```
#PO 841
#12/3/2014
#LAB 11: INTERACTION TERMS AND OTHER MODEL MANIPULATIONS
# ========================================================

# --------------------
# INTERACTION TERMS
# --------------------

#Let's revisit the OECD data on national health outcomes:

library(foreign)
health<-read.dta(file.choose(),convert.factors=F)
names(health)
health$country

#Let's regress public health expenditures against GNP per capita and
 plot the relationship between the two variables:

reghealthgdppc<-lm(pubhth~gnppc, data=health)
plot(pubhth~gnppc, data=health, pch=19)
abline(reghealthgdppc)

#Last week, we identified the country outliers. Recall that the
 labels=country argument tells R to label the points with the data set's
 country names; if we leave the labels=() argument out, R will label the
 plot with identification (or case) numbers. (Also remember to hit ESC
 when done identifying.)

identify(health$gnppc, health$pubhth, labels=health$country)

#Because we see a few outliers that represent different regions of the
 world (e.g., Sweden, Turkey), it seems that region may matter for this
 relationship. So let's create some new categorical variables that
 represent region. First, we'll create a new variable specifying whether
 or not the country is in Europe or not. This will return a logical
 vector that will specify whether or not a country is in Europe (FALSE
 or TRUE).

health$country
```

```r
europe<-health$country %in% c("Portugal", "Greece", "Spain", "Italy",
"Ireland", "United Kingdom", "Netherlands", "Belgium", "Austria",
"France", "Germany", "Denmark", "Finland", "Norway", "Sweden",
"Switzerland")

#We're telling R here, essentially, for all values of health$country,
is the country one of the following (Portugal, Greece, Spain, etc.),
true or false?

europe
summary(europe)

#Last week, we showed how to add a dummy variable (in this case,
"europe") to a regression model and plot the relationship between GDP
and health expenditures conditioned upon whether a country was in
Europe or not. We added the dummy variable "europe" to the original
bivariate model and ran a regression:

reghealthgdppceurope<-lm(pubhth ~ gnppc + europe, data=health) #The
coefficient will give you the effect on Y of "europe"'s being TRUE.
summary(reghealthgdppceurope)

#Then we subsetted our data according to country location:

eur<-subset(health, europe=="TRUE")
eur
noneur<-subset(health, europe=="FALSE")
noneur

#And then ran separate regressions for each subset:

regeur<-lm(eur$pubhth~eur$gnppc, data=eur)
summary(regeur)

regnoneur<-lm(noneur$pubhth~noneur$gnppc, data=noneur)
summary(regnoneur)

#And then plotted the data:
```

```
plot(pubhth~gnppc, main="Effect of GNPPC on Public Health Spending for
 European vs. Non-European Countries", ylab="Percent of GDP on Health",
 xlab="GNP per capita", pch=19, data=health)

#And finally, we superimposed our regression lines -- black for the
 relationship between GNPPC and public health spending in European
 countries, and red for the relationship between GNPPC and public health
 spending in non-European countries (as well as the original bivariate
 regression line estimating the relationship between public health and
 GNPPC across all OECD countries):

abline(regeur, col="black", lty=1, lwd=2)
abline(regnoneur, col="red", lty=1, lwd=2)
abline(reghealthgdppc, col="green", lty=1, lwd=2)

legend("bottomright", legend=c("Europe","Other OECD Countries", "All
 OECD Countries"), lty=1, lwd=2,  col=c("black", "red", "green"))

#As we can easily see, the magnitude of the relationship between GNP
 and public health spending is different for different regions. We can
 think of this difference in magnitude of relationships (and possibly
 even in the direction of relationships, though obviously not in this
 example) as two separate regressions performed on two separate sets of
 data. But the term for how a relationship changes based on certain
 conditions is the "interaction effect," expressed in a regression model
 by an interaction term. We can write interaction terms like this:

lm(pubhth ~ gnppc * europe, data=health) #"GNPPC" is interacted with
 "europe"

#Simply put, an interaction term measures the effect on the DV when X1
 is interacted with X2. In other words, what is the relationship of X1
 to Y when X2 equals a certain value?

