

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	B	C	D	E	F	G	H	I	J	K
1	id: country	country_code	vote	unrest	importhuman	Date	unrest	swestn	para	obstart
2	6 US	US	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
3	6 Canada	CA	no	1	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
4	6 Cuba	CU	yes	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
5	6 Dominican Republic	DO	abstain	2	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
6	6 Mexico	MX	yes	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
7	6 Guatemala	GT	no	1	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
8	6 Honduras	HN	yes	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
9	6 El Salvador	SV	abstain	1	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
10	6 Paraguay	PY	yes	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
11	6 Panama	PA	abstain	2	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
12	6 Colombia	CO	abstain	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
13	6 Venezuela, Bolivarian Republic of	VE	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
14	6 Ecuador	EC	yes	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
15	6 Peru	PE	yes	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
16	6 Brazil	BR	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
17	6 Bolivia (Plurinational State of)	BO	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
18	6 Paraguay	PY	abstain	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
19	6 Chile	CL	yes	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
20	6 Argentina	AR	abstain	1	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
21	6 Uruguay	UY	yes	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
22	6 UK & NI	GB	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
23	6 Netherlands	NL	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
24	6 Belgium	BE	no	1	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
25	6 Luxembourg	LU	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
26	6 Costa Rica	CR	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
27	6 Poland	PL	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
28	6 Czechoslovakia	CZ	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
29	6 Yugoslavia	YU	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
30	6 Greece	GR	no	1	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
31	6 Russian Federation	RU	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
32	6 Ukraine	UA	no	2	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
33	6 Belarus	BY	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
34	6 Norway	NO	no	1	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS
35	6 Denmark	DK	no	3	0	04/01/1946 R/1/1/07	0	0	0	DECLARATION OF HUMAN RIGHTS

(a) MS Excel

```
.mixed ln_w grade age c.age#c.age ttl_exp tenure c.tenure#c.tenure || id: tenure, cov(unstruct)
```

Performing EM optimization ...

Performing gradient-based optimization:

- Iteration 0: log likelihood = **-8985.3816**
- Iteration 1: log likelihood = **-8966.1706**
- Iteration 2: log likelihood = **-8965.8119**
- Iteration 3: log likelihood = **-8965.8119**

Computing standard errors ...

Mixed-effects ML regression
Group variable: **idcode**

	Number of obs	=	28,099
	Number of groups	=	4,697
	Obs per group:		
min	=	1	
avg	=	6.0	
max	=	15	

Wald ch2(6) = **6767.13**
Prob > ch2 = **0.0000**

Log likelihood = **-8965.8119**

ln_wage	Coefficient	Std. err.	z	P> z	[95% conf. interval]
grade	.06590318	.0017985	38.60	0.000	.0655264 - .0725372
age	.03221872	.0027988	11.53	0.000	.0267174 - .037657
c.age#c.age	-.0006574	.0000466	-14.09	0.000	-.0007488 - .000566

Command
.mixed ln_w grade age c.age#c.age ttl_exp tenure c.tenure#c.tenure || id: tenure, cov(unstruct)

Variables

Name	Label
idcode	NLS ID
year	Interview year
birth_yr	Birth year
age	Age in current year
race	Race
mp	1 if married, spouse present
net_mar	1 if never married
grado	Current grade completed
college	1 if college graduate
net_ress	1 if not SMSA
c_city	1 if central city
etc	1 if etc

Properties

Variables

Data

Frame default

Filename nlwork.dta

Label National Longitudinal Survey

Type

Format

Value label

Notes

Observations 28,099

Size 947,402

Memory 64MB

Sorted by idcode year

(b) Stata

R version 4.0.2 (2020-06-22) -- "Taking Off Again"
 Copyright (C) 2020 The R Foundation for Statistical Computing
 Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
 You are welcome to redistribute it under certain conditions.
 Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
 Type 'contributors()' for more information and
 'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
 'help.start()' for an HTML browser interface to help.
 Type 'q()' to quit R.

[R.app GUI 1.72 (7847) x86_64-apple-darwin17.0]

[History restored from /Users/mine/.Rapp.history]

> |

(c) RStudio

academy-launch - master - RStudio

avotes < Data Import Dataset Environment History Connections Git Tutorial

rcid	country	country_code	vote	session	importantvote	date	unres	amend	para	short
1	US	US	no	1	0	04/01/1946	RJ/1/107	0	D DECLA	
2	Canada	CA	no	1	0	04/01/1946	RJ/1/107	0	D DECLA	
3	Cuba	CU	yes	1	0	04/01/1946	RJ/1/107	0	D DECLA	
4	Dominican Republic	DO	abstain	1	0	04/01/1946	RJ/1/107	0	D DECLA	
5	Mexico	MX	yes	1	0	04/01/1946	RJ/1/107	0	D DECLA	
6	Guatemala	GT	no	1	0	04/01/1946	RJ/1/107	0	D DECLA	
7	Honduras	HN	yes	1	0	04/01/1946	RJ/1/107	0	D DECLA	
8	Bolivia	SV	abstain	1	0	04/01/1946	RJ/1/107	0	D DECLA	
9	Nicaragua	NI	yes	1	0	04/01/1946	RJ/1/107	0	D DECLA	
10	Panama	PA	abstain	1	0	04/01/1946	RJ/1/107	0	D DECLA	
11	Colombia	CO	abstain	1	0	04/01/1946	RJ/1/107	0	D DECLA	
12	Venezuela, Bolivarian Republic of	VE	no	1	0	04/01/1946	RJ/1/107	0	D DECLA	
13	Ecuador	EC	yes	1	0	04/01/1946	RJ/1/107	0	D DECLA	
14	Peru	PE	yes	1	0	04/01/1946	RJ/1/107	0	D DECLA	
15	Brazil	BR	no	1	0	04/01/1946	RJ/1/107	0	D DECLA	
16	Bolivia (Plurinational State of)	BO	no	1	0	04/01/1946	RJ/1/107	0	D DECLA	
17	Paraguay	PY	abstain	1	0	04/01/1946	RJ/1/107	0	D DECLA	
18	Chile	CL	yes	1	0	04/01/1946	RJ/1/107	0	D DECLA	
19	Argentina	AR	abstain	1	0	04/01/1946	RJ/1/107	0	D DECLA	
20	Uruguay	UY	yes	1	0	04/01/1946	RJ/1/107	0	D DECLA	

Showing 1 to 20 of 768,674 entries, 14 total columns

Console Terminal Jobs

~(Desktop/academy-launch) ~

R version 4.0.2 (2020-06-22) -- "Taking Off Again"
 Copyright (C) 2020 The R Foundation for Statistical Computing
 Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
 You are welcome to redistribute it under certain conditions.
 Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

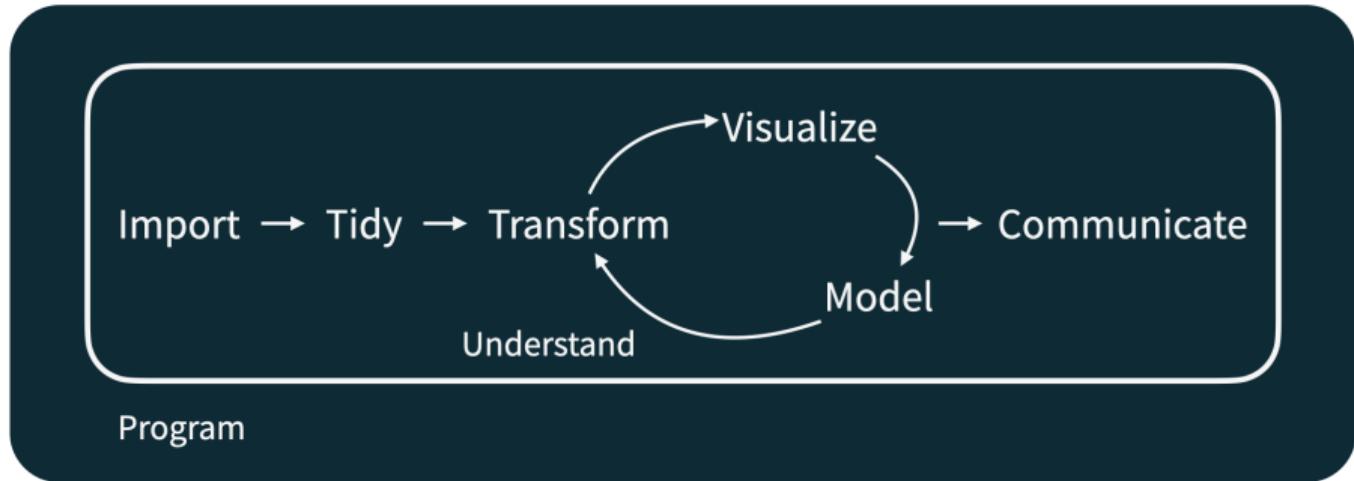
R is a collaborative project with many contributors.
 Type 'contributors()' for more information and
 'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
 'help.start()' for an HTML browser interface to help.
 Type 'q()' to quit R.

