Introduction to Data Science

Welcome and Motivation Session

Nith Kosal

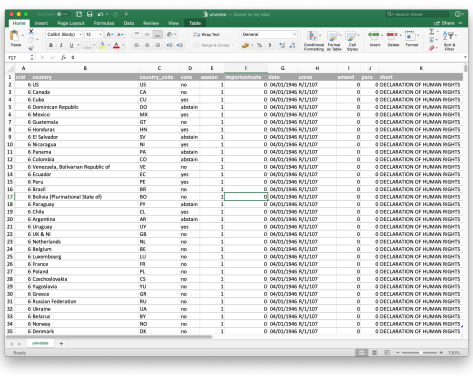
Future Forum

Future Forum, April 9, 2022

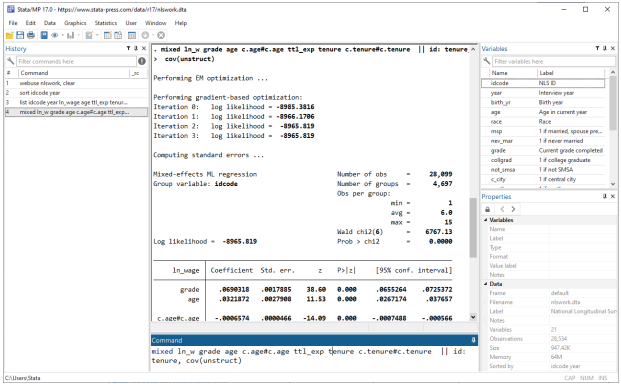
Dataset Terminology

- Data science is an exciting discipline that allows you to turn raw data into understanding, insight, and knowledge.
- We're going to learn to do this in a tidy way – more on that later!
- This is a course on introduction to data science, with an emphasis on statistical thinking.

