**DATA SPECIALIZATION**

***Unit 7: Regression Models***

# Introduction

**Link**: <https://www.coursera.org>

**Overview**: We believe that the key word in Data Science is 'science'. Our course track is focused on providing you with three things: (1) an introduction to the key ideas behind working with data in a scientific way that will produce new and reproducible insight, (2) an introduction to the tools that will allow you to execute on a data analytic strategy, from raw data in a database to a completed report with interactive graphics, and (3) on giving you plenty of hands on practice so you can learn the techniques for yourself.

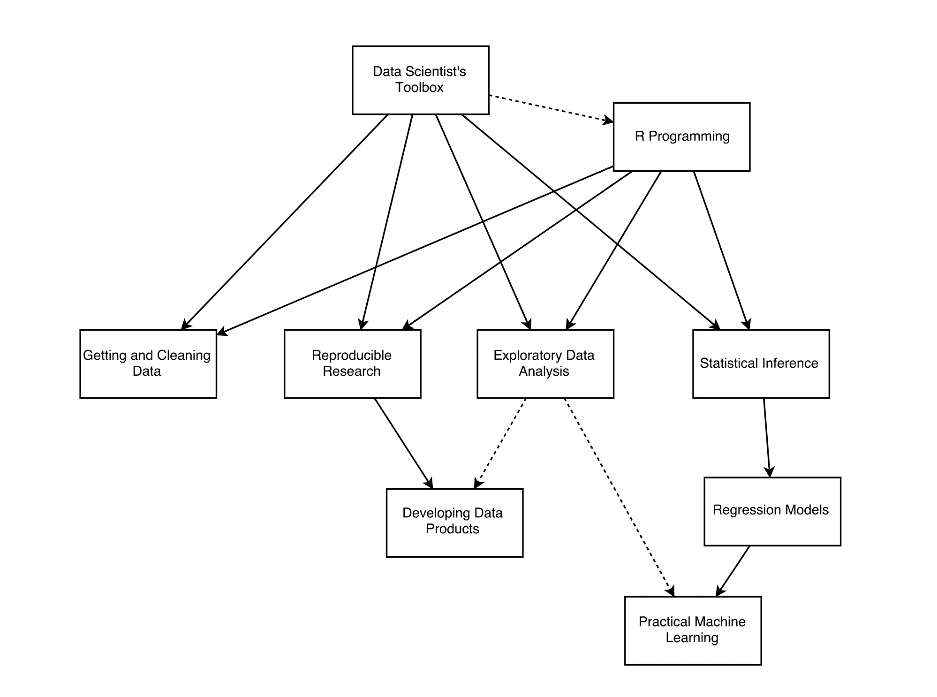
Regression Models represents a both fundamental and foundational component of the series, and it presents the single most practical data analysis toolset. Using only a bare minimum of mathematics, we will attempt to provide you with the fundamentals for the application and practice of regression.

We are excited about the opportunity to attempt to scale Data Science education. We intend for the courses to be self-contained, fast-paced, and interactive, and we intend to run them frequently to give people with busy schedules the opportunity to work on material at their own pace.

Brian Caffo and the Data Science Track Team

**About this Course:** Linear models, as their name implies, relates an outcome to a set of predictors of interest using linear assumptions. Regression models, a subset of linear models, are the most important statistical analysis tool in a data scientist’s toolkit. This course covers regression analysis, least squares and inference using regression models. Special cases of the regression model, ANOVA and ANCOVA will be covered as well. Analysis of residuals and variability will be investigated. The course will cover modern thinking on model selection and novel uses of regression models including scatterplot smoothing.

**Objective**: Ask the right questions, manipulate data sets, and create visualizations to communicate results.

This Specialization covers the concepts and tools you'll need throughout the entire data science pipeline, from asking the right kinds of questions to making inferences and publishing results. In the final Capstone Project, you’ll apply the skills learned by building a data product using real-world data. At completion, students will have a portfolio demonstrating their mastery of the material.

**Who:**

Jeff Leek – professor JHU School of Public Health. Statistics of genetic data.

* <http://biostat.jhsph.edu/~jleek/>, <http://simplystatistics.org/>, <https://github.com/jtleek>

Roger Peng

* <http://www.biostat.jhsph.edu/~rpeng/>, http://simplystatistics.org/

Brian Caffo – statistics to analyze brain image data.

* [www.bcaffo.com](http://www.bcaffo.com), <https://githup.com/bcaffo>

**Plagiarism**

Johns Hopkins University defines plagiarism as "...taking for one's own use the words, ideas, concepts or data of another without proper attribution. Plagiarism includes both direct use or paraphrasing of the words, thoughts, or concepts of another without proper attribution." We take plagiarism very seriously, as does Johns Hopkins University.

We recognize that many students may not have a clear understanding of what plagiarism is or why it is wrong. Please see the JHU referencing guide for more information on plagiarism.

It is critically important that you give people/sources credit when you use their words or ideas. If you do not give proper credit -- particularly when quoting directly from a source -- you violate the trust of your fellow students.

The Coursera Honor code includes an explicit statement about plagiarism:

I will register for only one account. My answers to homework, quizzes and exams will be my own work (except for assignments that explicitly permit collaboration). I will not make solutions to homework, quizzes or exams available to anyone else. This includes both solutions written by me, as well as any official solutions provided by the course staff. I will not engage in any other activities that will dishonestly improve my results or dishonestly improve/hurt the results of others.

**Reporting plagiarism on course projects**

One of the criteria in the project rubric focuses on plagiarism. Keep in mind that some components of the projects will be very similar across terms and so answers that appear similar may be honest coincidences. However, we would appreciate if you do a basic check for obvious plagiarism and report it during your peer assessment phase. It is currently very difficult to prove or disprove a charge of plagiarism in the MOOC peer assessment setting. We are not in a position to evaluate whether or not a submission actually constitutes plagiarism, and we will not be able to entertain appeals or to alter any grades that have been assigned through the peer evaluation system. But if you take the time to report suspected plagiarism, this will help us to understand the extent of the problem and work with Coursera to address critical issues with the current system.

# Regression Models

Started: July 11, 2016

Assignment due: August 8, 2016

Ends: August 14, 2016

# Week 1: Least Squares and Linear Regression

This week, we focus on least squares and linear regression.

### Welcome to Regression Models

I am happy that you've chosen to take Regression Models, part of the Johns Hopkins Data Science Specialization on Coursera! This course presents the fundamentals of regression modeling that you will need for the rest of the specialization and ultimately for your work in the field of data science.

We believe that the key word in Data Science is "science". Our course track is focused on providing you with three things: (1) an introduction to the key ideas behind working with data in a scientific way that will produce new and reproducible insight, (2) an introduction to the tools that will allow you to execute on a data analytic strategy, from raw data in a database to a completed report with interactive graphics, and (3) on giving you plenty of hands on practice so you can learn the techniques for yourself.

Regression Models represents a both fundamental and foundational component of the series, and it presents the single most practical data analysis toolset. Using only a bare minimum of mathematics, we will attempt to provide you with the fundamentals for the application and practice of regression.

We are excited about the opportunity to attempt to scale Data Science education. We intend for the courses to be self-contained, fast-paced, and interactive, and we intend to run them frequently to give people with busy schedules the opportunity to work on material at their own pace.

Some Basics

A couple of first week housekeeping items. First, make sure that you've had [R Programming](https://coursera.org/learn/rprog), the [Data Scientist's Toolbox](https://coursera.org/learn/datascitoolbox), [Reproducible Research](https://coursera.org/learn/repdata) and [Statistical Inference](https://coursera.org/learn/statinference) before taking this class. At a minimum you must know: very basic git, basic R and most of the Statistical Inference Coursera class. The small amount of knitr that you need for the project you can pick up quickly.

An important aspect of this class is to peruse the materials in the github repository. All of the most up to date material can be found here: [https://github.com/bcaffo/courses/tree/master/07\_RegressionModels](https://eventing.coursera.org/redirect/Kbp848YCgjxdpeGihUcxr4_PL8JM5tJxrEVr91e09B5KidCoGXWxe8c2AEKCoEXbBAaW4JMUUXI0R-KgvzNd7g.JywEThkg4eCE-iu2MYP7iw.u4F02Fpu8cveIoeaVeh3pyTDNTaRJByZyhIGJfoDbxSyWgvqb0RD-DHdGbcyGY3zs-TN62p7m08WdsmXoq7n1VM-HxV5Uf4gFt-i8N_pBBvfQE4bjJnLqJ5eeK8CuxbHQc8pesyaEIjrGmYf3lxvZHPq-ut2iPxKZgr1BXWTpzieJA8wNnfsXKZ46FHpt1cuQqXktkHcOliBOtM2J5derfGslUUlJ92-hUi4HWYjt6haT5hYL5w6DZZv2AifOtsBp_l9kceFDG51fpkyyXRAdyao4gbWdXraU1iwz1cbUjaN4MMtMsjFSg_2vkiDpohJxZLN_Cbt2c9mFMvgzjoq7Q)

You should clone this repository as your first step in this class and make sure to fetch updates periodically. (Please send pull requests too!) It is one of the most essential components of the Specialization that you start to use Git frequently. We're practicing what we preach as well by using the tools in the series to create the series, especially git.

You can clone the whole repo with (http)

git clone <https://github.com/bcaffo/courses.git>

or (ssh)

git clone git@github.com:bcaffo/courses.git

The lectures are in the index.Rmd lecture files. In [Developing Data Products](https://coursera.org/learn/devdataprod), we'll cover how to create these sorts of slides. However, for the time being, you should be able to open them in R Studio and look at their contents. You will see all of the R code to recreate the lectures. Going through the R code is the best way to familiarize yourself with the lecture materials.

If you'd prefer to watch the videos on YouTube, you can find them here: <https://www.youtube.com/playlist?list=PLpl-gQkQivXjqHAJd2t-J_One_fYE55tC>

If you'd like to keep up with the instructors I'm @bcaffo on twitter, Roger is @rdpeng and Jeff is @jtleek. The Department of Biostat here is @jhubiostat.

### Book: Regression Models for Data Science in R

Got it.

### Syllabus

This class is co-taught by Roger Peng and Jeff Leek. In addition, Sean Kross and Nick Carchedi have been helping greatly.

Course Description

Linear models, as their name implies, relates an outcome to a set of predictors of interest using linear assumptions. Regression models, a subset of linear models, are the most important statistical analysis tool in a data scientist's toolkit. This course covers regression analysis, least squares and inference using regression models. Special cases of the regression model, ANOVA and ANCOVA will be covered as well. Analysis of residuals and variability will be investigated. The course will cover modern thinking on model selection and novel uses of regression models including scatterplot smoothing.

Course Content

This class has three main components

Least squares and linear regression

Multivariable regression

Generalized linear models

The full list of topics are as follows:

Module 1, least squares and linear regression

01\_01 Introduction

01\_02 Notation

01\_03 Ordinary least squares

01\_04 Regression to the mean

01\_05 Linear regression

01\_06 Residuals

01\_07 Regression inference

Module 2, Multivariable regression

02\_01 Multivariate regression

02\_02 Multivariate examples

02\_03 Adjustment

02\_04 Residual variation and diagnostics

02\_05 Multiple variables

Module 3, Generalized linear models

03\_01 GLMs

03\_02 Binary outcomes

03\_03 Count outcomes

03\_04 Olio

Module 4, Logistic Regression and Poisson Regression

Logistic Regression

Poisson Regression

Hodgepodge

Github repository

The most up to date information on the course lecture notes will always be in the Github repository. Please issue pull requests so that we may improve the materials. Note my GitHub repo will generally be more up to date than the Data Science Specialization Repo.

Youtube videos

If you'd prefer to watch the videos on youtube, they can be found here:

<https://www.youtube.com/playlist?list=PLpl-gQkQivXhdgUCdaUQcdb31CRe8Mm2y>

Book: Regression Models for Data Science in R

A companion book is available here: https://leanpub.com/regmods

The book is published via leanpub, and the suggested price is $14.99. You can get it for free or pay what you feel it is worth.

Quizzes

There are four weekly quizzes.

You must earn a grade of at least 80% to pass a quiz

You may attempt each quiz up to 3 times in 8 hours.

The score from your most successful attempt will count toward your final grade.

Course Project

The Course Project is an opportunity to demonstrate the skills you have learned during the course. It is graded through peer assessment. You must earn a grade of at least 80% to pass the peer assessment.

Grading Policy

You must score at least 80% on all assignments (Quizzes & Project) to pass the course.

Your final grade will be calculated as follows:

Quiz 1 = 15%

Quiz 2 = 15%

Quiz 3 = 15%

Quiz 4 = 15%

Course Project = 40%

swirl Programming Assignment (optional)

In this course, you have the option to use the swirl R package to practice some of the concepts we cover in lectures.

While these lessons will give you valuable practice and you are encouraged to complete as many as possible, please note that they are completely optional and you can get full marks in the class without completing them.

Differences of opinion

Keep in mind that currently data analysis is as much art as it is science - so we may have a difference of opinion - and that is ok! Please refrain from angry, sarcastic, or abusive comments on the message boards. Our goal is to create a supportive community that helps the learning of all students, from the most advanced to those who are just seeing this material for the first time.

Plagiarism

Johns Hopkins University defines plagiarism as "...taking for one's own use the words, ideas, concepts or data of another without proper attribution. Plagiarism includes both direct use or paraphrasing of the words, thoughts, or concepts of another without proper attribution." We take plagiarism very seriously, as does Johns Hopkins University.

We recognize that many students may not have a clear understanding of what plagiarism is or why it is wrong. Please see the following guide for more information on plagiarism:

<http://www.jhsph.edu/academics/degree-programs/master-of-public-health/current-students/JHSPH-ReferencingHandbook.pdf>

It is critically important that you give people/sources credit when you use their words or ideas. If you do not give proper credit -- particularly when quoting directly from a source -- you violate the trust of your fellow students.

The Coursera Honor code includes an explicit statement about plagiarism:

I will register for only one account. My answers to homework, quizzes and exams will be my own work (except for assignments that explicitly permit collaboration). I will not make solutions to homework, quizzes or exams available to anyone else. This includes both solutions written by me, as well as any official solutions provided by the course staff. I will not engage in any other activities that will dishonestly improve my results or dishonestly improve/hurt the results of others.

Reporting plagiarism on course projects

One of the criteria in the project rubric focuses on plagiarism. Keep in mind that some components of the projects will be very similar across terms and so answers that appear similar may be honest coincidences. However, we would appreciate if you do a basic check for obvious plagiarism and report it during your peer assessment phase.

It is currently very difficult to prove or disprove a charge of plagiarism in the MOOC peer assessment setting. We are not in a position to evaluate whether or not a submission actually constitutes plagiarism, and we will not be able to entertain appeals or to alter any grades that have been assigned through the peer evaluation system.

