

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



datasciencebox.org

AutoSave OFF

unvotes — Saved to my Mac

Search Sheet

Home Insert Page Layout Formulas Data Review View Table

Paste **B** *I* U **A** *A* **A** Wrap Text General Conditional Formatting as Table Cell Styles Insert Delete Format Sort & Filter

Merge & Center

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

File Edit View Insert Cell Help Addins

unvotes

Filter

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

>

Environment History Connections Git Tutorial

Import Dataset Global Environment

Data unvotes 768674 obs. of 14 variables

Files Plots Packages Help Viewer

New Folder Delete Rename More

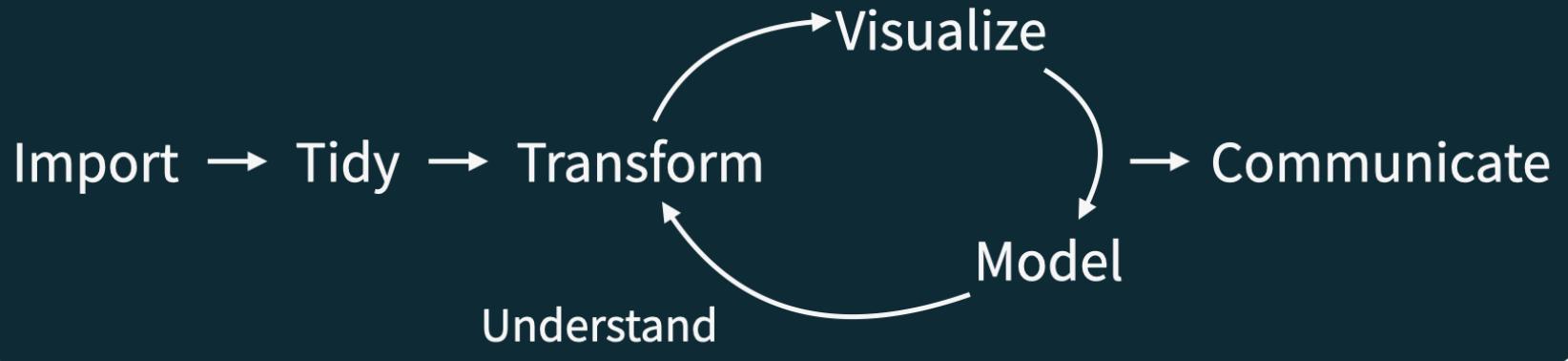
Home > Desktop > academy-launch

Name	Size	Modified
..		
.gitignore	29 B	Aug 18, 2020, 10:18
academy-launch.Rproj	235 B	Aug 18, 2020, 10:32
data		
unvotes.Rmd	2.8 KB	Aug 17, 2020, 2:01



