Creating and maintaining teaching materials using R(Markdown)

Abhijit Dasgupta

Zansors

November 8, 2019

Who am I?

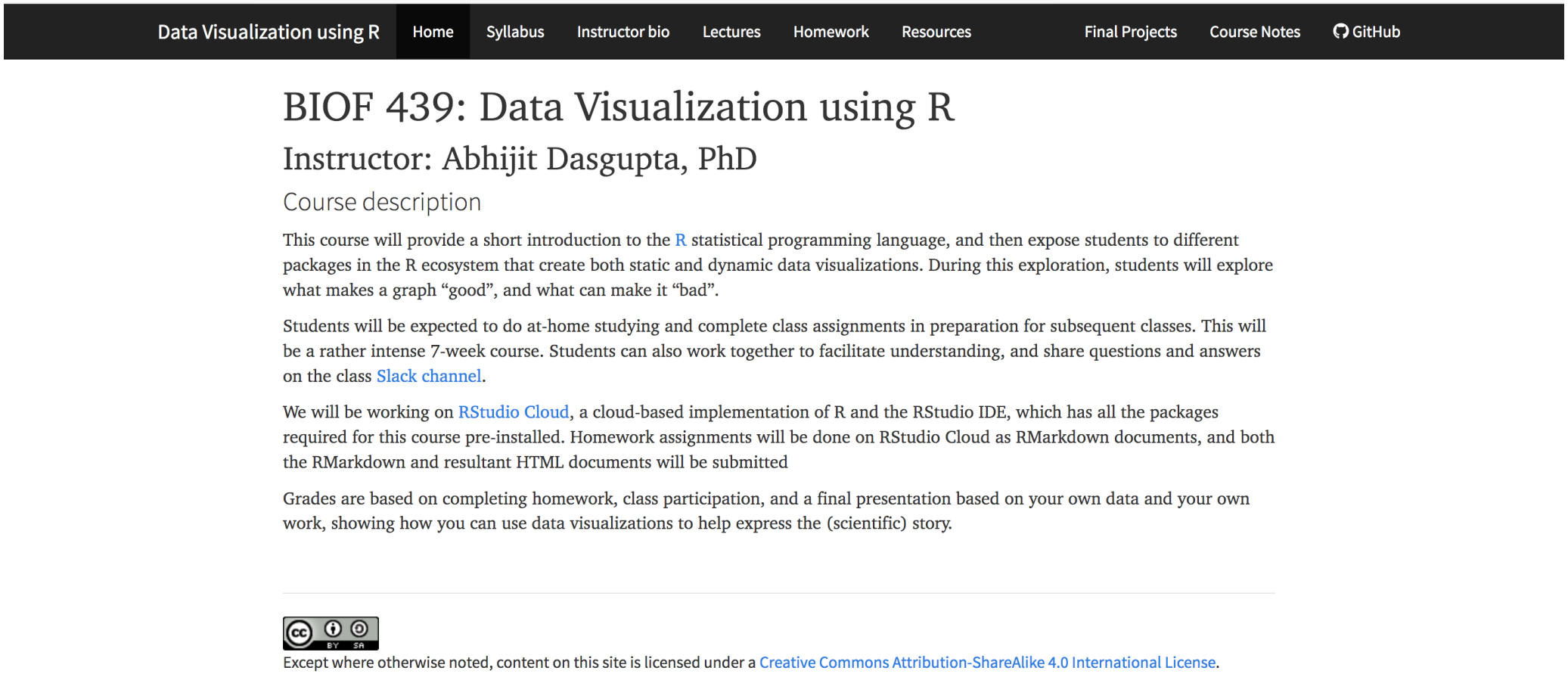
- Co-founded Stat Programming DC with Marck
- Sat on the Board of DCDC
- Biostatistican by trade (NIH)
    - Research, collaboration and reviews
- Data science at a startup (Zansors)
    - stats, signal processing, business dev
- Teaching R (FAES Graduate School at NIH, State Dept, elsewhere)

Who am I?

