**DATA SPECIALIZATION**

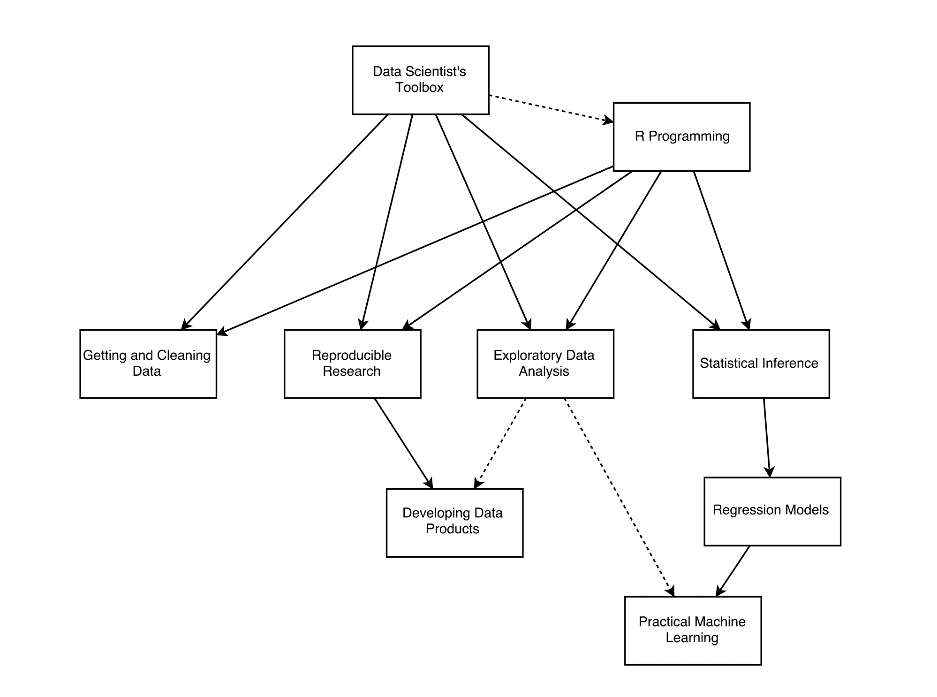
***Unit 5: Reproducible Research***

# Introduction

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

**Overview**: This course focuses on the concepts and tools behind reporting modern data analyses in a reproducible manner. Reproducible research is the idea that data analyses, and more generally, scientific claims, are published with their data and software code so that others may verify the findings and build upon them. The need for reproducibility is increasing dramatically as data analyses become more complex, involving larger datasets and more sophisticated computations. Reproducibility allows for people to focus on the actual content of a data analysis, rather than on superficial details reported in a written summary. In addition, reproducibility makes an analysis more useful to others because the data and code that actually conducted the analysis are available. This course will focus on literate statistical analysis tools which allow one to publish data analyses in a single document that allows others to easily execute the same analysis to obtain the same results.

**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.

# Reproducible Research

Started: May 04, 2016

Ends: June 5, 2016

## Week 1: Concepts, Ideas, Structures

### Introduction

This week will cover the basic ideas of reproducible research since they may be unfamiliar to some of you. We also cover structuring and organizing a data analysis to help make it more reproducible. I recommend that you watch the videos in the order that they are listed on the web page, but watching the videos out of order isn't going to ruin the story.

[Roger D. Peng](http://www.biostat.jhsph.edu/~rpeng/)

* Reproducible Research – applies much wider for data analysis
  + Communicate effort in way that it can be reconstructed
* Hard to ensure effort is reproducible with more complex analyses
* This course follows basic principles
  + Communicate exactly what you done
* Case studies for where reproducibility went right or horribly wrong.

### Course Description

In this course you will learn the ideas of reproducible research and reporting of statistical analyses. Topics covered include literate programming tools, evidence-based data analysis, and organizing data analyses. In this course you will learn to write a document using R markdown, integrate live R code into a literate statistical program, compile R markdown documents using knitr and related tools, publish reproducible documents to the web, and organize a data analysis so that it is reproducible and accessible to others.

Course Content

* Structuring and organizing a data analysis
* Markdown and R Markdown
* knitr / RPubs
* Reproducible research check list
* Evidence-based data analysis
* Case studies in air pollution epidemiology and high-throughput biology

### Course Book: Report Writing for Data Science in R

The book [*Report Writing for Data Science in R*](https://leanpub.com/reportwriting?utm_source=coursera&utm_medium=syllabus&utm_campaign=CourseraSyllabus) serves as the official course reference textbook. It comes in electronic form (PDF, ePub, and mobi) and is available from Leanpub. The book contains the lecture material for the course and can serve as a reference for you once the course is completed.

This book teaches the fundamental concepts and tools behind reporting modern data analyses in a reproducible manner. As data analyses become increasingly complex, the need for clear and reproducible report writing is greater than ever. The material for this book was developed as part of the industry-leading Johns Hopkins Data Science Specialization. Printed versions are available through Lulu (see link below).

### What is Reproducible Research About?

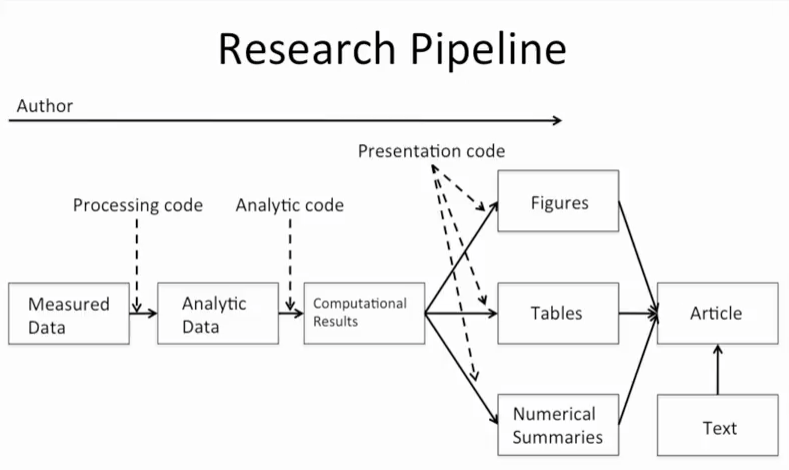
* Introduce idea of “reproducibility”.
* Reproducibility is critical for any data analysis and being able to communicate our work.
* Analogous to performing music
  + simple tunes are easy to reproduce
  + complex symphonies are difficult to reproduce (or require a large team to approach)
* Written music is how musicians communicate their compositions
* Symphony score: written music for every piece in a synchronized matrix
* Data analysis score communicates what was done and how to reproduce the analysis
* Fundamental problem: no standard or agreement on how to notate the analysis steps for data analysis
* Goal: share data with dynamic documents that communicate the data analysis to be reproduced.

