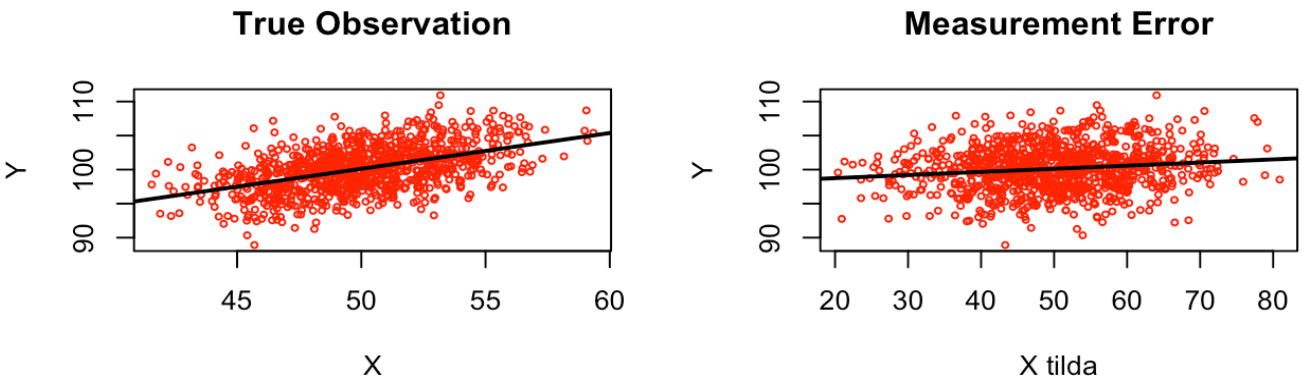
First, linear regression of Y on X:

```
# regression with correct sample (no measurement error)
regress1 <- lm(formula = Y ~ X)
stargazer(regress1, type="latex") # to generate the summary table in LaTeX format
beta.01 <- coef(regress1)
beta.0 <- as.numeric(beta.01["(Intercept)"])
beta.1 <- as.numeric(beta.01["X"])
# plot regress1
par(mfrow=c(2,2))
plot(X, Y, col = "red", xlab = "X", main = "True Observation", cex=.5)
abline(beta.01, lwd = 2)
```

Second, linear regression of Y on $\widetilde{X}$:

```
# regression with errored sample
regress2 <- lm(formula = Y ~ Xt)
stargazer(regress2, type="latex")
beta.hat.01 <- coef(regress2)
beta.hat.0 <- as.numeric(beta.hat.01["(Intercept)"])
beta.hat.1 <- as.numeric(beta.hat.01["Xt"])
# plot regress2
plot(Xt, Y, col = "red", xlab = "X tilda", main = "Measurement Error", cex=.5)
abline(beta.hat.01, lwd = 2)
```

Above two regression plot the following graphs:



Inferential statistics results of the two regressions are as below

The first regression presents that slope and intercept are approximately 0.526 and 73.815 respectively. They are both statistically very significant. According to its R^2, the model explains 23.8% of the total variability. This approximately corresponds to the squared value of correlation that we computed at the beginning. Overall, the model is well performing.

Table 1: linear regression of Y on X

	Dependent variable:
	Y
X	0.526***
	(0.030)
Constant	73.815***
	(1.492)
Observations	1,000
R^2	0.238
Adjusted R^2	0.237
Residual Std. Error	2.854 (df = 998)
F Statistic	311.620*** (df = 1; 998)
Note:	*p<0.1; **p<0.05; ***p<0.01

In addition to the information provided by the table above, ANOVA test can give us further insights.

```
# ANOVA test regression n.1
anova(regress1)

######### Result #########
Analysis of Variance Table
Response: Y
          Df Sum Sq Mean Sq F value Pr(>F)
X          1 2537.4 2537.41 311.62 < 2.2e-16 ***
Residuals 998 8126.4  8.14
---
Signif. codes: 0 â *** â  0.001  â ** â  0.01  â * â  0.05  â . â  0.1  â   â  1
```

F-value is huge and is significant according to the very low p-value. It concludes that there is a strong relation between X and Y.

The second regression shows different results. It presents slope and intercept that are approximately equal to 0.046 and 97.827 respectively. They are both statistically very significant. According to its R^2, the model explains only 2% of the total variability. It is a huge decrease compared to the regression with no-measurement-error sample. The model has poor performance.

