UNR Data Science Conference 2025 – 20 February 2025

Document Preparation: From R and Python Notebooks to Final Documents using LaTeX

Paul J. Hurtado & Theresa McKim
University of Nevada, Reno

Pre-session survey link & QR code

https://forms.gle/R9SoByuQE1oD7PEJ6

www.pauljhurtado.com ← **Download workshop files**
theresamckim.wordpress.com

Introduction

Workshop Outline

Introduction (15 min)

- Introductions
- Workflow overview: from scripts to final reports
- Overview: R Markdown in R Studio, TeXstudio for LaTeX

Section 1: R Markdown (45 min)

- Basics of R Markdown using R Studio: Syntax, creating documents, and embedding code
- Customizing outputs: HTML, PDF, Word
- LaTeX syntax for formatting equations & formulas
- Activity: Reformatting an R script as an R Markdown document. (15min)

Section 2: Python Notebooks (45 min)

- Overview of Colab, Jupyter Notebooks, & Anaconda for Python
- Writing and running Python code in Colab & Jupyter
- Creating shareable notebooks with annotations and visualizations
- Activity: Work in a Colab notebook & visualize data (15 min)

Section 3: LaTeX (30min):

- Overview of LaTeX for professional-quality documents
- Templates for academic and technical documents
- Embedding R or Python code in a LaTeX document for advanced formatting
- Activity: Add R code to a LaTeX (knitr) document (15min)

Wrap-up (15 min):

- Closing remarks & Resources for further learning

Meet your Instructors

Dr. Paul Hurtado

- Mathematics & Statistics Faculty, UNR
- Ecology, Evolution, & Conservation Biology (EECB) Graduate Program faculty member
- "Stats Chats" Co-facilitator
- Ph.D. in Applied Mathematics

  Research: Mathematical Biology
- Birding! NV Bird Records Committee member, eBird regional reviewer, etc.

Dr. Theresa McKim

- Biology Faculty, UNR
- Assistant Teaching Professor & Faculty Advisor (Neuroscience major)
- Ph.D. in Psychology & Neuroscience
- Courses taught: Intro to Data Science (Python), Neurobiology, Neuroscience techniques lab course
- Research interests: human neuroimaging (fMRI & EEG) and brain connectomics
- For fun: yoga 🧘‍♀️, baking 🍪, traveling 🗺️

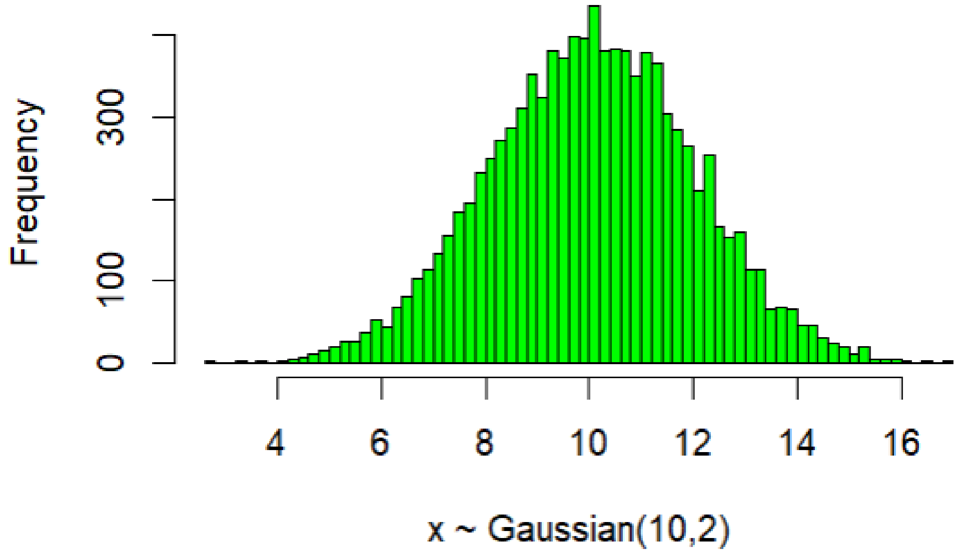
Documents, Directories, & Project Workflows

A common project progression from start to finish:

- Preliminary scripts, data, literature review (directory of PDFs):
  - Data prep/cleaning, data visualization, preliminary analyses
  - Organize your files thoughtfully!
- “Laboratory notebooks”:
  - These add context and organize information better than scripts alone: progress notes, important theory/methodology, organized outputs, etc.
  - Preliminary drafts of final results & outputs (figures, tables, etc.)
- Final products:
  - Technical reports, manuscript drafts and preprints, thesis chapters, publication submissions, book chapters, etc…

What is **R**?