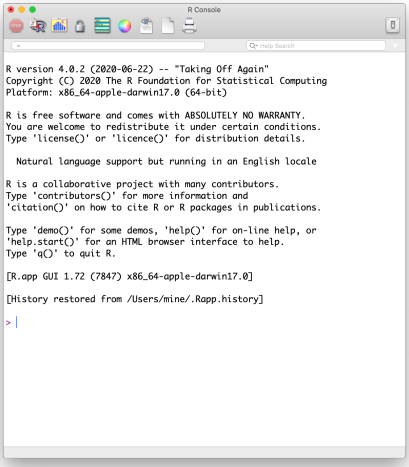
Theoretical Model



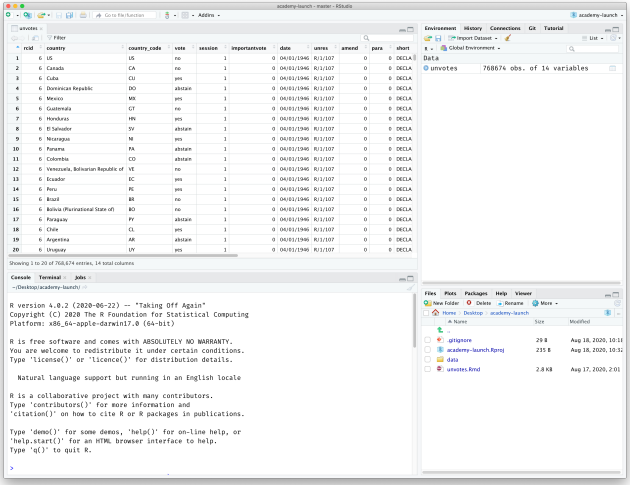
(a) MS Excel



(b) Stata

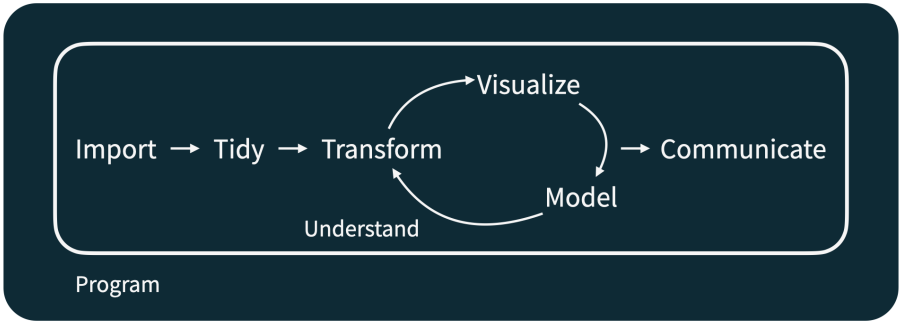


(c) RStudio

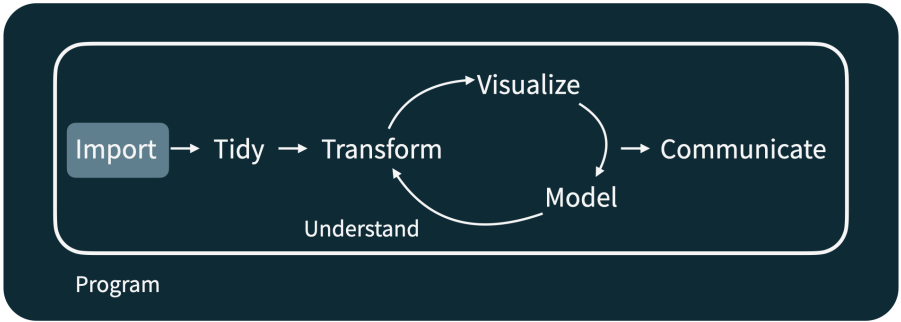


(d) RStudio

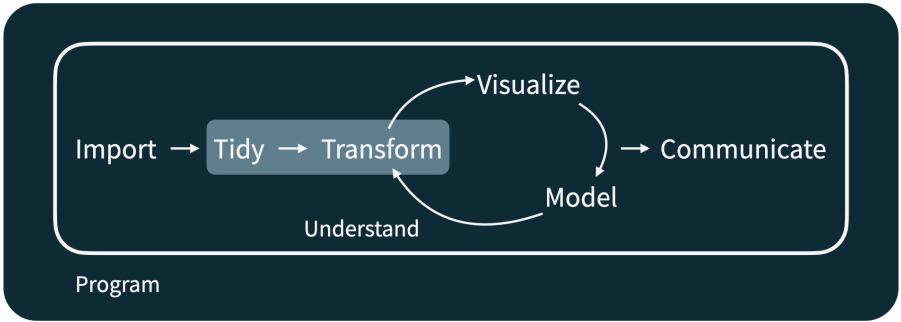
Data Science Life Cycle



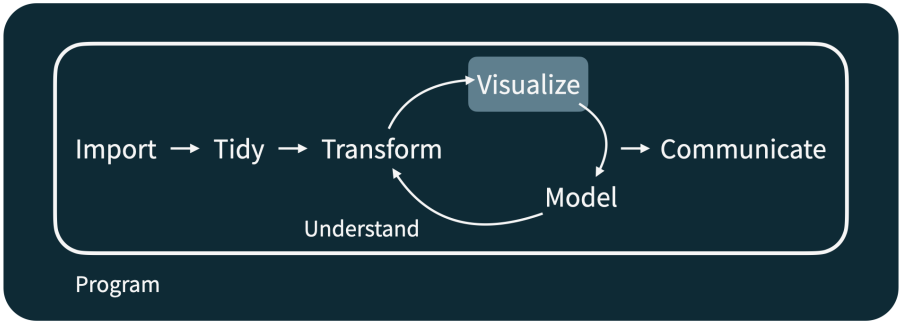
Data Science Life Cycle



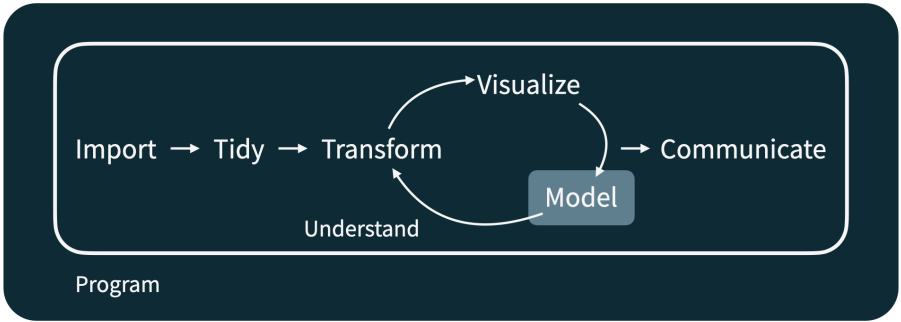
Data Science Life Cycle



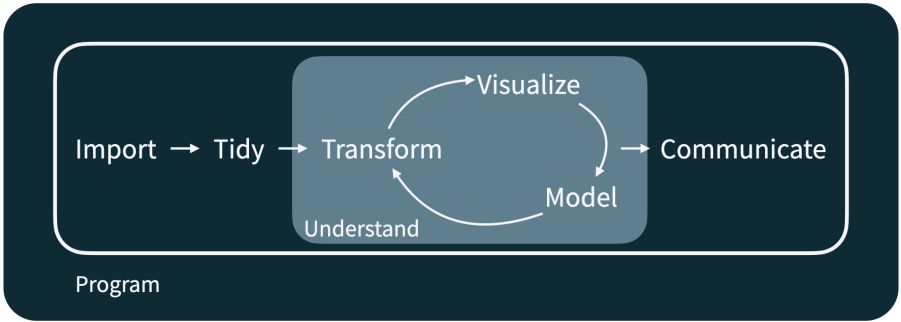
Data Science Life Cycle



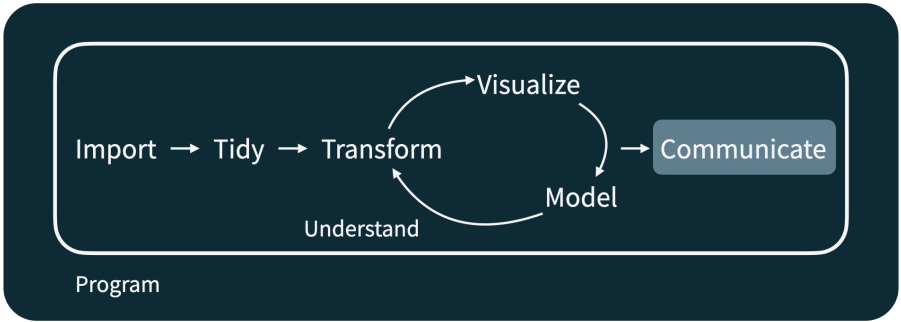
Data Science Life Cycle



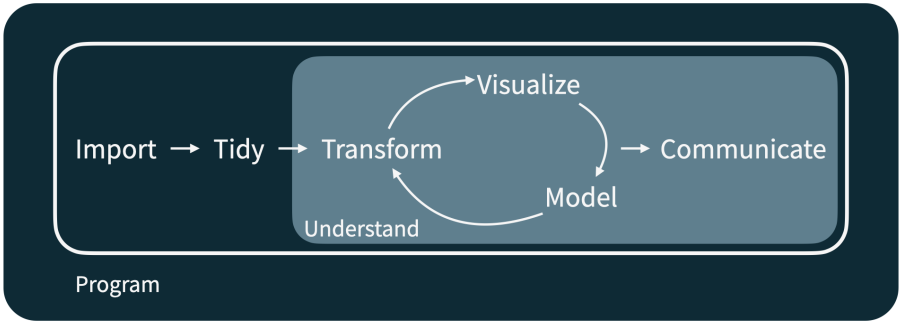
Data Science Life Cycle



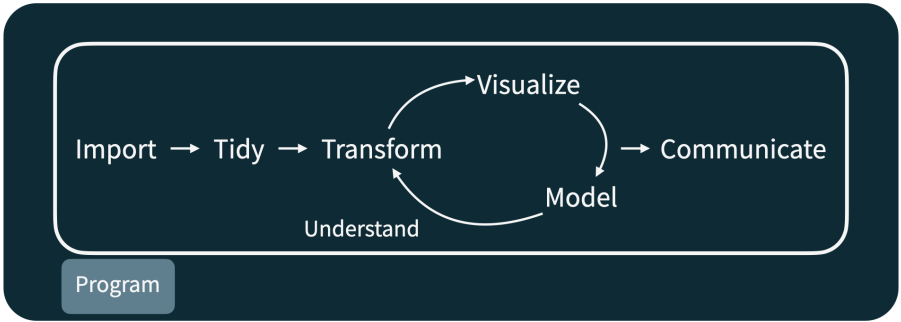
Data Science Life Cycle

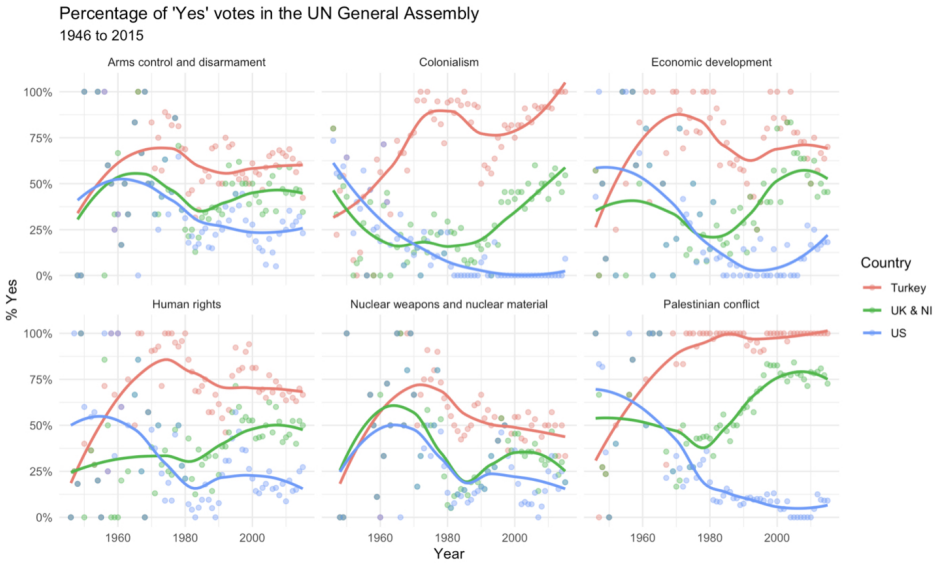


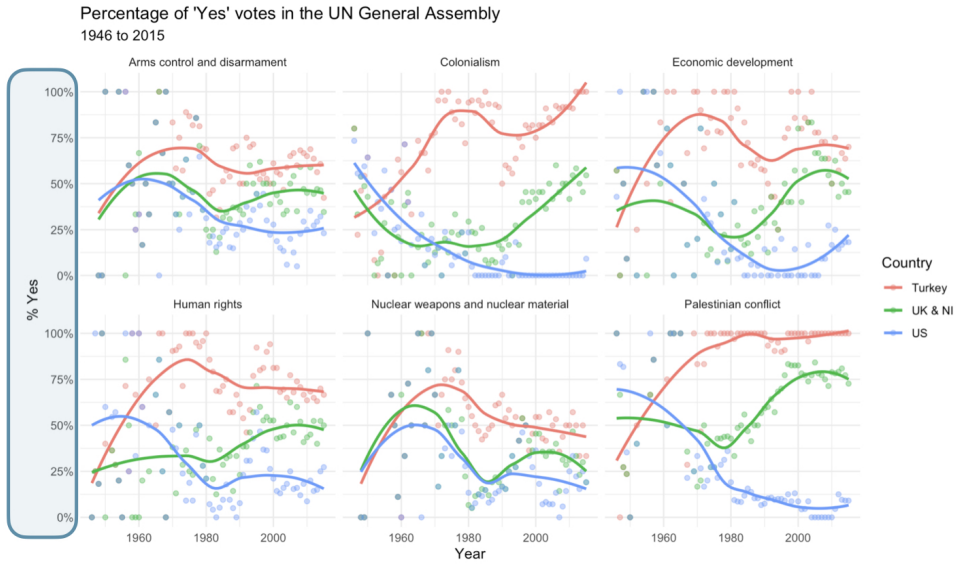
Data Science Life Cycle

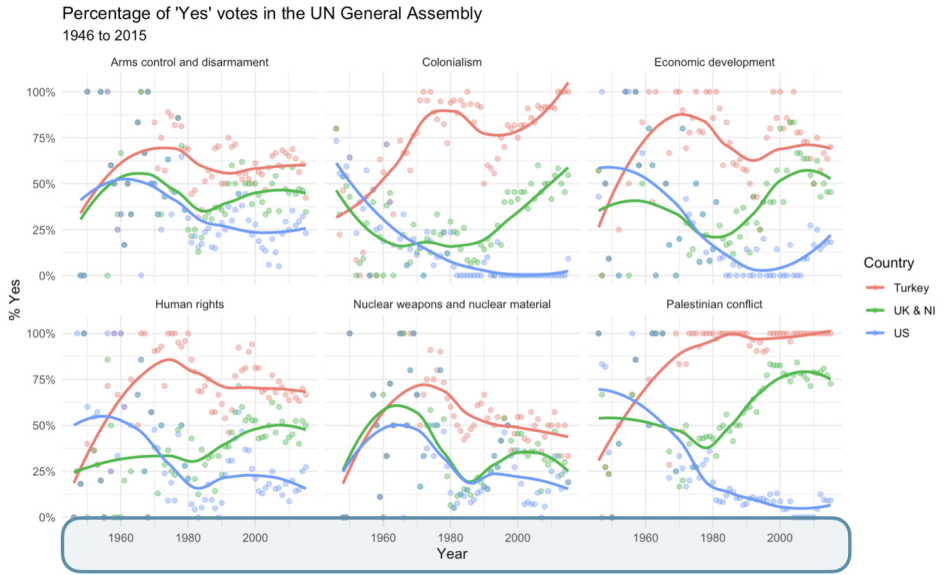


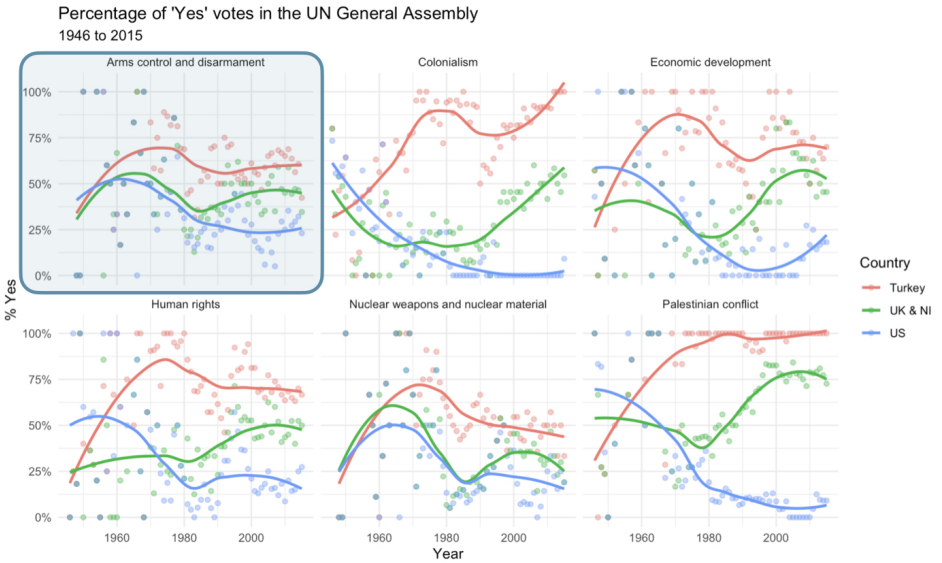
Data Science Life Cycle

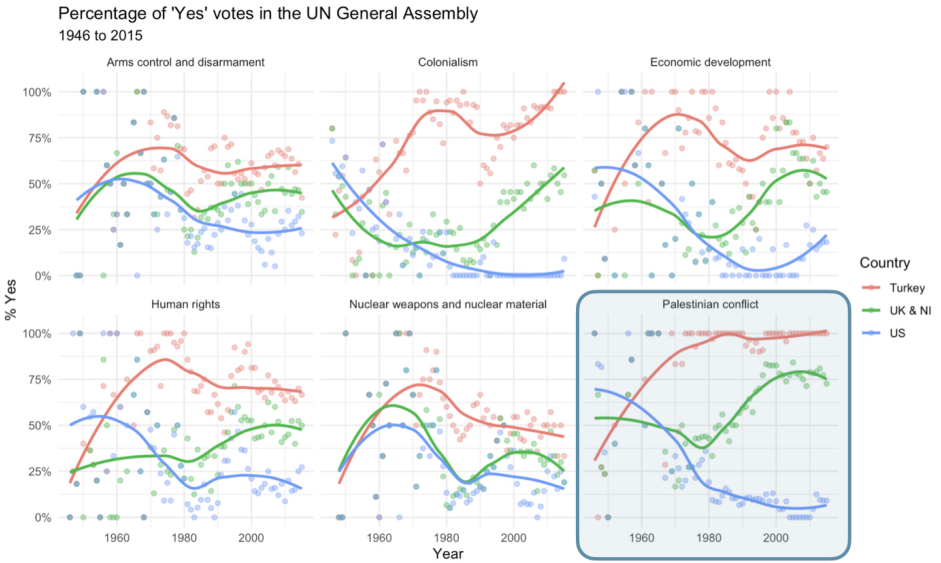


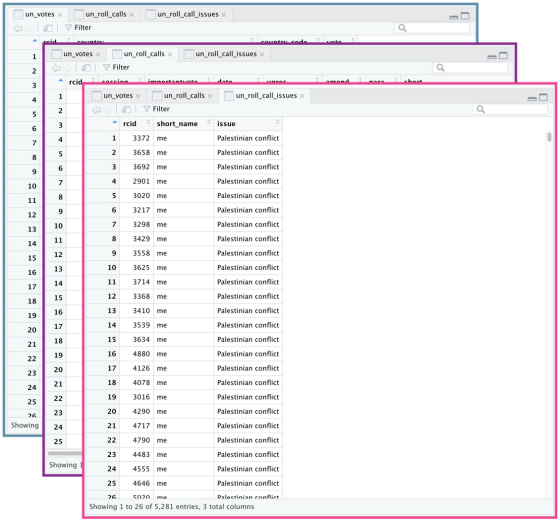

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US

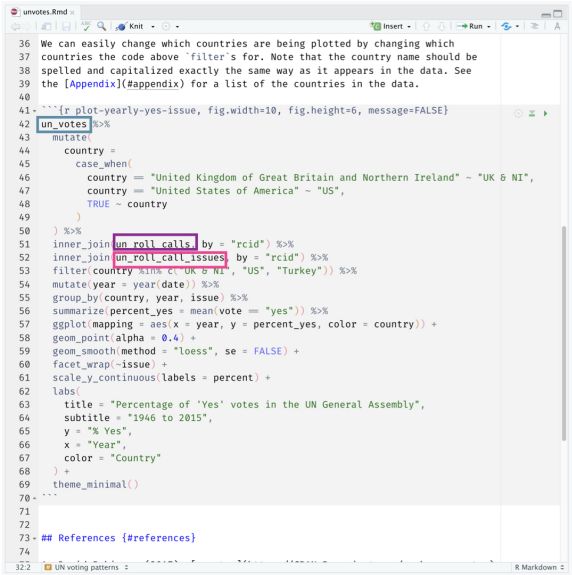

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US

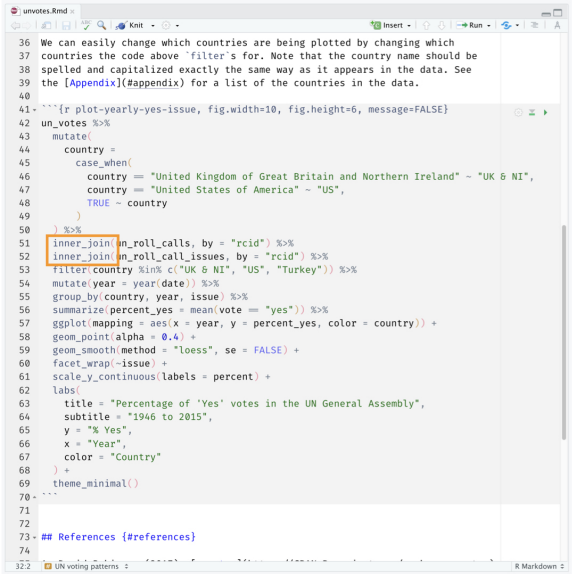

