Empirical Analysis of the Role of Energy in Economic Growth

Caleb Reese[a], Lucas Timmer[a], Matthew Kuperus Heun[a,*]

[a] *Engineering Department, Calvin College, Grand Rapids, MI 49546, USA*

Abstract

*********** Add abstract ***********

Keywords: economic growth, energy, cobb-douglas, CES, LINEX

```
#################################
# This function loads data given a country name.
#
# countryName = the name of the country for which data is to be loaded. The file r
#   named <countryName>Data.txt. The file is assumed to exist in a subfolder of th
#
# returns a matrix with the data that has been loaded
##
loadData <- function(countryName){
  # Read the data file as a table with a header.
  fileName <- paste("data/", countryName, "Data.txt", sep="")
  dataTable <- read.table(fileName, header = TRUE)
  # Identifies the header names associated with dataTable
  names(dataTable)
  return(dataTable)
}
```

*Corresponding author

Email address: mkh2@calvin.edu, tel: +1 (616) 526-6663, fax: +1 (616) 526-6501 (Matthew Kuperus Heun)

1. **Cobb-Douglas Without Energy**

```
cobbDouglas <- function(countryName){

  # Load the data that we need.
  dataTable <- loadData(countryName)

  # Establish guess values for alpha and lambda.
  lambdaGuess <- 0.0 # guessing lambda = 0 means there is no technological progres
  alphaGuess <- 0.3 # a typical value for alpha, the coefficient on capital stock

  # Runs a non-linear least squares fit to the data. We've replaced beta with 1-al
  modelCD <- nls(iGDP ~ exp(lambda*iYear) * iCapStk^alpha * iLabor^(1 - alpha),
                 start=(list(lambda=lambdaGuess,alpha=alphaGuess)),
                 data=dataTable)

  # Checks validity of the model. AIC stands for Akaike's Information Criterion.
  aicCD  <- AIC(modelCD, k=2); aicCD

  summaryCD <- summary(modelCD) # Gives the nls summary table.
  print(summaryCD)
  ciCD <- confint(modelCD, level = 0.95); ciCD # Displays confidence intervals for

  # Calculate beta and its confidence interval and report it.
  alpha <- as.numeric(coef(modelCD)["alpha"])
  beta <- 1.0 - alpha
  beta.est <- deltaMethod(modelCD, "1 - alpha"); beta.est # Estimates beta and its
  # Now calculate a confidence interval on beta
  dofCD <- summaryCD$df[2]; dofCD # Gives the degrees of freedom for the model.
  tvalCD <- qt(0.975, df = dofCD); tvalCD
  betaCICD <- with(beta.est, Estimate + c(-1.0, 1.0) * tvalCD * SE); betaCICD # Gi

  coef(modelCD)

  # Combine all estimates and their confidence intervals into data frames with int
  estCD <- data.frame(lambda = coef(modelCD)["lambda"], alpha = coef(modelCD)["alp
  row.names(estCD) <- "Cobb-Douglas"
```

```
  #row.names(estCD) <- "Cobb-Douglas: $y = e^{\\lambda t}k^{\\alpha}l^{\\beta}$"
  # The [1] subscripts pick off the lower confidence interval
  lowerCD <- data.frame(lambda = ciCD["lambda","2.5%"], alpha = ciCD["alpha", "2.5
  row.names(lowerCD) <- "- 95% CI"
  # The [2] subscripts pick off the lower confidence interval
  upperCD <- data.frame(lambda = ciCD["lambda","97.5%"], alpha = ciCD["alpha", "97
  row.names(upperCD) <- "+ 95% CI"

  # Now create the data for a table.
  dataCD <- rbind(upperCD, estCD, lowerCD); dataCD
  colnames(dataCD) <- c("$\\lambda$", "$\\alpha$", "$\\beta$", "$\\gamma$")
  tableCD <- xtable(dataCD, caption=paste(countryName, " Cobb-Douglas, 1980-2011",
  return(tableCD)
}

tableCD <- cobbDouglas("US")


Formula: iGDP ~ exp(lambda * iYear) * iCapStk^alpha * iLabor^(1 - alpha)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
lambda 0.010256   0.000682   15.03  1.7e-15
alpha  0.270030   0.028311    9.54  1.4e-10

Residual standard error: 0.0178 on 30 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 3.81e-07

Waiting for profiling to be done...


countries <- c("US", "CN")
tablesCD <- apply(countries, 1, cobbDouglas)

Error:  dim(X) must have a positive length
```

```r
print(tableCD, caption.placement="top")
```