But if you take the time to report suspected plagiarism, this will help us to understand the extent of the problem and work with Coursera to address critical issues with the current system.

### Pre-Course Survey

Completed

### Data Science Specialization Community Site

<http://datasciencespecialization.github.io/>

### Where to get more advanced material

If you want more advanced material, I've been working on another version of this class. Eventually I hope to have a second Coursera class as well. Currently, you can get the E-Book in progress here: <https://leanpub.com/lm> (it's variable pricing including free!)

In addition, you can watch the videos as they're being developed here:<https://www.youtube.com/playlist?list=PLpl-gQkQivXhdgUCdaUQcdb31CRe8Mm2y>

## Introduction to regression and least squares

### Regression

Regression models are the workhorse of data science. They are the most well described, practical and theoretically understood models in statistics. A data scientist well versed in regression models will be able to solve an incredible array of problems.

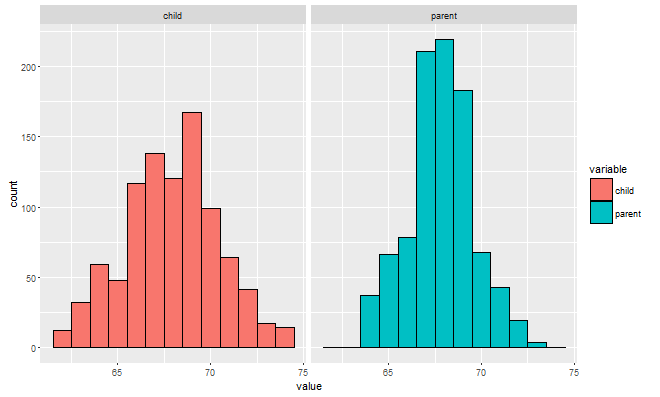
Perhaps the key insight for regression models is that they produce highly interpretable model fits. This is unlike machine learning algorithms, which often sacrifice interpretability for improved prediction performance or automation. These are, of course, valuable attributes in their own rights. However, the benefit of simplicity, parsimony and intrepretability offered by regression models (and their close generalizations) should make them a first tool of choice for any practical problem.

### Introduction to Regression

* linear regression (or generalization linear models) – goto method for many data science applications.
  + parsimonious and easily described mean relationship (regression is good at simple)
    - machine learning often comes up with some crazy shit
  + investigate residual variation (that appears unrelated)
  + quantify impact other information has beyond a primary factor to explain response
  + assumptions are needed to generalize findings beyond the data in question (tools of inference)
* Francis Galton – invented term and concept of linear regression (1885)
  + founded Biometricka journal and cousin to Charles Darwin
  + predicted child height from parent height in Victorian era
  + still relevant today
* Regression to the mean

### Introduction: Basic Least Squares

* look at the distributions of each group



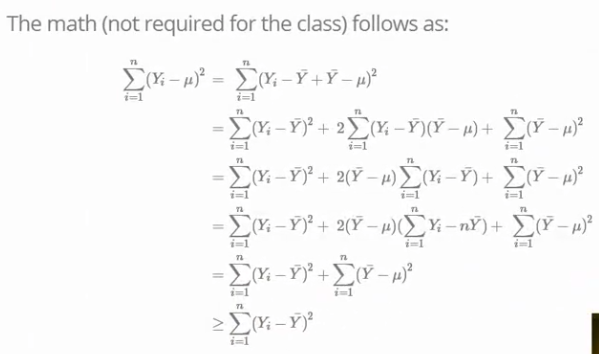
* Find the middle of least squares as center of mass of the histograms = the mean of the sample

### Technical details

Skip if you want to.

### Technical Details (Skip if you'd like)

* proof that Y-bar minimizes the square difference in means
* Yi are observed data points



### Introductory Data Example

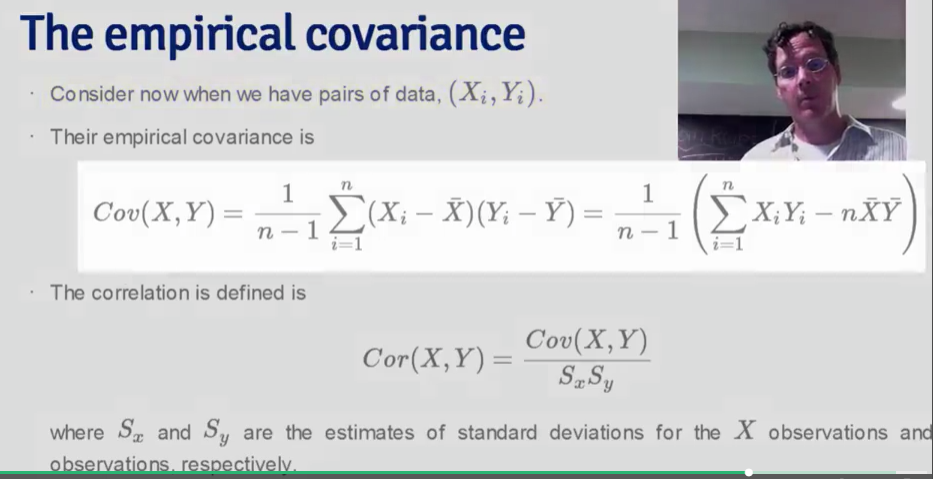
* scatter plot of the child vs. parent heights has problems
  + it is overplotted (different data on top of each other at specific points)
  + a better plot has points that vary with size and color
    - size = represent frequency of that value
    - color = represent frequency of that value

## Linear least squares

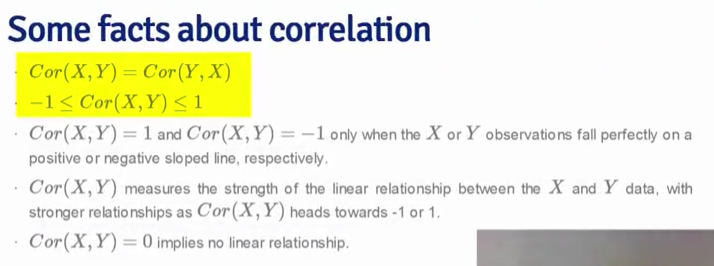
Ordinary least squares (OLS)is the workhorse of statistics. It gives a way of taking complicated outcomes and explaining behavior (such as trends) using linearity. The simplest application of OLS is fitting a line through some data. In the next few lectures, we cover the basics of linear least squares.

### Notation and Background

* Greek letters for ***population statistics*** we don’t know and try to estimate
* Arabic letters to indicate things that can be ***observed (sample statistics)***
* ***X-bar*** indicates sample mean
* ***X-tilda*** is mean ***centered*** data (have sample mean of zero) called centering
* ***Beta-0-hat*** is an estimated value for a parameter (hat)
* Sample mean (or empirical mean) is the least squares solution
* Empirical Standard Deviation and variance
  + shortcut version on right for **S2**
  + standard deviation , ***s***, is in same units as the data
  + ***X-i/s*** have empirical standard deviation of 1 (this is called ***scaling*** the data)
* ***Normalizing*** Data (centered and scaled) mean of zero and standard deviation of 1.
  + makes datasets comparable for variance/deviation
* Empirical ***Covariance***
  + ***Correlation*** is scaled version of the ***Covariance***

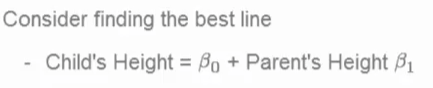


* Facts about correlation
  + bounded between -1 to 1
  + measures the strength of the linear relationship between X and Y data
  + 0 means no linear relationship
  + +/- is positive or negative complete correlation

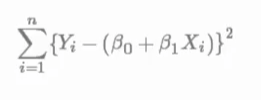


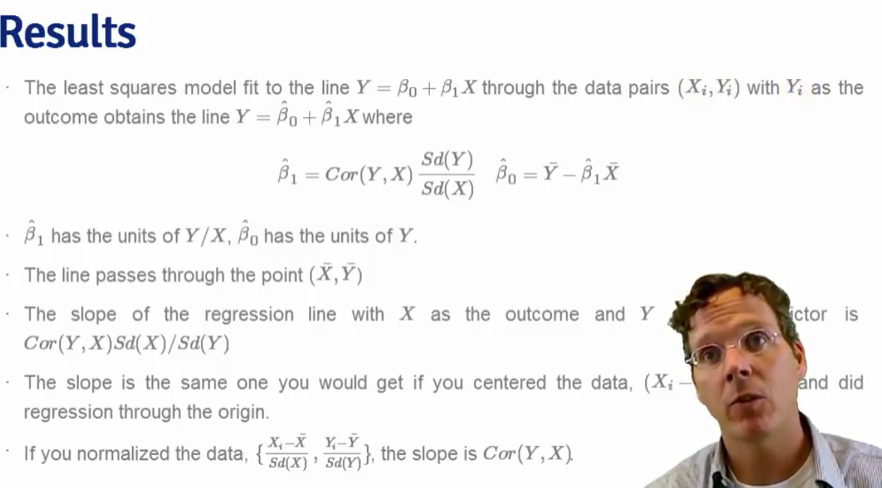
### Linear Least Squares

* general least squares for linear regression
* child-parent height example:



* Least squares criteria, such that you minimize the sum of squared distances between the data y-values and the y-value of the line.





### Linear Least Squares Coding Example

* Galton data. See the R markdown file.

### Technical Details (Skip if you'd like)

* See video

## Regression to the Mean

Here is a fundamental question. Why is it that the children of tall parents tend to be tall, but not as tall as their parents? Why do children of short parents tend to be short, but not as short as their parents? Conversely, why do parents of very short children, tend to be short, but not a short as their child? And the same with parents of very tall children?

We can try this with anything that is measured with error. Why do the best performing athletes this year tend to do a little worse the following? Why do the best performers on hard exams always do a little worse on the next hard exam?

These phenomena are all examples of so-called regression to the mean. Regression to the mean, was invented by Francis Galton in the paper “Regression towards mediocrity in hereditary stature” The Journal of the Anthropological Institute of Great Britain and Ireland , Vol. 15, (1886). The idea served as a foundation for the discovery of linear regression.

### Regression to the mean

* historically famous idea – sustained outliers don’t last forever when measuring with error.
* Example: simulate pairs of standard normal
* Question is what is cause of the regression? Is high/low value due to noise (error in measurement) or is it intrinsic (internal response to the input).
  + quiz performance: a good performer might have been benefit from noise in the test.
* Normalized data (X for child height, and Y parent height) so they both have mean 0 and variance 1
  + regression line passes through 0,0 (the mean of X, Y)
  + slope of line is Cor(X,Y) regardless of which variable is the outcome (both standard deviations are 1.
  + If X is output the slope is 1/Cor(X,Y)

## Practical R Exercises in swirl Part 1

install.packages(“swirl”)

packageVersion(“swirl”)

library(swirl)

install\_from\_swirl(“Regression Models”)

### Practice Programming Assignment: swirl Lesson 1: Introduction

* Galton: taller parents had children that were tall, but closer to the average.

| 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.

...

|======= | 10%

| 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.

...

|=========== | 14%

| 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!

...

|============== | 19%

| 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.

...

|================== | 24%

| 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.

...

|===================== | 29%

| 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.

...

|========================= | 33%

| Now's your chance to plot in R. Type "plot(child ~ parent, galton)" at the R prompt.

> plot(child ~ parent, galton)

| You got it!

|============================ | 38%

| 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.

...

|================================ | 43%

| Now it's your turn to try. Just type "plot(jitter(child,4) ~ parent,galton)" and see the magic.

> plot(jitter(child, 4) ~ paernt, galton)

Error in eval(expr, envir, enclos) : object 'paernt' not found

> plot(jitter(child, 4) ~ parent, galton)

| You're the best!

|=================================== | 48%

| 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).

...

|======================================= | 52%

| So generate the regression line and store it in the variable regrline. Type "regrline <- lm(child ~ parent, galton)"

> regrline <- lm(child ~ parent, galton)

| You're the best!

|========================================== | 57%

| 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')"

> abline(regrline, lwd=3, col='red')

| That's the answer I was looking for.

|============================================== | 62%

| 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.

> regrline <- lm(child ~ parent, galton)

| Not quite, but you're learning! Try again. Or, type info() for more options.

| This one's easy. Type "summary(regrline)"

> summary(regrline)

Call:

lm(formula = child ~ parent, data = galton)

Residuals:

Min 1Q Median 3Q Max

-7.8050 -1.3661 0.0487 1.6339 5.9264

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 23.94153 2.81088 8.517 <2e-16 \*\*\*

parent 0.64629 0.04114 15.711 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.239 on 926 degrees of freedom

Multiple R-squared: 0.2105, Adjusted R-squared: 0.2096

F-statistic: 246.8 on 1 and 926 DF, p-value: < 2.2e-16

| All that practice is paying off!

|================================================= | 67%

| The slope of the line is the estimate of the coefficient, or muliplier, 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?

1: .64629

2: 23.94153

3: .04114

Selection: 1

| You nailed it! Good job!

|===================================================== | 71%

| What is the standard error of the slope?

1: .64629

2: .04114

3: 23.94153

Selection: 2

| All that practice is paying off!

|======================================================== | 76%

| 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.

...

|============================================================ | 81%

| 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.)

| 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.)

...

|=============================================================== | 86%

| 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).

...

|=================================================================== | 90%

| 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).

...

|====================================================================== | 95%

| This concludes our lesson on regression toward the mean. We hope you found it above average!

...

|==========================================================================| 100%

### Practice Programming Assignment: swirl Lesson 2: Residuals

* generate a regression line with lm()
* verify the residuals are uncorrelated with the predictor variables with cov()
* var(data) = var(estimates) + var(residuals)
  + variance of residuals is always less than the variance of data

| Residuals. (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/)

...

|==== | 3%

| This lesson will focus on the residuals, the distances between the actual children's heights and the estimates given by

| the regression line. Since all lines are characterized by two parameters, a slope and an intercept, we'll use the least

| squares criteria to provide two equations in two unknowns so we can solve for these parameters, the slope and intercept.

...

|======= | 6%

| The first equation says that the "errors" in our estimates, the residuals, have mean zero. In other words, the residuals

| are "balanced" among the data points; they're just as likely to be positive as negative. The second equation says that our

| residuals must be uncorrelated with our predictors, the parents’ height. This makes sense - if the residuals and

| predictors were correlated then you could make a better prediction and reduce the distances (residuals) between the actual

| outcomes and the predictions.

...

|=========== | 9%

| We'll demonstrate these concepts now. First regenerate the regression line and call it fit. Use the R function lm. Recall

| that by default its first argument is a formula such as "child ~ parent" and its second is the dataset, in this case

| galton.

