

# **Streamline Your Workflow: Integrating SAS, LaTeX, and R into a Single Reproducible Document**

---

Lucy D'Agostino McGowan

 @LucyStats

April 4, 2017

GO HEELS



Figure 1: #GDTBATH

# Background

---

# Goals

 easy to implement

# Goals

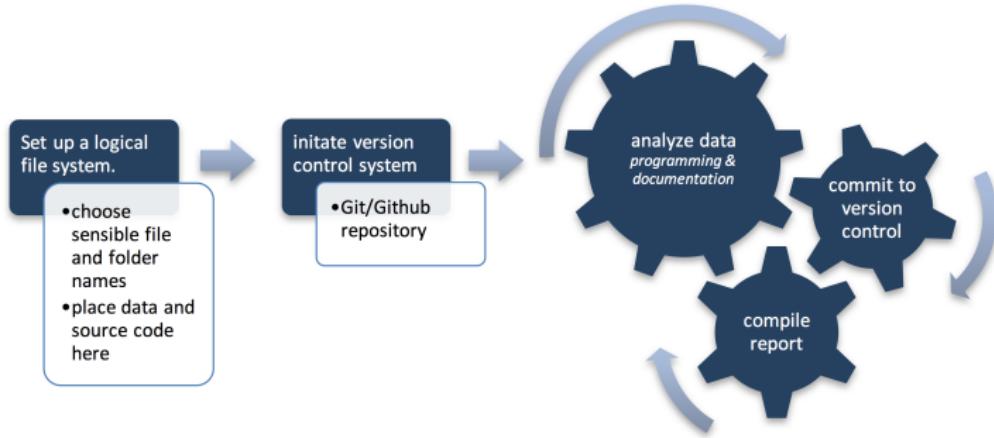
 easy to implement

 minimizes errors

## Goals

-  easy to implement
-  minimizes errors
-  collaborators can understand & contribute

# Background



**Figure 2:** Streamlined workflow summary

## Definitions

**shell:** This can be referred to as the terminal, bash, Command Prompt, etc. We will be navigating to different folders and running commands here.

## Definitions

chunk: A section of a programming language, in this case SAS or R.

## Definitions

Git: A version control system for tracking file changes

GitHub: Web based Git repository

## Definitions

**LATEX**: A type-setting system designed for scientific documentation.  
*Pronounced lah-tek*

## Best Practices

---

# File Organization

Consistent file organization is key. I prefer the following setup:

project-name

↳ code

↳ data

↳ reports

# File Organization

↓ All lower case

# File Organization

- ↓ All lower case
- █ Separated by underscores (\_) for field separation and dashes (-) for term/word separation (*for example, 2017-01-01\_new-years-resolutions*).

# File Organization

- ↓ All lower case
- █ Separated by underscores (\_) for field separation and dashes (-) for term/word separation (*for example, 2017-01-01\_new-years-resolutions*).
- 📅 If including a date, begin file name with YYYY-MM-DD  
(*this ensures easy sorting*)

# File Organization

- ↓ All lower case
- █ Separated by underscores (\_) for field separation and dashes (-) for term/word separation (*for example, 2017-01-01\_new-years-resolutions*).
- If including a date, begin file name with YYYY-MM-DD

(*this ensures easy sorting*)

⚠ Avoid things like

analysis\_final\_final\_this-is-really-final\_2.tex

# Version Control

The screenshot shows a GitHub repository page for 'LucyMcGowan / report-example'. The top navigation bar includes links for 'This repository', 'Search', 'Pull requests', 'Issues', 'Gist', and user profile icons. Below the header, the repository name 'LucyMcGowan / report-example' is displayed, along with a star count of 0, a fork count of 0, and a 'Unwatch' button. A navigation bar below the header offers links to 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area features a title 'Integrating SAS, LATEX, and R into a Single Reproducible Document' and an 'Edit' button. A 'Add topics' link is also present. Below the title, summary statistics are shown: 1 commit, 1 branch, 0 releases, and 1 contributor. A dropdown menu indicates the branch is 'master'. Buttons for 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download' are available. A list of commits is displayed, each with a small icon, the author's name ('LucyMcGowan'), the commit message, and the date ('Mar 2'). The commits are: 'first commit' (code), 'first commit' (data), and 'first commit' (reports). At the bottom, a note encourages adding a README, with a 'Add a README' button.