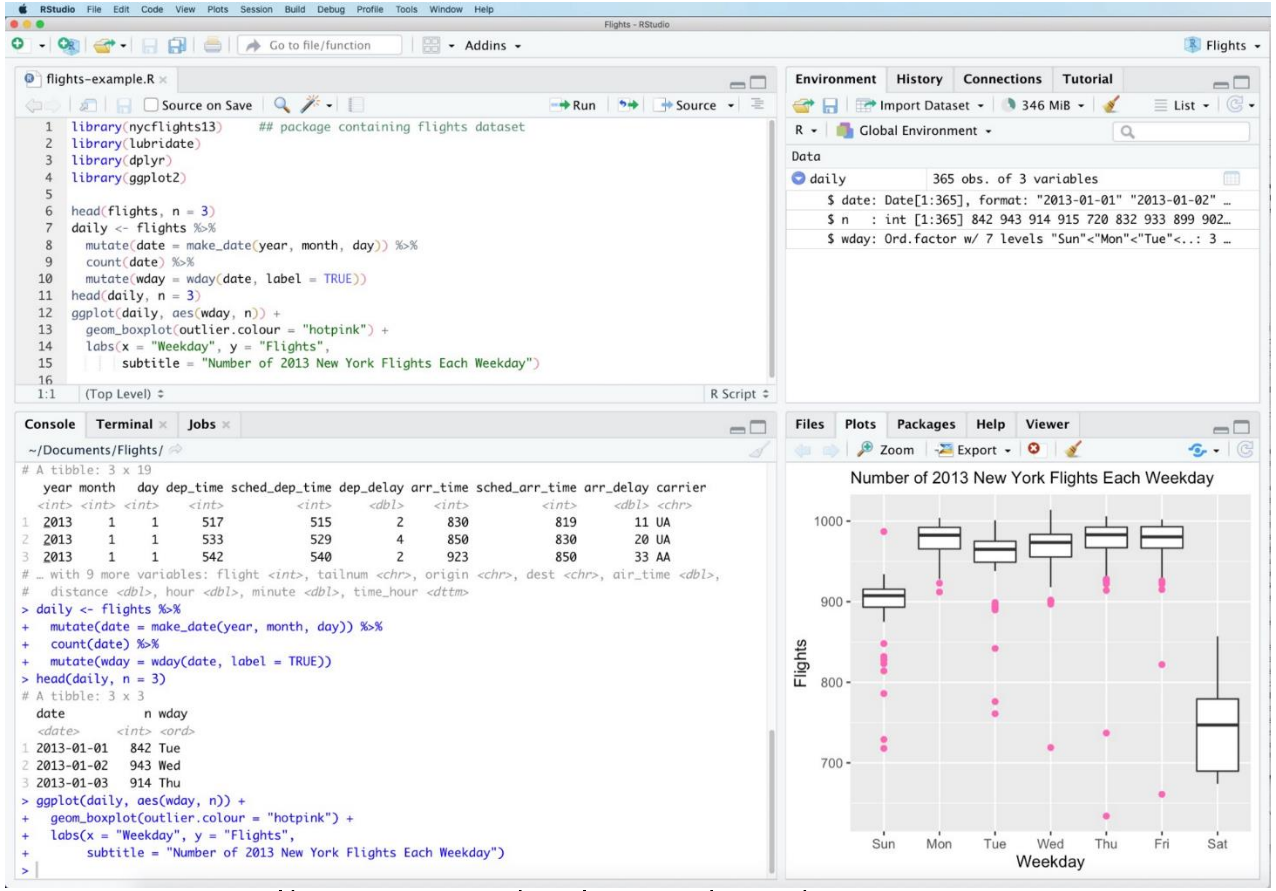
- "High level" programming language primarily for statistics applications
- Command line or script based
- Free, open source, cross-platform
- Modular (R Packages) like Python
- … not fast
- … not a CAS (computer algebra system)
- … not menu based (cf. Excel)
- … not the only option for Data Science




```
>
> 1+1
[1] 2
> sin(pi)
[1] 1.224606e-16
> sin(pi/2)
[1] 1
> sin(0)
[1] 0
> seq(0,10,by=2)
[1]  0  2  4  6  8 10
> hist(rnorm(10000,10,2),n=100,
+      col="green",main="",
+      xlab="x ~ Gaussian(10,2)")
> |
```

