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

Caleb, put your LaTeX code here.

```
createCountryFactorsGraph <- function(countryName){
  dataTable <- loadData(countryName)
  graphType <- "l"
  lineTypes <- c(1, 5, 2, 4, 6) #line types. See http://en.wikibooks.org/wiki/R_Pr
  lineWidths <- c(1, 1, 1, 1, 1) #line widths
  colors <- c("black", "black", "black", "black", "black") #line and point colors
  lineSpec <- list(lty=lineTypes, lwd=lineWidths, col=colors)
  graph <- xyplot(iCapStk+iLabor+iQ+iX+iU ~ Year, data=dataTable,
                  type=graphType,
                  par.settings = list(superpose.line = lineSpec),
                  key=list(text=list(c("k", "l", "q", "x", "u")),
                           type=graphType,
                           lines=lineSpec,
                           columns=1, x=0.0, y=0.98),
                  scales=list(cex=1.0, #controls text size on scales
                              tck=-0.5), #controls tick mark length
                  ylab="Indexed (1980=1)")
```

*Corresponding author

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

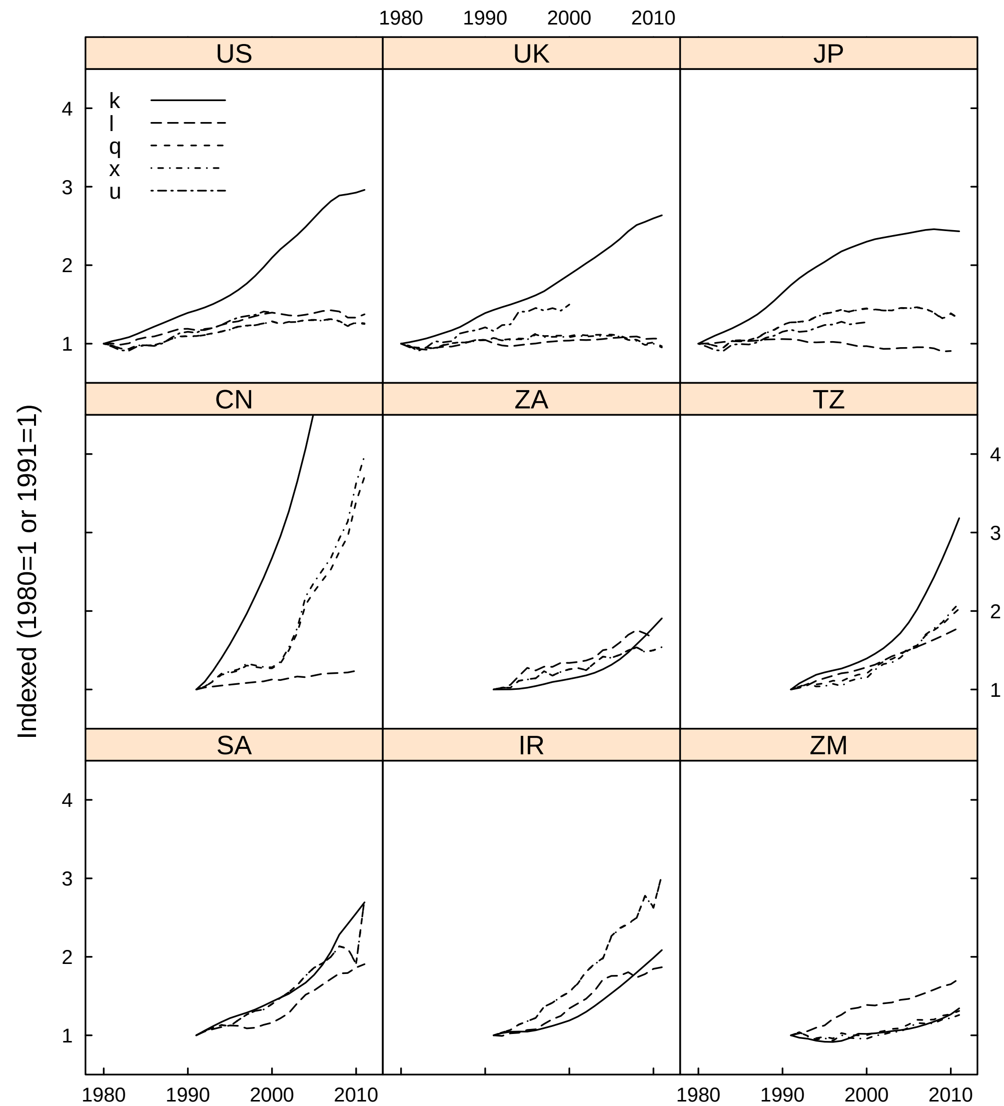

Figure 1: Factors of Production for All Countries. (China's indexed capital stock rises to $k = 9.2$ in 2011.)

```
  return(graph)
}

createGDPComparisonGraph <- function(countryName){
  dataTable <- loadData(countryName)
  graphType <- "l"
  lineTypes <- c(1) #line types. See http://en.wikibooks.org/wiki/R_Programming/Gr
  lineWidths <- c(1) #line widths
  colors <- c("black") #line and point colors
  lineSpec <- list(lty=lineTypes, lwd=lineWidths, col=colors)
  graph <- xyplot(iGDP ~ Year, data=dataTable,
                  key=list(text=list(c("GDP")),
                           type=graphType,
                           lines=lineSpec,
                           columns=1, x=0.0, y=0.98),
                  type=graphType,
                  par.settings = list(superpose.line = lineSpec),
                  scales=list(cex=1.0, #controls text size on scales
                              tck=-0.5), #controls tick mark length
                  ylab="Indexed (1980=1)")
  return(graph)
}


# createGDPComparisonGraph <- function(countryName){
#   dataTable <- loadData(countryName)
#   graphType <- "l"
#   colors <- c("black", "black", "black", "black") #line and point colors
#   lineWidths <- c(1, 1, 1, 1) #line widths
#   lineTypes <- c(1, 5, 2, 4) #line types. See http://en.wikibooks.org/wiki/R_Pro
#   lineSpec <- list(lty=lineTypes, lwd=lineWidths, col=colors)
#   pointSpec <- list(col=colors)
#   graph <- xyplot(iGDP ~ iYear, data=dataTable,
#                   key=list(text=list(c("GDP")),
#                            type=graphType,
#                            lines=lineSpec,
#                            columns=1, x=0.0, y=0.98),
```