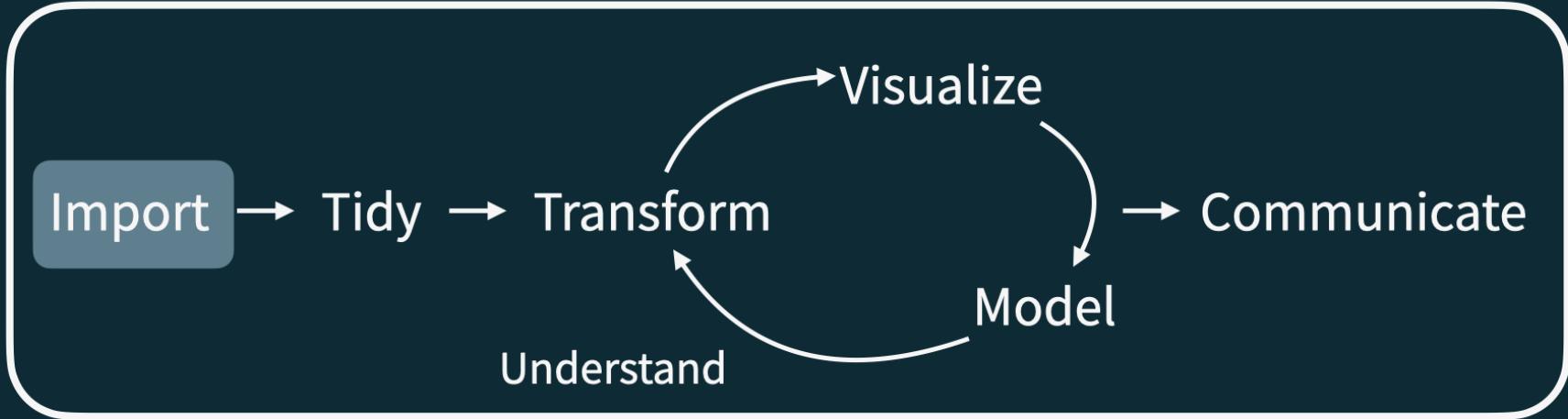
Data science life cycle





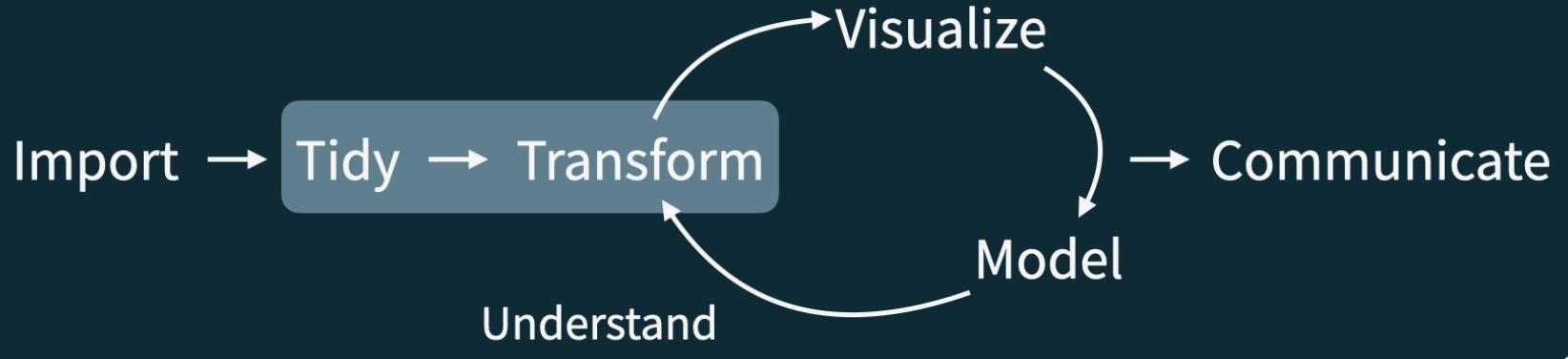
Program





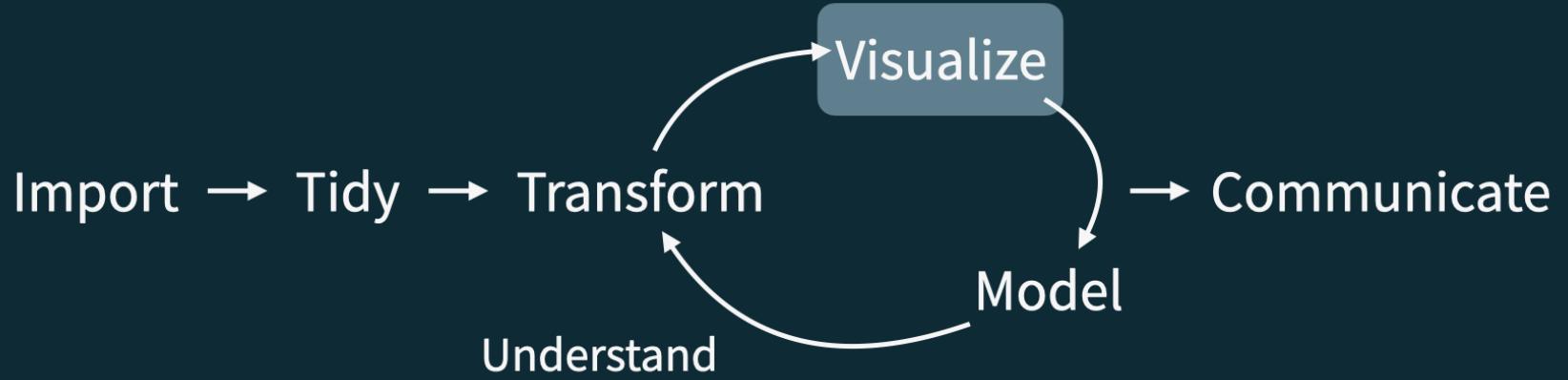
Program





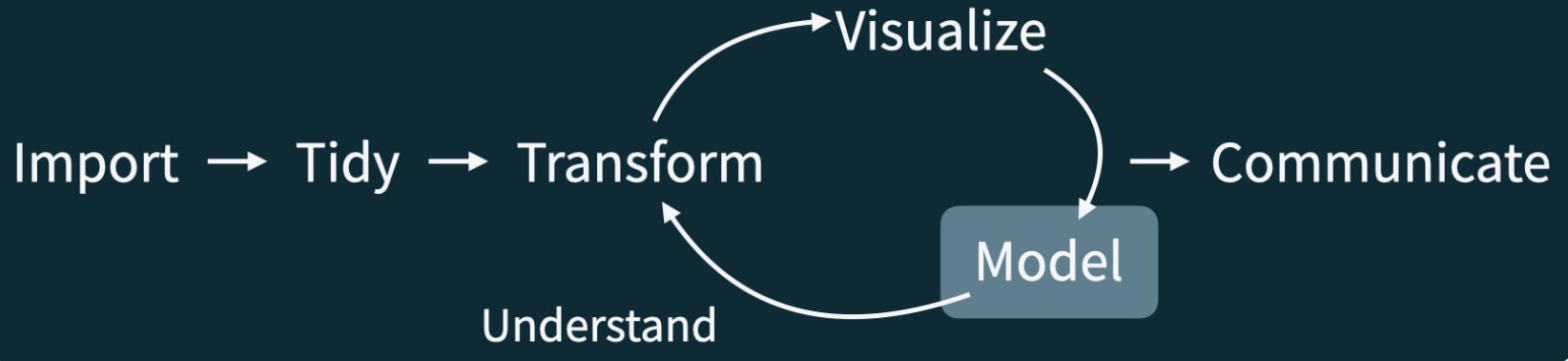
Program





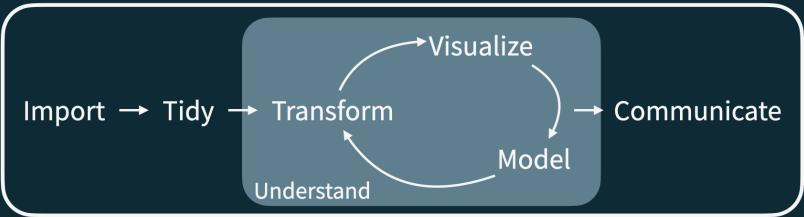
Program





Program

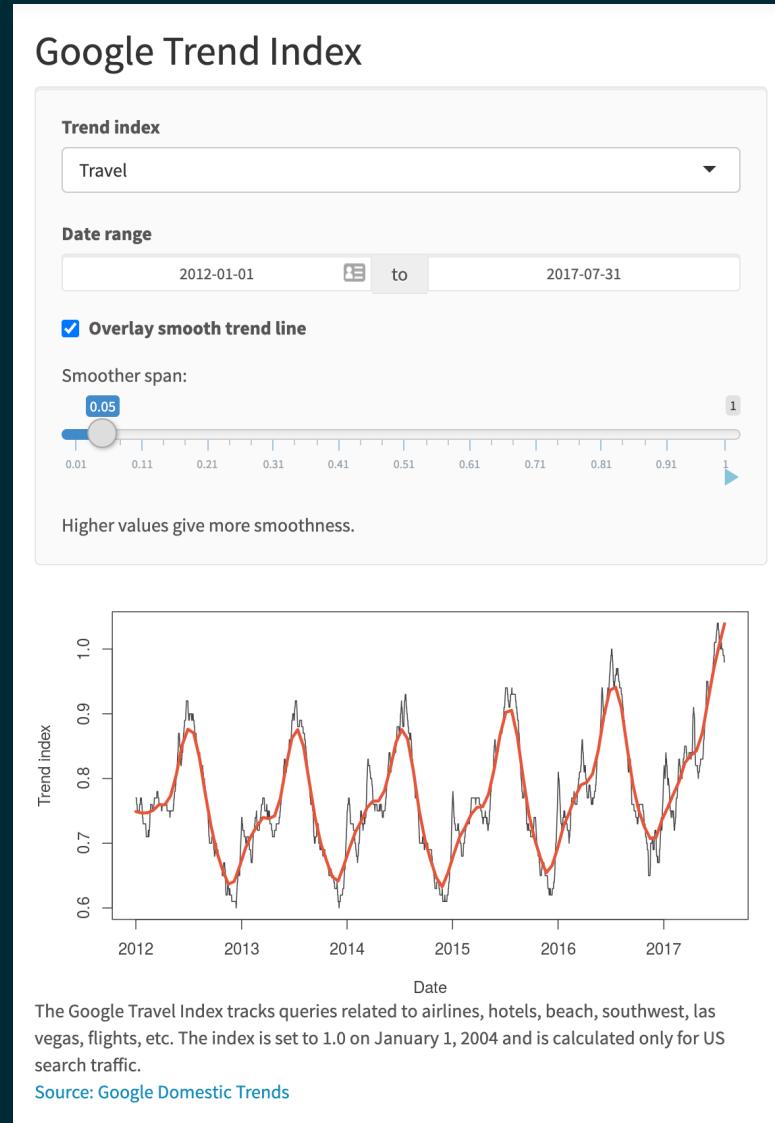
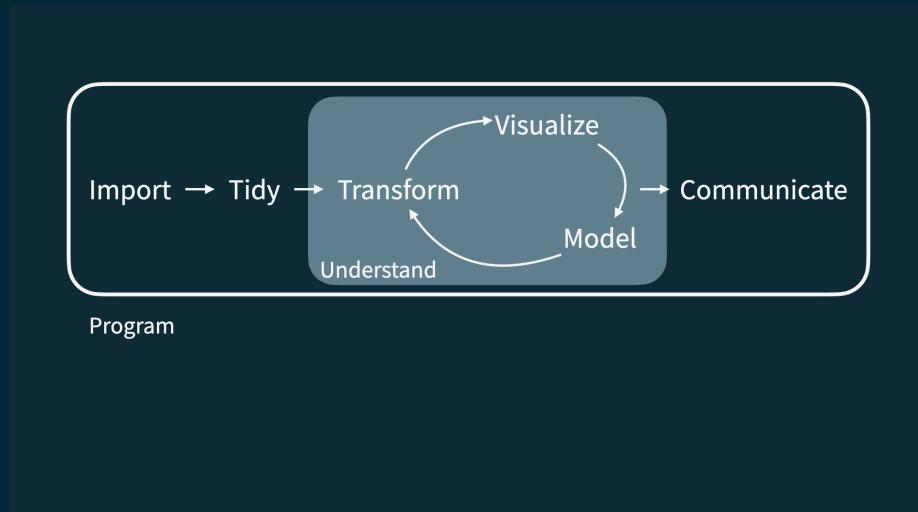