> fit <- lm(child ~ parent, galton)

| That's a job well done!

|============== | 12%

| Now we'll examine fit to see its slope and intercept. The residuals we're interested in are stored in the 928-long vector

| fit$residuals. If you type fit$residuals you'll see a lot of numbers scroll by which isn't very useful; however if you

| type "summary(fit)" you will see a more concise display of the regression data. Do this now.

> summary(fit)

Call:

lm(formula = child ~ parent, data = galton)

Residuals:

Min 1Q Median 3Q Max

-7.8050 -1.3661 0.0487 1.6339 5.9264

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 23.94153 2.81088 8.517 <2e-16 \*\*\*

parent 0.64629 0.04114 15.711 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.239 on 926 degrees of freedom

Multiple R-squared: 0.2105, Adjusted R-squared: 0.2096

F-statistic: 246.8 on 1 and 926 DF, p-value: < 2.2e-16

| You are amazing!

|================== | 16%

| First check the mean of fit$residuals to see if it's close to 0.

> mean(fit$residuals)

[1] -2.359884e-15

| Excellent work!

|====================== | 19%

| Now check the correlation between the residuals and the predictors. Type "cov(fit$residuals, galton$parent)" to see if

| it's close to 0.

> cov(fit$residuals, galton$parent)

[1] -1.790153e-13

| Perseverance, that's the answer.

|========================= | 22%

| As shown algebraically in the slides, the equations for the intercept and slope are found by supposing a change is made to

| the intercept and slope. Squaring out the resulting expressions produces three summations. The first sum is the original

| term squared, before the slope and intercept were changed. The third sum totals the squared changes themselves. For

| instance, if we had changed fit’s intercept by adding 2, the third sum would be the total of 928 4’s. The middle sum is

| guaranteed to be zero precisely when the two equations (the conditions on the residuals) are satisfied.

...

|============================= | 25%

| We'll verify these claims now. We've defined for you two R functions, est and sqe. Both take two inputs, a slope and an

| intercept. The function est calculates a child's height (y-coordinate) using the line defined by the two parameters,

| (slope and intercept), and the parents' heights in the Galton data as x-coordinates.

...

|================================ | 28%

| Let "mch" represent the mean of the galton childrens' heights and "mph" the mean of the galton parents' heights. Let "ic"

| and "slope" represent the intercept and slope of the regression line respectively. As shown in the slides and past

| lessons, the point (mph,mch) lies on the regression line. This means

1: mph = ic + slope\*mch

2: mch = ic + slope\*mph

3: I haven't the slightest idea.

Selection: 2

| You are quite good my friend!

|==================================== | 31%

| The function sqe calculates the sum of the squared residuals, the differences between the actual children's heights and

| the estimated heights specified by the line defined by the given parameters (slope and intercept). R provides the

| function deviance to do exactly this using a fitted model (e.g., fit) as its argument. However, we provide sqe because

| we'll use it to test regression lines different from fit.

...

|======================================== | 34%

| We'll see that when we vary or tweak the slope and intercept values of the regression line which are stored in fit$coef,

| the resulting squared residuals are approximately equal to the sum of two sums of squares - that of the original

| regression residuals and that of the tweaks themselves. More precisely, up to numerical error,

...

|=========================================== | 38%

| sqe(ols.slope+sl,ols.intercept+ic) == deviance(fit) + sum(est(sl,ic)ˆ2 )

...

|=============================================== | 41%

| Equivalently, sqe(ols.slope+sl,ols.intercept+ic) == sqe(ols.slope, ols.intercept) + sum(est(sl,ic)ˆ2 )

...

|================================================== | 44%

| The left side of the equation represents the squared residuals of a new line, the "tweaked" regression line. The terms

| "sl" and "ic" represent the variations in the slope and intercept respectively. The right side has two terms. The first

| represents the squared residuals of the original regression line and the second is the sum of squares of the variations

| themselves.

...

|====================================================== | 47%

| We'll demonstrate this now. First extract the intercept from fit$coef and put it in a variable called ols.ic . The

| intercept is the first element in the fit$coef vector, that is fit$coef[1].

| Now we'll show you some R code which generates the left and right sides of this equation. Take a moment to look it over.

| We've formed two 6-long vectors of variations, one for the slope and one for the intercept. Then we have two "for" loops

| to generate the two sides of the equation.

#Here are the vectors of variations or tweaks

sltweak <- c(.01, .02, .03, -.01, -.02, -.03) #one for the slope

ictweak <- c(.1, .2, .3, -.1, -.2, -.3) #one for the intercept

lhs <- numeric()

rhs <- numeric()

#left side of eqn is the sum of squares of residuals of the tweaked regression line

for (n in 1:6) lhs[n] <- sqe(ols.slope+sltweak[n],ols.ic+ictweak[n])

#right side of eqn is the sum of squares of original residuals + sum of squares of two tweaks

for (n in 1:6) rhs[n] <- sqe(ols.slope,ols.ic) + sum(est(sltweak[n],ictweak[n])^2)

| We'll demonstrate this now. First extract the intercept from fit$coef and put it in a variable called ols.ic . The

| intercept is the first element in the fit$coef vector, that is fit$coef[1].

> ols.ic <- fit$coeff[1]

| Not quite right, but keep trying. Or, type info() for more options.

| Type "ols.ic <- fit$coef[1]" at the R prompt.

> ols.ic <- fit$coef[1]

| You nailed it! Good job!

|========================================================== | 50%

| Now extract the slope from fit$coef and put it in the variable ols.slope; the slope is the second element in the fit$coef

| vector, fit$coef[2].

> ols.slope <- fit$coef[2]

| Perseverance, that's the answer.

|============================================================= | 53%

| Now we'll show you some R code which generates the left and right sides of this equation. Take a moment to look it over.

| We've formed two 6-long vectors of variations, one for the slope and one for the intercept. Then we have two "for" loops

| to generate the two sides of the equation.

Error in editor(file = file, title = title) :

argument "name" is missing, with no default

| Leaving swirl now. Type swirl() to resume.

> options(editor = "internal")

> swirl()

| Welcome to swirl! Please sign in. If you've been here before, use the same name as you did then. If you are new, call

| yourself something unique.

What shall I call you? rich

| Would you like to continue with one of these lessons?

1: Regression Models Residuals

2: No. Let me start something new.

Selection: 1

| Now we'll show you some R code which generates the left and right sides of this equation. Take a moment to look it over.

| We've formed two 6-long vectors of variations, one for the slope and one for the intercept. Then we have two "for" loops

| to generate the two sides of the equation.

...

|================================================================= | 56%

| Subtract the right side, the vector rhs, from the left, the vector lhs, to see the relationship between them. You should

| get a vector of very small, almost 0, numbers.

> lhs-rhs

[1] 1.264198e-09 2.527486e-09 3.801688e-09 -1.261469e-09 -2.522938e-09 -3.767127e-09

| You nailed it! Good job!

|==================================================================== | 59%

| You could also use the R function all.equal with lhs and rhs as arguments to test for equality. Try it now.

> all.equal(rhs,lhs)

[1] TRUE

| That's a job well done!

|======================================================================== | 62%

| Now we'll show that the variance in the children's heights is the sum of the variance in the OLS estimates and the

| variance in the OLS residuals. First use the R function var to calculate the variance in the children's heights and store

| it in the variable varChild.

> varChild <- var(child)

Error in is.data.frame(x) : object 'child' not found

> varChild <- var(child, dalton)

Error in is.data.frame(x) : object 'child' not found

> ?var

> varChild <- var(galton$child)

| Perseverance, that's the answer.

|=========================================================================== | 66%

| Remember that we've calculated the residuals and they're stored in fit$residuals. Use the R function var to calculate the

| variance in these residuals now and store it in the variable varRes.

> var(fit$residuals)

[1] 5.005688

| You're close...I can feel it! Try it again. Or, type info() for more options.

| Type "varRes <- var(fit$residuals)" at the R prompt.

> varRes <- var(fit$residuals)

| You got it right!

|=============================================================================== | 69%

| Recall that the function "est" calculates the estimates (y-coordinates) of values along the regression line defined by the

| variables "ols.slope" and "ols.ic". Compute the variance in the estimates and store it in the variable varEst.

> varEst <- var(est)

Error: is.atomic(x) is not TRUE

> varEst <- var(est(ols.slope, ols.ic))

| Excellent work!

|=================================================================================== | 72%

| Now use the function all.equal to compare varChild and the sum of varRes and varEst.

> all.equal(varRes, varEst)

[1] "Mean relative difference: 0.7334351"

| Try again. Getting it right on the first try is boring anyway! Or, type info() for more options.

| Type "all.equal(varChild,varEst+varRes)" at the R prompt.

> all.equal(varChild, varRes + varEst)

[1] TRUE

| All that practice is paying off!

|====================================================================================== | 75%

| Since variances are sums of squares (and hence always positive), this equation which we've just demonstrated,

| var(data)=var(estimate)+var(residuals), shows that the variance of the estimate is ALWAYS less than the variance of the

| data.

...

|========================================================================================== | 78%

| Since var(data)=var(estimate)+var(residuals) and variances are always positive, the variance of residuals

1: is greater than the variance of data

2: is unknown without actual data

3: is less than the variance of data

| Since var(data)=var(estimate)+var(residuals) and variances are always positive, the variance of residuals

1: is greater than the variance of data

2: is unknown without actual data

3: is less than the variance of data

Selection: 3

| You nailed it! Good job!

|============================================================================================= | 81%

| The two properties of the residuals we've emphasized here can be applied to datasets which have multiple predictors. In

| this lesson we've loaded the dataset attenu which gives data for 23 earthquakes in California. Accelerations are estimated

| based on two predictors, distance and magnitude.

...

|================================================================================================= | 84%

| Generate the regression line for this data. Type efit <- lm(accel ~ mag+dist, attenu) at the R prompt.

> efit <- lm(accel ~ mag+dist, attenu)

| Nice work!

|===================================================================================================== | 88%

| Verify the mean of the residuals is 0.

> mean(efit$residuals)

[1] -1.785061e-18

| All that practice is paying off!

|======================================================================================================== | 91%

| Using the R function cov verify the residuals are uncorrelated with the magnitude predictor, attenu$mag.

> cov(efit$residuals, attenu$mag)

[1] 5.338694e-17

| All that practice is paying off!

|============================================================================================================ | 94%

| Using the R function cov verify the residuals are uncorrelated with the distance predictor, attenu$dist.

> cov(efit$residuals, attenu$dist)

[1] 5.253433e-16

| All that practice is paying off!

|=============================================================================================================== | 97%

| Congrats! You've finished the course on Residuals. We hope it hasn't left a bad taste in your mouth.

### Practice Programming Assignment: swirl Lesson 3: Least Squares Estimation

* Ordinary Least Squares (OLS): minimize the error distance between the data and the regression line.
* Regression line contains the mean of the two sets of data
* Slope of regression line is correlation of the two sets of data multiplied by the ratio of the standard deviations (outcomes over predictors).
* Normalize data by subtracting mean and divide by standard deviation.
  + correlation value of normalized data is the same as original data.
  + slope of the OLS regression line is equal to the correlation (dependent ~ predictor)

myPlot <- function(beta){

y <- galton$child - mean(galton$child)

x <- galton$parent - mean(galton$parent)

freqData <- as.data.frame(table(x, y))

names(freqData) <- c("child", "parent", "freq")

plot(

as.numeric(as.vector(freqData$parent)),

as.numeric(as.vector(freqData$child)),

pch = 21, col = "black", bg = "lightblue",

cex = .15 \* freqData$freq,

xlab = "parent",

ylab = "child"

)

abline(0, beta, lwd = 3)

points(0, 0, cex = 2, pch = 19)

mse <- mean( (y - beta \* x)^2 )

title(paste("beta = ", beta, "mse = ", round(mse, 3)))

}

manipulate(myPlot(beta), beta = manipulate::slider(0.4, .8, step = 0.02))

| 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/)

...

|====== | 5%

| 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.

|============ | 11%

| 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).

...

|================== | 16%

| 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.

...

|======================== | 21%

| 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.

...

|============================== | 26%

| 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).

...

|==================================== | 32%

| 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.

...

|========================================== | 37%

| What RStudio graphics package allows the user to play with the data to see the effects of the changes?

1: manipulate

2: points

3: plot

4: abline

Selection: 1

| Perseverance, that's the answer.

|================================================ | 42%

| 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.

...

|====================================================== | 47%

| Which value of the slope minimizes the mean squared error?

1: .70

2: .44

3: .64

4: 5

Selection: 3

| You're the best!

|============================================================= | 53%

| What was the minimum mse?

1: .64

2: 44

3: .66

4: 5.0

Selection: 4

| You got it right!

|=================================================================== | 58%

| 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.

...

|========================================================================= | 63%

| Use R's function "cor" to compute the correlation between these normalized data sets.

> cor(gpa\_nor, gch\_nor)

[1] 0.4587624

| Your dedication is inspiring!

|=============================================================================== | 68%

| How does this correlation relate to the correlation of the unnormalized data?

1: It is smaller.

2: It is bigger.

3: It is the same.

Selection: 3

| You are doing so well!

|===================================================================================== | 74%

| 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.

> l\_nor <- lm(gch\_nor ~ gpa\_nor)

| You got it!

|=========================================================================================== | 79%

| What is the slope of this line?

1: I have no idea

2: 1.

3: The correlation of the 2 data sets

Selection: 3

| You're the best!

|================================================================================================= | 84%

| 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?

1: correlation(X,Y) \* sd(X)/sd(Y)

2: 1.

3: the same as the original

4: I have no idea

Selection: 1

| Perseverance, that's the answer.

|======================================================================================================= | 89%

| 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.

#plot the original Galton data points with larger dots for more freq pts

y <- galton$child

x <- galton$parent

freqData <- as.data.frame(table(galton$child, galton$parent))

names(freqData) <- c("child", "parent", "freq")

plot(as.numeric(as.vector(freqData$parent)),

as.numeric(as.vector(freqData$child)),

pch = 21, col = "black", bg = "lightblue",

cex = .07 \* freqData$freq, xlab = "parent", ylab = "child")

#original regression line, children as outcome, parents as predictor

abline(mean(y) - mean(x) \* cor(y, x) \* sd(y) / sd(x), #intercept

sd(y) / sd(x) \* cor(y, x), #slope

lwd = 3, col = "red")

#new regression line, parents as outcome, children as predictor

abline(mean(y) - mean(x) \* sd(y) / sd(x) / cor(y, x), #intercept

sd(y) / cor(y, x) / sd(x), #slope

lwd = 3, col = "blue")

#assume correlation is 1 so slope is ratio of std deviations

abline(mean(y) - mean(x) \* sd(y) / sd(x), #intercept

sd(y) / sd(x), #slope

lwd = 2)

points(mean(x), mean(y), cex = 2, pch = 19) #big point of intersection

| Congrats! You've concluded this lesson on ordinary least squares which are truly extraordinary!

