Lab 6

First, let's re-download the data from last week. Remember that this data includes a sample of people who ran the Cherry Blossom 10 miler in 2010.

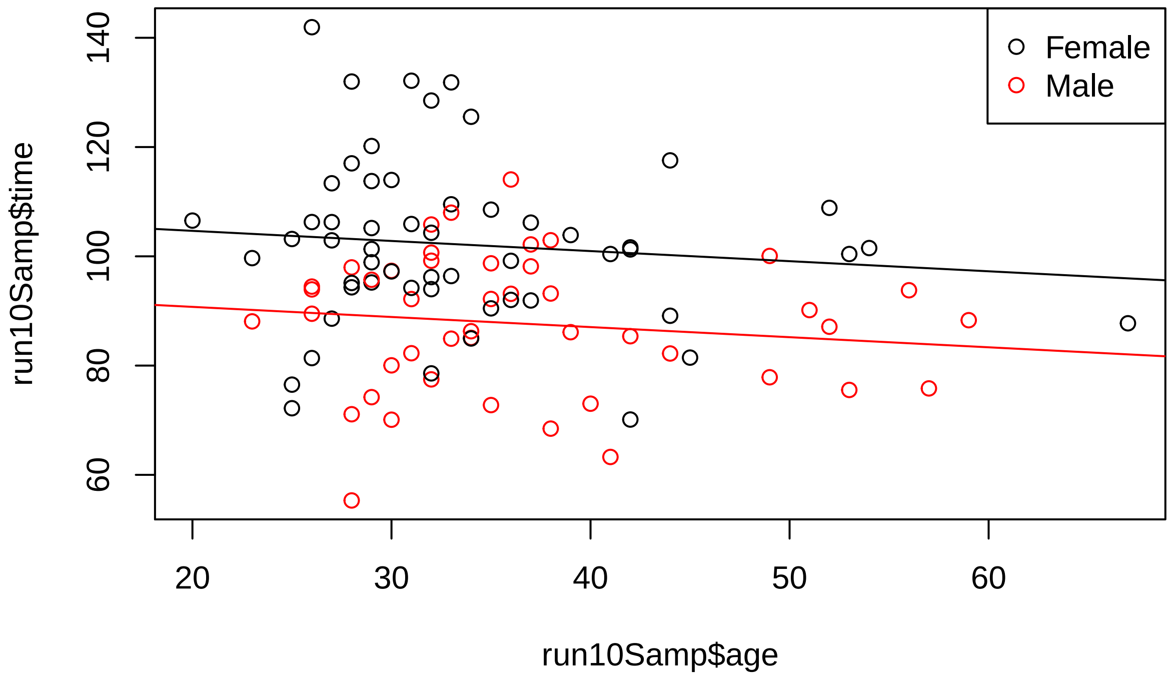
```
require(openintro)
data("run10Samp")
```

Last week, we ran a regression of time (DV) on age (IV) and gender (IV), where gender was a categorical variable with two possible values: male or female. This produced two lines showing the relationship between time and age for males and females, where the lines were allowed to have different intercepts:

```
lm1 = lm(time ~ age + gender, data=run10Samp)
summary(lm1)
```

```
##
## Call:
## lm(formula = time ~ age + gender, data = run10Samp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.956  -8.417   0.262   8.509  38.389
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 108.3630     5.7061  18.991  < 2e-16 ***
## age          -0.1851     0.1599  -1.157     0.25
## genderM     -13.9157     2.8698  -4.849 4.72e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.06 on 97 degrees of freedom
## Multiple R-squared:  0.2216, Adjusted R-squared:  0.2056
## F-statistic: 13.81 on 2 and 97 DF,  p-value: 5.284e-06
```

```
plot(run10Samp$age, run10Samp$time, col=as.numeric(run10Samp$gender))
legend("topright", c("Female", "Male"), col=c(1,2), pch=1)
abline(summary(lm1)$coef[1], summary(lm1)$coef[2])
abline(summary(lm1)$coef[1] + summary(lm1)$coef[3], summary(lm1)$coef[2], col=2)
```



Now suppose that we want to allow these lines to have different slopes. We can add an interaction term between gender and age as follows. The model is now $time = \beta_0 + \beta_1 * age + \beta_2 * gender + \beta_3 * age * gender$. In the output, the coefficient for beta_3 is shown as age:genderM. Note that $age * gender$ does the same thing as writing out the full functional form.

```
lm2 = lm(time ~ age * gender, data=run10Samp)
summary(lm2)
```

```
##
## Call:
## lm(formula = time ~ age * gender, data = run10Samp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.191  -8.594   0.809   8.278  37.798
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 110.9581     7.6539  14.497   <2e-16 ***
## age          -0.2622     0.2203  -1.190   0.2369
## genderM     -19.7203    11.7152  -1.683   0.0956 .
## age:genderM   0.1644     0.3217   0.511   0.6104
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.12 on 96 degrees of freedom
## Multiple R-squared:  0.2237, Adjusted R-squared:  0.1995
## F-statistic: 9.223 on 3 and 96 DF,  p-value: 2.026e-05
```