- Co-founded Stat Programming DC with Marck
- Sat on the Board of DCDC
- Biostatistican by trade (NIH)
    - Research, collaboration and reviews
- Data science at a startup (Zansors)
    - stats, signal processing, business dev
- Teaching R (FAES Graduate School at NIH, State Dept, elsewhere)

Teaching at FAES

Teaching at FAES



Data Visualization using R | Home | Syllabus | Instructor bio | Lectures | Homework | Resources | Final Projects | Course Notes | GitHub

BIOF 439: Data Visualization using R

Instructor: Abhijit Dasgupta, PhD

Course description

This course will provide a short introduction to the R statistical programming language, and then expose students to different packages in the R ecosystem that create both static and dynamic data visualizations. During this exploration, students will explore what makes a graph "good", and what can make it "bad".

Students will be expected to do at-home studying and complete class assignments in preparation for subsequent classes. This will be a rather intense 7-week course. Students can also work together to facilitate understanding, and share questions and answers on the class Slack channel.

We will be working on RStudio Cloud, a cloud-based implementation of R and the RStudio IDE, which has all the packages required for this course pre-installed. Homework assignments will be done on RStudio Cloud as RMarkdown documents, and both the RMarkdown and resultant HTML documents will be submitted

Grades are based on completing homework, class participation, and a final presentation based on your own data and your own work, showing how you can use data visualizations to help express the (scientific) story.

Teaching workshops

Programming with R | Home | Syllabus | Instructor bio | Lectures | Resources | Course Notes

PS 312: Programming with R

Abhijit Dasgupta, PhD

Welcome to PS 312.

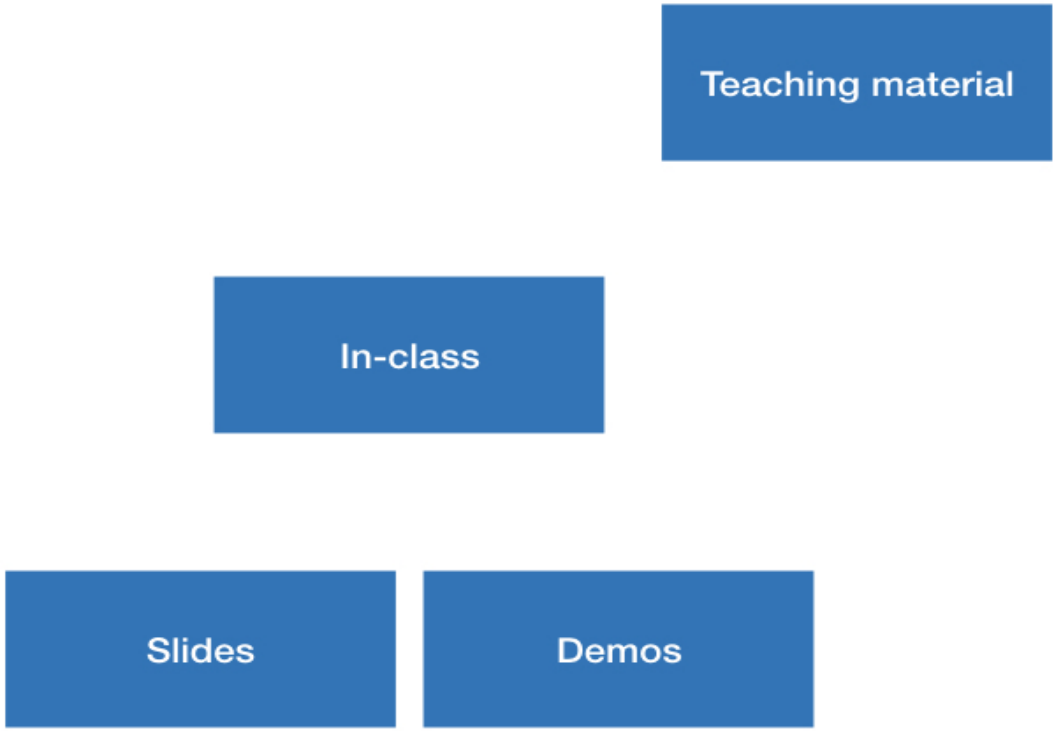
This 3-day workshop will introduce you to R, a statistical programming language that is widely used in government, industry and academia. R has grown in popularity both due to its price (free) and because it is a very high quality product. The R community covers the globe, and has contributed almost 2000 modules that cover the depth and breadth of analytics and data science. We will introduce a selection of useful modules in this class.

