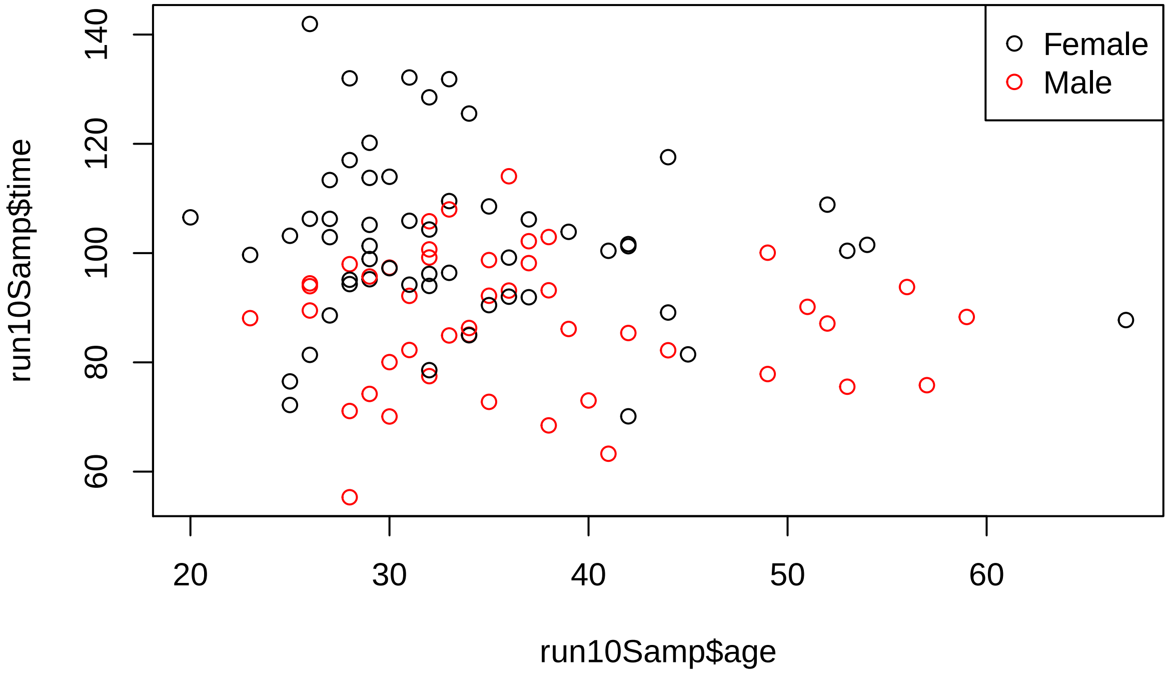
Lab 5

First, let's download some data. This data includes a sample of people who ran the Cherry Blossom 10 miler in 2010.

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

To investigate the relationship between time (in minutes), age, and gender we can make a scatter plot of the data and color code by gender:

```
plot(run10Samp$age, run10Samp$time, col=as.numeric(run10Samp$gender))
# note: use levels(run10Samp$gender) to figure out what values correspond to 1s vs 2s after as.numeric(
legend("topright", c("Female", "Male"), col=c(1,2), pch=1)
```



Based on this plot, if we want to predict race times, there is some evidence that it might be useful to use both age and gender. In order to add gender to the regression, we have some options:

```
# note that gender is a factor variable
class(run10Samp$gender)
```

```
## [1] "factor"
```

```
# note that there are two levels: F and M; F comes first, so it is the reference group
levels(run10Samp$gender)
```

```
## [1] "F" "M"
```

```
# We could recode the factor with integer values, but this is not necessary
# (and will yield a diff result for >2 categories)
# Can be useful for changing the ref group/order though
# If >2 categories, can do this and then use as.factor() to turn it back to a factor with new order

# Changes variable from a factor to numeric, with female=0, male=1
# Saves to new column called gender_numMaleRef
run10Samp$gender_numMaleRef = as.numeric(run10Samp$gender)
#Now: we can change the coding if we want:
run10Samp$gender_numMaleRef[which(run10Samp$gender=="F")] = 1 #change so that female=1, and
run10Samp$gender_numMaleRef[which(run10Samp$gender=="M")] = 0 #male=0

# Other ways to recode as numerics:
# Use ifelse:
#does the same as as.numeric, but could switch the 0 and 1:
run10Samp$gender_numFemRef = ifelse(run10Samp$gender == "F", 0, 1)


# We can also change ref group directly using relevel in stats
require(stats)
run10Samp$gender_factorMaleRef = relevel(run10Samp$gender, ref = "M")
levels(run10Samp$gender_factorMaleRef) #check that the order changed
```

```
## [1] "M" "F"
```

```
#Another note: if the variable was originally coded as a numeric or integer,
#we can use the as.factor() function to re-code it as a factor
run10Samp$gender_factorMaleRef2 = as.factor(run10Samp$gender_numMaleRef)
```