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US

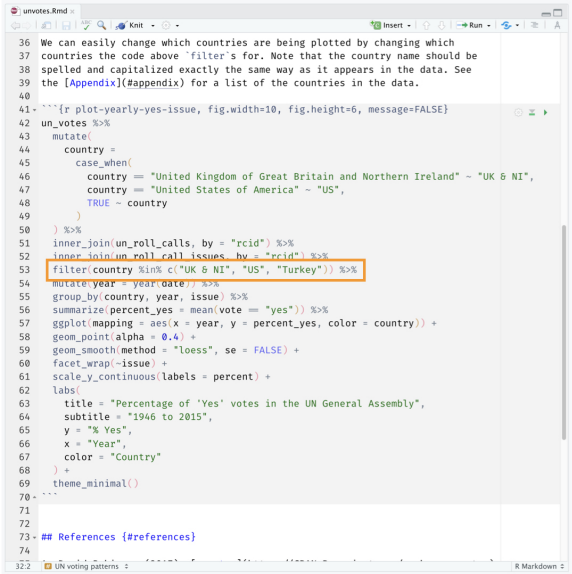

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US

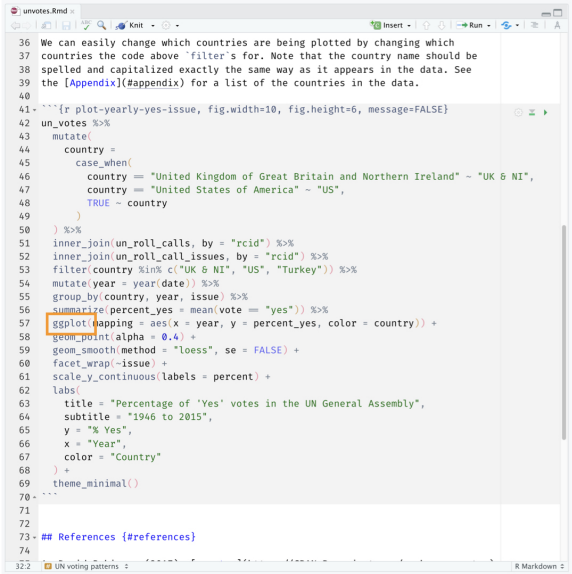

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US


un_votes
un_roll_calls
un_roll_call_issues
Filter
rcid
short_name
issue
1 3372 me Palestinian conflict
2 3658 me Palestinian conflict
3 3692 me Palestinian conflict
4 2901 me Palestinian conflict
5 3020 me Palestinian conflict
6 3217 me Palestinian conflict
7 3298 me Palestinian conflict
8 3429 me Palestinian conflict
9 3558 me Palestinian conflict
10 3625 me Palestinian conflict
11 3714 me Palestinian conflict
12 3368 me Palestinian conflict
13 3410 me Palestinian conflict
14 3539 me Palestinian conflict
15 3634 me Palestinian conflict
16 4880 me Palestinian conflict
17 4126 me Palestinian conflict
18 4078 me Palestinian conflict
19 3016 me Palestinian conflict
20 4290 me Palestinian conflict
21 4717 me Palestinian conflict
22 4790 me Palestinian conflict
23 4483 me Palestinian conflict
24 4555 me Palestinian conflict
25 4646 me Palestinian conflict
Showing 1 to 26 of 5,281 entries, 3 total columns


unvotes.Rmd