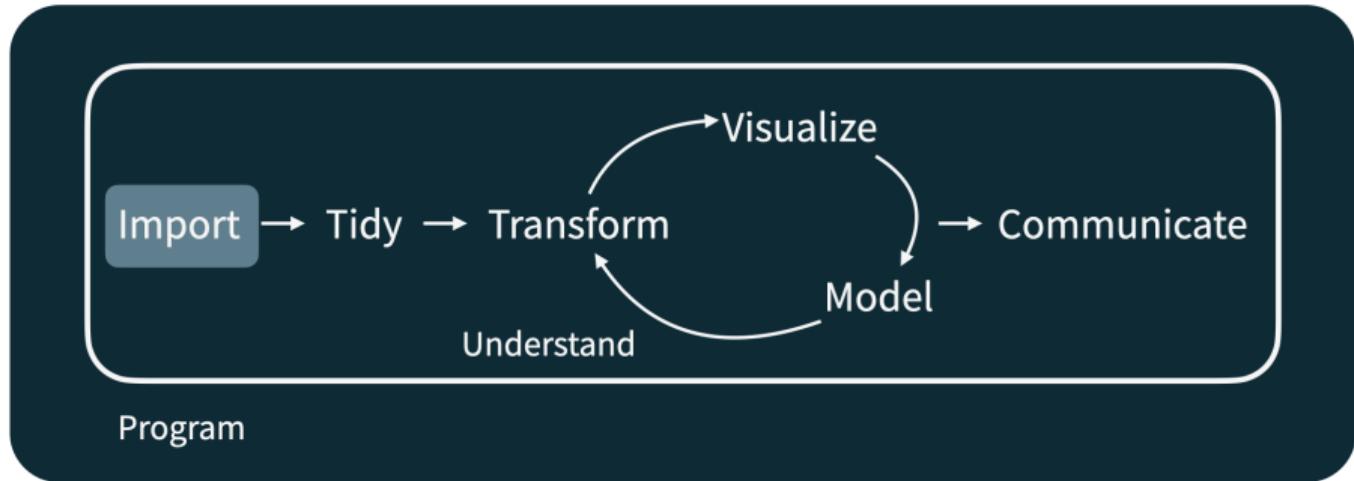
>

(d) RStudio

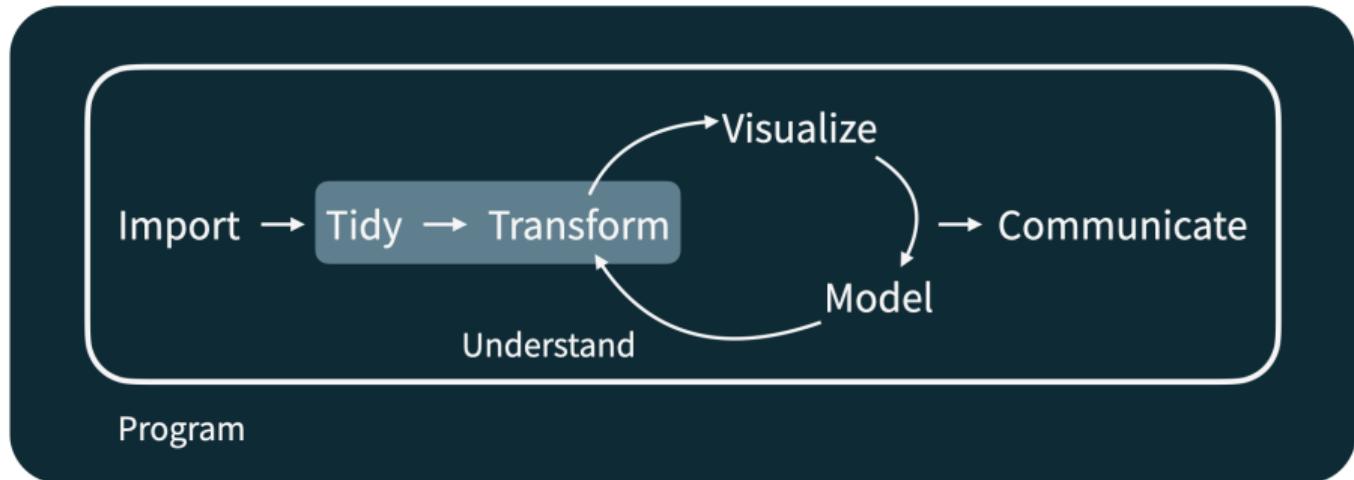
Data Science Life Cycle



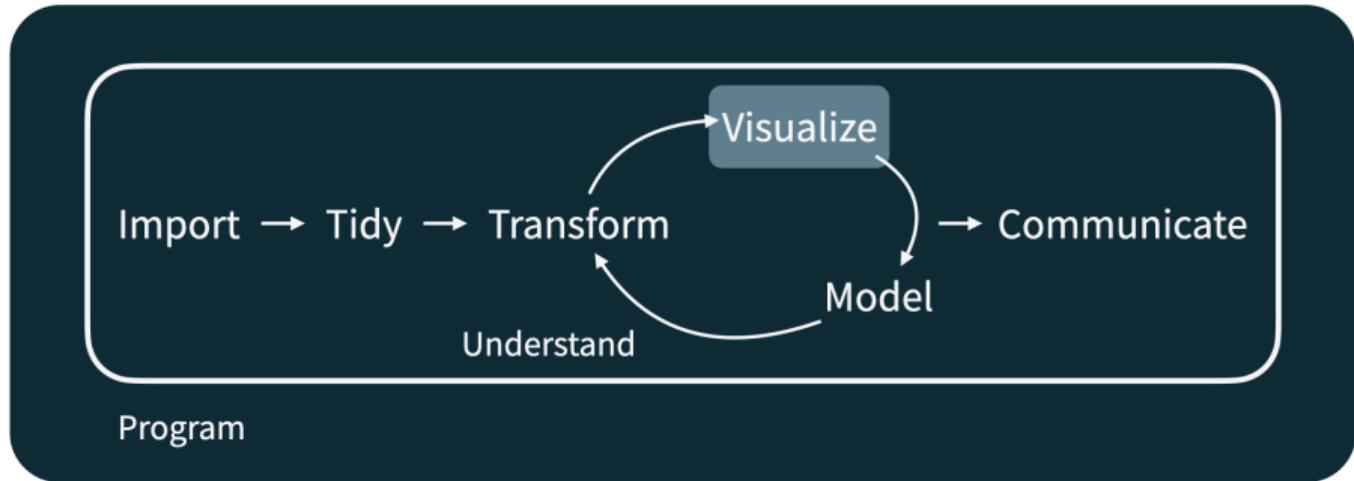
Data Science Life Cycle



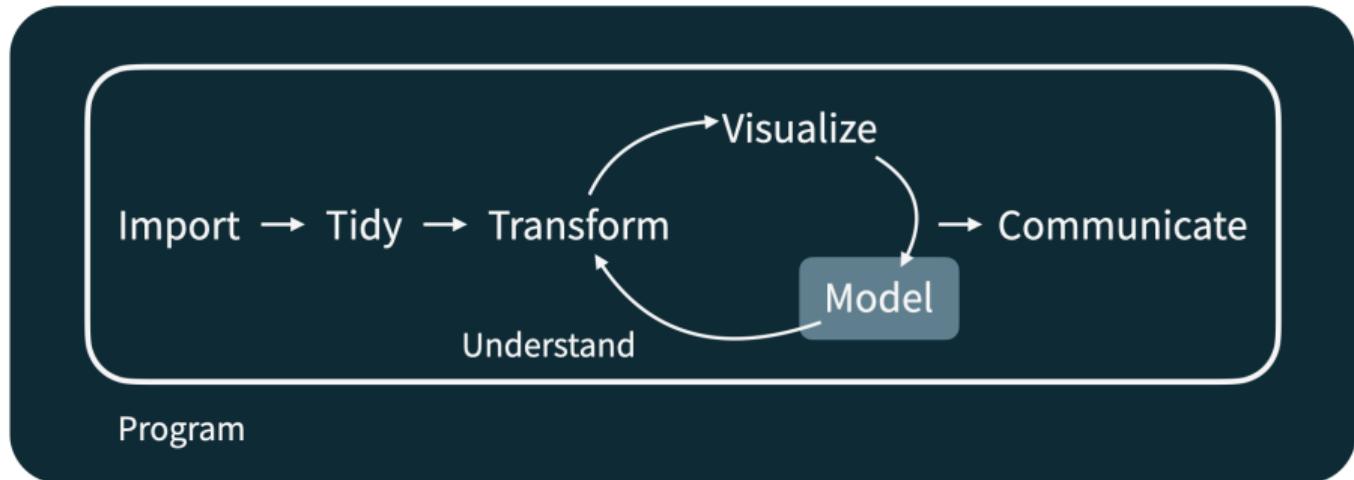
Data Science Life Cycle



