Welcome!

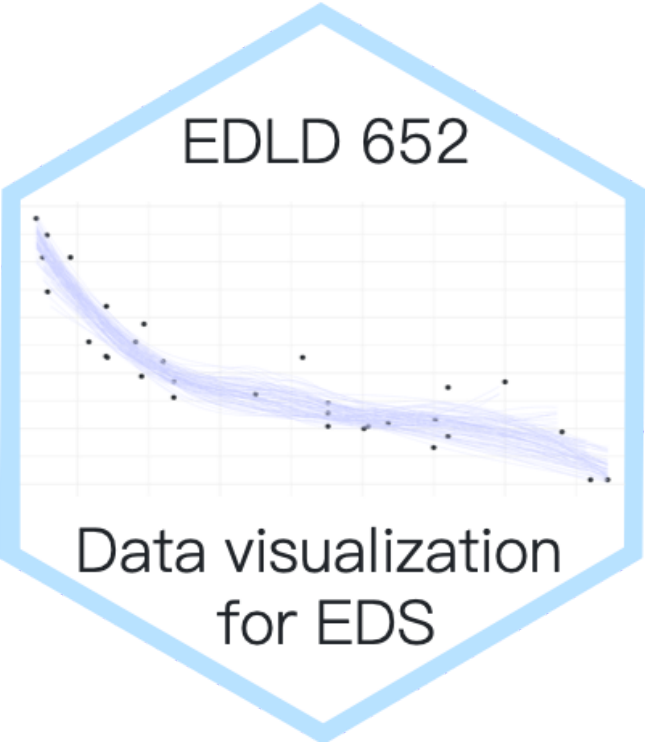
An overview of the course

Daniel Anderson

Week 1

Agenda

- Getting on the same page
- Syllabus
- Accessing and working with the course data
- If time allows – Intro to text data



whoami

- Research Associate Professor: Behavioral Research and Teaching
- Dad (two daughters: 9 and 7)
- Pronouns: he/him/his
- Primary areas of interest: 💗💗R💗💗, computational research, systemic inequities in opportunities and achievement, and variance between educational institutions

whoisyou?

- Introduce yourself
- Why are you here?
- What pronouns would you like us to use for you for this class?
- What was one thing you did not related to academic work over winter break?

A few class policies

- Be kind
- Be understanding and have patience, with others **and yourself**
- Help others whenever possible

Truly the most important part of this class. Important not just in terms of decency, but also in your learning, and most importantly, for equity.

A more specific policy

Kiddos in class

- All breastfeeding babies are welcome in class as often as necessary.
- Non–nursing babies and older children are welcome whenever alternate arrangements cannot be made. As a parent of two young children, I understand that babysitters fall through, partners have conflicting schedules, children get sick, and other issues arise that leave parents with few other options.

- In cases where children come to class, I invite parents/caregivers to sit close to the door so as to more easily excuse yourself to attend to your child's needs. Non–parents in the class: please reserve seats near the door for your parenting classmates.

- All students are expected to join with me in creating a welcoming environment that is respectful of your classmates who bring children to class.

Omicron

In–person class

- This class is scheduled to be in–person
- I am vaccinated and boosted (> 2 weeks ago)
- I plan to always double mask
- If you are not feeling well at all, even if you don't think it's COVID, please do not attend in person
- All courses will simultaneously be on zoom, and recordings will be posted

Last intro thing

- I'm here for you
- We won't have specific office hours, but know I'm always willing to meet
- This course, like all in the sequence, can be difficult. Don't suffer in silence. Don't do this alone.

Syllabus

Course Website(s)

website repo

Home Schedule Assignments Syllabus Ta

Data Visualization

for educational data science

Welcome to the second course in the Educational Data Science Specialization taught at the
the College of Education. This course will be taught through R, a free and open-source stat
and will provide students with the foundational principles and practice of data visualization,
scientific and technical data. We will have weekly lectures, covering a wide variety of topics
color theory, and principles of visual design. We will also cover mediums for communicatior
an emphasis on different web applications. Weekly hands-on laboratory sessions provide s
the lecture material into practice.

Materials

- Nearly everything will be distributed through the repo and through the website.
- Please clone the repo now, if you haven't already. Pull each week for the most recent changes.
- We'll use Canvas for grading, and that is essentially it.

R Markdown notes

- These slides were produced with **{xaringan}**, an R Markdown variant. I encourage you to try it out and use it for your final project presentation.
- The website was also produced with R Markdown (sort of)
    - It's a **{blogdown}** website with some custom CSS and Hugo shortcodes
- This course is not just about data viz, but also mediums for communication. This includes websites and data dashboards among other possibilities.

My assumptions about you

I assume you

- Understand the R package ecosystem (how to find, install, load, and learn about them)
- Can read "flat" (i.e., rectangular) datasets into R
    - I don't care what you use, but you should be using RStudio Projects & the {here} package
    - See Jenny Bryan's blog post for why.

- Can perform basic data wrangling and transformations in R, using the tidyverse
    - Leverage appropriate functions for introductory data science tasks (pipeline)
    - "clean up" the dataset using scripts and reproducible workflows
- Use version control with R via git and GitHub
- Use R Markdown to create reproducible dynamic reports

Learning objectives

- Transform data in a variety of ways to create effective data visualizations
- Understand and and be able to apply basic string operations and work with textual data
- Understand best practices in data visualization
- Customize ggplot2 graphics by reordering factors, creating themes, etc.
- Create an online data visualization portfolio using distill and/or flexdashboards to demonstrate key learning

Examples

Below are some links to final projects from students who have taken this class previously.