### Week 1 Quiz

Quiz: Quiz 1: 10 questions 10/10 correct.

Due on July 17, 2016

1. Consider the data set and weights give below:

x <- c(0.18, -1.54, 0.42, 0.95)

w <- c(2, 1, 3, 1)

Give the value of *μ* that minimizes the least squares equation ∑*ni*=1*wi*(*xi*−*μ*)2

> x <- c(0.18, -1.54, 0.42, 0.95)

> w <- c(2, 1, 3, 1)

> mean(x)

[1] 0.0025

> sum(w \* (x - mean(x))^2)

[1] 3.862994

> x - mean(x)

[1] 0.1775 -1.5425 0.4175 0.9475

> x2 <- (x - mean(x))^2

> x2

[1] 0.03150625 2.37930625 0.17430625 0.89775625

> w \* x2

[1] 0.0630125 2.3793062 0.5229187 0.8977563

> sum(w\*x2)

[1] 3.862994

> x <- c(0.18, -1.54, 0.42, 0.95)

> w <- c(2, 1, 3, 1)

> mean(x)

[1] 0.0025

>

> sum(x\*w)/sum(w)

[1] 0.1471429

> wm <- sum(x\*w)/sum(w)

> sum(w \* (x - wm)^2)

[1] 3.716543

2. Consider the following data set

x <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)

y <- c(1.39, 0.72, 1.55, 0.48, 1.19, -1.59, 1.23, -0.65, 1.49, 0.05)

Fit the regression through the origin and get the slope treating y as the outcome and x as the regressor. (Hint, do not center the data since we want regression through the origin, not through the means of the data.)

> x <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)

> y <- c(1.39, 0.72, 1.55, 0.48, 1.19, -1.59, 1.23, -0.65, 1.49, 0.05)

> lse <- lm(y ~ 0 + x)

> plot(x, y)

> abline(lse)

> summary(lse)

Call:

lm(formula = y ~ 0 + x)

Residuals:

Min 1Q Median 3Q Max

-2.0692 -0.2536 0.5303 0.8592 1.1286

Coefficients:

Estimate Std. Error t value Pr(>|t|)

x 0.8263 0.5817 1.421 0.189

Residual standard error: 1.094 on 9 degrees of freedom

Multiple R-squared: 0.1831, Adjusted R-squared: 0.09238

F-statistic: 2.018 on 1 and 9 DF, p-value: 0.1892

3. Do data(mtcars) from the datasets package and fit the regression model with mpg as the outcome and weight as the predictor. Give the slope coefficient.

> lse\_mpg <- lm(mpg ~ wt, mtcars)

> summary(lse\_mpg)

Call:

lm(formula = mpg ~ wt, data = mtcars)

Residuals:

Min 1Q Median 3Q Max

-4.5432 -2.3647 -0.1252 1.4096 6.8727

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 37.2851 1.8776 19.858 < 2e-16 \*\*\*

wt -5.3445 0.5591 -9.559 1.29e-10 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.046 on 30 degrees of freedom

Multiple R-squared: 0.7528, Adjusted R-squared: 0.7446

F-statistic: 91.38 on 1 and 30 DF, p-value: 1.294e-10

> plot(mtcars$wt, mtcars$mpg)

> abline(lse\_mpg)

> lse\_mpg$coefficients

(Intercept) wt

37.285126 -5.344472

4. Consider data with an outcome (Y) and a predictor (X). The standard deviation of the predictor is one half that of the outcome. The correlation between the two variables is .5. What value would the slope coefficient for the regression model with *Y* as the outcome and *X* as the predictor?

slope = cor(x,y) \* (sdY/sdX) = 0.5 \*(sdY / 0.5sdY) = 1

5. Students were given two hard tests and scores were normalized to have empirical mean 0 and variance 1. The correlation between the scores on the two tests was 0.4. What would be the expected score on Quiz 2 for a student who had a normalized score of 1.5 on Quiz 1?

cor(x,y) = 0.4 # this is the slope

y = f(x) = 0.4\*x = 0.4 \* 1.5 = 0.6

6. Consider the data given by the following

x <- c(8.58, 10.46, 9.01, 9.64, 8.86)

What is the value of the first measurement if x were normalized (to have mean 0 and variance 1)?

> e <- c(8.58, 10.46, 9.01, 9.64, 8.86)

> e\_norm <- e - mean(e)

> e\_norm <- e\_norm/sd(e)

> e\_norm

[1] -0.9718658 1.5310215 -0.3993969 0.4393366 -0.5990954

> scale(e)

[,1]

[1,] -0.9718658

[2,] 1.5310215

[3,] -0.3993969

[4,] 0.4393366

[5,] -0.5990954

attr(,"scaled:center")

[1] 9.31

attr(,"scaled:scale")

[1] 0.7511325

7. Consider the following data set (used above as well). What is the intercept for fitting the model with x as the predictor and y as the outcome?

x <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)

y <- c(1.39, 0.72, 1.55, 0.48, 1.19, -1.59, 1.23, -0.65, 1.49, 0.05)

> p <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)

> q <- c(1.39, 0.72, 1.55, 0.48, 1.19, -1.59, 1.23, -0.65, 1.49, 0.05)

> lse\_pq <- lm(y ~ x)

> summary(lse\_pq)

Call:

lm(formula = y ~ x)

Residuals:

Min 1Q Median 3Q Max

-2.1640 -0.5818 0.2010 0.6669 1.1928

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.567 1.252 1.252 0.246

x -1.713 2.105 -0.814 0.439

Residual standard error: 1.061 on 8 degrees of freedom

Multiple R-squared: 0.07642, Adjusted R-squared: -0.03903

F-statistic: 0.662 on 1 and 8 DF, p-value: 0.4394

> lse\_pq

Call:

lm(formula = y ~ x)

Coefficients:

(Intercept) x

1.567 -1.713

8. You know that both the predictor and response have mean 0. What can be said about the intercept when you fit a linear regression? Intercept is through 0.

9. Consider the data given by

x <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)

What value minimizes the sum of the squared distances betwe x <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)

y <- c(1.39, 0.72, 1.55, 0.48, 1.19, -1.59, 1.23, -0.65, 1.49, 0.05)en these points and itself?

> z <- c(0.8, 0.47, 0.51, 0.73, 0.36, 0.58, 0.57, 0.85, 0.44, 0.42)

> mean(z)

[1] 0.573

10. Let the slope having fit Y as the outcome and X as the predictor be denoted as *β*1. Let the slope from fitting X as the outcome and Y as the predictor be denoted as *γ*1. Suppose that you divide *β*1 by *γ*1; in other words consider *β*1/*γ*1. What is this ratio always equal to?

beta1 = cov(x,y) / var(x) # not sure this is correct (Vars?)

gamma1 = cov(x,y) / var(y)

beta1 / gamma1 = var(Y) / var(X)

beta1 = cor(x,y) \* sd(Y)/sd(X) # slope of Y outcome, X predictor

gamma1 = cor(x,y) \* sd(X)/sd(Y) # slope of X outcome, Y predictor

beta1 / gamma1 = ( cor(x,y) \* sd(Y)/sd(X) ) / ( cor(x,y) \* sd(X)/sd(Y) )

= sd(Y)^2 / sd(X)^2

= var(Y) / var(X)

# Week 2: Linear Regression and Multivariate Regression

This week, we will work through the remainder of linear regression and then turn to the first part of multivariable regression.

## Statistical linear regression models

Up to this point, we’ve only considered estimation. Estimation is useful, but we also need to know how to extend our estimates to a population. This is the process of statistical inference. Our approach to statistical inference will be through a statistical model. At the bare minimum, we need a few distributional assumptions on the errors. However, we’ll focus on full model assumptions under Gaussianity.

### Statistical Linear Regression Models

* Least squares is an estimation tool
* Statistical inference wants to draw conclusions about the population from the data on hand
* One way to do this is to add Gaussian error to the linear regression probabilistic model
  + errors are assumed iid with N(0, sigma^2)
  + the error variance is around the linear regression model, so it is less than variance of the model
  + error variance accounts for population variance of y at a given value of x.
* Expected value and Variance of population are the values of interest and what we use statistical inference to estimate.

### Interpreting Coefficients

* ***intercept***: expected value of y when regressor value (x) is equal to zero.
  + not always of interest in the study
  + new regression line with same slope, shift regressor by a constant value (a), gives a new intercept
  + often a = X-mean so that the intercept is interpreted as the expected response at the average value of X regressor
* ***slope***: expected change in response for a 1-unit change in predictor.
  + easy to understand
  + scaling or changing the units of X by constant (a) effects the slope by dividing by same value (a)

### Linear Regression for Prediction

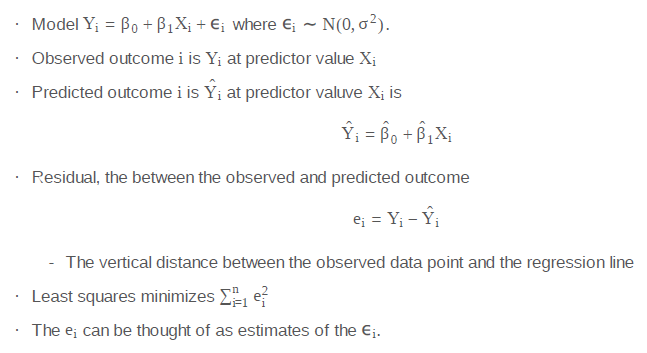
* get a prediction for Y = b-hat-not + b-hat-1\*X
  + there may be an uncertainty with prediction intervals (over extrapolation from data)
* ***See the R markdown file StatisticalInferenceRegressionModels.Rmd***

## Residuals

Residuals represent variation left unexplained by our model. We emphasize the difference between residuals and errors. The errors unobservable true errors from the known coefficients, while residuals are the observable errors from the estimated coefficients. In a sense, the residuals are estimates of the errors.

### Residuals

* Start out with a lot of variation in the price
* By considering mass with price, there is much less variation in price (around the regression line)
* the residual variation is the remaining variation around the regression model
  + can evaluate the model for poor fit
* Model for the residuals:
  + assume Gaussian errors ~ N(0, sigma^2)
  + the errors of the estimates is the estimates for the error value in the population model



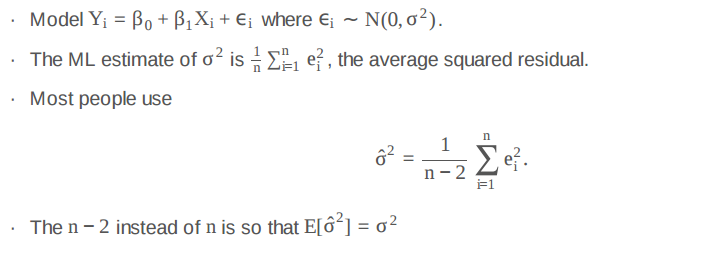
* Y-i-hat = predicted values from the linear regression model
  + uses the
* Residual = distance between the observed value and the predicted value
* Least squares minimizes the sum of squared errors.
* The residual can be thought of as estimates for the error term in the model, HOWEVER
  + you can increase residuals by adding irrelevant regressors into the model formation
* Properties of residuals:
  + expected value of zero
  + useful to determine poor model fit
  + residuals can be used to model outcome with calibration for regressors (removed?)
  + ***Different: residual variation (variation after removing the predictor) from systematic variation (variation explained by the regression model)***

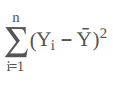
### Residuals, Coding Example

* Regression model is an estimate, may get meaningful information about trends without a precise model
  + for example a linear regression model for a non-linear system over a limited range.
* See the Rmd doc for the examples in R.

### Residual Variance

* Residual variation is the variation around the regression line.
* Sum of squares of the observed residuals is observed variance and is the expected value for the population variance.
* Degress of Freedom: in case of the regression line with two parameters (slope, intercept) you do not really have N observations (or degress of freedom). You are constrained to N-2 observations, since the model of a line with slope, intercept has two parameters that can always be determined from the data, thus taking two degrees of freedom away (N-2)

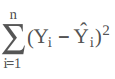




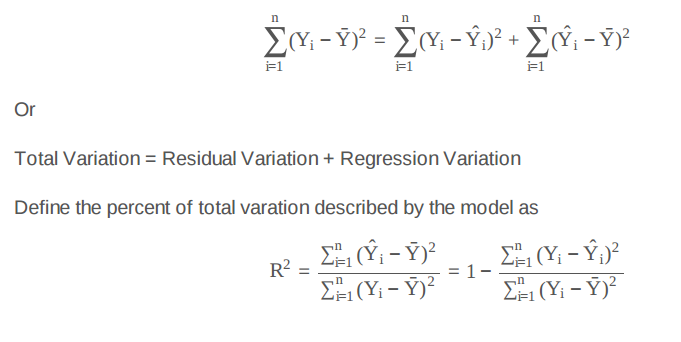
* ***Total Variability*** of the data is average squared deviation of the data around the mean.



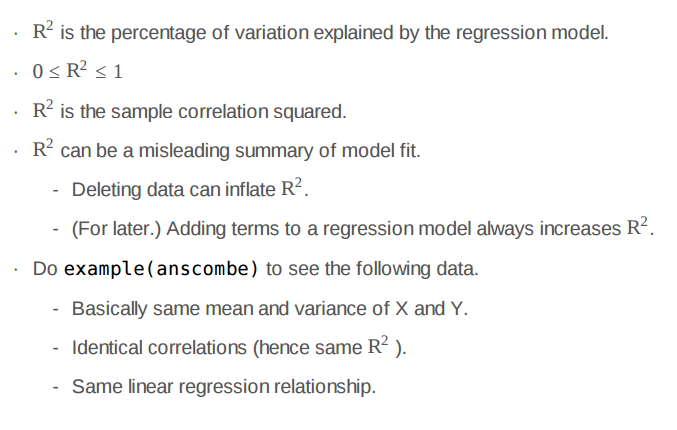
* The ***Regression Variability*** is the component of the variability that is explained by the regression model.



* The ***Residual Variability*** is the component of variability that is not explained by the regression model.