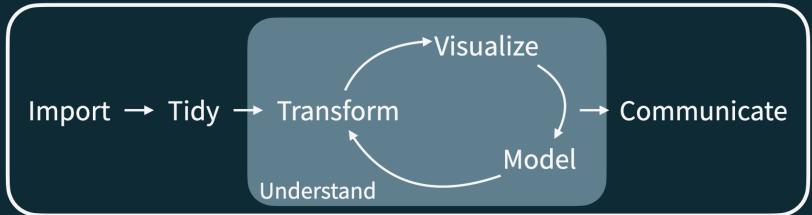
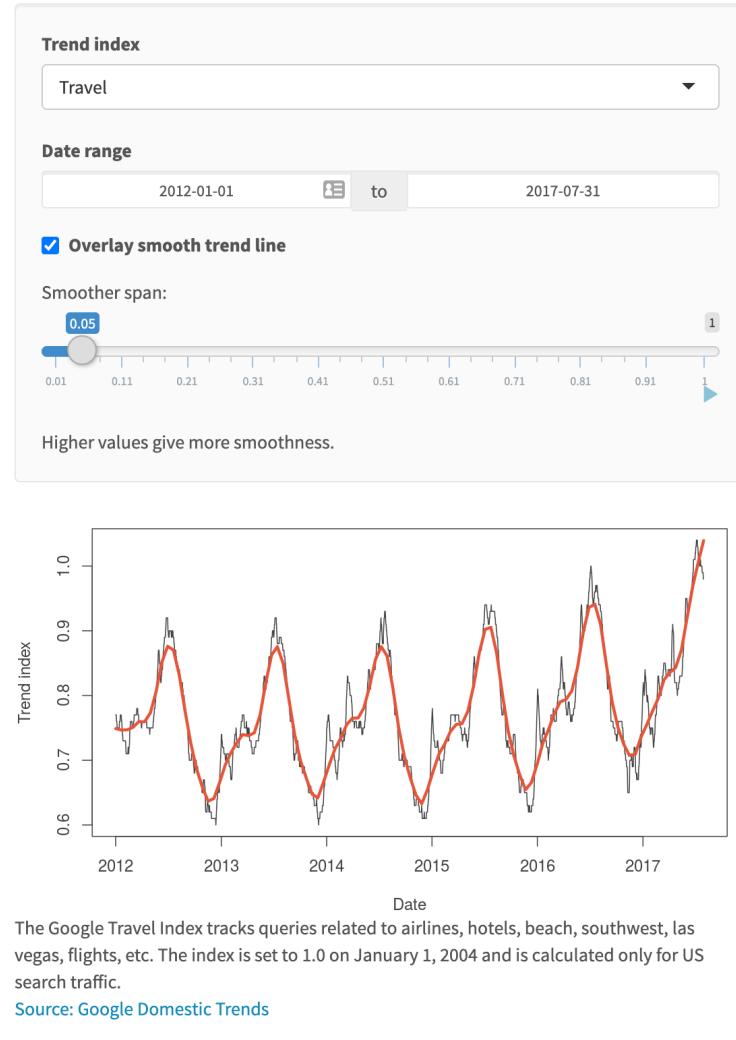
Program



