Lab 4B - What's the score?

Directions: Follow along with the slides and answer the questions in **bold** font in your journal.

## Previously

* In the previous lab, we learned we could make predictions about one variable by utilizing the information of another.
* In this lab, we will learn how to measure the accuracy of our predictions.
  + This in turn will let us evaluate how well a model performs at making predictions.
  + We'll also use this information later to compare different models to find which model makes the best predictions.

## Predictions using a line

* Load the arm\_span data again.
  + Create an xyplot with height on the y-axis and armspan on the x-axis.
  + Type add\_line() to run the add\_line function; you'll be prompted to click twice in the plot window to create a line that you think fits the data well.
* Fill in the blanks below to create a function that will make predictions of people's heights based on their armspan:

make\_predictions <- function(armspans) {  
 \_\_\_\_ \* armspans + \_\_\_\_  
}

## Make your predictions

* Fill in the blanks to include your predictions in the arm\_span data.

\_\_\_\_ <- mutate(\_\_\_\_, predictions = \_\_\_\_(\_\_\_\_))

* Now that we've made our predictions, we'll need to figure out a way to decide how accurate our predictions are.
  + We'll want to compare our *predicted heights* to the *actual heights*.
  + At the end, we'll want to come up with a single number summary that describes our model's accuracy.

## Sums of differences

* One method we might consider to measure our model's accuracy is to sum the differences in the actual heights minus our predicted heights.
  + **What do these differences measure?**
  + Fill in the blanks below to create a function which calculates the sum of differences:

accuracy <- function(actual, predicted) {  
 sum(\_\_\_\_ - \_\_\_\_)  
}

* Then fill in the blanks to calculate our accuracy summary.

summarize(\_\_\_\_, \_\_\_\_(\_\_\_\_, \_\_\_\_))

## Checking our work

* **Describe and interpret, in words, what the output of your accuracy summary means.**
  + **Compare your accuracy summary with a neighbors. Whose line was more accurate and why?**
* **Write down why adding positive and negative errors together is problematic for accessing prediction accuracy.**
  + **Why does calculating the squared values for the differences solve this problem?**
* Alter your accuracy function to first calculate the differences, then square them and finally take the mean of the squared differences. This is called the *mean squared error* (MSE).
  + Calculate the MSE of your line.

## On your own

* Create a *regression line* as you did in the previous lab, for height and armspan.
  + We also refer to *regression lines* as *linear models*.
  + Assign this model the name best\_fit.
* Making predictions with models R is familiar with is simpler than with lines, or models, we come up with ourselves.
  + Fill in the blanks to make predictions using best\_fit:

\_\_\_\_ <- mutate(\_\_\_\_, predictions = predict(\_\_\_\_))

* Calculate the MSE for these new predicted values.

## The magic of lm()

* The lm() function creates the *line of best fit* equation by finding the line that minimizes the *mean squared error*. Meaning, it's the *best fitting line possible*.
  + Compare the MSE value you calculated using the line you fitted with add\_line() to the the same value you calculated using the lm function.
  + Ask your neighbors if any of their lines beat the lm line in terms of the MSE. Were any of them successful?
* To see how the lm line fits your data, create a scatterplot and then run:

add\_line(intercept = \_\_\_\_, slope = \_\_\_\_)