```
#PO 841
#11/12/2014
#LAB 9: BIVARIATE REGRESSION

# ---------------------------
# LINEAR REGRESSION FUNCTIONS
# ---------------------------

#Bivariate regression in R can be done relatively simply using the lm()
 function (for Linear Model). The syntax is lm(y~x); this would
 correspond to the regression model y = b0 + b1x + e. (Replace "y" and
 "x" with the names of your actual y and x variables.)

#We can run bivariate regressions on data that you read in from an
 external source or manually-entered data. Let's say we are looking at
 what would seem intuitively to be a pretty obvious relationship --
 between test scores and hours studied:

scores<-c(48,76,50,76,60,54,92,69,65,91,69,83,78,98)
hours<-c(2,4,3,4,3,4,6,6,5,7,5,5,6,8)
names<-c("Joe", "Mike", "Bill", "Claire", "Sarah", "Kate", "Emily",
 "Michelle", "Matt", "Sue", "Jane", "Bob", "John", "Jennifer")

reg1<-lm(scores~hours) #Regress test scores against hours studied (always
 refer to a regression of Y against -- or on -- X)

#We saved the regression results as "reg1"; we can summarize them using
 summary(). This will tell us all the necessary information about the
 significance of the relationship between test scores and hours studied,
 including the actual values of the intercept and the beta coefficients
 along with the corresponding t and p values for each.

summary(reg1)

#Note: The p-values listed in regression summaries are generated from a
 2-tailed test.

#We can also generate a scatterplot of our data and superimpose the
 regression line that we generated:
```

```
plot(scores~hours, pch=19)
abline(reg1)

#Per Mike's question:

text(hours, scores, labels=names, cex=.6, pos=3) #Label all the data points; cex denotes font size, pos denotes position of the text

text(6, 78, labels="John", cex=.7, pos=3) #Label just one of the data points; specify the coordinates of your data point and what you want to label that point.


#We can identify the information we want from our regression object by using the dollar sign (the same one we use to identify variables within a data set) and "grab" only this data from the regression object like so:

reg1$coefficients #Returns just the values for the intercept and beta-1 coefficients
coef(reg1) #Another option that will return the coefficient values

summary(reg1)$coefficients #Returns values for the intercept and beta-1 with all accompanying information (standard errors, t scores and p-values)

#We can also store the summary of our regression and identify just that object:

sum<-summary(reg1)
sum$coefficients

sum$coefficients[,4] #Returns just the p-values for the intercept and beta-1
sum$coefficients[2,4] #Returns just the p-value for beta-1, and so on and so forth depending on which column and row you specify

sum$r.squared #Returns the value for the multiple R-squared (r-squared will always take a value between 0 and 1)
sum$adj.r.squared #Returns the adjusted-r-squared value
```

```
#The difference between multiple r-squared and adjusted r-squared doesn't
 mean much in bivariate regression. But in multivariate regression, which
 we'll cover next week, the adjusted-r-squared tries to correct for
 adding more predictors (independent variables) to your model. If you
 keep adding more and more predictors to your model, your r-squared will
 naturally improve because you're explaining more and more of the
 variation in Y -- but nevertheless, some of the explained variation in
 your model might still be due to randomness. The more predictors you
 add, the more your r-squared will increase, but this doesn't mean your
 multivariate model is necessarily a better fit for your data. The
 adjusted-r-squared statistic corrects for this problem, and adjusts
 itself according to the number of independent variables in your model.

#You can figure out how to identify the information you want (i.e., what
 should follow the dollar sign) using the attributes() command presented
 above or the str() command (str stands for "structure"):

str(sum)


# ------------------------
# EXAMINING RESIDUAL TERMS
# ------------------------

#Our regression object also stores all of our residual terms -- the
 difference between the actual observed values of our dependent variable
 (y) and our predicted values of the dependent variable (y-hat). In other
 words, for this example, our residual terms are the differences between
 the actual observed test scores of the class AND the test scores that
 our regression equation PREDICTS based on the number of hours studied.
 We can pull out the residual terms from our regression object like so:

reg1$residuals

#We have 14 data points, and thus have 14 residual terms:

length(scores)
length(reg1$residuals)
```

```r
#All our residual terms should sum to zero (or very very close to zero):

sum(reg1$residuals)

#We should also see a covariance of zero (or very very close to zero) between our residual terms and our independent variable. Our covariance should be approximately zero because we should observe no relationship between the error term and our independent variable; in other words, our residual terms should be randomly distributed on either side of the regression line:

cov(reg1$residuals,hours)

#Examining residual plots is the best way to investigate whether our assumptions about the error term -- e.g., homoskedasticity, no autocorrelation -- are intact. Residuals don't PROVE whether the assumptions are right or wrong but they help us make an educated guess. (We can add a horizontal line at y=zero to make interpretation easier.)

plot(hours, reg1$residuals, pch=19)
abline(h=0)

#As the plot demonstrates, the residuals appear to be randomly distributed on either side of the regression line, suggesting that there are no obvious problems (but this is obviously not always the case).