Table 2: linear regression of Y on $\widetilde{X}$

	Dependent variable:
	Y
Xt	0.046***
	(0.010)
Constant	97.827***
	(0.512)
Observations	1,000
R^2	0.020
Adjusted R^2	0.019
Residual Std. Error	3.235 (df = 998)
F Statistic	20.702*** (df = 1; 998)
Note:	*p<0.1; **p<0.05; ***p<0.01

In addition to the information provided by the table above, ANOVA test can give us further insights.

```
# ANOVA test regression n.2
anova(regress2)

######### Result #########
Analysis of Variance Table
Response: Y
          Df  Sum Sq Mean Sq F value  Pr(>F)
Xt         1   216.7 216.703 20.701 6.027e-06 ***
Residuals 998 10447.1 10.468
---
Signif. codes: 0 â*** â 0.001 â** â 0.01 â* â 0.05 â. â 0.1 â â 1
```

F-value is much smaller than the first ANOVA result but is still significant according to its low p-value. It concludes that even though the model's performance has diminished, there still resides a significant relation between $\widetilde{X}$ and Y.

Question 2

Use the information provided in the text above in order to

- suggest a correction for β_1;
- discuss what happens when $V[w_i] \to 0$ and when $V[w_i]$ grows very large with respect to $V[X_i]$.

Solution 2

Recall that with a measurement error w_i such that

$$\widetilde{X}_i = X_i + w_i$$
$$\rho_{w,u} = 0$$
$$\rho_{w,X} = 0,$$

OLS estimator for β_1 becomes

$$\hat{\beta}_1 \xrightarrow{p} \frac{\sigma_X^2}{\sigma_X^2+\sigma_w^2}\beta_1$$

where $\frac{\sigma_X^2}{\sigma_X^2+\sigma_w^2}$ is less than 1.

The simplest correction for this bias might be to invert the multiplier and multiply to estimated β_1 from the sample $\widetilde{X}$.

Let's check it:

```
# beta.hat probability convergence check
check1 <- ((var(X) + var(w))/var(X)) * beta.hat.1 # sample variances
beta.1 - check1
```

The result is

```
# > check1 <- ((var(X) + var(w))/var(X)) * beta.hat.1 # sample variances
# > beta.1 - check1
[1] 0.00158414
```

The difference is marginal, close to 0. The correction is good.

```
# beta corrected
beta.co.1 <- ((var(X) + var(w))/var(X)) * beta.hat.1
```

If $V[w_i]$ tends towards 0, estimated beta converges to true beta in probability because the multiplier will tend towards 1. Given that $E[w_i] = 0$, this is obvious because 0 variance would mean actually no measurement error. On the other hand, if $V[w_i]$ grows very large with respect to $V[X_i]$, it would increase the multiplier resulting in growing bias for estimated beta.

Question 3

Marco is very suspicious that the results you have obtained above depend on the specific sample you have drawn. Show, using a Monte Carlo simulation, that the OLS estimator for β_1 in the model with the measurement error is biased and inconsistent.

Solution 3

The question demands to evaluate unbiasedness and consistency of the OLS estimator. The former can be evaluated with Central Limit Theorem and the latter can be evaluated with Law of Large Numbers.

First, the seed need to be initialised.

```
set.seed(NULL)
```

Unbiasedness of the OLS estimator can be demonstrated with a fixed sample size and increasing iteration numbers for simulation loop.

```
fixed.samp <- 100
iter <- c(25,100,500,1000)
B.bias <- rep(0,length(iter))
for(i in 1:length(iter)){
  Bb <- rep(0,iter[i])
  for(j in 1:iter[i]){
    xy <- mvrnorm(fixed.samp,mu,sigma)
    x  <- xy[, 1]
    y  <- xy[, 2]
    wt <- rnorm(fixed.samp, mean = 0, sd = 10)
    xt <- x + wt

    reg.monte <- lm(formula = y ~ xt)
    beta.01.monte <- coef(reg.monte)
    beta.1.monte <- as.numeric(beta.01.monte["xt"])

    Bb[j]  <- beta.1.monte
  }
  mean.Bb <- mean(Bb)
  B.bias[i] <- mean.Bb
}
```

As the number of iteration increases, the mean should become more accurate. In other words, the mean should either constantly increase or decrease towards a certain number. But the result shows divergence of mean. **There is a bias.**

```
# > B.bias
[1] 0.03997102 0.04833214 0.04464190 0.04611004
```

Consistency of the OLS estimator can be demonstrated with a fixed iteration number and increasing sample sizes for simulation loop.

```
fixed.iter <- 100
samp <- c(25,100,500,1000)
B.cons <- rep(0,length(samp))
for(i in 1:length(samp)){
  Bc <- rep(0,fixed.iter)
  for(j in 1:fixed.iter){
    xy <- mvrnorm(samp[i],mu,sigma)
    x  <- xy[, 1]
    y  <- xy[, 2]
    wt <- rnorm(samp[i], mean = 0, sd = 10)
    xt <- x + wt

    reg.monte <- lm(formula = y ~ xt)
    beta.01.monte <- coef(reg.monte)
    beta.1.monte <- as.numeric(beta.01.monte["xt"])

    Bc[j]  <- beta.1.monte
  }
  sd.Bc <- sd(Bc)
  B.cons[i] <- sd.Bc
}
```