Dashboards

- Alexis Adams–Clark
- Brendan Cullen
- Maggie Osa

Blog post

- Teresa Chen
- Ouafaa Hmaddi
- Murat Kezer

Weekly learning objectives

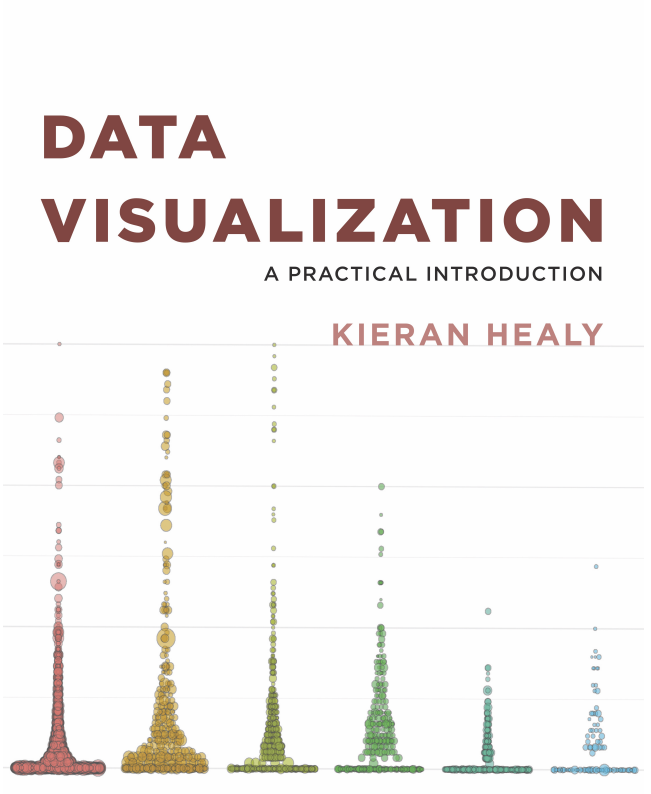
Provide you a frame for what you should be working to learn for that specific week.

This week's objectives

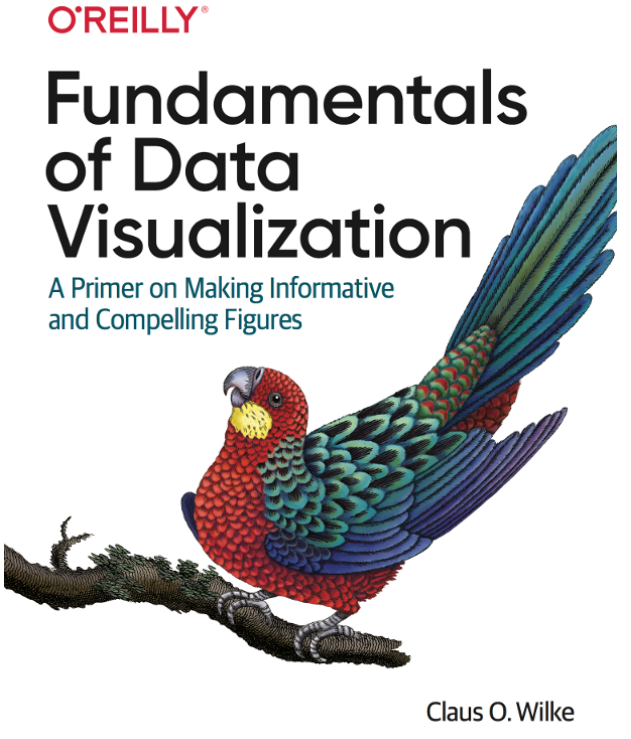
- Understand the requirements of the course
- Understand the requirements of the final project
- Be ready to go with *git* and GitHub
- Understand how to access the course data and documentation, begin playing with the data

Required Textbooks (free)

Healy

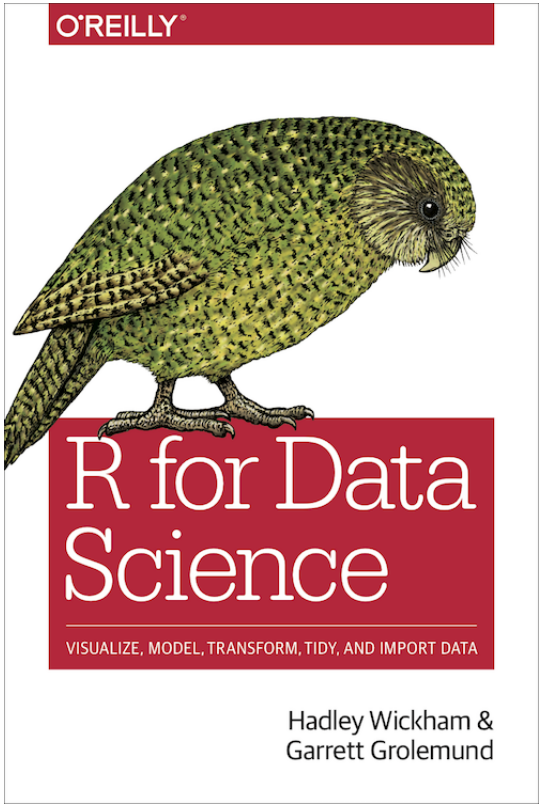


Wilke



Other books (also free)

Bryan



Wickham & Grolemund

Another resource

See the current draft here. Please read Chapter 8 on collaborating with git/GitHub. There is also a video lecture on this topic from last year that is linked on the website.

