

# STAT 306 Finding Relationships in Data

## Lab 3 - Simple Linear Regression Using 1m

David Lee

University of British Columbia  
Department of Statistics

January 27/29, 2015

# Regression in R

- Last week we saw how we can use the function `lsfit` to fit a simple linear regression model, and extract relevant information through the functions `ls.print` and `ls.diag`.
- `lm` (stands for linear models) is a more popular function for the same purpose. Almost every detail you need based on a linear regression model can be obtained through the `lm` object and some associated functions.

## Using `lm`

- Unlike `lsfit`, in `lm` the observations (predictors and responses) are specified through the use of a **formula** (and not as separate arguments), in the form **`response~predictor(s)`**.
- An optional data argument specifies the variable containing the data, stored as a data frame. When data is specified, you can simply use the column headers of that data frame in the formula.
- A summary of important quantities (such as coefficients, standard errors of fitted parameters etc) can be obtained using the function **`summary`**. Note that `summary` is a generic function; when called, this function looks for the type of object supplied and executes the relevant summary function for it. Therefore `?summary` will not give you a very useful help document.
- A fitted `lm` object is of the type `lm` (unsurprisingly), and the help document can be accessed by typing `summary.lm`.

# The predict function

- The `predict` function allows you to calculate pretty much everything given a future observation of  $x$ . Like `summary`, it is also a generic function, and the relevant help manual should be accessed through `predict.lm`.
- `predict` is very picky about the format of the new data supplied. It must be a data frame (or list) containing the same column headers as the predictors used in fitting, even if there is only one new observation or one predictor.
- If you are unsure of the column header you should use, type `model.frame(lmobj)` to get the model frame used by `lm`. Replace `lmobj` by the variable you stored the `lm` object to.

## The predict function (cont.)

- Failure to specify your new data properly will result in R using the original data for prediction, which will give you the fitted values instead. Sometimes R will not give any warning message about this.
- To make sure you are predicting on the correct  $x$ , you should always **check the length of the values returned** (and whether the values make sense).
- You can also use the `predict` function to calculate confidence or prediction intervals at any coverage probability.

## Lab question

- See WeBWork for this week's lab question. Question 1 is admittedly not very related to what we covered today, but Question 2 is (and should be very simple if you have been paying attention!). You need to submit your response by Friday 10pm.
- If you can't recall how to construct a matrix, refer to the R file for Lab 1.
- Don't hesitate to ask, especially if you are having trouble with something in R (manipulating vectors/matrices, algebraic operations, functions not working etc). This is what the labs are designed for.