

Welcome!

Data Science in a Box
datasciencebox.org

Modified by Tyler George



Hello world!



datasciencebox.org

Data science

- 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.



Course FAQ

Q - What data science background does this course assume?

A - None.

Q - Is this an intro stat course?

A - While statistics \neq data science, they are very closely related and have tremendous of overlap. Hence, this course is a great way to get started with statistics. However this course is *not* your typical high school statistics course.

Q - Will we be doing computing?

A - Yes.



Course FAQ

Q - Is this an intro CS course?

A - No, but many themes are shared.

Q - What computing language will we learn?

A - R.

Q: Why not language X?

A: We can discuss that over ☕.



Software



AutoSave OFF

unvotes — Saved to my Mac

Home Insert Page Layout Formulas Data Review View Table

F17 X ✓ fx | 0

	A	B	C	D	E	F	G	H	I	J	K
1	rcid	country	country_code	vote	session	importantvote	date	unres	amend	para	short
2	6	US	US	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
3	6	Canada	CA	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
4	6	Cuba	CU	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
5	6	Dominican Republic	DO	abstain	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
6	6	Mexico	MX	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
7	6	Guatemala	GT	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
8	6	Honduras	HN	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
9	6	El Salvador	SV	abstain	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
10	6	Nicaragua	NI	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
11	6	Panama	PA	abstain	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
12	6	Colombia	CO	abstain	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
13	6	Venezuela, Bolivarian Republic of	VE	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
14	6	Ecuador	EC	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
15	6	Peru	PE	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
16	6	Brazil	BR	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
17	6	Bolivia (Plurinational State of)	BO	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
18	6	Paraguay	PY	abstain	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
19	6	Chile	CL	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
20	6	Argentina	AR	abstain	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
21	6	Uruguay	UY	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
22	6	UK & NI	GB	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
23	6	Netherlands	NL	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
24	6	Belgium	BE	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
25	6	Luxembourg	LU	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
26	6	France	FR	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
27	6	Poland	PL	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
28	6	Czechoslovakia	CS	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
29	6	Yugoslavia	YU	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
30	6	Greece	GR	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
31	6	Russian Federation	RU	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
32	6	Ukraine	UA	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
33	6	Belarus	BY	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
34	6	Norway	NO	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
35	6	Denmark	DK	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS

R Console

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]

> |

academy-launch - master - RStudio

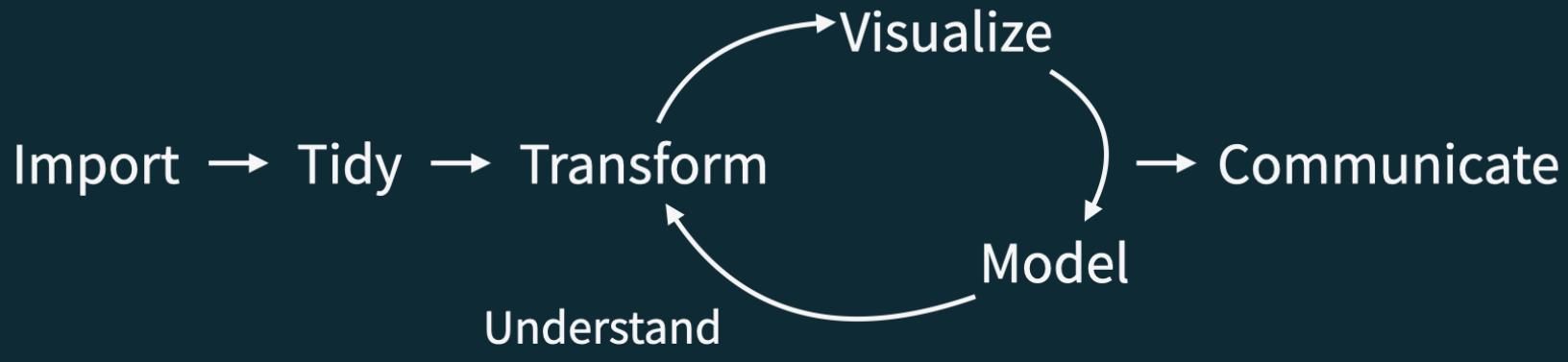
The screenshot shows the RStudio interface with the following components:

- Data View:** A grid view of the 'unvotes' dataset. The columns are: rcid, country, country_code, vote, session, importantvote, date, unres, amend, para, and short. The data shows various countries and their voting records.
- Environment View:** Shows the global environment with one object: 'unvotes' (768,674 obs. of 14 variables).
- Console View:** Displays the R startup message and the R command prompt (>).
- File View:** Shows the project directory structure: Home > Desktop > academy-launch. It includes files like .gitignore, academy-launch.Rproj, data, and unvotes.Rmd.