As the number of sample size increases, the variance decreases (this improves accuracy of the mean at the same time). The result shows clear decreasing trend of the variance. **The estimator is consistent**.

```
# > B.cons
[1] 0.066205420 0.029041998 0.012978031 0.009354562
```

Problem 2 – Panel Data and Standard Error

Set-up

In the U.S, there is an on-going debate on the extent to which the right to carry a gun influences crime. Proponents of so-called "Carrying a Concealed Weapon" (CCW) law argue that the deterrent effect of guns prevents crime, whereas opponents argue that the public availability of guns increases their usage and thus makes it easier to commit crimes. With the aim of contributing to this debate, using the "Guns" dataset available in the "AER" package available in R, you are considering to estimate the following regression model

$$\log(violent_i) = \beta_1 \cdot law_i + \beta_2 \cdot density_i + \beta_3 \cdot income_i + \beta_4 \cdot population_i + \beta_5 \cdot afam_i + \beta_6 \cdot cauc_i + \beta_7 \cdot male_i + u_i$$

Question 1

Consider the following models with one or two-way unobserved heterogeneity

$$\log(Y_{it}) = \alpha_i + \sum_i \beta_i X_{it} + u_{it}$$
$$\log(Y_{it}) = \lambda_t + \sum_i \beta_i X_{it} + u_{it}$$
$$\log(Y_{it}) = \alpha_i + \lambda_t + \sum_i \beta_i X_{it} + u_{it}$$

where Y_{it} is the proxy for the variable "violent" in the dataset and the X_{it} are: "law", "density", "income", "population", "afam", "cauc", and "male". Produce a table with descriptive statistics of these variables.

Solution 1

Import the dataset "Guns" from "AER" library and import other useful libraries.

```
library(plm) # panel data linear model
library(AER) # Applied Econometrics with R package
library(stargazer)
library(dplyr)
library(lmtest)
library(sandwich)

# Produce a table with descriptive statistics of these variables
data(Guns)
stargazer(Guns, type = "latex")
summary(Guns)
```

stargazer function gives the following table

Table 3: Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
violent	1,173	503.075	334.277	47.000	283.100	650.900	2,921.800
murder	1,173	7.665	7.523	0.200	3.700	9.800	80.600
robbery	1,173	161.820	170.510	6.400	71.100	192.700	1,635.100
prisoners	1,173	226.580	178.888	19	114	291	1,913
afam	1,173	5.336	4.886	0.248	2.202	6.851	26.980
cauc	1,173	62.945	9.762	21.780	59.940	69.200	76.526
male	1,173	16.081	1.732	12.214	14.653	17.526	22.353
population	1,173	4.816	5.252	0.403	1.188	5.686	33.145
income	1,173	13,724.800	2,554.542	8,554.884	11,934.760	15,271.010	23,646.710
density	1,173	0.352	1.355	0.001	0.032	0.178	11.102

In addition to table 3, summary function provides

```
      year             state          law
 1977   : 51   Alabama   : 23   no :888
 1978   : 51   Alaska    : 23   yes:285
 1979   : 51   Arizona   : 23
 1980   : 51   Arkansas  : 23
 1981   : 51   California: 23
 1982   : 51   Colorado  : 23
 (Other):867   (Other)   :1035
# there are other pieces of information provided by summary function but these are
   the only missed ones by stargazer function
```

According to the above descriptive statistics tables, Guns dataset is a balanced panel data with two factors state and year, and one binary variable law.

Question 2

Estimate the models avoce without any fixed-effects, with entity fixed-effects α_i, with time fixed-effects λ_i and with both. Compare the estimated coefficient associated with "law" across specifications.

Hint: in the two-way FE with a balanced panel data, the correct within transformation is

$$Y_{it} - \overline{Y}_{i.} - \overline{Y}_{.t} + \overline{Y}$$

where $\overline{Y}_{i.}$, $\overline{Y}_{.t}$, and $\overline{Y}$ are the time average within state, the state average within year, and the total average respectively.

Solution 2

First, all the factor proxies need to be declared as variables and also in matrix form for the sake of computation simplicity:

```
# declare columns as variables
violent <- Guns$violent
law <- as.numeric(Guns$law)
density <- Guns$density
income <- Guns$income
population <- Guns$population
afam <- Guns$afam
cauc <- Guns$cauc
male <- Guns$male
state <- Guns$state
year <- Guns$year

# build separate matrices for violent (Y) and other variables (X)
Y  <- cbind(violent)
X  <- cbind(law, density, income, population, afam, cauc, male)
```

