PRODUCTION FUNCTION ESTIMATION

PAUL SCHRIMPF

Problem 1: http://tryr.codeschool.com/ is an interactive introduction to R. Please work through it if you have not used R before. If you're already familiar with R, then you can skip this.

1. Explore the data

We will begin our analysis with some exploratory statistics and figures. There are at least two reasons for this. First, we want to check for any anomalies in the data, which may indicate an error in our code, our understanding of the data, or the data itself. Second, we should try to see if there are any striking patterns in the data that deserve extra attention.

The script downloadData.R downloads and loads into R the data from Rankin, Sderbom, and Teal (2006). Run this script by entering

```
source("downloadData.R")
```

You could enter this directly into R's command line, but it is always better to write your commands in a script and run the script. You are bound to make mistakes, and it will be easier to fix them if you save your commands in a file instead of entering them one by one.

Your workspace should now contain a data frame named "df." Data frames are how R stores data. A data frame is basically a matrix where the columns represent variables and the rows are observations. The variables/columns have names. To list the names of a data frame, run

```
names(df)
```

The meanings of these variables are listed in Table 1.

1.1. Descriptive statistics.

Show some summary statistics with

```
summary(df)
```

The builtin summary command is easy to use, but it does not quite provide all the information that we might want. For example, it does not show the standard deviation of each variable. We can calculate the standard deviation of a single variable with

```
sd(df$output, na.rm=TRUE)
```

or, we could calculate the standard deviation of all variables using the apply command,

```
apply(df, 2, FUN=function(x) { sd(x, na.rm=TRUE) })
```

Descriptive statistics are a very common thing to want to calculate, and not everyone will be statisfied by the summary command. As a result, many people have created alternate commands for calculating descriptive statistics. These commands come in R packages, which are very easy to install. Install the Hmisc package

```
install.packages("Hmisc")
```

You only need to install a package once (or at least once per update), but before you use commands from a package you have to load it

```
library(Hmisc)
```

Hmisc includes the describe command, which produces more descriptive statistics.

Date: Due: October 6th, 2014.

TABLE 1. Variable definitions

Variable	Definition
firm	Firm identifier
any.foreign.own	Indicator of foreign ownership
year	Year
firm.age	Firm age in years
exports	Indicator for whether the firm exports
exportna	Indicator for exporting outside of Africa
exporta	Indicator for exporting within Africa
rmawagus	
eduwgt	Weighted average education of employees
agewgt	Weighted average age of employees
tenwgt	Weighted average tenure of employees
ernus	
country	Country of firm
prateus	
kus_routus	
entex	Indicator for entry into exporting
exitex	Indicator for exit from exporting
industry	Industry of firm
labor	Labor input
output	Output (in 1991 US$)
capital	Capital (in 1991 US$)
materials	Material (in 1991 US$)
indirect.costs	(in 1991 US$)

I am unsure of the meaning of variables whose definitions are blank. I may have overlooked some documentation from http://www.csae.ox.ac.uk/datasets/cfld/cfld-main.html or from Rankin, Sderbom, and Teal (2006).

```
describe(df)
```

1.2. **Descriptive plots.** Let's make some plots of the data. Here's a histogram of log output

```
hist(log(df$output))
```

Here's a scatter plot of log labor and log output

```
plot(x=log(df$labor), y=log(df$output))
```

The builtin R plotting commands are convenient, but the ggplot2 package can create nicer looking figures. Creating nicer figures is not without a cost; the syntax for ggplot2 is far more verbose than the builtin plotting commands.

```
load.fun(ggplot2) # this is a function in downloadData.R that will
                  # install.packages("ggplot2") if needed, otherise it
                  # just call library(ggplot2).
## histograms by country and industry
output.histogram <- ggplot(df, aes(x=log(output),fill=country)) +
  geom_histogram() +
  facet_grid(country ~ industry) + theme_minimal() +
  scale_fill_brewer(type="qual",guide=FALSE)
output.histogram

## scatter plots by country and industry
scatter.capital <- ggplot(df, aes(x=log(capital), y=log(output),
                          color=country)) +
  geom_point() +
  facet_grid(country ~ industry) + theme_bw() +
  scale_color_brewer(type="qual",guide=FALSE)
scatter.capital
```

Problem 2: Create scatter plots of log output and log labor, log output and log materials, and log output and log indirect costs.

(1) Are there any strange patterns or other obvious problems with the data?
(2) Does it appear that Cobb-Douglas will be a good approximation for these firms' production functions? Why or why not?