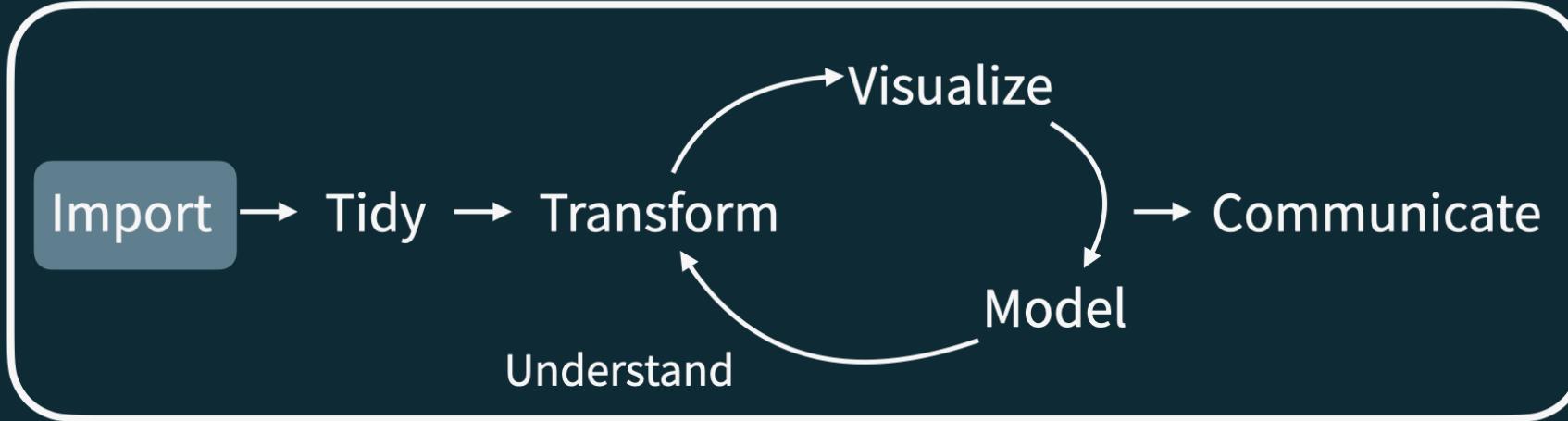


Data science life cycle

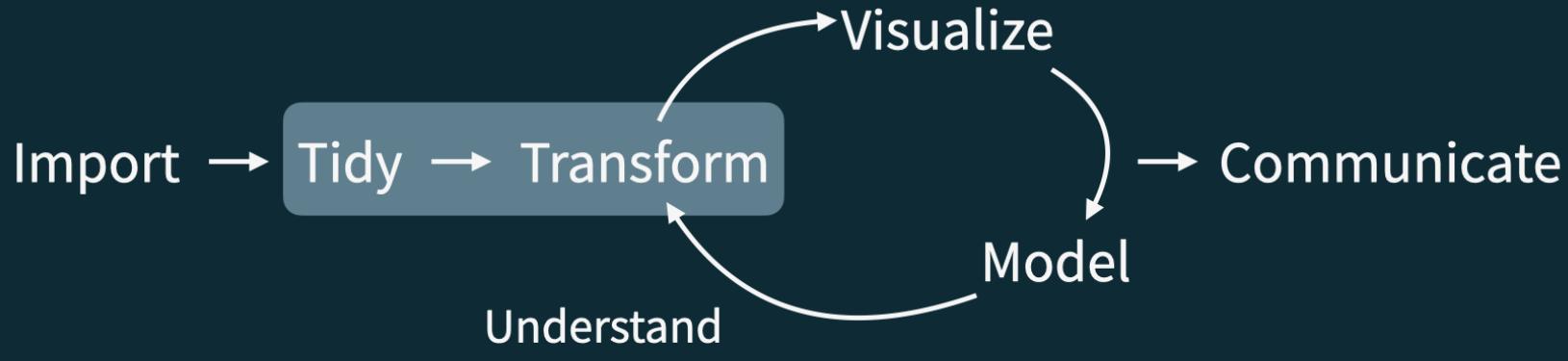




Program

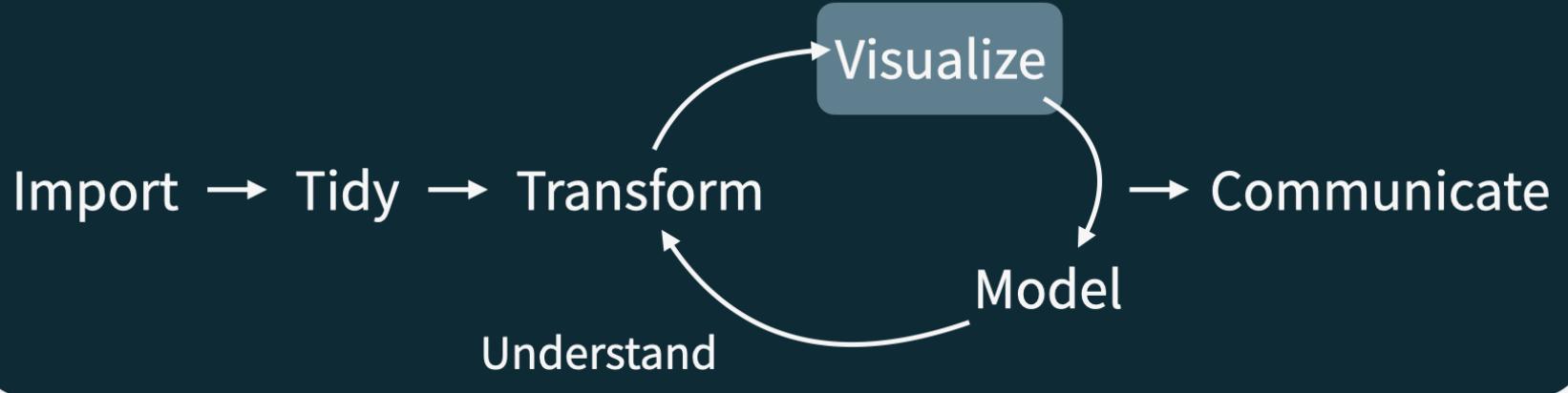


Program



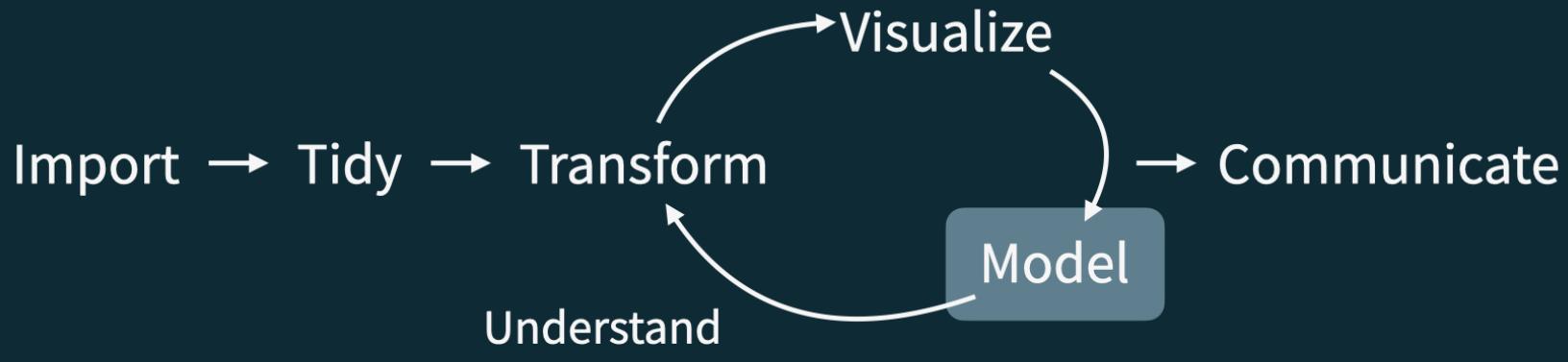
Program



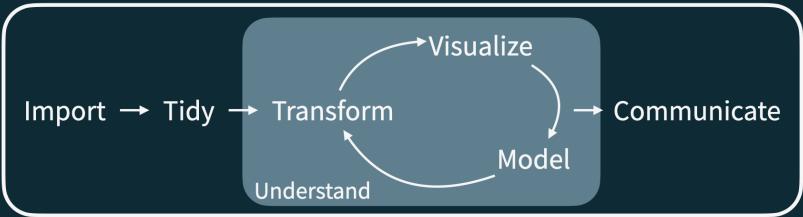


Program

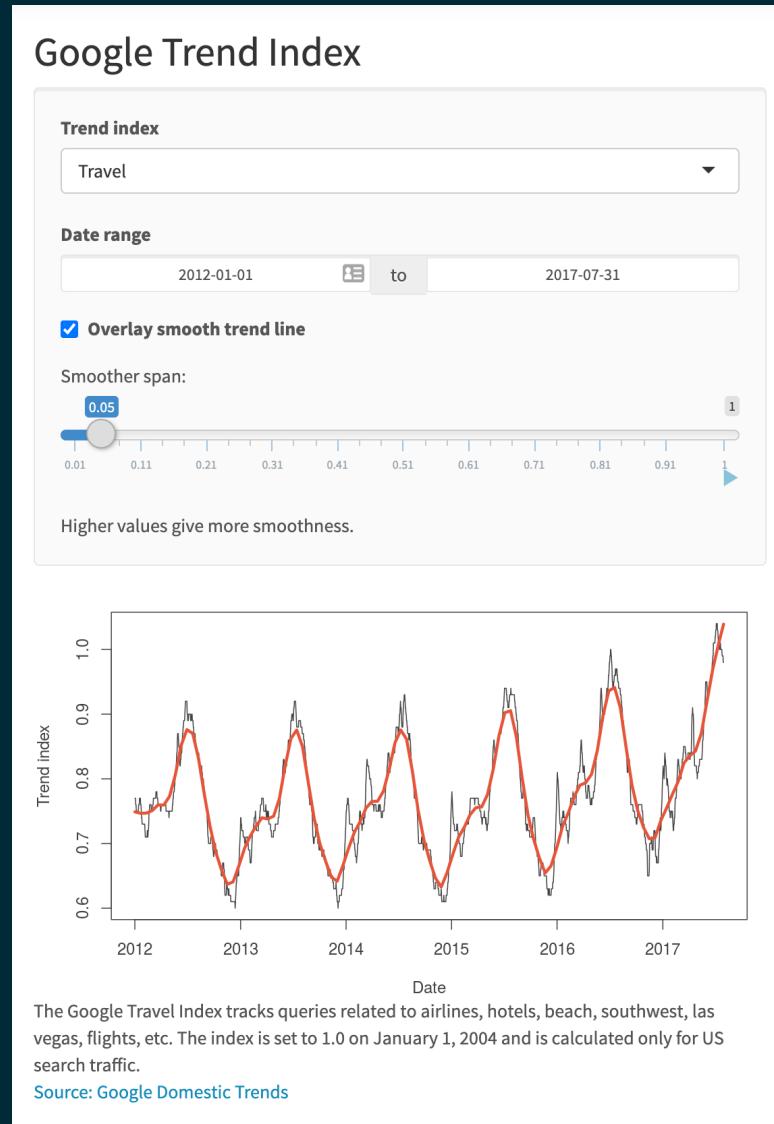
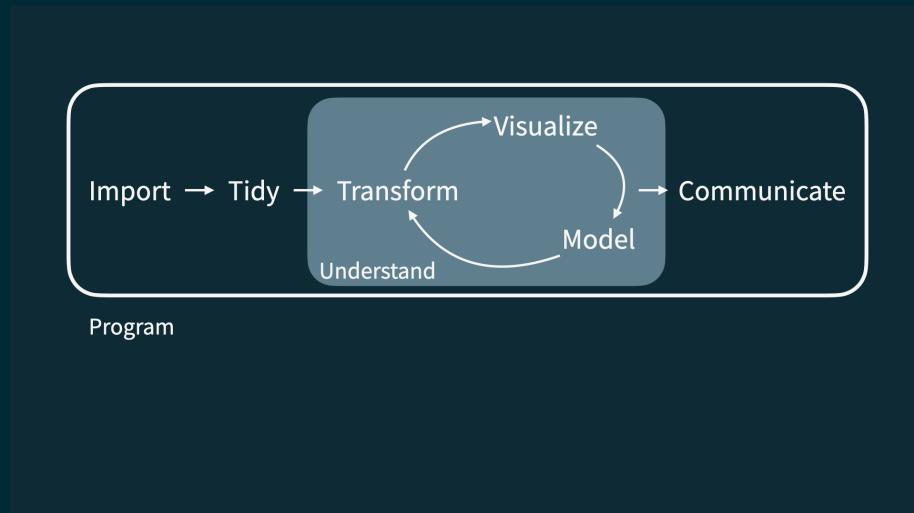




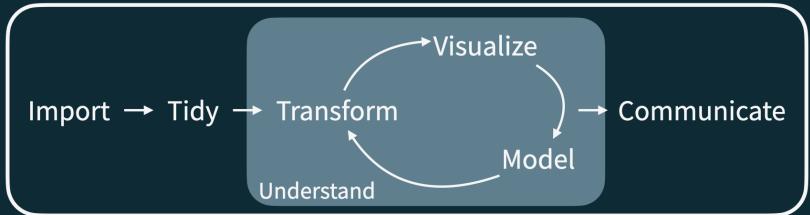
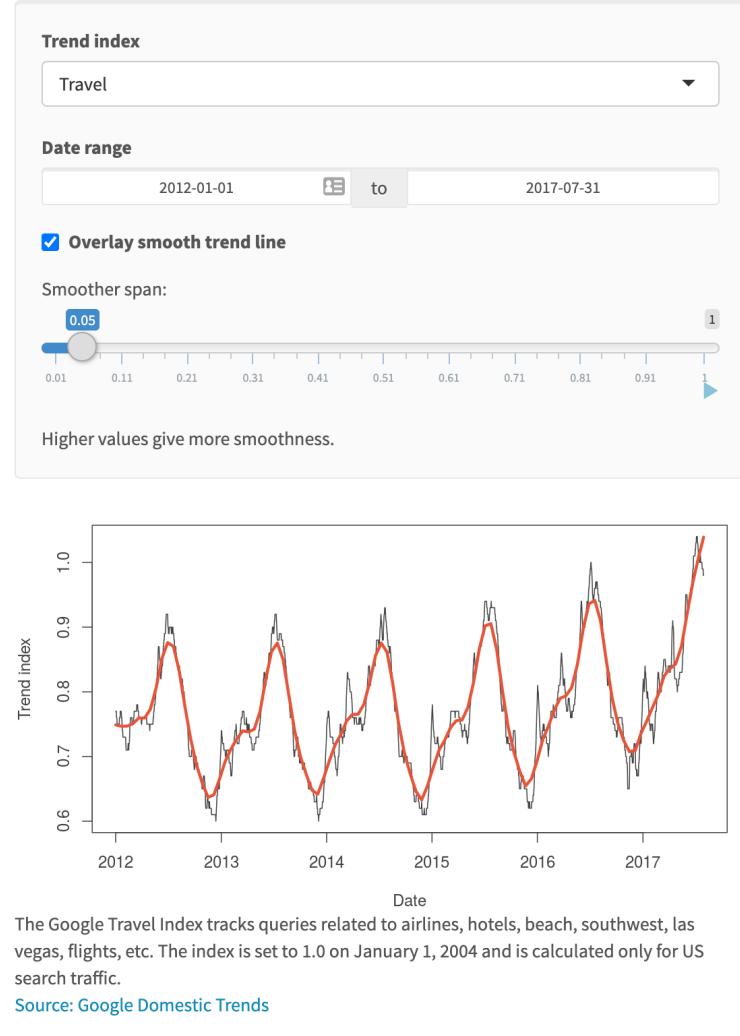
Program