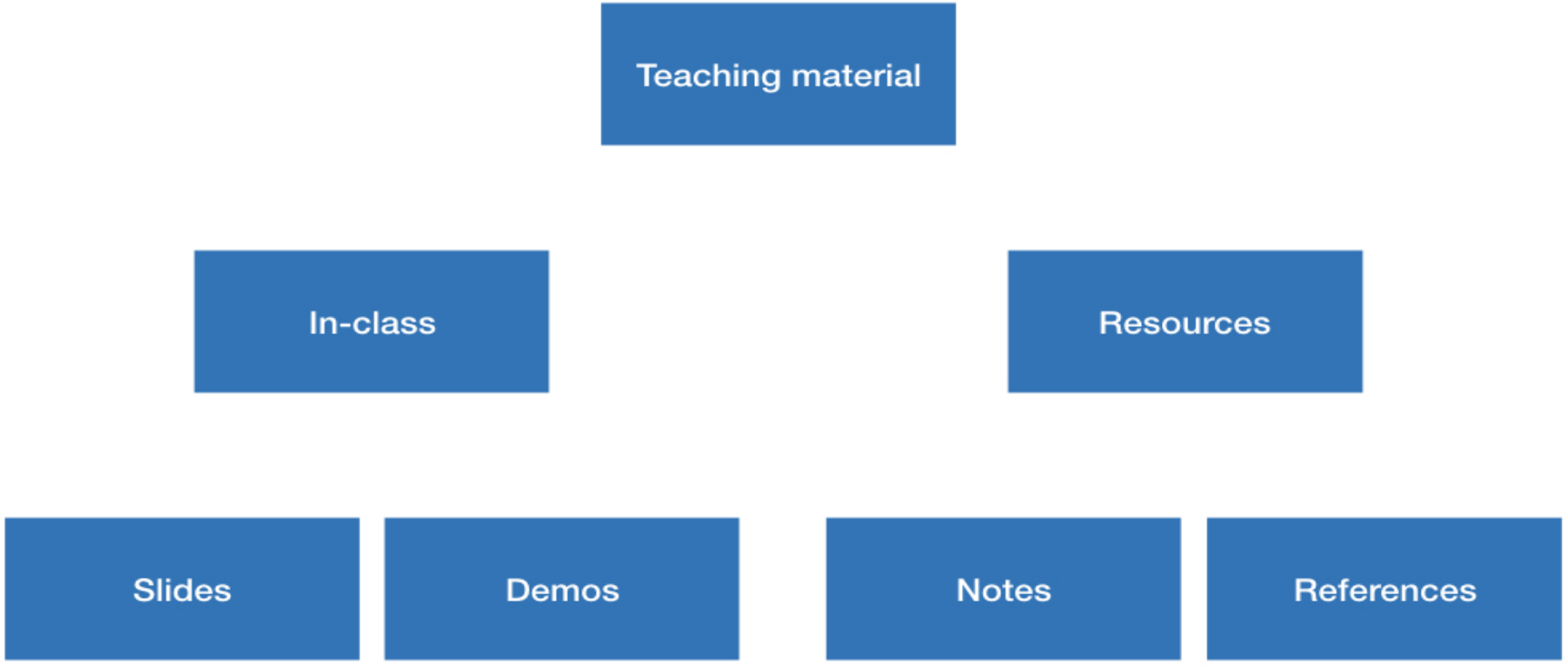
One of the great strengths of R has been and continues to be its data visualization capabilities; the ability to produce high quality, visually pleasing graphs easily. The ability to produce complex data visualizations and even interactive visualizations with few lines of code greatly reduces the effort needed to produce a high-quality product.

As a scripting language, R allows you to program once and re-use for different data sets in a completely transparent and reliable manner. It also allows you to create recipes and workflows that can automatically validate your data, process and analyze it and create automated reports. We will show how R can interact with Excel so that current workflows can be enhanced and analyses shared in a familiar manner.

We will of course merely touch the tip of this iceberg in this class, but there are plenty of online resources, mostly free, that will allow you to learn more and use R to meet your needs.