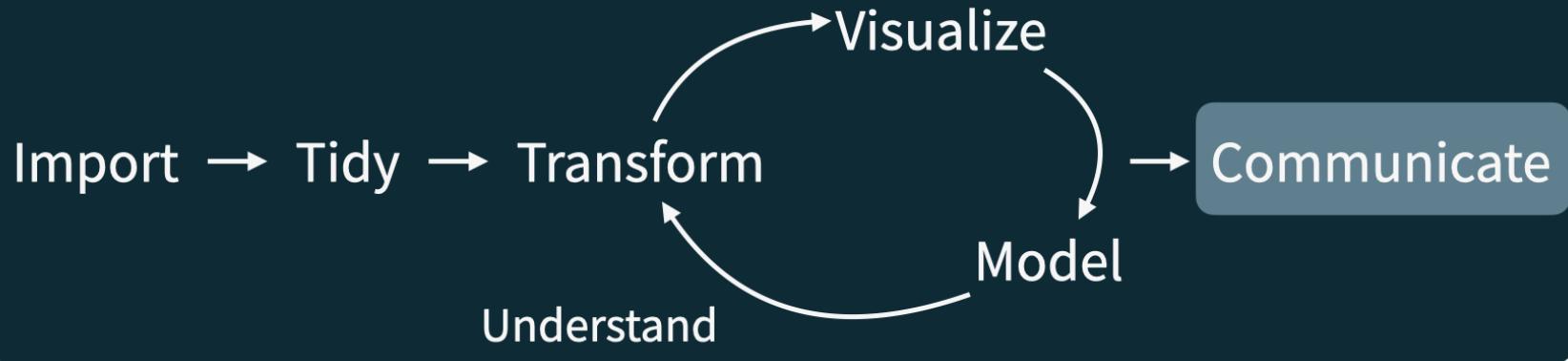
datasciencebox.org



Google Trend Index

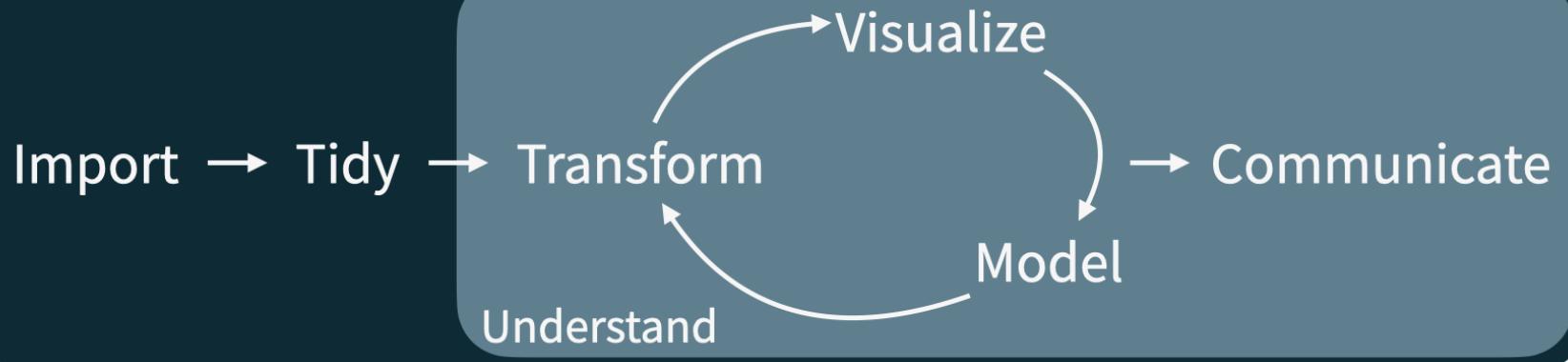


```
## # A tibble: 5 x 2
##   date      season
##   <chr>     <chr>
## 1 23 January 2017 winter
## 2 4 March 2017 spring
## 3 14 June 2017 summer
## 4 1 September 2017 fall
## 5 ...
```



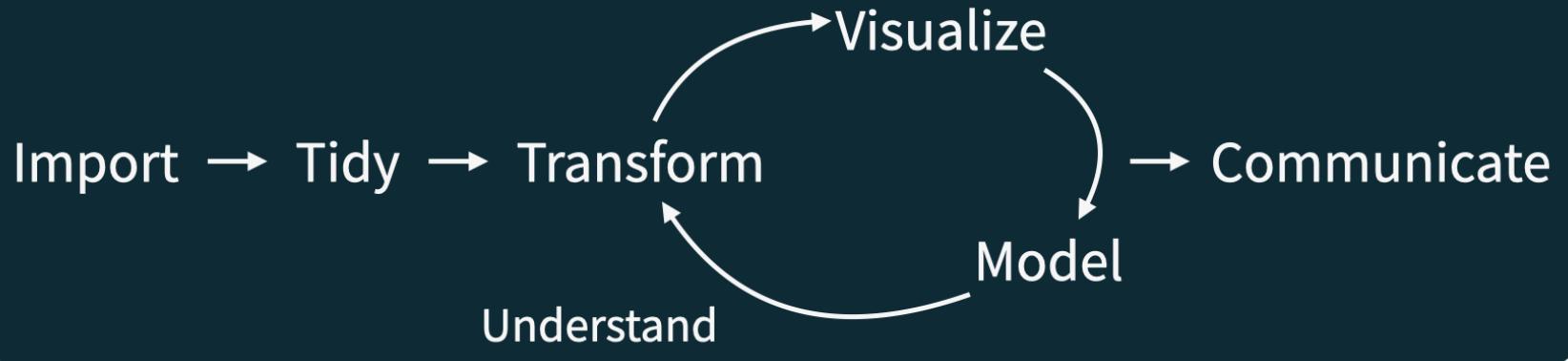
Program





Program





Program



datasciencebox.org



Let's dive in!