What is **R Studio**?

- An IDE (“front end”) for R
  - An improved interface for R
- Free and open source
- Added functionality
  - Syntax highlighting
  - Code completion
  - Project organization tools
  - Expanded document types, e.g., R Markdown “vignettes”
  - Server-to-browser interface
- Cross platform support



Source: https://en.wikipedia.org/wiki/RStudio#/media/File:RStudio_IDE_screenshot.png

What is **R Markdown**?

- A “scripting language” that integrates **formatted text**, **R code**, and **R code output** (figures, tables, etc.) in a single document.
- R Markdown (source) documents can use a mix of text formatting:
  - R Markdown + HTML and CSS
  - LaTeX equations (rendered by MathJax)
  - Python and other code works too!
    Example: https://pauljhurtado.com/R/RmdIntro.html (RmdIntro.Rmd)
- Output document formats can be either HTML, PDF, or Word
- Dynamic documents!
  - Interactive elements
  - Automatically update when given new data inputs

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins Project: (None)

exam1-review-solns.Rmd

Knit on Save Knit Run

```
to
$\infty$ and $-\infty$. Determine all horizontal and vertical asymptotes
of the graph $y = f(x)$ and sketch the function.

<div style="box-shadow: 5px 5px 5px grey; background-color:#E0E0E0;
margin:1em; padding:1em;">
**Solution:** As above, we can either divide numerator and denominator by $x$
to find the limit, or we can use the pre-calc shortcut and take the ratio of
terms with the largest degree, which gives
$$\lim_{x \to -\infty} \frac{2x + 3}{x - 1} = \lim_{x \to -\infty}
\frac{2x}{x} = 2$$
```

$$\lim_{x \to -\infty} \frac{2x+3}{x-1} = \lim_{x \to -\infty} \frac{2x}{x} = 2$$

```
$$\lim_{x \to \infty} \frac{2x + 3}{x - 1} = \lim_{x \to \infty} \frac{2x}{x}
= 2$$
```

$$\lim_{x \to \infty} \frac{2x+3}{x-1} = \lim_{x \to \infty} \frac{2x}{x} = 2$$

```
Therefore the function has a single horizontal asymptote at $y=2$.

Since the denominator has a zero at $x=1$ (which is not a zero of the
numerator), there is a vertical asymptote at $x=1$.

```{r, fig.align="center", echo=FALSE}
x=c(seq(-12,1-1e-10,length=300), 1, seq(1+1e-10,17,length=300))
plot(x,(2*x+3)/(x-1)*ifelse(x==1,NA,1),type="l",lwd=3, ylim=c(-10,10),
xlim=c(-10,15), ylab="f(x)")
abline(v=1,h=2,lty=3)
abline(v=0,h=0,lty=1)
```
</div>

11. Let $f(x) = x^2 + 3x$.
  a) Find the slope of the secant joining the points $P=\big(-2, f(-2)\big)$
and
```