```
#                   type=graphType,
#                   par.settings = list(superpose.line = lineSpec),
# #                                    superpose.symbol = pointSpec),
#                   scales=list(cex=1.0, #controls text size on scales
#                               tck=-0.5), #controls tick mark length
#                   ylab="Indexed (1980=1)")
#   return(graph)
#
#
# }



#createGDPComparisonGraph("US")

#createCountryFactorsGraph("US")
#lapply(countries, createCountryFactorsGraph)
```

1. Cobb-Douglas Without Energy

```
#################################################
# Calculates parameter estimates and confidence intervals
# for the Cobb-Douglas production function given a country.
#
# countryName is a string containing the 2-letter abbreviation for the country, e.
#
# returns a vector of data for the Cobb-Douglas model.
# First item is the +95% CI on all parameters
# Second item contains the parameter estimates
# Third item is the -95% CI on all parameters
# Each row has names: lambda, alpha, beta, gamma, corresponding to the parameters
##
```

```
cobbDouglasData <- 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),
#                  algorithm = "port",
                 start = list(lambda=lambdaGuess, alpha=alphaGuess),
#                  lower = list(lambda=-Inf, alpha=0),
#                  upper = list(lambda=Inf, alpha=1),
                 data=dataTable)

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

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

  # 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(ciHalfLevel, df = dofCD); tvalCD
  betaCICD <- with(beta.est, Estimate + c(-1.0, 1.0) * tvalCD * SE); betaCICD # Gi

  #print(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
  #print(estCD)
  row.names(estCD) <- "CD"
  #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)
  #print(dataCD)
  return(dataCD)
}

####################################
# Creates a LaTeX printable table from the Cobb Douglas data. This function first
#
# countryName is a string containint the 2-letter abbreviation for the country, e.
#
# returns a printable LaTeX table from xtable.
##
cobbDouglasTable <- function(countryName){
  dataCD <- cobbDouglasData(countryName)
  colnames(dataCD) <- c("$\\lambda$", "$\\alpha$", "$\\beta$", "$\\gamma$")
  tableCD <- xtable(dataCD, caption=paste(countryName, " Cobb-Douglas, 1980-2011",
  #print(tableCD)
  return(tableCD)
}
```

```
tablesCD <- lapply(countries, cobbDouglasTable)

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

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