Teaching material


Teaching material
In-class
Slides
Demos


Teaching material
In-class
Resources
Slides
Demos
Notes
References

Course requirements

Course requirements

Syllabus

Homework

Grading

Concept

Create a RMarkdown website

Slides with `xaringan`

Homework with RMarkdown

Notes with bookdown

But....

Need to produce new material and update

Update website and not rebuild everytime (make??)

Some kind of table of contents for lectures, homeworks

Automate processes (syllabus, receiving homework, grading, recording grades)

Syllabus

```
startDate <- lubridate::ymd('20190911')
classDate <- function(wk, startDate ){
  today <- startDate + 7*(wk-1)
  return(format(today, "%B %d, %Y"))
}

dts <- classDate(1:14, startDate)
topics <- c(
  'Introduction to R, RStudio and RMarkdown',
  'Data Structures in R (classes 5:30-7, 7-8:30)',
  'R packages, data import/export, munging',
  'Towards analytic data: Data Munging, continued',
  'Data exploration through visualization',
  'More data visualization and RMarkdown',
  'Statistical analyses: Table 1, estimation and confidence intervals, and more ggplot',
  'Statistical analyses: Classical hypothesis testing and computational inference',
  'Statistical learning: Regression models',
  "More data munging with `purrr`: grouping, mapping and functional programming",
  "Basic bioinformatics: Bioconductor and friends",
  "No class (Thanksgiving)",
  'Statistical learning: Cluster analysis and pattern recognition',
  "Project presentations")
D <- tibble(Date = dts, Topic=topics)
knitr::kable(D)
```

Automatically populate dates

Syllabus

Practical R | Home | Syllabus | Instructor bio | Lectures | Homework | Resources | Course Notes | Slack | GitHub

Outline of the class

Date	Topic
September 11, 2019	Introduction to R, RStudio and RMarkdown
September 18, 2019	Data Structures in R (classes 5:30-7, 7-8:30)
September 25, 2019	R packages, data import/export, munging
October 02, 2019	Towards analytic data: Data Munging, continued
October 09, 2019	Data exploration through visualization
October 16, 2019	More data visualization and RMarkdown
October 23, 2019	Statistical analyses: Table 1, estimation and confidence intervals, and more ggplot
October 30, 2019	Statistical analyses: Classical hypothesis testing and computational inference
November 06, 2019	Statistical learning: Regression models
November 13, 2019	More data munging with `purrr`: grouping, mapping and functional programming
November 20, 2019	Basic bioinformatics: Bioconductor and friends
November 27, 2019	No class (Thanksgiving)
December 04, 2019	Statistical learning: Cluster analysis and pattern recognition
December 11, 2019	Project presentations

Lectures

Table of contents

```
parse_title <- function(f){
  if(is.na(f)) return(NA)
  rmarkdown::yaml_front_matter(f)$title
}

rmds <- dir_ls('lectures', regex = '\\d{2}.*.Rmd')
htmls <- str_replace(rmds, 'Rmd','html')
pdfs <- str_replace(rmds, 'Rmd','pdf')

blah <- Reduce(full_join, list(htmls, pdfs,rmds)) %>%
  select(base, everything())
blah <- blah %>% tidyr::gather(variable, value, -base) %>%
  mutate(value = glue('[{variable}]({value})')) %>%
  mutate(value = ifelse(str_detect(value, 'NA'), '', value)) %>%
  spread(variable, value) %>%
  rename(Lecture=base) %>%
  mutate(Title = map_chr(str_match(Rmd, '\\((.+)\\)')[,2],parse_title))

kable(blah) %>%
  kable_styling(bootstrap_options='striped', full_width = T)
```

Lectures

Table of contents

```
parse_title <- function(f){
  if(is.na(f)) return(NA)
  rmarkdown::yaml_front_matter(f)$title
}

rmds <- dir_ls('lectures', regex = '\\d{2}.*.Rmd')
htmls <- str_replace(rmds, 'Rmd','html')
pdfs <- str_replace(rmds, 'Rmd','pdf')

blah <- Reduce(full_join, list(htmls, pdfs,rmds)) %>%
  select(base, everything())
blah <- blah %>% tidyr::gather(variable, value, -base) %>%
  mutate(value = glue('[{variable}]({value})')) %>%
  mutate(value = ifelse(str_detect(value, 'NA'), '', value)) %>%
  spread(variable, value) %>%
  rename(Lecture=base) %>%
  mutate(Title = map_chr(str_match(Rmd, '\\((.+)\\)')[,2],parse_title))

kable(blah) %>%
  kable_styling(bootstrap_options='striped', full_width = T)
```