* ***R-squared***: a statistic that shows the proportion of total variability that is explained by the model.
  + This can be calculated with knowing the variability of the residuals.
  + R = the correlation Cor(X,Y)



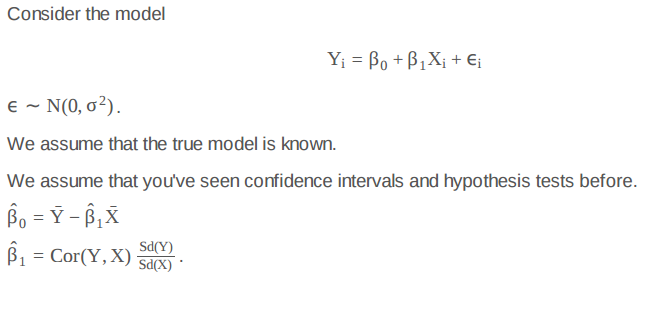
## Inference in regression

Inference is the process of drawing conclusions about a population using a sample. In statistical inference, we must account for the uncertainty in our estimates in a principled way. Hypothesis tests and confidence intervals are among the most common forms of statistical inference.

These statements apply generally, and, of course, to the regression setting that we've been studying. In the next few lectures, we'll cover inference in regression where we make some Gaussian assumptions about the errors.

### Inference in Regression

* The regression model



* Statistics have some general properties (within assumptions):
  + normally distributed (or Student T for a finite number of samples used to infer a population estimate from sample estimate)
  + Can be used to perform hypothesis testing
  + Can be used to create a confidence interval where the quantile is from a normal or Student T distribution
* For regression with iid sampling assumptions and normal errors, the inferences are similar to before
  + do not cover asymptotics for regression here…
* Need more variance in predictor to get lower variance in the outcome
  + close ball of samples doesn’t force the regression line to a slope very strongly (error higher)
* In practice, replace sigma by the estimate for sigma

### Coding Example

* See Rmd file

### Prediction

* See Rmd file

## For the project

### Really, really quick intro to knitr

You need to know a little bit of knitr. In this video, which you may have to refer back to when you start the project, will get you started on knitr.

* r options
  + cache = TRUE, store the calculations
  + eval = FALSE, don’t evaluate the code, just show it.
  + echo = FALSE, don’t show the code or results
  + results = hide, hide the results

## Practical R Exercises in swirl

### Practice Programming Assignment: swirl Lesson 1: Residual Variation

Discussion

* residual variation is variation remaining after systematic variation explained by the model
* given the model, Maximum Likelihood Estimate of variance of the random error is average squared residual error.
* unbiased estimate is given by accounting for degrees of freedom
  + linear model with one predictor has two parameters (intercept and slope)
  + intercept is really coefficient of special regressor with the same value, 1, at every sample.
* divide by n-2
* Total Variation = Residual Variation + Regression Variation
* sum(Yi – mean(Yi))^2 = sum((Yi – Yi-hat)^2 ) + sum( (Yi-hat – mean(Yi) )^2 )
* The term R^2 represents the percent of total variation described by the model, the regression variation (the term we didn't ask about in the preceding multiple choice questions).
* R-squared = (correlation )^2

### Practice Programming Assignment: swirl Lesson 2: Introduction to Multivariable Regression

* The regression in one variable given by lm(child ~ parent, galton) really involves two regressors, the variable, parent, and a regressor of all ones.
  + intercept is really coefficient of special regressor with the same value, 1, at every sample.
* Subtracting the means to eliminate the intercept is a special case of a general technique which is sometimes called Gaussian Elimination.
  + general technique is to pick one regressor and to replace all other variables by the residuals of their regressions against that one.
  + this difference is a residual of regression against the constant, 1
* The mean of a variable is the coefficient of its regression against the constant, 1.
  + subtracting the mean is equivalent to replacing a variable by the residual of its regression against 1.

# Regress the given variable on the given predictor,

# suppressing the intercept, and return the residual.

regressOneOnOne <- function(predictor, other, dataframe){

# Point A. Create a formula such as Girth ~ Height -1

formula <- paste0(other, " ~ ", predictor, " - 1")

# Use the formula in a regression and return the residual.

resid(lm(formula, dataframe))

}

# Eliminate the specified predictor from the dataframe by

# regressing all other variables on that predictor

# and returning a data frame containing the residuals

# of those regressions.

eliminate <- function(predictor, dataframe){

# Find the names of all columns except the predictor.

others <- setdiff(names(dataframe), predictor)

# Calculate the residuals of each when regressed against the given predictor

temp <- sapply(others, function(other)regressOneOnOne(predictor, other, dataframe))

# sapply returns a matrix of residuals; convert to a data frame and return.

as.data.frame(temp)

}

* Suppose we were given a multivariable regression problem involving an outcome and N regressors, where N > 1. Using only single-variable regression, how can the problem be reduced to a problem with only N-1 regressors?
  + Pick any regressor and replace the outcome and all other regressors by their residuals against the chosen one.

### Practice Programming Assignment: swirl Lesson 3: MultiVar Examples

* See notes

## Week 2 Quiz

### Quiz: Quiz 2

10 questions

Due July 24, 2016

* See R markdown file.

# Week 3: Multivariable Regression, Residuals, & Diagnostics

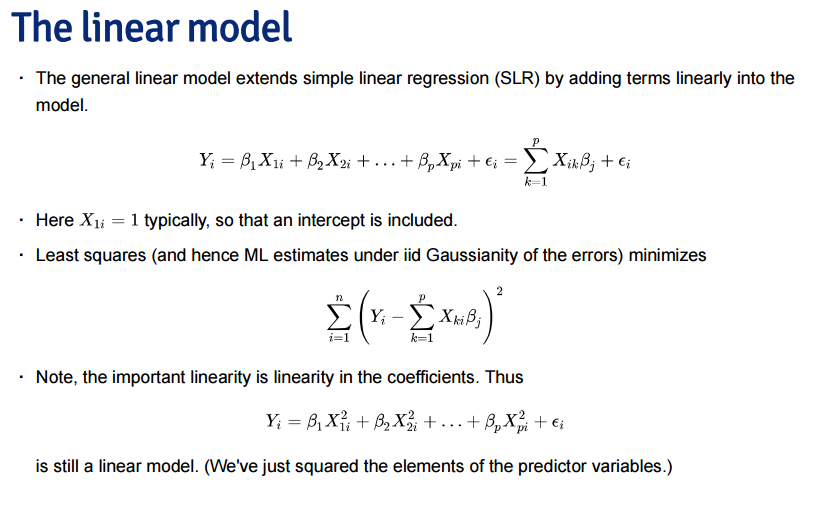
This week, we'll build on last week's introduction to multivariable regression with some examples and then cover residuals, diagnostics, variance inflation, and model comparison.

## Multivariable regression

We now extend linear regression so that our models can contain more variables. A natural first approach is to assume additive effects, basically extending our line to a plane, or generalized version of a plane as we add more variables. Multivariable regression represents one of the most widely used and successful methods in statistics.

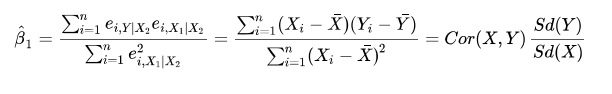
### Multivariable Regression part I

* Example: relationship between breath mint usage and pulmonary function (FEV)
* finding a significant regression relationship can be skeptical…what other variables that might explain the correlation relationship
  + Problem of multiplicity
* maybe condition on smoking status to compare like with like and see if relationship is still significant
* In other words, look for response relationship to one variable while holding the others constant
* Example 2: insurance company interested in how last years claims can predict time in the hospital this year
  + enormous amount of data to predict one variable number
  + simple linear regression is not suitable
  + How can one generalize SLR to incorporate lots of regressors for prediction
  + What are consequences of adding regressors or omitting regressors
* Multivariable Regression is a good start for prediction even though there may be other methods
  + machine learning methods did bring some more information (but not always to degree that SLR works)
* The Linear Model
  + linearity is linearity in the coefficients (you might square the predictor variables, the coefficients are still linear)
  + coefficient linearity is what defines an SLR



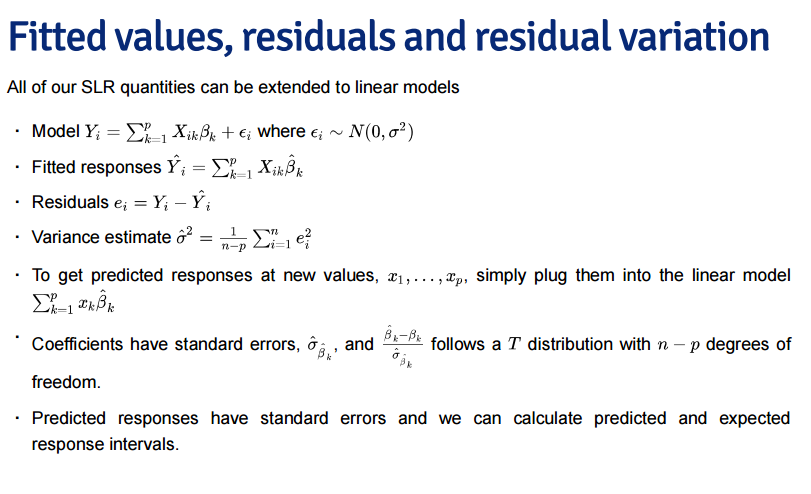
### Multivariable Regression part II

* Watch the video
* Mulltivariable regression coefficient “removes” linear effects of response and other regression variables
  + this is the movement along orthogonal axes of each variable
* ***adjusted*** sense of the model
  + More generally, multivariate regression estimates are exactly those having removed the linear relationship of the other variables from both the regressor and response
* When one of the regressors is intercept (=1) then the relationship looks like this:



### Multivariable Regression Continued

* demonstration in R
* see Rmd notes
* ***Interpretation of multivariable linear regression coefficient is the expected change in the response per unit change in the regressor, holding all other regressors fixed.***



* ***Overarching take-aways:***
  + Linear models are single most important applied statistical and machine learning technique
  + Linear models are parsimonious, easy to understand
  + Things you can accomplish with linear models:
    - decompose a signal into its harmonics (Discrete Fourier Transform)
    - flexibility fit complicated functions
    - fit factor variables as predictors (ANOVA, ANCOVA are special cases)
    - uncover complex multivariate relationships with the response
    - build accurate prediction models

## Multivariable regression tips and tricks

### Multivariable Regression Examples part I

* see the Rmd file
* Model selection will change your estimates
  + selection of which variables you are going to use
  + For example, all the variables give different coefficient for Agriculture than one with only Agriculture
  + Simpson’s Paradox – change in sign of the individual factors than when all are modeled

### Multivariable Regression Examples part II

* Dummy Variables are smart (uses binary values for some factor)
  + two-factor linear model can be used for a treatment group (1) and control group (0)
  + the coefficient is interpreted as the ***change in the mean*** of the treatment group compared to the control group
  + t-test for the coefficient is same as a two-group t-test where you assume a common variance.
* More than two levels
  + Republican, Democrat, Independents
  + if Republican is one coefficient and Democrat another, then the intercept is coefficient for Independents
  + Independents would be the ***reference level***, important for interpretation
  + Interpretation is comparing Republicans to Independents and Democrats to Independents
  + B1 – B2 compares Republicans to Democrats
* What your interpretation is depends on the reference level and inclusion or omission of an intercept.
* Re-leveling gives a different reference level (and interpretation).
* Other thoughts:
  + counts are bounded by 0 – violates assumption of normality of errors.
    - some counts are near zero, so assumption and the intent are both violated
  + Variance does not appear to be constant
    - the histogram distributions are skewed (violin plots)
    - Unequal variances is called ***heteroskedasticity***
  + Taking logs of counts could help (there are ‘zeros’ so using log(count + 1) )
  + Poisson GLMs are better for counts.

### Multivariable Regression Examples part III

* ANCOVA example
* see the Rmd file

### Multivariable Regression Examples part IV

* See the Rmd file for examples.

## Adjustment

Adjustment, is the idea of putting regressors into a linear model to investigate the role of a third variable on the relationship between another two. Since it is often the case that a third variable can distort, or confound if you will, the relationship between two others.

As an example, consider looking at lung cancer rates and breath mint usage. For the sake of completeness, imagine if you were looking at forced expiratory volume (a measure of lung function) and breath mint usage. If you found a statistically significant regression relationship, it wouldn’t be wise to rush off to the newspapers with the headline “Breath mint usage causes shortness of breath!”, for a variety of reasons. First off, even if the association is sound, you don’t know that it’s causal. But, more importantly in this case, the likely culprit is smoking habits. Smoking rates are likely related to both breath mint usage rates and lung function. How would you defend your finding against the accusation that it’s just variability in smoking habits?

If your finding held up among non-smokers and smokers analyzed separately, then you might have something. In other words, people wouldn’t even begin to believe this finding unless it held up while holding smoking status constant. That is the idea of adding a regression variable into a model as adjustment. The coefficient of interest is interpreted as the effect of the predictor on the response, holding the adjustment variable constant.

In this lecture, we’ll use simulation to investigate how adding a regressor into a model addresses the idea of adjustment.

### Adjustment Examples

* The coefficient of interest is interpreted as the effect of the predictor on the response, holding the adjustment variable constant.
* Example: two group variable, A-B test
* See the Rmd document.

## Residuals again

Recall from before that the vertical distances between the observed data points and the fitted regression line are called residuals. We can generalize this idea to the vertical distances between the observed data and the fitted surface in multivariable settings.

### Residuals and Diagnostics part I

* Start with linear model
* Assume that errors are iid (normal distribution)
* ***Leverage*** - how far from the center of mass.
* ***Influence*** - how far from the linear relationship.

### Residuals and Diagnostics part II

* Calling something an outlier is vague.
  + Rare or spurious processes
  + If it is not important – get rid of it. If it is important – keep it.
* R has influence measures to determine ‘outlier-ness’ of data.
* How do I use all of these things?
  + Be wary of simplistic rules for diagnostic plots and measures. The use of these tools is context specific. It's better to understand what they are trying to accomplish and use them judiciously.
  + Not all of the measures have meaningful absolute scales. You can look at them relative to the values across the data.
  + They probe your data in different ways to diagnose different problems.
  + Patterns in your residual plots generally indicate some poor aspect of model fit. These can include:
    - Heteroskedasticity (non constant variance).
    - Missing model terms.
    - Temporal patterns (plot residuals versus collection order).
  + Residual QQ plots investigate normality of the errors.
  + Leverage measures (hat values) can be useful for diagnosing data entry errors.
* Influence measures get to the bottom line, 'how does deleting or including this point impact a
* particular aspect of the model'.

### Residuals and Diagnostics part III

* See video and Rmd file.

## Model selection

These lectures represent a challenging question: ***“How do we chose what variables to include in a regression model?”***. Sadly, no single easy answer exists and the most reasonable answer would be ***“It depends.”.*** These concepts bleed into ideas of machine learning, which is largely focused on high dimensional variable selection and weighting. In the following lectures we cover some of the basics and, most importantly, the ***consequences of over- and under-fitting a model.***

### Model Selection part I

* Multivariable regression is a little different than prediction and machine learning.
* Prediction and machine learning is less focused on interpretability
* Modeling interest is looking for parsimonious, interpretable representations to better understand the underlying phenomena under study.