We can also create 3-d scatter plots. These are less useful to include in papers, but can be fun to look at.

```
load.fun(rgl) ## install & load the rgl package
open3d()
rgl.spheres(log(df$labor),log(df$capital), log(df$output),color="grey",radius
    =0.05,
            specular="white")  ## scatter plot of spheres
axes3d(c(’x’,’y’,’z’)) ##
title3d(’’,’’,’log labor’,’log capital’,’log output’) ## labels axes
grid3d(c("x", "y", "z"))
par3d(windowRect = c(100,100, 1280+100, 960+100))  ## resizes window
#play3d(spin3d(axis=c(0,0,1),rpm=3)) ## make it spin
```

2. Production function estimation

In this section, we will estimate production functions using various methods. I am not certain what the data means by "indirect costs." With no information, it seems reasonable to treat them as another flexible input; or simply subtract them from output; or add them to materials. You can choose to do whatever you wish.

```
df$Materials <- df$materials + df$indirect.costs
```

2.1. OLS. Estimate the production function using OLS with the lm() command:

```
summary(lm(log(output) ~ log(capital) + log(labor) + log(Materials), data=df))
```

lm() treats observations as independent and assumes homoskedasticity when calculating standard errors. Let's calculate heteroskedasticity robust standard errors clustered on firm. For this, we will use the plm package and some code in clusterFunctions.R

```
load.fun(plm) ## install & load the plm package
source("clusterFunctions.R") ## load function for clustered S.E.s
prod.ols <- plm(log(output) ~ log(capital) + log(labor) + log(Materials),
                data=df,model="pooling", index=c("firm","year")) # estimates
                    model
cl.plm(df, prod.ols, df$firm) # calculates standard errors
```

Problem 3: What are the estimated returns to scale?

Let's see whether it makes a difference if we add controls for year and industry. The way R lets you specify formulas for models is very flexible and makes this quite easy. As you hopefully know,

```
lm(y ~ x + z)
```

simply regresses y on x and z if x and z are numeric. If x or z is a "factor" (R's name for categorical or discrete variables), then R will regress y on a full set of dummies for x and/or z.

```
lm(y ~ x*z)
```

will regress y on a full set of interacted dummies for x and z. Thus, if we want to include industry and time dummies, we could run

```
plm(log(output) ~ log(capital) + log(labor) +
    log(Materials) + industry + as.factor(year),
    data=df,model="pooling", index=c("firm","year"))
```

If we want to let the input coefficients vary by industry, we could run

```
plm(log(output) ~ (log(capital) + log(labor) +
    log(Materials))*industry + as.factor(year),
    data=df,model="pooling", index=c("firm","year"))
```

Problem 4:

(1) Do year and industry effects appear to be important?
(2) Do the capital and labor coefficients appear to vary by industry?
(3) Add country dummies to the model. Do country effects appear important?

2.2. **Fixed effects.** The plm package can also estimate fixed effects models. To do so, run

```
prod.fe <- plm(fmla, data=df, model="within", index=c("firm","year"))
cl.plm(df, prod.fe, df$firm)
```

where fmla is the formula for whatever specification you prefer (i.e. with or without year, industry, and country controls).

Problem 5: How do the fixed effects estimates compare to the OLS estimates?

2.3. **Olley-Pakes.** In this section, we will estimate the production function using the control function approach of Olley and Pakes (1996). First, we must create an investment variable. We will just call the change in capital investment.

```
df <- df[order(df$firm, df$year), ] # sort data
df$invest <- c(df$capital[2:nrow(df)] - df$capital[1:(nrow(df)-1)],NA)
df$invest[c(df$firm[2:nrow(df)]!=df$firm[1:(nrow(df)-1)],FALSE)] <- NA
```

The first step of Olley-Pakes is to estimate

$$\begin{aligned} y_{it} =& \beta_k k_{it} + \beta_l l_{it} + \beta_m m_{it} + I_t^{-1}(k_{it}, I_{it}) + \epsilon_{it} \\ =& \beta_l l_{it} + \beta_m m_{it} + f_t(k_{it}, i_{it}) + \epsilon_{it}, \end{aligned}$$

where to keep things relatively simple, I have omitted controls for industry and country. You may include them if you want.

We will approximate f_t by a polynomial capital and investment. The poly() function in R makes this very easy, but poly does not like missing values, so we need to work with the subset of our data where nothing is missing.

```
notmissing <- !is.na(df$capital) & !is.na(df$invest)
df.nm <- subset(df,notmissing)
df.nm <- df.nm[order(df.nm$firm, df.nm$year),]
```

Now to estimate the first step,

```
op1 <- lm(log(output) ~ log(labor) + log(Materials) +
          poly(cbind(log(capital),invest),degree=4)*factor(year),
          data=df.nm)
```

This first step gives us consistent estimates of β_m and β_l.

Problem 6: How do the Olley-Pakes estimates of β_m and β_l compare to the OLS and fixed effects estimate? Is the difference what you would expect?