# -------------------------------
# BIVARIATE REGRESSION: EXAMPLES
# -------------------------------

#Let's now do a regression using real data. One major question in comparative politics is the relationship between wealth and democracy. A simple way to examine this is to look at the relationship between per-capita GDP and an index of democracy, like the Freedom House index, for a single country or region. The data we'll be using encompasses Latin American countries from 1974-present -- the data tells us the average Freedom House score for democracy and the percentage of GDP that is generated by agricultural output.
```

```
latindata<-read.csv(file.choose())
summary(latindata)

#Let's examine how GDP-per-capita varies over time, and how FH democratic indices vary over time:

par(mfrow=c(1,2))
plot(latindata$year, latindata$gdppc, pch=19)
plot(latindata$year, latindata$fh.score, pch=19)

#Examining both these plots indicates a relationship between GDP-per-capita and FH scores. To know for sure, let's perform a linear regression of the average Freedom House score for Latin America against the region's GDP per capita:

reg2<-lm(latindata$fh.score~latindata$gdppc) #Always list your dependent variable (what you are trying to explain) first, then tilda, then your independent variable (your predictor)

summary(reg2) #Here we see evidence for a strong, significant relationship between GDP per capita and FH scores.

#But we also have to check to see if any of the assumptions about our error term have been violated. Let's plot the data, and superimpose the regression line on our plot:

plot(latindata$gdppc, latindata$fh.score, pch=19)
abline(reg2)

#What OLS violation does this plot suggest? Let's take a look at the residuals plot to see the problem even more clearly:

plot(latindata$gdppc, reg2$residuals, pch=19) #Plot the independent variable against the residual terms and superimpose a horizontal line at y=0.
abline(h=0)

#We see clustering of the residual terms here, which suggests autocorrelation. As we discussed in lecture, autocorrelation occurs when there is some relationship among the error terms. There are of course
```

```
 statistical tests for autocorrelation, but eyeballing a scatterplot is
 the easiest way to tell if your data suffers from this problem: Does
 there appear to be a relationship between the residual terms, or do the
 residual terms appear to be evenly/randomly distributed on both sides of
 the regression line? Let's compare:

par(mfrow=c(1,2))
plot(hours, reg1$residuals, pch=19) #From our first example
abline(h=0)

plot(latindata$gdppc, reg2$residuals, pch=19)
abline(h=0)

#In the first plot, the residual terms appear to be randomly distributed
 around the regression line; in the second plot, we see clustering of our
 data points, which suggests that some of the residuals may be correlated
 with other residuals.



#A histogram of the residual terms also shows that the residual terms of
 our FH score ~ GDPPC regression are not normally distributed around mean
 0:

hist(reg2$residuals)

#So, we have convincing evidence that there is correlation among our
 error terms in our Latin-data regression model. What does this mean in
 real-world terms? It means that the error terms in our regression model
 y = a + bx + e are not independent. Remember, the error term "e" in a
 regression model reflects omitted variables -- variables not included in
 the model -- that might influence the dependent variable (in this case,
 the FH score). We don't assume that GDPPC can fully explain FH democracy
 scores for Latin America -- other factors also contribute to the FH
 score. These other factors are omitted variables that are encompassed by
 our error term.

#Often in time-series models, when data is measured over time, the values
 of a certain variable are very similar from one year to the next -- to
 use an obvious example, it makes sense that if life expectancy is high
 this year, there's a much greater chance that it will also be high next
 year, and so on. Thus, it's often the case that the effect of an omitted
 variable THIS year is likely to be related to the effect of an omitted
 variable NEXT year, and so on. If this correlation-across-time for our
 omitted variable(s) is true (which we can assess by looking at the plot
 of error terms against our chosen predictor variable X), then we have
 autocorrelation -- an apparent relationship among our error terms.

#What are the consequences of autocorrelation? The estimates for beta-hat
 coefficients will be accurate (i.e., unbiased), BUT the variance (and
 thus the standard error) of the beta-hat coefficient WILL be biased. We
 will tend to underestimate the standard error of beta-hat, which in turn
 means that we will be more likely to reject the null hypothesis (that
 beta-hat = 0). So autocorrelation can be potentially very dangerous for
 testing hypotheses. Time-series methods can help solve this problem.


#****************************************************


#Now let's now look at another data set -- the Mozaffar dataset on party
 systems in Africa that we examined a couple of weeks ago.