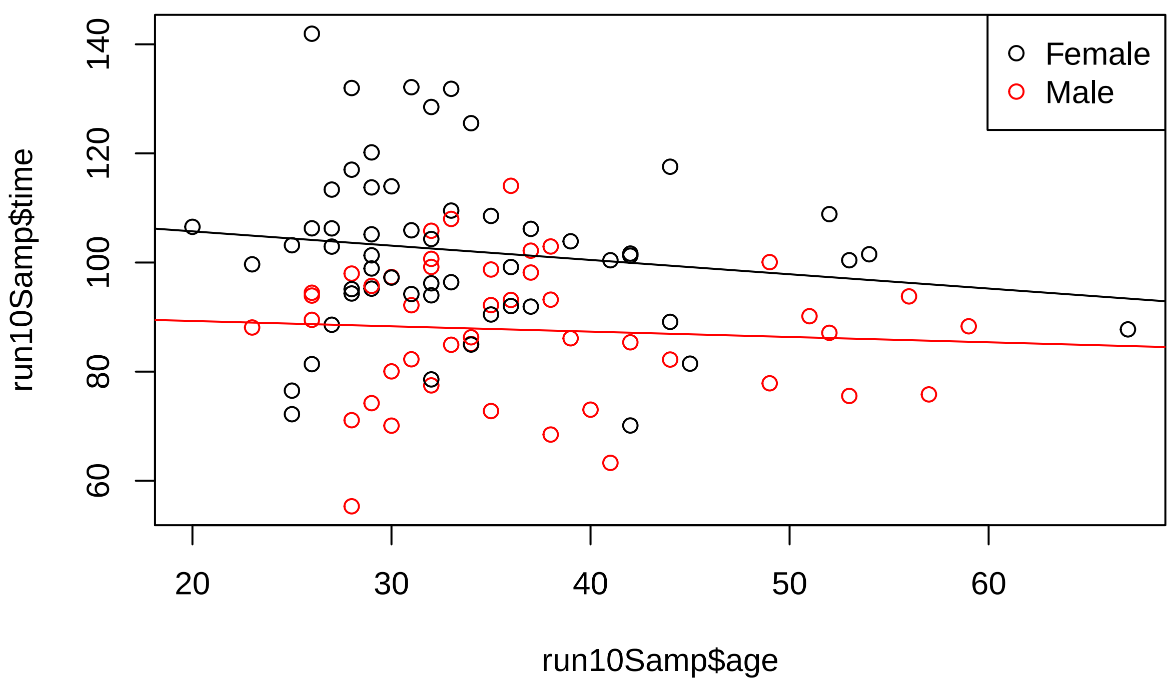
```
#same thing
lm3 = lm(time ~ age + gender + age*gender, data=run10Samp)
summary(lm3)
```

```
##
```

```
## Call:
## lm(formula = time ~ age + gender + age * gender, data = run10Samp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.191  -8.594   0.809   8.278  37.798
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 110.9581     7.6539  14.497   <2e-16 ***
## age          -0.2622     0.2203  -1.190   0.2369
## genderM     -19.7203    11.7152  -1.683   0.0956 .
## age:genderM   0.1644     0.3217   0.511   0.6104
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.12 on 96 degrees of freedom
## Multiple R-squared:  0.2237, Adjusted R-squared:  0.1995
## F-statistic: 9.223 on 3 and 96 DF,  p-value: 2.026e-05
```

Now let's think about the lines that we get when male = 0 vs. male = 1. When male=0, the line is $time = 110.96 - .26 * age - 19.72 * genderM + .16 * age * genderM = 110.96 - .26 * age$. Now, when male=1, the line is $time = 110.96 - .26 * age - 19.72 * genderM + .16 * age * genderM = 110.96 - .26 * age - 19.72 * 1 + .16 * age * 1 = (110.96 - 19.72) + (-.26 + .16) * age$. We can graph this as follows (note that the lines for male and female have different intercepts AND different slopes now):

```r
plot(run10Samp$age, run10Samp$time, col=as.numeric(run10Samp$gender))
legend("topright", c("Female", "Male"), col=c(1,2), pch=1)
abline(summary(lm2)$coef[1], summary(lm2)$coef[2])
abline(summary(lm2)$coef[1] + summary(lm2)$coef[3], summary(lm2)$coef[2]+summary(lm2)$coef[4], col=2)
```