Extra credit opportunity

5 points: Deep dive into a topic not covered by the course

Some options

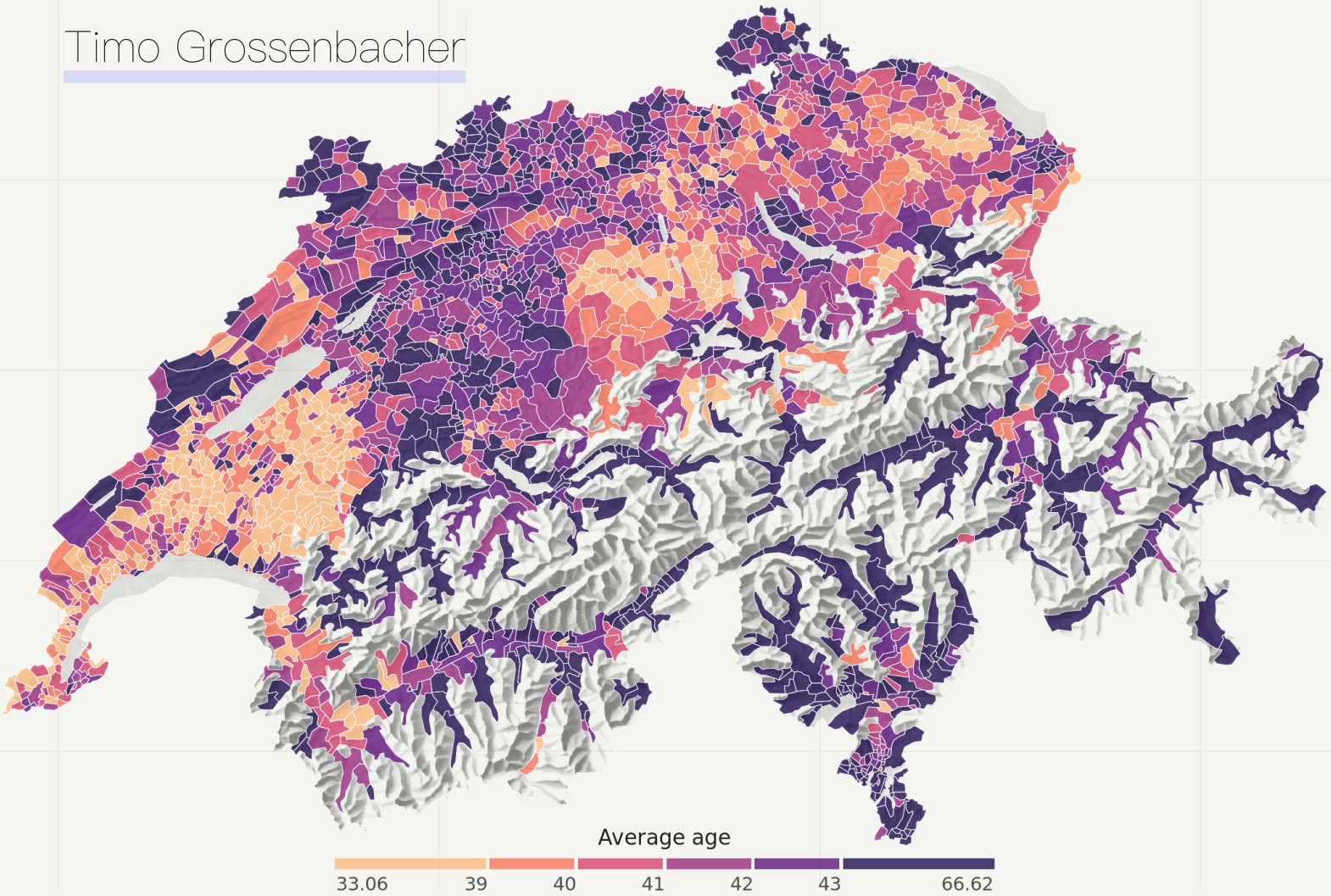
- Geographic data (we will discuss this, but it's relatively late and there's a ton we won't be able to get to)
- Network data
- DAGs
- Flow data (e.g., alluvial diagrams)
- Interactive plots
- Animated plots

Some examples

Switzerland's regional demographics

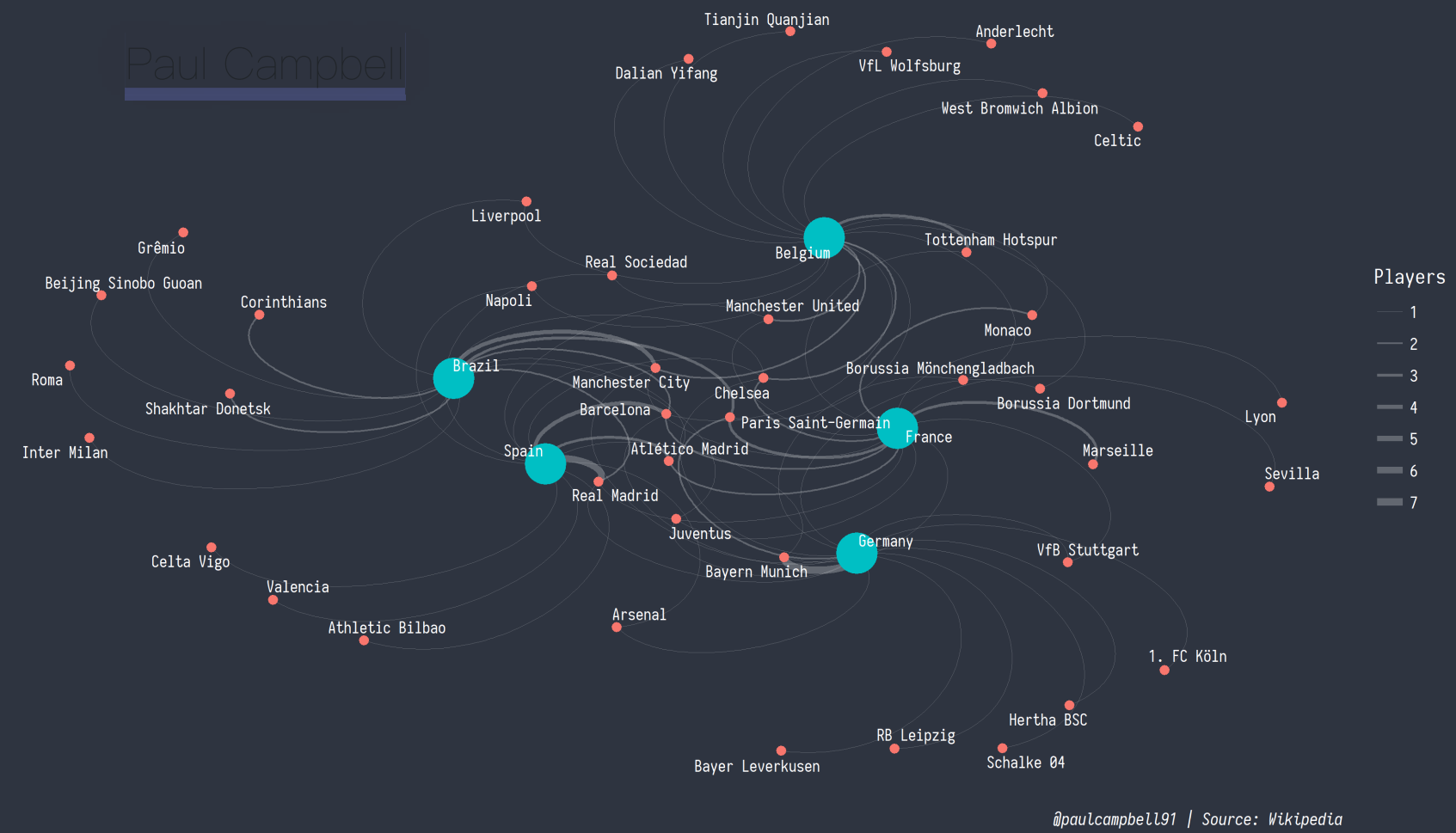
Average age in Swiss municipalities, 2015