mozaffar<-read.csv(file.choose())
names(mozaffar)

#One of the hypotheses that Mozaffar and the other researchers are
 testing is that countries with more ethnic cleavages will have a greater
 number of political parties. Let's examine a regression of the effective
 number of legislative parties (mozaffar$enlp) against ethnic
 fragmentation (mozaffar$ltfragto):

reg3<-lm(enlp~ltfragto, data=mozaffar)

#Note that you don't have to specify the variable names (using $) if you
 specify to R which data set you're referring to. This can save time when
 you're writing code for multiple regressions. We would have gotten
 identical results via:

reg3<-lm(mozaffar$enlp ~ mozaffar$ltfragto)
```

```
summary(reg3)

#There appears to be a positive relationship between ethnic cleavages and
 number of political parties, significant at the .05 level. Now let's see
 if there are any apparent violations of the assumptions necessary for
 OLS:

plot(enlp~ltfragto, data=mozaffar, pch=19)
abline(reg3)

plot(reg3$residuals~ltfragto, data=mozaffar, pch=19)
abline(h=0)

#We see clustering of the error terms, indicating non-independence, or
 autocorrelation. But our plots also suggest that the variance of the
 error terms might not be constant (i.e., homoskedastic). Error variance
 depends on the value of the independent variable. In this case, variance
 seems to be larger at higher levels of ethnic fragmentation. Thus, we
 have evidence of heteroskedasticity.

#What are the consequences of heteroskedasticity? Again, the estimates
 for beta-hat coefficients will be accurate (i.e., unbiased), BUT the
 variance and, thus, the standard errors of your beta-hat estimators will
 be biased, making it more or less likely that you will find significant
 results. Under conditions of heteroskedasticity, you will not minimize
 the variance of the estimates of your relationship -- so another model
 (not OLS) would provide a better estimate of the relationship between X
 and Y.


#But let's ignore the apparent heteroskedasticity for now. Part of
 Mozaffar et al.'s argument is that the expected relationship between
 fragmentation and number of parties holds ONLY in countries where ethnic
 groups are also highly concentrated. Let's test this hypothesis by
 running separate regressions for two subsets of the data: First, let's
 see if there is a relationship between fragmentation and number of
 parties in countries where concentration is above the median level.
 Second, let's test for that relationship in countries where
 concentration is at or below the median level.
```

```
#Note: According to the codebook, mozaffar$contot is the variable for
 ethnic concentration.

reg4<-lm(mozaffar$enlp~mozaffar$ltfragto, subset = mozaffar$contot >
 median(mozaffar$contot))
summary(reg4)

reg5<-lm(enlp~ltfragto, data=mozaffar, subset = contot <= median(mozaffar
 $contot))
summary(reg5)

#Examining the summary reports of our regressions, we see that the beta-
 hat for ethnic fragmentation is almost zero in countries where ethnic
 groups are geographically dispersed (i.e., where levels of ethnic
 concentration are at or below the median, as in regression 5). But beta-
 hat is large (and the regression has a much higher r-squared) where
 ethnic groups are highly concentrated (i.e., levels of ethic
 concentration are above the median, as in regression 4).

#Let's examine the differences in the beta-hat coefficients (the strength
 of the relationship between ethnic fragmentation and number of parties)
 for the two different levels of ethnic concentration side by side:

par(mfrow=c(1,2))
plot(enlp~ltfragto, data=mozaffar, subset=contot>median(contot), add=T,
 xlab="Ethnic Fragmentation", ylab="Number of Political Parties",
 main="Relationship Under Conditions of Ethnic Concentration", pch=19)
abline(reg4)
plot(enlp~ltfragto, data=mozaffar, subset=contot<=median(contot), add=T,
 xlab="Ethnic Fragmentation", ylab="Number of Political Parties",
 main="Relationship Under Conditions of Ethnic Dispersion", pch=19)
abline(reg5)

#Now let's examine the residuals plots for each. We can't simply do
 plot(reg4$residuals~ltfragto), because the length of the residual-terms
 vector for regression 4 is of a different length than independent
 variable ltfragto:

length(reg4$residuals)
length(mozaffar$ltfragto)

#The two variables are different lengths, of course, because our
 regressions were performed on a subset of the data conditioned upon the
 median of mozaffar$contot. Now let's create two new vectors, containing
 the values of mozaffar$ltfragto that meet the respective "contot"
 conditions:

which(colnames(mozaffar)=="contot") #Column 11
fragmentation.concentrated<-mozaffar$ltfragto[which(mozaffar[,11] >
 median(mozaffar$contot))]
fragmentation.dispersed<-mozaffar$ltfragto[which(mozaffar[,11] <=
 median(mozaffar$contot))]
length(fragmentation.concentrated)
length(fragmentation.dispersed)

#Now we can create residuals plots for both regressions, since the
 residual-terms vector and the vector of relevant values of "ltfragto"
 are the same length:

par(mfrow=c(1,2))
plot(reg4$residuals~fragmentation.concentrated, add=T, xlab="Ethnic
 Fragmentation", ylab="Residuals", main="Relationship Under Conditions of
 Ethnic Concentration", pch=19)
abline(h=0)
plot(reg5$residuals~fragmentation.concentrated, add=T, xlab="Ethnic
 Fragmentation", ylab="Residuals", main="Relationship Under Conditions of
 Ethnic Dispersion", pch=19)
abline(h=0)

#What assumptions appear to be violated here?
```