Program

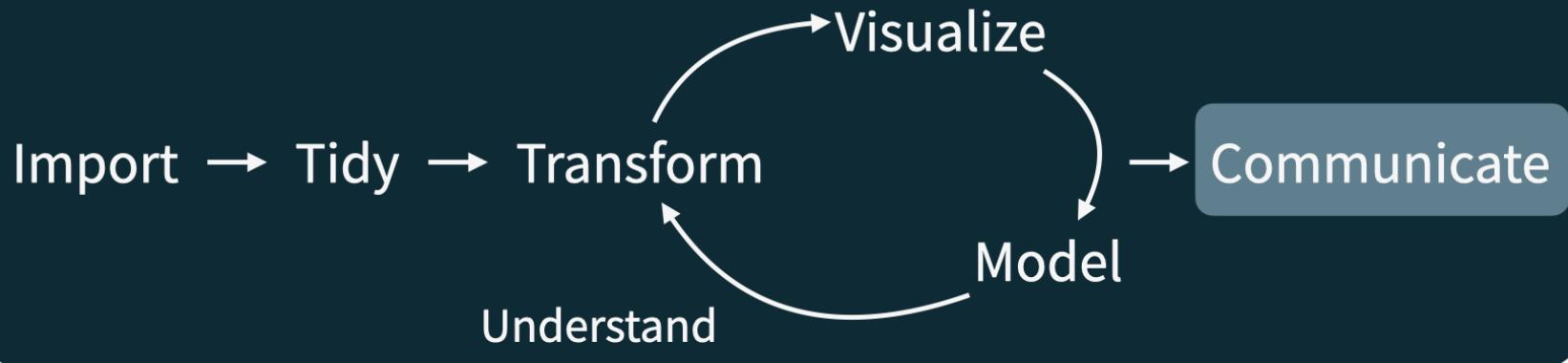


Google Trend Index



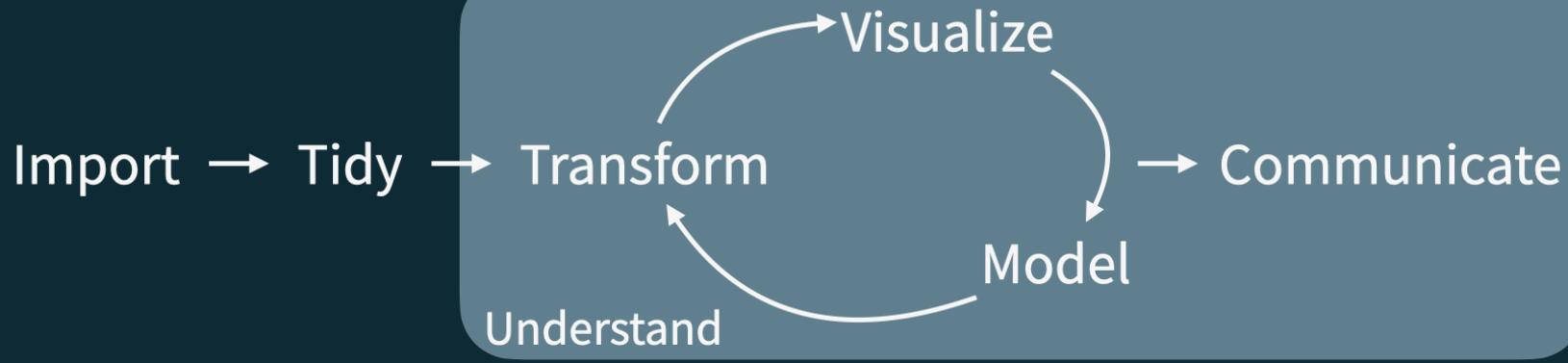
Program

```
## # A tibble: 5 x 2
##   date      season
##   <chr>     <chr>
## 1 23 January 2017 winter
## 2 4 March 2017 spring
## 3 14 June 2017 summer
## 4 1 September 2017 fall
## 5 ...
```

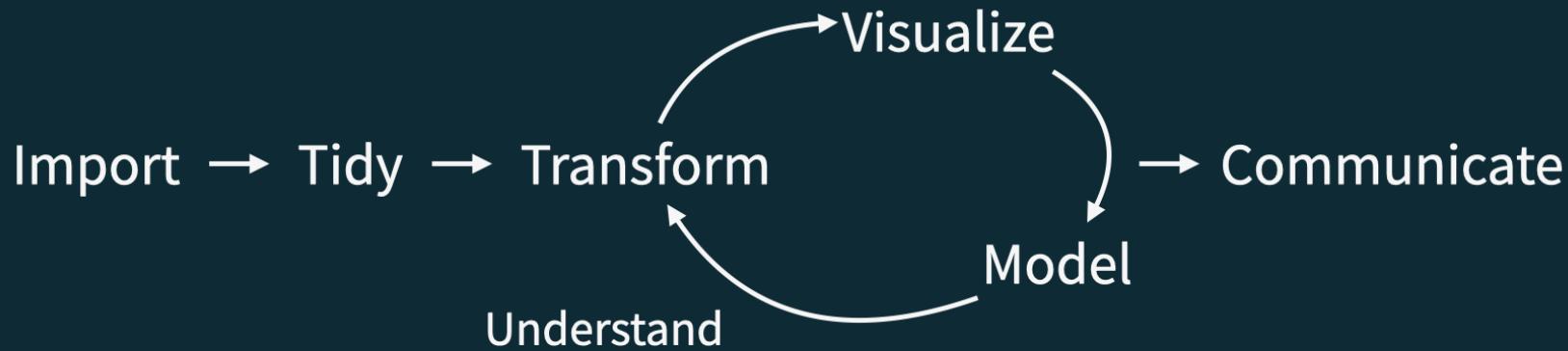


Program





Program



Program



academy-search - master | GitHub

Code Reviewers Add

```

1<-
2  title: "UN Votes"
3  authors: "Mine Cetinkaya-Rundel"
4  date: "2018-08-18"
5  output:
6    html_document
7    pdf_document
8    rmd_document
9    toc: yes
10   toc_float: yes
11   ...
12
13 #R Introduction
14
15 How do various countries vote in the United Nations General Assembly, how have
16 their voting patterns evolved throughout time, and how similarly or differently
17 do they view certain issues? Answering these questions (at a high level) is the
18 focus of this analysis.
19
20 We will use the tidyverse, gridExtra, and magrittr packages for the
21 data wrangling and visualization, and the DT package for interactive display
22 of tabular output. The data we're using come from the unvotes package.
23
24 ##(r load-packages, warning=FALSE, message=FALSE)
25 library(tidyverse)
26 library(gridExtra)
27 library(scales)
28 library(DT)
29 library(unvotes)
30
31
32 # UN voting patterns (plotting)
33
34 Let's create a data visualization that displays how the voting record of the
35 UK & US changed over time on a variety of issues, and compares it
36 to two other countries: US and Turkey.
37
38 We can easily change which countries are being plotted by changing which
39 countries the code above filters for. Note that the country name should be
40 spelled and capitalized exactly the same way as it appears in the data. See
41 the [Appendix] for a list of the countries in the data.
42
43 ##(r plot-yearly-vote-issue, figwidth=8, figheight=6, message=FALSE)
44 un_votes %>%
45   mutate(!)
46   country =
47   case_when(
48     country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK",
49     TRUE ~ identity
50   ) %>%
51   inner_join(un_bill_votes, by = "id") %>%
52 
```

Environment History Connections Git Tutorial

File Plots Packages Help Viewer

Introduction

UN voting patterns

References

Appendix

UN Votes

Mine Cetinkaya-Rundel
2018-08-18

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`, `gridExtra`, 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 (gridExtra)
Library (scales)
Library (DT)
Library (unvotes)