### Reproducible Research: Concepts and Ideas (part 1)

* Science – ***replication*** is ultimate standard for strengthening evidence
* Conduct replication studies with independent:
  + Investigators
  + Data
  + Analytical methods
  + Laboratories
  + Instruments
* ***Replication*** important for studies that can have broad impact or regulatory decisions
* What is wrong with replication (full repeat)?
  + some studies cannot be replicated
  + not time (lost opportunity)
  + Money
  + Unique equipment or research
* ***Reproducible*** research: make analytic data and code available so others may reproduce the findings
  + more likely to validate the analysis part of the research
* Replication (gold standard) - reproducibility - nothing [spectrum of replication of research]
* Why do we need reproducible research?
  + Data collection throughput is high volume and complexity
  + Databases can be merged into new “megadatabases”
  + Models, algorithms are much more complex
  + for every field “X” there is a field “Computational X”
* Example: Reproducible Air Pollution and Health
  + estimating small (but important) health effects in presence of much stronger signals
  + Results in substantial policy decisions affecting stakeholders (cost, effort)
  + Complex statistical methods are needed (requiring extensive scrutiny)
* Internet-based Health and Air Pollution Surveillance System (iHAPSS)

### Reproducible Research: Concepts and Ideas (part 2)

* Research Pipeline
  + Article (published) – normally all that you see



* Reproducible research attempts to allow the author and reader to meet somewhere in middle (computational results)
* Recent developments in RepResearch
  + Reproducible Research in Computational Science. Science. 11 Feb 2011. Roger D. Peng.

<https://psychology.stanford.edu/sites/all/files/Science-2011-Peng-1226-7.pdf>

* + 60 Minutes – Deception at Duke.

<http://www.cbsnews.com/news/deception-at-duke-fraud-in-cancer-care/>

<https://www.youtube.com/watch?v=eV9dcAGaVU8>

* + Evolution of Translational Omics
    - Report by National Institute of Health
    - <http://www.nationalacademies.org/hmd/Reports/2012/Evolution-of-Translational-Omics.aspx>
* Lessons Learned (medical)
  + Data/metadata used to develop test should be made publicly available
  + Computer code and computational procedures sustainably available
  + Code released encompasses all data steps
* What is reproducible research
  + Analytic Data are available (different than “raw” data)
  + Analytic Code are available (applied to analytic data to get key results)
  + Documentation of Code and Data
  + Standard means of distribution
* Who are the stakeholders?
  + Authors – need tools
  + Readers – need tools
* Challenges
  + Currently high effort for authors to make public
  + Readers high effort to download and piece together the data and analysis
  + Few tools to create and execute reproducible research
* REALITY BITES
  + Authors – drop stuff on web, few central databases
  + Readers – just download and try to figure out

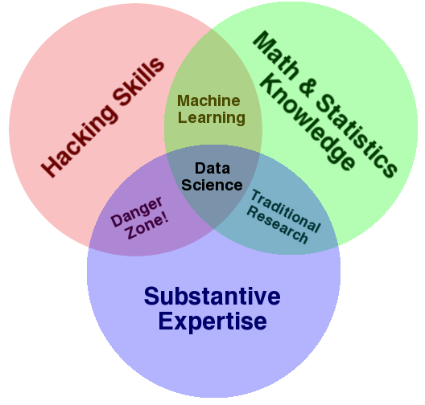
### Reproducible Research: Concepts and Ideas (part 3)

* ***Literate (Statistical) Programming***
  + ***article*** is a stream of ***text*** and ***code***
  + analysis code is divided into text and code “chunks”
  + Each code chunk loads data and computes results
  + Presentation code formats results (tables, figures, etc.)
  + Article text explains what is going on
  + Literate programs can be **weaved** to produce human readable documents and **tangled** to produce machine-readable documents
* Documentation language (humans)
* Programming language (machine)
* Sweave uses LaTeX in R
  + LaTeX is difficult to learn, used only by weirdos
  + lacks many features
* knitr is an alternate
  + brings many features of Sweave
  + can use LaTeX, Markdown, HTML
* Further infrastructure is needed to create and distribute reproducible documents

### Scripting Your Analysis

* ***Basic Principle to make work reproducible: SCRIPTING***
  + idea of writing everything down
  + Instead of notebook or on paper – use script
* Presentation of research is like hearing only the melody of music
  + there are other parts to the analysis that are not presented
    - rabbit holes
    - no-go ideas
    - problems, successes
* Score for music is how to write the entire piece of complex music
* Data Analysis Script notation is not highly standardized
  + SCRIPTING – instructions for computer

### Structure of a Data Analysis (part 1)

* Process of how data analysis unfolds (generally)
* Steps in data analysis
  + 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
* Define the Question
  + important problems don’t have all information in advance, too much information and need to find or filter the data to get what you need to solve the problem.
  + Do not start with the data!!
  + Science to Data to Applied Statistics (analysis) to Theoretical Statistics (generalized model)
    - Narrow the data to what is needed for the science question
  + Start with general question – then make it more concrete
    - 5 why’s or 5 what’s
* Define ideal data set
  + Data set depends on the goal
    - Descriptive – a whole population
    - Exploratory – random sample with many variables measured
    - Inferential – the right population and random sampling
    - Predictive – a training and test data set from same population
    - Causal – data from randomized study
    - Mechanistic – data about all components in system
* Determine what you can access
  + free data on the web
  + buy the data
  + follow the terms of use
  + may need to generate the data yourself
* Obtain the Data
  + try to obtain raw data
  + be sure to reference the source
  + polite emails go a long way
  + if you load data from the internet source (record the url, date, time)
* <http://archive.ics.uci.edu/ml/datasets/Spambase>
  + also available with kernlab package in R
* Clean the data
  + raw data needs to be processed
  + understand how the data was pre-processed (on collection)
  + understand the data source (census, sample, convenience sample, etc.)
  + May need reformatting, subsampling – record these steps
  + Determine if data are good enough