#let's take a look at how to interpret interaction term coefficients:

reg.interaction<-lm(pubhth ~ gnppc + europe + gnppc * europe,
 data=health)
summary(reg.interaction)
```

```
#What does the summary report tell us? The interaction term coefficient
 is -.000124; this tells us that when europe = TRUE, the slope
 coefficient for GNPPC changes by -.000124. Thus: When europe=FALSE, for
 every one-unit increase in GNPPC, public health spending increases by .
 0001928 (B1); when europe=TRUE, for every one-unit increase in GNPPC,
 public health spending increases by .0000688 (.0001928 - .000124, or B1
 + B3).

#How do we plot interaction terms? Exactly the way we plotted the
 different regressions on the different subsets of data above. Think of
 an interaction term as telling you what will happen to the relationship
 between X1 and Y if X2 is a certain value (i.e., if the data is subset
 according to X2). Let's go through it again, in a different way.

#First, let's subset our data according to region (i.e., the value of
 "europe"):

europe.TRUE<-data.frame(gnppc=range(health$gnppc), europe=T)
europe.FALSE<-data.frame(gnppc=range(health$gnppc), europe=F)

#Now let's create a regression line that will document the relationship
 between GNPPC and public health expenditures when europe=TRUE and when
 europe=FALSE. Last time we ran separate regressions for each subset of
 data; this time, let's tell R to draw lines based on what the
 regression model that includes our interaction term gnppc * europe
 would predict:

plot(pubhth~gnppc, main="Effect of GNPPC on Public Health Spending for
 European vs. Non-European Countries", ylab="Percent of GDP on Health",
 xlab="GNP per capita", pch=19, data=health)

lines(range(health$gnppc), predict(reg.interaction, europe.TRUE),
 lty=1, lwd=2, col="black")

lines(range(health$gnppc), predict(reg.interaction, europe.FALSE),
 lty=1, lwd=2, col="red")

legend("bottomright", legend=c("Europe","Other OECD Countries"), lty=1,
 lwd=2, col=c("black", "red"))
```

```r
#Let's compare and contrast -- my point here is to show that plotting interaction terms essentially amounts to plotting the different slopes for different subsets of the data:

par(mfrow=c(1,2))

plot(pubhth~gnppc, main="Effect of GNPPC on Public Health Spending for European vs. Non-European Countries", ylab="Percent of GDP on Health", xlab="GNP per capita", pch=19, data=health)
abline(regeur, col="black", lty=1, lwd=2)
abline(regnoneur, col="red", lty=1, lwd=2)

plot(pubhth~gnppc, main="Effect of GNPPC on Public Health Spending for European vs. Non-European Countries", ylab="Percent of GDP on Health", xlab="GNP per capita", pch=19, data=health)
lines(range(health$gnppc), predict(reg.interaction, europe.TRUE), lty=1, lwd=2, col="black")
lines(range(health$gnppc), predict(reg.interaction, europe.FALSE), lty=1, lwd=2, col="red")
legend("topleft", legend=c("Europe","Other OECD Countries"), lty=1, lwd=2, col=c("black", "red"))

#Interpreting and plotting interaction terms with dummy variables (e.g., europe) seems pretty straightforward, since conditionality is binary (the country is either in Europe or it isn't). What if we wanted to include a term that interacted two continuous variables?

#For example, what if we wanted to see how the relationship between GNPPC and public health spending would change when life expectancy = a certain value?

reg.interaction.2<-lm(pubhth ~ gnppc + lifexp + gnppc * lifexp, data=health)
summary(reg.interaction.2)

#What do these results tell us?

#B1: For every one-unit increase in GNPPC, public health spending will increase by .001021 IF life expectancy = zero (thus, the interpretation of B1 is substantively insignificant).