UN voting patterns

Let's create a data visualization that displays how the voting record of the UK & US 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", TRUE ~ country)
  )
  %>% 
  inner_join(un_bill_votes, by = "id") %>%
  inner_join(un_bill_votes, by = "id") %>%
  filter(country %in% "UK", "US", "TUR")
  mutate(issue = year(issue))
  group_by(issue, year, country)
  summarise(issue_mean = mean(issue))
  ggplot(mapping = aes(x = year, y = percent_yea, color = country)) +
  geom_point(size = 4, alpha = 0.5) +
  geom_smooth(method = "loess", se = FALSE) +
  facet_wrap(~country)
  ggplot(mapping = geom_bar(mapping = aes(x = year, y = count, color = country)))
  labs(
    title = "Percentage of 'Yea' votes in the UN General Assembly",
    subtitle = "(Data up to 2017)",
    x = "Year",
    y = "Count",
    color = "Country")
  )
  theme_minimal()

```

Let's dive in!

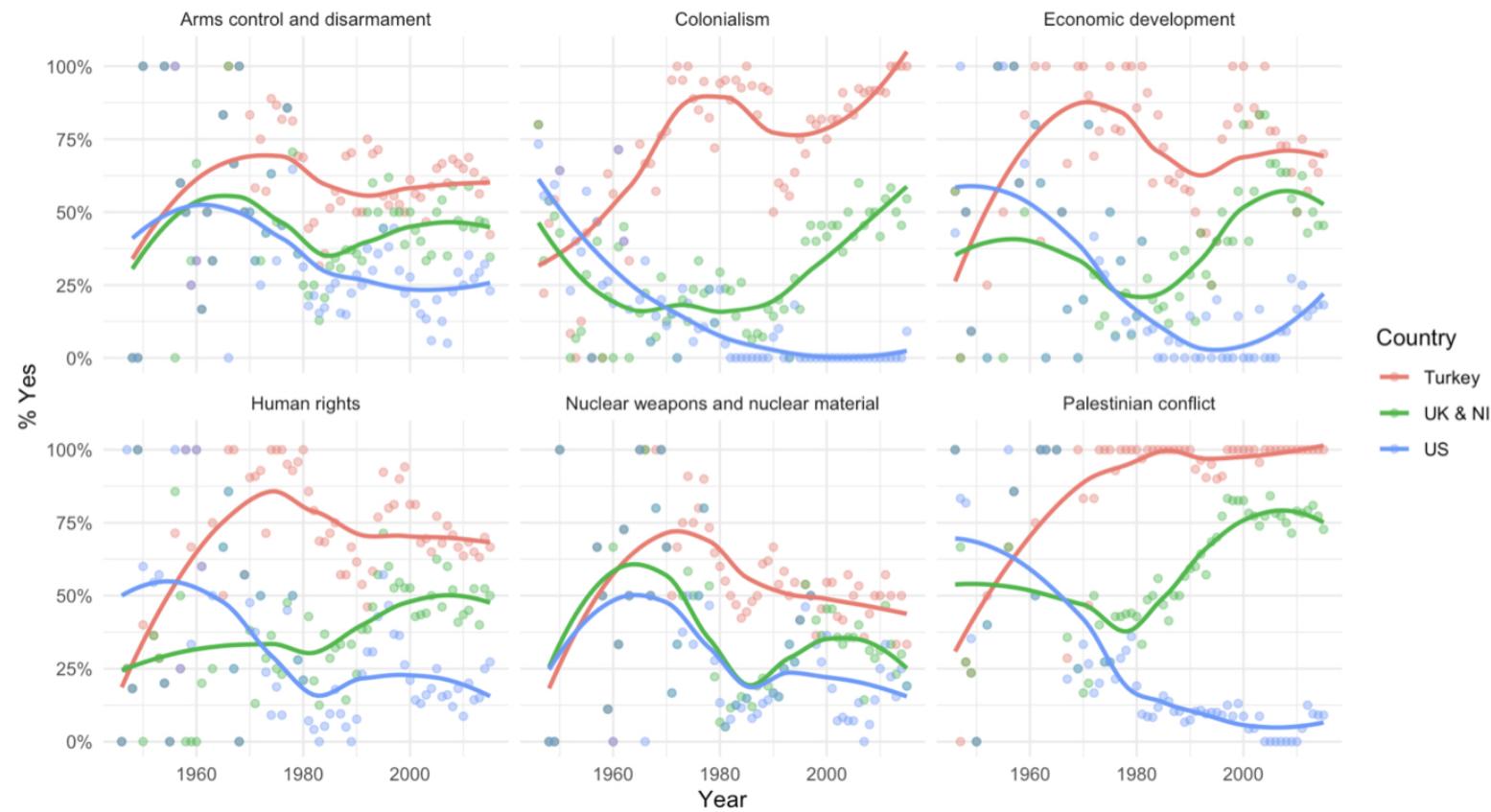


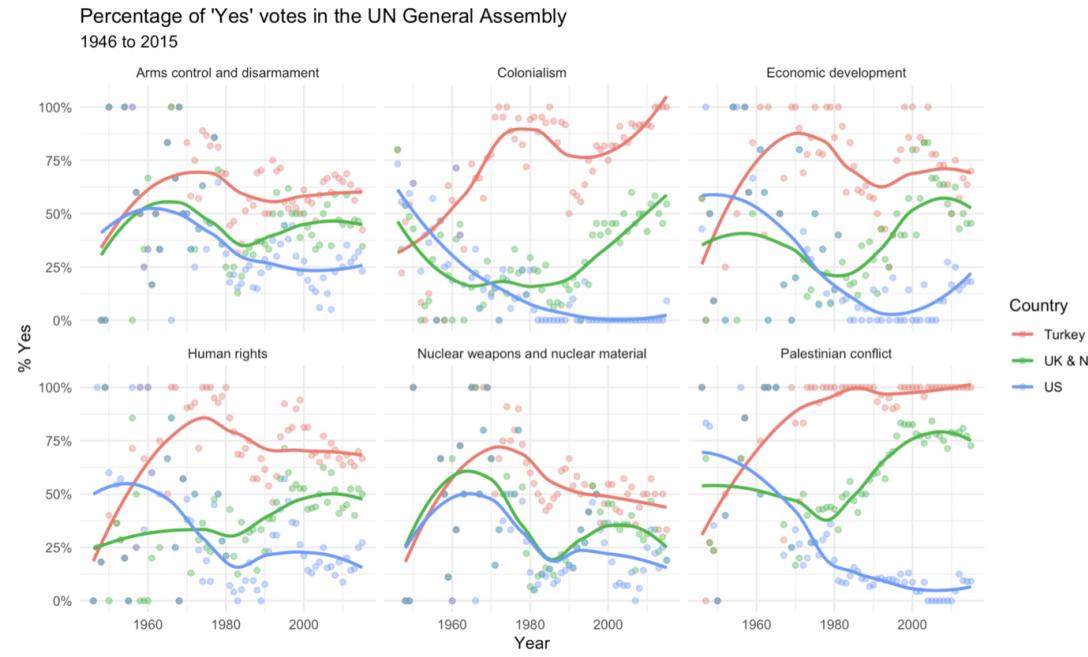
datasciencebox.org



datasciencebox.org

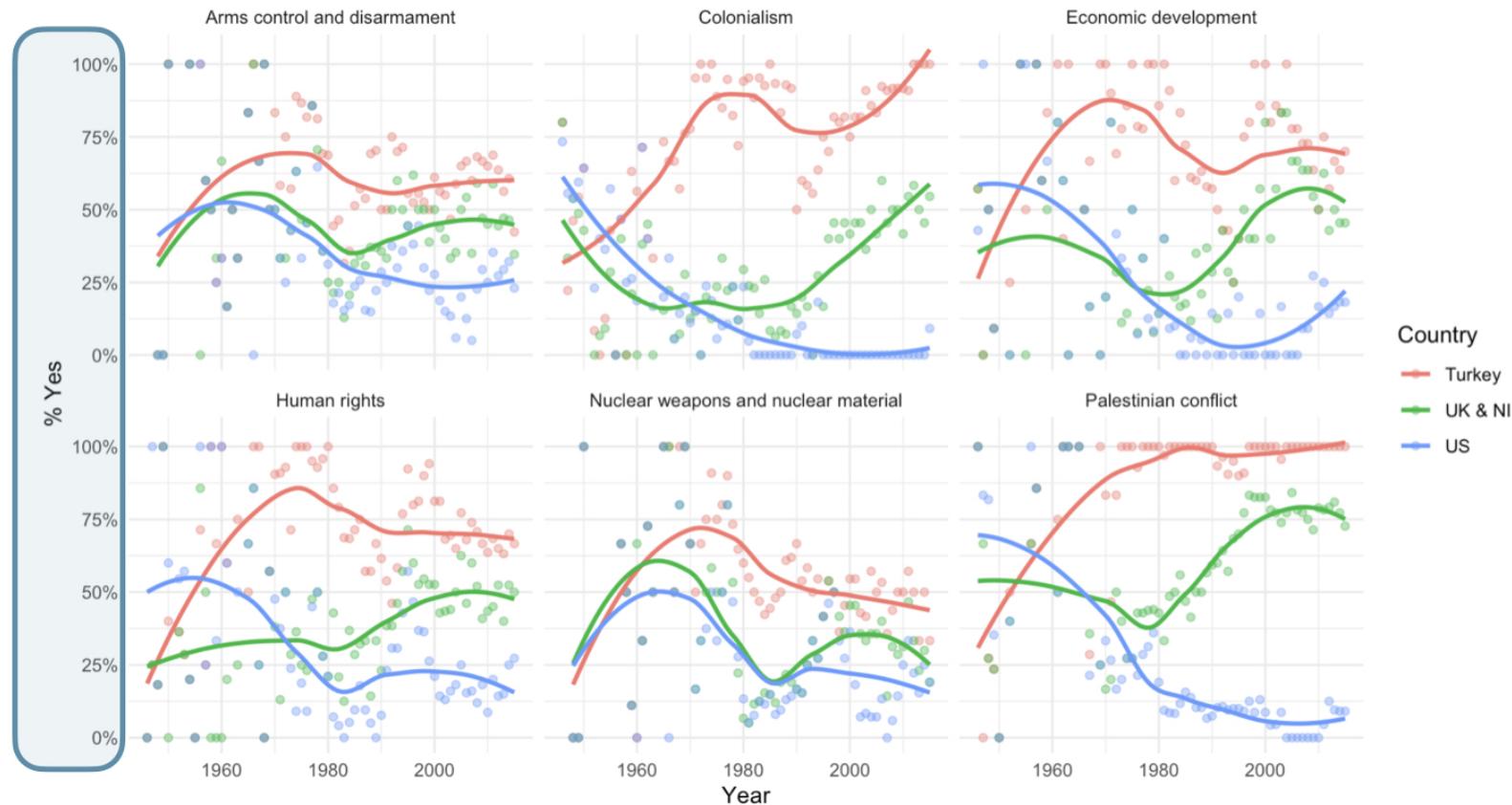
Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015