**Figure 3:** Follow along:

<https://github.com/LucyMcGowan/report-example>

# Version Control

## Motivation

The screenshot shows a GitHub repository page for the forked repository `LFOD/papr`, which is a fork of `LucyMcGowan/papr`. The repository has 99 commits, 7 branches, 0 releases, and 3 contributors. The main commit list shows various updates to files like `www`, `.gitignore`, `.httr-oauth`, `about.md`, `biorxiv_data.Rda`, `biorxiv_data.csv`, `full_biorxiv_data.csv`, `global.R`, `help.md`, `server.R`, and `term_pca_df.Rda`.

This branch is 53 commits ahead of LucyMcGowan:master.

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

\_nick Strayer updated commented out github install command to reflect the shinysens... [diff] Pull request Compare Latest commit c63cc6f 4 days ago

File	Commit Message	Time Ago
<code>www</code>	Made the rate ui centered.	25 days ago
<code>.gitignore</code>	add in recommendation engine	a month ago
<code>.httr-oauth</code>	I think I got the bug that made you accept the terms twice	5 days ago
<code>about.md</code>	update about section [user]	6 days ago
<code>biorxiv_data.Rda</code>	add recommendor engine	a month ago
<code>biorxiv_data.csv</code>	first commit	6 months ago
<code>full_biorxiv_data.csv</code>	first commit	6 months ago
<code>global.R</code>	Popup seems to be working well	5 days ago
<code>help.md</code>	make about and help match ui	26 days ago
<code>server.R</code>	last little css changes to shinypopup put in	4 days ago
<code>term_pca_df.Rda</code>	update recommender engine [car]	a month ago

# Version Control

## Motivation

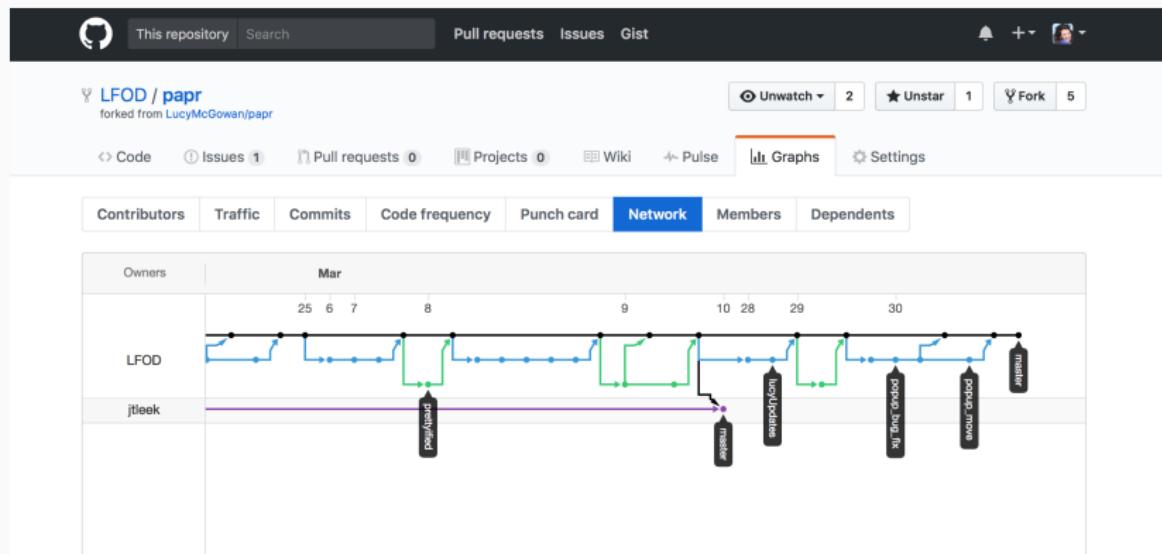


Figure 5: Github branching example

## GitHub

0. If you do not have a GitHub account, you can create one ([github.com](https://github.com)). If you do not have Git installed, install it.

## GitHub

0. If you do not have a GitHub account, you can create one ([github.com](https://github.com)). If you do not have Git installed, install it.
1. Log into your GitHub account and create a repository with the same name as the parent directory.