Furthermore, Guns dataset will be duplicated with a modification in data type of the law proxy because it's declared as "factor" type:

```
# new data.frame created from Guns modifying law's class from factor to numeric
# as.numeric(law) gives 1 for no 2 for yes so -1 makes it binary
Guns1 <- with(Guns, data.frame(violent, as.numeric(law) - 1, density, income,
    population, afam, cauc, male, as.factor(state), as.factor(year)))
# rename columns
colnames(Guns1)[c(2,9,10)] <- c("law","state","year")
```

Later on, the given log-linear fixed-effects model will be within-transformed to neutralise its numerous intercepts. For this purpose, compute time average of Y and X, and state average of Y and X, then use them to demean Y and X:

```
# Time average of Y and X within a state
ave.time.Y <- cbind(ave(Guns1$violent, Guns1$state, FUN = mean))
ave.time.X <- cbind(ave(Guns1$law, Guns1$state), ave(Guns1$density, Guns1$state),
                    ave(Guns1$income, Guns1$state), ave(Guns1$population, Guns1$state),
                    ave(Guns1$afam, Guns1$state), ave(Guns1$cauc, Guns1$state),
                    ave(Guns1$male, Guns1$state))

# Entity average of Y and X within a year
ave.state.Y <- cbind(ave(Guns1$violent, Guns$year, FUN = mean))
ave.state.X <- cbind(ave(Guns1$law, Guns1$year), ave(Guns1$density, Guns1$year),
                     ave(Guns1$income, Guns1$year), ave(Guns1$population, Guns1$year),
                     ave(Guns1$afam, Guns1$year), ave(Guns1$cauc, Guns1$year),
                     ave(Guns1$male, Guns1$year))

# demeaned Y and X with state index (Y/ave.time.Y because we will log it)
Y.dem.state = Y/ave.time.Y
X.dem.state = X - ave.time.X
# demeaned Y and X with time index (Y/ave.state.Y because we will log it)
Y.dem.time = Y/ave.state.Y
X.dem.time = X - ave.state.X
```

Everything is set. Let's first estimate the model without any fixed-effects. There are two ways to do it:

```
# estimation WITHOUT fixed effects
reg <- lm(formula= log(Y) ~ X, data= Guns)
stargazer(reg, type = "latex")
# PANEL linear model without fixed effects
regp <- plm(formula= log(Y) ~ X, data= Guns, index= c("state","year"), model=
    "pooling", effect= "individual")
stargazer(regp, type = "latex")
```

They produce the same result given in the table 4 below:

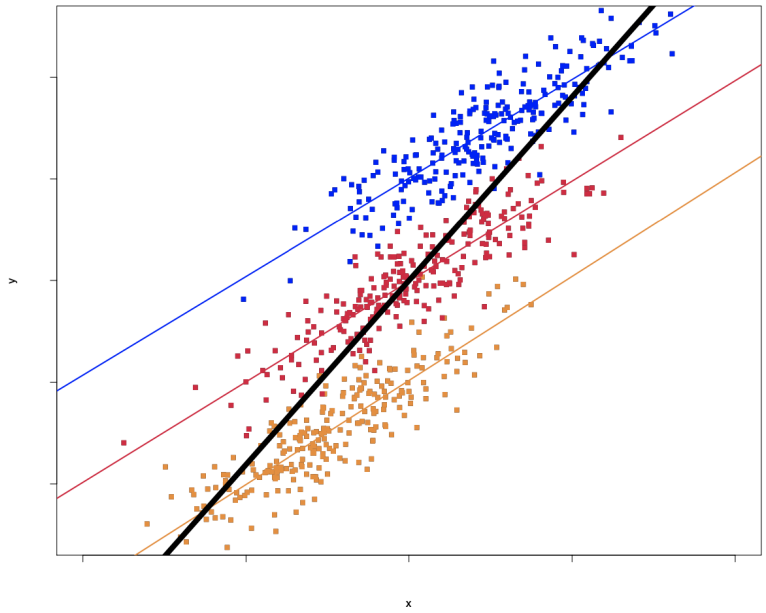
Table 4: Log-Linear Model without any Fixed Effects

	Dependent variable:
	log(Y)
Xlaw	-0.340^{***}
	(0.036)
Xdensity	0.100^{***}
	(0.013)
Xincome	0.00000
	(0.00001)
Xpopulation	0.046^{***}
	(0.003)
Xafam	0.102^{***}
	(0.018)
Xcauc	0.031^{***}
	(0.009)
Xmale	-0.057^{***}
	(0.011)
Constant	4.536^{***}
	(0.584)
Observations	1,173
R^2	0.479
Adjusted R^2	0.476
Residual Std. Error	0.467 (df = 1165)
F Statistic	153.300^{***} (df = 7; 1165)
Note:	*p<0.1; **p<0.05; ***p<0.01

(From now on, only "law" proxy will be interpreted. Other proxies will not be displayed in this report but are available with the attached code.)

The model looks fine with low p-values for every proxy except the income one. R^2 looks plausible as well. But, there is a problem in this kind of regression **when the dataset is a panel data**.