Now, we can run the regression with lm, specifying that we want to include age and gender as covariates. Let's see how including gender as a categorical vs. numerical variable with different reference groups changes the results. Note that we get basically the same results each time, but they are represented in slightly different ways. More on this later when we talk about categorical variables with more than 2 categories.

```
# with gender as a factor variable, female is the reference group
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
```

```
# with gender as numeric: female=0, male=1
lm2 = lm(time ~ age + gender_numFemRef, data=run10Samp)
summary(lm2)
```

```
##
## Call:
## lm(formula = time ~ age + gender_numFemRef, 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
## gender_numFemRef -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
```

```
# with gender as a factor variable, male is the reference group
lm3 = lm(time ~ age + gender_factorMaleRef2, data=run10Samp)
summary(lm3)
```

```
##
## Call:
## lm(formula = time ~ age + gender_factorMaleRef2, 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)             94.4472     6.2403  15.135  < 2e-16 ***
## age                     -0.1851     0.1599  -1.157     0.25
## gender_factorMaleRef21  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
```

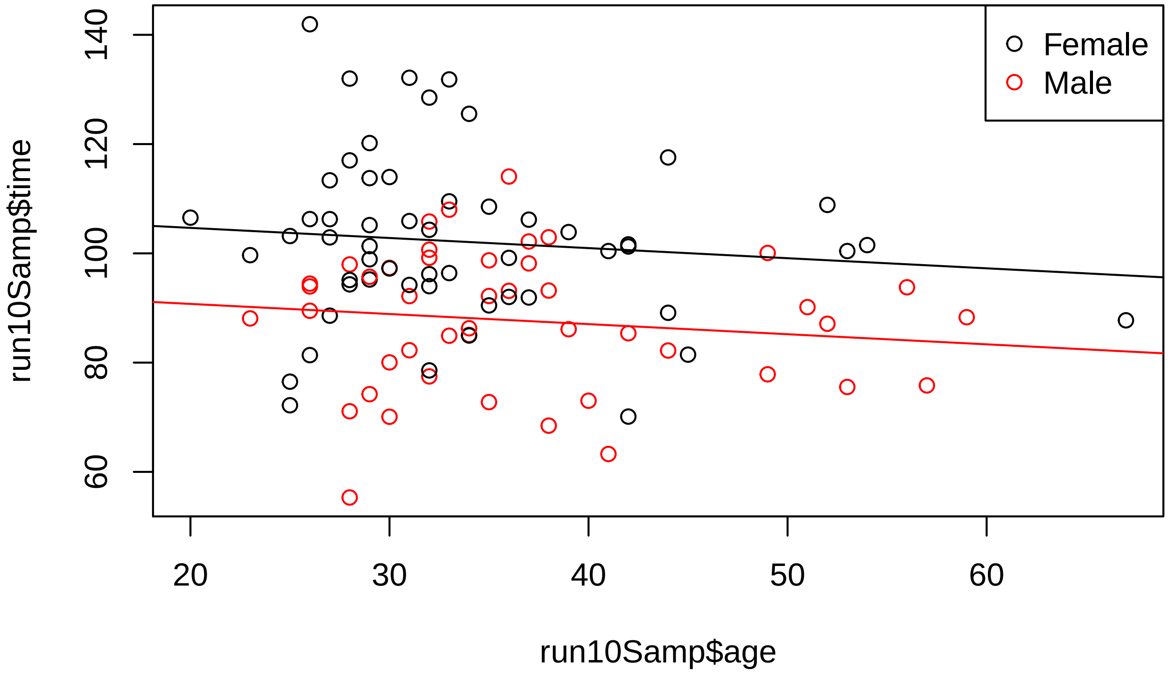
```
# with gender as a numeric variable, male=0 and female=1
lm3 = lm(time ~ age + gender_numMaleRef, data=run10Samp)
summary(lm3)
```

```
##
## Call:
## lm(formula = time ~ age + gender_numMaleRef, 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)        94.4472     6.2403  15.135  < 2e-16 ***
## age                -0.1851     0.1599  -1.157     0.25
## gender_numMaleRef  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
```

How do we interpret this?
We say that, among individuals of the same gender, we expect runners who are one year older to have finishing times that are .185 minutes shorter on average. We can also plot the lines representing this model (although this gets more complex for continuous covariates and we would need a 3D space to represent these relationships). Think about how we got the equations for these lines and why they make sense!

```
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)
```