Knit
Insert
Run
We can easily change which countries are being plotted by changing which
countries the code above `filter`s for. Note that the country name should be
spelled and capitalized exactly the same way as it appears in the data. See
the [Appendix](#appendix) for a list of the countries in the data.
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
mutate(
country =
case_when(
country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
country == "United States of America" ~ "US",
TRUE ~ country
)
) %>%
inner_join(un_roll_calls, by = "rcid") %>%
inner_join(un_roll_call_issues, by = "rcid") %>%
filter(country %in% c("UK & NI", "US", "Turkey")) %>%
mutate(year = year(date)) %>%
group_by(country, year, issue) %>%
summarize(percent_yes = mean(vote == "yes")) %>%
ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
geom_point(alpha = 0.4) +
geom_smooth(method = "loess", se = FALSE) +
facet_wrap(~issue) +
scale_y_continuous(labels = percent) +
labs(
title = "Percentage of 'Yes' votes in the UN General Assembly",
subtitle = "1946 to 2015",
y = "% Yes",
x = "Year",
color = "Country"
) +
theme_minimal()
```
References {#references}
32:2
UN voting patterns
R Markdown


unvotes.Rmd
Knit
Insert
Run
We can easily change which countries are being plotted by changing which
countries the code above `filter`s for. Note that the country name should be
spelled and capitalized exactly the same way as it appears in the data. See
the [Appendix](#appendix) for a list of the countries in the data.
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
mutate(
country =
case_when(
country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
country == "United States of America" ~ "US",
TRUE ~ country
)
) %>%
inner_join(un_roll_calls, by = "rcid") %>%
inner_join(un_roll_call_issues, by = "rcid") %>%
filter(country %in% c("UK & NI", "US", "Turkey")) %>%
mutate(year = year(date)) %>%
group_by(country, year, issue) %>%
summarize(percent_yes = mean(vote == "yes")) %>%
ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
geom_point(alpha = 0.4) +
geom_smooth(method = "loess", se = FALSE) +
facet_wrap(~issue) +
scale_y_continuous(labels = percent) +
labs(
title = "Percentage of 'Yes' votes in the UN General Assembly",
subtitle = "1946 to 2015",
y = "% Yes",
x = "Year",
color = "Country"
) +
theme_minimal()
```
References {#references}
32:2
UN voting patterns
R Markdown