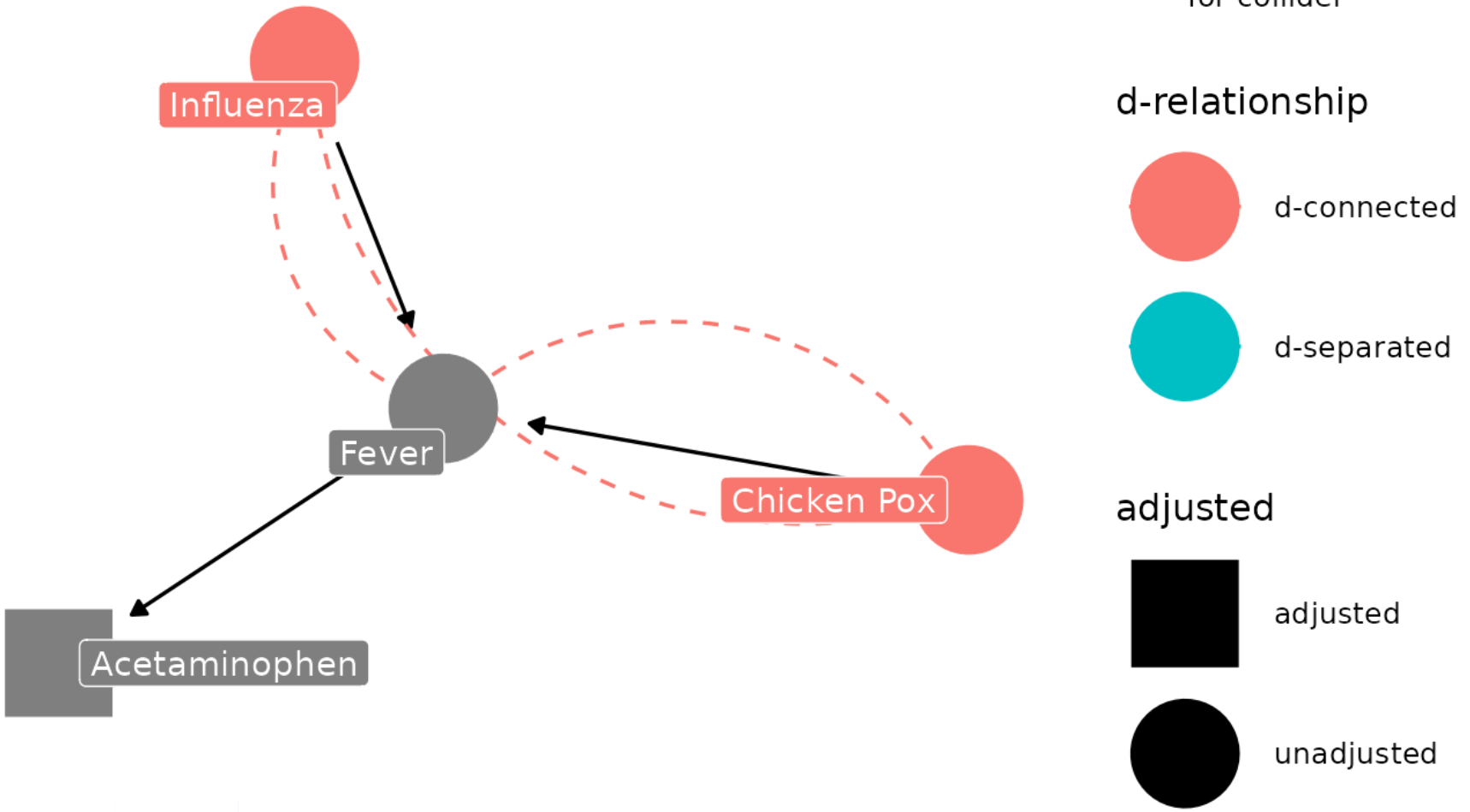
Timo Grossenbacher



World Cup 2018 | Club Country Network

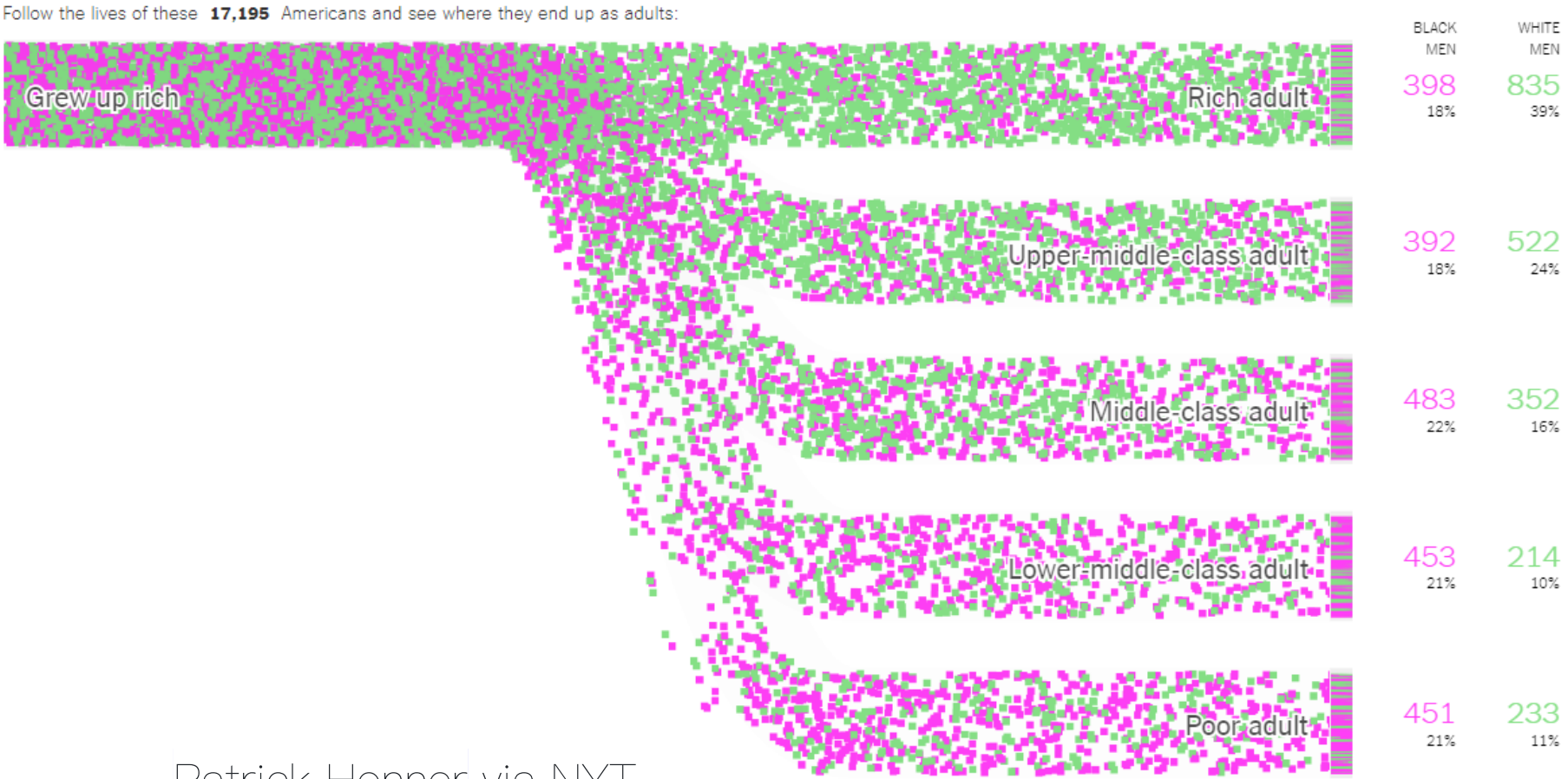
Belgium, Brazil, France, Germany, Spain



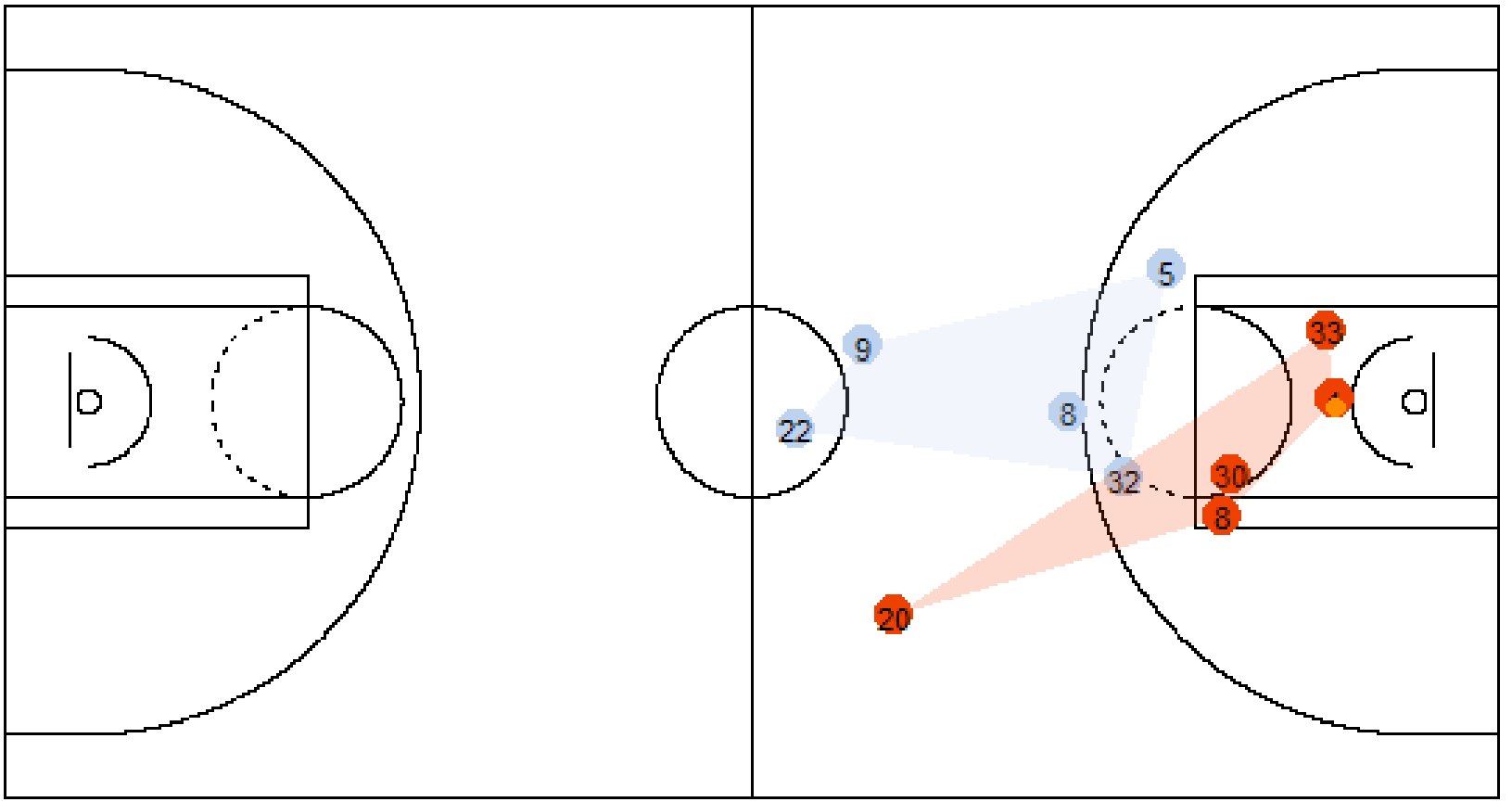


ggdag via Malcolm Barrett

Black and white boys raised in wealthy families



Patrick Honner via NYT



James Curley

Labs

See the assignments page of the website.

15 points each (45 points total; 30%)

1. Distributions, GitHub collabo, and working w/strings
2. Visual perception & plot reproducing
3. Color

Homework

Only one this time, worth 30 points (20%)

- Basically the same as the labs, but scored correct/incorrect, and no in–class time devoted to them.
- Okay to work on collaboratively – I actively encourage you to do so as long as you're using a shared repo

Topic: Visualizing uncertainty, tables, and plot refinement

Data viz "in the wild" presentations

Everyone will be randomly assigned a date to share two data visualizations you have found in publications, websites, or anywhere else IRL.

