Welcome!

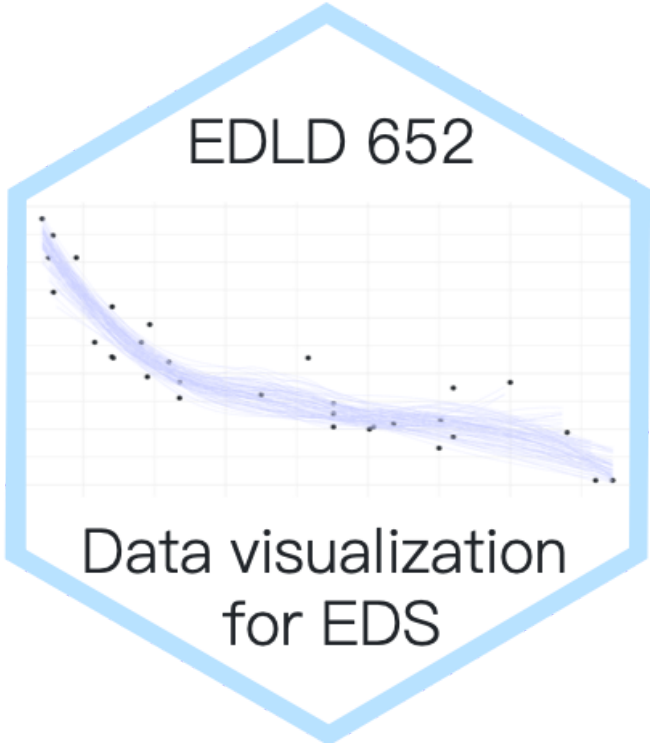
An overview of the course

Maithreyi Gopalan

Week 1

Agenda

- Getting on the same page
- Syllabus
- Discuss project details available for sign-up this term
- Finalize the personalized meeting schedule for the rest of the term



who am i

- Associate Professor
- Pronouns: she/her/hers
- Primary areas of interest: educational equity, policy analysis, social psychology,systemic inequities in opportunities and achievement
- Big secret: I was not a 💗R💗; have been a SAS and STATA user mostly, so learning R with y'all!

who is you?

- 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.

In-person class

In-person class

- This class is in-person
- Your class participation grade comes exclusively from your active participation in the class through discussions and hands-on lab sessions
- If you are not feeling well, please do not attend in person
- See syllabus for attendance policy