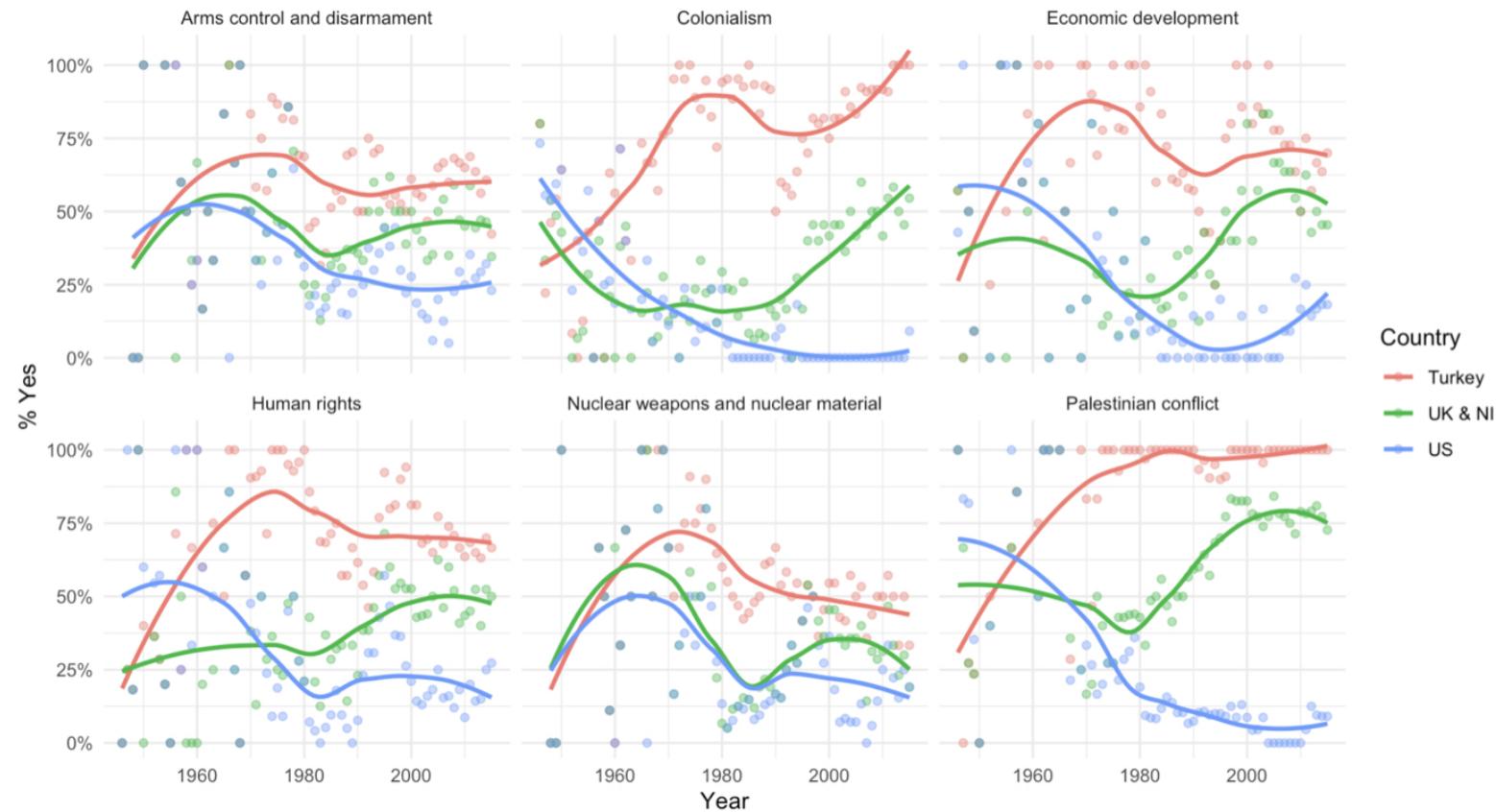


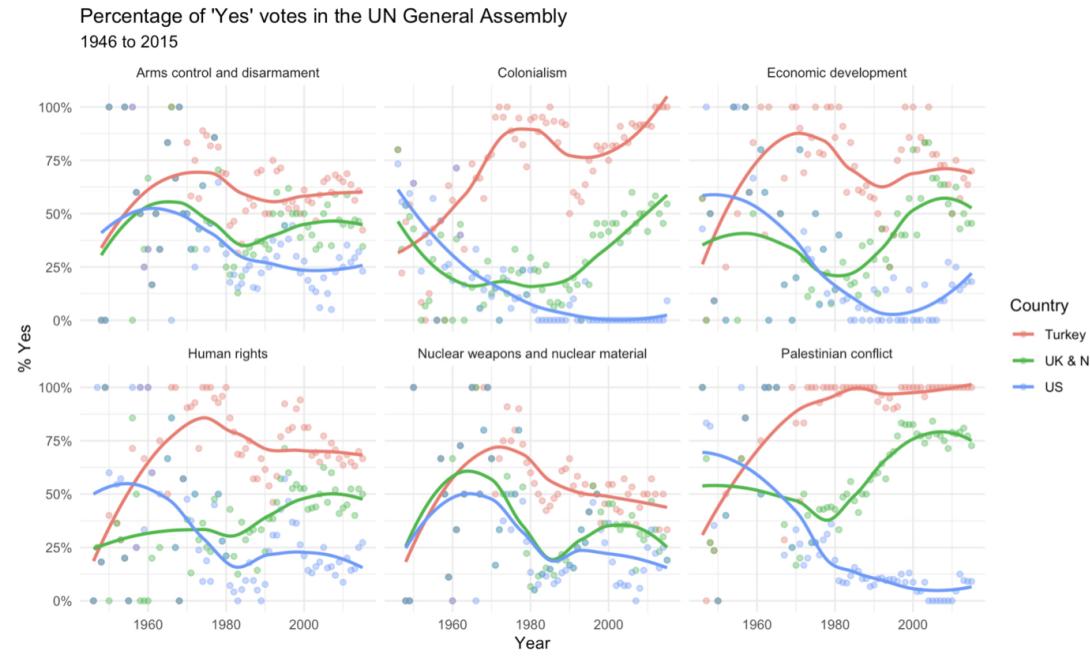
datasciencebox.org



datasciencebox.org

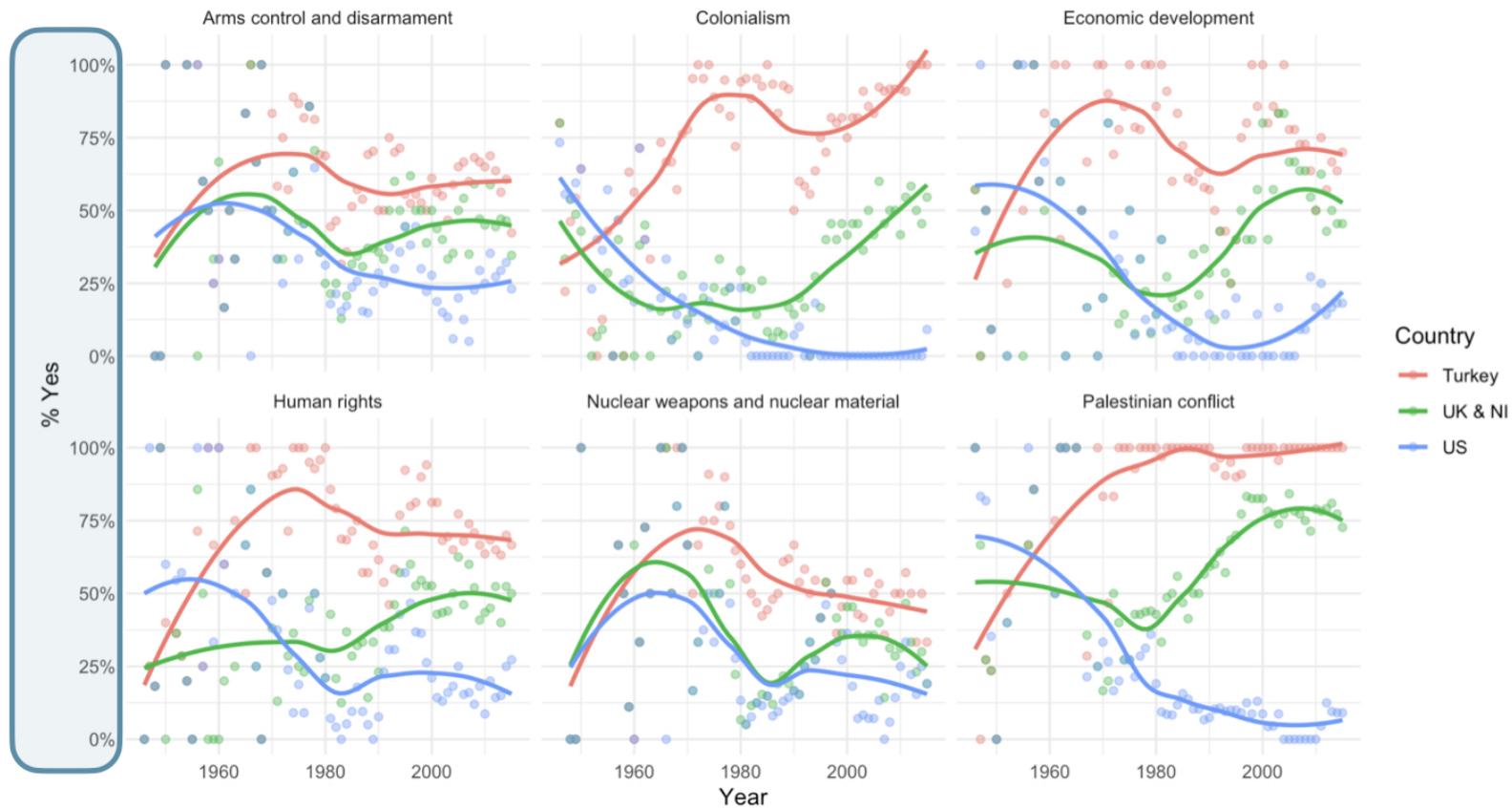
Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015