- Not a formal presentation
- Share the links with me before class – we'll look at it as a group and discuss
- You note where you found it and what you like/dislike about it

Presentation order

Date	Presenter
2022–01–10	Futing
2022–01–10	Zach
2022–01–10	Cano
2022–01–17	Ian
2022–01–17	Abbie
2022–01–24	Havisha
2022–01–24	Tingyu
2022–01–31	Dillon
2022–01–31	Eliott
2022–02–07	Merly
2022–02–07	Esmeralda

Date	Presenter
2022–02–14	Amy
2022–02–14	Diana
2022–02–21	Errol
2022–02–21	Mandi
2022–02–28	Adriana
2022–02–28	Rebecca

I will email this out as well.

Final Project

70 points total (46.66%)

Group project

- Please try to finalize your group by the end of today. You will have time when exploring the course data to work together.
- No fewer than 2, no more than 3.
- Although the final is the only mandated group project, I encourage you to work with your group for all labs and the homework assignment as well.

Five parts

- Proposal (5 points): Due 1/24/22
- Draft (10 points): Due 2/21/22
- Peer review (10 points): Assigned, 2/21/21; Due 2/28/22
- Presentation (5 points): 3/7/21 (Week 10)
- Product (40 points): Due 11:59:59 PM, 3/14/21

Product

Four components:

- A web–deployed portfolio showcasing your #dataviz skills.
  - distill (what I'll lecture on), R Markdown, or blogdown website
  - Technical document with pagedown or bookdown
  - Scientific poster with pagedown
  - flexdashboard

- At least three finalized data displays, with each accompanied by a strong narrative/story, as well as the history of how the visualization changed over time.
- Housed on GitHub
    - Fully reproducible
- Deployed through GitHub pages (or netlify or similar)

Proposal

Four components:

- Show me some evidence that you've at least played around with the course data and that you have some ideas of what you want to do
- *Very* preliminary visualizations, and/or hand–sketches of visuals you'd like to make, noting the data sources/columns to be used
- Identification of the intended audience for each viz
- The intended message to be communicated for each viz

Main point – feedback!

Draft

- Expected to still be a work in progress
    - Data visualizations should be largely complete
- Deployment not expected
- Provided to your peers so they can learn from you as much as you can learn from their feedback

Peer Review

- We are all professionals here. It is imperative we act like it.
- Understand the purpose of the exercise.
- Zero tolerance policy for inappropriate comments
- Should be vigorously encouraging

Utilizing GitHub

You'll be assigned three proposals to review (3 points each, plus one bonus point for free)

- Fork their repo, embed comments & suggest changes to their code, submit a PR

Presentation

Order randomly assigned. Basically a chance to share what you created!

- Presentation length will be determined later, but likely to be in the 10–15 minute range (note – you will present as a group)
- Share the final products
- Share the prior iterations
- Discuss the progression along the way and why specific changes were made
- What challenges did you face along the way? What victories did you have that you are particularly proud of?

Grading

Points

150 points total

- 3 labs at 15 points each (45 points; 30%)
- 1 homework assignments at 30 points each (20%)
- 1 Data Viz in the Wild (5 points; 3%)
- Final Project (70 points; 50%)
  - Proposal (5 points; 3%)
  - Draft (10 points; 7%)
  - Peer review (10 points; 7%)
  - Presentation (5 points; 3%)
  - Product (40 points; 27%)

Grading

Lower percent	Lower point range	Grade	Upper point range	Upper percent
0.970+	(146 pts or more)	A+		
0.930	(140 pts)	A	(145 pts)	0.969
0.900	(135 pts)	A–	(139 pts)	0.929
0.870	(131 pts)	B+	(134 pts)	0.899
0.830	(125 pts)	B	(130 pts)	0.869
0.800	(120 pts)	B–	(124 pts)	0.829
0.770	(116 pts)	C+	(119 pts)	0.799
0.730	(110 pts)	C	(115 pts)	0.769
0.700	(105 pts)	C–	(109 pts)	0.739
		F	(104 pts or less)	0.699


COMMUNICATION

Data Visualization Competition

Optional: opt–in/opt–out

- This term, we are hosting a data visualization competition hosted by USAFacts, who provided us with the course data
- This is completely optional and you should feel under no obligation to be part of the competition
- As a group, you neeed to decide whether to opt–in or opt–out by the start of class in Week 3

A note on competition

- Sometimes competition can lead to toxic environments. Let's not do that.
- As previous portions of the syllabus should have made clear, this class is inherently collaborative **as a class** (i.e., you will be pointing out ways to improve your peers visuals through your peer review).
- Should not be stress–inducing. Intended to be a fun way to challenge yourself to do your best work.

Competition

- Week 10, all student groups will present on their final projects.
- Those who opt–in will provide their presentations to three judges (one from UO, one from USAFacts, and one external to both organizations)
- Judges evaluate the visuals using a rating scale (which I will create) and note strengths/weaknesses
- Judges will make ratings independently initially, then will confer to declare a winner
- You will receive the judges' feedback, in addition to mine (which all groups will receive at the end of the term)

Competition

- Will be in a room besides this one
- I will developer a flier to advertise the competition and invite attendees
- Will be both live and virtual (i.e., there will be people joining via zoom)

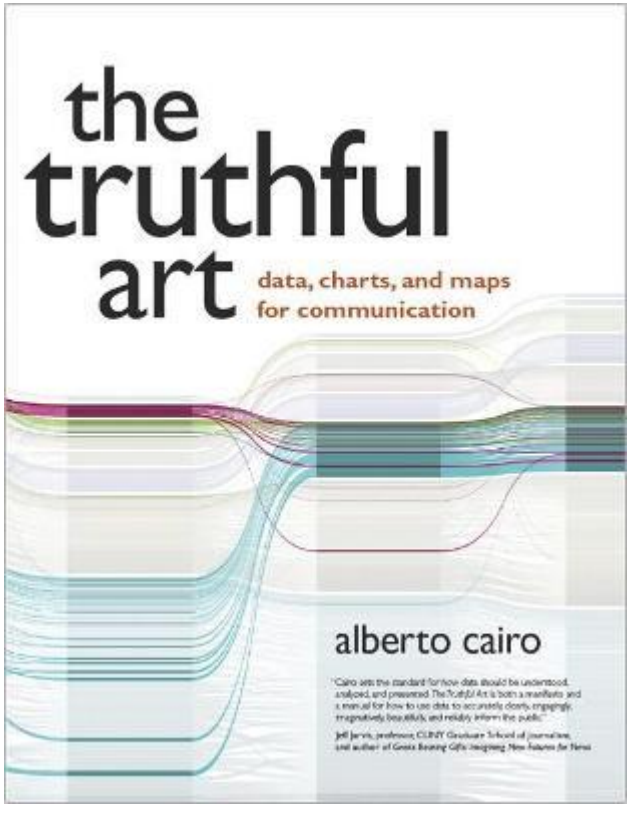
Conditions for participating

- Must opt–in
- Must build your visuals to match the USAFacts style guide
    - Some of this is a bit tricky/finnicky with ggplot2 theming. I would be happy to help you with this outside of class

Why participate?