Table 1: US Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0116	0.33	0.79	
Cobb-Douglas	0.0103	0.27	0.73	0.0
- 95% CI	0.0089	0.21	0.67	

```r
# According to http://cran.r-project.org/web/packages/xtable/vignettes/xtableGalle
# be able to use the "sanitize.text.function" parameter to allow markup in column
# line is not working at the present time. --MKH, 18 Jan 2012.
# print(tableCD, caption.placement="top", sanitize.text.function = function(x){x})
```

2. Cobb-Douglas With Energy

We can force α, β, and γ to be in $[0, 1]$ by a reparameterization:

$$a \in [0, 1], b \in [0, 1], \alpha = \min(a, b), \beta = |b - a|, \gamma = 1 - \max(a, b)$$

```r
###############################
# This function fits the Cobb Douglas production function including energy to hist
# The functional form is
#
# iGDP = exp(lambda * itime) * iCapStk^alpha * iLabor^beta * iEnergy^gamma
#
# iYear: time indexed to 0.0 at the first year [years since beginning of data set]
# iGDP = time series GDP data indexed to 1.0 at the first year [-]
# iCapStk = time series capital stock data indexed to 1.0 at the first year [-]
# iLabor = time series labor data indexed to 1.0 at the first year [-]
# iEnergy = time series energy data indexed to 1.0 at the first year [-].
#      This may be any of heat (iQ), exergy (iX), or useful work (iU).
#
# returns a data object containing three rows and 5 columns.
```

```
#   col 1: lambda
#   col 2: alpha
#   col 3: beta
#   col 4: gamma
#   row 1: -95% CI for each parameter
#   row 2: estimate for each parameter
#   row 3: +95% CI for each parameter
#
# countryName is a string containint the 2-letter abbreviation for the country, e.
# energyColumnName is the name of the column to be used for energy.
#   This may be any of heat ("iQ"), exergy ("iX"), or useful work ("iU").
# energyString is the string to be used in table captions reprsenting the type of
##
cobbDouglasEnergy <- function(countryName, energyColumnName, energyString){

  # Load the data that we need.
  dataTable <- loadData(countryName)

  # Reparameterize to ensure that we meet the constraints:
  # * alpha + beta + gamma = 1.0.
  # * alpha, beta, and gamma are all between 0.0 and 1.0.
  # To do this, we reparameterize as
  # * 0 < a < 1
  # * 0 < b < 1
  # * alpha = min(a, b)
  # * beta = b - a
  # * gamma = 1 - max(a, b)
  iEnergy <- dataTable[energyColumnName] #grabs the desired energy column
  lambdaGuess <- 0.0 # guessing lambda = 0 means there is no technological progres
  alphaGuess <- 0.3 # a typical value for alpha
  betaGuess <- 0.7 # a typical value for beta
  modelCDe <- nls(iGDP ~ exp(lambda*iYear) *
                    iCapStk^min(a,b) * iLabor^abs(b-a) *
                    iEnergy^(1.0 - max(a,b)),
                  algorithm = "port",
                  start = list(lambda=lambdaGuess, a=alphaGuess, b=alphaGuess+beta
                  lower = list(lambda=-Inf, a=0, b=0),
```

```
                upper = list(lambda=Inf, a=1, b=1),
                data = dataTable)
aicCDe <- AIC(modelCDe, k=2) # Checks validity of the model. AIC stands for Akai
print(aicCDe)

summaryCDe <- summary(modelCDe) # Gives the nls summary table
print(summaryCDe)

# Provides confidence intervals on lambda, a, and b. But, we need CIs on alpha a
ciCDe <- confint(modelCDe, level = 0.95)
print(ciCDe)

a <- as.numeric(coef(modelCDe)["a"])
b <- as.numeric(coef(modelCDe)["b"])
lambda <- as.numeric(coef(modelCDe)["lambda"])
alpha <- a
beta <- b - a
gamma <- 1.0 - alpha - beta

# Report results with SE
beta.est <- deltaMethod(modelCDe, "b-a") # Reports results for beta, because bet
gamma.est <- deltaMethod(modelCDe, "1-b") # Reports results for gamma, because g

# Now calculate confidence intervals.
dofCDe <- summaryCDe$df[2] # Gives the degrees of freedom for the model.
print(dofCDe)
tvalCDe <- qt(0.975, df = dofCDe)
print(tvalCDe)
betaCICDe <- with(beta.est, Estimate + c(-1.0, 1.0) * tvalCDe * SE) # Gives the
print(betaCICDe)
gammaCICDe <- with(gamma.est, Estimate + c(-1.0, 1.0) * tvalCDe * SE) # Gives th
print(gammaCICDe)

# Combine all estimates and their confidence intervals into data frames with int
estCDe <- data.frame(lambda = lambda, alpha = alpha, beta = beta, gamma = gamma)
print(estCDe);
#row.names(estCDe) <- "Cobb-Douglas with e: $y = e^{\\lambda t}k^{\\alpha}l^{\\b
```

```
  row.names(estCDe) <- paste("C-D with ", energyString, sep="")
  # The [1] subscripts pick off the lower confidence interval
  lowerCDe <- data.frame(lambda = ciCDe["lambda","2.5%"], alpha = ciCDe["a", "2.5%
  row.names(lowerCDe) <- "- 95% CI"
  # The [2] subscripts pick off the lower confidence interval
  upperCDe <- data.frame(lambda = ciCDe["lambda","97.5%"], alpha = ciCDe["a", "97.
  row.names(upperCDe) <- "+ 95% CI"

  # Now create the data for a table.
  dataCDe <- rbind(upperCDe, estCDe, lowerCDe)
  print(dataCDe)

  return(dataCDe)
}
```