We can also use the margins package to look at the marginal (i.e., average) "effect" (i.e., coefficient on) of age when gender=M and gender=F. The values in the age column are just what we calculated for the slopes of the lines above. The values in the genderF column are the average coefficients for "genderF" across all ages in the dataset.

```
#install.packages("margins")
require(margins)
margins(lm2, at = list(gender = c("M", "F")))
```

```
##  at(gender)      age genderF
##           M -0.09774   13.96
##           F -0.26217   13.96
```

```
#this is where those values for genderF are coming from:
-summary(lm2)$coef[3]-(summary(lm2)$coef[4]*mean(run10Samp$age))
```

```
## [1] 13.95703
```

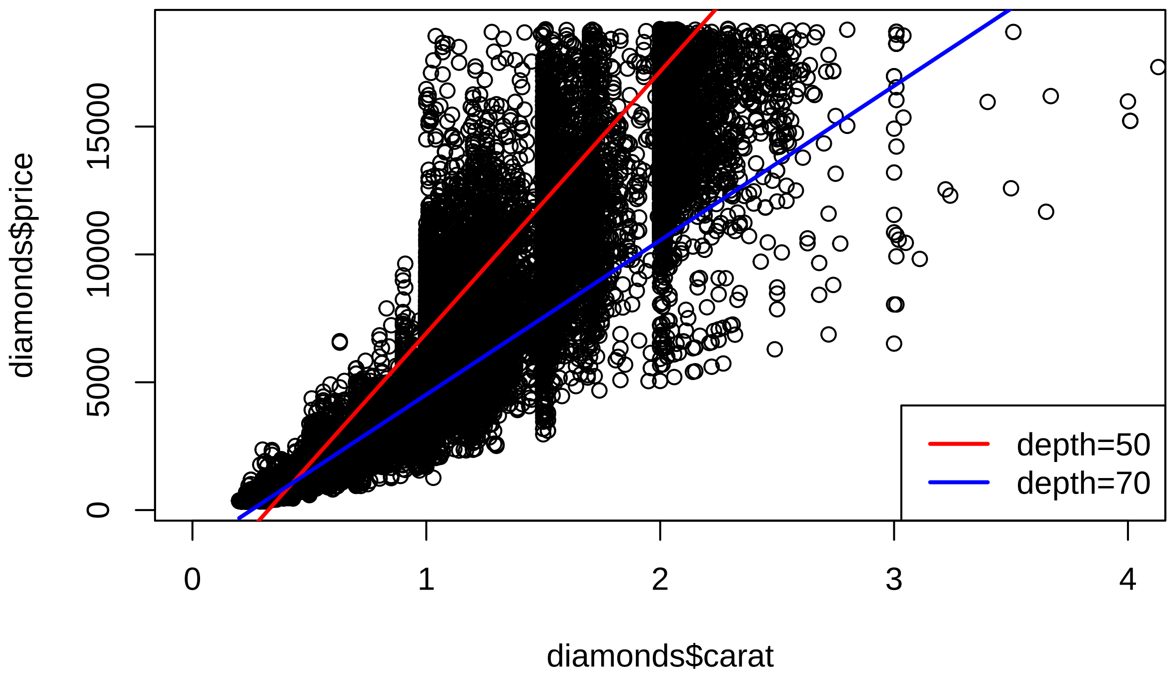
We can also add interaction terms between continuous variables, but then we essentially get infinitely many lines.
Lets load the diamonds dataset from ggplot2, which has information for diamond prices and various characteristics. We can estimate price by regressing on carat, depth, and an interaction between carat and depth.

```
require(ggplot2)
data(diamonds)
lm_diamonds = lm(price ~ carat * depth, data=diamonds)
summary(lm_diamonds)
```

```
##
## Call:
## lm(formula = price ~ carat * depth, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15039.2   -799.3    -18.1    539.6  12666.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7823.738    592.049 -13.215   <2e-16 ***
## carat       20742.600    567.672  36.540   <2e-16 ***
## depth          90.043      9.588   9.391   <2e-16 ***
## carat:depth  -210.075      9.187 -22.868   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1534 on 53936 degrees of freedom
## Multiple R-squared:  0.8521, Adjusted R-squared:  0.8521
## F-statistic: 1.036e+05 on 3 and 53936 DF,  p-value: < 2.2e-16
```