## GitHub

0. If you do not have a GitHub account, you can create one ([github.com](https://github.com)). If you do not have Git installed, install it.
1. Log into your GitHub account and create a repository with the same name as the parent directory.
2. In the shell, navigate to your project-name folder and type (or copy and paste) the following, changing your-user-name to your GitHub username and project-name to your project name

# Version Control



**Figure 6:** Git as told by [xkcd.com/1597](https://xkcd.com/1597)

# Version Control

```
git init
git add .
git commit -m "first commit"
git remote add origin
    https://github.com/your-user-name/project-name.git
git push -u origin master
```

# Version Control

Every time you make a change to a file, repeat the following commands in the shell

```
git add filename
```

```
git commit -m "describe your file change here"
```

```
git push -u origin master
```

## Version Control

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDFKLJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Figure 7: Git Commits as told by xkcd.com/1296

## GitHub Resources

1. Hu, J. (2013). The Hitchhiker's Guide to Github: SAS Programming Goes Social. SESUG. URL  
<http://analytics.ncsu.edu/sesug/2013/PA-04.pdf>.
2. Philp, S. (2012). An Introduction to Git Version Control for SAS Programmers. WUSS. URL  
<http://www.wuss.org/proceedings12/94.pdf>.
3. Bryan, J. (2017). Happy Git and GitHub for the useR. URL  
<http://happygitwithr.com/>

# Literate Programming

MAN, YOU'RE BEING INCONSISTENT  
WITH YOUR ARRAY INDICES. SOME  
ARE FROM ONE, SOME FROM ZERO.

DIFFERENT TASKS CALL FOR  
DIFFERENT CONVENTIONS. TO  
QUOTE STANFORD ALGORITHMS  
EXPERT DONALD KNUTH,  
"WHO ARE YOU? HOW DID  
YOU GET IN MY HOUSE?"



WAIT, WHAT?

WELL, THAT'S WHAT HE  
SAID WHEN I ASKED  
HIM ABOUT IT.



# Literate Programming

*"I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: "Literate Programming." 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."* (Knuth 1984).

# Literate Programming

explanation + code

# **Setup**

---

# Setup



🎓 SAS University Edition



pdflatex (version 1.3 or later)

pdflatex (version 1.3 or later)

StatRep

## StatRep



open-source software project

## StatRep



open-source software project



reads the code and markup & creates a SAS program

## StatRep



open-source software project



reads the code and markup & creates a SAS program



results that are displayed in your document

## StatRep

<http://support.sas.com/rnd/app/papers/statrep.html>

1. Copy the StatRep macros to a local directory. *Remember where you are saving these macros, you will need this filepath soon.*

## StatRep

<http://support.sas.com/rnd/app/papers/statrep.html>

1. Copy the StatRep macros to a local directory. *Remember where you are saving these macros, you will need this filepath soon.*
2. Install the LATEX package

1. Install [https://www.sas.com/en\\_us/software/university-edition.html](https://www.sas.com/en_us/software/university-edition.html)

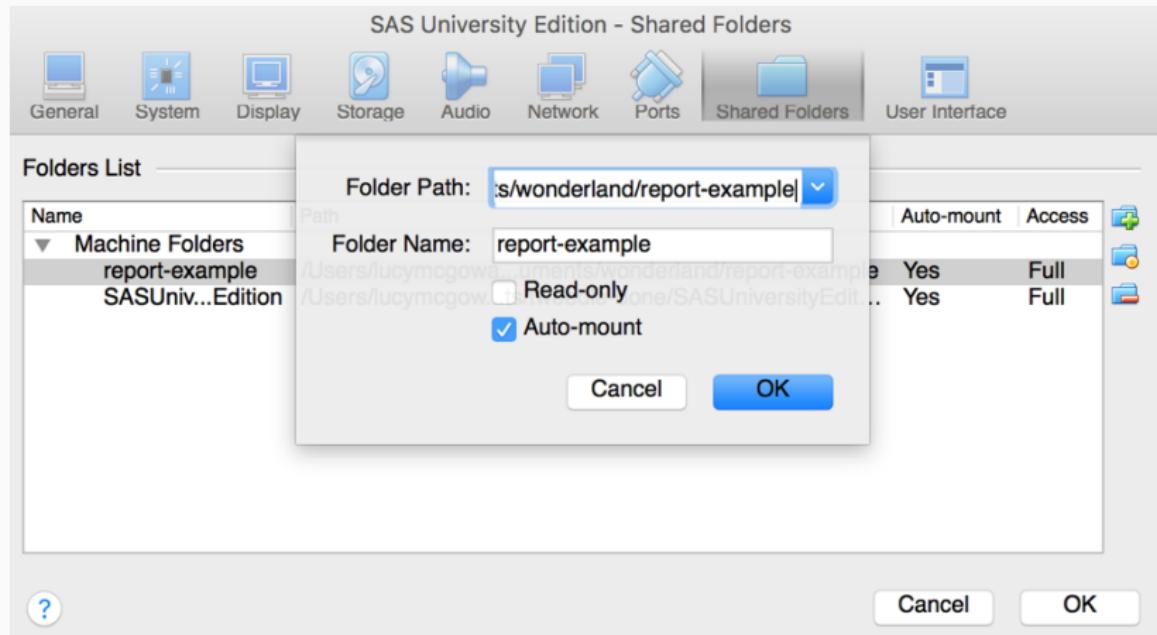
1. Install [https://www.sas.com/en\\_us/software/university-edition.html](https://www.sas.com/en_us/software/university-edition.html)
2. Create a folder with the title of your project, mine will be called report-example