```

```
#B2: For every one-unit increase in life expectancy, public health
 spending will increase by .2178 IF GNPPC = zero (again, B2 is
 substantively insignificant in this case).

#B0: Public health spending will be about -.1351 percent of GDP when
 GNPPC and life expectancy = 0 (the intercept is extremely substantively
 insignificant!)

#B3: The interaction term coefficient, however, tells us what will
 happen to the relationship between GNPPC and public health spending as
 life expectancy increases. For a one-unit change in life expectancy,
 the effect of GNPPC on public health expenditures is .00100914 (i.e, B1
 + B3, or .001021 - .00001186). In effect, B3 (-.00001186, the
 interaction coefficient) tells us what happens to B1 with every one-
 unit change in X2. So, if life expectancy increased by 2 years, the
 effect of GNPPC on public health spending would be . 001021 + 2(-.
 00001186), and so on and so forth.

#The interaction term coefficient also tells us what will happen to the
 relationship between life expectancy and public health spending as
 GNPPC increases. For a one-unit change in GNPPC, we can expect the
 relationship between life expectancy and public health spending to
 change by -.00001186 (B3).

#But we're likely not going to worry about the change in the
 relationship between life expectancy and public health spending for
 every single unit change in GNPPC. Instead, we might be interested in
 seeing how the relationship between life expectancy and public health
 spending varies according to specific barometers -- say, countries with
 a GNPPC under $15,000, countries with GNPPCs from $15,000-$20,000 and
 countries with GNNPC > $20,000. We can plot these differences easily.

#First, let's subset our data according to the conditions we are
 interested in:

gnppc.low<-subset(health, health$gnppc < 15000)
summary(gnppc.low)

gnppc.middle<-subset(health, health$gnppc >= 15000 & health$gnppc <
```

```
 20000)
summary(gnppc.middle)

gnppc.high<-subset(health, health$gnppc >= 20000)
summary(gnppc.high)

#Now let's run regressions on each of our three subsets of data:

reg.gnppc.low<-lm(pubhth ~ lifexp, data=gnppc.low)
summary(reg.gnppc.low)

reg.gnppc.middle<-lm(pubhth ~ lifexp, data=gnppc.middle)
summary(reg.gnppc.middle)

reg.gnppc.high<-lm(pubhth ~ lifexp, data=gnppc.high)
summary(reg.gnppc.high)

#And now let's plot the different slopes for each of our three
 categories:

plot(pubhth ~ lifexp, main="Life Expectancy's Effect on Public Health
 Spending, by GNPPC Bracket", ylab="Public Health Spending",
 xlab="National Life Expectancy", pch=19, data=health)

abline(reg.gnppc.low, col="black", lty=1, lwd=2)
abline(reg.gnppc.middle, col="red", lty=1, lwd=2)
abline(reg.gnppc.high, col="green", lty=1, lwd=2)

legend("bottomright", legend=c("GNPPC < $15,000","GNPPC between $15,000
 - $20,000", "GNPPC >= $20,000"), lty=1, lwd=2,  col=c("black", "red",
 "green"))

# ---------------
# CURVE-FITTING
# ---------------

#Let's say that you suspect a few of your OLS assumptions are being
 violated, and you want to see if a curvilinear model (as opposed to a
 linear model -- a  straight line) will fit your data better.
```

```r
plot(pubhth ~ gnppc, data=health, pch=19)
abline(reghealthgdppc)

plot(health$gnppc, reghealthgdppc$residuals, pch=19)
abline(h=0)

#Looks like our OLS assumptions are being violated. But would a
 curvilinear model fit the data any better than a straight line? We can
 add a quadratic term to our original regression model and see:

reghealthgdppc.quadratic<-lm(pubhth ~ gnppc + I(gnppc^2), data=health)
summary(reghealthgdppc.quadratic)

#The quadratic term is not significant, so we fail to reject the
 hypothesis that a linear model is acceptable for these data (the
 implied alternative is that a curvilinear model is better for these
 data).

