Lec 03 - Intro to Git and GitHub

Statistical Programming

Sta 323 | Spring 2022

Dr. Colin Rundel

Responsible research and reproducibility

Seizure study retracted after authors realize data got "terribly mixed"

- From the authors of **Low Dose Lidocaine for Refractory Seizures in Preterm Neonates**:
- "The article has been retracted at the request of the authors. After carefully re-examining the data presented in the article, they identified that data of two different hospitals got terribly mixed. The published results cannot be reproduced in accordance with scientific and clinical correctness."

From http://retractionwatch.com/2013/02/01/seizure-study-retracted-after-authors-realize-data-got-terribly-mixed/

Bad spreadsheet merge kills depression paper, quick fix resurrects it

- The authors informed the journal that the merge of lab results and other survey data used in the paper resulted in an error regarding the identification codes. Results of the analyses were based on the data set in which this error occurred. Further analyses established the results reported in this manuscript and interpretation of the data are not correct.
- **Original conclusion:** Lower levels of CSF IL-6 were associated with current depression and with future depression [...].
- **Revised conclusion:** Higher levels of CSF IL-6 and IL-8 were associated with current depression [...].

From http://retractionwatch.com/2014/07/01/bad-spreadsheet-merge-kills-depression-paper-quick-fix-resurrects-it/

Study of social media retracted when authors can’t provide data

- "A business journal has retracted a 2016 paper about how social media can encourage young consumers to become devoted to particular brands, after discovering flaws in the data and findings."
- Reasons for retraction:
  - Error in data
  - Error in results and/or conclusions
  - Results not reproducible

From http://retractionwatch.com/2017/07/31/study-social-media-retracted-authors-cant-provide-data/

Heart pulls sodium meta-analysis over duplicated, and now missing, data

- "The journal Heart has retracted a 2012 meta-analysis after learning that two of the six studies included in the review contained duplicated data. Those studies, it so happens, were conducted by one of the co-authors."
- From the retraction notice, "The Committee considered that without sight of the raw data on which the two papers containing the duplicate data were based, their reliability could not be substantiated. Following inquiries, it turns out that the raw data are no longer available having been lost as a result of computer failure."
- Reasons for retraction:
  - Duplication of data
  - Results not reproducible

From http://retractionwatch.com/2013/05/02/heart-pulls-sodium-meta-analysis-over-duplicated-and-now-missing-data/

Teaching Reproducibility

1. Convince researchers to adopt a reproducible research workflow.
2. Train new researchers who don’t have any other workflow.

Donald Knuth "Literate Programming" (1983)

"Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do."

"The practitioner of literate programming [...] strives for a program that is comprehensible because its concepts have been introduced in an order that is best for human understanding, using a mixture of formal and informal methods that reinforce each other."

- These ideas have been around for years!
- Tools for putting them to practice have also been around.
- They have never been as accessible as the current tools.

Reproducibility checklist

- Are the tables and figures reproducible from the code and data?
- Does the code actually do what you think it does?
- In addition to what was done, is it clear why it was done? (e.g., how were parameter settings chosen?)
- Can the code be used for other data, especially future updates to the current data?
- Can you extend the code to do other things?

Ambitious goal

We need an environment where

- data, analysis, and results are tightly connected, or better yet, inseparable,
- reproducibility is built in,
  - the original data remains untouched
  - all data manipulations and analyses are inherently documented
- documentation is human readable and syntax is minimal.

Reproducible data analysis

- Scriptability → R or Python
- Literate programming → R Markdown or Jupyter Notebooks
- Version control → Git / GitHub

Could these tools have prevented some of the aforementioned retractions?

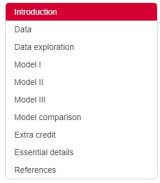
What is markdown?

- Markdown is a lightweight markup language for creating HTML (and other formatted) documents.
- Markup languages are designed to produce documents from human readable text (and annotations).
- Some of you may be familiar with LaTeX. This is another (less human friendly) markup language for creating pdf documents.
- Why markdown is great:
  - Easy to learn and use.
  - Focus on **content**, rather than **coding** and debugging **errors**.
  - Once you have the basics down, you can get fancy via HTML, JavaScript, and CSS.
  - Used by both R Markdown and Jupyter Notebooks

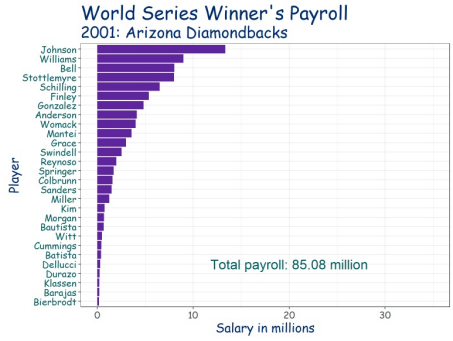
R Markdown

Something simple

Something fancy



Exam 2



Introduction

You will work with yearly baseball statistics for each team from the years 2000 to 2015. You do not need any deep insight into the sport of baseball to successfully complete this exam. However, if you think you do not understand a specific variable and how it relates to baseball then ask questions.