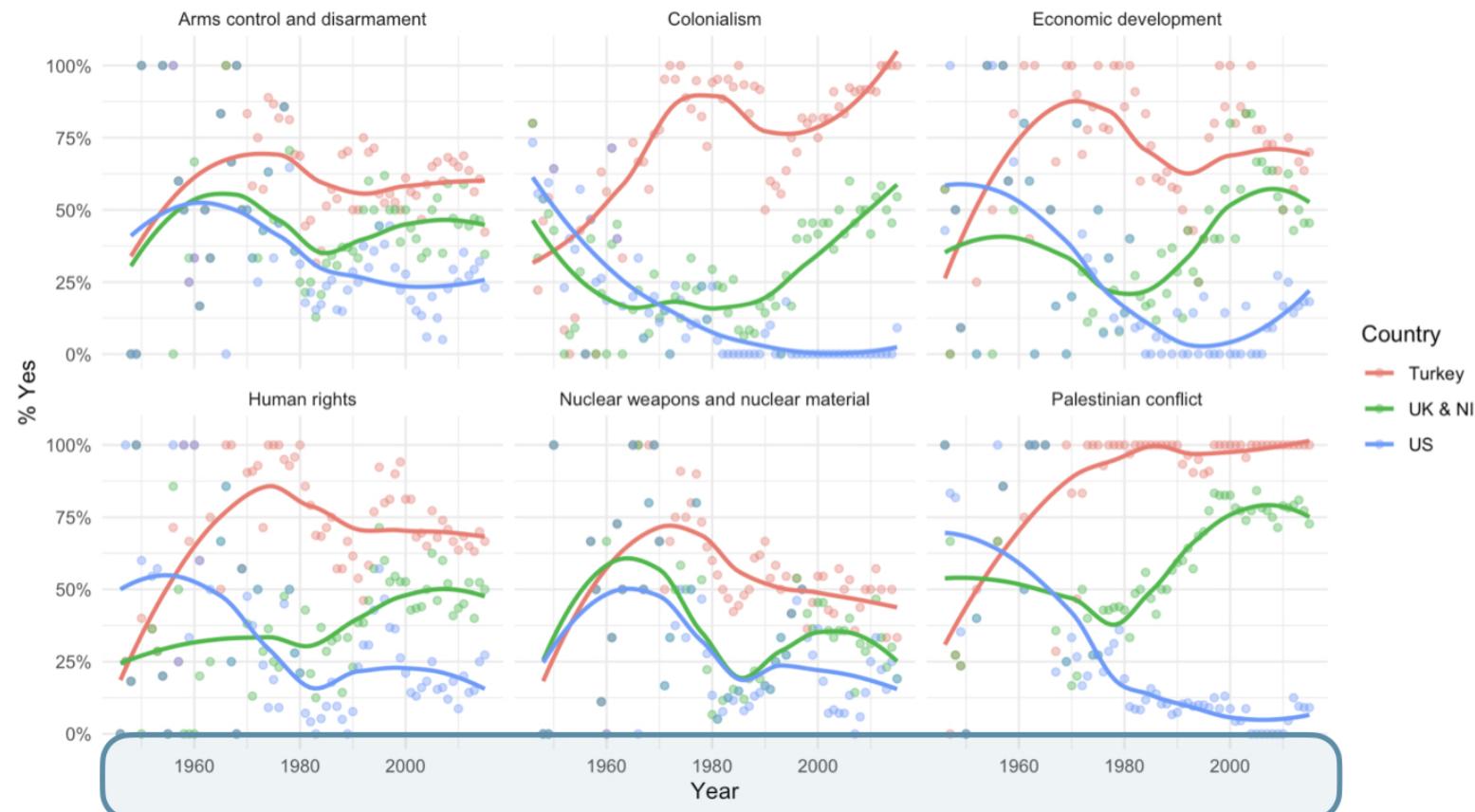
- Where can we find the issues for each visualization?
- Which countries are being visualized?
- What is our response? What is on our y-axis?
- What is our predictors? What is on the x-axis?

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

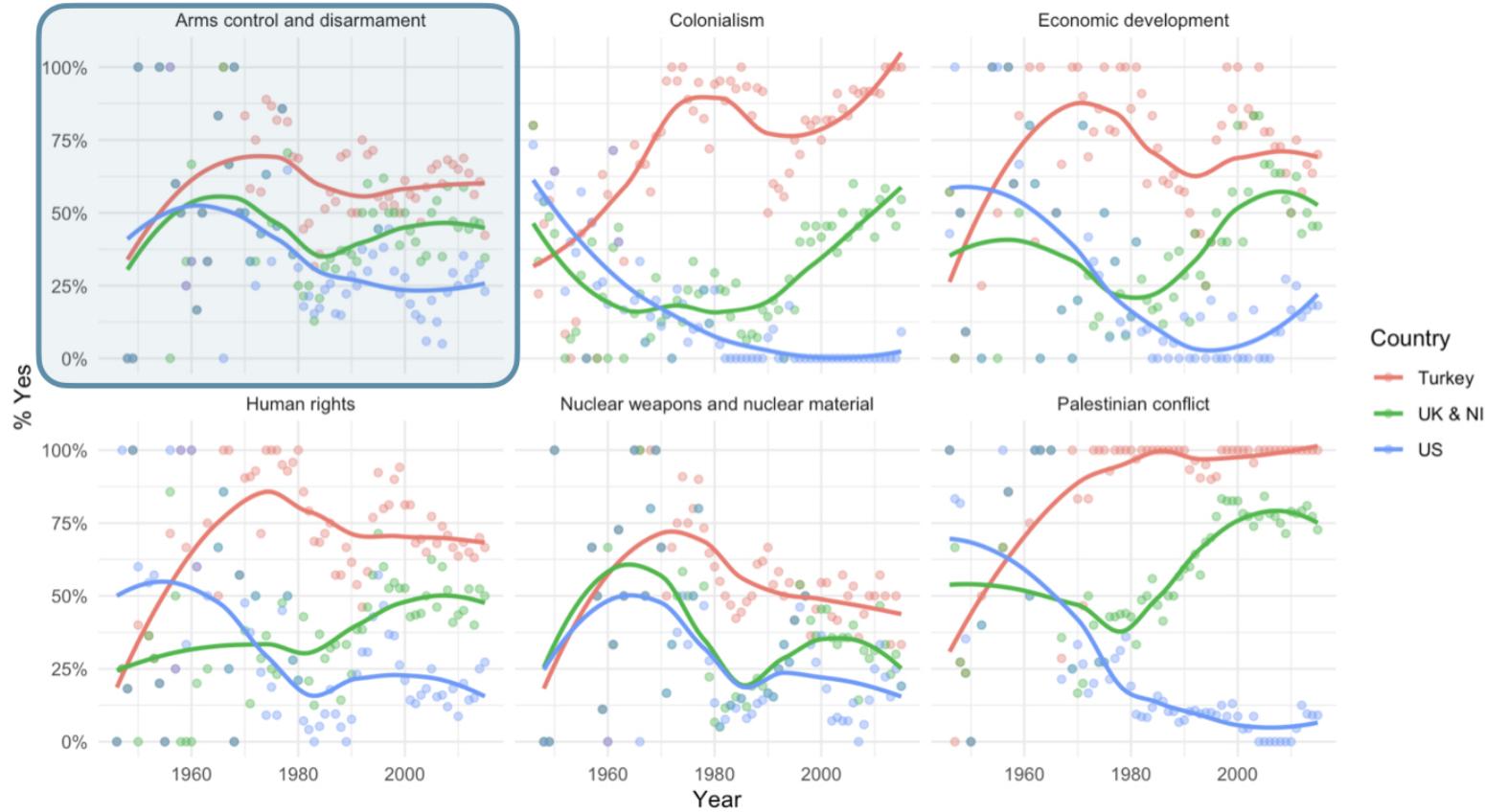


Country
— Turkey
— UK & NI
— US

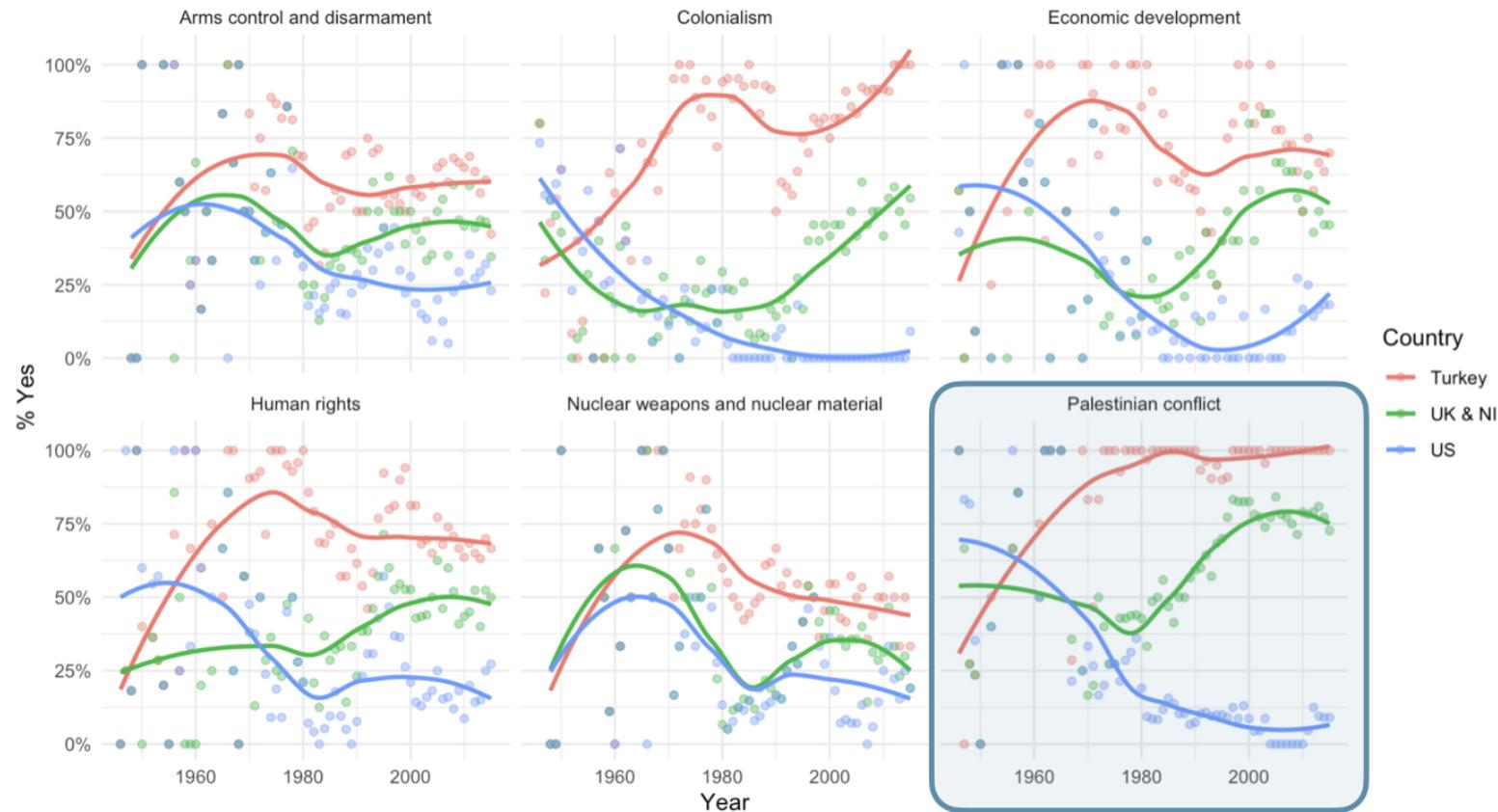
Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015



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 displays three stacked data grids, each with a header row and numbered rows from 1 to 26. The top grid has columns: rcid, session, importantvote, date, unres, amend, para, short. The middle grid has columns: rcid, short_name, issue. The bottom grid has columns: rcid, short_name, issue. All grids show data related to 'Palestinian conflict'.

	rcid	session	importantvote	date	unres	amend	para	short
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	5020	me	Palestinian conflict					

Showing 1 to 26 of 5,281 entries, 3 total columns