### Structure of a Data Analysis (part 2)

* subsampling into train and test data sets
  + use randomization to separate the data
* Exploratory Data Analysis
  + look at summaries of data
  + check for missing data
  + create exploratory plots
  + perform exploratory analyses (i.e. clustering)
  + Build the exploration on the TRAINING SET – leave the test set alone!
* Statistical Prediction/Modeling
  + Should be informed by the results of exploratory analysis
  + Exact methods based on question of interest
  + account for transformations
  + measures of uncertainty reported
* Get a Measure of Uncertainty
  + create predictions for the test set
  + build classification table
  + calculate the error rate
* Interpret the Results
  + Use appropriate language
    - describes
    - correlates with / associated with
    - leads to / causes
    - predicts
* Challenge the Results (Counter-Argumenting)
  + Challenge all steps (Question, Data Source, Processing, Analysis, Conclusions)
  + challenge the measures of uncertainty
  + challenge choices of terms to include in models
  + think of potential alternative analyses (good to write in the what is next section)
* Synthesize/Write-Up results
  + lead with question
  + summarize the analyses in the story
  + don’ t include every branch
    - only if it is necessary for the story
    - only if needed to address a challenge
  + Order analyses according to the story vs. chronologically completed (think of movie production)
  + Include publication quality figures to contribute to the story
* Document the analysis / Create reproducible code

### Organizing Your Analysis

* Data analysis files
  + Data
    - Raw data (store in analysis folder, document the data source)
    - processed data (well named what script processed, doc in README, make it tidy)
  + Figures
    - exploratory (simple, quick figures to get initial look at the data)
    - final publication figures (must be polished, organized and clear)
  + R code
    - Raw/unused scripts (use comments, version numbers)
    - final scripts (CLEAR comments show processing details, only final results)
    - R Markdown files (useful and easy to generate in RStudio)
  + Text
    - README files (explain what is in directory and step-by-step for analyses
    - Text of analysis / report
* Paper outline
  + Title
  + Introduction (motivation)
  + Methods (statistics you used)
  + Results (include measure of uncertainty)
  + Conclusions (include potential problems or alternate methods
* Case studies:
  + The Duke Saga Starter Set
  + Reproducible Research and Biostatistics
  + Managing a statistical analysis project guidelines and best practices
  + Project Template
* [“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)

### Use R version 3.1.1

* Current version of base package installed is now 3.2.2 [May 6, 2016]

### Quiz: Week 1

### Readings

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

* Math education fails to teach the mathematical reasoning that is needed
  + Application of math processes to the world around us
* Why math classes are such failures
  + lack of initiative
  + lack of perseverance
  + lack of retention
  + aversion to word problems
  + eagerness for formula
* Media pervasiveness
  + sitcoms affect patience for unsolved problems
  + need more imagination
* Math teaching textbooks
  + equivalent to sitcoms
  + textbook does not stress critical thinking or reasoning
    - follow the arrows….
  + Smooth straight path for the students
* Typical problem solving pattern (poor reasoning development)
  + Visual (usually cheap chart junk)
  + Structure (graphs, diagrams)
  + Steps (given to student in formulaic or recipie format)
* Why bad to society
  + Patient problem solving
* Formulation of definition of the problem is the most important
  + Einstein
* Better method for problems solving teaching
  + Visual (use the real world as it is – photos and videos, live specimens)
  + Question (decide what it is, USE AS FEW WORDS as possible)
  + Structure (decide what structure to use to help solve the problem)
  + Steps (not given, but derived from the investigation and reasoning so far)
* What to do about it
  + Use multimedia
  + Encourage student intuition
  + ask the shortest question you can
  + let students build the problems
  + be less helpful
* FOLLOW ON IDEAS:
  + is there a format, like reproducible research that can be used to markdown create learning materials
    - move away from prime contractor and central controlled media
    - on the spot and adaptive to the students needs for education, not the budget’s need to distribute wealth

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

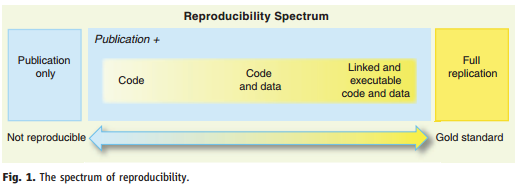
* many pictures and systems descriptions

<https://www.youtube.com/watch?v=eV9dcAGaVU8>

* Duke University cancer researchers
  + Claimed to have identified a method to correctly match cancer patients with best effective drug
* DNA is unique so each tumor is different
  + drugs don’t work for everyone same
* Discovery failure known in 2010
  + Unknown at time at how discovery was deceptive
* Dr. Potti – Rhodes Scholar
  + Claimed 80% success
* Dr. Nevins – research lab director
* Genomics DNA decoding results in huge data file
* University of Texas Graduate School of Biomedical Sciences
  + Upon reviewing the data they found oddities…
  + Kevin Kooms, Keith Bagerly
* Errors kept happening after more research
* National Cancer Institute – also found errors
* Investigation Committee
  + Found Dr. Potti to be correct
* Paul Goldberg – The Cancer Letter
  + Found the Rhodes Scholar (Australia) – fellowship
* The data was found to be manipulated
* One of the most significant retractions from medical journal
* Conclusion: Research misconduct