Now, if we want to visualize the relationship between these variables, it is harder because we used two continuous covariates. But, we could show the relationship between carat and price for different depths (note: think about how we could derive the equations for these lines: just plug in the given value for depth!):

```
carats = seq(min(diamonds$carat), max(diamonds$carat), .05)
preds_50 = predict(lm_diamonds, data.frame(depth = rep(50,length(carats)), carat=carats))
preds_70 = predict(lm_diamonds, data.frame(depth = rep(70,length(carats)), carat=carats))
plot(diamonds$carat, diamonds$price, xlim=c(0,4))
lines(carats, preds_50, col=2, lwd=2)
lines(carats, preds_70, col=4, lwd=2)
legend("bottomright", c("depth=50", "depth=70"), col=c(2,4), lty=1, lwd=2)
```



We can also use margins to calculate average coefficients across all possible values of the other covariates

```
margins(lm_diamonds)
```

```
##   carat   depth
##    7771 -77.58
```

```
#how we can get these values by hand:
summary(lm_diamonds)$coef[2]+summary(lm_diamonds)$coef[4]*mean(diamonds$depth)
```

```
## [1] 7770.573
```

```
summary(lm_diamonds)$coef[3]+summary(lm_diamonds)$coef[4]*mean(diamonds$carat)
```

```
## [1] -77.58424
```

Finally, let's talk about what happens when we include categorical variables in a regression that have more than two categories.

Lets start with a simple linear model with one continuous and one categorical variable. In this case we will use the predictor 'x' and the cut to predict price. Let us first take a look at the cut column:

```
levels(diamonds$cut)
```

```
## [1] "Fair"      "Good"      "Very Good" "Premium"   "Ideal"
```

Here we can see that there are 5 levels to this variable in some order. The first one in the output is "Fair". This tells use that R will read this group as the reference group.

Why do we need a reference category? Remember that to find the beta hats using matrix algebra, we'll need to multiply the transpose of the X matrix with the X matrix, and then we need to take the inverse of this square matrix. (note: the matrix below is basically how this gets coded in R when you run the linear regression)

```
lm_diamonds2 = lm(price~cut, data=diamonds)
model.matrix(lm_diamonds2)[1:5,]
```

```
##   (Intercept) cutGood cutVery Good cutPremium cutIdeal
## 1           1       0            0          0        1
## 2           1       0            0          1        0
## 3           1       1            0          0        0
## 4           1       0            0          1        0
## 5           1       1            0          0        0
```

If we add in the reference group, our matrix columns are no longer linearly independent, which means the matrix is not full rank. We cannot take the inverse of a matrix that is not full rank, so we won't be able to solve for the beta hats.

If we regress price on cut we get the following output:

```r
lm_diamonds2 = lm(price~cut, data=diamonds)
summary(lm_diamonds2)
```

```
##
## Call:
## lm(formula = price ~ cut, data = diamonds)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4258  -2741  -1494   1360  15348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4358.76      98.79  44.122  < 2e-16 ***
## cutGood       -429.89     113.85  -3.776 0.000160 ***
## cutVery Good  -377.00     105.16  -3.585 0.000338 ***
## cutPremium     225.50     104.40   2.160 0.030772 *
## cutIdeal      -901.22     102.41  -8.800  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3964 on 53935 degrees of freedom
## Multiple R-squared:  0.01286,    Adjusted R-squared:  0.01279
## F-statistic: 175.7 on 4 and 53935 DF,  p-value: < 2.2e-16
```

The intercept represents the mean price of diamonds with cut=fair. All of the other categories basically became their own binary variable, and the coefficient on each one now represents the difference in average price between cut=Fair and the given category.

Challenge to discuss: what if we include an interaction between cut and carat. Now how do we interpret all these coefficients? Try writing out the equation (we'll do this on the board).

```r
lm_diamonds3 = lm(price~cut*carat, data=diamonds)
summary(lm_diamonds3)
```

```
##
## Call:
## lm(formula = price ~ cut * carat, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14878.3   -793.0    -23.0    546.3  12706.2
##
## Coefficients:
```

```
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)        -1839.07      84.35 -21.802  < 2e-16 ***
## cutGood             -583.65      95.77  -6.094 1.11e-09 ***
## cutVery Good        -578.59      88.73  -6.521 7.06e-11 ***
## cutPremium          -540.83      88.12  -6.137 8.46e-10 ***
## cutIdeal            -461.30      86.57  -5.329 9.93e-08 ***
## carat               5924.50      72.31  81.933  < 2e-16 ***
## cutGood:carat       1555.14      86.30  18.021  < 2e-16 ***
## cutVery Good:carat  2011.48      78.16  25.737  < 2e-16 ***
## cutPremium:carat    1883.26      76.43  24.641  < 2e-16 ***
## cutIdeal:carat      2267.90      76.05  29.820  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1498 on 53930 degrees of freedom
## Multiple R-squared:  0.8591, Adjusted R-squared:  0.859 
## F-statistic: 3.653e+04 on 9 and 53930 DF,  p-value: < 2.2e-16
```