1. Install [https://www.sas.com/en\\_us/software/university-edition.html](https://www.sas.com/en_us/software/university-edition.html)
2. Create a folder with the title of your project, mine will be called report-example
3. Within that folder create 3 folders: **data, code, reports**

1. Install [https://www.sas.com/en\\_us/software/university-edition.html](https://www.sas.com/en_us/software/university-edition.html)
2. Create a folder with the title of your project, mine will be called report-example
3. Within that folder create 3 folders: **data, code, reports**
4. Add project folder to SAS University Edition

1. Install [https://www.sas.com/en\\_us/software/university-edition.html](https://www.sas.com/en_us/software/university-edition.html)
2. Create a folder with the title of your project, mine will be called report-example
3. Within that folder create 3 folders: **data, code, reports**
4. Add project folder to SAS University Edition
5. Add macro folder to SAS University Edition

# SAS University Edition



**Figure 8:** Add a Shared Folder to SAS University Edition

/folders/myshortcuts/report-example/

1. Download R (<https://www.r-project.org>)
2. Install the knitr package:

```
install.packages('knitr')
```

## Methods

---

# Methods

## 1. Document Setup

# Methods

1. Document Setup
2. Programming and Documentation

# Methods

1. Document Setup
2. Programming and Documentation
3. Rendering

## Document Setup

 analysis.Rnw

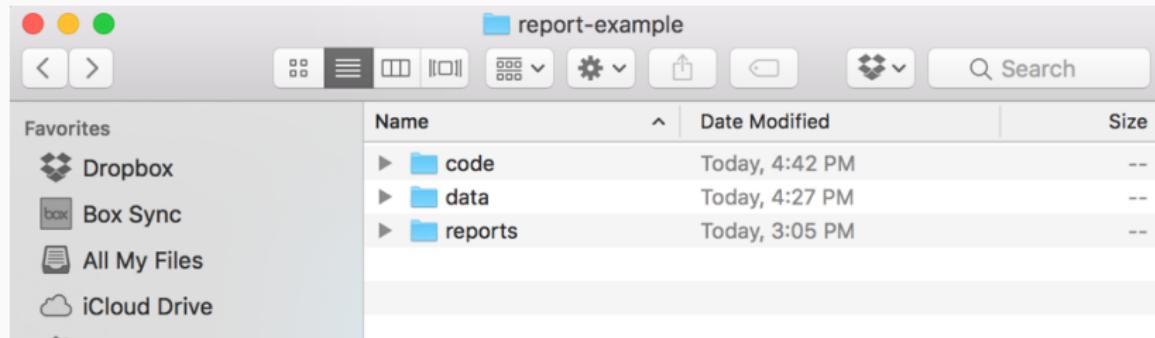
## Document Setup

```
\documentclass{article}
\usepackage{statrep}
\usepackage{parskip,xspace,hyperref}
\def\SRrootdir{/folders/myshortcuts/report-example/code}
\def\SRmacropath{/folders/myshortcuts/statrep_macros.sas}
\title{Your title}
\author{Your name}
\date{\today}
\begin{document}
\maketitle

[we will add code here]

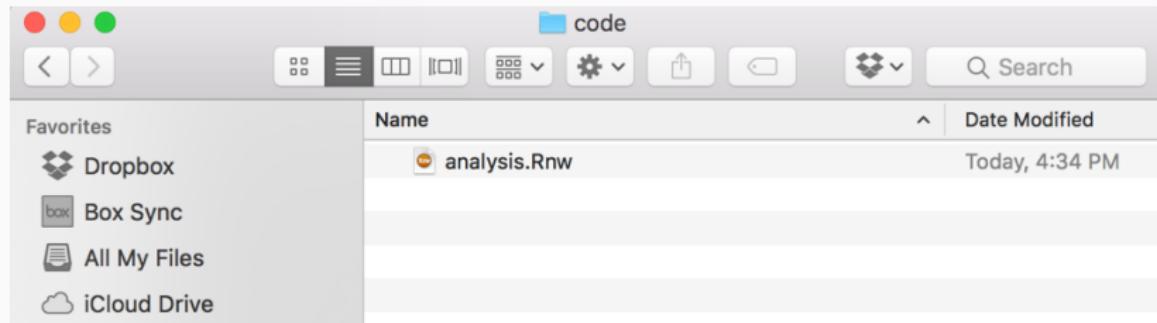
\end{document}
```