2.1. Cobb-Douglas with Q

```
countryName <- "US"
energyColumn <- "iQ"; energyString <- "q"
dataCDq <- cobbDouglasEnergy(countryName, energyColumn, energyString)

[1] -161.1

Formula: iGDP ~ exp(lambda * iYear) * iCapStk^min(a, b) * iLabor^abs(b -
    a) * iEnergy^(1 - max(a, b))

Parameters:
       Estimate Std. Error t value Pr(>|t|)
lambda  0.01049    0.00108    9.67  1.4e-10
a       0.26322    0.03795    6.94  1.3e-07
b       0.97890    0.07647   12.80  1.9e-13

Residual standard error: 0.0181 on 29 degrees of freedom

Algorithm "port", convergence message: relative convergence (4)

Waiting for profiling to be done...
```

```
          2.5%  97.5%
lambda 0.00885 0.0127
a      0.18580 0.3283
b      0.82175     NA
[1] 29
[1] 2.045
[1] 0.5947 0.8367
[1] -0.1353  0.1775
   lambda  alpha   beta  gamma
1 0.01049 0.2632 0.7157 0.0211
            lambda  alpha   beta   gamma
+ 95% CI   0.01270 0.3283 0.8367  0.1775
C-D with q 0.01049 0.2632 0.7157  0.0211
- 95% CI   0.00885 0.1858 0.5947 -0.1353

# TODO Develop a better caption for the table based on the data automatically.
tableCDq <- xtable(dataCDq, caption=paste(countryName, ", 1980-2011.", sep=""), di
#estCDe <- dataCDe[2]
#dataAll <- rbind(estCD, estCDe); dataAll
#tableAll <- xtable(dataAll, caption="U.S. 1980-2011, All Models.", digit = c(4, 4
```

```
print(tableCDq, caption.placement="top")
```

Table 2: US, 1980-2011.

	lambda	alpha	beta	gamma
+ 95% CI	0.0127	0.33	0.84	0.177
C-D with q	0.0105	0.26	0.72	0.021
- 95% CI	0.0089	0.19	0.59	-0.135

```
# According to http://cran.r-project.org/web/packages/xtable/vignettes/xtableGalle
# be able to use the "sanitize.text.function" parameter to allow markup in column
# line is not working at the present time. --MKH, 18 Jan 2012.
# print(tableCDe, sanitize.text.function = function(x){x})
```

```
#print(tableAll, caption.placement="top")
```

