```
#multiple regression of diamonds data

diamonds = read.csv("diamonds.csv", header = T)
head(diamonds)

#exploratory data analysis
plot(y = diamonds$Price, x = diamonds$Carats, xlab = "Carats", ylab = "Price")
boxplot(Price~Color, data = diamonds, ylab = "Price", xlab = "Color")
boxplot(Price~Clarity, data = diamonds, ylab = "Price", xlab = "Clarity")
boxplot(Price~Certification, data = diamonds, ylab = "Price", xlab = "Certification")

#suggest nonconstant variance - try log(price).  quadratic relationship with carats: try
carats^2
#these suggest some differences in prices across the categorical variables, but each
#all plots do not account for the effects of other variables (which is why we do multiple
regression)

#make log of price
diamonds$Logprice = log(diamonds$Price)

#it helps interpretation to substract the mean from numerical predictors
diamonds$CaratsCent = diamonds$Carats - mean(diamonds$Carats)

#make carats squared
diamonds$Carats2 = diamonds$CaratsCent^2


#let's continue with the EDA. lattice is a great graphics library.
#if you have not installed it, type  install.packages("lattice")
#load it in to R each time you start R
library(lattice)

#you also can make these plots with ggplot -- I will post those commands later.

#let's look for interaction effects using trellis plots.  Let's use LogPrice on Carats by
each color
xyplot(Logprice~Carats | Color, data = diamonds)

# slope in each plot is similar, so no strong evidence of interaction between Carats and
Color

#repeat for clarity and certification
xyplot(Logprice~Carats | Clarity, data = diamonds)
xyplot(Logprice~Carats | Certification, data = diamonds)

#same result: similar slopes in each panel, so no strong evidence of interactions with Carats

#we also can examine trellis plots for box plots to see if there are interactions among
categoical vars

bwplot(Logprice~Color | Clarity, data = diamonds)

#for any clarity category, the color box plots have a reasonably similar ordering.
#So, no strong evidence of interaction between Clarity and Color
#repeat for Color and Certification, and Clarity and Certification (we did not do this in
class)

#### fitting the multiple regression  ####

#so, our candidate model based on EDA is Logprice ~ Carats + Carats^2 + Color + Clarity +
Certification + error

#we need to make dummy variables for all the categorical data

#sample size
```

```
n = nrow(diamonds)

#create series of indicator variables for color
diamonds$colorD = rep(0, n)
diamonds$colorD[diamonds$Color == "D"] = 1

diamonds$colorE = rep(0, n)
diamonds$colorE[diamonds$Color == "E"] = 1

diamonds$colorF = rep(0, n)
diamonds$colorF[diamonds$Color == "F"] = 1

diamonds$colorG = rep(0, n)
diamonds$colorG[diamonds$Color == "G"] = 1

diamonds$colorH = rep(0, n)
diamonds$colorH[diamonds$Color == "H"] = 1

diamonds$colorI = rep(0, n)
diamonds$colorI[diamonds$Color == "I"] = 1

#create series of indicator variables for clarity
diamonds$clarityIF = rep(0, n)
diamonds$clarityIF[diamonds$Clarity == "IF"] = 1

diamonds$clarityVS1 = rep(0, n)
diamonds$clarityVS1[diamonds$Clarity == "VS1"] = 1

diamonds$clarityVS2 = rep(0, n)
diamonds$clarityVS2[diamonds$Clarity == "VS2"] = 1

diamonds$clarityVVS1 = rep(0, n)
diamonds$clarityVVS1[diamonds$Clarity == "VVS1"] = 1

diamonds$clarityVVS2 = rep(0, n)
diamonds$clarityVVS2[diamonds$Clarity == "VVS2"] = 1

#create series of indicator variables for certification
diamonds$certGIA = rep(0, n)
diamonds$certGIA[diamonds$Certification == "GIA"] = 1

diamonds$certIGI = rep(0, n)
diamonds$certIGI[diamonds$Certification == "IGI"] = 1

diamonds$certHRD = rep(0, n)
diamonds$certHRD[diamonds$Certification == "HRD"] = 1


#Let's fit a multiple linear regression model without transformations, just to see the impact

regPonAll = lm(Price~CaratsCent + colorE + colorF + colorG + colorH + colorI + clarityVS1 +
clarityVS2 + clarityVVS1 + clarityVVS2 + certIGI + certHRD, data = diamonds)

#to view results, see
summary(regPonAll)

#what happens if we make a different color (color F in this case) the baseline?
regPonAllf = lm(Price~CaratsCent + colorD + colorE + colorG + colorH + colorI + clarityVS1 +
clarityVS2 + clarityVVS1 + clarityVVS2 + certIGI + certHRD, data = diamonds)
summary(regPonAllf)
#notice that the dummy variables for all the color categories changed, as did the intercept,
but nothing else changed.