unvotes.Rmd
Knit
Insert
Run
We can easily change which countries are being plotted by changing which
countries the code above `filter`s for. Note that the country name should be
spelled and capitalized exactly the same way as it appears in the data. See
the [Appendix](#appendix) for a list of the countries in the data.
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
mutate(
country =
case_when(
country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
country == "United States of America" ~ "US",
TRUE ~ country
)
) %>%
inner_join(un_roll_calls, by = "rcid") %>%
inner_join(un_roll_call_issues, by = "rcid") %>%
filter(country %in% c("UK & NI", "US", "Turkey")) %>%
mutate(year = year(date)) %>%
group_by(country, year, issue) %>%
summarize(percent_yes = mean(vote == "yes")) %>%
ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
geom_point(alpha = 0.4) +
geom_smooth(method = "loess", se = FALSE) +
facet_wrap(~issue) +
scale_y_continuous(labels = percent) +
labs(
title = "Percentage of 'Yes' votes in the UN General Assembly",
subtitle = "1946 to 2015",
y = "% Yes",
x = "Year",
color = "Country"
) +
theme_minimal()
```
References {#references}
32:2
UN voting patterns
R Markdown


unvotes.Rmd
Knit
Insert
Run
We can easily change which countries are being plotted by changing which
countries the code above `filter`s for. Note that the country name should be
spelled and capitalized exactly the same way as it appears in the data. See
the [Appendix](#appendix) for a list of the countries in the data.
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
mutate(
country =
case_when(
country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
country == "United States of America" ~ "US",
TRUE ~ country
)
) %>%
inner_join(un_roll_calls, by = "rcid") %>%
inner_join(un_roll_call_issues, by = "rcid") %>%
filter(country %in% c("UK & NI", "US", "Turkey")) %>%
mutate(year = year(date)) %>%
group_by(country, year, issue) %>%
summarize(percent_yes = mean(vote == "yes")) %>%
ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
geom_point(alpha = 0.4) +
geom_smooth(method = "loess", se = FALSE) +
facet_wrap(~issue) +
scale_y_continuous(labels = percent) +
labs(
title = "Percentage of 'Yes' votes in the UN General Assembly",
subtitle = "1946 to 2015",
y = "% Yes",
x = "Year",
color = "Country"
) +
theme_minimal()
```
References {#references}
32:2
UN voting patterns
R Markdown

unvotes.Rmd

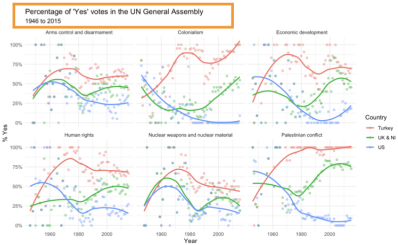
Knit · Insert · Run

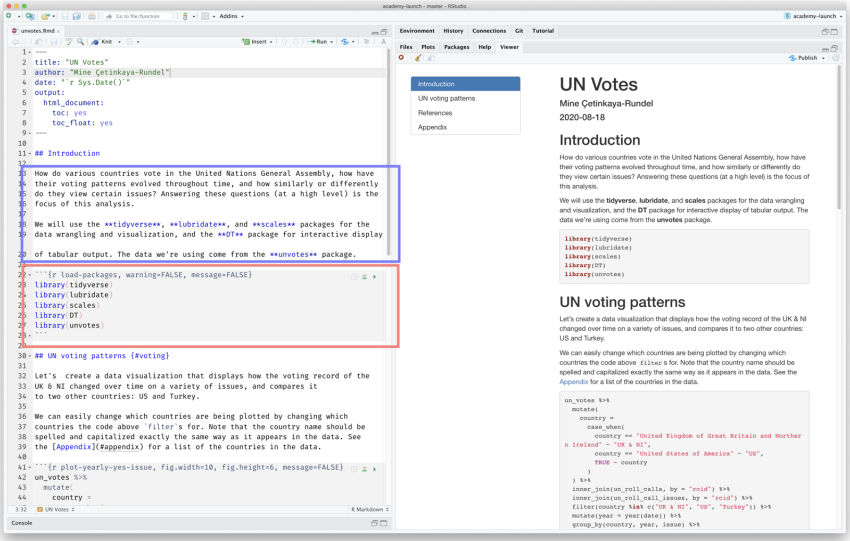
```
We can easily change which countries are being plotted by changing which
countries the code above `filter`s for. Note that the country name should be
spelled and capitalized exactly the same way as it appears in the data. See
the [Appendix](#appendix) for a list of the countries in the data.

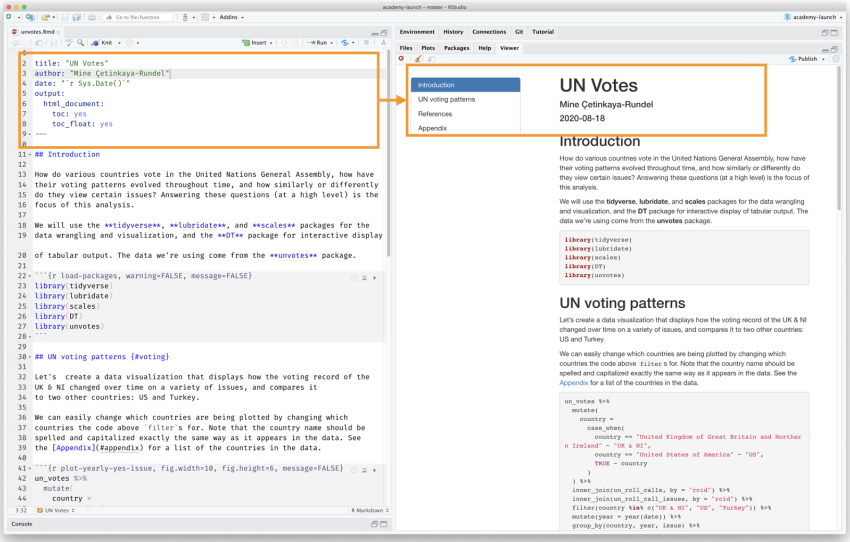
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
  mutate(
    country =
      case_when(
        country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
        country == "United States of America" ~ "US",
        TRUE ~ country
      )
  ) %>%
  inner_join(un_roll_calls, by = "rcid") %>%
  inner_join(un_roll_call_issues, by = "rcid") %>%
  filter(country %in% c("UK & NI", "US", "Turkey")) %>%
  mutate(year = year(date)) %>%
  group_by(country, year, issue) %>%
  summarize(percent_yes = mean(vote == "yes")) %>%
  ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
  facet_wrap(~issue) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "Percentage of 'Yes' votes in the UN General Assembly",
    subtitle = "1946 to 2015",
    y = "% Yes",
    x = "Year",
    color = "Country"
  ) +
  theme_minimal()
```


