

# QSS20: Modern Statistical Computing

Unit 05: LaTeX and catchup (GitHub & functions)




# Goals for today

- ▶ **Project & assignment updates**
- ▶ Git/GitHub: [lecture & activity](#), review pset2 submission instructions (from previous class)
- ▶ User-defined functions: [lecture](#) & [activity](#) (from last week)
- ▶ LaTeX/Overleaf

# Where we're headed

## DataCamp deadlines:

- ▶ **Wednesday 09/26:** Chapters on exact matching
- ▶ **Wednesday 10/05:** Chapter on regular expressions for matching

TITLE ▾	ASSIGNEES ▾	STATUS	DUE BY ▴
 Joining Data with pandas Data Merging Basics Chapter	Organization	<b>DUE SOON</b>	Sep 28, 15:30 EDT
 Joining Data with pandas Merging Tables With Different Join Types Chapter	Organization	<b>DUE SOON</b>	Sep 28, 15:30 EDT
 Regular Expressions in Python Regular Expressions for Pattern Matching Chapter	Organization	Active	Oct 5, 15:30 EDT

## Class and problem sets:

- ▶ **This Wednesday & next week in class:** exact merging, regular expressions, probabilistic merging
- ▶ **Grades for pset1:** if you submitted on time, hopefully next Monday
- ▶ **Sunday 10/09 at 11:59 PM:** problem set two due

## More about START model (for SIP)

- ▶ SIP data dictionaries now available for [START Information Reporting System \(SIRS\)](#) and soon for medical student training data
- ▶ You can [read the START SIP proposal here](#)
- ▶ From [website of National Center of START Services](#):

*The START Model was developed and implemented in 1988 by Dr. Joan B. Beasley and her team to provide community-based crisis intervention for individuals with intellectual and developmental disabilities and mental health needs. The model is evidence-informed and utilizes a national database. It is a person-centered, solutions-focused approach that employs positive psychology and other evidence-based practices. **START** stands for **S**ystemic, **T**herapeutic, **A**ssessment, **R**esources, and **T**reatment.*

# Final project surveys

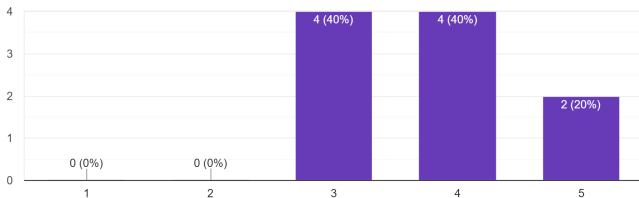
- ▶ Thanks to the 8 of you that have filled it out!
- ▶ **Extension for final project survey: due Friday, 09/30, 11:59 PM (then we'll put you in groups)**
- ▶ If you've formed a team already, feel free to get started on final project milestones (first one *not due until 10/21*)

# Pset 1 feedback so far

How would you rate the difficulty of problem set 1?

 Copy

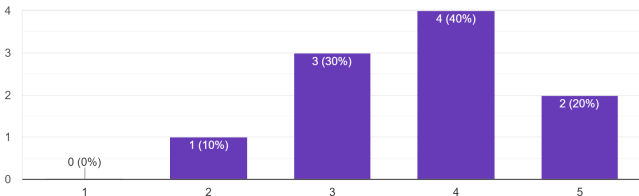
10 responses



How would you rate the overall pace of the course?

 Copy

10 responses



## Note on difficulty of activities/psets vs. DataCamp

1. **DataCamp**: meant as gentle intro before pset challenges; not realistic for entry-level data science jobs; provides a lot of handholding in terms of noting (1) exactly which commands to use; (2) helper code; (3) very simplified/cleaned data
2. **Real-world data science**: more difficult than the problem set; would be asked “hey, did this policy reduce or widen disparities” and start with a blank notebook and be 100% reliant on google/stackoverflow
  - ▶ **Translating question into concrete approach**: define disparities (charges, incarceration or not, sentencing conditional on incarceration); find which variables measure that; deal with duplicates
  - ▶ **Data cleaning without scaffolding**: recognizing the errors in the datetime and that `errors_coerce = True` would set a lot of valuable data to missing; further deduplication of judge names; investigating PROMIS CONVERSION (eg coding that to missing)
  - ▶ A lot of these things won't throw errors if you run an analysis without fixing but will lead to flawed results/incorrect policy conclusions

Before we code, let's group!

**Groups for today are same as pset1:**

<b>Partner A</b>	<b>Partner B</b>	<b>Partner C</b>
Max Konzerowsky	Omar Corral-Williams	
Anish Sikhnam	Justin Sapun	
Emma Johnson	Nate Haile	
Andrew Cho	Luca D'Ambrosio	
Saige Gitlin	Daniel Céspedes	Giulio Frey
Nick Romans	Kayla Hamann	
Daniel Xu	Filippo de Min	
Andy Ilie	Rachael Williams	



# Goals for today

- ▶ Project & assignment updates
- ▶ **Git/GitHub: lecture & activity, review pset2 submission instructions (from previous class)**
- ▶ User-defined functions: lecture & activity (from last week)
- ▶ LaTeX/Overleaf

# Goals for today

- ▶ Project & assignment updates
- ▶ Git/GitHub: **lecture & activity**, review pset2 submission instructions (from previous class)
- ▶ **User-defined functions: lecture & activity** (from last week)
- ▶ LaTeX/Overleaf