The hidden problem with this is model is that the regression **pools** all the data amongst which some may be clustered like below:



Recall that panel data encompasses data coming from several entities. Pooled panel data ignore the heterogeneity proper to each cluster that are either entity and/or time (here state and year). It's equivalent to say that pooled panel regression will suffer from omitted variables that are heterogeneous across entities and time (or both). If the regression does not take this heterogeneity into account, the OLS estimator becomes inaccurate.

In order to capture the heterogeneous omitted variables, entity fixed-effects, time fixed-effects, and two-way fixed-effects are introduced:

```
# estimation with ENTITY fixed effects
reg.state <- lm(formula= log(Y) ~ X + state - 1, data= Guns)
stargazer(reg.state, type = "text")

# estimation with TIME fixed effects
reg.year <- lm(formula= log(Y) ~ X + year - 1, data= Guns)
stargazer(reg.year, type = "text")

# estimation with ENTITY and TIME fixed effects
reg.state.year <- lm(formula= log(Y) ~ X + state + year - 1, data= Guns)
stargazer(reg.state.year, type = "text")
```

The code produces the following results:

Table 5: State Fixed-Effects Regression Model

	Dependent variable:
	log(Y)
Xlaw	-0.048^{**}
	(0.019)
(Other coefficients)	⋮
stateAlabama	4.081^{***}
	(0.383)
(50 other states)	⋮
R^2	0.999
Adjusted R^2	0.999
Residual Std. Error	0.161 (df = 1115)
F Statistic	$28{,}759.810^{***}$ (df = 58; 1115)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 6: Time Fixed-Effects Regression Model

	Dependent variable:
	log(Y)
Xlaw	-0.330^{***}
	(0.038)
(Other coefficients)	⋮
year1977	4.646^{***}
	(0.637)
(22 other years)	⋮
R^2	0.994
Adjusted R^2	0.994
Residual Std. Error	0.467 (df = 1143)
F Statistic	$6{,}555.501^{***}$ (df = 30; 1143)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 7: Two-way Fixed-Effects Regression Model

	Dependent variable:
	log(Y)
Xlaw	−0.028
	(0.017)
(Other coefficients)	⋮
stateAlabama	4.110***
	(0.451)
(50 other states)	⋮
year1978	0.059**
	(0.028)
(21 other years)	⋮
Observations	1,173
R^2	1.000
Adjusted R^2	0.999
Residual Std. Error	0.140 (df = 1093)
F Statistic	27,471.620*** (df = 80; 1093)
Note:	*p<0.1; **p<0.05; ***p<0.01

Both state and time fixed-effects models have significant coefficient for law proxy but have extremely high R^2 value. This could be a sign of existence of poorly estimated standard errors of regression (SER) coming from violation of Gauss-Markov assumption or autocorrelation (trends over time).

Two-way fixed-effects model has insignificant coefficient for law proxy and it also has extremely high R^2 value.

There is also within-transformed fixed-effects model which is mathematically equivalent to the initial fixed-effects regression model. The latter carries intercepts for N different entities and T different time periods, which is quite cumbersome. Within-transformed model removes theses from the formula.

Since within-transformation model is equivalent to fixed-effects model, the OLS estimators should remain the same. But let's see other features such as R^2 stays the same.

Two different ways to code within transformation of fixed-effects model are proposed using lm function and plm function:

```
# LINEAR within transformed model with ENTITY fixed effects
l.within.state <- lm(formula= log(Y.dem.state) ~ X.dem.state, data= Guns1)
stargazer(l.within.state, type= "text")
# PANEL within transformed model with entity fixed effects
within.state <- plm(formula= log(Y) ~ X, data= Guns, index= c("state","year"),
                    model= "within", effect= "individual")
stargazer(within.state, type = "text")
fixef(within.state)

# LINEAR within transformed model with TIME fixed effects
l.within.time <- lm(formula= log(Y.dem.time) ~ X.dem.time, data= Guns1)
stargazer(l.within.time, type= "text")
# PANEL within transformed model with TIME fixed effects
within.year <- plm(formula= log(Y) ~ X, data= Guns, index= c("state","year"),
                   model= "within", effect= "time")
stargazer(within.year, type = "text")
fixef(within.year)

# LINEAR within transformed model with TWO-WAY fixed effects
ave.Y <- ave(Y)
ave.X <- ave(X)
Y.dem.twoway <- ((Y * ave.Y)/(ave.state.Y * ave.time.Y))
X.dem.twoway <- X - ave.state.X - ave.time.X + ave.X
l.within.twoway<- lm(formula= log(Y.dem.twoway) ~ X.dem.twoway, data= Guns1)
stargazer(l.within.twoway, type= "text")
# PANEL within transformed model with TWO-WAY fixed effects
within.state.year <- plm(formula= log(Y) ~ X, data= Guns, index= c("state","year"),
    model= "within", effect= "twoways")
stargazer(within.state.year, type="text")
fixef(within.state.year)
```

Each couple of linear within transformation model and panel within transformation model displays the same result of regression. The models were well specified.