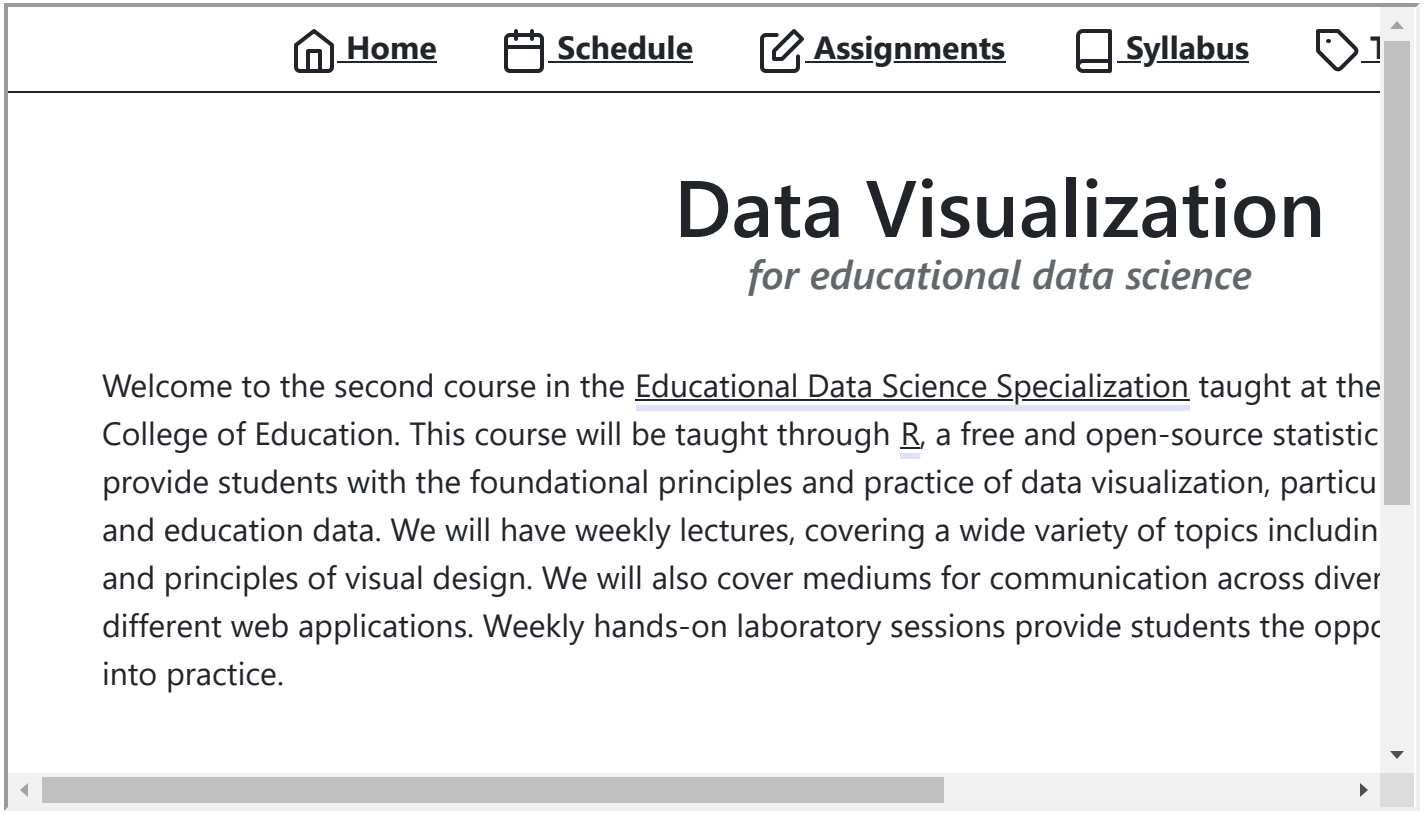
Last intro thing

- 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



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
- Indeed, all of today's lab is going to be about git and next week will be refresher on R Markdowns!

Learning objectives

- Transform data in a variety of ways to create effective data visualizations
- Understand best practices in data visualization
- Create and customize graphics in a variety of ways using best practices (e.g., visual perception, color choices, text annotations, categorical axis ordering, uncertainty)
- Build web-based platforms for sharing data visualizations

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

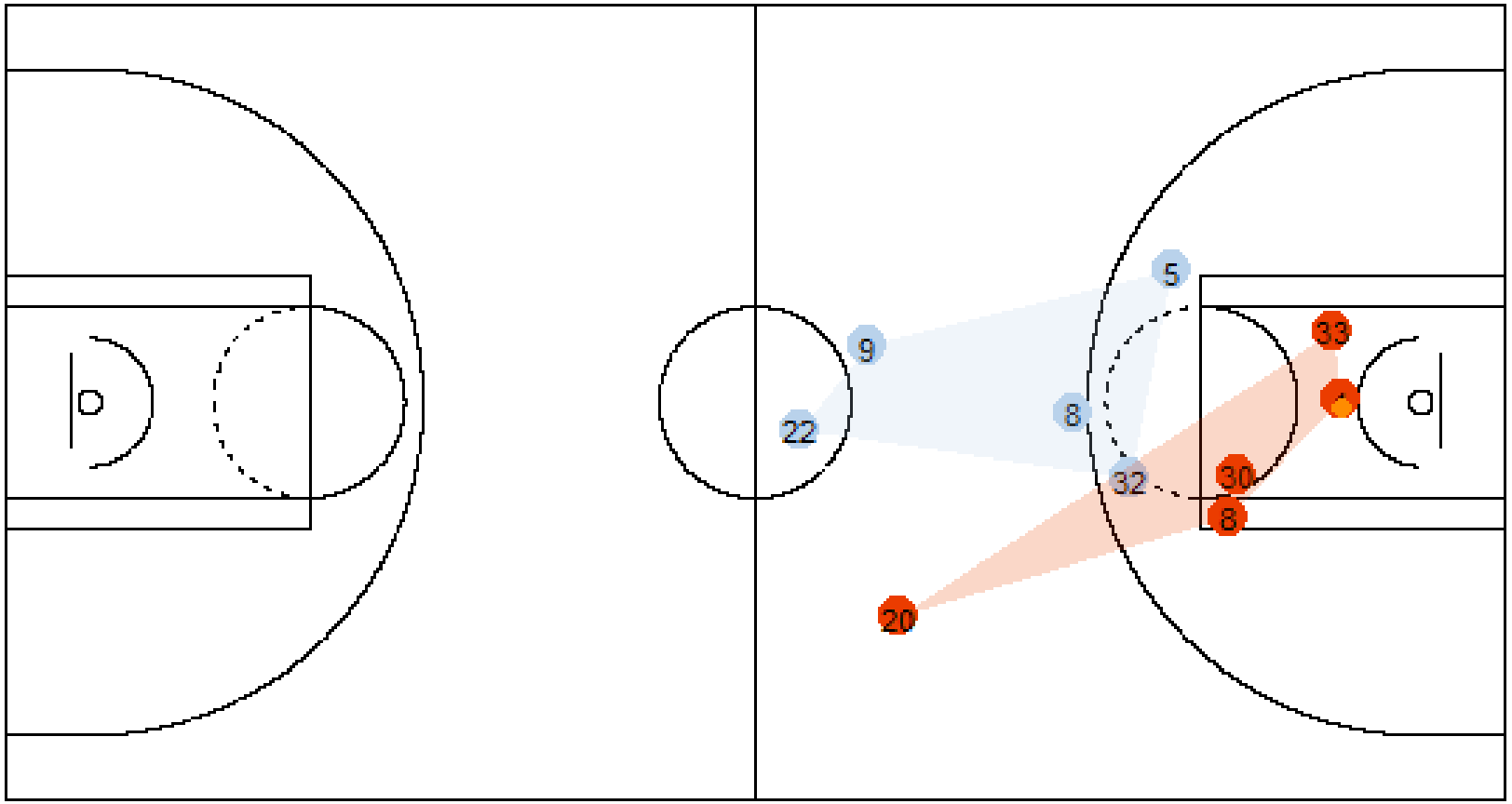
- Understand the requirements of the course
- Understand the requirements of the final project
- Be ready to go with *git* and GitHub
- Understand how to access the course data and documentation, begin playing with the data