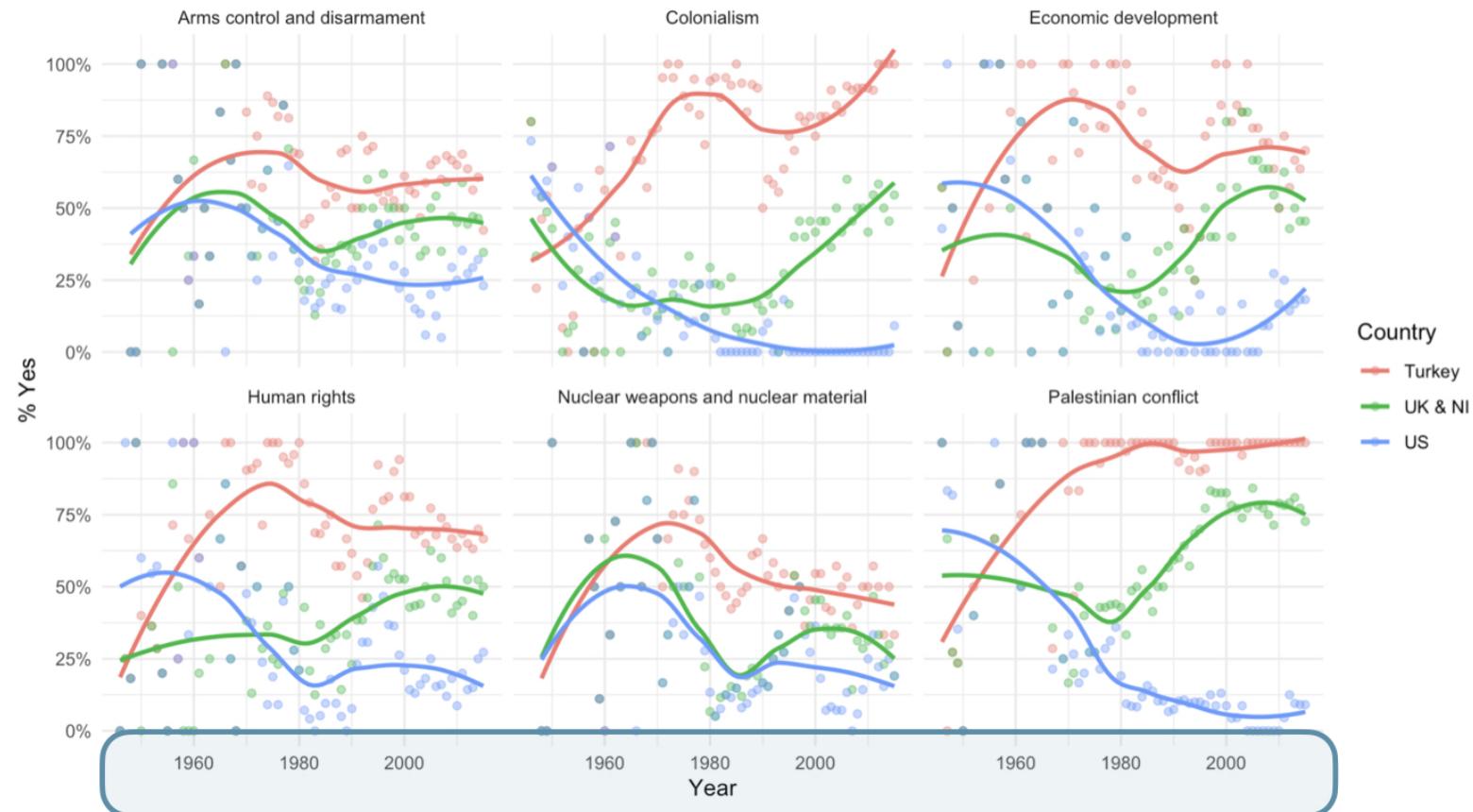


- Where can we find the issues for each visualization?
- Which countries are being visualized?
- What is our response? What is on our y-axis?
- What is our predictors? What is on the x-axis?

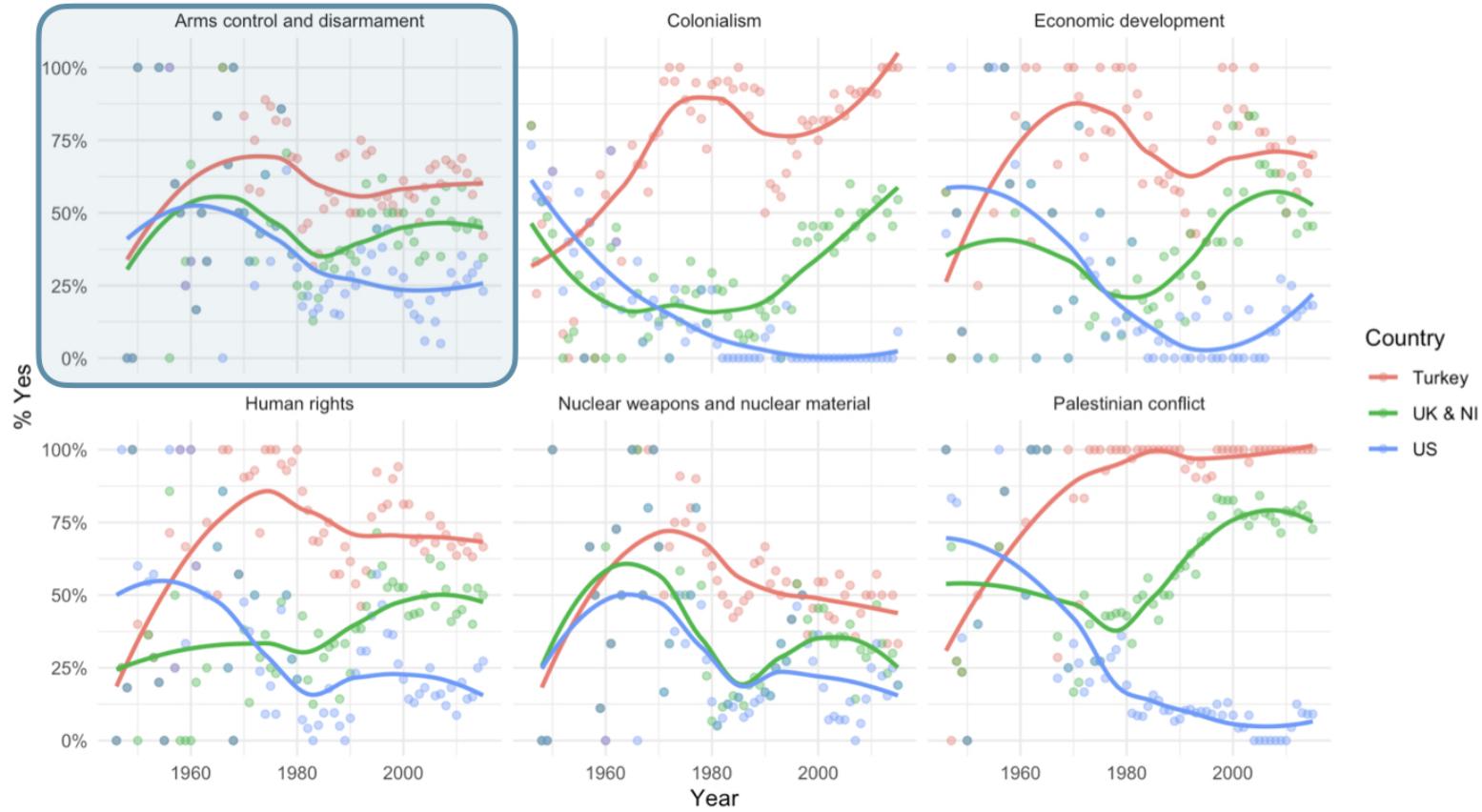
Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015