References {#references}
```

32:2 UN voting patterns · R Markdown




UN Votes
Mine Çetinkaya-Rundel
2020-08-18
Introduction
How do various countries vote in the United Nations General Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do they view certain issues? Answering these questions (at a high level) is the focus of this analysis.
We will use the tidyverse, lubridate, and scales packages for the data wrangling and visualization, and the DT package for interactive display of tabular output. The data we're using come from the unvotes package.
UN voting patterns
Let's create a data visualization that displays how the voting record of the UK & NI changed over time on a variety of issues, and compares it to two other countries: US and Turkey.
We can easily change which countries are being plotted by changing which countries the code above filter s for. Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. See the Appendix for a list of the countries in the data.
Introduction
UN voting patterns
References
Appendix


academy-launch - master - RStudio
unvotes.Rmd
title: "UN Votes"
author: "Mine Çetinkaya-Rundel"
date: "`r Sys.Date()`"
output:
html_document:
toc: yes
toc_float: yes
Introduction
How do various countries vote in the United Nations General Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do they view certain issues? Answering these questions (at a high level) is the focus of this analysis.
We will use the **tidyverse**, **lubridate**, and **scales** packages for the data wrangling and visualization, and the **DT** package for interactive display of tabular output. The data we're using come from the **unvotes** package.
```{r load-packages, warning=FALSE, message=FALSE}
library(tidyverse)
library(lubridate)
library(scales)
library(DT)
library(unvotes)
```
UN voting patterns {#voting}
Let's create a data visualization that displays how the voting record of the UK & NI changed over time on a variety of issues, and compares it to two other countries: US and Turkey.
We can easily change which countries are being plotted by changing which countries the code above `filter`s for. Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. See the [Appendix](#appendix) for a list of the countries in the data.
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
mutate(
country =
Environment
History
Connections
Git
Tutorial
Files
Plots
Packages
Help
Viewer
Publish
Introduction
UN voting patterns
References
Appendix
UN Votes
Mine Çetinkaya-Rundel
2020-08-18
Introduction
How do various countries vote in the United Nations General Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do they view certain issues? Answering these questions (at a high level) is the focus of this analysis.
We will use the tidyverse, lubridate, and scales packages for the data wrangling and visualization, and the DT package for interactive display of tabular output. The data we're using come from the unvotes package.
library(tidyverse)
library(lubridate)
library(scales)
library(DT)
library(unvotes)
UN voting patterns
Let's create a data visualization that displays how the voting record of the UK & NI changed over time on a variety of issues, and compares it to two other countries: US and Turkey.
We can easily change which countries are being plotted by changing which countries the code above filters for. Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. See the Appendix for a list of the countries in the data.
un_votes %>%
mutate(
country =
case_when(
country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
country == "United States of America" ~ "US",
TRUE ~ country
)
) %>%
inner_join(un_roll_calls, by = "rcid") %>%
inner_join(un_roll_call_issues, by = "rcid") %>%
filter(country %in% c("UK & NI", "US", "Turkey")) %>%
mutate(year = year(date)) %>%
group_by(country, year, issue) %>%
Console
```

Learning Goals

By the end of the course, you will be able to...

- gain insight from data
- gain insight from data, reproducibly
- gain insight from data, reproducibly, using modern programming tools and techniques
- gain insight from data, reproducibly and collaboratively, using modern programming tools and techniques
- gain insight from data, reproducibly (with literate programming and version control) and collaboratively, using modern programming tools and techniques

Reproducible Data Analysis

- Near-term goals:
  - Are the tables and figures reproducible from the code and data?
  - Does the code actually do what you think it does?
  - In addition to what was done, is it clear why it was done?
- Long-term goals:
  - Can the code be used for other data?
  - Can you extend the code to do other things?

Toolkit for Reproducibility

- Scriptability $\rightarrow$ R
- Literate programming (code, narrative, output in one place) $\rightarrow$ R Markdown
- Version control $\rightarrow$ Git / GitHub

R and RStudio

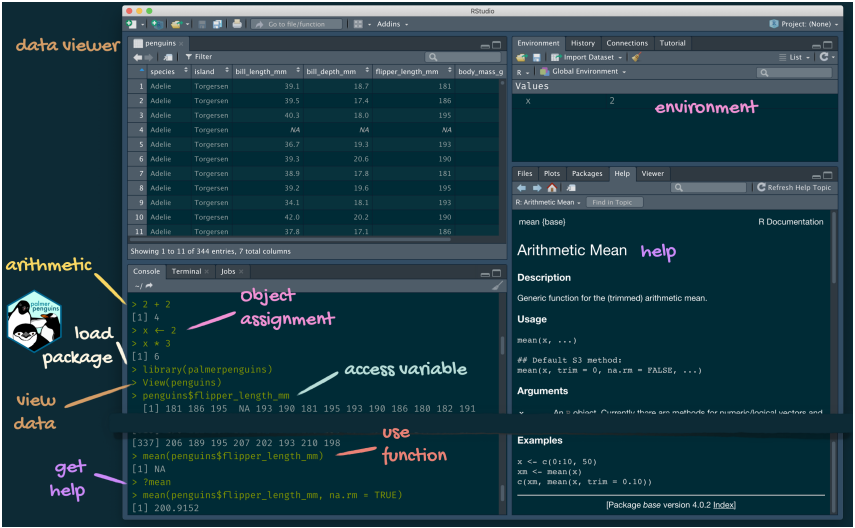
- R is an open-source statistical programming language
- R is also an environment for statistical computing and graphics
- It's easily extensible with *packages*

- RStudio is a convenient interface for R called an IDE (integrated development environment), e.g. *"I write R code in the RStudio IDE"*
- RStudio is not a requirement for programming with R, but it's very commonly used by R programmers and data scientists

R Packages

- Packages are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data
- As of September 2020, there are over 16,000 R packages available on CRAN (the Comprehensive R Archive Network)
- We're going to work with a small (but important) subset of these!
- The tidyverse is an opinionated collection of R packages designed for data science.
- All packages share an underlying philosophy and a common grammar

Tour: R and RStudio



Tour: R and RStudio

Functions are (most often) verbs, followed by what they will be applied to in parentheses:

```
do_this(to_this)
do_that(to_this, to_that, with_those)
```

Packages are installed with the install.packages function and loaded with the library function, once per session:

```
install.packages("package_name")
library(package_name)
```

Columns (variables) in data frames are accessed with $:

```
dataframe$var_name
```

