Lab 4F - Some models have curves

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

## Making models do yoga

* In the previous lab, we saw that prediction models could be improved by including additional variables.
  + But using straight lines for all the variables in a model might not really fit what's happening in the data.
* In this lab, we'll learn how we can turn our lm() models using straight lines into lm() models using quadratic curves.
* Load the movie data and split it into two sets:
  + A set named training that includes 75% of the data.
  + And a set named testing that includes the remaining 25%.
  + Remember to use set.seed.

## Problems with lines

* Calculate the *slope* and *intercept* of a linear model that predicts audience\_rating based on critics\_rating for the training data.
  + Then create a scatterplot of the two variables using the testing data and use add\_line() to include the *line of best fit* based on the training data..
  + **Describe, in words, how the line fits the data? Are there any values for critics\_rating that would make obviously poor predictions?**
* **Compute the MSE of the model for the testing data and write it down for later.**

## Adding flexibility

* You don't need to be a full-fledged Data Scientist to realize that trying to fit a line to curved data is a poor modeling choice.
  + If our data is curved, we should try model it with a curve.
* So instead of using an lm() like
* y = a + bx
* We could use an lm() like
* y = a + bx + cx2
* This is called a *quadratic* curve.

## Making bend-y models

* To fit a quadratic model in R, we can use the poly() function.
  + Fill in the blanks below to predict audience\_rating using a quadratic polynomial for critics\_rating.

lm(\_\_\_\_ ~ poly(\_\_\_\_, 2), data = training)

* **What is the role of the number 2 in the poly() function?**
* **Write down the model equation in the form:**
* y = a + bx + cx2
* Assign this model a name and calculate the MSE for the testing\_data.

## Comparing lines and curves

* Create a scatterplot with audience\_rating on the y-axis and critics\_rating on the x-axis using your testing data.
  + Add the *line of best fit* for the training data to the plot.
  + Then use the name of the model in the code below to add your *quadratic* model:

add\_curve(\_\_\_\_)

* **Compare how the *line of best fit* and the *quadratic* model fit the data. Use the difference in each model's testing MSE to describe why one model fits better than the other.**

## On your own

* Create a model that predicts audience\_rating using a 3 degree polynomial (called a *cubic* model) for the critics\_rating using the training data.
  + By using a plot, describe why you think a 2 or 3 degree polynomial will make better predictions for the testing data.
  + Compute the MSE for the model with a 3 degree polynomial and use the MSE to justify whether the 2 or 3 degree polynomial fits the testing data better.
  + Using the linear model from above which has the smallest MSE, include a different numerical variable to the model and recompute the MSE. Does modeling the variable you chose as a quadratic polynomial improve the MSE further?