### model diagnostics!
#to do checks of residuals versus each predictor, use
plot(regPonAll$resid, x=diamonds$CaratsCent, ylab = "Residuals")
```

```
abline(0,0)
boxplot(regPonAll$resid~diamonds$Color, ylab = "Residuals")
boxplot(regPonAll$resid~diamonds$Clarity, ylab = "Residuals")
boxplot(regPonAll$resid~diamonds$Certification, ylab = "Residuals")

#problems show up in plots.  Let's repeat based on our EDA suggested model.

reglogPonAll = lm(Logprice~CaratsCent + Carats2 + colorE + colorF + colorG + colorH + colorI
+ clarityVS1 + clarityVS2 + clarityVVS1 + clarityVVS2 + certIGI + certHRD, data = diamonds)


#to view results, see
summary(reglogPonAll)

#to do checks of residuals versus each predictor, use
plot(reglogPonAll$resid, x=diamonds$CaratsCent, ylab = "Residuals")
abline(0,0)
boxplot(reglogPonAll$resid~diamonds$Color, ylab = "Residuals")
boxplot(reglogPonAll$resid~diamonds$Clarity, ylab = "Residuals")
boxplot(reglogPonAll$resid~diamonds$Certification, ylab = "Residuals")

#No real patterns, so things look good!

#confidence intervals for coefficients
confint(reglogPonAll)

#exponentiate to get interpretation as ratio of medians
exp(confint(reglogPonAll))

#commands for predicting price of a new diamond, as well as visualing the relationship
between price and carats, at end of script.

### coding tip for dummy variables
#a quicker way to let R make the dummy variables uses the as.factor( ) command.

reglogPonAll2 = lm(Logprice~CaratsCent + Carats2 + as.factor(Color) + as.factor(Clarity) +
as.factor(Certification), data = diamonds)
summary(reglogPonAll2)

#the quick way is especially useful if you don't care which is the baseline, and in initial
stages of model fitting where you are trying
#to find a good model first before worrying about interpretation.

#if you want to control the level with the as.factor way, use the following.
#suppose you want to change the reference category to Color == G.  Then, use
reglogPonAll2 = lm(Logprice~CaratsCent + Carats2 + relevel(as.factor(Color), ref = "G") +
as.factor(Clarity) + as.factor(Certification), data = diamonds)

#let's look at the results
summary(reglogPonAll2)


### nested f test to see if a whole set of dummy variables are significant

#let's do a nested F test to see if certification as a whole is important
#first we fit the model without certification

reglogPonAllNocert = lm(Logprice~CaratsCent + Carats2 + colorE + colorF + colorG + colorH +
colorI + clarityVS1 + clarityVS2 + clarityVVS1 + clarityVVS2, data = diamonds)
anova(reglogPonAll, reglogPonAllNocert)

#p value is 0.088.  This is borderline evidence.  The organization that certified the diamond
may well have an effect on the price, although
#we don't have strong evidence to rule out the possibility that it does not.
```

```
##### Interactions  #####

#use lattice plots to search for possible interaction effects -- see commands at the
beginning of script.

#after fitting regression, look for interaction effects in residual plots.
#if you see different patterns in residuals vs. some predictor based on values of another
predictor, try an interaction effect in the regression model

#let's look for interactions in the diamonds regression.

xyplot(reglogPonAll$residual~ diamonds$CaratsCent | diamonds$Color)
xyplot(reglogPonAll$residual~ diamonds$CaratsCent | diamonds$Clarity)
xyplot(reglogPonAll$residual~ diamonds$CaratsCent | diamonds$Certification)

#not much. maybe we can try the interaction of Carats with Clarity.