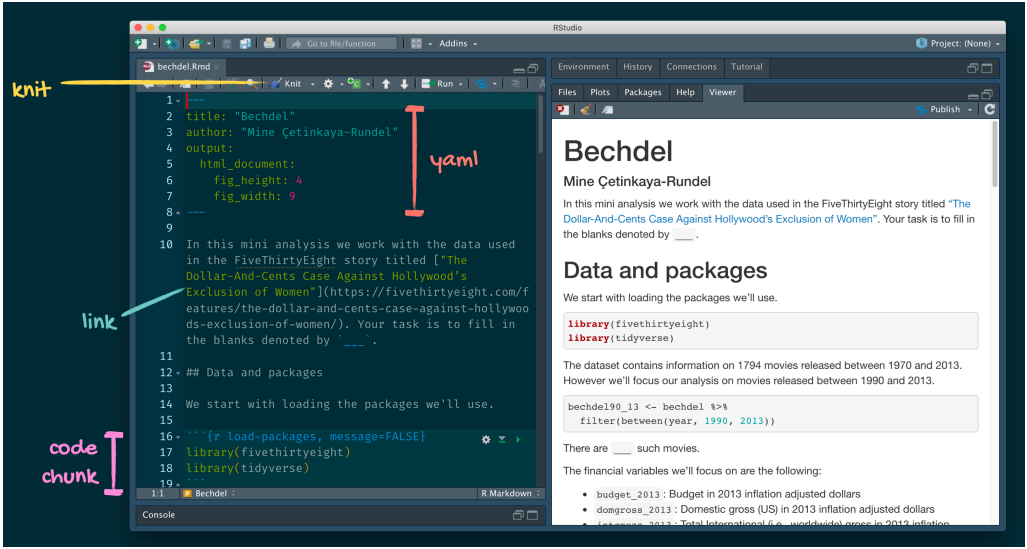
Object documentation can be accessed with ?

```
?mean
```

R Markdown

- rmarkdown and the various packages that support it enable R users to write their code and prose in reproducible computational documents.
- We will generally refer to R Markdown documents (with .Rmd extension), e.g. "Do this in your R Markdown document" and rarely discuss loading the rmarkdown package
- Fully reproducible reports – each time you knit the analysis is ran from the beginning
- Simple markdown syntax for text
- Code goes in chunks, defined by three backticks, narrative goes outside of chunks

Tour: R Markdown

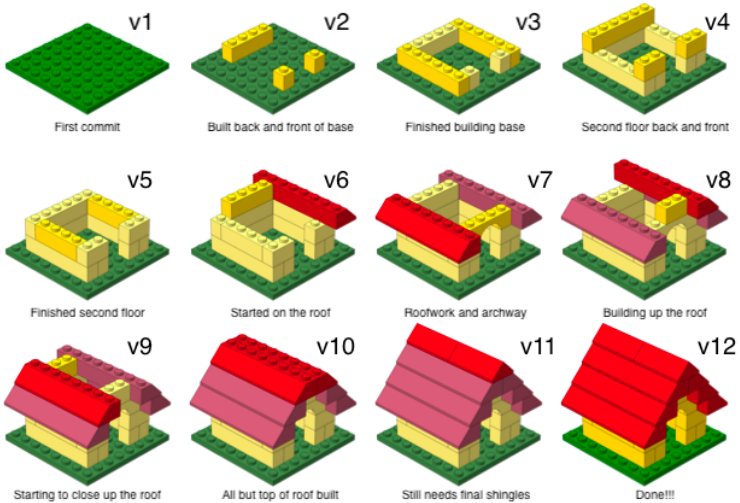


Version Control and Collaboration

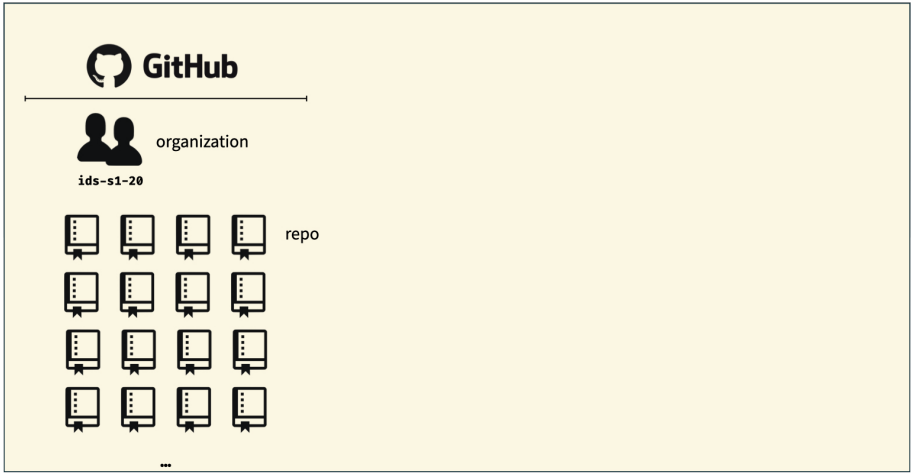
- Git is a version control system – like ?Track Changes? features from Microsoft Word, on steroids
- It's not the only version control system, but it's a very popular one

- GitHub is the home for your Git-based projects on the internet – like DropBox but much, much better
- We will use GitHub as a platform for web hosting and collaboration

Versioning with Human Readable Messages

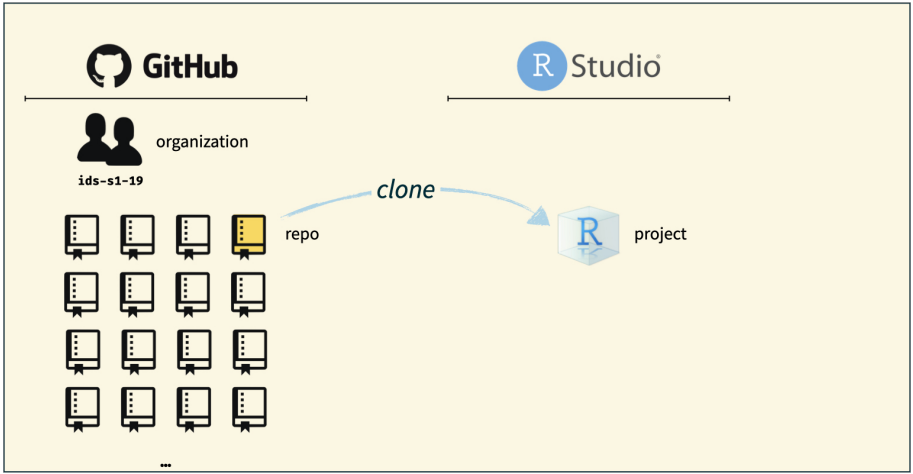


How Will We Use Git and GitHub?



How Will We Use Git and GitHub?

How Will We Use Git and GitHub?



How Will We Use Git and GitHub?