- The winning group will have their best visual, or possibly all of their visuals, featured on the USAFacts website

- The winning group will also all get copies of Alberto Cairo's the truthful art.



Questions?

Full lecture here

Please do watch the video and read the chapter.

Quick pop quiz

Talk with your neighbor. What do these terms mean?

- stage
- commit
- push
- pull
- clone
- fork
- branch
- merge
- merge conflict
- pull request
- stash

Clone the course repo

Why would we probably not want to fork the repo?

Course data

Getting started

- To make it as easy as possible, I wrote a small R package to make accessing the data easier
- Install with

```
remotes::install_github("datalorax/edld652")
```

Setting your key

- When you first load the package, you will see a message asking you to set a key.
- There is a document on canvas showing you how to do this. We'll go through it together now.
- You only need to do this once, then you can forget about it.
- Please do not share this key with others outside of this class – don't commit it to any repo.
- After you've set your key, go to **Session** on your menu and select **Restart R**.

Check to see if all is working

After you've done everything on the prior slide, run the following to make sure it's working

```
library(edld652)
list_datasets()
```

```
##  [1] "EDFacts_acgr_lea_2011_2019"
##  [2] "EDFacts_acgr_sch_2011_2019"
##  [3] "EDFacts_math_achievement_lea_2010_2019"
##  [4] "EDFacts_math_achievement_sch_2010_2019"
##  [5] "EDFacts_math_participation_lea_2013_2019"
##  [6] "EDFacts_math_participation_sch_2013_2019"
##  [7] "EDFacts_rla_achievement_lea_2010_2019"
##  [8] "EDFacts_rla_achievement_sch_2010_2019"
##  [9] "EDFacts_rla_participation_lea_2013_2019"
## [10] "EDFacts_rla_participation_sch_2013_2019"
## [11] "NCES_CCD_fiscal_district_2010"
## [12] "NCES_CCD_fiscal_district_2011"
## [13] "NCES_CCD_fiscal_district_2012"
## [14] "NCES_CCD_fiscal_district_2013"
```

Accessing a dataset

- The `list_datasets()` function shows you a list of all available datasets
- You can import any of these into R with the `get_data()` function by passing the name of the dataset as a string.

For example: Average cohort graduate rates for local education agency data, 2011 to 2019

```
acgd <- get_data("EDFacts_acgr_lea_2011_2019")
```

```
##
  |
  |                                                      |   0%
  |
  |                                                      |   1%
  |
  |=                                                     |   2%
  |
  |=                                                     |   3%
```

```
acgd
```

```
## # A tibble: 11,326 × 29
##    ALL_COHORT ALL_RATE CWD_COHORT CWD_RATE
##         <dbl> <chr>          <dbl> <chr>
##  1        252 80                 3 PS
##  2        398 75                47 70-79
##  3       1020 89                51 40-49
##  4        750 91                35 60-69
##  5        128 55-59             15 LT50
##  6        166 90-94              9 GE50
##  7        336 90                30 60-79
##  8        273 77                11 LT50
##  9        134 70-74              4 PS
## 10        266 58                33 50-59
## # … with 11,316 more rows, and 25 more variables:
## #   DATE_CUR <chr>, ECD_COHORT <dbl>,
## #   ECD_RATE <chr>, FIPST <chr>, FILEURL <chr>,
## #   LEAID <chr>, LEANM <chr>, LEP_COHORT <dbl>,
## #   LEP_RATE <chr>, MAM_COHORT <dbl>,
## #   MAM_RATE <chr>, MAS_COHORT <dbl>,
## #   MAS_RATE <chr>, MBL_COHORT <dbl>, …
```

Accessing documentation

- The names of the datasets themselves can sometimes be a bit cryptic
- The variable names are often not interpretable at all (particularly the financial data)
- You can access the documentation for any dataset with the `get_documentation()` function, again passing the name of the dataset
- This function operates slightly differently on Mac/Windows

- Mac
  - Creates a folder in your current working directory called `data-documentation`
  - Downloads the documentation and places it in that folder
  - Opens the documentation
  - If the same documentation is requested again, skip the download and just open
- Windows
  - Prints a link to your console where documentation can be downloaded

Note – if any Windows users want to let me borrow their computer for a bit after class one day, I might be able to get it working for Windows as well.

Data demo

For the next 30 minutes or so we will:

- Walk through the overview of the course data together, and then
- Work in small groups to continue to explore the data and come up with new visualizations on your own.

Intro to textual data

Structured vs unstructured

- Most every dataset you've ever worked with is what is referred to as a **structured** dataset – it has rows and columns.
- But there is an incredible amount of data out there that is **unstructured** – it just sort of exists
- Most text data is unstructured. How would you analyze the contents of a book? No rows or columns there

Getting text data

There are **many** ways to get text data. Any digital text could potentially be used as textual data.

How about Wikipedia?

Anything that lives on the web is a common use case. Social media data being perhaps primary among them.

"Screen" scraping

Short foray into web scraping. It's not expected you fully follow this. More about "exposure" and less about building competencies.

Use the rvest package to scrape the data you see "on the screen".

