#

# **Gold Price Function1978-2013**

#

# Regression for: Gold = …

#

# include library for Wald, RESET, and other tests

library(survey)

library(lmtest)

library(tseries)

library(fUnitRoots)

# eliminate at most one independent variable from regression

elim\_regress\_variable <- function(vars, elim="")

{

for (i in 1:NROW(vars))

{

if(elim==vars[i])

{

vars\_new <- vars[-i]

}

}

return(vars\_new)

}

# eliminate variable from group listings

elim\_group\_variable <- function(indep\_var\_groups, elim="")

{

groups <- indep\_var\_groups

# find group with this possible elimination candidate

for (i in 1:NROW(groups))

{

grp <- groups[[i]]

grp\_size <- NROW(grp)

for (j in 1:grp\_size)

{

#print(grp[j])

if (!is.na(grp[j]) && elim==grp[j])

{

grp <- grp[-j]

}

}

groups[[i]] <- grp

}

return(groups)

}

# create regression string for lm()

create\_regress\_str <- function(dep\_var, indep\_var, long\_term)

{

str <- paste(dep\_var," ~ ")

first <- 1

for (i in 1:NROW(indep\_var))

{

if (first==1)

{

str <- paste(str,indep\_var[i],sep="")

first <- 0

}

else

{

str <- paste(str,indep\_var[i],sep=" + ")

}

}

for (i in 1:NROW(long\_term))

{

str <- paste(str,long\_term[i],sep=" + ")

}

return(str)

}

# automatically eliminate least significant variable making sure

# to retain at least one per group; stop when all are at least

# significant at 10% level or are last in respective groups.

choose\_elim\_variable <- function(vars, groups, results)

{

sig\_level = 0.1

results\_data <- results$coeff

probs <- results\_data[,4]

prob\_order <- order(probs,decreasing=TRUE)

elim = "-1"

index = 1

max\_probs <- NROW(probs)

#print(paste("max\_probs: ",max\_probs, sep=''))

#print(probs)

#print(prob\_order)

repeat

{

if (index > max\_probs) break

var\_name <- names(probs[prob\_order[index]])

var\_value <- probs[[prob\_order[index]]]

#print(paste("index=",index,sep=''))

#print(paste("var\_name=",var\_name,sep=''))

#print(paste("var\_value=",var\_value,sep=''))

# find group with this possible elimination candidate

for (i in 1:NROW(groups))

{

grp <- groups[[i]]

grp\_size <- NROW(grp)

for (j in 1:grp\_size)

{

if (grp[j] == var\_name)

{

#print(grp[j])

if (grp\_size > 1 && var\_value > sig\_level)

{

return(var\_name)

}

}

}

}

index <- index + 1

}

return(elim)

}

# make the input file a CSV file from your Excel spreadsheet, and write to the output file in sink().

# you need to have the full directory name included!

goldc <- read.csv("1978.csv",header=TRUE)

sink("1978goldintg.pdf", type="output", split=TRUE)

# import all data as time series and take logs if needed; the names after the dollar sign

# are the column names in the spreadsheet (or \*.csv file)

PG<- ts(log(goldc$PG))

Note<- ts(log(goldc$Note))

SP<- ts(log(goldc$SP))

Dollar<- ts(log(goldc$Dollar))

U<- ts(log(goldc$U))

CPI<- ts(goldc$CPI)

# get first difference of each variable

dPG <- diff(PG)

dNote <- diff(Note)

dSP <- diff(SP)

dDollar <- diff(Dollar)

dU<-diff(U)

dCPI<-diff(CPI)

# generate lag variables

lPG <- lag(PG,-1)

lNote <- lag(Note,-1)

lSP <- lag(SP,-1)

lDollar <- lag(Dollar,-1)

lU<-lag(U,-1)

lCPI<-lag(CPI,-1)

dPG1 <- lag(dPG,-1)

dPG2 <- lag(dPG,-2)

dPG3 <- lag(dPG,-3)

dPG4 <- lag(dPG,-4)

dPG5 <- lag(dPG,-5)

dPG6 <- lag(dPG,-6)

dPG7 <- lag(dPG,-7)

dPG8 <- lag(dPG,-8)

dPG9 <- lag(dPG,-9)

dPG10 <- lag(dPG,-10)

dPG11 <- lag(dPG,-11)

dPG12 <- lag(dPG,-12)

dNote1 <- lag(dNote,-1)

dNote2 <- lag(dNote,-2)

dNote3 <- lag(dNote,-3)

dNote4 <- lag(dNote,-4)

dNote5 <- lag(dNote,-5)

dNote6 <- lag(dNote,-6)