They produce the following tables:

Table 8: Within Transformed Entity Fixed-Effects Model

	Dependent variable:
	log(Y)
Xlaw	-0.048^{**}
	(0.019)
(Other coefficients)	⋮
Observations	1,173
R^2	0.217
Adjusted R^2	0.177
F Statistic	44.245^{***} (df = 7; 1115)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 9: Within Transformed Time Fixed-Effects Model

	Dependent variable:
	log(Y)
Xlaw	-0.330^{***}
	(0.038)
(Other coefficients)	⋮
Observations	1,173
R^2	0.475
Adjusted R^2	0.462
F Statistic	147.827^{***} (df = 7; 1143)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 10: Within Transformed Two-way Fixed-Effects Model

	Dependent variable:
	log(Y)
Xlaw	−0.028
	(0.017)
(Other coefficients)	⋮
Observations	1,173
R^2	0.056
Adjusted R^2	−0.013
F Statistic	9.217*** (df = 7; 1093)
Note:	*p<0.1; **p<0.05; ***p<0.01

Interpretation

Table 8: Within Transformed Entity Fixed-Effects Model

- First and foremost, the OLS estimator is significant according to its low p-value.
- The coefficient for "law" proxy tells the fact that the shall carry law is in effect in a given year for a given state has a negative impact on violent crime rate per 100,000 people of 4.8% ($100 \times \beta_1$%).
- This regression model explains 21.7% of the total variability of the dependent variable.
- State fixed effects eliminated the risk of a bias due to omitted variables that vary across states **but not over time**.

Table 9: Panel within transformed regression model with time fixed-effects

- First and foremost, the OLS estimator is significant according to its low p-value.
- The fact that the shall carry law is in effect in a given year for a given state has a negative impact on violent crime rate per 100,000 people of 33.0% ($100 \times \beta_1$%).
- This regression model explains 47.5% of the total variability of the dependent variable.
- Time fixed effects eliminated the risk of a bias due to omitted variables that vary across time **but not among states**.

Table 10: Panel within transformed regression model with time fixed-effects

- First and foremost, the OLS estimator is **not significant** according to its high p-value (at least above 10%). The interpretation should be stopped at this stage.

Question 3

Replicate the same results you have just obtained using the package "lfe"

Solution 3

The lfe package is no longer available at CRAN repository. However, it can be manually installed using R tools. But R tools is only available in .exe extension that works on Windows but not on MAC.

Since I am using a MAC notebook, instead of lfe package, I decided to use plm function from plm package and manual computation with lm function.

Question 4

From now on, consider only the specification with the entity fixed-effects.

Write the code to estimate the standard errors under homoskedasticity, those that correct for heteroskedasticity and those clustered at the entity level. Focus again on your variable of interest "law" and comment similarities and differences of the SE across estimators.

Hint: here you should first write the within estimator in matrix form, then estimate the var-cov matrix of the $\hat{\beta}_{OLS}$ in the three cases. Part of the exercise consists in finding the proper expressions of the SE.

Solution 4

Define first entity fixed-effects regression model as fit:

```
# both same entity fixed-effects regression model
lfit <- reg.state    # class lm
pfit <- within.state # class plm panelmodel
```

Generate var-cov matrices under homoskeadsticity, heteroskedasticity, and heteroskedasticity across clusters:

```
vcov.homo    <- sandwich::vcovHC.default(lfit, type = "const") # homoskedasticity
vcov.hetero  <- sandwich::vcovHC.default(lfit, type = "HC0")   # heteroskedasticity
vcov.cluster <- plm::vcovHC.plm(pfit, type=c("HC0"), cluster=c("group")) # clustered
```

Test homoscedastic standard errors, heteroskedasticity-robust standard errors, and clustered standard errors:

```
stargazer(coeftest(lfit, vcov = vcov.homo), type= "text") # SE = 0.018715
stargazer(coeftest(lfit, vcov = vcov.hetero), type= "text") # SE = 0.019168
stargazer(coeftest(pfit, vcov = vcov.cluster), type= "text") # SE = 0.041169
```

The results for "law" proxy are

```
Homoscedastic Standard Errors = 0.018715 # same as SE from summary(reg.state)
Heteroskedasticity-Robust Standard Errors = 0.019168
Cluester-Robust Standard Errors = 0.041169
```

Heteroskedasticity-robust standard errors estimation is useful when the Gauss-Markov assumption is suspected to be violated. It is usually larger than homoscedastic (non-robust) standard errors, which is the case here.

Cluster-robust standard errors account for heteroskedasticity across clusters of observations. Cluster-robust standard errors' estimate is much larger than the non-robust or robust standard errors. There exists heteroskedasticity across clusters.