<https://psychology.stanford.edu/sites/all/files/Science-2011-Peng-1226-7.pdf>

* computational science improvements enabled by:
  + increased computing power
  + methodological advances
  + new communication technologies
* many scientific areas are able to collect complex high-dimensional data
  + large data sets require more computation by scientists
  + large data sets mean computational experts are doing more science
  + publicly available datasets give rise to more research by a wider range of people
* Sloan Digital Sky Survey
  + large publicly available astronomical survey of the Norther Hemisphere
  + astronomers without telescope can make discoveries
  + Similar databases exist for biology and epidemiology
* Replication – ultimate standard by which scientific claims are judged
  + independence
  + build evidence for or against a scientific hypothesis
  + culture of replication quickly weeds out spurious claims and enforces a disciplined approach
* Replication is murkier within environment of complex large data sets
  + computational resources constrained (e.g. climate science)
  + data size and access constraints
  + time and cost to replicate
* Reproducible Research
  + data and computer codes used for analysis be made publicly available
  + Not same as replication since the same data set is used
    - Difficult to recreate the entire data set (Sloan Digital Sky Survey)
  + Limited exploration of data and analysis code is possible and may be sufficient to give a quality valuation to the scientific claim
  + Fills the gap between Full Replication and No Replication



* Reproduciblity Spectrum
  + Publication Only
  + Publication Plus:
    - Code
    - Code and Data
    - Linked and Executable Code and Data
  + Full Replication
* Case studies of microarray gene expression analyses
  + Not Reproducible
  + Partially Reproducible with some discrepancies
  + Reproducible
  + Availability of Data and MetaData explains nearly all the discrepancies
* Reproducibility Standard
  + every computational experiment has a detailed log of every action taken by the computer
  + computer code is more valuable than natural language explanations in Journals
* Code Stability is a Problem
  + settings and actions are not saved concretely
  + packages are deprecated or unknowingly used and not recorded (environment)
  + Non-Open source software has proprietary (hidden) elements and methods
* Upholding a Reproducibility Standard
  + Journals make peer review efforts require reproducibility reviews
    - Make online repo of the code and data associated with the paper
    - Badges for Data, Code, Reproducibility
  + Reproducibility alone does not equate to high quality or validation
    - can be critical to track down the bugs (or the more interesting questions)
  + Integrated Infrastructure for distributing reproducible research
    - central data repository
    - GitHub: <http://github.com>
    - SourceForge: <http://sourceforge.net>
  + Culture of Reproducibility for Computation Science
    - require it for published claims
    - non-proprietary formats
  + DataMed Central and CodeMed Central
    - similar to PubMed Central

## Week 2: Markdown & knitr

Start: May 9, 2016

Due: May 15, 2016

This week we cover some of the core tools for developing reproducible documents. We cover the literate programming tool knitr and show how to integrate it with Markdown to publish reproducible web documents. We also introduce the first peer assessment which will require you to write up a reproducible data analysis using knitr.

### Coding Standards in R

* Text editor to write/save code in text formatted files
  + Standard text formats depend on languages in use
  + Tools can be used for books, webpages, etc.
  + Text editor is lowest common denominator in R code
* Indent code
  + makes structure of code much easier to determine
  + change in R Studio with Preferences – Code Format
    - 4 spaces minimum
    - 8 spaces more ideal
  + larger indents makes you think about code differently
    - more likely to design code with functions
* Limit width of code (80 columns?)
* Limit length of code in individual functions
  + keep the function to one logical activity
  + single page view keeps the function better to read/edit/understand
  + significantly helps with debugging (traceback, debugger, profiler)

### Markdown

* useful language for creating documents without complications of more extensive markup languages
  + “Markdown” conceptually more simple than markup (tags, rules, data types, etc.)
* text to HTML conversion tool for web writers
* easy to read, easy to write
* valid XHTML (or HTML)
* created by John Gruber
* Syntax: ***check the cheat sheets***

<https://help.github.com/categories/writing-on-github/>

<http://daringfireball.net/projects/markdown/basics>

### R Markdown

* Markdown implemented with R code (interactive, or code output produced with R code)
  + R code is live and embedded in the document
  + results from the R code are inserted into the markdown document
* Core tool of ***literate statistical programming***
* R markdown can be converted using standard markdown using the **knitr** package in R
* Markdown can be converted to HTML with **markdown** package in R
* Tool chain:
  + write R markdown file
  + convert to markdown in R Studio
  + convert markdown to HTML
* Slides written in R markdown converted with **slidify** package in R
* You can use raw HTML
* You can use LaTeX notation with knitr (add mathematical notations)

### R Markdown Demonstration

* see video

### knitr (part 1)

* Problems
  + Author effort to put on web
  + Quality of presentation on web is skill related
  + Reader effort to “pull” data that is unorganized or confusing
* Solution
  + single document that integrates text and code in a logical and easy to follow format
* Literate Programming
  + Don Knuth (Stanford)
    - Originally for computer programming: Document the code as you write the code
  + ***Article*** is a stream of text and code explains what is going on in the analysis process
  + ***Analysis code*** is divided into text and code ***chunks***
  + ***Presentation code*** formats the results (tables, figures, etc.)
  + Literate programs are ***weaved*** to produce human readable documents
  + Literate programs are ***tangled*** to produce machine readable documents
* Literate Programming implementation
  + documentation language
  + programming language
* **Sweave** – developed by Friedrich Leisch used LaTeX and R
* **knitr** – supports variety of documentation languages.
* How to make my work Reproducible?
  + Decide to do it (early is better)
  + Keep track of things (version control, snapshots)
  + Use software whose operation can be coded
    - generaly rules out GUI interfaces
  + Don’t save the outputs
    - Keep raw data + preprocessing code
  + Save data in non-proprietary formats
    - not as many proprietary data formats anymore
* Pros for literate programming
  + text and code in one place, logical order
  + data ad results auto updated to reflect external changes
  + code is live – keeps from introducing errors into analysis
* Cons for literate programming
  + can make documents difficult to read
  + must process each time to make human readable (processing time)

### knitr (part 2)

* knitr is an R package written by Yihui Xie as student in Iowa State
* support packages available in CRAN
* can use many different documentation
  + HTML and LaTeX are markup languages
  + Markdown
* what knitr is good for:
  + manuals
  + short/medium length technical documents
  + tutorials
  + reports (periodic builds)
    - watch production or study in progress
  + data preprocessing of documents and summaries
    - documents how data was cleaned (outliers, transformation, NA’s, etc.)
* what knitr not good for:
  + very long research documents
  + complex computations
  + documents that require precise formatting (ad-hoc formats)