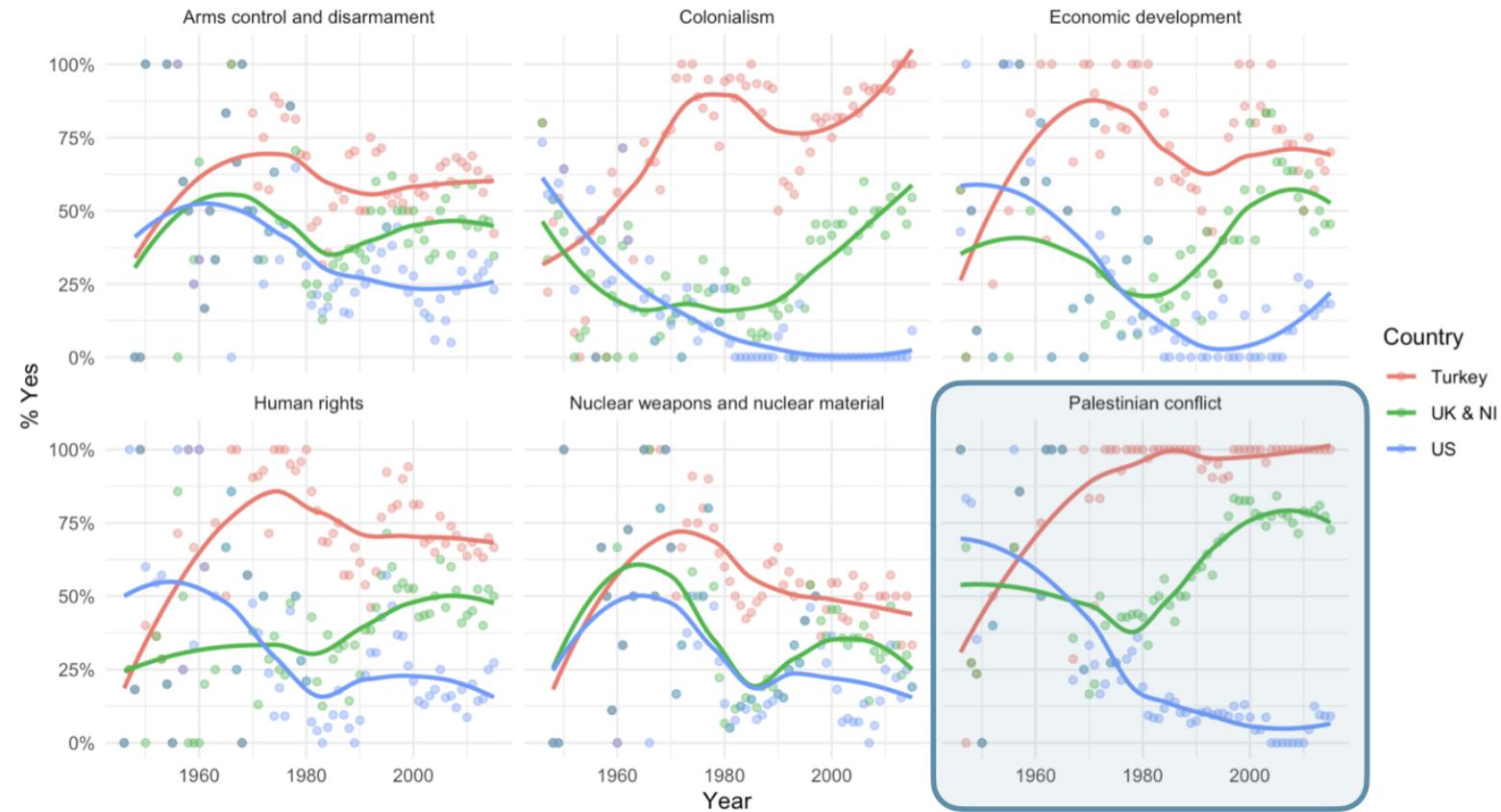
Percentage of 'Yes' votes in the UN General Assembly 1946 to 2015



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 stacked Jupyter Notebook cells. The top cell has a blue border and displays the first few rows of the 'un_votes' dataset. The middle cell has a purple border and displays the first few rows of the 'un_roll_calls' dataset. The bottom cell has a pink border and displays the first few rows of the 'un_roll_call_issues' dataset. All three cells have a 'Filter' button at the top right.

un_votes

un_roll_calls

un_roll_call_issues

Filter

rcid country_code vote

1 un_votes un_roll_calls un_roll_call_issues

2 Filter

3 rcid session importantvote date unres amend para short

4 un_votes un_roll_calls un_roll_call_issues

5 Filter

6 rcid short_name issue

7 1 3372 me Palestinian conflict

8 2 3658 me Palestinian conflict

9 3 3692 me Palestinian conflict

10 4 2901 me Palestinian conflict

11 5 3020 me Palestinian conflict

12 6 3217 me Palestinian conflict

13 7 3298 me Palestinian conflict

14 8 3429 me Palestinian conflict

15 9 3558 me Palestinian conflict

16 10 3625 me Palestinian conflict

17 11 3714 me Palestinian conflict

18 12 3368 me Palestinian conflict

19 13 3410 me Palestinian conflict

20 14 3539 me Palestinian conflict

21 15 3634 me Palestinian conflict

22 16 4880 me Palestinian conflict

23 17 4126 me Palestinian conflict

24 18 4078 me Palestinian conflict

25 19 3016 me Palestinian conflict

26 20 4290 me Palestinian conflict

27 21 4717 me Palestinian conflict

28 22 4790 me Palestinian conflict

29 23 4483 me Palestinian conflict

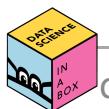
30 24 4555 me Palestinian conflict

31 25 4646 me Palestinian conflict

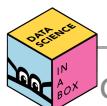
32 26 5020 me Palestinian conflict

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

```
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
```



```
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
```



```
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
```