Solution Code for Problem 1

```
library(stargazer)
# 1. Generate artificial data
## set seed
set.seed(123)
## normal distribution parameters
mu1  <- 50; mu2  <- 100
sd1  <- sqrt(10); sd2 <- sqrt(10)
cov12 <- 5
cor12 <- (cov12/(sd1*sd2))
## bivariate normal distribution parameters
mu   <- c(50,100)
sigma <- matrix(c(sd1^2, sd1*sd2*cor12, sd2*sd1*cor12, sd2^2),2)
## random sample generation
N  <- 1000
XY <- mvrnorm(N,mu,sigma)
X  <- XY[, 1]
Y  <- XY[, 2]
## measurement error
w  <- rnorm(N, mean = 0, sd = 10)
Xt <- X + w
# regression with correct sample
regress1 <- lm(formula = Y ~ X)
stargazer(regress1, type="latex") # to generate the summary table in LaTeX format
beta.01 <- coef(regress1)      # intercept and slope of regress1 stored in beta.01
beta.0 <- as.numeric(beta.01["(Intercept)"])
beta.1 <- as.numeric(beta.01["X"])
# plot regress1
par(mfrow=c(2,2))
plot(X, Y, col = "red", xlab = "X", main = "True Observation", cex=.5)
abline(beta.01, lwd = 2)
# regression with errored sample
regress2 <- lm(formula = Y ~ Xt)
stargazer(regress2, type="latex")
beta.hat.01 <- coef(regress2)
beta.hat.0 <- as.numeric(beta.hat.01["(Intercept)"])
beta.hat.1 <- as.numeric(beta.hat.01["Xt"])
# plot regress2
plot(Xt, Y, col = "red", xlab = "X tilda", main = "Measurement Error", cex=.5)
abline(beta.hat.01, lwd = 2)
# ANOVA test
anova(regress1)
anova(regress2)
# beta.hat probability convergence check
check1 <- ((var(X) + var(w))/var(X)) * beta.hat.1 # sample variances
beta.1 - check1
# beta corrected
beta.co.1 <- ((var(X) + var(w))/var(X)) * beta.hat.1
```

```r
## Monte Carlo simulation
# bias
set.seed(NULL)
fixed.samp <- 100
iter <- c(25,100,500,1000)
B.bias <- rep(0,length(iter))
for(i in 1:length(iter)){
  Bb <- rep(0,iter[i])
  for(j in 1:iter[i]){
    xy <- mvrnorm(fixed.samp,mu,sigma)
    x  <- xy[, 1]
    y  <- xy[, 2]
    wt <- rnorm(fixed.samp, mean = 0, sd = 10)
    xt <- x + wt

    reg.monte <- lm(formula = y ~ xt)
    beta.01.monte <- coef(reg.monte)
    beta.1.monte <- as.numeric(beta.01.monte["xt"])

    Bb[j]  <- beta.1.monte
  }
  mean.Bb <- mean(Bb)
  B.bias[i] <- mean.Bb
}
# accuracy of mean does not improve

# consistency
fixed.iter <- 100
samp <- c(25,100,500,1000)
B.cons <- rep(0,length(samp))
for(i in 1:length(samp)){
  Bc <- rep(0,fixed.iter)
  for(j in 1:fixed.iter){
    xy <- mvrnorm(samp[i],mu,sigma)
    x  <- xy[, 1]
    y  <- xy[, 2]
    wt <- rnorm(samp[i], mean = 0, sd = 10)
    xt <- x + wt

    reg.monte <- lm(formula = y ~ xt)
    beta.01.monte <- coef(reg.monte)
    beta.1.monte <- as.numeric(beta.01.monte["xt"])

    Bc[j]  <- beta.1.monte
  }
  sd.Bc <- sd(Bc)
  B.cons[i] <- sd.Bc
}
# variance decreases
```