A baseball team beats its opponent when it outscores its opponent. This happens when a team scores more runs than it allows in a game. The win percentage (percentage of games played that resulted in a win) of a team is related to the number of runs a team scores versus the number of runs a team allows over the course of a full season (usually 162 games). Runs scored and runs allowed are a function of hits and walks, and hits allowed and walks allowed, respectively. You will explore three models that investigate the relationship between win percentage, and runs scored and runs allowed. This is the first step if you ultimately want to quantify players' value to a team.

Data

Below is a preview of the data set. Consult the data dictionary for further details on the variables. Definitions of variables are available if you load package `Lahman` and type `?Teams` in your Console. The file `mlb.csv` is a subset of data frame `Teams`. You should not work with the `Teams` data frame.

To get started, load packages `tidyverse` and `broom`. Next, read in `mlb.csv` with function `read_csv()` and save it as an object named `mlb`.

R Markdown resources

- In RStudio, go to `Help > Cheatsheets` and select
  - R Markdown Cheat Sheet
  - R Markdown Reference Guide
- Check out the official R Markdown book: R Markdown: The Definitive Guide by Yihui Xie, J. J. Allaire, and Garrett Grolemund
- Check out bookdown: Authoring Books and Technical Documents with R Markdown by Yihui Xie.
- Take a look at RPubs web published R Markdown documents.

A note on environments

- Your R Markdown document and your Console do not share their global environments.
- This is good for reproducibility, but can sometimes result in frustrating errors.
- This also means any packages or data needed for your analysis need to be loaded in your R Markdown document as well.

R Markdown suggestions

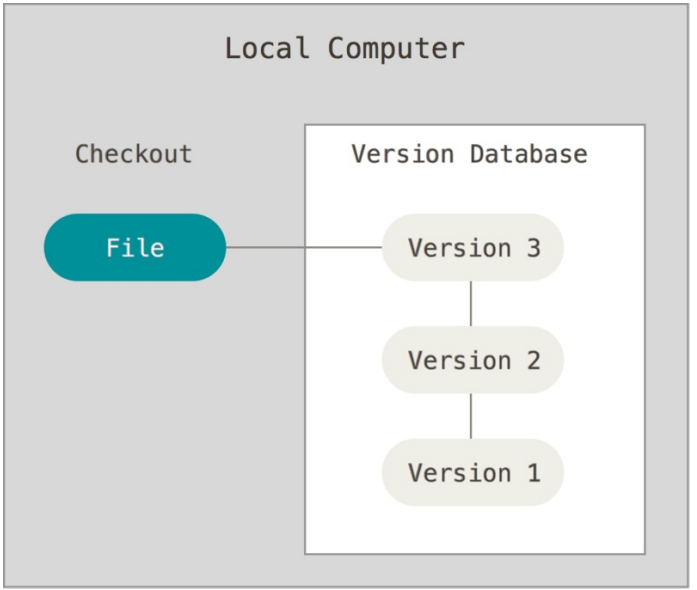
- Remember to name your code chunks
- Familiarize yourself with chunk options (https://yihui.org/knitr/options/)
    - Use global chunk options to reduce duplication
- Load packages at the start of a document, generally the chunk after your setup chunk
- Familiarize yourself with various outputs: Make slides with `output: ioslides_presentation` or `xaringan`, make websites with `blogdown`, author a book with `bookdown`, etc.

These slides were made with R Markdown and `xaringan`.

Git and GitHub

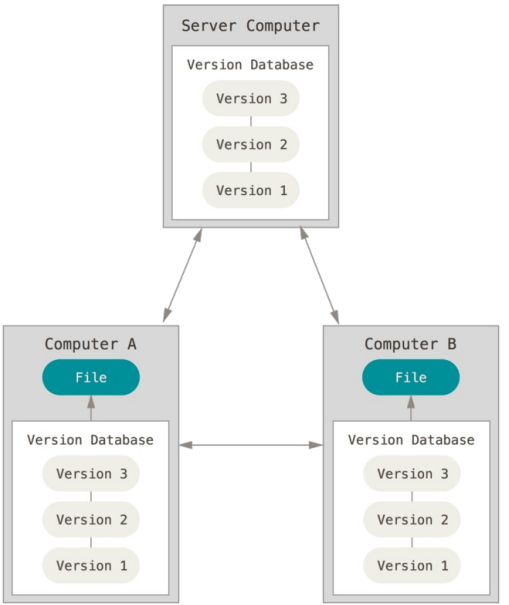
Why version control?

