```
Course: Regression Models
Lesson: Introduction


- Class: text
  Output: "Introduction to Regression Models. (Slides for this and other Data Science courses may be found at github https://github.com/DataScienceSpecialization/courses if you want to use them. They must be downloaded as a zip file and viewed locally. This lesson corresponds to Regression_Models/01_01_introduction.)"

- Class: text
  Output: This is the first lesson on Regression Models. We'll begin with the concept of "regression toward the mean" and illustrate it with some pioneering work of the father of forensic science, Sir Francis Galton.

- Class: text
  Output: Sir Francis studied the relationship between heights of parents and their children. His work showed that parents who were taller than average had children who were also tall but closer to the average height. Similarly, parents who were shorter than average had children who were also shorter than average but less so than mom and dad. That is, they were closer to the average height. From one generation to the next the heights moved closer to the average or regressed toward the mean.

- Class: text
  Output: For this lesson we'll use Sir Francis's parent/child height data which we've taken the liberty to load for you as the variable, galton. (Data is from John Verzani's website, http://wiener.math.csi.cuny.edu/UsingR/.) So let's get started!

- Class: figure
  Output: Here is a plot of Galton's data, a set of 928 parent/child height pairs. Moms' and dads' heights were averaged together (after moms' heights were adjusted by a factor of 1.08). In our plot we used the R function "jitter" on the children's heights to highlight heights that occurred most frequently. The dark spots in each column rise from left to right suggesting that children's heights do depend on their parents'. Tall parents have tall children and short parents have short children.
  Figure: plot1_children_vs_parents.R
  FigureType: new

- Class: figure
  Output: Here we add a red (45 degree) line of slope 1 and intercept 0 to the plot. If children tended to be the same height as their parents, we would expect the data to vary evenly about this line. We see this isn't the case. On the left half of the plot we see a concentration of heights above the line, and on the right half we see the concentration below the line.
  Figure: plot2_identity_line.R
  FigureType: add

- Class: figure
  Output: Now we've added a blue regression line to the plot. This is the line which has the minimum variation of the data around it. (For theory see the slides.) Its slope is greater than zero indicating that parents' heights do affect their children's. The slope is also less than 1 as would have been the case if children tended to be the same height as their parents.
  Figure: plot3_regression_line.R
  FigureType: add

- Class: cmd_question
  Output: Now's your chance to plot in R. Type "plot(child ~ parent, galton)" at the R prompt.
  CorrectAnswer: plot(child ~ parent, galton)
  AnswerTests: omnitest(correctExpr='plot(child ~ parent, galton)')
  Hint: Type "plot(child ~ parent, galton)" at the R prompt.
  Figure: restore_1.R
  FigureType: new

- Class: text
  Output: You'll notice that this plot looks a lot different than the original we displayed. Why? Many people are the same height to within measurement error, so points fall on top of one another. You can see that some circles appear darker than
```

```
  others. However, by using R's function "jitter" on the children's heights, we can
  spread out the data to simulate the measurement errors and make high frequency
  heights more visible.


- Class: cmd_question
  Output: Now it's your turn to try. Just type "plot(jitter(child,4) ~ parent,galton)"
  and see the magic.
  CorrectAnswer: plot(jitter(child,4) ~ parent,galton)
  AnswerTests: omnitest(correctExpr='plot(jitter(child,4) ~ parent,galton)')
  Hint: You can do it! Type "plot(jitter(child,4) ~ parent,galton)"
  Figure: restore_2.R
  FigureType: new

- Class: text
  Output: Now for the regression line. This is quite easy in R. The function lm (linear
  model) needs a "formula" and dataset. You can type "?formula" for more information,
  but, in simple terms, we just need to specify the dependent variable (children's
  heights) ~ the independent variable (parents' heights).

- Class: cmd_question
  Output: So generate the regression line and store it in the variable regrline. Type
  "regrline <- lm(child ~ parent, galton)"
  CorrectAnswer: regrline <- lm(child ~ parent, galton)
  AnswerTests: omnitest(correctExpr='regrline <- lm(child ~ parent,
  galton)');expr_creates_var('regrline')
  Hint: You CAN do it! Type "regrline <- lm(child ~ parent, galton)"

- Class: cmd_question
  Output: Now add the regression line to the plot with "abline". Make the line wide and
  red for visibility. Type "abline(regrline, lwd=3, col='red')"
  CorrectAnswer: abline(regrline, lwd=3, col='red')
  AnswerTests: omnitest(correctExpr='abline(regrline, lwd=3, col=\'red\')')
  Hint: Yes, you can! Type "abline(regrline, lwd=3, col='red')"
  Figure: restore_3.R
  FigureType: add

- Class: cmd_question
  Output: The regression line will have a slope and intercept which are estimated from
  data. Estimates are not exact. Their accuracy is gauged by theoretical techniques and
  expressed in terms of "standard error." You can use  "summary(regrline)" to examine
  the Galton regression line. Do this now.
  CorrectAnswer: summary(regrline)
  AnswerTests: omnitest(correctExpr='summary(regrline)')
  Hint: This one's easy. Type "summary(regrline)"

- Class: mult_question
  Output: The slope of the line is the estimate of the coefficient, or multiplier, of
  "parent", the independent variable of our data (in this case, the parents' heights).
  From the output of "summary" what is the slope of the regression line?
  AnswerChoices: .64629;.04114;23.94153
  CorrectAnswer: .64629
  AnswerTests: omnitest(correctVal= '.64629')
  Hint: Look at the line labelled "parent" and the column "Estimate"

- Class: mult_question
  Output: What is the standard error of the slope?
  AnswerChoices: .64629;.04114;23.94153
  CorrectAnswer: .04114
  AnswerTests: omnitest(correctVal= '.04114')
  Hint: Look at the line labelled "parent" and the column "Std. Error."

- Class: text
  Output: A coefficient will be within 2 standard errors of its estimate about 95% of
  the time. This means the slope of our regression is significantly different than
  either 0 or 1 since (.64629) +/- (2*.04114) is  near neither 0 nor 1.

- Class: figure
  Output: We're now adding two blue lines to indicate the means of the children's
```

```
    heights (horizontal) and the parents' (vertical). Note that these lines and the regression line all intersect in a point. Pretty cool, huh? We'll talk more about this in a later lesson. (Something you can look forward to.)
  Figure: plot4_mean_heights.R
  FigureType: add

- Class: figure
  Output: The slope of a line shows how much of a change in the vertical direction is produced by a change in the horizontal direction. So, parents "1 inch" above the mean in height tend to have children who are only .65 inches above the mean. The green triangle illustrates this point. From the mean, moving a "1 inch distance" horizontally to the right (increasing the parents' height)  produces a ".65 inch" increase in the vertical direction (children's height).
  Figure: plot5_triangle1.R
  FigureType: add

- Class: figure
  Output: Similarly, parents who are 1 inch below average in height have children who are only .65 inches below average height. The purple triangle illustrates this. From the mean, moving a "1 inch distance" horizontally to the left (decreasing the parents' height)  produces a ".65 inch" decrease in the vertical direction (children's height).
  Figure: plot5_triangle2.R
  FigureType: add

- Class: text
  Output: This concludes our lesson on regression toward the mean. We hope you found it above average!

- 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: ""

```