Solution Code for Problem 2

```
library(plm)
library(AER)
library(stargazer)
library(dplyr)
library(ggplot2)
library(lmtest)
library(sandwich)

# Produce a table with descriptive statistics of these variables
data(Guns)
stargazer(Guns, type = "text")
summary(Guns)

# declare columns as variables
violent <- Guns$violent
law <- as.numeric(Guns$law)
density <- Guns$density
income <- Guns$income
population <- Guns$population
afam <- Guns$afam
cauc <- Guns$cauc
male <- Guns$male
state <- Guns$state
year <- Guns$year

# build separate matrices for violent (Y) and other variables (X)
Y  <- cbind(violent)
X  <- cbind(law, density, income, population, afam, cauc, male)

# new data.frame created from Guns modifying law's class from factor to numeric
Guns1 <- with(Guns, data.frame(violent, as.numeric(law) - 1, density, income,
    population, afam, cauc, male,
                            as.factor(state), as.factor(year)))
# rename columns
colnames(Guns1)[c(2,9,10)] <- c("law","state","year")

# Time average of Y and X within a state
ave.time.Y <- cbind(ave(Guns1$violent, Guns1$state, FUN = mean))
ave.time.X <- cbind(ave(Guns1$law, Guns1$state), ave(Guns1$density, Guns1$state),
                  ave(Guns1$income, Guns1$state), ave(Guns1$population, Guns1$state),
                  ave(Guns1$afam, Guns1$state), ave(Guns1$cauc, Guns1$state),
                  ave(Guns1$male, Guns1$state))

# Entity average of Y and X within a year
ave.state.Y <- cbind(ave(Guns1$violent, Guns$year, FUN = mean))
ave.state.X <- cbind(ave(Guns1$law, Guns1$year), ave(Guns1$density, Guns1$year),
                    ave(Guns1$income, Guns1$year), ave(Guns1$population, Guns1$year),
```

```
                    ave(Guns1$afam, Guns1$year), ave(Guns1$cauc, Guns1$year),
                    ave(Guns1$male, Guns1$year))

# demeaned Y and X with state index (Y/ave.time.Y because we will log it)
Y.dem.state = Y/ave.time.Y
X.dem.state = X - ave.time.X
# demeaned Y and X with time index (Y/ave.state.Y because we will log it)
Y.dem.time = Y/ave.state.Y
X.dem.time = X - ave.state.X

# estimation WITHOUT fixed effects
reg <- lm(formula= log(Y) ~ X, data= Guns)
stargazer(reg, type = "text")
# PANEL linear model without fixed effects
regp <- plm(formula= log(Y) ~ X, data= Guns, index= c("state","year"), model=
    "pooling", effect= "individual")
stargazer(regp, type = "text")

# estimation with ENTITY fixed effects
reg.state <- lm(formula= log(Y) ~ X + state - 1, data= Guns)
stargazer(reg.state, type = "text")

# estimation with TIME fixed effects
reg.year <- lm(formula= log(Y) ~ X + year - 1, data= Guns)
stargazer(reg.year, type = "text")

# estimation with ENTITY and TIME fixed effects
reg.state.year <- lm(formula= log(Y) ~ X + state + year - 1, data= Guns)
stargazer(reg.state.year, type = "text")

# LINEAR within transformed model with ENTITY fixed effects
l.within.state <- lm(formula= log(Y.dem.state) ~ X.dem.state, data= Guns1)
stargazer(l.within.state, type= "text")
# PANEL within transformed model with ENTITY fixed effects
within.state <- plm(formula= log(Y) ~ X, data= Guns, index= c("state","year"),
                    model= "within", effect= "individual")
stargazer(within.state, type = "text")
fixef(within.state)

# LINEAR within transformed model with TIME fixed effects
l.within.time <- lm(formula= log(Y.dem.time) ~ X.dem.time, data= Guns1)
stargazer(l.within.time, type= "text")
# PANEL within transformed model with TIME fixed effects
within.year <- plm(formula= log(Y) ~ X, data= Guns, index= c("state","year"),
                   model= "within", effect= "time")
stargazer(within.year, type = "text")
fixef(within.year)

# LINEAR within transformed model with TWO-WAY fixed effects
ave.Y <- ave(Y)
```

```
ave.X <- ave(X)
Y.dem.twoway <- ((Y * ave.Y)/(ave.state.Y * ave.time.Y))
X.dem.twoway <- X - ave.state.X - ave.time.X + ave.X
l.within.twoway<- lm(formula= log(Y.dem.twoway) ~ X.dem.twoway, data= Guns1)
stargazer(l.within.twoway, type= "text")
# PANEL within transformed model with TWO-WAY fixed effects
within.state.year <- plm(formula= log(Y) ~ X, data= Guns, index= c("state","year"),
    model= "within", effect= "twoways")
stargazer(within.state.year, type="text")
fixef(within.state.year)

# SE under homo, hetero, and hetero-cluster (for only ENTITY fixed effects)
lfit  <- reg.state
pfit  <- within.state

# variance-covariance matrices
vcov.homo    <- sandwich::vcovHC.default(lfit, type = "const") # homoskedasticity
vcov.hetero  <- sandwich::vcovHC.default(lfit, type = "HC0") # heteroskedasticity
vcov.cluster <- plm::vcovHC.plm(pfit, type = c("HC0"), cluster = c("group")) #
    clustered
# SE test
stargazer(coeftest(lfit, vcov = vcov.homo), type= "text") # SE = 0.018715 # compare
    with summary(reg.state)
stargazer(coeftest(lfit, vcov = vcov.hetero), type= "text") # SE = 0.019168
stargazer(coeftest(pfit, vcov = vcov.cluster), type= "text") # SE = 0.041169
```