***A model is a lense thorugh which you look at your data. – Scott*** Zeger (model is neither right nor wrong, only a way to look at the world.)

* good model connects the data to a true, parsimonious relationship.

***There are known knowns. These are things we know that we know. There are unknown knowns. That is to say there are things we know we don’t know. But there are also unknown unknowns. There are things we don’t know we don’t know. - Rumsfeldian Triplet***

* unknown unknowns – regressors that we don’t know about that we should have included in the model.
* known unknowns – regressors we would like to include in model but don’t have.
* known knowns – regressors we have and know we should check to include in the model.

**General Rules**

* Omitting variables results in bias in coefficients of interest.
  + Randomize treatments, tries to uncorrelate treatment indicator
* include variables that we should not have increases the standard errors of the regression variables
  + increases the bias of the model
  + Randomization tries to prevent this from happening by chance or unintentionally.
  + If there are too many unobserved confounding variables, randomization won’t help
  + Don’t idly throw models into the model.
  + A bunch of standard normal variables will not bias the model, but will increase the actual standard errors.
* R^2 increases monotonically as more regressors are included
  + the more regressors the better the R^2 looks!
* SSE decreases monotoically as more regressors are included.

### Model Selection part II

* ***Variance Inflation***
  + include unimportant regressors
  + unimportant regressors don’t bring in new information, may depend on another variable.
  + Do not put highly correlated variables in model unnecessarily! Increases actual Variance.
* ***Variance Inflation Factor (VIF)***
  + comparison of the variance inflation from what you have vs. ideal setting of orthogonal correlation matrix.
* Residual Variance estimation
  + if we ***underfit*** (omit important variables) the model then residual variance is biased
  + if we ***overfit*** the model then the residual variance is unbiased
  + if we overfit the model then the variance of the variance is biased (less accurate model)
* ***Principal Component Analysis*** on covariates are often useful for reducing complex covariate spaces.
* Good experimental design can eliminate a complex model search at analysis.
  + often control of the design is limited
  + Field blocking for two types of seeds (take out factor of the field soil, slope, sun, etc.)
* One interesting automated model search technique is ***nested variables***
  + nested likelihood ratio tests
  + each model contains the terms of each previous model (+ others)
  + See Rmd examples.

### Model Selection part III

* All Models are Wrong, Some Models are Useful. – George Box
* Useful models can be used as a lens to learn something useful from your data set.

## Practical R Exercises in swirl Part 3

### Practice Programming Assignment: swirl Lesson 1: MultiVar Examples2

* Completed 8/9/2016
* notes: swirl\_MultiVar\_Examples2.md

### Practice Programming Assignment: swirl Lesson 2: MultiVar Examples3

### Practice Programming Assignment: swirl Lesson 3: Residuals Diagnostics and Variation

## Week 3 Quiz

### Quiz: Quiz 3

7 questions: 6/7 for 85% score on 8/9/2016

Overdue July 31, 2016

# Week 4: Logistic Regression and Poisson Regression

This week, we will work on generalized linear models, including binary outcomes and Poisson regression.

### GLMs

Generalized linear models (GLMs) were a great advance in statistical modeling. The original manuscript with the GLM framework was from Nelder and Wedderburn in 1972. in the Journal of the Royal Statistical Society. The McCullagh and Nelder book is the famous standard treatise on the subject.

Recall linear models. Linear models are the most useful applied statistical technique. However, they are not without their limitations. ***Additive response models don’t make much sense if the response is discrete, or strictly positive.*** Additive error models often don’t make sense, for example, if the outcome has to be positive. Transformations, such as taking a cube root of a count outcome, are often hard to interpret. In addition, there’s value in modeling the data on the scale that it was collected. Particularly interpretable transformations, natural logarithms in specific, aren’t applicable for negative or zero values.

The generalized linear model is family of models that includes linear models. By extending the family, it handles many of the issues with linear models, but at the expense of some complexity and loss of some of the mathematical tidiness. A GLM involves three components

* An exponential family model for the response.
* A systematic component via a linear predictor.
* A link function that connects the means of the response to the linear predictor.

The three most famous cases of GLMs are: linear models, binomial and binary regression and Poisson regression. We’ll go through the GLM model specification and likelihood for all three. For linear models, we’ve developed them previously. The next two modules will be devoted to binomial and Poisson regression. We’ll only focus on the most popular and useful link functions.

* Review: Linear Models
  + most useful applied statistical technique
  + Limitations
    - assumption of additivity is hard with binary data
    - transformations often hard to interpret (logs might work, others don’t so well)
    - There is value in modeling the data in the scale it was collected
    - transformations (natural logarithm) aren’t applicable to negative or zero values
    - problems when error Gaussian has large probability of being negative
* Generalized Linear Models
  + introduced in 1972 paper by Nelder and Wedderburn to Royal Society of
  + GLM has three components
    - an ***exponential family*** model for response (random component)
    - the linear predictor – what we are modeling (systematic component)
    - connect the response to the linear predictor (link function)
* Example:
  + Y ~ N(mu, sigma^2) # guassian distribution is an exponential family distribution
  + linear predictor, eta
  + link function is such that mu gives the linear predictor
    - for linear models, the link function returns mu, an identity type function, mu and linear predictor are equal
* Logistic Regression
  + useful for binomial factor (0 or 1), positive and bounded by number of flips
  + assume that Y\_i ~ Bernoulli(mu\_i) so that the E[Y\_i] = mu where mu is between 0 and 1
  + binary values don’t come from normal distribution (coin flips)
  + linear predictor is same as linear regression
  + link function is the log of the odds
    - same as the natural log odds referred to as the ***logit***
  + transformation of the means of the distribution, not the Y of the data itself
  + can go backwards
  + we try to maximize the likelihood to get the estimate of the parameter
* Poisson Regression
  + useful for modeling count data
    - positive and unbounded
  + Y\_i ~ Poisson(mu\_i) so that E[Y\_i] = mu\_i
  + Linear predictor is the same (family of exponentials)
  + Link function is g(mu) = eta = log(mu)
  + Going backwards is e^eta to get back to mu
* All these cases share the link function
  + the maximum likelihood has a certain case that is maximum
* About Variances – important for GLMs
  + for linear model has assumption of constant variance
  + for Bernoulli case the variance changes with the number of data
  + Poisson variance definition is similar with changing mean with new data
  + Standard Options in R (quasi blank options in R)
    - poisson and quasipoisson (slightly more flexible variance model)
* Overveiw
  + Normal equations have to be solved iteratively (resulting in coefficients for each iteration)
  + Predictor responses obtained with Eta equation
  + Coefficients in GLM are the expected change in the response with a unit change in the regressor holding all other regressors constant.
  + We lose the closed form inference (no t-scores, etc.)
    - another body of mathematics to calculate a p-value
  + larger sample sizes required for GLMs since they are based on asymptotic

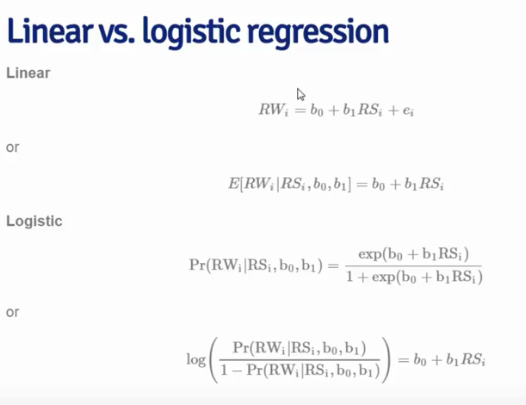
## Logistic Regression

Binary GLMs come from trying to model outcomes that can take only two values. Some examples include: survival or not at the end of a study, winning versus losing of a team and success versus failure of a treatment or product. Often these outcomes are called Bernoulli outcomes, from the Bernoulli distribution named after the famous probabilist and mathematician.

If we happen to have several exchangeable binary outcomes for the same level of covariate values, then that is binomial data and we can aggregate the 0’s and 1’s into the count of 1’s. As an example, imagine if we sprayed insect pests with 4 different pesticides and counted whether they died or not. Then for each spray, we could summarize the data with the count of dead and total number that were sprayed and treat the data as binomial rather than Bernoulli.

### Logistic Regression part I

* For ***binary*** data, two levels (0 or 1)
  + survival analysis
  + wins/loses
  + success/failure
* Called Binary or Bernoulli
* ***Binomial*** random variables are handled by binary logistical regression when covariance matrix is constant
* Example: Raven win/loss data – does Raven score predict wins or loss
  + fiting binary data to a linear regression model is not very good – but can be a first, quick start
  + Next try modeling the odds:
    - model probability the Ravens win
    - ***odds = (probability of win) / (1 – probability of win)***
    - ***probability of win = (odds ) / (1 + odds)***
  + Log odds = log( of the odds ) # called the logit



* Expected value of binomial is the probability of the outcome.
* The log of the odds is the linear regression relationship
* **Interpreting Logistic Regression**
  + log odds are the bottom equation in slide above
  + ***b0*** is log odds of a ravens win if they score zero points
  + ***b1*** is log odds ratio of win probability for each point scored (compared to zero points)
  + ***exp(b1)*** is odds ratio of win probability for each point scored (compared to zero points)
* **Natural Logarithm**: <https://en.wikipedia.org/wiki/Natural_logarithm>
  + The **natural logarithm** of a number is its [logarithm](https://en.wikipedia.org/wiki/Logarithm) to the [base](https://en.wikipedia.org/wiki/Base_(exponentiation)) of the mathematical constant [*e*](https://en.wikipedia.org/wiki/E_(mathematical_constant)), where *e* is an [irrational](https://en.wikipedia.org/wiki/Irrational_number) and [transcendental](https://en.wikipedia.org/wiki/Transcendental_number) number approximately equal to2.718281828459. The natural logarithm of *x* is generally written as ln *x*, log*e* *x*, or sometimes, if the base *e* is implicit, simply log *x*.[[1]](https://en.wikipedia.org/wiki/Natural_logarithm#cite_note-1) [Parentheses](https://en.wikipedia.org/wiki/Parentheses) are sometimes added for clarity, giving ln(*x*), log*e*(*x*) or log(*x*). This is done in particular when the argument to the logarithm is not a single symbol, to prevent ambiguity.
  + The natural logarithm of *x* is the [power](https://en.wikipedia.org/wiki/Exponentiation) to which *e* would have to be raised to equal *x*.
  + The natural logarithm function, if considered as a real-valued function of a real variable, is the inverse function of the exponential function, leading to the identities:
  + Like all logarithms, the natural logarithm maps multiplication into addition:
* Odds
  + idea of a fair game flipping coin with success probability p
  + win you earn X, if you lose you earn Y
  + what should we set X and Y for game to be fair
  + the odds can be interpreted as “how much should you be willing to pay for a p probability of winning a dollar”
    - if p > 0.5, you pay more if you lose than if you win
    - if p < 0.5 you pay less if you lose than if you win
  + horse track will give you the odds against the horse winning, 10/1
    - quickly see the payout for your bet
    - probabilities are set adaptively as the bets come in
    - the money coming in and out is balanced by the house (neutral cash flow)
    - over a long period of watching horses by many people, the market tends to be correct
    - money is made through a transaction fee – the house always wins.

There are a few Rmd notes.

Try the Ravens score experiment in full.

## Poisson Regression

### Count Data

Many data take the form of unbounded count data. For example, consider the number of calls to a call center or the number of flu cases in an area or the number of hits to a web site.

In some cases the counts are clearly bounded. However, modeling the counts as unbounded is often done when the upper limit is not known or very large relative to the number of events.

***If the upper bound is known, the techniques we’re discussing can be used to model the proportion or rate.*** The starting point for most count analysis is the the Poisson distribution.

In the following lectures, we go over some of the basics of modeling count data.

### Poisson Regression part I

* see video and Rmd

### Poisson Regression part II

* see video and Rmd

## Hodgepodge

This lecture is a bit of an mishmash of interesting things that one can accomplish with linear models.

### Hodgepodge

* see video and Rmd file.

## Practical R Exercises in swirl Part 4

### Practice Programming Assignment: swirl Lesson 1: Variance Inflation Factor

* done, notes

### Practice Programming Assignment: swirl Lesson 2: Overfitting and Underfitting

* done, notes

### Practice Programming Assignment: swirl Lesson 3: Binary Outcomes

* done, notes

### Practice Programming Assignment: swirl Lesson 4: Count Outcomes

* done, notes

## Week 4 Quiz

Quiz: Quiz 4

6 questions

Overdue on 8/7/2016

## Course Project

### Peer Graded Assignment: Regression Models Course Project1h

Overdue: 8/7/2016

Submitted on 8/10/2016 at midnight. Grading in progress.

## Review Your Peers: Regression Models Course Project

Due August 10, 11:59 PM PDT

### Post-Course Survey

# ENDNOTES

## Discussion Forum Notes

"Illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials"

So, as Jerome said, we're only concerned about the **distribution of the mean** of 40 exp and not the exponential distribution itself:

In other words:

* random draw - a mean of randomly chosen 40 exponentials
* population - all possible random draws = means of all possible sets [of size 40] of exponentials
* sample - subset of the population = means of 1000 randomly chosen sets [of size 40] of exponentials
* theoretical distribution - a distribution of the population [of all means...] which - by CLT - is approximately normal N(mu, sigma/sqrt(n))
* sample distribution/mean/variance - a distribution/mean/variance of the sample [of the means...]

 3 Upvote

Based on the central limit theorem, the expected variance for the distribution of sample means is sigma^2 / N, not sigma^2. If we're taking samples of 40 exponentials, then N = 40. The expected standard deviation would be sigma / sqrt(N).

In your report, if you've created an array of 1000 means of 40 exponentials you can run **pastecs::stat.desc()** to obtain a variety of descriptive statistics on the array of means, including variance and standard deviation.

Yiannis - here is my understanding:

* if you take ***n*** observations from a population, that's a sample
* You calculate a statistic. In this case, it's the mean
* What you have now is a sample mean or, alternatively, a mean of the sample
* You repeat that process***N*** times. For each calculation, you get a different mean.
* The collection of those means form their own kind of distribution, just like exponentials or binomials or what have you.
* The term for that is sampling distribution or, in plainer English, a distribution of statistics calculated on N samples, and a collection of samples is called a sampling.

Sadly, Statistics needs help from English majors to make it less obtuse.

I think that the term used for the collection of those means/statistics described at your last two steps is **sampling distribution** - correct me if I am wrong. In the assignment script, however, it is called **sample distribution** (if I get it correct), which I think is confusing.