```r
print(tablesCD[["US"]], caption.placement="top")
```

Table 1: US Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0116	0.34	0.79	
CD	0.0102	0.27	0.73	0.0
- 95% CI	0.0087	0.21	0.66	

```r
print(tablesCD[["UK"]], caption.placement="top")
```

Table 2: UK Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0303	1.12	1.24	
CD	0.0097	0.44	0.56	0.0
- 95% CI	-0.0104	-0.25	-0.13	

```r
print(tablesCD[["JP"]], caption.placement="top")
```

```r
print(tablesCD[["ZA"]], caption.placement="top")
```

Table 3: JP Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0081	0.61	0.56	
CD	0.0048	0.52	0.48	0.0
- 95% CI	0.0014	0.44	0.39	

Table 4: ZA Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0022	0.73	0.54	
CD	0.0008	0.60	0.40	0.0
- 95% CI	-0.0007	0.46	0.26	

```
print(tablesCD[["CN"]], caption.placement="top")
```

Table 5: CN Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0779	1.32	0.89	
CD	0.0188	0.71	0.29	0.0
- 95% CI	-0.0405	0.11	-0.32	

```
print(tablesCD[["SA"]], caption.placement="top")
```

```
print(tablesCD[["IR"]], caption.placement="top")
```

```
print(tablesCD[["TZ"]], caption.placement="top")
```

Table 6: SA Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	-0.0087	0.68	0.78	
CD	-0.0123	0.45	0.55	0.0
- 95% CI	-0.0159	0.21	0.32	

Table 7: IR Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0045	0.70	0.51	
CD	0.0039	0.60	0.40	0.0
- 95% CI	0.0032	0.49	0.30	

```
print(tablesCD[["ZM"]], caption.placement="top")
```

```
# 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:

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

2.1. Cobb-Douglas with Q

```
# Note that the anlaysis of ZA is taking a long time here. Need to figure out why.
CDqTables <- lapply(countries, cobbDouglasEnergyTable, energyType="Q")
```

Table 8: TZ Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0068	0.95	0.50	
CD	0.0015	0.73	0.27	0.0
- 95% CI	-0.0039	0.50	0.05	

Table 9: ZM Cobb-Douglas, 1980-2011

	λ	α	β	γ
+ 95% CI	0.0280	1.57	-0.25	
CD	0.0249	1.41	-0.41	0.0
- 95% CI	0.0218	1.25	-0.57	

```
print(CDqTables[["US"]], caption.placement="top")
print(CDqTables[["ZA"]], caption.placement="top")
# 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*

```
# Note that the anlaysis of ZA is taking a long time here. Need to figure out why.
CDxTables <- lapply(countries, cobbDouglasEnergyTable, energyType="X")
```

```
print(CDxTables[["US"]], caption.placement="top")
print(CDxTables[["ZA"]], caption.placement="top")
```

2.3. *Cobb-Douglas With U*

```
CDuTables <- lapply(countries, cobbDouglasEnergyTable, energyType="U")
```

```
print(CDuTables[["US"]], caption.placement="top")
print(CDuTables[["ZA"]], caption.placement="top")
```

3. CES

```
cesData <- function(countryName, energyType){
  energyColumnName <- paste("i", energyType, sep="")
  # Load the data that we need.
  dataTable <- loadData(countryName)

  # Establish guess values for phi beta, zeta, lambda_L and lambda_E.
  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 port
  lambda_EGuess <- 0.008 #assuming no technical progress on the energy portion of

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

  aicCES <- AIC(modelCES, k=2) # Checks validity of the model. AIC stands for Akai
  print(aicCES)

  # Gives the nls summary table
```

```
summaryCES <- summary(modelCES) # Gives the nls summary table
print(summaryCES)

# Provides confidence intervals on phi, beta, zeta, lambda_L, and lambda_E. But,
ciCES <- confint(modelCES, level = ciLevel)
print(ciCES)

# Get the estimate for alpha
beta <- as.numeric(coef(modelCES)["beta"])
alpha <- 1.0 - beta
alpha.est <- deltaMethod(modelCES, "1 - beta") # Estimates alpha and its standar
print(alpha.est)