##fitting interaction effects:  two ways
#the quick way is to add a * between the two variables that you want to interact

reglogPonAllInt = lm(Logprice ~ CaratsCent*as.factor(Clarity) + Carats2*as.factor(Clarity) +
as.factor(Color) + as.factor(Certification), data = diamonds)

#a more concise way to write this is:  reglogPonAllInt = lm(Logprice ~ (CaratsCent +
Carats2)*as.factor(Clarity) + as.factor(Color) + as.factor(Certification), data = diamonds)

#the long way to include an interaction is to create a new variable equal to the product of
the two variables you want to interact.  then include it in the regression.
#when interacting with a categorical variable, create interaction variables for all levels of
the categorical variables (except the baseline)

#let's look at the results
summary(reglogPonAll2)

#see the R script called "interactions.txt" for interpretations of these results.

#let's do a nested F test to see if the entire set of interactions is useful (this test is
also on the interactions.txt script)

anova(reglogPonAll, reglogPonAllInt)

#Set of interactions appears to be a useful predictor. However, we note that the interaction
coefficients only are
#significant for VS2, which confirms what we saw in the xyplot for the residuals.  I don't
know of any reason why
#the relationship of carats should differ for VS2 diamonds but not others.  So, I am
skeptical whether or not this
#set of interactions is "too specific" to these particular diamonds.  That is, it is helping
with the fit of these particular prices
#but might not be significant if we had another set of data.  We'd want some expert advice to
decide whether or not to believe this interaction effect
#is practically meaningful.


####  plot for visualizing relationship of carats with logprice
#make several values of carat deviations from the average
caratchanges = seq(from = -.5, to = .5, by = .05)

#compute the value of B1*caratchanges + B2*caratchanges^2

#list the coefficients
coef(reglogPonAll)

#we take the second and third values of the coefficient vector to get B1 and B2
avglogp = coef(reglogPonAll)[2]*caratchanges + coef(reglogPonAll)[3]*caratchanges^2
```

```
#plot them on the log scale
plot(y = avglogp, x = caratchanges, xlab = "Change in Carats", ylab = "Change in average log
price")

#not especially interpretable, so transform to percent change in medians (recall the
interpretation of logs and medians)
plot(y = exp(avglogp), x = caratchanges, xlab = "Change in Carats", ylab = "Percent change in
median price")


#### predicting a new  diamond price


#how about making a prediction for a new diamond with 1 carat, color g, clarity vvs2, and GIA
certification?
#we need a new dataset with the same format as the diamonds data.  So, let's copy one row of
diamond to initialize our new dataset

newdata = diamonds[1,]

#now we replace each value, ignoring anything to do with price or logprice since we won't use
those in the prediction.

#start by replacing the value of carats. Since we have only one row as prediction
newdata$Carats[1] = 1

#now to Color
newdata$Color[1] = "G"

#Clarity
newdata$Clarity[1] = "VVS2"

#Certification
newdata$Certification[1] = "GIA"

#ignore columns 5 and 6, which deal with price and logprice.

#Caratscentered
newdata$CaratsCent[1] = newdata$Carats[1] - mean(diamonds$Carats)

#carats centered squared
newdata$Carats2[1] = newdata$CaratsCent[1]^2

#zero out the indicators for all colors but D
newdata$colorD[1] = 0
newdata$colorE[1] = 0
newdata$colorF[1] = 0
newdata$colorG[1] = 1
newdata$colorH[1] = 0
newdata$colorI[1] = 0

#zero out all the indicators for all clarity by VVS2
newdata$clarityIF[1] = 0
newdata$clarityVS1[1] = 0
newdata$clarityVS2[1] = 0
newdata$clarityVVS1[1] = 0
newdata$clarityVVS2[1] = 1

#zero out all certification but GIA
newdata$certGIA[1] = 1
newdata$certIGA[1] = 0
newdata$certHRD[1] = 0

#look at newdata to make sure it is what you actually wanted!
newdata
```

```
#now the prediction interval for the logwage
predlogprice = predict(reglogPonAll, newdata, interval = "prediction")
predlogprice
      fit      lwr     upr
#1 9.221311 9.102371 9.34025

#let's exponentiate to get on a scale for predicting the price

exp(predlogprice)
       fit      lwr      upr
#1 10110.31 8976.555 11387.25
```