```yaml
Course: Regression_Models
Lesson: Least_Squares_Estimation


- Class: text
  Output: "Least Squares Estimation. (Slides for this and other Data Science courses may be found at github https://github.com/DataScienceSpecialization/courses. If you care to use them, they must be downloaded as a zip file and viewed locally. This lesson corresponds to Regression_Models/01_03_ols. Galton data is from John Verzani's website, http://wiener.math.csi.cuny.edu/UsingR/)"

- Class: text
  Output: In this lesson, if you're using RStudio, you'll be able to play with some of the code which appears in the slides. If you're not using RStudio, you can look at the code but you won't be able to experiment with the function "manipulate". We provide the code for you so you can examine it without having to type it all out.  In RStudio, when the edit window displays code, make sure your flashing cursor is back in the console window before you hit "Enter" or any keyboard buttons, otherwise you might accidentally alter the code. If you do alter the file, in RStudio, you can hit Ctrl z in the editor until all the unwanted changes disappear. In other editors, you'll have to use whatever key combination performs "undo" to remove all your unwanted changes.

- Class: figure
  Output: Here are the Galton data and the regression line seen in the Introduction. The regression line summarizes the relationship between parents' heights (the predictors) and their children's (the outcomes).
  Figure: plot1.R
  FigureType: new

- Class: text
  Output: We learned in the last lesson that the regression line is the line through the data which has the minimum (least) squared "error", the vertical distance between the 928 actual children's heights and the heights predicted by the line. Squaring the distances ensures that data points above and below the line are treated the same. This method of choosing the 'best' regression line (or 'fitting' a line to the data) is known as ordinary least squares.

- Class: figure
  Output: As shown in the slides, the regression line contains the point representing the means of the two sets of heights. These are shown by the thin horizontal and vertical lines. The intersection point is shown by the triangle on the plot. Its x-coordinate is the mean of the parents' heights and y-coordinate is the mean of the childrens' heights.
  Figure: meanpt.R
  FigureType: add

- Class: text
  Output: As shown in the slides, the slope of the regression line is the correlation between the two sets of heights multiplied by the ratio of the standard deviations (childrens' to parents' or outcomes to predictors).

- Class: figure
  Output: Here we show code which demonstrates how changing the slope of the regression line affects the mean squared error between actual and predicted values. Look it over to see how straightforward it is.
  Figure: demofile.R
  FigureType: new

- Class: mult_question
  Output: What RStudio graphics package allows the user to play with the data to see the effects of the changes?
  AnswerChoices: manipulate; plot; abline; points
  CorrectAnswer: manipulate
  AnswerTests: omnitest(correctVal='manipulate')
  Hint: Three of the four choices all plot.

- Class: figure
  Output: Now you can actually play with the code to use R's manipulate function and
```

```
    find the minimum squared error. You can adjust the slider with the left mouse button
    or use the right and left arrow keys to see how changing the slope (beta) affects the
    mean squared error (mse). If the slider disappears you can call it back by clicking
    on the little gear in the upper left corner of the plot window.
  Figure: sourceit.R
  FigureType: new

- Class: mult_question
  Output:  Which value of the slope minimizes the mean squared error?
  AnswerChoices: .64; .44; .70; 5
  CorrectAnswer: .64
  AnswerTests: omnitest(correctVal='.64')
  Hint: If you list the choices from least to biggest pick one of the two middle choices.

- Class: mult_question
  Output:  What was the minimum mse?
  AnswerChoices: 5.0; .64; 44; .66
  CorrectAnswer: 5.0
  AnswerTests: omnitest(correctVal='5.0')
  Hint: You don't want an error that's too big or too small.

- Class: text
  Output: Recall that you normalize data by subtracting its mean and dividing by its
    standard deviation. We've done this for the galton child and parent data for you.
    We've stored these normalized values in two vectors, gpa_nor and gch_nor, the
    normalized galton parent and child data.

- Class: cmd_question
  Output: Use R's function "cor" to compute the correlation between these normalized
    data sets.
  CorrectAnswer: cor(gpa_nor,gch_nor)
  AnswerTests: ANY_of_exprs('cor(gpa_nor,gch_nor)','cor(gch_nor,gpa_nor)')
  Hint: Type "cor(gpa_nor,gch_nor)" at the R prompt.

- Class: mult_question
  Output:  How does this correlation relate to the correlation of the unnormalized data?
  AnswerChoices: It is the same.; It is bigger.; It is smaller.
  CorrectAnswer: It is the same.
  AnswerTests: omnitest(correctVal='It is the same.')
  Hint: Have you really changed anything?

- Class: cmd_question
  Output: Use R's function "lm" to generate the regression line using this normalized
    data. Store it in a variable called l_nor. Use the parents' heights as the predictors
    (independent variable) and the childrens' as the predicted (dependent). Remember,
    'lm' needs a formula of the form dependent ~ independent. Since we've created the
    data vectors for you there's no need to provide a second "data" argument as you have
    previously.
  CorrectAnswer: l_nor <- lm(gch_nor ~ gpa_nor)
  AnswerTests: omnitest(correctExpr='l_nor <- lm(gch_nor ~ gpa_nor)')
  Hint: Type "l_nor <- lm(gch_nor ~ gpa_nor)" at the R prompt.

- Class: mult_question
  Output:  What is the slope of this line?
  AnswerChoices: The correlation of the 2 data sets; I have no idea; 1.
  CorrectAnswer:  The correlation of the 2 data sets
  AnswerTests: omnitest(correctVal='The correlation of the 2 data sets')
  Hint: Think correlation.

- Class: mult_question
  Output:  If you swapped the outcome (Y) and predictor (X) of your original
    (unnormalized) data, (for example, used childrens' heights to predict their parents),
    what would the slope of the new regression line be?
  AnswerChoices: correlation(X,Y) * sd(X)/sd(Y); the same as the original; I have no
    idea; 1.
  CorrectAnswer: correlation(X,Y) * sd(X)/sd(Y)
  AnswerTests: omnitest(correctVal='correlation(X,Y) * sd(X)/sd(Y)')
  Hint: Since you're swapping X and Y, swap the X and Y in the formula. Swapping X and
    Y in the correlation function doesn't change anything.
```

```
- Class: figure
  Output: We'll close with a final display of source code from the slides. It plots the galton data with three regression lines, the original in red with the children as the outcome,  a new blue line with the parents' as outcome and childrens' as predictor, and a black line with the slope scaled so it equals the ratio of the standard deviations.
  Figure: demofile2.R
  FigureType: new

- Class: text
  Output: Congrats! You've concluded this lesson on ordinary least squares which are truly extraordinary!

- Class: mult_question
  Output: "Would you like to receive credit for completing this course on
    Coursera.org?"
  CorrectAnswer: NULL
  AnswerChoices: Yes;No
  AnswerTests: coursera_on_demand()
  Hint: ""
```