Some examples

Timo Grossenbacher

Paul Campbell

Patrick Honner via NYT



James Curley

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

[1] 1

Date	Presenter
2025-01-15	Michelle
2025-01-15	EmilyW
2025-01-22	Nakyung
2025-01-22	Aden
2025-01-29	Elizabeth
2025-01-29	Erick

Date	Presenter
2025-02-05	Saratessa
2025-02-05	Sophia
2025-02-12	Will
2025-02-12	Maiko
2025-02-19	Songyi
2025-02-19	EmilyM

I will email this out as well.

Final Project

70 points total (35%)

Group project

- Please try to finalize your group by the end of today. You will have time during lab today 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.

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 four 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

Three 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!

Peer Review (as part of Lab 8)

- 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?

Questions?

Lab 1

Quick refresher on git here

You can also watch a Full lecture after class 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

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 and
## the county seat of Lane County, Oregon, United
## States. It is located 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 Coast.[10]
```

Print the fourth paragraph

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

```
## The first people to settle in the Eugene area
## were 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.[20][21] They were known as the Chifin
## Kalapuyans and called the Eugene area where they
## lived "Chifin", sometimes recorded as "Chafin" or
## "Chiffin".[22][23]
```

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: 133 × 2
##    paragraph description
##        <int> <chr>
##  1         1 ""
##  2         2 "Eugene (/juːˈdʒiːn/ yoo-JEEN) is a city in and the county
##  3         3 "The second-most populous city in Oregon, Eugene had a popu
##  4         4 "Eugene is home to the University of Oregon, Bushnell Unive
##  5         5 "The first people to settle in the Eugene area were the Kal
##  6         6 "Other Kalapuyan tribes occupied villages that are also now
##  7         7 "According to archeological evidence, the ancestors of the
##  8         8 "French fur traders had settled seasonally in the Willamett
##  9         9 "In July 1830, \"intermittent fever\" struck the lower Colu
## 10        10 "As the demographic pressure from the settlers grew, the re
```

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: 8,242 × 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 and
## 10         2 the
## # ℹ 8,232 more rows
```

Not perfect, but pretty good

What to do now?

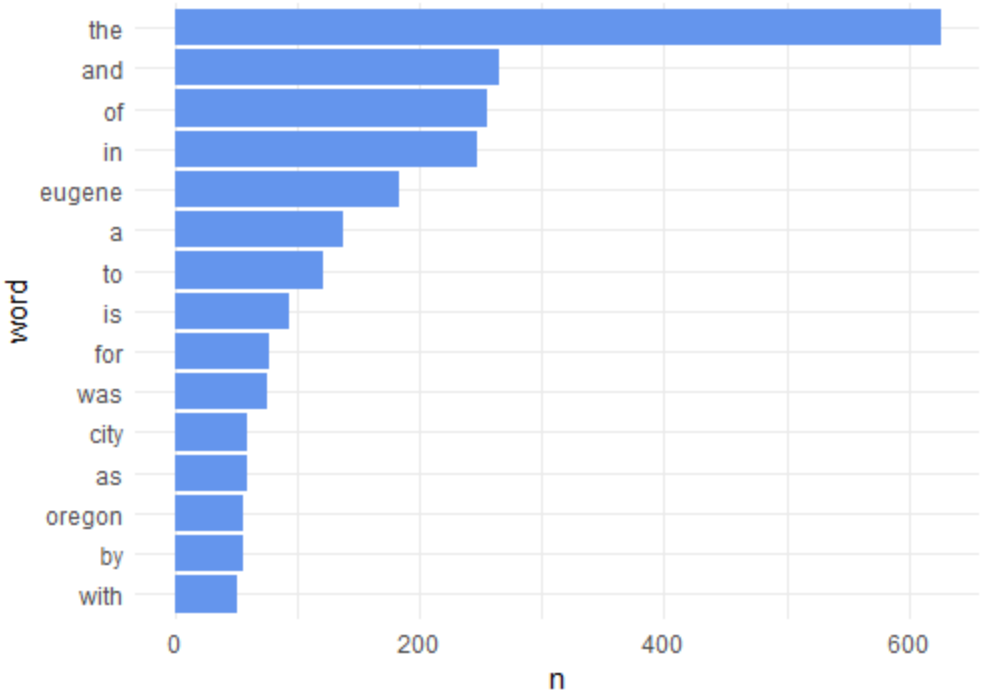
Let's count some words!

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