dNote7 <- lag(dNote,-7)

dNote8 <- lag(dNote,-8)

dNote9 <- lag(dNote,-9)

dNote10 <- lag(dNote,-10)

dNote11 <- lag(dNote,-11)

dNote12 <- lag(dNote,-12)

dSP1<- lag(dSP,-1)

dSP2<- lag(dSP,-2)

dSP3 <- lag(dSP,-3)

dSP4 <- lag(dSP,-4)

dSP5 <- lag(dSP,-5)

dSP6 <- lag(dSP,-6)

dSP7 <- lag(dSP,-7)

dSP8 <- lag(dSP,-8)

dSP9 <- lag(dSP,-9)

dSP10 <- lag(dSP,-10)

dSP11 <- lag(dSP,-11)

dSP12 <- lag(dSP,-12)

dDollar1 <- lag(dDollar,-1)

dDollar2 <- lag(dDollar,-2)

dDollar3 <- lag(dDollar,-3)

dDollar4 <- lag(dDollar,-4)

dDollar5 <- lag(dDollar,-5)

dDollar6 <- lag(dDollar,-6)

dDollar7 <- lag(dDollar,-7)

dDollar8 <- lag(dDollar,-8)

dDollar9 <- lag(dDollar,-9)

dDollar10 <- lag(dDollar,-10)

dDollar11 <- lag(dDollar,-11)

dDollar12 <- lag(dDollar,-12)

dU1 <- lag(dU,-1)

dU2 <- lag(dU,-2)

dU3 <- lag(dU,-3)

dU4 <- lag(dU,-4)

dU5 <- lag(dU,-5)

dU6 <- lag(dU,-6)

dU7 <- lag(dU,-7)

dU8 <- lag(dU,-8)

dU9 <- lag(dU,-9)

dU10 <- lag(dU,-10)

dU11 <- lag(dU,-11)

dU12 <- lag(dU,-12)

dCPI1 <- lag(dCPI,-1)

dCPI2 <- lag(dCPI,-2)

dCPI3 <- lag(dCPI,-3)

dCPI4 <- lag(dCPI,-4)

dCPI5 <- lag(dCPI,-5)

dCPI6 <- lag(dCPI,-6)

dCPI7 <- lag(dCPI,-7)

dCPI8 <- lag(dCPI,-8)

dCPI9 <- lag(dCPI,-9)

dCPI10 <- lag(dCPI,-10)

dCPI11 <- lag(dCPI,-11)

dCPI12 <- lag(dCPI,-12)

# create dependent variable string and vector of all long term variables

dep\_var = "dPG"

long\_term <- c("lPG","lNote","lSP","lDollar","lU","lCPI")

# create vector of all independent variable names

indep\_vars <- c("dPG1","dPG2","dPG3","dPG4","dPG5","dPG6","dPG7","dPG8","dPG9","dPG10","dPG11","dPG12",

"dNote","dNote1","dNote2","dNote3","dNote4","dNote5","dNote6","dNote7","dNote8","dNote9","dNote10","dNote11","dNote12",

"dSP","dSP1","dSP2","dSP3","dSP4","dSP5","dSP6","dSP7","dSP8","dSP9","dSP10","dSP11","dSP12",

"dDollar","dDollar1","dDollar2","dDollar3","dDollar4","dDollar5","dDollar6","dDollar7","dDollar8","dDollar9","dDollar10","dDollar11","dDollar12",

"dU","dU1","dU2","dU3","dU4","dU5","dU6","dU7","dU8","dU9","dU10","dU11","dU12",

"dCPI","dCPI1","dCPI2","dCPI3","dCPI4","dCPI5","dCPI6","dCPI7","dCPI8","dCPI9","dCPI10","dCPI11","dCPI12")

# create groups of independent variables such that at least one member of

# each group must remain when using Henry's general to specific elimination

indep\_vars\_groups <- list(c("dPG1","dPG2","dPG3","dPG4","dPG5","dPG6","dPG7","dPG8","dPG9","dPG10","dPG11","dPG12"),

c("dNote","dNote1","dNote2","dNote3","dNote4","dNote5","dNote6","dNote7","dNote8","dNote9","dNote10","dNote11","dNote12"),

c("dSP","dSP1","dSP2","dSP3","dSP4","dSP5","dSP6","dSP7","dSP8","dSP9","dSP10","dSP11","dSP12"),

c("dDollar","dDollar1","dDollar2","dDollar3","dDollar4","dDollar5","dDollar6","dDollar7","dDollar8","dDollar9","dDollar10","dDollar11","dDollar12"),

c("dU","dU1","dU2","dU3","dU4","dU5","dU6","dU7","dU8","dU9","dU10","dU11","dU12"),

c("dCPI","dCPI1","dCPI2","dCPI3","dCPI4","dCPI5","dCPI6","dCPI7","dCPI8","dCPI9","dCPI10","dCPI11","dCPI12"))