# Recap



**Figure 9:** File setup demonstration: project-name

# Recap



**Figure 10:** File setup demonstration: code folder

# Recap

The screenshot shows the 'SAS University Edition - Shared Folders' window. At the top, there is a toolbar with icons for General, System, Display, Storage, Audio, Network, Ports, Shared Folders (which is selected and highlighted in grey), and User Interface. Below the toolbar is a section titled 'Folders List' containing a table.

Name	Path	Auto-mount	Access
Machine Folders			
myfolders	/Users/lucymcgowan/Documents/sas-university-edition/myfolders	Yes	Full
report-example	/Users/lucymcgowan/Documents/wonderland/report-exam...	Yes	Full
sas-macros	/Users/lucymcgowan/Documents/tweedle-done/sas-macros	Yes	Full
Transient Folders			

**Figure 11:** File setup demonstration: SAS University Edition shared folders

# Programming and Documentation



data

# Programming and Documentation

The screenshot shows a GitHub repository page for the user 'fivethirtyeight'. The repository name is 'fivethirtyeight / data'. The page includes a navigation bar with links for 'Pull requests', 'Issues', and 'Gist'. On the right side of the header are buttons for 'Watch', 'Star', 'Fork', and a profile icon. Below the header, there are tabs for 'Code', 'Issues 12', 'Pull requests 10', 'Projects 0', 'Wiki', 'Pulse', and 'Graphs'. A dropdown menu for the branch 'master' is open, showing the path 'data / star-wars-survey /'. To the right of the path are buttons for 'Create new file', 'Upload files', 'Find file', and 'History'. The main content area displays a file named 'StarWars.csv' with the description 'star wars survey data' and a commit message from 'andrewflowers' dated 'Latest commit cc60359 on Jul 22, 2014'. The file was last updated '3 years ago'.

**Figure 12:** Five Thirty Eight Star Wars Survey Data

## Programming and Documentation

 [we will add code here]

# Programming and Documentation

## SAS Chunk

```
\begin{Sascode}  
/* SAS CODE HERE */
```

```
\end{Sascode}
```

# Programming and Documentation

## R Chunk

```
<<>>=
```

```
#R CODE HERE
```

```
@
```

# Programming and Documentation



**Figure 13:** Han shot first (1977)

# Programming and Documentation



**Figure 14:** Greedo shot first (1997)

# Programming and Documentation

## Importing

I obtained data from a 538 Star Wars Survey and will read it into SAS in order to analyze whether age, gender, education level, or whether the individual is a Star Trek fan are associated with incorrectly believing that Greedo shot first.

```
\begin{Sascode}
libname data "/folders/myshortcuts/report-example/data";
filename reffile
  '/folders/myshortcuts/report-example/data/data.csv';
```

## Importing

```
proc import datafile=reffile  
    dbms=csv  
    out=data.starwars;  
    getnames=yes;  
run;  
\end{Sascode}
```

# Programming and Documentation

## Modeling

```
\begin{Sascode}[store = logistic]
ods graphics on;

proc logistic data = data plots = oddsratio;
  class age (ref = FIRST) gender college star_trek_fan;
  model wrong (event = "1") = age gender college
    star_trek_fan;
run;

ods graphics off;
\end{Sascode}
```