# Now calculate a confidence interval on alpha
dofCES <- summaryCES$df[2]
print(dofCES) # Gives the degrees of freedom for the model.
tvalCES <- qt(ciHalfLevel, df = dofCES); tvalCES
# Get confidence intervals for each parameter in the model
alphaCICES <- with(alpha.est, Estimate + c(-1.0, 1.0) * tvalCES * SE) # CI on al
print(alphaCICES)

# Assemble the data into data frames for the table.
estCES <- data.frame(phi = coef(modelCES)["phi"], alpha = alpha,
                     beta = coef(modelCES)["beta"], zeta = coef(modelCES)["zeta"
                     lambda_L = coef(modelCES)["lambda_L"], lambda_E = coef(mode
row.names(estCES) <- paste("CES with ", energyType, sep="")
#print(estCES)
# The [1] subscripts pick off the lower confidence interval
lowerCES <- data.frame(phi = ciCES["phi","2.5%"], alpha = alphaCICES[1],
                       beta = ciCES["beta", "2.5%"], zeta = ciCES["zeta", "2.5%"
                       lambda_L = ciCES["lambda_L", "2.5%"], lambda_E = ciCES["l
row.names(lowerCES) <- "- 95% CI"
# The [2] subscripts pick off the lower confidence interval
upperCES <- data.frame(phi = ciCES["phi","97.5%"], alpha = alphaCICES[2],
                       beta = ciCES["beta", "97.5%"], zeta = ciCES["zeta", "97.5
                       lambda_L = ciCES["lambda_L", "97.5%"], lambda_E = ciCES["
row.names(upperCES) <- "+ 95% CI"
```

```
  # Now create the data for a table.
  dataCES <- rbind(upperCES, estCES, lowerCES)
  print(dataCES)
  return(dataCES)

  #xyplot( resid(modelCESQ) ~ fitted(modelCESQ) )
  #histogram( ~resid(modelCESQ) )
  #qqmath( ~resid(modelCESQ) )
}

####################################
# Creates a LaTeX printable table from the CES data. This function first calls ces
#
# countryName is a string containint the 2-letter abbreviation for the country, e.
# energyType is a string to be used in table captions reprsenting the type of ener
#
# returns a printable LaTeX table from xtable.
##
cesTable <- function(countryName, energyType){
  dataCESe <- cesData(countryName, energyType)
  tableCESq <- xtable(dataCESe, caption=paste(countryName, ", 1980-2011.", sep="")
}
```

3.1. CES with Q

```
countryName <- "US"
energyType <- "Q"
tableCESq <- cesTable(countryName, energyType)

[1] -194

Formula: iGDP ~ ((1 - zeta) * (exp(lambda_L * iYear) * iCapStk^(1 - beta) *
    iLabor^beta)^phi + zeta * (exp(lambda_E * iYear) * iQ)^phi)^(1/phi)

Parameters:
          Estimate Std. Error t value Pr(>|t|)
```

```
phi      -3.96e+01   2.43e+01   -1.63   0.1144
beta      6.09e-01   3.45e-02   17.64  2.4e-16
zeta      2.09e-06   1.32e-05    0.16   0.8758
lambda_L  7.98e-03   6.68e-04   11.95  2.8e-12
lambda_E  8.57e-03   2.48e-03    3.45   0.0018

Residual standard error: 0.0105 on 27 degrees of freedom

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

Waiting for profiling to be done...

             2.5%      97.5%
phi            NA -10.290831
beta     0.514667   0.665371
zeta           NA         NA
lambda_L 0.006428   0.009152
lambda_E 0.000715   0.012468
         Estimate      SE
1 - beta   0.3911 0.03453
[1] 27
[1] 0.3202 0.4619
              phi  alpha   beta      zeta lambda_L lambda_E
+ 95% CI   -10.29 0.4619 0.6654        NA 0.009152 0.012468
CES with Q -39.64 0.3911 0.6089 2.085e-06 0.007979 0.008570
- 95% CI       NA 0.3202 0.5147        NA 0.006428 0.000715


#CESqTables <- lapply(countries, cesTable, energyType="Q")
```

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

```
#print(CESqTables[["US"]], caption.placement="top")
#print(CESqTables[["ZA"]], caption.placement="top")
```

Table 10: US, 1980-2011.

	phi	alpha	beta	zeta	lambda_L	lambda_E
+ 95% CI	-10.3	0.46	0.67		0.00915	0.01247
CES with Q	-39.6	0.39	0.61	0.000002	0.00798	0.00857
- 95% CI		0.32	0.51		0.00643	0.00071