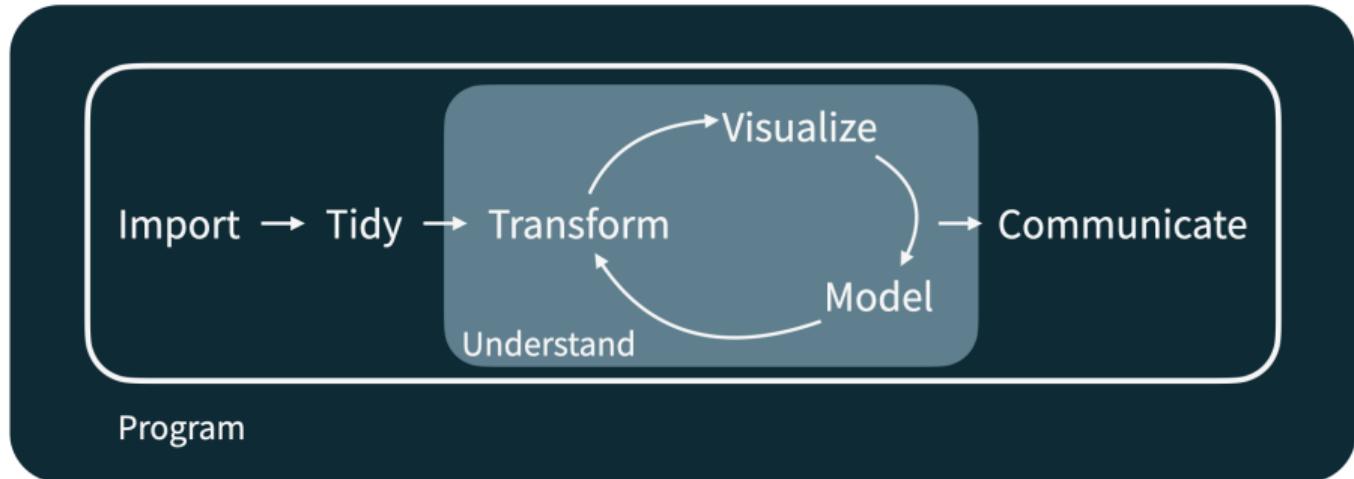
Data Science Life Cycle



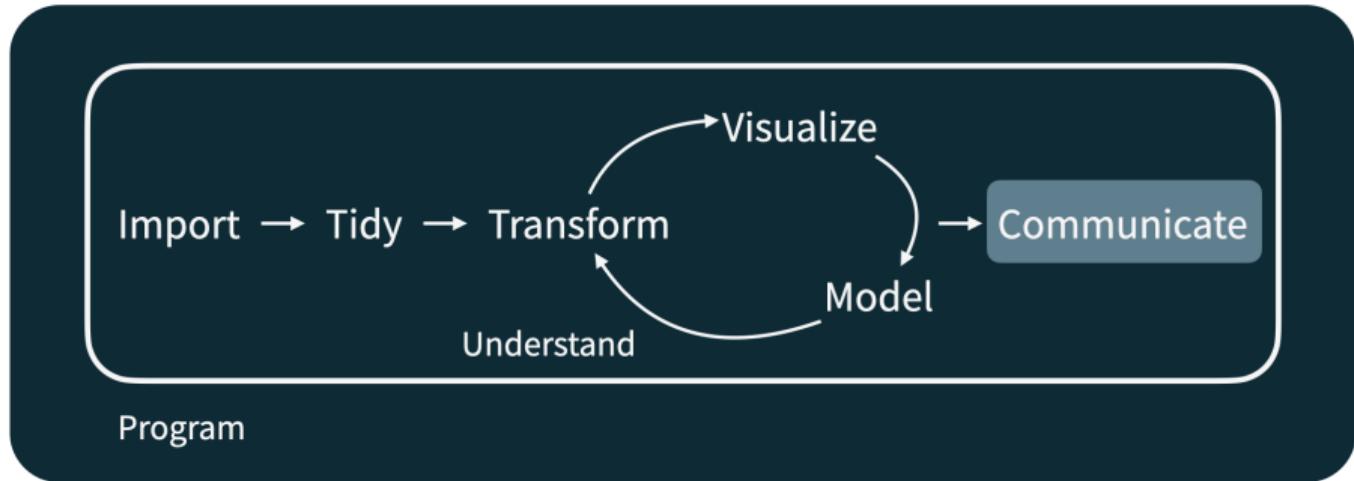
Data Science Life Cycle



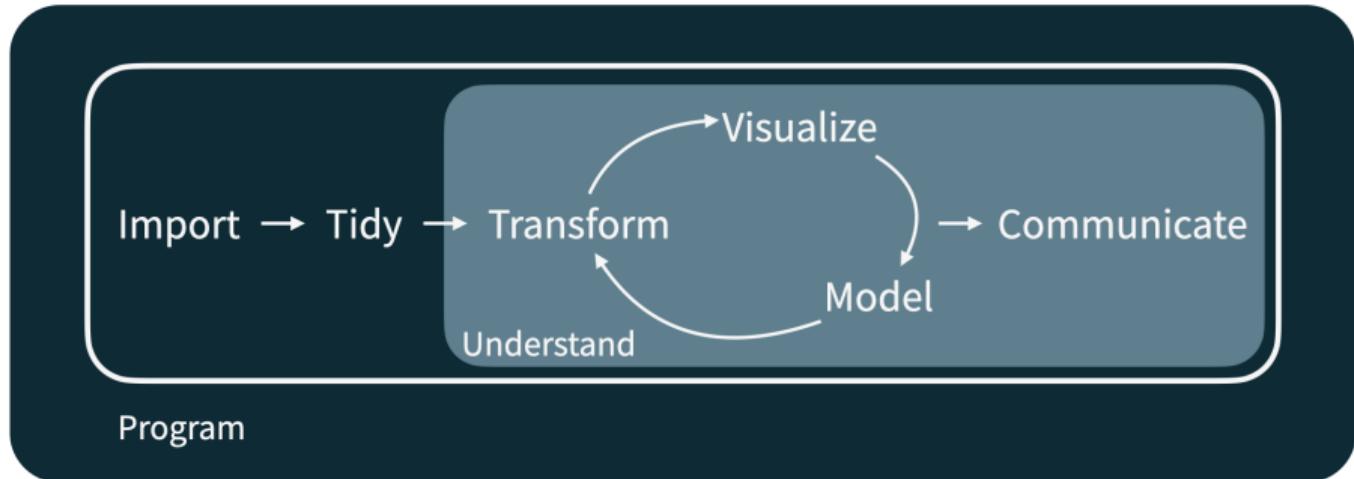
Data Science Life Cycle



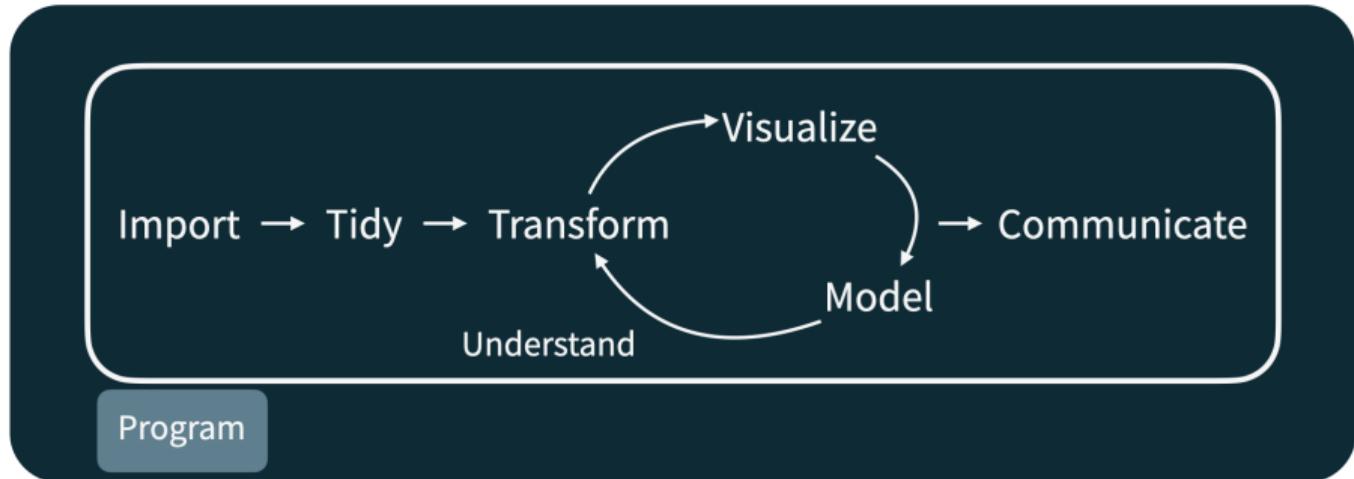
