section3_PDM

Eline van Groningen, Paola Priante, Valery Maasdamme, Yuhu Wang

9/25/2020

```
# Load libraries
library(tidyverse)
library(stargazer)
library(wbstats)
library(ggplot2)
library(plyr)
library(plm)

# Load world bank data
dfExport <- wb_data(indicator=c("IC.EXP.TMBC",       # Time to export
                                "NY.GDP.PCAP.CD",    # GDP per capita
                                "TG.VAL.TOTL.GD.ZS", # Merchandise trad
e % GDP
                                "NE.EXP.GNFS.ZS",    # Exports of goods
 and services (% of GDP)
                                "IC.EXP.CSDC.CD"),   # Cost to export
                                country = "countries_only",
                                start_date = 2014,
                                end_date = 2019)

# Rename column names
colnames(dfExport)[colnames(dfExport) == "date"]              <- "Year"
colnames(dfExport)[colnames(dfExport) == "country"]           <- "Count
ry"
colnames(dfExport)[colnames(dfExport) == "date"]              <- "Year"
colnames(dfExport)[colnames(dfExport) == "IC.EXP.TMBC"]       <- "TimeE
xport"
colnames(dfExport)[colnames(dfExport) == "NY.GDP.PCAP.CD"]    <- "GDPPe
rCap"
colnames(dfExport)[colnames(dfExport) == "TG.VAL.TOTL.GD.ZS"] <- "Merch
andiseGDP"
colnames(dfExport)[colnames(dfExport) == "NE.EXP.GNFS.ZS"]    <- "Expor
tGoodsServices"
colnames(dfExport)[colnames(dfExport) == "IC.EXP.CSDC.CD"]    <- "CostE
xport"

# Subset complete observations, and implement an admittedly arbitrary
# observation period
dfExport.sub <- dfExport[complete.cases(dfExport),]

# Generate list with all countries with complete observations
```

```
complete <- dfExport.sub %>%
  dplyr::count(Country) %>%
  filter(n == 6)
completeCountry <- as.vector(complete$Country)

# Generate data frame only containing countries with complete observati
ons
dfExport.sub.cmplt <- dfExport.sub %>%
  filter(Country %in% completeCountry)

# Convert to data frame
dfExport.sub.cmplt <- as.data.frame(dfExport.sub.cmplt)

# Generate table with summary statistics
stargazer(dfExport.sub.cmplt)

# Plot Cost Export

subCountries <- c("Australia", "Bolivia", "Brazil", "Portugal", "Thaila
nd",
                  "Zimbabwe", "Bangladesh", "Bulgaria", "China", "Denma
rk",
                  "France", "Finland", "India")

dfExport.sub.cmplt <-
  dfExport.sub.cmplt[dfExport.sub.cmplt$Country %in% subCountries,]

ggplot(dfExport.sub.cmplt, aes(x=CostExport, y=TimeExport))+
  #add the annual outcomes coloured by Country
  geom_point(aes(color=Country), size=1)+
  #add regression lines for the countries
  geom_smooth(method="lm", se=FALSE, colour="dark grey")+
  #label the axis
  xlim(0, 300) + ylim(0, 70)+
  xlab("Cost of Export")+
  ylab("Time of Export")+
  theme(axis.title= element_text(size=rel(1)),
        axis.text= element_text(size=rel(1)))+
  guides(colour = guide_legend(override.aes = list(size=1)))
```