Let's read in the Wikipedia page on Eugene

```
library(rvest)
eugene <- read_html("https://en.wikipedia.org/wiki/Eugene%2C_Oreg
```

Grab paragraphs

The `"#mw-content-text > div.mw-parser-output > p"` is the CSS selector that I pulled from the website

```
paragraphs <- eugene %>%
  html_elements("#mw-content-text > div.mw-parser-output > p") %>
  html_text2()
```

The first paragraph is just an empty line, so they are numbered p + 1

Print the first paragraph

```
cat(stringr::str_wrap(paragraphs[2], 50))
```

```
## Eugene (/juːˈdʒiːn/ yoo-JEEN) is a city in the
## U.S. state of Oregon, in the Pacific Northwest. It
## is at the southern end of the Willamette Valley,
## near the confluence of the McKenzie and Willamette
## rivers, about 50 miles (80 km) east of the Oregon
```

Print the fourth paragraph

```
cat(stringr::str_wrap(paragraphs[5], 50))
```

```
## The first people to settle in the Eugene area were
## known as the Kalapuyans, also written Calapooia
## or Calapooya. They made "seasonal rounds," moving
## around the countryside to collect and preserve
## local foods, including acorns, the bulbs of the
## wapato and camas plants, and berries. They stored
## these foods in their permanent winter village.
## When crop activities waned, they returned to their
## winter villages and took up hunting, fishing, and
## trading.[19][20] They were known as the Chifin
## Kalapuyans and called the Eugene area where they
## lived "Chifin", sometimes recorded as "Chafin" or
## "Chiffin".[21][22]
```

Analysis

How do we analyze the text? What we we even analyze?

First, let's structure it! Turn the text into a simple data frame.

```
library(tidyverse)
eugene_df <- tibble(
  paragraph = seq_along(paragraphs),
  description = paragraphs
)
eugene_df
```

```
## # A tibble: 130 × 2
##    paragraph
##        <int>
##  1         1
##  2         2
##  3         3
##  4         4
##  5         5
##  6         6
##  7         7
##  8         8
##  9         9
```

Can we analyze it now?

Not really... what would we analyze?

Words!

Let's break it into words. This is where the tidytext package comes into play.

The `unnest_tokens()` function

Just like most functions in the tidyverse, we pipe our data to `unnest_tokens()`

- First argument is the name of the new column we want in our data
- Second argument is the text data to process
- Third argument is how the text should processed. The default is `"words"`, meaning the text will be broken into words.

Example

```
library(tidytext)
eugene_tidy_words <- eugene_df %>%
  unnest_tokens(word, description)
eugene_tidy_words
```

```
## # A tibble: 7,814 × 2
##    paragraph word
##        <int> <chr>
##  1         2 eugene
##  2         2 juːˈdʒiːn
##  3         2 yoo
##  4         2 jeen
##  5         2 is
##  6         2 a
##  7         2 city
##  8         2 in
##  9         2 the
## 10         2 u.s
## # … with 7,804 more rows
```

Not perfect, but pretty good

What to do now?

Let's count some words!

```
eugene_tidy_words %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 2,478 × 2
##    word       n
##    <chr>  <int>
##  1 the      597
##  2 and      249
##  3 of       231
##  4 in       230
##  5 eugene   178
##  6 a        136
##  7 to       116
##  8 is        95
##  9 for       76
## 10 was       76
## # … with 2,468 more rows
```

Plot the top 15 words

```
eugene_tidy_words %>%
  count(word, sort = TRUE) %>%
  mutate(word = reorder(word, n)) %>% # make y-axis ordered by n
  slice(1:15) %>% # select only the first 15 rows
  ggplot(aes(n, word)) +
    geom_col(fill = "cornflowerblue")
```

Not very informative

Why?

Most of the words are common words like "the", "and", "of" (top three words)

These are referred to as "stop words".

Luckily, **tidytext** provides us with a dictionary of stop words. We can use an `anti_join()` with this dictionary to remove these words.

Quick refresher

A `semi_join()` works just like an `inner_join()`, but without adding any columns. A `semi_join()` works by **keeping** only rows that are in common with the two datasets.

An `anti_join()` does basically the opposite, by **removing** any rows that are in common between the two datasets.

Look at the stop words

This dataset is available to you as soon as you load **tidytext**.

There are three lexicons – I usually use all three.

```
stop_words
```

```
## # A tibble: 1,149 × 2
##    word        lexicon
##    <chr>       <chr>
##  1 a           SMART
##  2 a's         SMART
##  3 able        SMART
##  4 about       SMART
##  5 above       SMART
##  6 according   SMART
##  7 accordingly SMART
##  8 across      SMART
##  9 actually    SMART
## 10 after       SMART
## # … with 1,139 more rows
```

Count

Let's try counting again without the stop words included.

```
eugene_tidy_words %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 2,199 × 2
##    word            n
##    <chr>       <int>
##  1 eugene        178
##  2 city           54
##  3 oregon         50
##  4 university     40
##  5 community      27
##  6 lane           24
##  7 eugene's       23
##  8 college        20
##  9 home           20
## 10 center         19
## # … with 2,189 more rows
```

So much more informative!

Plot the top 15 words

```
eugene_tidy_words %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE) %>%
  mutate(word = reorder(word, n)) %>% # make y-axis ordered by n
  slice(1:15) %>% # select only the first 15 rows
  ggplot(aes(n, word)) +
    geom_col(fill = "cornflowerblue")
```

Add the headers in

I hid the code here because it's weird and inefficient but the HTML structure made it difficult. You can look at the source if you want. The new dataframe is called `eugene_tidy_words2`.

```
## # A tibble: 130 × 3
##    paragraph header
##        <int> <chr>
##  1         1 Intro
##  2         2 Intro
##  3         3 Intro
##  4         4 Intro
##  5         5 History
##  6         6 History
##  7         7 History
##  8         8 History
##  9         9 History
## 10        10 History
## # … with 120 more rows, and 1 more variable:
## #   description <chr>
```

Count words by header

Not surprisingly, "eugene" appears to be the most common among multiple categories.

We might want to remove "eugene" as well.

```
eugene_tidy_words2 %>%
  unnest_tokens(word, description) %>%
  count(header, word, sort = TRUE) %>%
  anti_join(stop_words)
```

```
## # A tibble: 3,029 × 3
##    header            word       n
##    <chr>             <chr>  <int>
##  1 Arts and culture  eugene    54
##  2 History           eugene    21
##  3 Infrastructure    eugene    20
##  4 Arts and culture  church    19
##  5 Education         eugene    15
##  6 Geography         eugene    15
##  7 Economy           eugene    14
##  8 Education         school    14
##  9 Government        eugene    14
```

Plot

Top 15 words by header

```
p <- eugene_tidy_words2 %>%
  unnest_tokens(word, description) %>%
  count(header, word, sort = TRUE) %>%
  anti_join(stop_words) %>%
  group_by(header) %>%
  slice(1:15) %>%
  ggplot(aes(n, word)) +
    geom_col(fill = "cornflowerblue") +
    facet_wrap(~header, scales = "free_y")
```

Next time

- Finish up on text data
- Discuss string manipulations
- Discuss distribution/binning