Based on this model, what is the predicted average finishing time of 30 year old males? 30 year old females?

```
summary(lm1)$coef[1] + summary(lm1)$coef[2]*30 #female
```

```
## [1] 102.8112
```

```
summary(lm1)$coef[1] + summary(lm1)$coef[3] + summary(lm1)$coef[2]*30 #male
```

```
## [1] 88.8955
```

```
#or:
predict(lm1, data.frame(age = 30, gender="F"))
```

```
##        1
## 102.8112
```

```r
predict(lm1, data.frame(age = 30, gender="M"))
```

```
##       1
## 88.8955
```

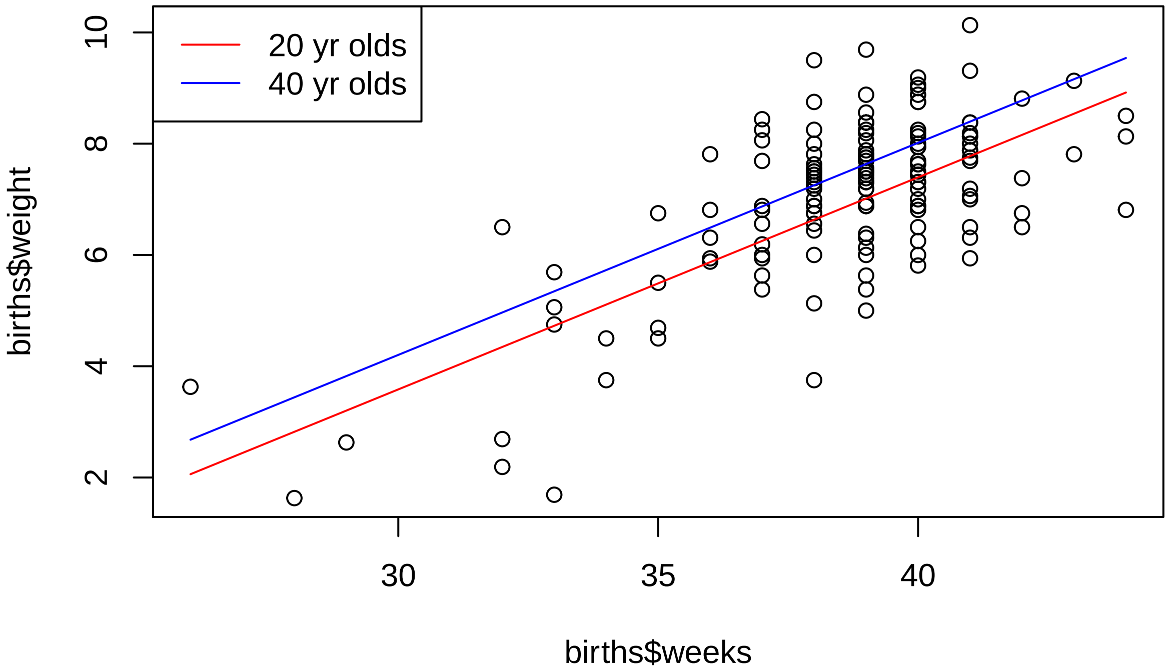
Lets now download a new dataset! This dataset includes birth information for 150 babies born in NC. Suppose that we want to predict the birthweight of these babies using the mother's age and weeks of pregnancy as covariates. We can do this as follows:

```r
data("births")
lm_births = lm(weight ~ mAge + weeks, data=births)
summary(lm_births)
```

```
## 
## Call:
## lm(formula = weight ~ mAge + weeks, data = births)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -3.00703 -0.63549  0.04663  0.64072  2.95998 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -8.47135    1.35853  -6.236 4.52e-09 ***
## mAge         0.03100    0.01437   2.158   0.0326 *  
## weeks        0.38117    0.03254  11.715  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.083 on 147 degrees of freedom
## Multiple R-squared:  0.4837, Adjusted R-squared:  0.4767 
## F-statistic: 68.87 on 2 and 147 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 weeks and birthweight for mothers of a particular age:

```r
weeks = seq(min(births$weeks), max(births$weeks), 1)
preds_20 = predict(lm_births, data.frame(mAge = rep(20,length(weeks)), weeks=weeks))
preds_40 = predict(lm_births, data.frame(mAge = rep(40,length(weeks)), weeks=weeks))
plot(births$weeks, births$weight)
lines(weeks, preds_20, col=2)
lines(weeks, preds_40, col=4)
legend("topleft", c("20 yr olds", "40 yr olds"), col=c(2,4), lty=1)
```



We can also predict birthweight of a baby born at a particular number of weeks to a mother who is a particular age. For example, let's predict birthweight of babies born at 40 weeks to 30 year old mothers:

```
# By hand:
summary(lm_births)$coef[1] + summary(lm_births)$coef[2]*30 + summary(lm_births)$coef[3]*40
```