```
unvotes.Rmd x ABC Knit 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
```



```
unvotes.Rmd x ABC 🔍 Knit ⚙️ 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
```



```
unvotes.Rmd x ABC Knit 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
```



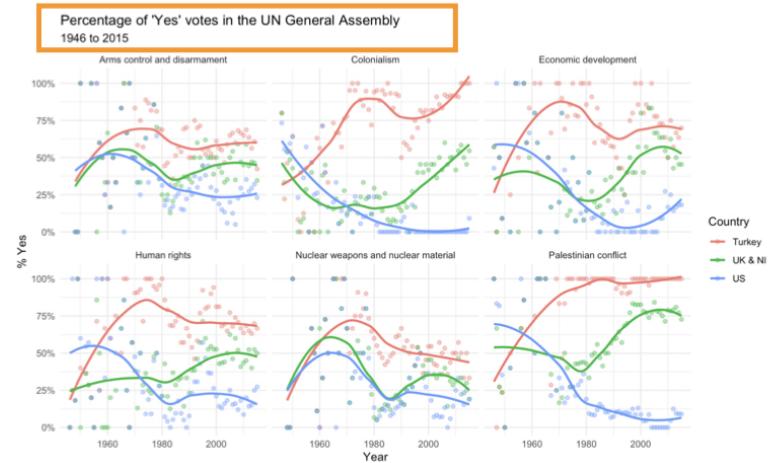
```
unvotes.Rmd x ABC Knit ▾ Insert ▾ Run ▾ A
36 We can easily change which countries are being plotted by changing which
37 countries the code above `filter`'s for. Note that the country name should be
38 spelled and capitalized exactly the same way as it appears in the data. See
39 the [Appendix](#appendix) 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

```



academy-launch - master - RStudio

unvotes.Rmd

```

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 ```
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 `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       country =
46         case_when(
47           country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK & NI",
48           country == "United States of America" ~ "US",
49           TRUE ~ country
50         )
51       ) %>%
52       inner_join(un_roll_calls, by = "rcid") %>%
53       inner_join(un_roll_call_issues, by = "rcid") %>%
54       filter(country %in% c("UK & NI", "US", "Turkey")) %>%
55       mutate(year = year(date)) %>%
56       group_by(country, year, issue) %>%

```

Environment History Connections Git Tutorial

Files Plots Packages Help Viewer

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 `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.

```

un_votes %>%
  mutate(
    country =
      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) %>%

```

academy-launch - master - RStudio

unvotes.Rmd

```

1 title: "UN Votes"
2 author: "Mine Çetinkaya-Rundel"
3 date: `r Sys.Date()`
4 output:
5   html_document:
6     toc: yes
7     toc_float: yes
8 ---
9
10 ## Introduction
11
12 How do various countries vote in the United Nations General Assembly, how have
13 their voting patterns evolved throughout time, and how similarly or differently
14 do they view certain issues? Answering these questions (at a high level) is the
15 focus of this analysis.
16
17 We will use the tidyverse, lubridate, and scales packages for the
18 data wrangling and visualization, and the DT package for interactive display
19
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 ````

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 `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 =

```

3:2 UN Votes

Console

Environment History Connections Git Tutorial

Files Plots Packages Help Viewer

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 `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.

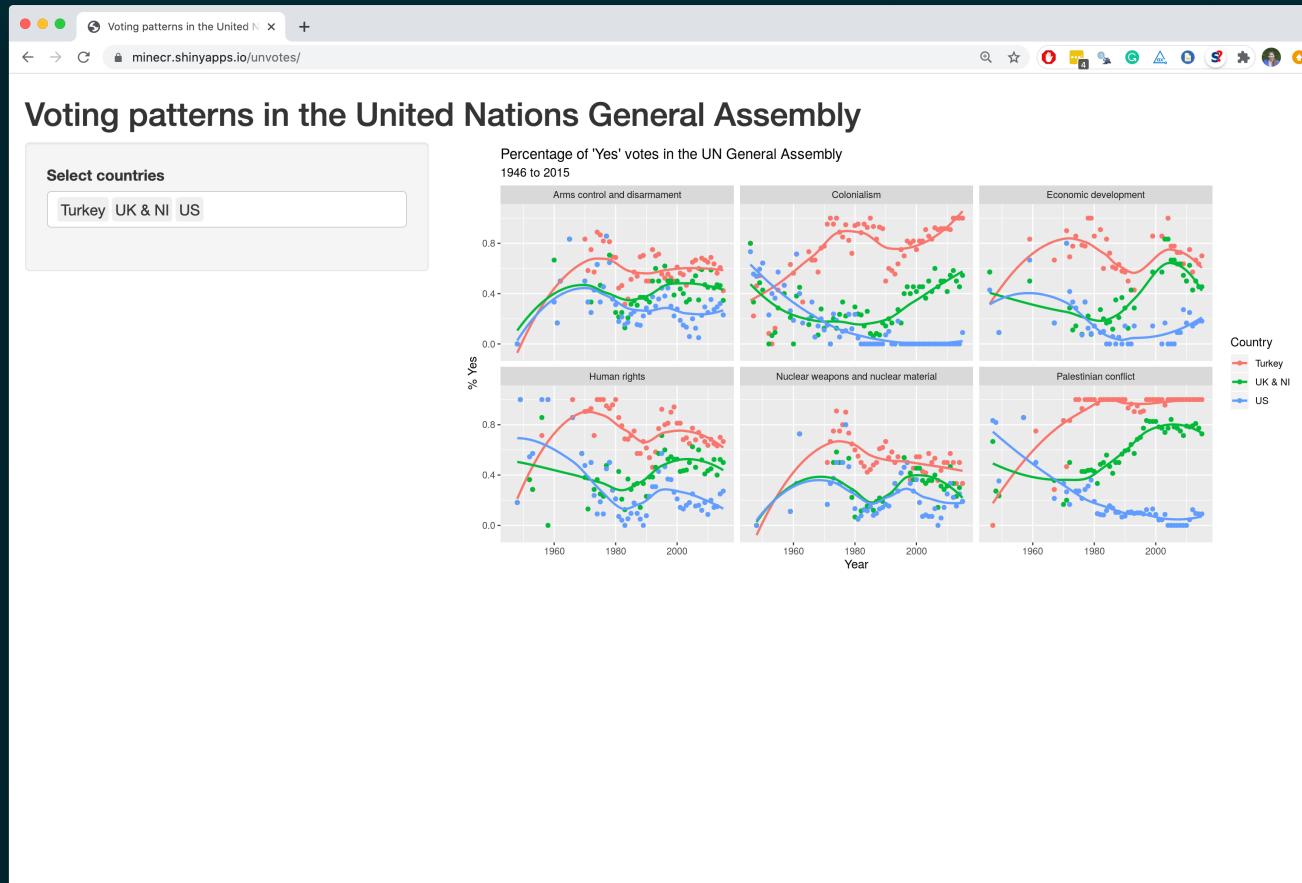
```

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) %>%

```



minecr.shinyapps.io/unvotes



Course toolkit

Course operation

- Moodle

Doing data science

- Programming:
 - R
 - RStudio
 - tidyverse
 - R Markdown
- Version control and collaboration:
 - Git
 - GitHub



Learning goals

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

- use data carefully and ethically
- 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



Reproducibility checklist

What does it mean for a data analysis to be "reproducible"?

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 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 data¹
- As of November 2021, there are over 18,000 R packages available on **CRAN** (the Comprehensive R Archive Network)²
- We're going to work with a small (but important) subset of these!

¹ Wickham and Bryan, R Packages.

² CRAN contributed packages.



Tour: R and RStudio

The screenshot shows the RStudio interface with several annotated features:

- data viewer**: Points to the Data Viewer pane displaying the "penguins" dataset.
- arithmetic**: Points to the Console pane showing the calculation `> 2 + 2`.
- load package**: Points to the Console pane showing the command `> library(palmerpenguins)`.
- view data**: Points to the Data Viewer pane showing the first 11 rows of the penguins dataset.
- get help**: Points to the Help pane showing the documentation for the `mean` function.
- Object assignment**: Points to the Console pane showing the assignment `> x <- 2`.
- access variable**: Points to the Console pane showing the access `> x * 3`.
- use function**: Points to the Console pane showing the use of the `mean` function `> mean(penguins$flipper_length_mm)`.
- environment**: Points to the Environment pane showing the variable `x` with value `2`.

Help Pane (Documentation for `mean`):

- Description**: Generic function for the (trimmed) arithmetic mean.
- Usage**: `mean(x, ...)`
- Arguments**: `## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)`
- Examples**:

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

[Package base version 4.0.2 Index]

A short list (for now) of R essentials

- 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)
```