2.2. Cobb-Douglas With X

```
energyColumn <- "iX"; energyString <- "x"
dataCDx <- cobbDouglasEnergy(countryName, energyColumn, energyString)

[1] -161.4

Formula: iGDP ~ exp(lambda * iYear) * iCapStk^min(a, b) * iLabor^abs(b -
    a) * iEnergy^(1 - max(a, b))

Parameters:
       Estimate Std. Error t value Pr(>|t|)
lambda  0.01071    0.00105   10.16  4.6e-11
a       0.25689    0.03666    7.01  1.0e-07
b       0.95700    0.07467   12.82  1.8e-13

Residual standard error: 0.018 on 29 degrees of freedom

Algorithm "port", convergence message: relative convergence (4)

Waiting for profiling to be done...

           2.5%   97.5%
lambda 0.008902 0.01287
a      0.182137 0.32622
b      0.803506      NA
[1] 29
[1] 2.045
[1] 0.5791 0.8211
[1] -0.1097  0.1957
   lambda  alpha   beta gamma
1 0.01071 0.2569 0.7001 0.043
              lambda  alpha   beta   gamma
+ 95% CI    0.012867 0.3262 0.8211  0.1957
C-D with x 0.010714 0.2569 0.7001  0.0430
- 95% CI    0.008902 0.1821 0.5791 -0.1097
```

```
tableCDx <- xtable(dataCDx, caption=paste(countryName, ", 1980-2011.", sep=""), di
```

```
print(tableCDx, caption.placement="top")
```

Table 3: US, 1980-2011.

	lambda	alpha	beta	gamma
+ 95% CI	0.0129	0.33	0.82	0.196
C-D with x	0.0107	0.26	0.70	0.043
- 95% CI	0.0089	0.18	0.58	-0.110

2.3. Cobb-Douglas With U

```
energyColumn <- "iU"; energySTring <- "u"
dataCDu <- cobbDouglasEnergy(countryName, energyColumn, energyString)

Error:  Missing value or an infinity produced when evaluating the
model

tableCDu <- xtable(dataCDu, caption=paste(countryName, ", 1980-2011.", sep=""), di

Error:  object 'dataCDu' not found
```

```
print(tableCDu, caption.placement="top")

Error:  error in evaluating the argument 'x' in selecting a method
for function 'print':  Error:  object 'tableCDu' not found
```

3. CES With Q

```r
# Establish guess values for alpha, beta, and lambda.
phiGuess <- -20
betaGuess <- 0.5 # a typical value for beta (exponent on labor)
zetaGuess <- 0.0004 # a small value
lambda_LGuess <- 0.007 #assuming no technical progress on the labor-capital portic
lambda_EGuess <- 0.008 #assuming no technical progress on the energy portion of th

# Runs a non-linear least squares fit to the data with constraints
modelCESQ <- nls(iGDP ~ ((1-zeta) * (exp(lambda_L*iYear) * iLabor^beta * iCapStk^(
                         + zeta*(exp(lambda_E*iYear) * iQ)^phi)^(1/phi),
                 algorithm = "port",
                 control = nls.control(maxiter = 500, tol = 1e-06, minFactor = 1/1
                                       printEval = FALSE, warnOnly = FALSE),
                 start = list(phi=phiGuess, beta=betaGuess, zeta=zetaGuess, lambda
                              lambda_E=lambda_EGuess),
                 lower = list(phi=-Inf, beta=0, zeta=0, lambda_L=-Inf, lambda_E=-I
                 upper = list(phi=0, beta=1, zeta=1, lambda_L=Inf, lambda_E=Inf),
                 data=dataTable)

Error:  object 'dataTable' not found


# Gives the nls summary table
summary(modelCESQ)

Error:  object 'modelCESQ' not found

xyplot( resid(modelCESQ) ~ fitted(modelCESQ) )

Error:  object 'modelCESQ' not found

histogram( ~resid(modelCESQ) )

Error:  object 'modelCESQ' not found

qqmath( ~resid(modelCESQ) )

Error:  object 'modelCESQ' not found
```