Automatically grab slide titles from the slides

Lectures

Table of contents

```
parse_title <- function(f){
  if(is.na(f)) return(NA)
  rmarkdown::yaml_front_matter(f)$title
}

rmds <- dir_ls('lectures', regex = '\\d{2}.*.Rmd')
htmls <- str_replace(rmds, 'Rmd','html')
pdfs <- str_replace(rmds, 'Rmd','pdf')

blah <- Reduce(full_join, list(htmls, pdfs,rmds)) %>%
  select(base, everything())
blah <- blah %>% tidyr::gather(variable, value, -base) %>%
  mutate(value = glue('[{variable}]({value})')) %>%
  mutate(value = ifelse(str_detect(value, 'NA'), '', value)) %>%
  spread(variable, value) %>%
  rename(Lecture=base) %>%
  mutate(Title = map_chr(str_match(Rmd, '\\((.+)\\)')[,2],parse_title))

kable(blah) %>%
  kable_styling(bootstrap_options='striped', full_width = T)
```

I use `glue` a lot for automating link creation

Lectures

Homeworks

```
solutions <- dir_ls('HW', glob = '*Solution.html', fail=F)
hws <- dir_ls('HW', regex="HW\\d+.html", fail=F)
hws <- setdiff(hws, solutions)
rmds <- dir_ls('HW', regexp = 'HW\\d+.Rmd')
numbers <- sort(unique(stringr::str_match(hws, 'HW(\\d+)')[,2]))
ind <- match(glue('HW/HW{i}_Solution.html', i=numbers), solutions)
out_data <- tibble(n = numbers, rmds = sort(rmds), hws = sort(hws), solutions = solutions[ind])

out <- tibble('Homework' = glue_data(out_data,'[Homework {n}]({hws})'),
              'R Markdown' = glue_data(out_data,
                                       '[R Markdown]({rmds})'),
              'Solutions' = ifelse(!is.na(out_data$solutions),
                                   glue_data(out_data,'[Solution]({solutions})'),
                                   ''))
```

Automatically create homework and solution links

Provide RMD files as template for submission

Homework

Homeworks

Homework	R Markdown	Solutions
Homework 1	R Markdown	
Homework 2	R Markdown	Solution
Homework 3	R Markdown	Solution
Homework 4	R Markdown	Solution
Homework 5	R Markdown	Solution
Homework 6	R Markdown	Solution
Homework 7	R Markdown	
Homework 8	R Markdown	

Homework

Homework 8: Summaries

Abhijit Dasgupta

Due Tuesday, November 5 at midnight

Submission Link

```
library(reticulate)
source_python('create_request.py')
a <- py$create_request('2019-11-07T00:00:00',
                       'Homework 8',
                       '/BIOF339_Fall2019/Homeworks/H
```

```
def create_request(timestamp, name, directory, late=T
  """
  Create a file request in Dropbox, with a deadline
  (only for Business or Professional accounts)

  INPUTS:

  timestamp: The deadline date and time in ISO 8601 f
             in local time
  name:      The title of the file request
  directory: The destination directory for the file r
             It must start with '/' for the top level
  late :     True/False; if True a grace period of 1
  token:     Dropbox authentication token
  """
```

Homework evaluation

- Homework submission as RMarkdown documents via Dropbox
  - Dropbox attaches student names to files
- I run the document, provide comments in RMarkdown, and put grade on last line
  - Emphasize standard project structure, `here`
- Script to parse grades from submissions, with names, and update a CSV file
  - Can see who has submitted which homework, and scores
- Email commented homework back
  - `gmailr` & class email list
- Have not played with `learnr` and `grader`