14:15 (Top Level) R Markdown

Console Terminal Render Jobs

.../exam-reviews/exam1-review-solns.Rmd

```
t.md --to html4 --from markdown+autolink_bare_uris+tex_math_single_backslash --output
 exam1-review-solns.html --lua-filter "C:\Users\phurtado\Documents\R\win-library\4.1
\rmarkdown\rmarkdown\lua\pagebreak.lua" --lua-filter "C:\Users\phurtado\Documents\R\w
in-library\4.1\rmarkdown\rmarkdown\lua\latex-div.lua" --self-contained --variable bs3
=TRUE --standalone --section-divs --template "C:\Users\phurtado\Documents\R\win-libra
ry\4.1\rmarkdown\rmd\h\default.html" --no-highlight --variable highlightjs=1 --variab
le theme=bootstrap --include-in-header "C:\Users\phurtado\AppData\Local\Temp\RtmpOG6A
```

C:/Users/phurtado/Dropbox/UNR2015/Teaching/0-MATH-181-2025-SP/exam-reviews/exam1-review-solns.html

exam1-review-solns.html Open in Browser Find Publish

10. Let $f(x) = \frac{2x+3}{x-1}$. Find the limits of f as x goes to ∞ and $-\infty$. Determine all horizontal and vertical asymptotes of the graph $y = f(x)$ and sketch the function.

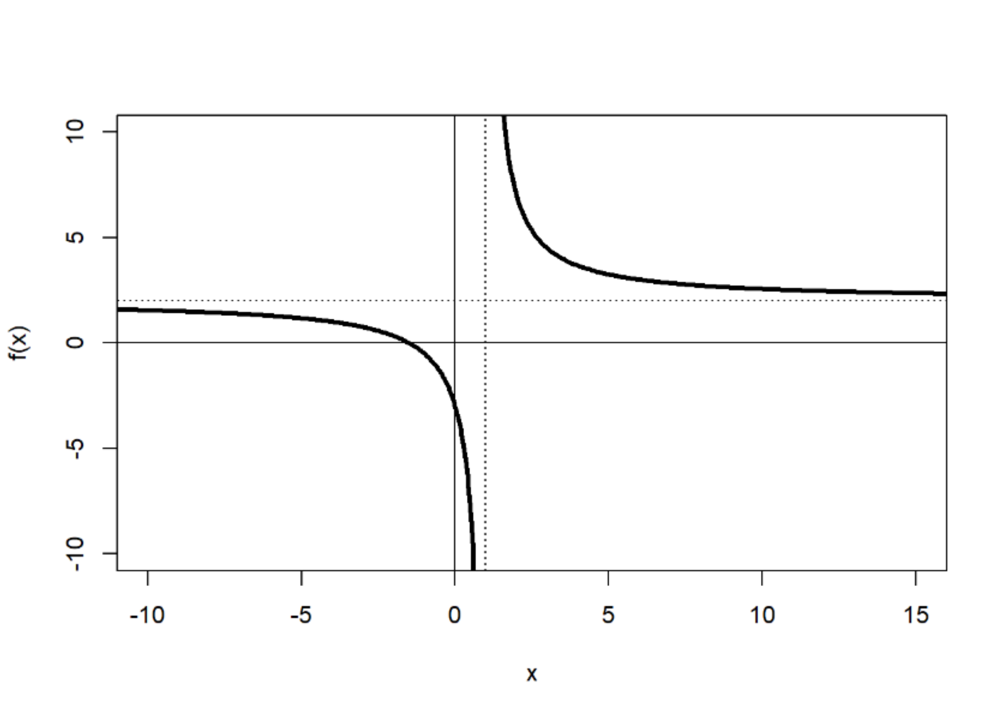
Solution: As above, we can either divide numerator and denominator by x to find the limit, or we can use the pre-calc shortcut and take the ratio of terms with the largest degree, which gives

$$\lim_{x \to -\infty} \frac{2x+3}{x-1} = \lim_{x \to -\infty} \frac{2x}{x} = 2$$

$$\lim_{x \to \infty} \frac{2x+3}{x-1} = \lim_{x \to \infty} \frac{2x}{x} = 2$$

Therefore the function has a single horizontal asymptote at $y = 2$.

Since the denominator has a zero at $x = 1$ (which is not a zero of the numerator), there is a vertical asymptote at $x = 1$.



What is **LaTeX/TeX**?