**More Advanced Exploratory Analysis:**

<https://github.com/lgreski/datasciencectacontent/blob/master/markdown/edaInToothGrowthAnalysis.md>

<https://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test>

you recommend the [Shapiro-Wilk Test for Normality](https://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test) and a Q-Q plot. Is it expected that we use these methods? I don't remember going over them in lecture.

All of these statistics are available in the stat.desc() function within the [pastecs package](https://cran.r-project.org/web/packages/pastecs/pastecs.pdf).

I noticed that as well. I've used Q-Q plots before and am using them in both parts of the project. Not hard to use: **qqnorm(x) and qqline(x)** should be all you need. You're right, we didn't cover them in the material, and I'm going to use only Q-Q plot for my work. I would suggest googling for the other if you think you'd like to use that technique.

**Variance Mistake:**

Now that I've completed 6 or so peer reviews, I see where I made a critical mistake in my analysis but don't understand why. The mistake I made is this: I did not use (1/lamba)^2/n for my theoretical variance. I understand now that, given 1/lambda as the standard deviation for the distribution, the variance is (1/lambda)^2. How did the sample size become the divisor? Thanks!

Variance has two nice properties that produce this result:

* Var(A + B) = Var(A) + Var(B) provided that A and B are independent (more technically, that covariance between A and B is nil)
* Var(A/n) = (1/n^2) \* Var(A) for any constant n

In this case, you want Var[ (A1 + A2 + . . . + An) / n ] based on n independent draws from the same population A.

This becomes (1/n^2) \* [ Var(A1) + Var(A2) + . . . + Var(An) ].

And since the expected variance is 1/lambda^2 for each of n draws from population A (all independent), this reduces to (1/n^2) \* [ n\*(1/lambda^2) ] = (1/n) \* (1/lambda^2).

## Reading: Statistical Inference for Data Science

<file:///C:/Users/rich/Downloads/LittleInferenceBook.pdf>

* ***Statistical Inference***: process of generating conclusions about a population from a noisy sample.
  + with statistical inference we are trying to generate new knowledge
  + without statistical inference we are simply living in our data (tacit knowledge only)
* ***Knowledge***: is a familiarity, awareness or understanding of someone or something, such as facts, information, descriptions, or skills, which is acquired through experience or education by perceiving, discovering, or learning.
  + justified true belief
  + scientific knowledge
  + religious knowledge
  + communicating knowledge – transfer of knowledge through symbolism or language
  + situated knowledge – tacit knowledge, first hand knowledge
  + partial knowledge – bounded rationality, incomplete information
  + Intuition – ability to acquire knowledge without inference or use of reason
* ***Parsimony***: using simplest reasonable model to explain complex phenomena
  + Probability models are parsimonious description of the world
  + Probability models are connection between data (sample) and the world (population)
* Concerns with using data to draw general conclusions about population:
  + is the sample representative of the population
  + are there known and observed, known and unobserved, or unknown and unobserved variables that contaminate the conclusions
  + is there systematic bias created by missing data or the experimental design
  + what randomness exists in the data, how to account for it
    - explicit randomness = result of randomization or random sampling
    - implicit randomness = aggregation of many complex unknown processes
  + are we estimating and underlying mechanistic model of phenomena (confounding)
* Goals of Inference (examples)
  + Estimate and quantify the uncertainty of an estimate of population quantity
  + Determine if population quantity is a benchmark value
  + Infer a mechanistic relationship when quantities are measured with noise
  + Determine impact of a policy
  + Discuss probability of an event occurring
* Tools of Statistical Inference
  + ***Randomization***: balancing unobserved variables that may confound inferences
  + ***Random*** ***sampling***: obtain data that is representative of the population (minimize bias)
  + ***Sampling*** ***models***: create a model for the sampling process, most common is “iid”
  + ***Hypothesis*** ***testing***: decision making in the presence of uncertainty
  + ***Confidence*** ***Intervals***: quantify uncertainty in estimation
  + ***Probability*** ***models***: formal connection between data and population of interest
    - often assumed or approximated
  + ***Study*** ***design***: process of designing experiment to minimize biases and variability
  + ***Nonparametric*** ***bootstrapping***: process of using data to, with minimal probability model assumptions, create inferences
  + ***Permutation***: randomization and exchangeability testing, the process of using data permutations to perform inferences
* ***Frequency style inference***: using frequency probability derived from long run proportion of event occurrences
  + decisions based on given data and controlling for tolerance level of error in judgement
* ***Bayesian style inference***: use Bayes representations of beliefs to make inference
  + given prior subjective beliefs and objective information from data, what is the posterior belief?
* **Probability**
  + ***randomness***: any process occurring without apparent deterministic patterns
    - we often treat things as random when they are deterministic (but not sure how)
  + ***probability***: long run proportion of times event occurs in repeated unrelated realizations.
  + Kolmogorov’s 3 rules: a possible random outcome must:
    - assign a number between 0 and 1
    - requires the probability that something occurs as 1
    - required that probability of union of any two sets of outcomes that are ***mutually*** ***exclusive*** (nothing in common) is sum of respective probabilities

# Data Analysis Style Guide (R)

* Download the data
  + check if it exists
  + give location
  + download the file
  + read the file (csv, html)
* Record the analysis architecture and environment with systemInfo()
* Create a data structure
* Explore data
  + sizeof()
  + head()
  + tail()
  + names()
  + str()
  + class()
  + Range of data: summary statistics (min, max, median, std dev)
  + table(data.frame$variable)
* Clean and prepare data for analysis
  + missing values
  + Not available (N/A)
  + not a number (NAN)
* Analysis branches
* Run Script
  + source(“file.R”)

**Download**

if(!file.exists(“./data”)){dir.create(“./data”)}

fileUrl <- “https://data.baltimorecity.gov/api/views/etc”

download.file(fileUrl, destfile=”./data/cameras.csv”, method=”wb”)

cameraData <- read.csv(“./data/cameras.csv”)get

Example to set the column classes:

pollution <- read.csv(“data/avgpm25.csv”, colClasses = c(“numeric”,

“character”, “factor”, “numeric”, “numeric”) )

**Profiling R Code**

**For a block of code:**

# Start the clock!

ptm <- proc.time()

# block of code to evaluate process time

code block

# Stop the clock

proc.time() - ptm

**For a single function:**

# For a single line of R code

system.time( a <- function(x) )

**Introduction to swirl**

* Statistics With R Learning
* Optional for the course.
* Install swirl: install.packages(“swirl”)
* Check version: packageVersion(“swirl”)
* Load swirl: library(swirl)
* Install R Programming course in swirl: install\_from\_swirl(“R Programming”)
* Start swirl: swirl()

Error in editor(file = file, title = title) :

argument "name" is missing, with no default

| Leaving swirl now. Type swirl() to resume.

> options(editor = "internal")

> swirl()

**Data Size Estimate**

* Calculate Memory Reqs
  + NROWS x NCOLS x (8 bytes/numeric) / 2^20 bytes/MB
  + 2^10 MB per GB
  + ***Overhead is about twice as much memory to read in the data frame***

s

* Look at the size of an R object
  + object.size()
  + format(object.size(x), units=”Mb”)

Reproducibility Data

* Record the session info for your architecture and system: **sessionInfo()**

Writing Forumlas in HTML, Posts, RMarkdown using Mathjax

* <https://github.com/lgreski/datasciencectacontent/blob/master/markdown/mathjaxWithGithubMarkdown.md>
* able to write clean formulas

# Git Repository Commands

build a remote repository on GitHub then:

echo "# cleaningdataproject" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/rbmorrison/cleaningdataproject.git

git push -u origin master

To push forked repo if the master has been updated

git add <files>

git commit -m “notes for the commit”

git pull –rebase origin master # there are two dashes

git push -u origin master

To check the current origin master:

* git remote show origin # shows remote origin fetch/push URLs, Head and Remote branches
* git remote -v # shows the origin fetch/push URLs

To fork another repo:

* Log on to GitHub, navigate to the repository.
* In the top-right corner of the page, click Fork.

To clone the repo to work on:

1. Navigate to your fork on GitHub.
2. Copy the URL for the forked repository.
3. git clone and paste the URL

To remove folder/directory or file only from git repository and not from the local try 3 simple steps.

**Steps to remove directory**

git rm -r --cached File-or-FolderName

git commit -m "Removed folder from repository"

git push origin master

# CodeBook Examples

## Example 1: Simple / Text

Getting and Cleaning Data: Course Project

=========================================

Introduction

------------

This repository contains my work for the course project for the Coursera course "Getting and Cleaning data", part of the Data Science specialization.

What follows first are my notes on the original data.

About the raw data

------------------

The features (561 of them) are unlabeled and can be found in the x\_test.txt.

The activity labels are in the y\_test.txt file.

The test subjects are in the subject\_test.txt file.

The same holds for the training set.

About the script and the tidy dataset

-------------------------------------

I created a script called run\_analysis.R which will merge the test and training sets together.

Prerequisites for this script:

1. the UCI HAR Dataset must be extracted and..

2. the UCI HAR Dataset must be availble in a directory called "UCI HAR Dataset"

After merging testing and training, labels are added and only columns that have to do with mean and standard deviation are kept.

Lastly, the script will create a tidy data set containing the means of all the columns per test subject and per activity.

This tidy dataset will be written to a tab-delimited file called tidy.txt, which can also be found in this repository.

About the Code Book

-------------------

The CodeBook.md file explains the transformations performed and the resulting data and variables.

## Example 2: Detailed / R Markdown

Codebook template

*Your name here*

*The date here*

This is an explanation for the Getting and Cleaning Data Course Project.

The data set was obtained from: <https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip>

A full description is available at the site where the data was obtained:<http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

License:

Use of this dataset in publications must be acknowledged by referencing the following publication [1]

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012

For each record it is provided:

* Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
* Triaxial Angular velocity from the gyroscope.
* A 561-feature vector with time and frequency domain variables.
* Its activity label.
* An identifier of the subject who carried out the experiment.

The dataset includes the following files:

* ‘README.txt’
* ‘features\_info.txt’: Shows information about the variables used on the feature vector.
* ‘features.txt’: List of all features.
* ‘activity\_labels.txt’: Links the class labels with their activity name.
* ‘train/X\_train.txt’: Training set.
* ‘train/y\_train.txt’: Training labels.
* ‘test/X\_test.txt’: Test set.
* ‘test/y\_test.txt’: Test labels.

The following files are available for the train and test data. Their descriptions are equivalent.

* ‘train/subject\_train.txt’: Each row identifies the subject who performed the activity for each window sample. Its range is from 1 to 30.
* ‘train/Inertial Signals/total\_acc\_x\_train.txt’: The acceleration signal from the smartphone accelerometer X axis in standard gravity units ‘g’. Every row shows a 128 element vector. The same description applies for the ‘total\_acc\_x\_train.txt’ and ‘total\_acc\_z\_train.txt’ files for the Y and Z axis.
* ‘train/Inertial Signals/body\_acc\_x\_train.txt’: The body acceleration signal obtained by subtracting the gravity from the total acceleration.
* ‘train/Inertial Signals/body\_gyro\_x\_train.txt’: The angular velocity vector measured by the gyroscope for each window sample. The units are radians/second.

Notes:

* Features are normalized and bounded within [-1,1].
* Each feature vector is a row on the text file.

Project Methodology:

The data set was originally separated into test and train subsets. The script “run\_analysis.R” was written to process the data for the course project. This script performs the following steps:

Merges the training and the test sets to create one data set.  
The original data set was randomly separated into separate train and test sets. The data from these sets are combined into one set in the script.

Extracts only the measurements on the mean and standard deviation for each measurement. Uses descriptive activity names to name the activities in the data set Appropriately labels the data set with descriptive variable names. From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.

Project Description

Short description of the project

Study design and data processing

Collection of the raw data

Description of how the data was collected.

Notes on the original (raw) data

Some additional notes (if avaialble, otherwise you can leave this section out).

Creating the tidy datafile

Guide to create the tidy data file

Description on how to create the tidy data file (1. download the data, …)/

Cleaning of the data

Short, high-level description of what the cleaning script does. link to the readme document that describes the code in greater detail

Description of the variables in the tiny\_data.txt file

General description of the file including: - Dimensions of the dataset - Summary of the data - Variables present in the dataset

(you can easily use Rcode for this, just load the dataset and provide the information directly form the tidy data file)

Variable 1 (repeat this section for all variables in the dataset)

Short description of what the variable describes.

Some information on the variable including: - Class of the variable - Unique values/levels of the variable - Unit of measurement (if no unit of measurement list this as well) - In case names follow some schema, describe how entries were constructed (for example time-body-gyroscope-z has 4 levels of descriptors. Describe these 4 levels).

(you can easily use Rcode for this, just load the dataset and provide the information directly form the tidy data file)

Notes on variable 1:

If available, some additional notes on the variable not covered elsewehere. If no notes are present leave this section out.

Sources

Sources you used if any, otherise leave out.

Annex

If you used any code in the codebook that had the echo=FALSE attribute post this here (make sure you set the results parameter to ‘hide’ as you do not want the results to show again)

## Example 3: Research Pipeline Notes

Scientific Question

Protocol

Nature

Measured Data

* Original ***data*** collected. Should be protected read-only and preserved, never modified.

Processing Code

* Code written to process a ***cached*** copy of the measured raw data.
* Configuration (***config***) files and settings saved for each processing step.
* Diagnostic code that is used to clean or munge the data set (check for errors, data validation, NA, etc.)
* Helper functions written to make the code work.

Analytic Code

* Source code that analyzes the data set to produce the computational results (summaries, pattern classifications, modeling, etc.)

Computational Results

* The final set of data that answers the question or provides the insight to answer the question.

Presentation Code

* Figures
* Tables
* Numerical Summaries

Article

* Natural language text sections
* Figures
* Tables
* Summaries
* References

# Example README

### README for swfdr folder

Copyright (C) 2011 Jeffrey T. Leek (<http://www.biostat.jhsph.edu/~jleek/contact.html>) and Leah R. Jager ([jager@usna.edu](mailto:jager@usna.edu))

Note: These functions were written on a Mac and may have difficulties when read on Windows machines.

### getPvalues.R

This file contains the code to scrape the P-values from pubmed (either run it first, or use the already calculated pvalueData.rda)

### calculateSwfdr.R

This file contains the function to estimate the science-wise false discovery rate

### journalAnalysis.R

This file contains the code to reproduce the quantities and figures in the Jager/Leek paper

### journalAnalysisHelp.R

This file contains a helper function necessary for journalAnalysis.R

### pvalueData.rda

The pre-computed p-value data used for the Jager/Leek paper in .rda format.

### simulation.R

A simulation study comparing our estimates to the truth when the assumptions hold and when they are badly violated.