R essentials (continued)

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

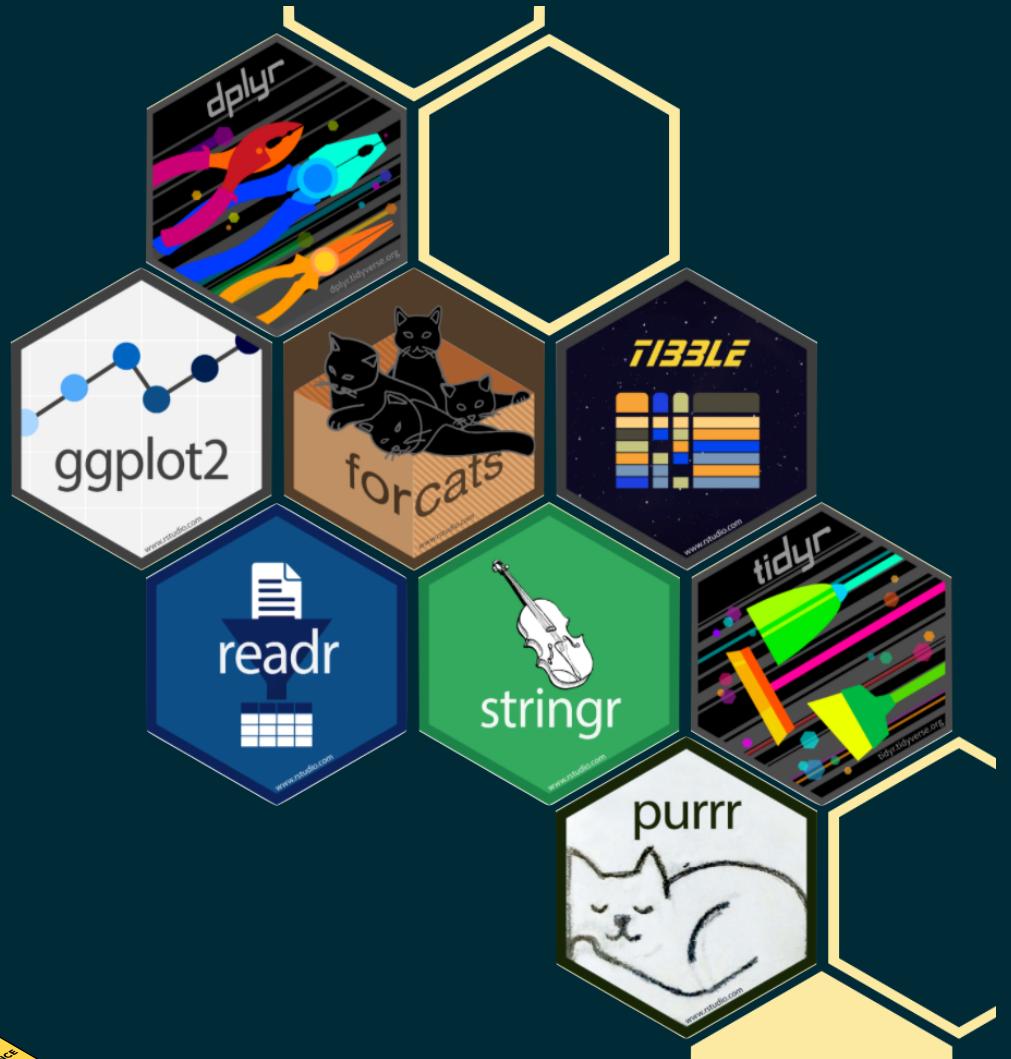
```
dataframe$var_name
```

- Object documentation can be accessed with ?

```
?mean
```



tidyverse



tidyverse.org

- The **tidyverse** is an opinionated collection of R packages designed for data science
- All packages share an underlying philosophy and a common grammar

rmarkdown

rmarkdown.rstudio.com

- **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



R Markdown



R Markdown

- 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 includes YAML front matter and an R code chunk. A yellow arrow labeled "knit" points to the "Knit" button in the toolbar. A red arrow labeled "yaml" points to the YAML section. A green arrow labeled "link" points to the URL in the text. A pink arrow labeled "code chunk" points to the R code chunk. The right pane displays the rendered HTML output, which includes the YAML metadata, the descriptive text, the "Data and packages" section, and a code block for loading packages. The rendered output also shows the dataset summary and financial variables.

```
---  
title: "Bechdel"  
author: "Mine Çetinkaya-Rundel"  
output:  
  html_document:  
    fig_height: 4  
    fig_width: 9  
---  
  
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"](https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-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.  
  
```{r load-packages, message=FALSE}  
library(fivethirtyeight)
library(tidyverse)
...
```
```

bechdel190_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 adjusted dollars

Environments

The environment of your R Markdown document is separate from the Console!

Remember this, and expect it to bite you a few times as you're learning to work with R Markdown!



Environments

First, run the following in the console

```
x <- 2  
x * 3
```

All looks good, eh?

Then, add the following in an R chunk in your R Markdown document

```
x * 3
```

What happens? Why the error?



R Markdown help

R Markdown Cheat Sheet
Help -> Cheatsheets

This Cheat Sheet (and others) will be on Moodle

R Markdown :: CHEAT SHEET

What is R Markdown?

.Rmd files - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand the work and the Reproducible Research - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents - You can choose to export the rendered report in a variety of formats, including HTML, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

Workflow

Open a new .Rmd file at File > New File > R Markdown. Use the wizard that opens to populate the file with a template.

Write document by editing template

Knit document to create report; use knit button or render()

Preview Output in IDE window

Publish (optional) to web server

Examine build log in R Markdown console

Use output file that is saved along side.Rmd

render

Use `markdown::render()` to render/knit at cmd line. Important args:

| | | | | |
|-------------------------------------|--|--------------------------|-------------------------|---|
| <code>input</code> - file to render | <code>output_options</code> - List of render options (as YAML) | <code>output_file</code> | <code>output_dir</code> | <code>params</code> - list of params to use |
|-------------------------------------|--|--------------------------|-------------------------|---|

Embed code with knitr syntax

INLINE CODE

Insert with `r <>`. Results appear as text without code.
Built with `r getRVersion()` Built with 3.2.3

CODE CHUNKS

One or more lines surrounded with ````{r}` and `````. Place chunk options within curly braces, after r. Insert with `r`.

```
```{r echo=TRUE}
getRVersion()
```
R: 3.2.3
```

GLOBAL OPTS

Set with `knitr::opts_chunk$set(...)`

```
```{r include=FALSE}
knitr::opts_chunk$set(echo=TRUE)
```
R: 3.2.3
```

Markdown Quick Reference
Help -> Markdown Quick Reference

Link on Moodle and [HERE](#)

Files Plots Packages Help Viewer

Markdown Quick Reference Find in Topic

Markdown Quick Reference

R Markdown is an easy-to-write plain text format for creating dynamic documents and reports. See [Using R Markdown](#) to learn more.

Emphasis

`*italic*` `**bold**`
`_italic_` `__bold__`

Headers

Header 1
Header 2
Header 3

Lists

Unordered List

- * Item 1
- * Item 2
 - + Item 2a
 - + Item 2b

Ordered List

1. Item 1
2. Item 2
3. Item 3
 - + Item 3a
 - + Item 3b

Manual Line Breaks

End a line with two or more spaces:
Roses are red,
Violets are blue.

Links

Use a plain http address or add a link to a phrase:
<http://datasciencebox.org>



How will we use R Markdown?

- Every assignment / report / project / etc. is an R Markdown document
- You'll always have a template R Markdown document to start with
- The amount of scaffolding in the template will decrease over the semester



What's with all the hexes?



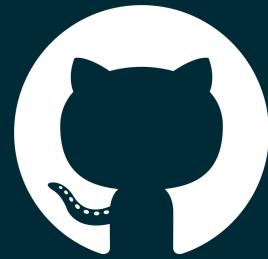
Mitchell O'Hara-Wild, useR! 2018 feature wall