- Simple formal system for tracking chang
- Time machine for your projects
  - Track blame and/or praise
  - Remove the fear of breaking things
- Learning curve can be a bit steep, but wh



Why Git?

- Distributed
  - Work online or offline
  - Collaborate with large groups
- Popular and Successful
  - Active development
  - Shiny new tools and ecosystems
  - Fast
- Tracks any type of file
- Branching
  - Smarter merges



Verifying Git is installed

Git is already set-up on the DSS servers. In the terminal tab you can verify this by

```
[cr173@numeric1 ~]$ git --version
git version 2.20.1
```

```
[cr173@numeric1 ~]$ which git
/usr/bin/git
```

To install Git on your own computer follow the directions in Happy Git and GitHub for the useR.

Git sitrep

```
usethis::git_sitrep()
## Git config (global)
## ● Name: <unset>
## ● Email: <unset>
## ● Vaccinated: FALSE
## ℹ See `?git_vaccinate` to learn more
## ℹ Defaulting to https Git protocol
## ● Default Git protocol: 'https'
## GitHub
## ● Default GitHub host: 'https://github.com'
## ● Personal access token for 'https://github.com': <unset>
## x Call `gh_token_help()` for help configuring a token
## Git repo for current project
## ℹ No active usethis project
```

Configure Git

The following will tell Git who you are, and other common configuration tasks.

```
usethis::use_git_config(
  user.name = "Colin Rundel",
  user.email = "rundel@gmail.com",
  push.default = "simple",
  pull.rebase = FALSE
)
```

You will need to do this configuration once on each machine in which you choose to use Git.

This can also be done via the terminal with,

```
$ git config --global user.name "Colin Rundel"
$ git config --global user.email "rundel@gmail.com"
$ git config --global push.default simple
```

Configure Git verification

To verify you configured Git correctly, run

```
usethis::git_sitrep()
## Git config (global)
## ● Name: 'Colin Rundel'
## ● Email: 'rundel@gmail.com'
## ● Vaccinated: FALSE
## ℹ See `?git_vaccinate` to learn more
## ● Default Git protocol: 'https'
## GitHub
## ● Default GitHub host: 'https://github.com'
## ● Personal access token for 'https://github.com': <unset>
## x Call `gh_token_help()` for help configuring a token
## Git repo for current project
## ℹ No active usethis project
```

You should see output similar to above.

Configure SSH and GitHub (authentication)

We will be authenticating with GitHub using SSH based public / private keys. We can create a new key pair (if necessary) by running the following in RStudio's console:

```
credentials::ssh_setup_github()
```

```
## No SSH key found. Generate one now?
##
## 1: Yes
## 2: No
##
## Selection: 1
## Generating new RSA keyspair at: /home/guest/.ssh/id_rsa
## Your public key:
##
##  ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC/wH7pT3UXdOMJSX2wMaPVTyGnYkS8OPmcfjct6h8Q+44/9UG3sOibjjUCxIxVe
##
## Please copy the line above to GitHub: https://github.com/settings/ssh/new
## Would you like to open a browser now?
##
## 1: Yes
## 2: No
##
## Selection: 1
```

Getting started today

In order to get started, you need to obtain today's files from GitHub. The steps below will give you access.

1. Log in to GitHub
2. Navigate to https://github.com/sta323-sp22/
3. Find the repository named `git_demo`-username and open it
4. Copy the link under `Clone or Download` to clone with SSH.
5. In RStudio go to `File > New Project > Version Control > Git`
6. Paste the URL, that you copied in step 4, in the box under `Repository URL:`

You now should have all the files in the repository in a directory on the server or your own computer.

Version control best practices

- Commit early, often, and with complete code.
- Write clear and concise commit summary messages.
- Test code before you commit.
- Use branches.
- Communicate with your team.

Git and GitHub resources

- Git's Pro Git book, Chapters Getting Started and Git Basics will be most useful if you are new to Git and GitHub
- Git cheatsheet by Atlassian
- GitHub's interactive tutorial
- Free online course from Udacity
- Happy Git with R by Jenny Bryan