Data Science Life Cycle



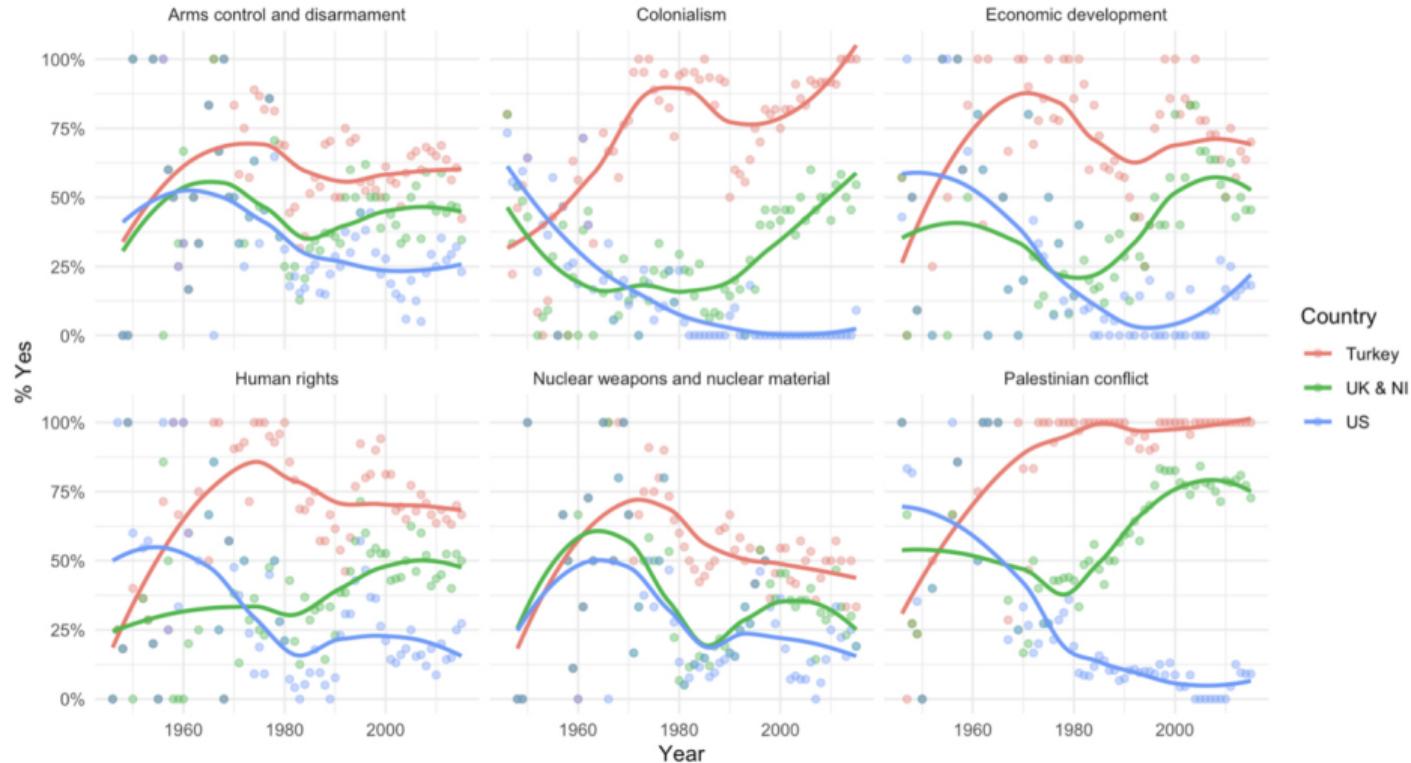
Data Science Life Cycle



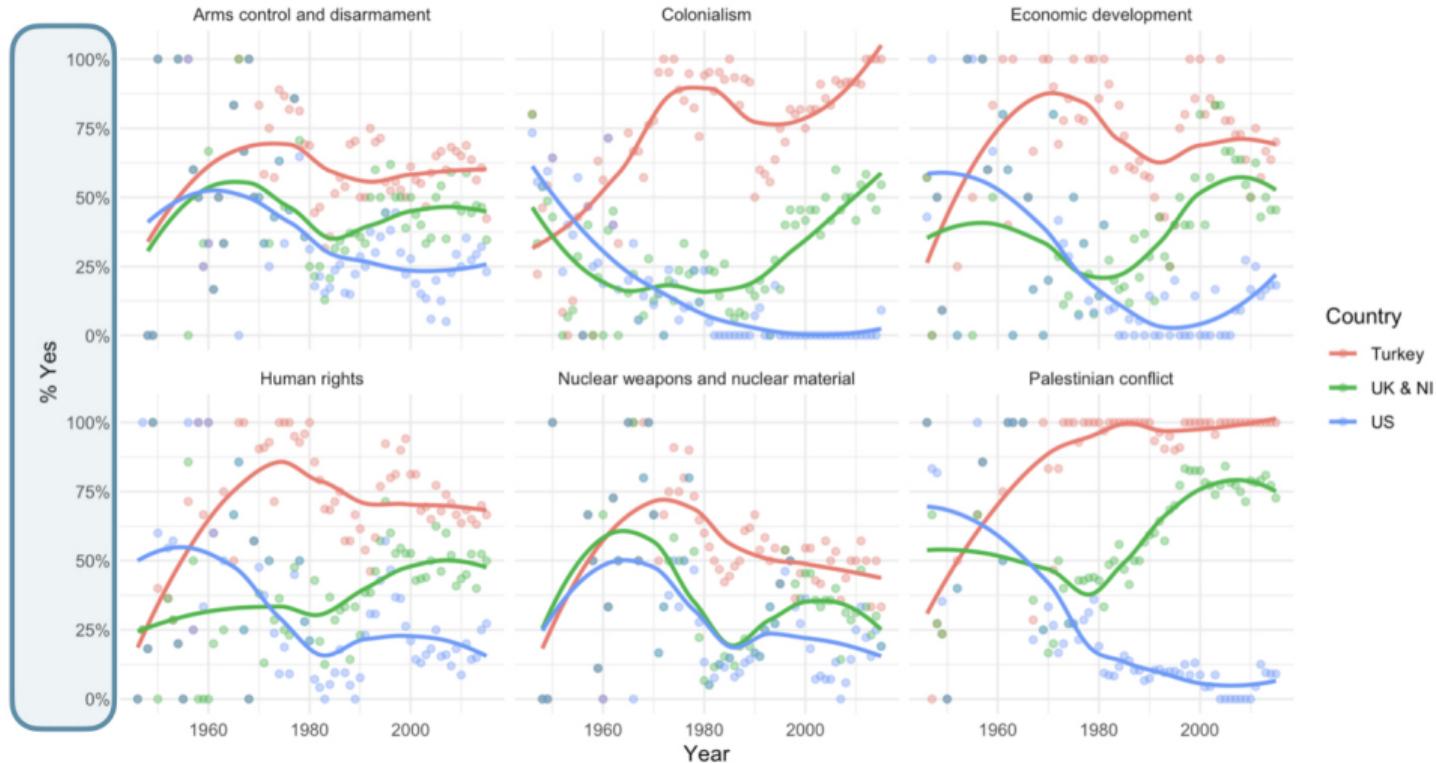
Data Science Life Cycle



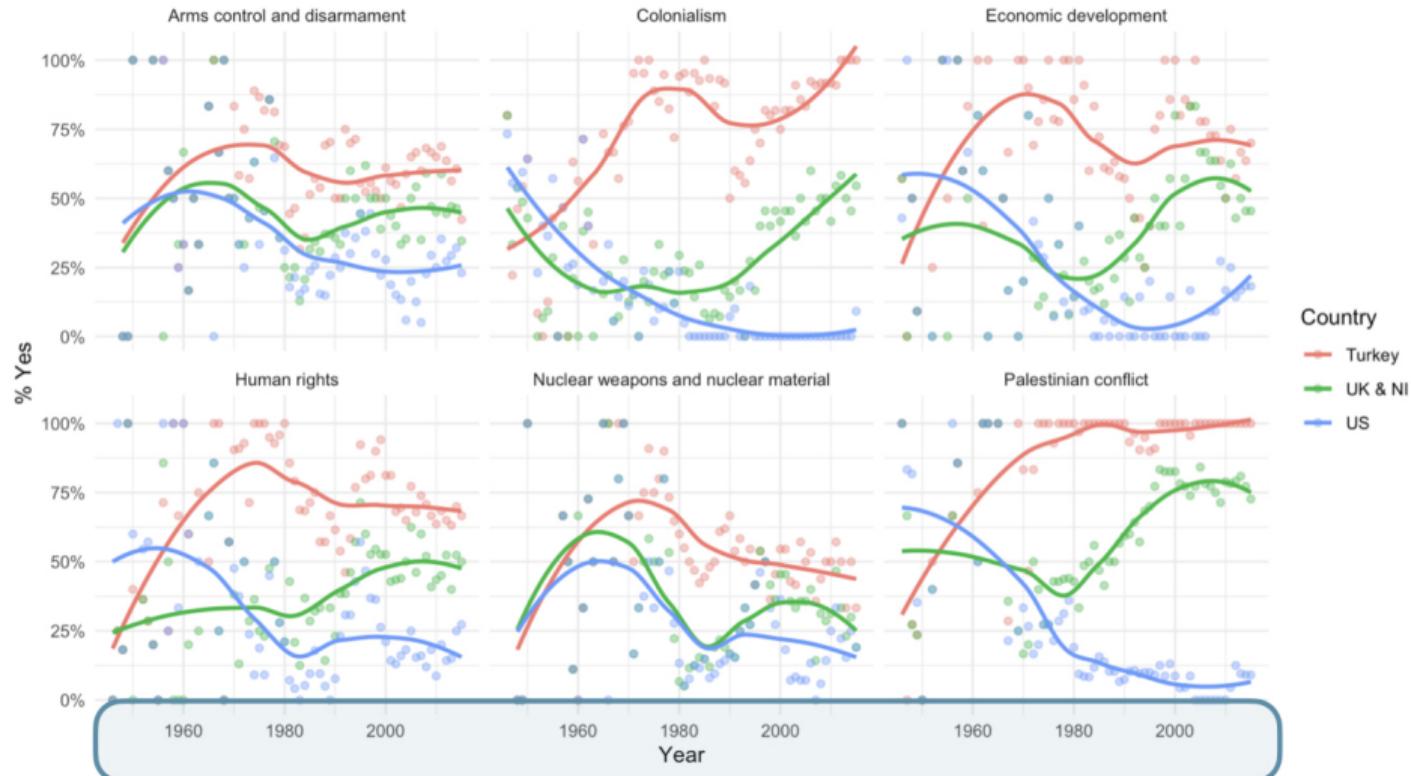
Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015