Preparing data for regression

```
# Determine country averages of the included variables, as well as the
number of
# non missing observations during the selected observation period
dfExport.sub.cmplt.avg <-
  ddply(dfExport.sub.cmplt, .(Country), summarise,
        avg.TimeExport   = mean(TimeExport, na.rm=TRUE),
```

```
        avg.GDPPerCap    = mean(GDPPerCap, na.rm=TRUE),
        avg.CostExport   = mean(CostExport, na.rm=TRUE),
        avg.ExportGoodsServices     = mean(ExportGoodsServices, na.rm=T
RUE),
        avg.MerchandiseGDP   = mean(MerchandiseGDP, na.rm=TRUE),
        numValid         = length(Country))

# Merge averages in dfWorld.avg with dfWorld.sub (this can be done with

# 'mutate', but then the concise data frame with country average will n
ot be
# made available
dfExport.sub.cmplt <- merge(dfExport.sub.cmplt, dfExport.sub.cmplt.avg,

                            by="Country")

attach(dfExport.sub.cmplt)
dfExport.sub.cmplt$diff.TimeExport   <- TimeExport   - avg.TimeExport
dfExport.sub.cmplt$diff.GDPPerCap    <- GDPPerCap    - avg.GDPPerCap
dfExport.sub.cmplt$diff.CostExport   <- CostExport   - avg.CostExport
dfExport.sub.cmplt$diff.ExportGoodsServices <- ExportGoodsServices  -
  avg.ExportGoodsServices
dfExport.sub.cmplt$diff.MerchandiseGDP   <- MerchandiseGDP   -
  avg.MerchandiseGDP
detach(dfExport.sub.cmplt)
```

Pooled Regression

```
#Formulate the model (very ad hoc)
mdlA <- TimeExport ~ GDPPerCap + CostExport + ExportGoodsServices +
  MerchandiseGDP

#Make between and within group data frames

#For convenience two datasets are made that contain the model
#variables for the within group differences and the between
#group difference

# find the variable of interest
mdlvars <- all.vars(mdlA)
mdlvars.avg <- paste0("avg.", mdlvars)
mdlvars.diff <- paste0("diff.", mdlvars)

# Select variables from the data frames
dfExport.between <- dfExport.sub.cmplt.avg[mdlvars.avg]
dfExport.within <-dfExport.sub.cmplt[mdlvars.diff]

# Rename column names in order to make use  of the same model specifica
tion
```

```
# mdlA, and to conveniently merge the regression objects in stargazer

colnames(dfExport.within) <-
  gsub("diff\\.", "", colnames(dfExport.within))
colnames(dfExport.between) <-
  gsub("avg\\.", "", colnames(dfExport.between))

## Estimation of the pooled model
rsltPool <- lm(mdlA, data= dfExport.sub.cmplt)
summary(rsltPool)
stargazer::stargazer(rsltPool, align=TRUE, no.space=TRUE,
                     intercept.bottom=FALSE, type="text")
```

Between regression

```
rsltwithin <- lm(mdlA, data= dfExport.within)
summary(rsltwithin)
rsltBetween <- lm (mdlA, data= dfExport.between)
summary(rsltBetween)

stargazer::stargazer(rsltPool, rsltBetween, aling=TRUE, no.space=TRUE,
                     intercept.bottom= FALSE, type= "text")
```

Fixed Effect Regression

```
rsltFE.Country <- plm(mdlA, data= dfExport.sub.cmplt,
                      index= c("Country", "Year"), model="within")
#Tabulate the results
summary(rsltFE.Country)
stargazer::stargazer(rsltPool, rsltFE.Country, align=TRUE, no.space=TRU
E,
                     intercept.bottom=FALSE, type="text")
#Explore the estimated intercepts
summary(fixef(rsltFE.Country, type="dmean"))
```

Random Effect Regression

```
#Estimate random effect model ('random')
rsltRE.Country <- plm(mdlA, data=dfExport.sub.cmplt,
                      index=c("Country", "Year"), model= "random")

#Tabulate the results
summary(rsltRE.Country)
stargazer::stargazer(rsltPool, rsltFE.Country, rsltRE.Country,
                     align=TRUE, no.space=TRUE, intercept.bottom=FALSE,

                     type="text")

# Evaluate the fixed effects model versus the pooled regression model
# Last minute of tutorial #4 Panel Data
# An insignificant tests tells that all models are consistent
```

```r
# A significant tests rejects the hypothesis in favor of the fix effect
s model
pFtest(rsltFE.Country, rsltPool)

# How do we now when to use fixed and when to use random?
# Hausman test: compare random and fixed effects models
# Under H0, no correlation between disturbance and explanatory variable
s,
# both RE and FE are consistent (though FE is not efficient), under H1,

# correlation between disturbance, only FE consistent
# Last two minutes of tutorial #5 Panel Data
phtest(rsltFE.Country, rsltRE.Country)
```