- Software for typesetting documents (launched in 1984!)
- LaTeX markup describes the **layout**, **content**, and **content formatting** of the document; text script is parsed to produce PDF outputs.
- Widely used for journal article submissions, technical books, graduate theses, etc. particularly in the mathematical sciences.
- Challenging learning curve, ***but*** …
  - **Equation** formatting is exceptional!
  - **Automatic numbering and referencing** of figures, tables, citations, etc!
  - Easy to reformat/restyle documents (e.g. from thesis chapter to preprint to journal article submission)
  - Sweave/knitr: R code can be embedded just like an R Markdown document!
- **TeXstudio** is a super helpful interface for learning and using LaTeX

LaTeX Example: Equations

AMS Short Math Guide for LaTeX

https://tug.ctan.org/info/short-math-guide/short-math-guide.pdf

LaTeX Script

```
Let $f(x)=3x^2+1$, then …\\
Since $y_i=\beta_0+\beta_1\,x_i+\epsilon_i$\\
\begin{align}
  A=\begin{bmatrix} a & b\\ c & d\end{bmatrix}
\end{align}
```

Output

Let $f(x)=3x^2+1$, then …
Since $y_i=\beta_0+\beta_1\,x_i+\epsilon_i$

$$A=\begin{bmatrix} a & b\\ c & d\end{bmatrix} \tag{1}$$

$$\frac{dx}{dt} = rx\left(1-\left(\frac{x}{K}\right)^{\theta}\right) \tag{36}$$

has an analytical solution, then modify the code above to find a numerical solution using `ode` in R. Use initial conditions $x(0) = 10$, $r = 1$, and $K = 1000$, and explore θ values above, below, and at $\theta = 1$. Plot both curves together on the same plot to compare the exact and approximate solution curves.

Keeping Numerical Solutions Positive: The Log-Transform Trick

The above approach to computing numerical solutions to ODEs can sometimes be problematic for ODE models of biological systems. Often, in biological models (e.g., like the SIR model mentioned above) one or more equations are of the form

$$\frac{dx}{dt} = g(x)x. \tag{37}$$

Importantly, this implies that trajectories $x(t)$ slow as they asymptotically approach $x = 0$, and therefore can never pass through $x = 0$. However, numerical methods may take an approximated, discrete-time step that results in the simulated x trajectory erroneously crossing zero (which should be impossible!). This can lead to state variables running off to $\pm\infty$, and other undesirable behaviors. Fortunately, there is a clever trick that allows us to avoid these numerical errors: if we assume that $x(t) > 0$ over the time period for which we seek a numerical solution, then we can let $X = \log(x)$ and by the properties of the natural log function[14] it follows that

$$\begin{aligned}\frac{dX}{dt} &= \frac{d}{dt}\log(x) = \frac{1}{x}\frac{dx}{dt} \\ &= \frac{1}{x}g(x)x = g(\exp(X))\end{aligned} \tag{38}$$

That is, if we transform a strictly positive state variables x to a real-valued variable X using the natural log function, then numerical solutions of X can take on both positive and negative values that can later be transformed back to strictly positive x values, since $x = \exp(X)$. This transformation of variables is the default approach some modelers take to computing numerical solutions of models where trajectories should remain strictly positive over a finite time interval.

One caveat to this approach is that, since $X(t) \to -\infty$ as $x(t) \to 0$, it can lead to numerical errors due to finite limits on the size of floating point values on a computer. While it may seem contradictory to trade one numerical problem for another, it is usually far less common (and much more manageable) in practice to have a numerical solution diverging to $-\infty$ and hitting the computer's floating point limit. Thus, this approach should be reserved for situations where $x(t)$ remains bounded away from 0 or converges slowly to 0. A second caveat is that often times solutions

[14] Here the notation `log(x)` is used for the natural log function, instead of `ln(x)`, following the convention used in most modern scientific programming languages. Likewise, `exp(x)`$=e^x$.

that erroneously cross through 0 do so because of typos in the code that defines the derivative function. Thus, carefully checking that the equations, as implemented in the code, are free of errors is strongly advised before assuming that numerical methods are to blame for seemingly erroneous numerical solutions.