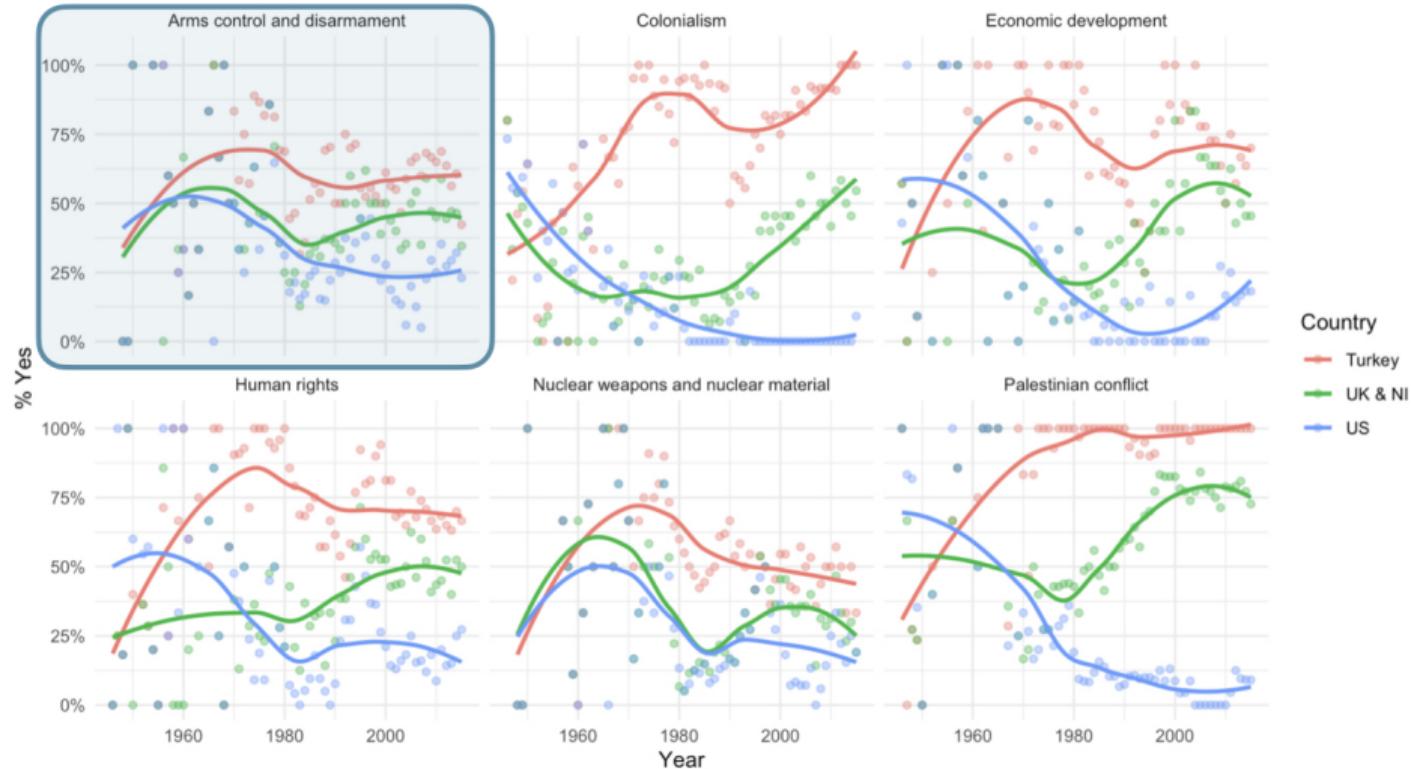
Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015



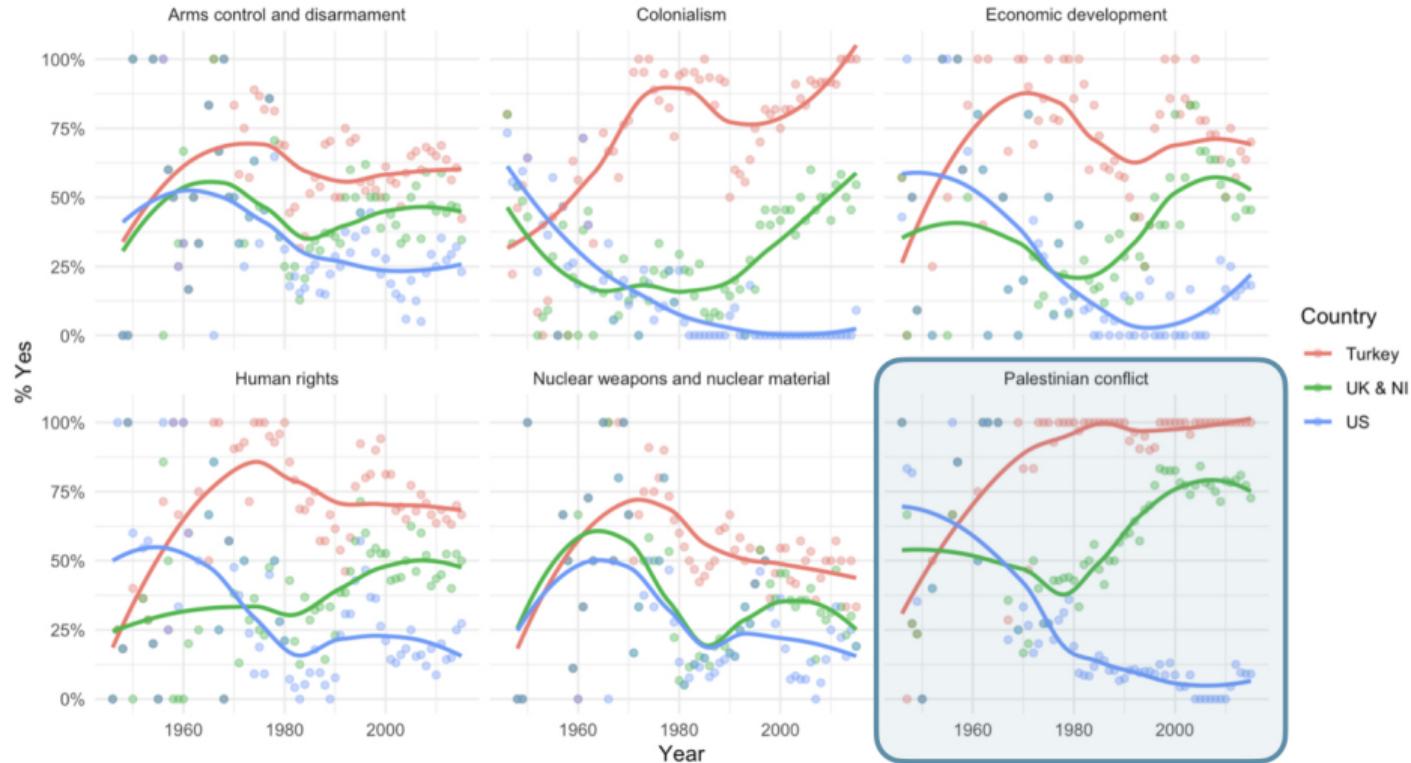
Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015



Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015



Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015



The screenshot shows three vertically stacked Jupyter Notebook cells, each displaying a table from a database. The top cell has a blue border and displays the 'un_votes' table. The middle cell has a pink border and displays the 'un_roll_calls' table. The bottom cell has a red border and displays the 'un_roll_call_issues' table. All three tables have columns: 'rcid', 'short_name', and 'issue'. The 'issue' column contains the value 'Palestinian conflict' for all rows.