```
## [1] 7.705366
```

```
# Using predict:
predict(lm_births, data.frame(mAge = 30, weeks=40))
```

```
##        1
## 7.705366
```

Note: the last portion of this lab can be done/discussed using Ying's simulation tool!

Simulation tool: https://a3sr.shinyapps.io/QM_Regression_Lab_/

Finally, this week, we'll look at the hypothesis tests that we perform in linear regression. Remember that for each covariate, we perform a hypothesis test of whether the coefficient is 0 (null) or not 0 (alternative). We can think about two types of errors that we might make: Type I error means that we reject the null when the null is true. Type II error means that we do not reject the null when the null is false. For the purposes of this exercise, we're going to simulate a situation where one beta coefficient is 0 and another coefficient is not 0. Then we can estimate the type I and type II error rates:

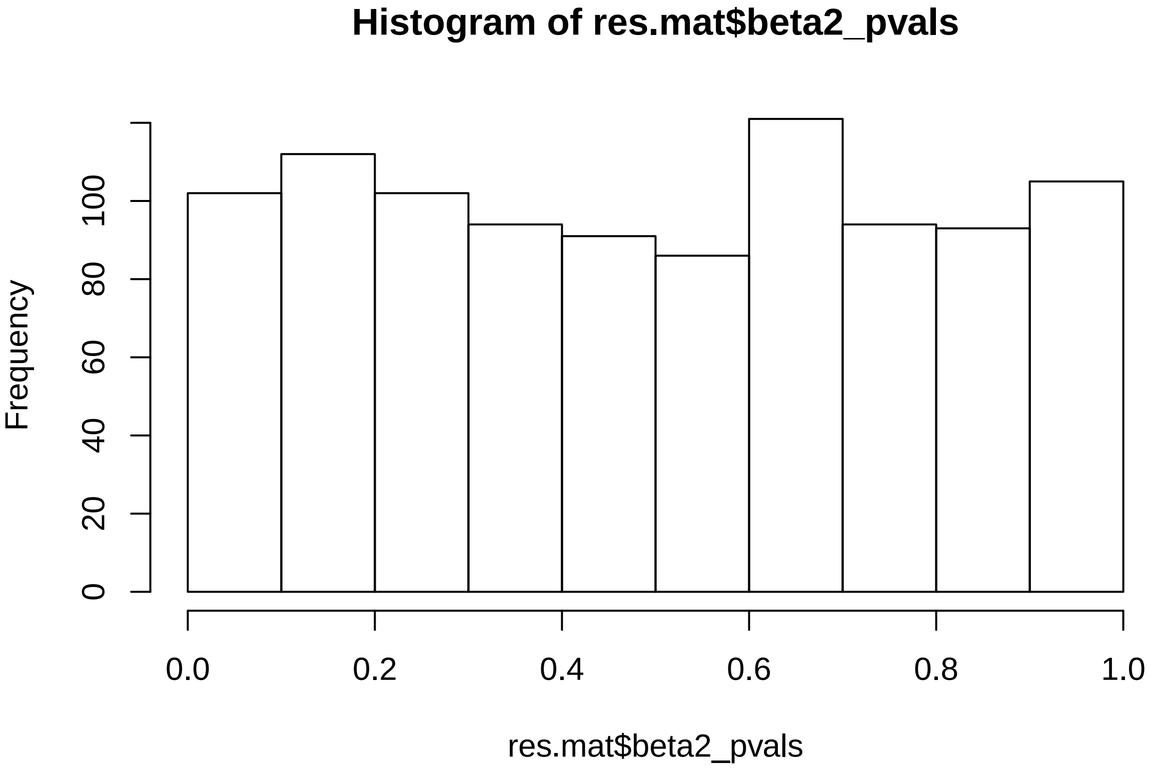
```
set.seed(123)
beta0 <-1
beta1 <- 0.2
beta2 <- 0

n<-100
nsim <- 1000

res.mat <- data.frame(beta1_pvals=rep(NA, nsim),
                      beta2_pvals=rep(NA, nsim))

for (s in 1:nsim) {
  X1<- rnorm(n,0, 2)
  X2<- rnorm(n,0, 3)
  Y <- beta0 + beta1*X1 + beta2*X2 + rnorm(n, 0, 1)
  res <- summary(lm(Y~X1+X2))$coefficients
  res.mat$beta1_pvals[s]<- res[2, 4]
  res.mat$beta2_pvals[s]<- res[3, 4]
}

#type 1 error
sum(res.mat$beta2_pvals<.05)/nsim
```

```
## [1] 0.049
```

```
#type 2 error
sum(res.mat$beta1_pvals>.05)/nsim
```

```
## [1] 0.028
```

```
#Density of errors where null is true -- what do you expect this to look like??
hist(res.mat$beta2_pvals)
```


Histogram of res.mat$beta2_pvals
Frequency
res.mat$beta2_pvals