### knitr (part 3)

* Demo knitr document
* Easy method in R Studio:
  + write an R Markdown document
  + knit HTML
* More Complicated:
  + write the R markdown document in text editor
  + use knitr package from R

library ( knitr )

setwd (<working directory>)

knit2html (“document.Rmd”)

browseURL(“document.html”)

**NOTE: could not get this to work in R Studio (errors)**

* knitr produces Markdown from the R code chunks (evaluates them and adds the output to the document)
* A few recommendations:
  + knitr will fill a new document with filler text: delete it
  + By default all code chunks are echoed – will see the output.
  + add names to the code chunks – helps with graphics

```{r firstchunk}

## R code here

```

### knitr (part 4)

* knitr document chain:
  + you write RMarkdown document (.Rmd)
  + knitr produces Markdown document (.md)
  + knitr converts the Markdown document to HTML (by default)
  + Do not edit (or save) the .md or .html documents until you are finished
* Options:

Here is a code chunk.

```{r simulation,echo=FALSE,results=HIDE}

set.seed(1)

x <- rnorm(100)

mean(x)

```

* Graphics:
  + the results of graphics or plots is written as embedded Base64 image data (not file link)
* Tables using xtable:
  + see example, ensure xtable package is installed
* Set global options for every code chunk in document.
  + examples: suppress echoing code and results output
  + This can get a little confusing.

```{r setoptions,echo=FALSE}

opts\_chunk$set(echo=FALSE, results=”hide”)

```

* + Output:
    - results: asis or hide
    - echo: TRUE, FALSE
* Caching Computations
  + useful for complex code chunk
  + cache=TRUE - after first run will be loaded from chunks.
    - must be re-run if something changes
    - dependencies are not explicitly checked
    - chunks with side effects may not be cacheable

### Week 2 Quiz

5/5

Saved to pdf.

### Introduction to Unit 5, Project 1

* Two steps
  + do analysis
  + make a report using knitr
* Submit the GitHub Repository URL
  + in the address bar (or use the link generator)
* Submit the SHA-1 for that commit in the GitHub repo (submission version)
  + click on the # commits link in the repo page
  + go to the commits list
  + look at the SHA-1 hash use the copy to clipboard link

### Assignment: Unit 5, Project 1

Turned in on May 17, 2016 at 0106.

Grade:

## Week 3: Reproducible Research Checklist & Evidence-based Data Analysis

This week covers what one could call a basic check list for ensuring that a data analysis is reproducible. While it's not absolutely sufficient to follow the check list, it provides a necessary minimum standard that would be applicable to almost any area of analysis.

### Communicating Results

* Communicate results to be understood and can be reproduced if needed
* Generate a lot of data
  + must pass the right level of detail of information
  + there is a general hierarchy
* Managers and leaders are busy
* First cut in email, results of analysis in oral form
* Get email response from busy/important people
  + send no more than one email per day
  + 3 sentences or less (as much in subject line)
  + ask yes/no questions
  + indicate action taken if no response by time specified
  + be specific and concise
  + Keywords (help search)
* Hierarchy of Information built into a research paper
  + Title/Author list
  + Abstract
  + Body/Results
  + Supplemental Materials (details)
  + Code / Data (the whole pile)
* Hierarchy of Information: Email
  + subject line /sender info
    - summary in one sentence if you can
  + Email body
    - brief description of problem / context
    - recall proposed and executed (1-2 paragraphs)
    - action options and make concrete
    - make questions yes/no if possible
  + attachments
    - code file
    - report
    - concise and keep relevant to the email
  + links
    - supplementary material
    - GitHub repository / project web site

### RPubs

* <http://rpubs.com>
  + account
  + allows you to publish markdown and knitr documents
* knitr in RStudio and then hit “publish” button.
* all files are public so be careful.

### Reproducible Research Checklist (parts 1-3)

* ***DO: start with good science***
  + garbage in, garbage out
  + coherent, focused question simplifies many research problems
  + work with good collaborators reinforces good practice
  + something interesting will motivate good habits
* ***DON’T: do things by hand***
  + edit spreadsheet to clean the data by hand (not reproducible)
    - removing outliers (NO)
    - QA/ QC on data – changing values
    - validate measurements
  + Edit tables or figures (rounding, formatting by hand is not reproducible)
  + Download data from websites (not reproducible if done by hand)
    - may be difficult to obtain the data again
  + moving data around computer; splitting and reformatting data files
    - better to automate so can be reproducible
  + “Do this only once….”
    - still document and write into repeatable code
  + Anything done by hand must be precisely documented and may be so detailed and hard to explain it is not worth the effort.
* ***DON’T: Use Point and Click software (GUI interfaces)***
  + GUI actions are difficult to record and reproduce
  + Log files may be useful, but could be complicated
  + Be wary of data analysis software that is highly interactive = more non-reproducible
  + text editors are somewhat ok – be sure to record the default formats, indent, etc.
* ***DO: Teach a Computer; Automate***
  + Instructions for computer are more well organized/structured, possibly better thought out
  + Teaching a computer almost guarantees reproducibility
  + Payback will be later than up front
  + Example: download file with ***download.file()***
    - Full URL is specified
    - Name of file specified
    - Directory is specified
    - Code can be executed by someone else
* ***DO: Use some version control***
  + Slow things down; think before commit
    - add changes in small chunks
  + Tracks and tag the snapshots; able to revert
  + Also helps to publish results
* ***DO: Keep Track of Your Software Environment***
  + Computer Architecture: CPU (Intel, AMD, ARM, GPU)
  + Operating System: Windows, Mac OS, Linux / Unix
  + Software toolchain:
    - Compiler
    - Interpreter
    - Command Shell
    - Programming Language
    - Database Backend
    - Data Analysis Software
  + Supporting Software
    - Libraries
    - Packages
    - Dependencies
  + External Dependencies:
    - Web sites
    - data repositories
    - remote databases
    - software repositories
  + Version Numbers: everything
  + in R: ***sessionInfo()***

> sessionInfo()

R version 3.2.3 (2015-12-10)

Platform: x86\_64-w64-mingw32/x64 (64-bit)

Running under: Windows >= 8 x64 (build 9200)

locale:

[1] LC\_COLLATE=English\_United States.1252 LC\_CTYPE=English\_United States.1252

[3] LC\_MONETARY=English\_United States.1252 LC\_NUMERIC=C

[5] LC\_TIME=English\_United States.1252

attached base packages:

[1] stats graphics grDevices utils datasets methods base

other attached packages:

[1] lattice\_0.20-33 dplyr\_0.4.3

loaded via a namespace (and not attached):

[1] Rcpp\_0.12.3 assertthat\_0.1 digest\_0.6.9 crayon\_1.3.1 bitops\_1.0-6 grid\_3.2.3

[7] R6\_2.1.2 DBI\_0.3.1 magrittr\_1.5 swirl\_2.4.1 httr\_1.1.0 stringi\_1.0-1

[13] testthat\_0.11.0 rmarkdown\_0.9.5 tools\_3.2.3 stringr\_1.0.0 RCurl\_1.95-4.7 parallel\_3.2.3

[19] yaml\_2.1.13 memoise\_1.0.0 htmltools\_0.3.5 knitr\_1.12.3

* ***DON’T: Save Output***
  + avoid saving data analysis output (table, figures, summaries)
    - maybe temp for efficiency
    - delete all else – keeps process robust and reproducible
  + avoid not understanding where a piece of the analysis chain came from
  + Best to save the code and the data
* ***DO: Set your Random Number Generator Seed***
  + set.seed()
  + allows the random numbers to be exactly reproducible
  + Always set the seed
* ***DO: Think about the entire Pipeline***
  + Data analysis is a lengthy process
  + Rad data -> processed data -> analysis -> report
  + the more the pipeline is reproducible, the better
* ***Summary: Checklist***
  + 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)

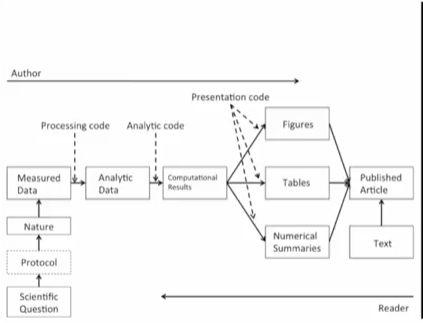
reproducible?

### Evidence-based Data Analysis (part 1)

* ***Replication***
  + Focus on validity of the scientific claim
  + Is the claim true?
  + Ultimate standard for strengthening scientific evidence
  + Uses new investigators, ***new data***, different analysis
  + Particularly important for policy and regulation decisions
* ***Reproducibility***
  + Focus on validity of the data anlysis
  + Can we trust this analysis?
  + Arguably minimum standard for any scientific study
  + New investigators, ***same data***, same methods
  + Important when replication is impossible
* ***Background Trends***
  + studies cannot be replicated: time, money, unique opportunities
  + technology increasing data collection
  + bigger databases (merging)
  + computing power allows more complex analysis
  + For every field X there is a field Computational X
* ***Trend Results***
  + basic analysis is hard to describe
  + heavy computation thrust on people not familiar with methods
    - computing
    - statistics
  + errors more easily introduced in long pipelines
  + knowledge transfer goes down with complexity
  + ***End result is complicated analyses are not trusted.***

### Evidence-based Data Analysis (part 2)

* Research Pipeline



* What does Reproducible Research Solve?
  + Transparency
  + Data Availability
  + Software / Methods available
  + Improved transfer of knowledge
  + DO NOT GET VALIDITY OR CORRECTNESS
    - can be reproducible and still wrong
* Can requiring reproducibility deter bad analysis or methods?
* Problems with reproducibility?
  + addresses the downstream aspect of research process – post-publication
  + assumes everyone plays by the same rules
  + assumes everyone wants to achieve same goals (i.e., scientific discovery)

### Evidence-based Data Analysis (part 3)

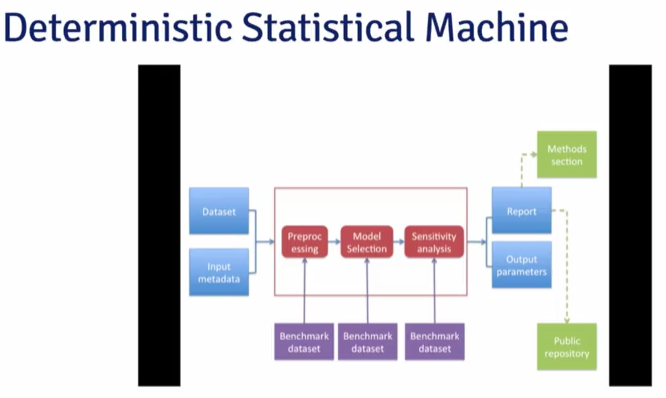
* Scientific Dissemination Process
  + Research Conducted
  + Paper submitted to journal
    - Editor’s Judgement
  + Paper Publication
    - Peer review
    - Maybe reproducible research could be done here (maybe unrealistic)
  + Post-Publication Review
    - This is where reproducible research can intervene/improve

### Evidence-based Data Analysis (part 4)

* Who reproduces research?
  + re-run analysis – check for matching results
  + check code for bugs/errors
  + try alternate approaches; check sensitivity
  + inherited from the traditional notion of replication
* Who and what are their goals?
  + Original Investigator
  + Reproducers – I don’t care (general public)
  + Reproducers – the truth agrees (scientists)
  + Reproducers – The truth is something else (scientists)
  + Reproducers – the truth is not whatever you come up with (????)
* Secondary Analyses are inevitably colored by interests/motivations of others
* Reproducibility does bring a degree of transparency (code and data)
* Can we trust analyses? - not addressed by reproducibility
* ***Evidence Based Data Analysis***
  + should be evidence to justify the application of a given method
  + for example what bandwidth to use in a histogram

### Evidence-based Data Analysis (part 5)

* Goal is to create standardized, accepted analytic pipelines from evidence-based components
* Analysis with a transparent box
* Analogous to a pre-specified clinical trial protocol (includes how data will be analyzed)
* Deterministic Statistical Machine: <http://goo.gl/Qvlhuv>



* Case Study Example:

1. check for outliers, high leverage, overdispersion
2. fill in missing data – NO
3. model selection
   1. estimate degrees of freedom to adjust for unmeasured confounders
   2. other aspects of model not as critical
4. Multiple lag analysis
5. Sensitivity Analysis WRT
   1. Unmeasured confounder adjustment
   2. Influential points

* Idea: create archives of curated libraries of data analysis
  + packages that encode data analysis pipelines for given problems, technologies, questions
  + experts in field
  + changes introduced after passing benchmarks and unit tests

***BLUF: Reproducible Research improves transparency, but does not answer the question CAN YOU TRUST THE ANALYSIS? -- missing part is the standardized acceptance of methods (curated archive idea)***

### Readings

<http://simplystatistics.org/2012/08/27/a-deterministic-statistical-machine/>

* problem most “user-friendly” statistical software suffers from: It makes it really easy to screw up a data analysis.
  + tell you something is significant and if you don’t like that it isn’t, you can keep slicing and dicing the data until it is.
  + GUI improvements won’t prevent false positives that plague science and cost business big $$.
* The key issue behind getting insight from data is knowing when you are fooling yourself with ***confounders***, or ***small effect sizes***, or ***overfitting***.
* **“deterministic statistical machine”:** you input a data set and then specify the question you are asking (is variable Y related to variable X? can i predict Z from W?) then, depending on your question, it uses a deterministic set of methods to analyze the data.
  + Say regression for inference, linear discriminant analysis for prediction, etc.
  + method is fixed and deterministic for each question.
  + performs a pre-specified set of checks for outliers, confounders, missing data, maybe even data fudging.
  + generates a report with a markdown tool and then immediately publishes the result to figshare.

<https://zoom.us/recording/play/XlETWuVKT2WzFduea9XeJ8-TWshcE-XMTi-9PDi9OKuPh5N5IbB23MKPyeTsRBQC>

**Data Science, Uses, Excuses, and Abuses**

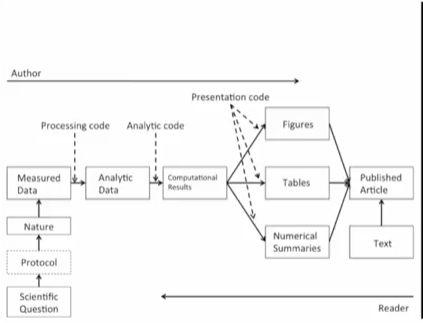
* Marketing Research blog
* The Modeling Agency – teach clients how to use the tools.
* Data Science
  + broader than analytics
  + more technical form of business analytics
    - could be an inaccurate stretch from sound business strategy
  + hype is really big right now
  + Topics:
    - data acquisition
    - data delivery
    - data analysis
    - data consumption
  + Team effort really
  + Software and Statistics
* Similar problems with general R&D product design
  + the team works in a vacuum and does not consider the customer
  + capable
* Too much market research is focused on consumers and not actual people
  + too narrow view on a data set
* Value is to help understand the right questions
* Subject Matter Experts are needed to get the right questions but they are biased
* Cross Industry Standard for Data Mining
  + <https://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining>
* Data silos
  + pull data into an analytic sandbox
  + know what resources are available and how they can integrate
* Hype: invested a lot of money in analytics departments
  + Rest of company finds very hard to interact with it
  + Hard to explain what it is you do and what the value is

## Week 4: Case Studies & Commentaries

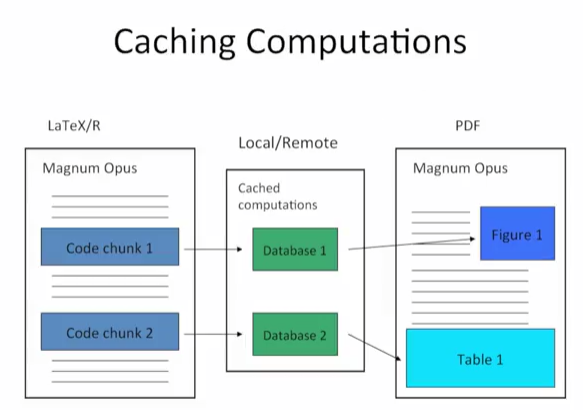
This week there are two case studies involving the importance of reproducibility in science for you to watch.

### Caching Computations (lecture notes)

* Literate Statistical Programming
  + article is a stream of text and code
  + analysis code is divided into code and text chunks
  + each code chunk loads data and computes results
  + more from previous lecture….
* Research Pipeline
  + review of prior lecture



* Caching Computations



* **cacher** (R package)
  + evaluate code written in files and stores results in key-value database
  + SHA-1 hash values so changes tracked and code re-evaluated
  + cacher packages able to be distributed
  + also can look at snippets
* Use cacher
  + parse R source file
  + create the cache directories and subdirectories
  + cloning creates directories and leaves data as optional (use the cache)
  + Analysis Graphs – view the chunk steps conceptually
* cacher functions
  + objectcode – check specific object (from the graph)
  + runcode - Running code does not compute where there are cache stored
  + checkcode – evaluate from scratch
  + loadcache() – load the cached object values (notes the transfer)
* Authors can package their analysis
  + Readers can have flexibilitiy to not have to re-create the entire environment for review