To illustrate this approach, here is the above SIR model code rewritten in terms of the log-transformed values $X = \log(S)$ and $Y = \log(I)$ where we have also omitted the R equation since the constant population size allows us to calculate $R(t)$ once the $S(t)$ and $I(t)$ curves are obtained using $R(t) = N(0) - S(t) - I(t)$.

```
# A function for ode() to compute numerical solutions
# of the log-transformed SIR model.
logSI = function(tval, Xs, params) {
   X = Xs[["X"]]        # X = log(S)
   Y = Xs[["Y"]]        # Y = log(I)
   B=params[["beta"]]
   g=params[["gamma"]]
   # Now compute the derivatives
   dX = -B*exp(Y)       # I = exp(Y)
   dY = B*exp(X) - g    # S = exp(X)
   # Return the derivatives in a list to use with ode()
   return(list(c(dX,dY)))
}

# Initial conditions, log-transformed
logX0 = c(X=log(X0[["S"]]), Y=log(X0[["I"]]))
logXout = ode(logX0, Time, logSI, Pars) # requires deSolve

# Convert back to S and I, and calculate R
Xout = cbind(Time,S=exp(logXout[,2]), I=exp(logXout[,3]),
            R=sum(X0) - exp(logXout[,2]) - exp(logXout[,3]))
head(Xout,3)
```

```
          Time        S        I             R
[1,] 0.0000000 998.0000 2.000000 -2.273737e-13
[2,] 0.7537688 997.5126 2.324242  1.631915e-01
[3,] 1.5075377 996.9464 2.700730  3.528366e-01
```

```
# Plot the numerical solution curves as before
par(mfrow=c(1,3)) # 3 panels in 1 row
plot(Time, Xout[,2], ylab="S", ylim=c(0,sum(X0)),
    type="l", lwd=2);   abline(h=0)
plot(Time, Xout[,3], ylab="I", ylim=c(0,sum(X0)),
    type="l", lwd=2);   abline(h=0)
plot(Time, Xout[,4], ylab="R", ylim=c(0,sum(X0)),
    type="l", lwd=2);   abline(h=0)
```

Exercise 8. Modify the code above to simulate the Rosenzweig–MacArthur model, eqs. (23), using the parameter values from Fig. 1 and using both direct implementation and the log-transformation technique. Plot both results to compare approaches.

Challenge Problem 11. Look at the different methods available through the `ode` function in the `deSolve` package ([156]; type `?ode` into R to view the documentation). When should you use one method over the other? Compare these to

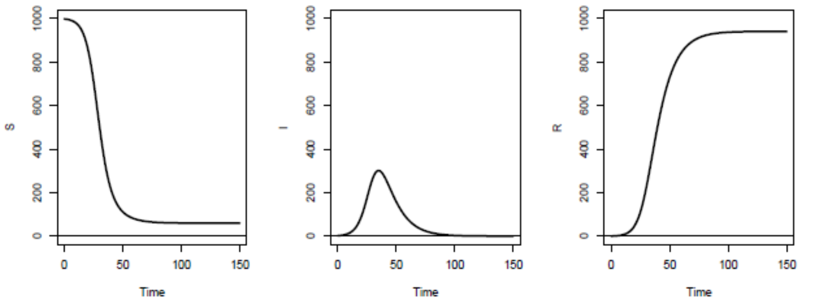


Fig. 3 A numerical solution to the log-transformed SIR model using the same parameters and initial conditions as in Fig. 2. See the code in the main text for details.

the methods available in Matlab, Python, or other computing platforms (see also [138, 115]).

Challenge Problem 12. It can be challenging to obtain numerical solutions to stiff systems. Find resources like [116, 138] to help you better understand which numerical methods work well for stiff (multiple time scale) systems in the R package `deSolve` as well as in Matlab or similar software, and look up some stiff models to use as examples. Use the different methods to find numerical solutions for the same parameter values and time values, and compare these to methods that should perform poorly (e.g., Euler). How different are the results from different methods? How do R's stiff solvers compare to those in Matlab or other software?