Maintenance and upkeep

Use `drake` (one choice out of many)

```
slides_plan <- drake_plan(
  create_slides_html = target(
    rmarkdown::render(knitr_in(rmdf), output_dir=slid
    transform = map(rmdf=!!slides_rmdfiles)
  ),
  create_slides_pdf = target(
    coursedown::slide2pdf(knitr_in(rmdf)),
    transform = map(rmdf = !!slides_rmdfiles)
  ),
  create_index = target(
    rmarkdown::render_site(knitr_in('slides/index.Rmd
    trigger = trigger(depend =T, file=T, command=T,
                      change = max(map_dbl(!!slides_r
                                           function(x) fi
  )
)
```

```
hw_plan <- drake_plan(
  create_hw_html = target(
    rmarkdown::render(knitr_in(rmdf), output_dir = hw
    transform = map(rmdf = !!hw_rmdfiles)
  ),
  create_index = target(
    rmarkdown::render_site(knitr_in('slides/index.Rmd
    trigger = trigger(depend=T, file=T, command=T,
                      change = max(map_dbl(!!hw_rmdfi
                                           function(x
  )
)
```

Maintenance and upkeep

Use `drake` (one choice out of many)

```
slides_plan <- drake_plan(
  create_slides_html = target(
    rmarkdown::render(knitr_in(rmdf), output_dir=slid
    transform = map(rmdf=!!slides_rmdfiles)
  ),
  create_slides_pdf = target(
    coursedown::slide2pdf(knitr_in(rmdf)),
    transform = map(rmdf = !!slides_rmdfiles)
  ),
  create_index = target(
    rmarkdown::render_site(knitr_in('slides/index.Rmd
    trigger = trigger(depend =T, file=T, command=T,
                      change = max(map_dbl(!!slides_r
                                           function(x) fi
  )
)
```

```
hw_plan <- drake_plan(
  create_hw_html = target(
    rmarkdown::render(knitr_in(rmdf), output_dir = hw
    transform = map(rmdf = !!hw_rmdfiles)
  ),
  create_index = target(
    rmarkdown::render_site(knitr_in('slides/index.Rmd
    trigger = trigger(depend=T, file=T, command=T,
                      change = max(map_dbl(!!hw_rmdfi
                                           function(x
  )
)
```

Uses `pagedown::chrome_print`.

My students seem to demand PDFs

Maintenance and upkeep

Use `drake` (one choice out of many)

```
slides_plan <- drake_plan(
  create_slides_html = target(
    rmarkdown::render(knitr_in(rmdf), output_dir=slid
    transform = map(rmdf=!!slides_rmdfiles)
  ),
  create_slides_pdf = target(
    coursedown::slide2pdf(knitr_in(rmdf)),
    transform = map(rmdf = !!slides_rmdfiles)
  ),
  create_index = target(
    rmarkdown::render_site(knitr_in('slides/index.Rmd
    trigger = trigger(depend =T, file=T, command=T,
                      change = max(map_dbl(!!slides_r
                                           function(x) fi
  )
)
```

```
hw_plan <- drake_plan(
  create_hw_html = target(
    rmarkdown::render(knitr_in(rmdf), output_dir = hw
    transform = map(rmdf = !!hw_rmdfiles)
  ),
  create_index = target(
    rmarkdown::render_site(knitr_in('slides/index.Rmd
    trigger = trigger(depend=T, file=T, command=T,
                      change = max(map_dbl(!!hw_rmdfi
                                           function(x
  )
)
```

Automatically updates the TOC

The coursedown package (at github.com/webbedfeet/coursedown)

- Structure project as a website with 3 sub-websites
  - Lectures, Homework, Notes
- Common top level links
- Assumes deployment via GitHub Pages
- Creates directory structure
- Provides templates for slides, homeworks, notes, top-level `index.Rmd` for each website

Very much a work in progress

`_course.yml`

```
course_shortname: BIOF339
course_name: Practical R
instructor: Abhijit Dasgupta
description: ''
course_url: http://www.araastat.com/BIOF339
dates: Fall, 2019
slack: 'biof339'
short_repo: araastat/BIOF339
twitter: webbedfeet
course_shortname1: BIOF339
github: https://www.github.com/araastat/BIOF339
slack_channel: https://biof339.slack.com
```

- Common branding and labeling via yaml

Wrapping up

- Many opinionated choices, based on what I wanted to learn
- Would love collaboration to fix the kinks

Thank you

webbedfeet.netlify.com

@webbedfeet

webbedfeet/coursedown