## Results

```
\Listing[store = logistic,  
objects = OddsRatios,  
caption = {Wrong about who shot first OR}]{logisticOR}
```

# Programming and Documentation

## Results

The LOGISTIC Procedure				
Odds Ratio Estimates				
Effect		Point Estimate	95% Wald Confidence Limits	
Age	30-44 vs 18-29	1.566	0.941	2.605
Age	45-60 vs 18-29	1.232	0.730	2.079
Age	> 60 vs 18-29	2.190	1.218	3.936
Gender	Fema vs Male	1.402	0.970	2.026
college	College degree vs No college deg	1.027	0.705	1.497
Star_Trek_fan	No vs Yes	1.194	0.818	1.742

**Figure 15:** Wrong about who shot first Odds Ratios

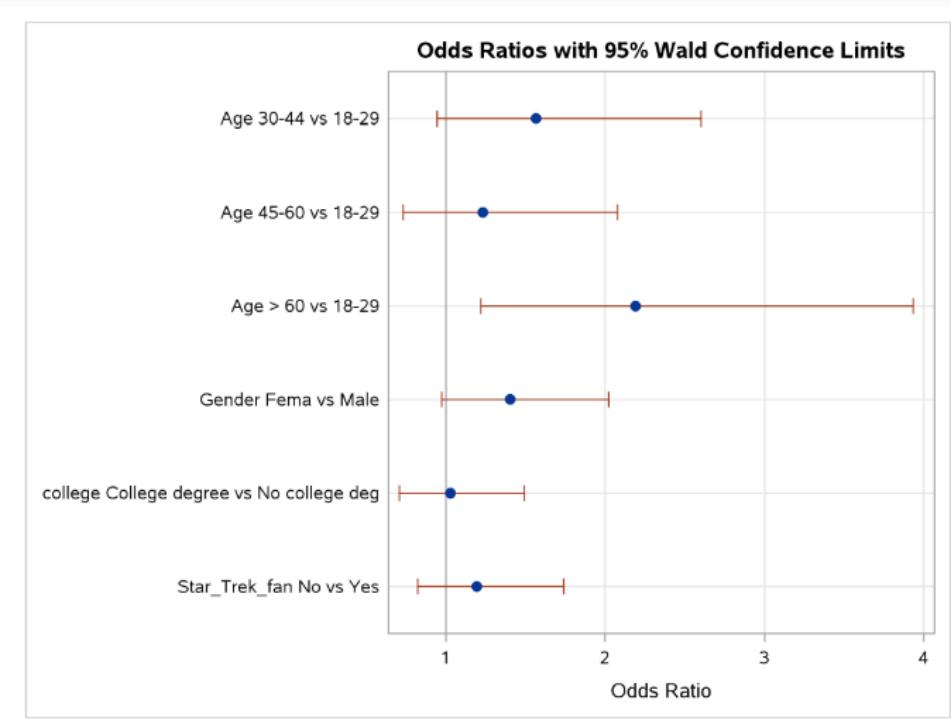
# Programming and Documentation

## Results

```
\Graphic[store=logistic,  
objects=ORPlot,  
caption={Wrong about who shot first plots}]{ORplot}
```

# Programming and Documentation

## Results



# Programming and Documentation

## Export

```
\begin{Sascode}
proc export data=data.starwars
    outfile =
        '/folders/myshortcuts/report-example/data/data_.csv'
replace
dbms = dlm;
delimiter = ',';
run;
\end{Sascode}
```