2.7 *ODEs as Statistical Models*

This final section in §2 introduces some useful statistical concepts and basic approaches to using an ODE model as the basis for a statistical model.[15] Increasingly, ODE models are being used as components of statistical models, e.g., for making inferences about underlying mechanisms or for forecasting. Applications include making inferences from model parameters estimated from time series data (i.e., a sequence of state variable measurements taken at multiple time points, that in some way corresponds to a sequence of state variable values), quantifying uncertainty in those parameter estimates, forecasting, and conducting statistical tests to determine whether a certain parameter is significantly different from zero or whether one model fits the data better than another model (i.e., *model selection*).

[15] For related resources in R, see the packages such as `CollocInfer` [87], `deBInfer` [19], or browse the relevant CRAN Task Views (`https://cran.r-project.org/web/views/`).

Part I: R and R Markdown

R Studio & R Markdown Basics

- Coding in R: R Scripts
  - R scripts & R Studio *projects*
  - support.posit.co/hc/en-us/articles/200526207-Using-RStudio-Projects
- R Markdown
  - Text Formatting
  - Code Chunks
  - Output types
  - Other document types: R slides, etc.
- R Studio IDE
- Equations using LaTeX

R Coding in a Research Context

- R Scripts: simplest "R code" files
- R Studio *projects* let scripts & markdown documents **operate from a "home" directory** using *relative* file names.
- It's good to have a well thought out directory structure for your projects!

  Example directory structure from https://martinctc.github.io/blog/rstudio-projects-and-working-directories-a-beginner's-guide/

R Markdown

- Flexible mix of **R code**, **formatted text**, and **R output**.
- An excellent medium for a notebook!
  - Document your work activities
  - Document data prep, analyses, important decisions, results, etc.
  - Draft analyses and prepare content for final products
  - Archive your work so that others (including “future you”) can review it
- Getting started is EASY in R Studio: **File > New File… > R Markdown**
  - Template includes a link to rmarkdown.rstudio.com (a great resource!)

Exercise #1: Make a new R Markdown document, and let’s practice some basic text formatting and modifying “code chunks”.

- **Optional “Homework”:** View and modify the slide (*.Rpres) example.

R Markdown Exercise: NV Bird Diversity

Exercise #2: ***Do county area & population size correlate with observed county level bird diversity in Nevada?***

- Open the “NV Bird Example/” directory
- **Main goal:** Convert the R script “**scratch.R**” to an R Markdown document
    - Add sections to organize the document
    - Add equations for the linear regressions
    - Add notes and other details that feel like they’re missing
    - Customize figures and other outputs

Consult the “cheat sheet” built into R Studio, and other resources, as needed!
https://rmarkdown.rstudio.com/lesson-15.html

Workflow & R Markdown Remarks

<u>Project Workflow: Start to Finish</u>

- **Organize files thoughtfully** – what are your goals & final products?
  - "Scratch" scripts and written notes → notebook docs → Final products
  - Well documented, reproducible records of your activities
- Target audience: (1) "Future you"; (2) Collaborators, etc.; (3) 3rd parties?

<u>R Markdown</u>

- Find **useful resources** & bookmark them; borrow code from other docs!
- **Learn R Studio**'s capabilities and built-in resources
- Learn some HTML and LaTeX, but anticipate quirky behavior
- Version control? Low-tech tip: Save copies with dates appended to file names.

Part II: Python Notebooks

Part III: LaTeX

What is **LaTeX/TeX**?

- Typesetting software (launched in 1984!): parses text into a PDF
- LaTeX markup specifies document layout and content.
- Widely used for journal article submissions, technical books, graduate theses, etc. particularly in the mathematical sciences.
- There’s a learning curve, ***but*** …
  - Equation formatting is exceptional!
  - Automatic numbering of, and referencing of, figures, tables, citations, etc!
  - Easy to reformat documents (e.g. from thesis chapter to preprint server upload to journal article submission)
  - Sweave/knitr: R code can be embedded just like an R Markdown document!
- **TeXstudio** is a super helpful interface for learning and using LaTeX

LaTeX Basics (Part I): Text & Equations

There are many excellent resources for learning LaTeX

- https://www.pauljhurtado.com/latex has links to some resources, and and an example-based overview of editing LaTeX files
- Tip: gain familiarity by modifying existing documents and templates!