# create time series dataset including the dependent and all possible independent variables

tsdata <- ts.union(dPG,dPG1,dPG2,dPG3,dPG4,dPG5,dPG6,dPG7,dPG8,dPG9,dPG10,dPG11,dPG12,

dNote,dNote1,dNote2,dNote3,dNote4,dNote5,dNote6,dNote7,dNote8,dNote9,dNote10,dNote11,dNote12,

dSP,dSP1,dSP2,dSP3,dSP4,dSP5,dSP6,dSP7,dSP8,dSP9,dSP10,dSP11,dSP12,

dDollar,dDollar1,dDollar2,dDollar3,dDollar4,dDollar5,dDollar6,dDollar7,dDollar8,dDollar9,dDollar10,dDollar11,dDollar12,

dU,dU1,dU2,dU3,dU4,dU5,dU6,dU7,dU8,dU9,dU10,dU11,dU12,

dCPI,dCPI1,dCPI2,dCPI3,dCPI4,dCPI5,dCPI6,dCPI7,dCPI8,dCPI9,dCPI10,dCPI11,dCPI12,

lPG,lNote,lSP,lDollar,lU,lCPI)

# begin with a string that sets up the regression equation with dependent variable

# followed by "~" and then ALL independent variables set

str <- create\_regress\_str(dep\_var, indep\_vars,long\_term)

regress <- lm(str, data=tsdata)

results <- summary(regress)

print(results)

# repeat regression eliminating one variable at a time for "elim" either by

# 1. TESTING: enter variables until "-1" is entered in console for testing

# 2. Calling choose\_elim\_variable() to eliminate completely through variables

repeat

{

#elim <- readline(paste("Which variable to eliminate (e.g. dlimp4 or -1 to end)? ", sep=""))

elim <- choose\_elim\_variable(indep\_vars, indep\_vars\_groups, results)

#print(elim)

if (elim=="-1") break

indep\_vars <- elim\_regress\_variable(indep\_vars, elim)

indep\_vars\_groups <- elim\_group\_variable(indep\_vars\_groups, elim)

str <- create\_regress\_str(dep\_var, indep\_vars, long\_term)

regress <- lm(str, data=tsdata)

results <- summary(regress)

}

print(results)

# RUN TESTS!!!

# Wald test

wald <- regTermTest(regress, ~lPG+lNote+lSP+lCPI+lDollar+lU, method="Wald")

print(wald)

coeffs <- regress$coefficients

coeff\_c <- -coeffs["(Intercept)"]/coeffs["lPG"]

coeff\_lNote <- -coeffs["lNote"]/coeffs["lPG"]

coeff\_lSP <- -coeffs["lSP"]/coeffs["lPG"]

coeff\_lDollar <- -coeffs["lDollar"]/coeffs["lPG"]

coeff\_lU <- -coeffs["lU"]/coeffs["lPG"]

coeff\_lCPI <- -coeffs["lCPI"]/coeffs["lPG"]

# store standard errors in "stderrors"

stderrors <- results$coefficients[,2]

stderror\_c <- stderrors["(Intercept)"]

stderror\_lNote <- stderrors["lNote"]

stderror\_lSP <- stderrors["lSP"]

stderror\_lDollar <- stderrors["lDollar"]

stderror\_lU <- stderrors["lU"]

stderror\_lCPI <- stderrors["lCPI"]

# store t-stats values in "tstats"

tstats <- results$coefficients[,3]

tstats\_c <- tstats["(Intercept)"]

tstats\_lNote <- tstats["lNote"]

tstats\_lSP <- tstats["lSP"]

tstats\_lDollar <- tstats["lDollar"]

tstats\_lU <- tstats["lu"]

tstats\_lCPI <- tstats["lCPI"]

# store probabilities in "probs"

results\_data <- results$coeff

probs <- results\_data[,4]

probs\_c <- probs["(Intercept)"]

probs\_lNote <- probs["lNote"]

probs\_lSP <- probs["lSP"]

probs\_lDollar <- probs["lDollar"]

probs\_lU <- probs["lU"]

probs\_lCPI <- probs["lCPI"]