### To reproduce the results in the Jager/Leek paper

To get the simulated results you should run sensitivity.R (supplementary sensitivity analysis) and simulation.R (main text sensivity analysis)

Note, because of the bootstrapping calculations, these functions may take a while (think order hours) to run

# Steps in data analysis

* Process steps
  + Define question
  + Define the ideal data set
  + Determine what data you can access
  + Obtain the data
  + Clean the data
  + Exploratory data analysis
  + Statistical Prediction/Modeling
  + Interpret Results
  + Challenge Results
  + Synthesize/write up results
  + Create reproducible code
* Interpret the Results
  + Use appropriate language
    - describes
    - correlates with / associated with
    - leads to / causes
    - predicts
* Tidy Data
  + Pipeline
    - Raw Data
    - Processing Script
    - Tidy data
    - Data analysis
    - Data communication
  + Four things you should have:
    - Raw Data [no software, no manipulation, no remove, no summary]
    - Tidy Data
    - Code book [explaining variables, values, summary choices, experimental design]
    - An explicit and exact process to go from data to analysis results
  + 'Tidy Data' paper: <http://vita.had.co.nz/papers/tidy-data.pdf>
    - Each variable forms a column
    - Each observation forms a row
    - Each type of observational unit forms a table
  + Un-Tidy errors:
    - Column headers are values, not variable names
    - Variables are stored in both rows and columns
    - A single observational unit is stored in multiple tables
    - Multiple types of observational units are stored in same table
    - Multiple variables are stored in one column
  + Tidy R functions
    - gather(), spread(), mutate(), numeric(), select(), group\_by(), separate(),
* ***Summary: Checklist for Reproducible Research***
  + Are we doing good science?
  + Was any part of this analysis done by hand?
    - If so, are those parts *precisely* document?
    - Does the documentation match reality?
  + Have we taught a computer to do as much as possible (i.e. coded)?
  + Are we using a version control system?
  + Have we documented our software environment?
  + Have we saved any output that we cannot reconstruct from original data + code?
  + How far back in the analysis pipeline can we go before our results are no longer (automatically)

reproducible?

# Project Template

<http://projecttemplate.net/architecture.html>

ProjectTemplate is based on the idea that you should structure all of your data analysis projects in the same way so that you can exploit conventions instead of writing boilerplate code. Because so much of ProjectTemplate’s functionality is based on conventions, it’s worth explaining ProjectTemplate’s idealized project in some detail.

Full Project Layout

As far as ProjectTemplate is concerned, a good statistical analysis project should look like the following:

* project/
  + cache/
  + config/
  + data/
  + diagnostics/
    - 1.R
  + doc/
  + graphs/
  + lib/
    - helpers.R
  + logs/
  + munge/
  + profiling/
    - 1.R
  + reports/
  + src/
  + tests/
    - 1.R
  + README
  + TODO

Each of these directories and files serves a specific purpose, which we describe below:

* **cache**: Here you’ll store any data sets that (a) are generated during a preprocessing step and (b) don’t need to be regenerated every single time you analyze your data. You can use the **cache()** function to store data to this directory automatically. Any data set found in both the **cache** and **data** directories will be drawn from **cache** instead of **data** based on ProjectTemplate’s priority rules.
* **config**: Here you’ll store any configurations settings for your project. Use the DCF format that the **read.dcf()** function parses.
* **data**: Here you’ll store your raw data files. If they are encoded in a supported file format, they’ll automatically be loaded when you call **load.project()**.
* **diagnostics**: Here you can store any scripts you use to diagnose your data sets for corruption or problematic data points.
* **doc**: Here you can store any documentation that you’ve written about your analysis.
* **graphs**: Here you can store any graphs that you produce.
* **lib**: Here you’ll store any files that provide useful functionality for your work, but do not constitute a statistical analysis per se. Specifically, you should use the **lib/helpers.R** script to organize any functions you use in your project that aren’t quite general enough to belong in a package.
* **logs**: Here you can store a log file of any work you’ve done on this project. If you’ll be logging your work, we recommend using the [log4r](https://github.com/johnmyleswhite/log4r) package, which ProjectTemplate will automatically load for you if you turn the **logging** configuration setting on.
* **munge**: Here you can store any preprocessing or data munging code for your project. For example, if you need to add columns at runtime, merge normalized data sets or globally censor any data points, that code should be stored in the **munge** directory. The preprocessing scripts stored in **munge** will be executed sequentially when you call**load.project()**, so you should append numbers to the filenames to indicate their sequential order.
* **profiling**: Here you can store any scripts you use to benchmark and time your code.
* **reports**: Here you can store any output reports, such as HTML or LaTeX versions of tables, that you produce. Sweave or brew documents should also go in the **reports** directory.
* **src**: Here you’ll store your final statistical analysis scripts. You should add the following piece of code to the start of each analysis script: **library('ProjectTemplate); load.project()**. You should also do your best to ensure that any code that’s shared between the analyses in **src**is moved into the **munge** directory; if you do that, you can execute all of the analyses in the**src** directory in parallel. A future release of ProjectTemplate will provide tools to automatically execute every individual analysis from **src** in parallel.
* **tests**: Here you can store any test cases for the functions you’ve written. Your test files should use **testthat** style tests so that you can call the **test.project()** function to automatically execute all of your test code.
* **README**: In this file, you should write some notes to help orient any newcomers to your project.
* **TODO**: In this file, you should write a list of future improvements and bug fixes that you plan to make to your analyses.

#### **Minimal Project Layout**

A minimal project, which you can create using **create.project(minimal = TRUE)**, only contains a subset of the full project layout:

* project/
  + cache/
  + config/
  + data/
  + munge/
  + src/
  + README

This is designed for newcomers who don’t need the more advanced subdirectories that ProjectTemplate normally creates.

# TODO and LOOKUPS

* Heritage Health Prize
* Knit to HTML
* <https://www.springboard.com/blog/free-public-data-sets-data-science-project/>
* Books.google.com/ngrams
* [www.census.gov/census2010](http://www.census.gov/census2010)
* <http://www.sdss.org/>
* DarwinTunes – audio files
* <http://www.data.gov>
* <http://www.data.gov/safety/results-second-annual-safety-datapalooza/>
* UCI Machine Learning Repository: <http://archive.ics.uci.edu/ml/datasets/Flags>
* Data sets that come with R: > library(datasets) > data(iris)
* <http://www.netlib.org/lapack>
* <http://adv-r.had.co.nz/Profiling.html#profiling>
* <http://www.r-tutor.com/content/r-tutorial-ebook>
* United States Department of Agriculture's PLANTS Database: <http://plants.usda.gov/adv_search.html>
* There is a school of thought that this approach is backwards, that we should teach ggplot2 first. See <http://varianceexplained.org/r/teach_ggplot2_to_beginners/>
* <https://data.baltimorecity.gov>
* <http://www.stat.berkeley.edu/~statcur/Workshop2/Presentations/XML.pdf>
* avatar update; gravatar update
* JSON refs:
  + <http://www.json.org>
  + Good tutorial: <http://www.r-bloggers.com/new-package-jsonlite-a-smarter-json-encoderdecoder/>
  + jsonlite vignette
* <http://vita.had.co.nz/papers/tidy-data.pdf>
* <http://research.collegeboard.org/programs/sat/data/archived/cb-seniors-2013>’
* <http://en.wikipedia.org/wiki/List_of_tz_database_time_zones>
* 2011 Journal of Statistical Software paper titled 'Dates and Times Made Easy with lubridate'.
* Original data sources:
  + <http://data.worldbank.org/data-catalog/GDP-ranking-table>
  + <http://data.worldbank.org/data-catalog/ed-stats>
* ***Grammar of Graphics (Leland Wilkinson)***
* ggplot2 – ggplot2.org (best documentation)

Open Government Sites

* United Nations: <http://data.un.org/>
* U.S. <http://www.data.gov/>
  + List of cities and states with open data
* United Kingdom: <http://data.gov.uk/>
* France: <http://www.data.gouv.fr/>
* Ghana: <http://data.gov.gh/>
* Australia: <http://data.gov.au/>
* Germany: <https://www.govdata.de/>
* Hong Kong: <http://www.gov.hk/en/theme/psi/datasets/>
* Japan: <http://www.data.go.jp/>
* Many More: <http://www.data.gov/opendatasites>

Gapminder: <http://www.gapminder.org> human health data

Survey from U.S. :<http://www.asdfree.com/> help with access and analysis in R

Data Marketplace: <http://www.infochimps.com/marketplace>

Kaggle: <http://www.kaggle.com/> Data science competitions

Collections by data scientists:

* Hilary Mason: <http://bitly.com/bundles/hmason/1>
* Peter Skomoroch: <https://delicious.com/pskomoroch/dataset>
* Jeff Hammerbacher: <http://www.quora.com/Jeff-Hammerbacher/Introduction-to-Data-Science-Data-Sets>
* Gregory Piatetsky-Shapiro: <http://www.kdnuggets.com/gps.html>
* <http://blog.mortardata.com/post/67652898761/6-dataset-lists-curated-by-data-scientists>

Human activity using smartphone data sets

<http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

Spam classifying email

<http://archive.ics.uci.edu/ml/datasets/Spambase>

Specialized Collections:

* Stanford Large Network Data
* UCI Machine Learning
* KDD Nugets Datasets
* CMU Statlib # famous canonical data sets
* Gene expression omnibus
* ArXiv Data
* Public Data Sets on Amazon Web Services

API’s with R interfaces

* twitter (and twitter package)
* figshare and rfigshare
* PLoS and rplos
* rOpenSci
* Facebook and RFacebook
* Google maps and RGoogleMaps

You can use the quantmod (<http://www.quantmod.com/>) package to get historical stock prices for publicly traded companies on the NASDAQ and NYSE. Use the following code to download data on Amazon's stock price and get the times the data was sampled.

library(quantmod)

amzn = getSymbols("AMZN",auto.assign=FALSE)

sampleTimes = index(amzn)

Resources for graphing in R

* R Graph Gallery
* R Bloggers

http://rgraphgallery.blogspot.com/2013/04/rg68-get-google-map-and-plot-data-in-it.html

Hierarchical Clustering:

Rafa’s Distances and Clustering Video

Elements of Statistical Learning

K-Means Clustering:

Rafael Irizarry’s Distances and Clustering Video

Elements of Statistical Learning

Alternative methods:

* Factor Analysis
* Independent Components Analysis
* Latent Semantic Analysis

very nice concise tutorial on creating heatmaps in R exists at:

http://sebastianraschka.com/Articles/heatmaps\_in\_r.html#clustering.

UCI's Center for

| Machine Learning and Intelligent Systems. You can find out more about the data at

| http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones. As this

| address indicates, the data involves smartphones and recognizing human activity.

* + EPA Air Pollution Data: <http://goo.gl/soQZHM>
  + Technology Transfer Network
  + Air Quality System (AQS) and AQS Data Mart

**The Air Quality System (AQS) and AQS Data Mart websites have been updated and moved.**

The new **AQS** website is at [www2.epa.gov/aqs](https://www.epa.gov/aqs)

The new **AQS Data Mart** is at <https://aqs.epa.gov/aqsweb/documents/data_mart_welcome.html>

* Internet-based Heatlh and Air Pollution Surveillance System (iHAPSS)
* <https://psychology.stanford.edu/sites/all/files/Science-2011-Peng-1226-7.pdf>
* <http://simplystatistics.org/>
* <https://www.youtube.com/watch?v=eV9dcAGaVU8>

<https://www.ted.com/talks/dan_meyer_math_curriculum_makeover>

<https://www.google.com/about/datacenters/inside/>

* [“Workflow for statistical analysis and report writing”](http://stackoverflow.com/questions/1429907/workflow-for-statistical-analysis-and-report-writing)
* [“Organizing R Source Code”](http://stackoverflow.com/questions/2284446/organizing-r-source-code)
* [“How to organize large R programs?”](http://stackoverflow.com/questions/1266279/how-to-organize-large-r-programs)
* [“R and version control for the solo data analyst”](http://stackoverflow.com/questions/2712421/r-and-version-control-for-the-solo-data-analyst)
* [“How does software development compare with statistical programming/analysis ?”](http://stackoverflow.com/questions/2295389/how-does-software-development-compare-with-statistical-programming-analysis)
* [“How do you combine “Revision Control” with “WorkFlow” for R?”](http://stackoverflow.com/questions/2286831/how-do-you-combine-revision-control-with-workflow-for-r)
* [How to efficiently manage a statistical analysis project?](http://stats.stackexchange.com/questions/2910/how-to-efficiently-manage-a-statistical-analysis-project)

<http://projecttemplate.net/>

[www.sdss.org](http://www.sdss.org) Sloan Digital Sky Survey

<http://simplystatistics.org/>

ETHICS:

Office of Research Integrity at the U.S. Department of Health and Human Services.

<https://ori.hhs.gov/>

Mathematical Biostatistics Boot Camp

* YouTube: [www.youtube.com/playlist?list=PLpl-gQkQivXhk6qSyiNj51qamjAtZISJ-](http://www.youtube.com/playlist?list=PLpl-gQkQivXhk6qSyiNj51qamjAtZISJ-)

Diagnostic Likelihood Ratio – find examples and tutorials.

* how related to ROC curves, machine effectiveness

Read: <http://nick.brown.free.fr/stapel/FakingScience-20141214.pdf>

Read: http: <https://statistics.stanford.edu/~ckirby/brad/papers/2010LSIexcerpt.pdf>

Try QQ Plots: All of these statistics are available in the stat.desc() function within the [pastecs package](https://cran.r-project.org/web/packages/pastecs/pastecs.pdf).

* + - * see week 4 discussion on Statistical Infererence.

Read: <http://www.smart-stats.org/> blog for stats at Johns Hopkins University

Paper: Regression to the mean, was invented by Francis Galton in the paper “Regression towards mediocrity in hereditary stature” The Journal of the Anthropological Institute of Great Britain and Ireland , Vol. 15, (1886). The idea served as a foundation for the discovery of linear regression.

# References

*The Elements of Data Analytic Style* [https://leanpub.com/datastyle/](https://eventing.coursera.) [other R, and data science books in leanpub]

The book [*Report Writing for Data Science in R*](https://leanpub.com/reportwriting?utm_source=coursera&utm_medium=syllabus&utm_campaign=CourseraSyllabus)

R-Studio: <http://www.rstudio.com/>

R reference card: <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>

The Comprehensive R Archive Network: <https://cran.r-project.org/>

The R Journal: <https://journal.r-project.org/>

RSeek: <http://www.rseek.org> - custom front-end to Google to find R help.

GitHub:

Tidy Data: <http://vita.had.co.nz/papers/tidy-data.pdf>

Lecture Notes for more subsetting:

<http://www.biostat.jhsph.edu/~ajaffe/lec_winterR/Lecture%202.pdf>