	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
26	5070	me	Palestinian conflict

```
unvotes.Rmd x
Insert | Run | A
36 We can easily change which countries are being plotted by changing which
37 countries the code above `filter`'s for. Note that the country name should be
38 spelled and capitalized exactly the same way as it appears in the data. See
39 the [Appendix](#appendix) for a list of the countries in the data.
40
41 ~~~{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
42 un_votes %>%
43   mutate(
44     country =
45       case_when(
46         country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
47         country == "United States of America" ~ "US",
48         TRUE ~ country
49       )
50   ) %>%
51   inner_join(un_roll_calls, by = "rcid") %>%
52   inner_join(un_roll_call_issues, by = "rcid") %>%
53   filter(country %in% c("UK & NI", "US", "Turkey")) %>%
54   mutate(year = year(date)) %>%
55   group_by(country, year, issue) %>%
56   summarize(percent_yes = mean(vote == "yes")) %>%
57   ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
58   geom_point(alpha = 0.4) +
59   geom_smooth(method = "loess", se = FALSE) +
60   facet_wrap(~issue) +
61   scale_y_continuous(labels = percent) +
62   labs(
63     title = "Percentage of 'Yes' votes in the UN General Assembly",
64     subtitle = "1946 to 2015",
65     y = "% Yes",
66     x = "Year",
67     color = "Country"
68   ) +
69   theme_minimal()
70 ~~~
71
72
73 ## References {#references}
74
```

```
uvotes.Rmd x
Insert | Run | A
36 We can easily change which countries are being plotted by changing which
37 countries the code above `filter`'s for. Note that the country name should be
38 spelled and capitalized exactly the same way as it appears in the data. See
39 the [Appendix](#appendix) for a list of the countries in the data.
40
41 ~~~{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
42 un_votes %>%
43   mutate(
44     country =
45       case_when(
46         country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
47         country == "United States of America" ~ "US",
48         TRUE ~ country
49       )
50     ) %>%
51   inner_join(m_roll_calls, by = "rcid") %>%
52   inner_join(m_roll_call_issues, by = "rcid") %>%
53   filter(country %in% c("UK & NI", "US", "Turkey")) %>%
54   mutate(year = year(date)) %>%
55   group_by(country, year, issue) %>%
56   summarize(percent_yes = mean(vote == "yes")) %>%
57   ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
58   geom_point(alpha = 0.4) +
59   geom_smooth(method = "loess", se = FALSE) +
60   facet_wrap(~issue) +
61   scale_y_continuous(labels = percent) +
62   labs(
63     title = "Percentage of 'Yes' votes in the UN General Assembly",
64     subtitle = "1946 to 2015",
65     y = "% Yes",
66     x = "Year",
67     color = "Country"
68   ) +
69   theme_minimal()
70 ~
71
72
73 ## References {#references}
74
```

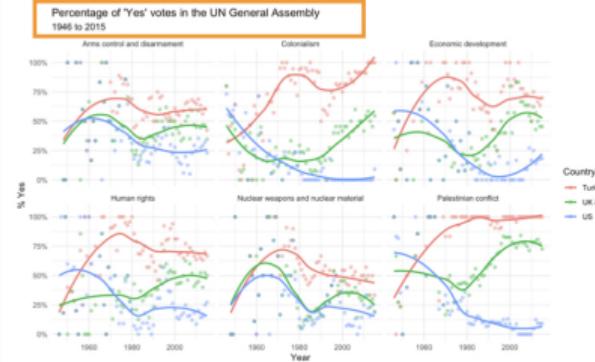
```
uvotes.Rmd x
File Edit View Insert Knit Run A
36 We can easily change which countries are being plotted by changing which
37 countries the code above `filter`'s for. Note that the country name should be
38 spelled and capitalized exactly the same way as it appears in the data. See
39 the [Appendix](#appendix) for a list of the countries in the data.
40
41 ~~~{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
42 un_votes %>%
43   mutate(
44     country =
45       case_when(
46         country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
47         country == "United States of America" ~ "US",
48         TRUE ~ country
49       )
50   ) %>%
51   inner_join(un_roll_calls, by = "rcid") %>%
52   inner_join(un_roll_call_issues, by = "rcid") %>%
53   filter(country %in% c("UK & NI", "US", "Turkey")) %>%
54   mutate(year = year(date)) %>%
55   group_by(country, year, issue) %>%
56   summarize(percent_yes = mean(vote == "yes")) %>%
57   ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
58   geom_point(alpha = 0.4) +
59   geom_smooth(method = "loess", se = FALSE) +
60   facet_wrap(~issue) +
61   scale_y_continuous(labels = percent) +
62   labs(
63     title = "Percentage of 'Yes' votes in the UN General Assembly",
64     subtitle = "1946 to 2015",
65     y = "% Yes",
66     x = "Year",
67     color = "Country"
68   ) +
69   theme_minimal()
70 ~~~
71
72
73 ## References {#references}
74
```

```
uvotes.Rmd x
Insert | Run | A
36 We can easily change which countries are being plotted by changing which
37 countries the code above `filter`'s for. Note that the country name should be
38 spelled and capitalized exactly the same way as it appears in the data. See
39 the [Appendix](#appendix) for a list of the countries in the data.
40
41 ~~~{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
42 un_votes %>%
43   mutate(
44     country =
45     case_when(
46       country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
47       country == "United States of America" ~ "US",
48       TRUE ~ country
49     )
50   ) %>%
51   inner_join(un_roll_calls, by = "rcid") %>%
52   inner_join(un_roll_call_issues, by = "rcid") %>%
53   filter(country %in% c("UK & NI", "US", "Turkey")) %>%
54   mutate(year = year(date)) %>%
55   group_by(country, year, issue) %>%
56   summarize(percent_yes = mean(vote == "yes")) %>%
57   ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
58   geom_point(alpha = 0.4) +
59   geom_smooth(method = "loess", se = FALSE) +
60   facet_wrap(~issue) +
61   scale_y_continuous(labels = percent) +
62   labs(
63     title = "Percentage of 'Yes' votes in the UN General Assembly",
64     subtitle = "1946 to 2015",
65     y = "% Yes",
66     x = "Year",
67     color = "Country"
68   ) +
69   theme_minimal()
70 ~
71
72
73 ## References {#references}
74
```

```

36 We can easily change which countries are being plotted by changing which
37 countries the code above `filter`'s for. Note that the country name should be
38 spelled and capitalized exactly the same way as it appears in the data. See
39 the [Appendix](#appendix) for a list of the countries in the data.
40
41 ``{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
42 un_votes %>%
43   mutate(
44     country =
45       case_when(
46         country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
47         country == "United States of America" ~ "US",
48         TRUE ~ country
49       )
50     ) %>%
51   inner_join(un_roll_calls, by = "rcid") %>%
52   inner_join(un_roll_call_issues, by = "rcid") %>%
53   filter(country %in% c("UK & NI", "US", "Turkey")) %>%
54   mutate(year = year(date)) %>%
55   group_by(country, year, issue) %>%
56   summarize(percent_yes = mean(vote == "yes")) %>%
57   ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
58   geom_point(alpha = 0.4) +
59   geom_smooth(method = "loess", se = FALSE) +
60   facet_wrap(~issue) +
61   scale_y_continuous(labels = percent) +
62   labs(
63     title = "Percentage of 'Yes' votes in the UN General Assembly",
64     subtitle = "1946 to 2015",
65     y = "% Yes",
66     x = "Year",
67     color = "Country"
68   ) +
69   theme_minimal()
70 ```
71
72
73 ## References {#references}
74

```



The screenshot shows an RStudio interface with a Shiny application running in the background. The Shiny app has a sidebar with 'Introduction', 'UN voting patterns', 'References', and 'Appendix' options. The main content area displays the title 'UN Votes' by Mine Çetinkaya-Rundel, dated 2020-08-18, with a section titled 'Introduction' and a note about the analysis focus.

Code in the Shiny App:

```

1: ---
2: title: "UN Votes"
3: author: "Mine Çetinkaya-Rundel"
4: date: "r Sys.Date()"
5: output:
6:   html_document:
7:     toc: yes
8:     toc_float: yes
9: ---
10: 
11: ## Introduction
12: 
13: How do various countries vote in the United Nations General Assembly, how have
14: their voting patterns evolved throughout time, and how similarly or differently
15: do they view certain issues? Answering these questions (at a high level) is the
16: focus of this analysis.
17: 
18: We will use the **tidyverse**, **lubridate**, and **scales** packages for the
19: data wrangling and visualization, and the **DT** package for interactive display
20: of tabular output. The data we're using come from the **unvotes** package.
21: 
22: ```{r load-packages, warning=FALSE, message=FALSE}
23: library(tidyverse)
24: library(lubridate)
25: library(scales)
26: library(DT)
27: library(unvotes)
28: ```

30: ## UN voting patterns {#voting}
31: 
32: Let's create a data visualization that displays how the voting record of the
33: UK & NI changed over time on a variety of issues, and compares it to two other countries:
34: US and Turkey.
35: 
36: We can easily change which countries are being plotted by changing which
37: countries the code above 'filter's for. Note that the country name should be
38: spelled and capitalized exactly the same way as it appears in the data. See
39: the [Appendix](#appendix) for a list of the countries in the data.
40: 
41: ```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
42: un_votes %>%
43:   mutate(
44:     country =

```

Code in the RStudio Console:

```

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) %>

```

The screenshot shows a Shiny application running in a web browser. The left pane displays the R code for the application, and the right pane shows the resulting user interface.