#In order to plot the quadratic model, let's create a grid of possible
 GNPPC values and, using the quadratic model and R's predict() function,
 predict the values of y-hat for each value of x. (We have to do this so
 that we can actually plot the curve; we cannot use the abline()
 function, which only works for linear relationships. So, we need to
 write the code ourselves for a vector of predicted y-hats for the model
 that includes the quadratic term.)

gnppc.new<-seq(1,350000, 1000)

#Indicate that R should predict y-hat for the newly defined values of
 GNPPC ("gnppc.new")
predictions<-predict(reghealthgdppc.quadratic,
 data.frame(gnppc=gnppc.new, gnppc2=gnppc.new^2))

#Plot the original data and then superimpose both the linear regression
 line and the quadratic regression line:
plot(health$gnppc, health$pubhth, pch=19)
abline(reghealthgdppc, lwd=2, col="green")
lines(gnppc.new, predictions, lwd=2, col="red")
```

```r
#You can also use the predict() function to plot the relationship
 between, say, a logged x variable -- log(x) -- and y, or a cubed x
 variable -- x^3 -- and y, and so on:

#For a logged model:

reghealthgdppc.log<-lm(pubhth ~ gnppc + log(gnppc), data=health)
summary(reghealthgdppc.log)
gnppc.new.2<-seq(1,350000, 1000)
predictions.2<-predict(reghealthgdppc.log, data.frame(gnppc=gnppc.new.
 2, gnppc2=log(gnppc.new.2)))
plot(health$gnppc, health$pubhth, pch=19)
abline(reghealthgdppc, lwd=2, col="green")
lines(gnppc.new.2, predictions.2, lwd=2, col="red")


#Or a cubed model:

reghealthgdppc.cubed<-lm(pubhth ~ gnppc + I(gnppc^3), data=health)
summary(reghealthgdppc.cubed)
gnppc.new.3<-seq(1,350000, 1000)
predictions.3<-predict(reghealthgdppc.cubed,
 data.frame(gnppc=gnppc.new.3, gnppc2=gnppc.new.3^3))
plot(health$gnppc, health$pubhth, pch=19)
abline(reghealthgdppc, lwd=2, col="green")
lines(gnppc.new.3, predictions.3, lwd=2, col="red")

#Compare the various models (none of them seem to fit the data
 especially well):

plot(health$gnppc, health$pubhth, pch=19)
abline(reghealthgdppc, lwd=2, col="black")
lines(gnppc.new, predictions, lwd=2, col="red")
lines(gnppc.new.2, predictions.2, lwd=2, col="green")
lines(gnppc.new.3, predictions.3, lwd=2, col="blue")

legend("bottomright", legend=c("Linear model","Quadratic Model",
 "Logged Model", "Cubed Model"), lty=1, lwd=2,  col=c("black", "red",
 "green", "blue"))
```

```r
# ----------------------
# MATRIX ALGEBRA IN R
# ----------------------

#Let's go through a few simple operations in R.

x<-matrix(c(4, 9 ,8, 5), nrow=2, ncol=2)
x

t(x) #X'

x %*% t(x) #X'X -- use %*% to multiply matrices

solve(x) #X-1 if X is a square matrix

#Let's take the example from Homework #9, and use R to find the matrix
 of beta coefficients:

x<-matrix(c(1,1,1,1,0,1,3,4), nrow=4, ncol=2)
x
y<-c(5,7,8,10)
y

colMeans(x)
colSums(x)
rowMeans(x)
rowSums(x)

#Find X'X:
x.prime.x<-t(x) %*% x
x.prime.x

#Find X'X-1:
x.prime.x.inverse<-solve(x.prime.x)
x.prime.x.inverse

#Find X'Y:
x.prime.y<-t(x) %*% y
x.prime.y
```

```
#Find X'X-1X'Y:
x.prime.x.inverse %*% x.prime.y
```