Let's begin by exploring TeXstudio

- Open TeXstudio, then make a new document:
  File > New from template… > article
- Explore the "LaTeX" menu.
- Explore the ***AMS Short Math Guide for LaTeX*** linked on the next slide

LaTeX Example: Equations

AMS Short Math Guide for LaTeX
https://tug.ctan.org/info/short-math-guide/short-math-guide.pdf

LaTeX Script

```
Let $f(x)=3x^2+1$, then …\\
Since $y_i=\beta_0+\beta_1\,x_i+\epsilon_i$\\
\begin{align}
  A=\begin{bmatrix} a & b\\ c & d\end{bmatrix}
\end{align}
```

Output

Let $f(x) = 3x^2 + 1$, then …
Since $y_i = \beta_0 + \beta_1\,x_i + \epsilon_i$

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{1}$$

Exercise: Open a new *article* or *report* and recreate the output above.

LaTeX Basics (Part II): Document layout

- The main **content** of the document is after “\begin{document}”
- **Above that** are commands that specify document style and format
- Explore the following:
  - \usepackage{geometry}
    \geometry{top=1in,left=1.5in,right=1in,bottom=1.25in}
  - \usepackage{lineno}
  - \usepackage{parskip}
  - Other options for \documentclass?
- **Optional:** Visit your favorite quantitative journal’s website, find their author instructions, and download their LaTeX template.

LaTeX Examples: Templates

Activity: Explore these templates to see how their “headers” differ

- Homework
- Presentation Slides (Beamer)
- Poster
- Thesis
- ArXiv template (arxiv.org is a preprint server for mathematics, statistics, mathematical sciences started in 1991; it inspired bioRxiv & medRxiv)

Explore some of the LaTeX content as well, time permitting!

Combining LaTeX + R: Sweave/knitr

- The *knitr* package lets us embed blocks of R code (nearly identical to R Markdown code chunks) into LaTeX documents!

- If TeXstudio has not been configured to build Sweave files using knitr, see details at www.pauljhurtado.com/teaching/software.html

- You can also include Python and other code! For an example, see
  pauljhurtado.com/R/tutorial-intro-040716.pdf
  pauljhurtado.com/R/tutorial-files-040716.zip (see *.Rnw file)

Exercise: Add a code-generated figure one of the knitr templates.

LaTeX Remarks

LaTeX equations for R Markdown

- The **AMS Short Math Guide to LaTeX** is an excellent resource!
- Some environments require additional "math mode" delimiters ($)
- Equation numbering is possible, but it requires some extra setup.
  Example: https://pauljhurtado.com/R/RmdIntro.html (RmdIntro.Rmd)

LaTeX and knitr/Sweave Documents

- Final products: LaTeX with R "code chunks" parsed by knitr in Texstudio.
- Develop R code in R Studio ***then*** copy (& reformat) text, eqs., & code

Closing Remarks

Documents and Workflows

- Organize your project files with the final product(s) in mind!
- R scripts are your “scratch paper”
- R Markdown documents are your “lab notebook”
  - Document your work, successes and failures!
  - “Future you” appreciates well annotated code and thorough explanations ☺
  - Version control can be helpful
    - Fancy options like GitHub are great if you know how to use them!
    - Low tech options work too! **Ex:** Save copies of files with dates appended to the names.
- Results can then be polished and moved into their final forms
  - Presentation slides, thesis chapters, preprint articles, etc.

R / R Markdown & LaTeX Resources

- **www.pauljhurtado.com/R** ← Links to other resources!
- R Tutorial: kevintshoemaker.github.io/R-Bootcamp/
- bookdown.org/yihui/rmarkdown/ & yihui.org/knitr/
- docs.posit.co/connect/user/rmarkdown/
- rstudio.github.io/cheatsheets/html/rmarkdown.html

- **www.pauljhurtado.com/latex** ← Links to **templates** & other resources!
- LaTeX equations: ctan.org/tex-archive/info/**short-math-guide**
- Short Course: ctan.org/tex-archive/info/examples/Math_into_LaTeX-4
- BibTex reference manager: www.jabref.org