R Code (Left):

```
2 title: "UN Votes"
3 author: "Mine Çetinkaya-Rundel"
4 date: r Sys.Date()
5 output:
6   html_document:
7     toc: yes
8     toc_float: yes
9 ...
11 ## Introduction
12
13 How do various countries vote in the United Nations General Assembly, how have
14 their voting patterns evolved throughout time, and how similarly or differently do
15 they view certain issues? Answering these questions (at a high level) is the focus of this analysis.
16
17 We will use the tidyverse, lubridate, and scales packages for the data wrangling and visualization, and the DT package for interactive display
18
19 of tabular output. The data we're using come from the unvotes package.
20
21
22 ##(r load-packages, warning=FALSE, message=FALSE)
23 library(tidyverse)
24 library(lubridate)
25 library(scales)
26 library(DT)
27 library(unvotes)
28 ...
29
30 ## UN voting patterns #voting
31
32 Let's create a data visualization that displays how the voting record of the
33 UK & NI changed over time on a variety of issues, and compares it
34 to two other countries: US and Turkey.
35
36 We can easily change which countries are being plotted by changing which
37 countries the code above filters for. Note that the country name should be
38 spelled and capitalized exactly the same way as it appears in the data. See
39 the [Appendix](#appendix) for a list of the countries in the data.
40
41 ##(r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE)
42 un_votes %>%
43   mutate(
44     country =
45       country =
```

UI (Right):

```
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer
Publish
```

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 =
      country =
        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(issue = year(date) - issue) %>%
    group_by(country, year(issue)) %>%
```

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 → R
- ▶ Literate programming (code, narrative, output in one place) → R Markdown
- ▶ Version control → 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

The screenshot illustrates the RStudio interface with several annotations:

- Data Viewer:** Shows a table of penguin data with columns: species, Island, bill_length_mm, bill_depth_mm, flipper_length_mm, and body_mass_g.
- Environment:** Shows the variable `x` assigned the value 2.
- Console:** Displays R code and its output. Annotations include:
 - arithmetic:** Points to the expression `> 2 + 2`.
 - load package:** Points to the command `> library(palmerpenguins)`.
 - view data:** Points to the command `> View(penguins)`.
 - get help:** Points to the command `> ?mean`.
 - Object assignment:** Points to the assignment `> x <- 2`.
 - access variable:** Points to the command `> penguins$flipper_length_mm`.
 - use function:** Points to the command `> mean(penguins$flipper_length_mm)`.
 - mean output:** Points to the output `[1] 181 186 195 NA 193 190 181 195 193 190 186 180 182 191`.
 - mean function call:** Points to the command `> mean(penguins$flipper_length_mm, na.rm = TRUE)`.
 - mean output (continued):** Points to the output `[1] 200.9152`.
- R Documentation:** Shows the `mean` function documentation, including the description, usage, arguments, examples, and notes.

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

The screenshot shows the RStudio interface with an R Markdown file named "bechdel.Rmd" open in the left pane. The code is as follows:

```
knit
1 ---  
2 title: "Bechdel"  
3 author: "Mine Çetinkaya-Rundel"  
4 output:  
5   html_document:  
6     fig_height: 4  
7     fig_width: 9  
8 ---  
9  
10 In this mini analysis we work with the data used  
11 in the FiveThirtyEight story titled ["The  
12 Dollar-And-Cents Case Against Hollywood's  
13 Exclusion of Women"](https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/). Your task is to fill in  
14 the blanks denoted by _____.  
15  
16 ## Data and packages  
17  
18 We start with loading the packages we'll use.  
19  
20 library(fivethirtyeight)  
21 library(tidyverse)  
22  
23
```

Annotations on the left side of the code editor:

- "knit" is highlighted in yellow.
- "yaml" is highlighted in red.
- "link" is highlighted in green.
- "code chunk" is highlighted in pink.

The right pane shows the rendered HTML output:

Bechdel

Mine Çetinkaya-Rundel

In this mini analysis we work with the data used in the FiveThirtyEight story titled ["The Dollar-And-Cents Case Against Hollywood's Exclusion of Women"](#). Your task is to fill in the blanks denoted by _____.

Data and packages

We start with loading the packages we'll use.

```
library(fivethirtyeight)  
library(tidyverse)
```

The dataset contains information on 1794 movies released between 1970 and 2013. However we'll focus our analysis on movies released between 1990 and 2013.