### Case Study: Air Pollution

* Case: identify the harmful elements of particulate matter air pollution
* Particulate matter air pollution
  + dust is not a monolithic entity
  + Constituents range from benign to harmful (chemicals, metals, VOC, etc.)
  + Identifying the most harmful constituents can allow targeted mitigation measures.
* What causes toxic PM
  + certain sources create chemical components
* Current regulation
  + Only affect the TOTAL amount in air
  + Does not account for the sources of PM
  + Could be improved by focusing on the most harmful constitutents
* National Morbidity Mortality and Air Pollution Study (NMMAPS)
  + national study on short-term health effects of ambient air pollution
  + focused on PM10 and Ozone (O3)
  + Key publications:
    - <http://www.ncbi.nlm.nih.gov/pubmed/11098531>
    - <http://www.ncbi.nlm.nih.gov/pubmed/113554823>
  + Funded by Health Effects Institute
  + Health outcomes include morbidity and hospitalizations for cardiovascular and respiratory disease
  + One of the most reproducible studies at NLM
    - <http://www.ihapss.jhsph.edu>
    - <http://www.ncbi.nlm.nih.gov/pubmed/22475833>
  + Methodology test bed, often cited
* Cardiovascular Effects of Nickel in Ambient Air
  + <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1665439>
  + Claim: larger concentrations of nickel show higher community effects from PM10
    - nickel appeared to be a more significant indicator
    - Too simple to be true? – very attractive conclusion!
  + Rexamined in <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2137127>
    - Hypothesis: results driven by NYC results with high nickel content PM
    - Plotted % increase in mortality vs. long term nickel concentration
    - outliers give a significant regression model with statistical significance (p<0.01)
    - take out outliers (all NYC counties) and regression is not significant (p<0.31)
    - regression is sensitive to the NYC counties included in data
      * most information (strength of relationship) come from 3 counties in NYC
      * NYC does have higher concentrations of nickel and vanadium
      * there is evidence of nickel effect on toxicity of PM10
    - Original evidence may be not as strong as suggested – MORE WORK NEEDED.
* Reproducibility allowed this case to move toward more scientific knowledge

### Case Study: High Throughput Biology (lecture)

* Kieth Baggerly lecture – University of Texas M.D. Anderson Cancer Center – 9/4/2010
* Duke university regarding high dimensional genomics
* Importance of reproducible research in high velocity biology
  + more data is more complex
  + ability to reproduce is decreased due to the computational resources required
  + Intuition about what makes sense breaks down in high dimensions
    - genomic structures as biomarkers needs high verification
    - humans will find patterns where there are none
  + Documentation critical to avoid errors – poor documentation leads to backtracking
* Paper published by Potti et al (2006) in Nature Medicine 12:1294-1300
  + ***claim***: we can use microarray data from cell lines (NCI60) to define drug response signatures which can then be used to predict patient response
  + 7 examples
  + MDA very excited
  + look at most resistant and most sensitive gene lines for a particular treatment agent
  + compare the two to determine the principal component genes (PCA) – they call meta-genes
  + these meta-genes fit a model which can then be a predictor of patient response
  + All the data was public
* Baggerly and Coombes - reproduced the research using the same the data (it is public data)
  + did not get same results
  + the clusters in the test set did not separate like the training data did
    - WHAT????
  + extreme separation in heatmaps using 2 sample t-test
    - NOT SAME AS PUBLISHED???
  + side by side comparison of the data shows an off by one indexing error
  + Software use – no documentation
    - recreated documentation
    - two input files – one with header one without
  + 4 of 6 matching gene lists included unexplained outliers
  + Data was transposed – swapped interpretation of ‘0’ and ‘1’ values
    - child leukemia patient data showed resistance …. for something that works
* 92 genes from test set
  + overlap of 14 genes (one contiguous block)
  + 5 other genes are used to explain the biology
  + no overlap between genes listed for: genes work test, genes work train, genes explain biology
* Reused samples
  + column of sample data when correlated with itself and no other sample (unique samples)
    - main diagonal should be all that is shown
  + only 85 of 124 samples were unique
  + authors noted 95 samples
    - 15 repeats – some of them labeled both sensitive and resistant
* They started clinical trials – without checking the reported errors.
  + Duke suspended and then restarted the trials
  + possible people assigned to KNOWN INNEFFECTIVE treatments
* Validation data set shows that every sample was incorrect while clinical trials underway
  + NCI received report that was not public
  + NCI is federally funded and required to follow FOIA
  + NCI sends redacted report
* One of the scientists claimed to have been a Rhodes Scholar (false)
  + among other C.V. fudging
* Joint letter from biostatisticians to DoD, NCI, Duke, et. al. to stop the trials
  + <http://groups.google.com/group/reproducible-research>
* CONCLUSION: the raw data was different than the data used in the analysis
  + the research was reproducible – i.e. the methods worked for the data and code provided
  + the research was not replicateable – using different raw data produced different results
  + Also, the analysis data could not be traced back to the raw data sets
    - In fact they were using analyzed data not raw samples and still could not trace back
* ROOT CAUSES:
  + Shoddy or manipulated data handling
    - cautionary tale for all of us about the value of securing the chain of data custody
  + Shoddy or manipulated reviews
    - assuring that we release data and code when we publish.
  + Shoddy or manipulated internal reviews once indicators of incorrectness provided
    - a reasonable balance of skepticism about our colleagues and the need to support them
  + Insufficient oversight by funding organizations (NIH)
    - The funding agencies should be in that position, but they are not. They have multiple constraints and pressures, including their funded scientists whose interests are skewed toward not sharing data, pressures from their institutions who want to protect data and IP, and things like the Byah-Dole act which encourage institutions to patent their results.
* Data handling errors
  + Confounding in the experimental design
  + mixing up the sample labels
  + mixing up the gene labels
  + mixing up the group labels
  + most mixes involve simple switches or offsets
  + Simplicity of mixes often hidden
    - incomplete documentation
    - common mistakes (honest or not)
  + Intuition is poor so mistakes are more common than you want to admit
* Normalization of keeping data safe
  + Data
  + Code
  + Literate Programming (Sweave or knitr reports
  + Reuse templates
  + Report structure standardized
  + Executive summaries
  + Appendices
    - SessionInfo()
    - Saves (caches)
    - File locations

### Commentaries on Data Analysis

* Preventative data errors and poor analyses
* Framing the question in data analysis
  + Wrong question can give problems with reproducible
  + For example highly-exploratory or high-descriptive analysis
    - may not lead to firm results
  + Confirmation and results
  + ***The type of analysis matters….***

### Course Project 2

* NOAA storm event database
* <http://www.rpubs.com/rdpeng/13396>

Data file:

<https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2>

Documentation:

<https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf>

### Assignment: Course Project 2

Due May 29, 11:59 PM PDT

### Review Classmates: Course Project 21h 00m

Due June 1, 11:59 PM PDT

### Post-Course Survey

# Data Analysis Style Guide (R)

* Download the data
  + check if it exists
  + give location
  + download the file
  + read the file (csv, html)
* 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***
* Look at the size of an R object
  + object.size()
  + format(object.size(x), units=”Mb”)

# 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/>

# 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>