# print ecm coeffs, std-errors, t-stats, and probs

cat("Long Run Elasticities:\n")

cat("\t\t","Coeff\t\t","Std-Errors\t","t-stat\t\t","prob","\n")

cat("(Intercept)\t",coeff\_c,"\t",stderror\_c,"\t",tstats\_c,"\t",probs\_c,"\n")

cat("lNote\t\t",coeff\_lNote,"\t",stderror\_lNote,"\t",tstats\_lNote,"\t",probs\_lNote,"\n")

cat("lSP\t",coeff\_lSP,"\t",stderror\_lSP,"\t",tstats\_lSP,"\t",probs\_lSP,"\n")

cat("lDollar\t",coeff\_lDollar,"\t",stderror\_lDollar,"\t",tstats\_lDollar,"\t",probs\_lDollar,"\n")

cat("lCPI\t\t",coeff\_lCPI,"\t",stderror\_lCPI,"\t",tstats\_lCPI,"\t",probs\_lCPI,"\n")

cat("lU\t\t",coeff\_lU,"\t",stderror\_lU,"\t",tstats\_lU,"\t",probs\_lU,"\n\n\n")

# create ecm solution for long-range approx.

ecm <- PG - coeff\_c - coeff\_lNote \* Note - coeff\_lSP \* SP - coeff\_lDollar \* Dollar - coeff\_lU \* U - coeff\_lCPI \* CPI

ecm1 <- lag(ecm, -1)

tsdata <- ts.union(dPG,dPG1,dPG2,dPG3,dPG4,dPG5,dPG6,dPG7,dPG8,dPG9,dPG10,dPG11,dPG12,

dNote,dNote1,dNote2,dNote3,dNote4,dNote5,dNote6,dNote7,dNote8,dNote9,dNote10,dNote11,dNote12,

dSP,dSP1,dSP2,dSP3,dSP4,dSP5,dSP6,dSP7,dSP8,dSP9,dSP10,dSP11,dSP12,

dDollar,dDollar1,dDollar2,dDollar3,dDollar4,dDollar5,dDollar6,dDollar7,dDollar8,dDollar9,dDollar10,dDollar11,dDollar12,

dU,dU1,dU2,dU3,dU4,dU5,dU6,dU7,dU8,dU9,dU10,dU11,dU12,

dCPI,dCPI1,dCPI2,dCPI3,dCPI4,dCPI5,dCPI6,dCPI7,dCPI8,dCPI9,dCPI10,dCPI11,dCPI12,

lPG,lNote,lSP,lDollar,lU,lCPI,

ecm1)

# do FINAL regression WITH ecm

long\_term <- "ecm1"

str <- create\_regress\_str(dep\_var, indep\_vars, long\_term)

regress <- lm(str, data=tsdata)

results <- summary(regress)

print(regress)

print(results)

# Durbin-Watson test

dw <- dwtest(regress,data=tsdata)

print(dw)

# Breusch-Godfrey test

# bgtest(formula, order = 1, type = c("Chisq", "F"), data = list())

bp <- bgtest(regress,order=4,type="F",data=tsdata)

print(bp)

# Ramsey RESET test, fitted with quadratic

# resettest(formula, power = 2:3, type = c("fitted", "regressor", "princomp"), data = list())

ramsey <- reset(regress, power=2:3, type="fitted", data=tsdata)

print(ramsey)

# White's heteroskedastic test

#white <- vcovHC(regress, data=tsdata)

#white <- summaryw(regress)

#white <- ncv.test(regress)

#print(white)

#hetero\_cov\_mat <- hccm(regress)

#print(hetero\_cov\_mat)

# Jarque-Bera normality test

# skip first 3 NA values because of volatility

jarque\_test <- jarque.bera.test(ecm[4:length(ecm)])

print(jarque\_test)

# Dickey-Fuller unit root test

# skip first 3 NA values because of volatility

adf\_test <- adf.test(ecm[4:length(ecm)])

print(adf\_test)

# Unit root tests

print("Unit Root Test")

unitRootTest1 <- unitrootTest(ecm[4:length(ecm)],lags=4)

print(unitRootTest1)

# Augmented Dickey-Fuller test for unit roots using "urdfTest"

dickey <- urdfTest(ecm[4:length(ecm)], lags=4)

print(dickey)

# Phillips-Perron test for unit roots

phillips <- urppTest(ecm[4:length(ecm)], use.lag=4, doplot=TRUE)

print(phillips)

# Elliott-Rothenberg-Stock test for unit roots

elliot <- urersTest(ecm[4:length(ecm)])

print(elliot)

# Schmidt-Phillips test for unit roots

schmidt <- urspTest(ecm[4:length(ecm)])

print(schmidt)

sink("1978goldintg.pdf", type="output", split=TRUE)