```
bechdel90_13 <- bechdel %>%  
  filter(between(year, 1990, 2013))
```

There are ____ such movies.

The financial variables we'll focus on are the following:

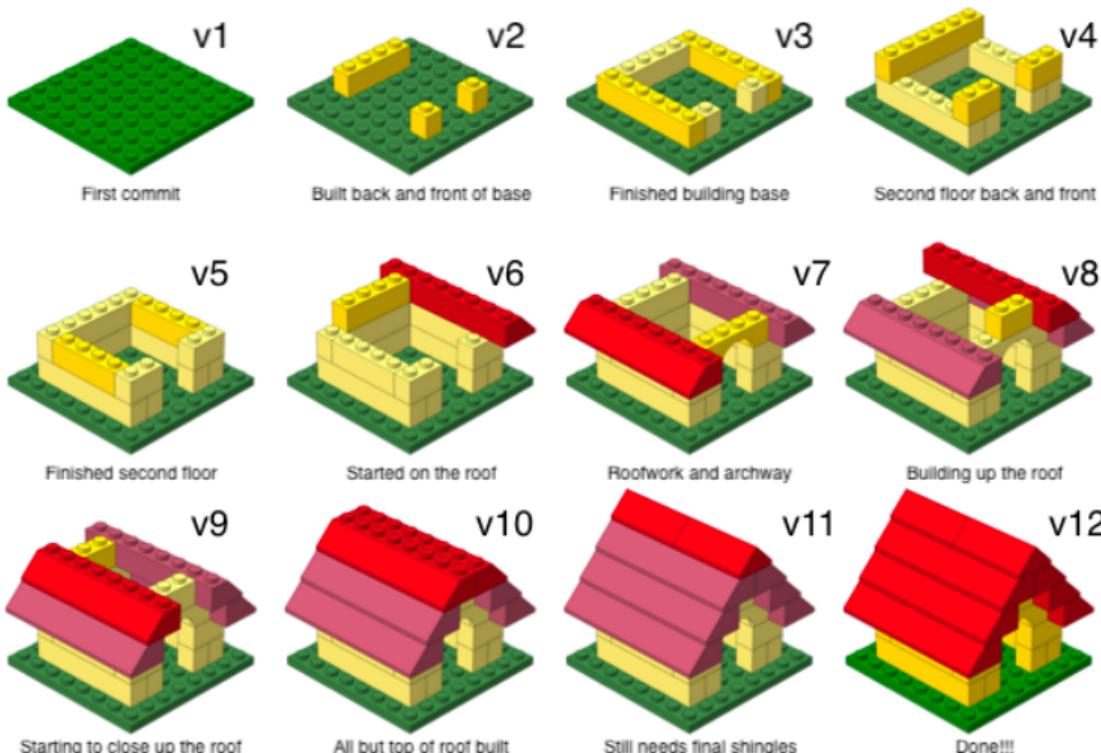
- budget_2013 : Budget in 2013 inflation adjusted dollars
- domgross_2013 : Domestic gross (US) in 2013 inflation adjusted dollars
- intgross_2013 : Total International (i.e., worldwide) gross in 2013 inflation

Version Control and Collaboration

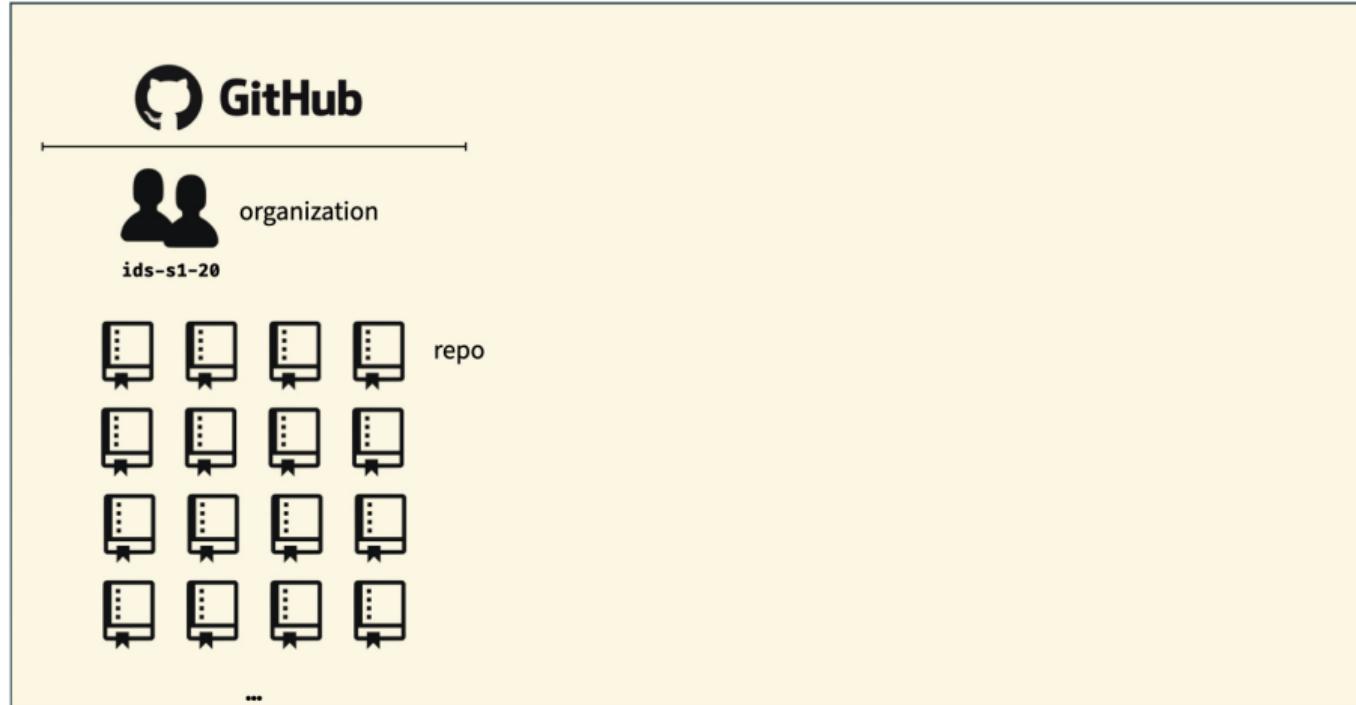


- ▶ Git is a version control system – like ?Track Changes? features from Microsoft Word, on steroids
- ▶ It's not the only version control system, but it's a very popular one
- ▶ GitHub is the home for your Git-based projects on the internet – like DropBox but much, much better
- ▶ We will use GitHub as a platform for web hosting and collaboration

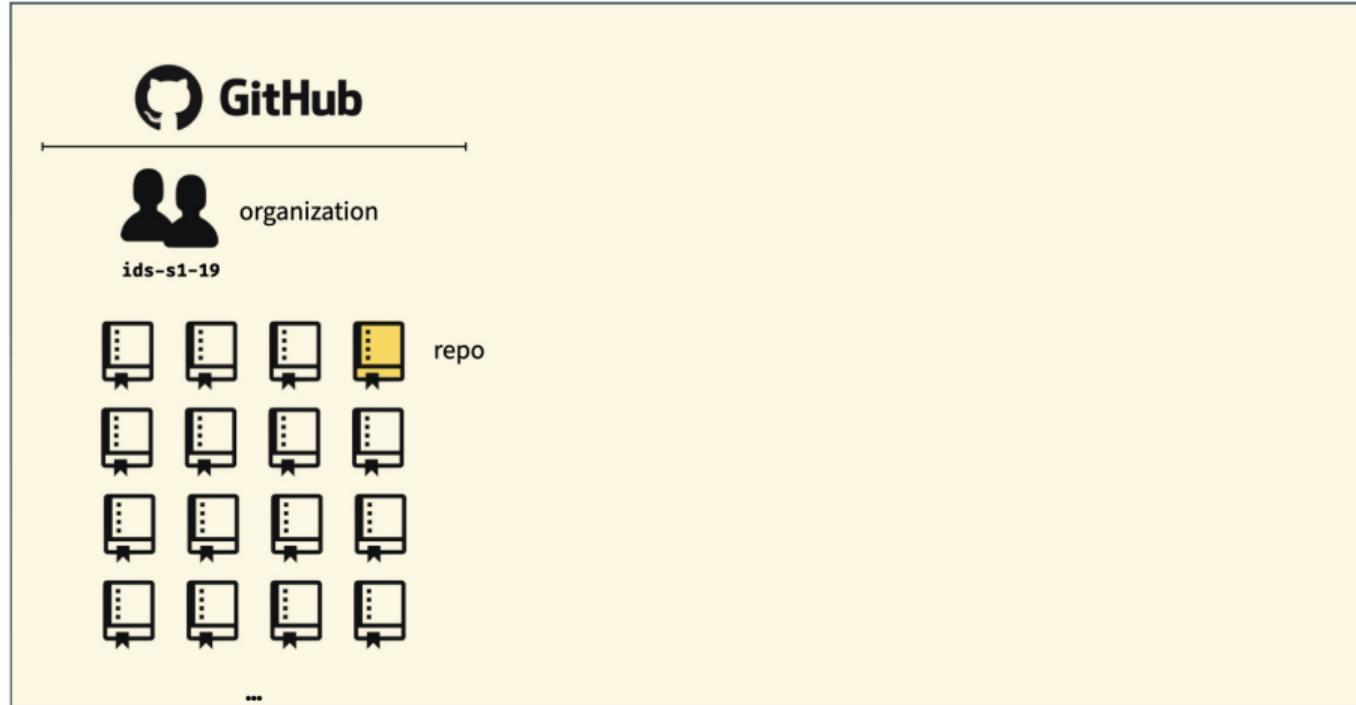
Versioning with Human Readable Messages



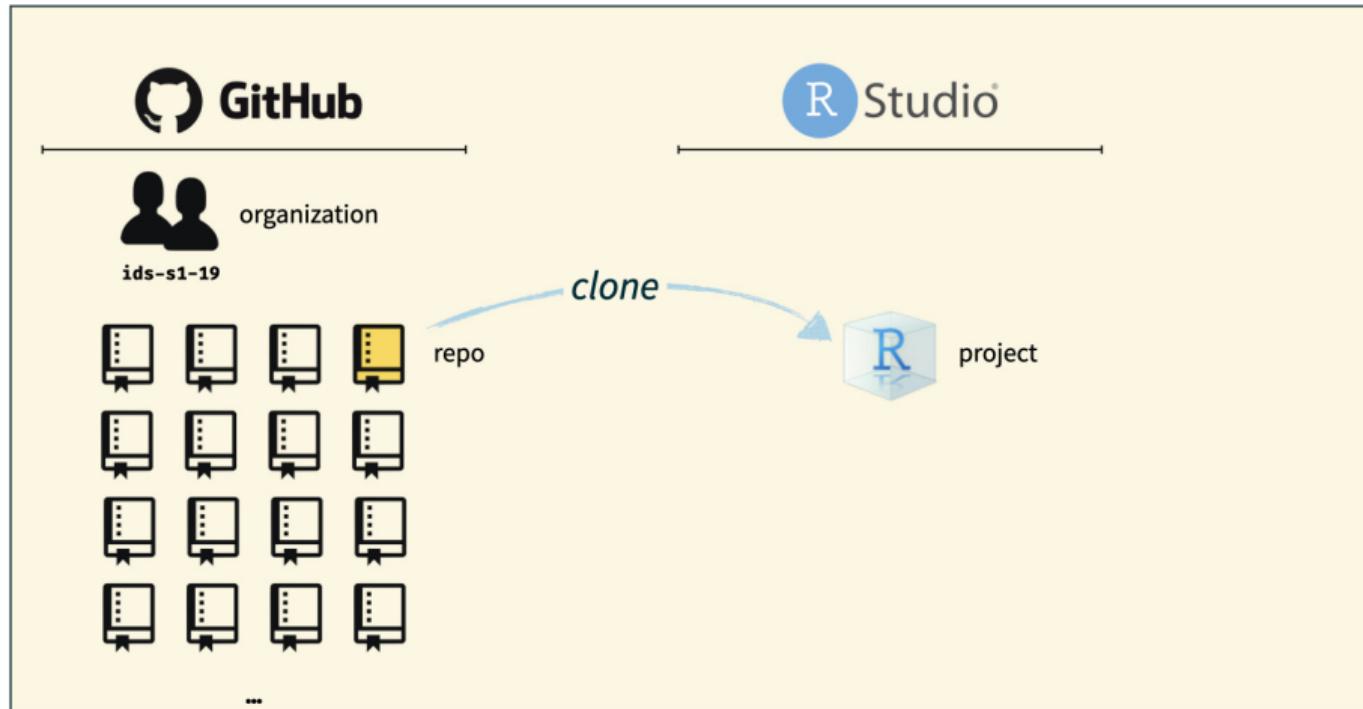
How Will We Use Git and GitHub?



How Will We Use Git and GitHub?



How Will We Use Git and GitHub?



How Will We Use Git and GitHub?