Git and GitHub

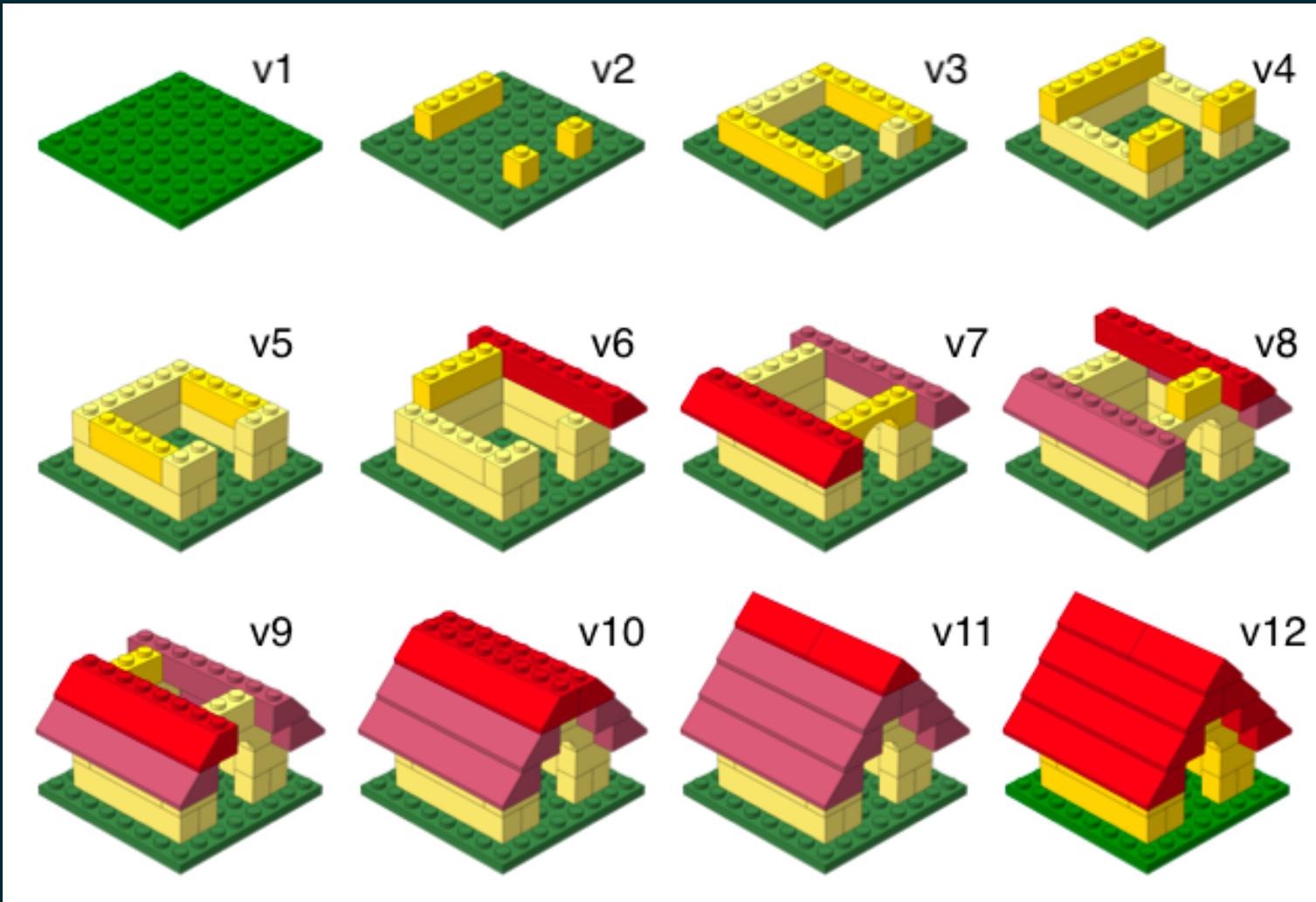


Git and GitHub

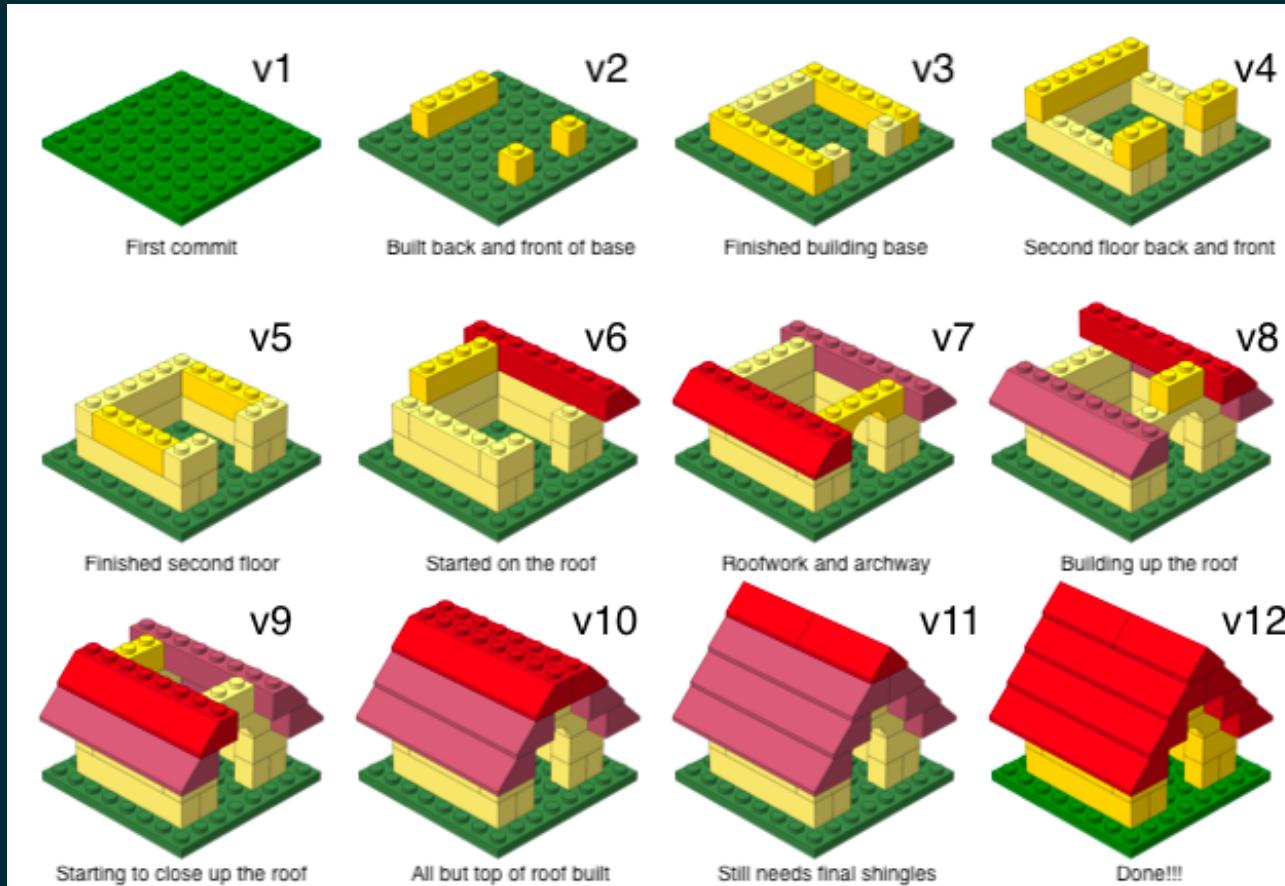


- Git is a version control system -- like “Track Changes” features from Microsoft Word, on steroids
- It's not the only version control system, but it's a very popular one
- GitHub is the home for your Git-based projects on the internet -- like DropBox but much, much better
- We will use GitHub as a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

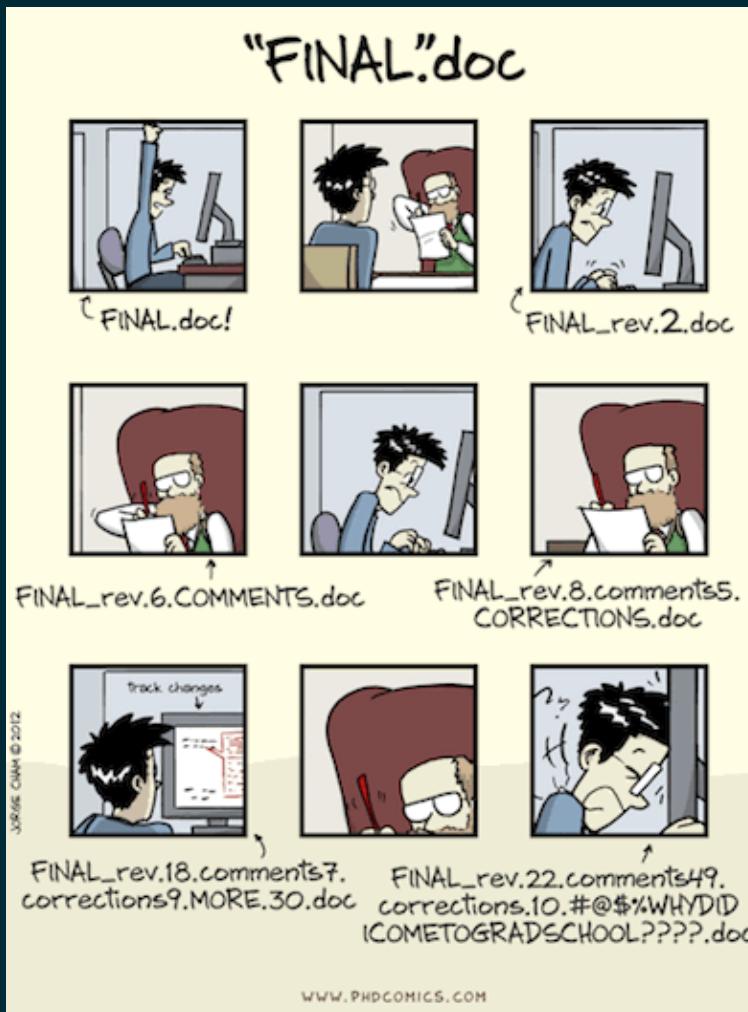
Versioning



Versioning with human readable messages



Why do we need version control?



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?



Git and GitHub tips

- There are millions of git commands -- ok, that's an exaggeration, but there are a lot of them -- and very few people know them all. 99% of the time you will use git to add, commit, push, and pull.
- We will be doing Git things and interfacing with GitHub through RStudio, but if you google for help you might come across methods for doing these things in the command line -- skip that and move on to the next resource unless you feel comfortable trying it out.
- There is a great resource for working with git and R: happygitwithr.com. Some of the content in there is beyond the scope of this course, but it's a good place to look for help.



Tour: Git and GitHub

- Create a GitHub account
- Verify your GitHub email
- Adjust your GitHub settings for a more pleasant GitHub experience
 - Settings > Emails > Uncheck "Keep my email address private"
 - Settings > Emails > Update name and photo

Next...

Work with R, RStudio, Git, and GitHub together![†]

[†]Just like a real data scientist!