# Goals for today

- ▶ Project & assignment updates
- ▶ Git/GitHub: [lecture](#) & [activity](#), review pset2 submission instructions (from previous class)
- ▶ User-defined functions: [lecture](#) & [activity](#) (from last week)
- ▶ **LaTeX/Overleaf**

# Overview before activity

- ▶ LaTeX: typesetting language
- ▶ Can work with locally using things like TexMaker, etc.
- ▶ Here, we'll be interacting with it via Overleaf, which is similar to Google docs but for LaTeX and facilitates collaboration/easier troubleshooting of compile errors

# Non-exhaustive list of things that can cause compilation errors

1. Underscores or certain special characteristics without an “escape” before them, e.g.:

```
## Ex. 1: this causes error due to underscore without escape
```

```
The file is called: file_here.R
```

```
## works
```

```
The file is called: file\_here.R
```

```
## Ex. 2: comments out rest of code after percent symbol
```

```
This increased by 5%
```

```
## this works
```

```
This increased by 5\%
```

2. Start entering math mode but fail to exit it, e.g.:

```
## Ex. 3: this causes errors
```

```
We calculate fraction as  $\frac{5}{10}$  and then do...
```

```
## this works
```

```
We calculate fraction as  $\frac{5}{10}$  and then do
```

## “Environments”, or ways to go beyond standard text

- ▶ Itemized list

```
\begin{itemize}  
\item First item...  
\item  
\end{itemize}
```

- ▶ Numbered list

```
\begin{enumerate}  
\item First item...  
\item  
\end{enumerate}
```

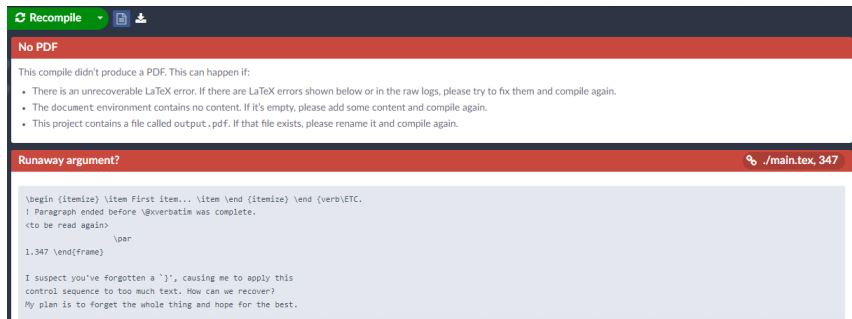
- ▶ Figure

```
\begin{figure}  
\caption{my caption}  
\label{fig:myfig}  
\includegraphics[scale = 0.5]{example_graphic.png}  
\end{figure}
```

# Leads to another set of compilation errors

- ▶ Runaway argument or forgotten end group
- ▶ Usually means you began an environment but forgot to end it; can happen with long tables, deeply nested lists, etc. where easy to lose track

Example:



The screenshot shows the Overleaf interface with a compilation error. At the top, a green bar says "Recompile". Below it, a red banner reads "No PDF". The message states: "This compile didn't produce a PDF. This can happen if:" followed by three bullet points: "There is an unrecoverable LaTeX error. If there are LaTeX errors shown below or in the raw logs, please try to fix them and compile again.", "The document environment contains no content. If it's empty, please add some content and compile again.", and "This project contains a file called output .pdf. If that file exists, please rename it and compile again." Below this, another red banner says "Runaway argument?" with a link icon and the file name ".main.tex, 347". The error message in the code editor reads: "\begin{itemize} \item First item... \item \end{itemize} \end{verb}ETC. ! Paragraph ended before \@xverbatim was complete. <to be read again> \par 1.347 \end{frame} I suspect you've forgotten a '}', causing me to apply this control sequence to too much text. How can we recover? My plan is to forget the whole thing and hope for the best."

Recompile

No PDF

This compile didn't produce a PDF. This can happen if:

- There is an unrecoverable LaTeX error. If there are LaTeX errors shown below or in the raw logs, please try to fix them and compile again.
- The document environment contains no content. If it's empty, please add some content and compile again.
- This project contains a file called output .pdf. If that file exists, please rename it and compile again.

Runaway argument? [./main.tex, 347](#)

```
\begin{itemize} \item First item... \item \end{itemize} \end{verb}ETC.
! Paragraph ended before \@xverbatim was complete.
<to be read again>
\par
1.347 \end{frame}

I suspect you've forgotten a '}', causing me to apply this
control sequence to too much text. How can we recover?
My plan is to forget the whole thing and hope for the best.
```

# Compilation errors

- ▶ Common w/ complicated docs
- ▶ Ways to address:
  1. **Recompile frequently!**
  2. Try to interpret and google the error—not always easy since error messages may not be clear/informative w.r.t. line numbers (esp. on Overleaf)



## Other useful commands

```
## create a numbered section and label it to cross-ref
\section{This is my section outlining disparities}
\label{sec:disparities}
```

```
## reference a section in text
In Section \ref{sec:disparities} I discuss...
```

```
## reference a table or fig in text
Table \ref{tab:tabname} and Figure \ref{fig:myfig} show...
```

```
## stop a figure or table from going into the next section
[! h] (inside \figure{} env; stay where it is in code)
\FloatBarrier (before \& after figure/table; don't float off)
(in addition to stuff at the start of the \begin{table})
```

## Break for LaTeX tables and figures activity

- ▶ [Link to template to copy over](#) (click 'Menu' in top-left then Actions/'Copy Project')
- ▶ Link to Python activity:

[03\\_latex\\_output\\_examples\\_blank.ipynb](#)