```
## # A tibble: 2,584 × 2
##    word       n
##    <chr>  <int>
##  1 the      626
##  2 and      266
##  3 of       256
##  4 in       247
##  5 eugene   184
##  6 a        138
##  7 to       121
##  8 is        94
##  9 for       78
## 10 was       76
## # ℹ 2,574 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
## # ℹ 1,139 more rows
```

Count

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

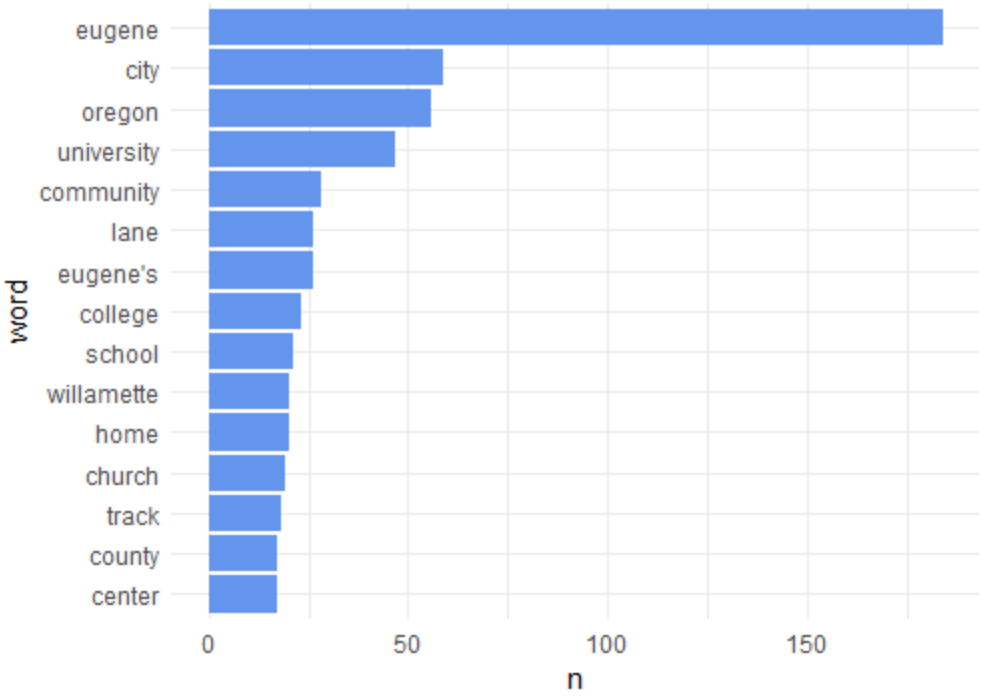
```
eugene_tidy_words %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 2,296 × 2
##    word            n
##    <chr>       <int>
##  1 eugene        184
##  2 city           59
##  3 oregon         56
##  4 university     47
##  5 community      28
##  6 eugene's       26
##  7 lane           26
##  8 college        23
##  9 school         21
## 10 home           20
## # ℹ 2,286 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")
```



Working with data

Getting started

- To make it as easy as possible, Daniel Anderson wrote package to make accessing a set of EDFacts data easier. Let's play with that one if time permits

- Install with

```
#detach("package:edld652", unload = TRUE)
remotes::install_github("datalorax/edld652", force = TRUE)
#set_key("maithgopalan_testkey")
```

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()
```

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")
```

```
##
  |
  |
  |
  |===========
  |
  |===============
  |
  |============================
  |
  |============================================================
```

```
acgdd <- get_documentation("EDFacts_acgr_lea_2011_2019")
# acgdd
```

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

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.

Next time

- Quick refresher on R Markdowns
- Discuss string manipulations
- Discuss distribution/binning