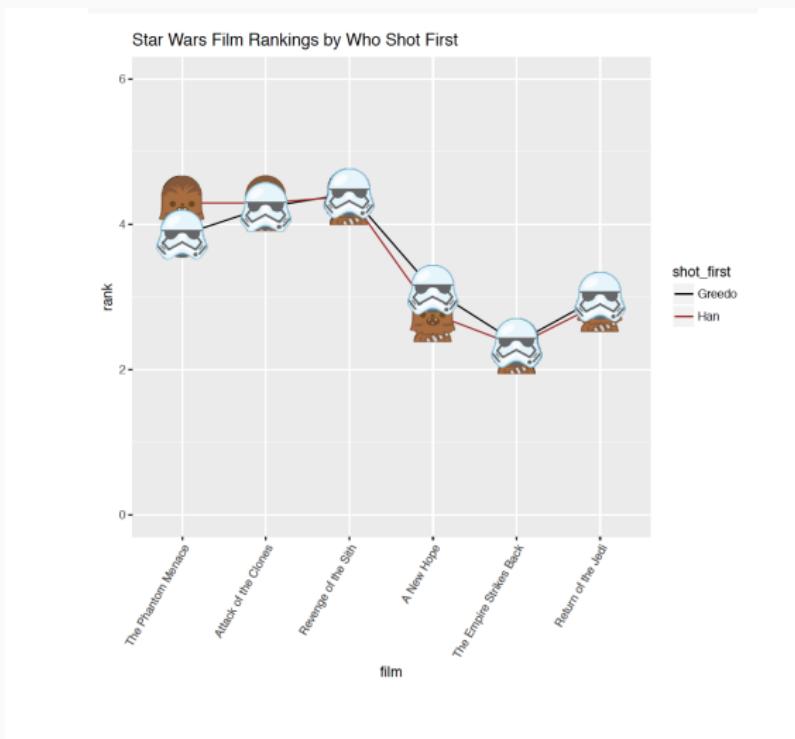
# Programming and Documentation

## R Code

```
<>>=  
filename = ".../data/starwars_sasedit.csv"  
if (file.exists(filename)){  
  
  starwars <- read.csv(filename)  
  
  # more R code here  
}  
@
```

# Programming and Documentation

R



## Rendering

1. Compile the .Rnw file. This is completed using knitr. In the shell, navigate to your code folder where analysis.Rnw is located and submit: RScript -e "library(knitr); knit('analysis.Rnw')"

## Rendering

1. Compile the .Rnw file. This is completed using knitr. In the shell, navigate to your code folder where analysis.Rnw is located and submit: RScript -e "library(knitr); knit('analysis.Rnw')"
2. Compile the pdflatex. Run the following in the shell: pdflatex analysis.tex

## Rendering

1. Compile the .Rnw file. This is completed using knitr. In the shell, navigate to your code folder where analysis.Rnw is located and submit: RScript -e "library(knitr); knit('analysis.Rnw')"
2. Compile the pdflatex. Run the following in the shell: pdflatex analysis.tex
3. Run the SAS code. Open SAS University Edition. Navigate to your code folder (it will be under Server Files and Folders). Open and run analysis\_SR.sas.

## Rendering

1. Compile the .Rnw file. This is completed using knitr. In the shell, navigate to your code folder where analysis.Rnw is located and submit: RScript -e "library(knitr); knit('analysis.Rnw')"
2. Compile the pdflatex. Run the following in the shell: pdflatex analysis.tex
3. Run the SAS code. Open SAS University Edition. Navigate to your code folder (it will be under Server Files and Folders). Open and run analysis\_SR.sas.
4. Re-compile the pdflatex. Run the following in the shell: pdflatex analysis.tex

## References

1. Arnold, T., & Kuhfeld, W. F. (2012). Using SAS and LATEX to Create Documents with Reproducible Results. URL <http://support.sas.com/resources/papers/proceedings12/324-2012.pdf>.
2. Arnold, T. (2015). The StatRep System for Reproducible Research: A Note for SAS University Edition Users. URL <http://support.sas.com/rnd/app/papers/statrep/statrepUE.pdf>.
3. Bryan, J. (2017). Happy Git and GitHub for the useR. URL <http://happygitwithr.com/>
4. FiveThirtyEight. (2014). America's Favorite 'Star Wars' Movies (And Least Favorite Characters). URL <https://fivethirtyeight.com/datalab/americas-favorite-star-wars-movies-and-least-favorite->  
Data found at: <https://github.com/fivethirtyeight/>

## References

5. Hu, J. (2013). The Hitchhiker's Guide to Github: SAS Programming Goes Social. SESUG. URL <http://analytics.ncsu.edu/sesug/2013/PA-04.pdf>.
6. Knuth, D. E. (1984). Literate programming. *The Computer Journal*, 27(2), 97-111.
7. Philp, S. (2012). An Introduction to Git Version Control for SAS Programmers. WUSS. URL <http://www.wuss.org/proceedings12/94.pdf>.
8. Wickham, H. (2015). R packages. O'Reilly Media, Inc.

# Contact

Lucy D'Agostino McGowan

Vanderbilt University

 ld.mcgowan@vanderbilt.edu

 www.lucymcgowan.com

 LucyMcGowan

 @LucyStats