The second step of Olley-Pakes is to estimate β_k by using nonlinear least-squares on:

$$y_{it} - \hat{\beta}_l l_{it} - \hat{\beta}_m m_{it} = \beta_k k_{it} + g\left(\hat{f}_{it-1} - \beta_k k_{it-1}\right) + \xi_{it} + \epsilon_{it}$$

That is, we want to solve

$$\min_{g,\beta_k} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(y_{it} - \hat{\beta}_l l_{it} - \hat{\beta}_m m_{it} - \beta_k k_{it} - g\left(\hat{f}_{it-1} - \beta_k k_{it-1}\right)\right)^2$$

A useful observation is that the solution to this minimization problem is the same whether we minimize simultaneously or first minimize by choice of g given β_k, and then minimize over β_k. That is, we can instead solve

$$\min_{\beta_k} \min_{g} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(y_{it} - \hat{\beta}_l l_{it} - \hat{\beta}_m m_{it} - \beta_k k_{it} - g\left(\hat{f}_{it-1} - \beta_k k_{it-1}\right)\right)^2$$

For a fixed β_k, the inner minimization is just a non-parametric regression of $y_{it} - \hat{\beta}_l l_{it} - \hat{\beta}_m m_{it} - \beta_k k_{it}$ on $\hat{f}_{it-1} - \beta_k k_{it-1}$. Thus, we can solve the inner minimization problem by running OLS of $y_{it} - \hat{\beta}_l l_{it} - \hat{\beta}_m m_{it} - \beta_k k_{it}$ on a polynomial of $\hat{f}_{it-1} - \beta_k k_{it-1}$. We then, just have to search for a minimal sum of residuals as we vary a single parameter, β_k. The following code implements this idea and creates an objective function for the minimization over β_k.

```
# calculate fhat
b1 <-op1$coefficients[c("log(labor)", "log(Materials)")]
xb1 <- log(as.matrix(df.nm[,c("labor", "Materials")])) %*% b1
fhat <- predict(op1,df.nm) - xb1

# function to lag fhat and capital
lag <- function(x, i=df.nm$firm, t=df.nm$year) {
  if (length(i) != length(x) || length(i) != length(t) ) {
    stop("Inputs not same length")
  }
  x.lag <- x[1:(length(x)-1)]
  x.lag[i[1:(length(i)-1)]!=i[2:length(i)] ] <- NA
  x.lag[t[1:(length(i)-1)]+1!=t[2:length(i)] ] <- NA
  return(c(NA,x.lag))
}
# create data frame for step 2 regression
df.step2 <- data.frame(lhs=(log(df.nm$output)-xb1),
                       k=log(df.nm$capital),fhat=fhat,
                       k.lag=log(lag(df.nm$capital)),
                       f.lag=lag(fhat))
# drop missing observations because they mess up poly()
df.step2 <- subset(df.step2, !apply(df.step2, 1, function(x)
                                    any(is.na(x))))
# objective function = sum of residuals^2
objective <- function(betaK, degree=4) {
  op2 <- lm(I(lhs - betaK*k) ~ poly(I(f.lag - betaK*k.lag),degree),
            data=df.step2)
  return(sum(residuals(op2)^2))
}
```

To estimate β_k we need to minimize objective with respect to β_k. Before trying to minimize a function, it is a good idea to plot it to see whether it is well-behaved. Here is how you could plot it.

```
fig.df <- data.frame(bk=seq(from=-0.02,to=0.3, by=0.005))
fig.df$obj <- sapply(fig.df$bk, objective)
ggplot(data=fig.df,aes(x=bk,y=obj)) + geom_point()
```

To minimize, we can use the optim command

```
opt.out <- optim(prod.ols$coefficients["log(capital)"],
                 fn=objective,method="Brent",lower=-1,upper=1)
betaK <- opt.out$par
```

Problem 7: How does the Olley-Pakes estimate of the capital coefficient compare to the OLS and fixed effects estimates?

Problem 8: The code above treats the labor as flexibly chosen each period. What would need to change if labor is inflexible and must be chosen a period in advance? Re-estimate the production function treating labor as inflexible. To do this you can modify the code above, or you can use the function production.cf.2step from productionEstimation.R. production.cf.2step does exactly what we did above, but wraps it into a nicer interface more similar to the built-in R function lm. It is somewhat painful to write such code; dealing with formulas and data.frames is especially tedious. However, having such code makes it very easy to change your specification. See the comments in productionEstimation.R for details on how to use it.

Problem 9: Levinsohn and Petrin (2003) propose using the firm's choice of materials, instead of investment, to form the control function. Either estimate the production function using materials to form the control function and report the results; or explain why such a procedure is a bad idea. If you choose to estimate using materials for the control function, how do the results differ from when using investment?

References

Levinsohn, James and Amil Petrin. 2003. "Estimating Production Functions Using Inputs to Control for Unobservables." *The Review of Economic Studies* 70 (2):pp. 317–341. URL http://www.jstor.org/stable/3648636.

Olley, G.S. and A. Pakes. 1996. "The dynamics of productivity in the telecommunications equipment industry." *Econometrica* 64 (6):1263–1297.

Rankin, Neil, Mns Sderbom, and Francis Teal. 2006. "Exporting from Manufacturing Firms in Sub-Saharan Africa." *Journal of African Economies* 15 (4):671–687. URL http://jae.oxfordjournals.org/content/15/4/671.abstract.