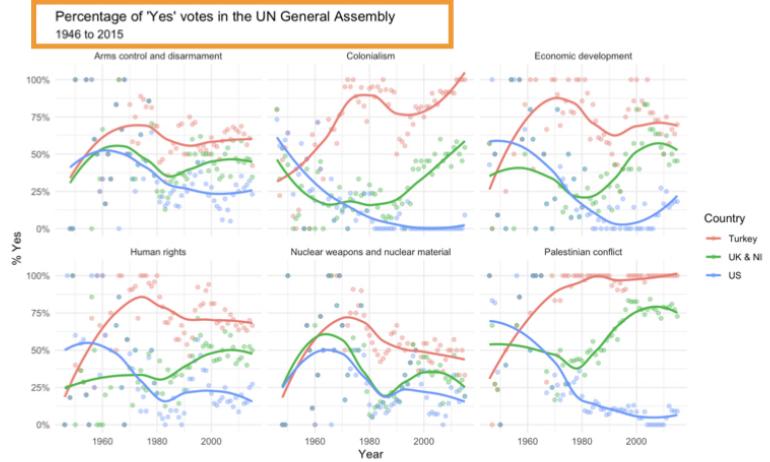
```
unvotes.Rmd x
Insert | Run | A
36 We can easily change which countries are being plotted by changing which
37 countries the code above `filter`'s for. Note that the country name should be
38 spelled and capitalized exactly the same way as it appears in the data. See
39 the [Appendix](#appendix) for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



```

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   )  

46 ````
```

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) %>%
```

academy-launch - master - RStudio

```

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 =

```

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.

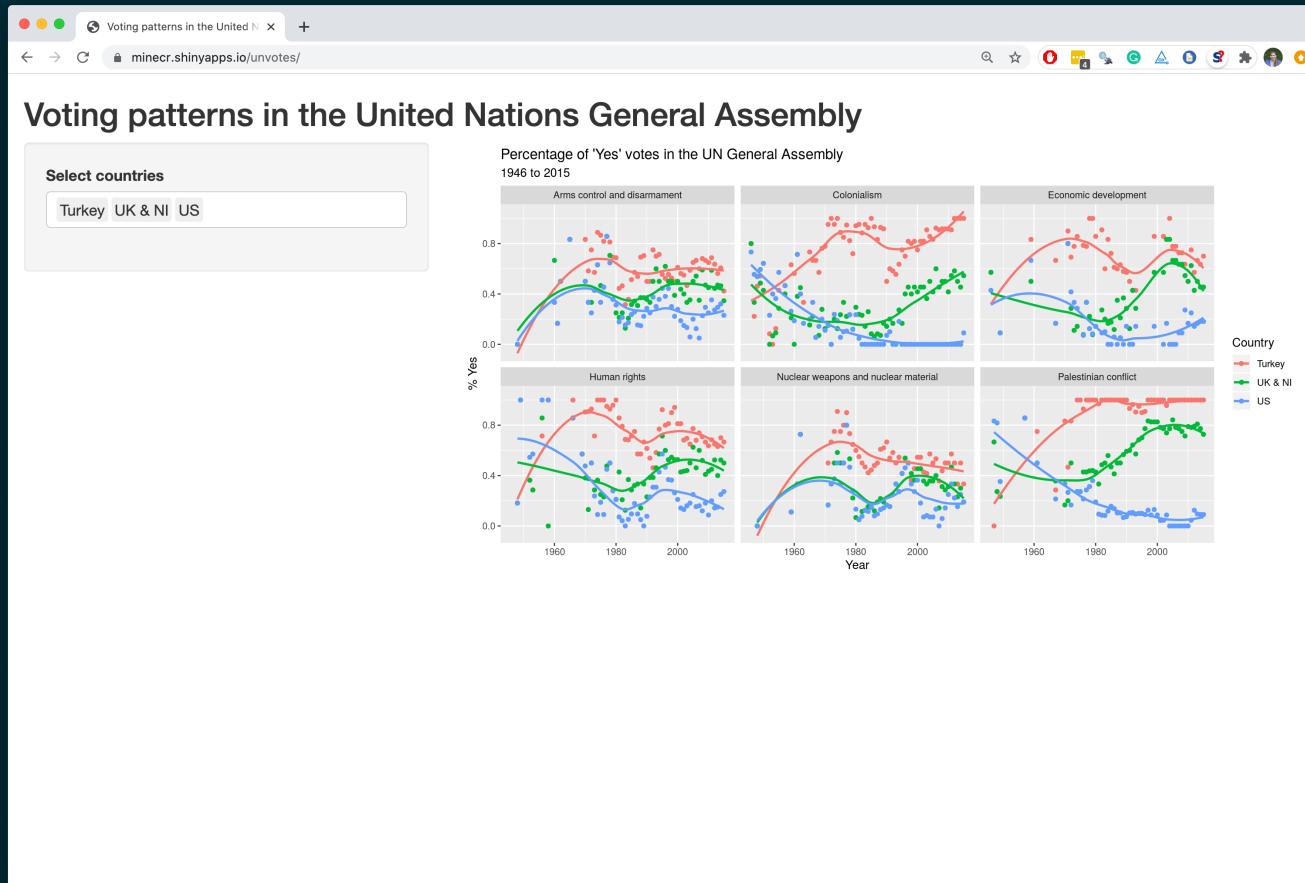
We can easily change which countries are being plotted by changing which countries the code above `filter`'s for. Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. See the [Appendix](#) for a list of the countries in the data.

```

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...



Learning goals

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

- use data carefully and ethically



Learning goals

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

- use data carefully and ethically
- gain insight from data



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**



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**



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



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"?



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 illustrates a workflow for data analysis in RStudio:

- data viewer**: A data frame titled "penguins" is displayed in the Data Viewer pane, showing columns for species, island, bill length, bill depth, flipper length, and body mass.
- arithmetic**: Basic arithmetic operations like addition and multiplication are shown in the Console.
- load package**: The palmerpenguins package is loaded using the command `> library(palmerpenguins)`.
- view data**: The penguins dataset is viewed using `> View(penguins)`.
- get help**: Help for the mean function is accessed using `> ?mean`.
- Object assignment**: An object named "x" is assigned the value 2 using `> x <- 2`.
- access variable**: The flipper length variable is accessed using `penguins$flipper_length_mm`.
- use function**: The mean function is used to calculate the mean flipper length, with and without NA values removed.
- environment**: The variable "x" is listed in the Global Environment pane with a value of 2.
- R Documentation**: The help page for the `mean` function is displayed, providing details on its usage, arguments, and examples.



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)
```



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
```



R essentials (continued)

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

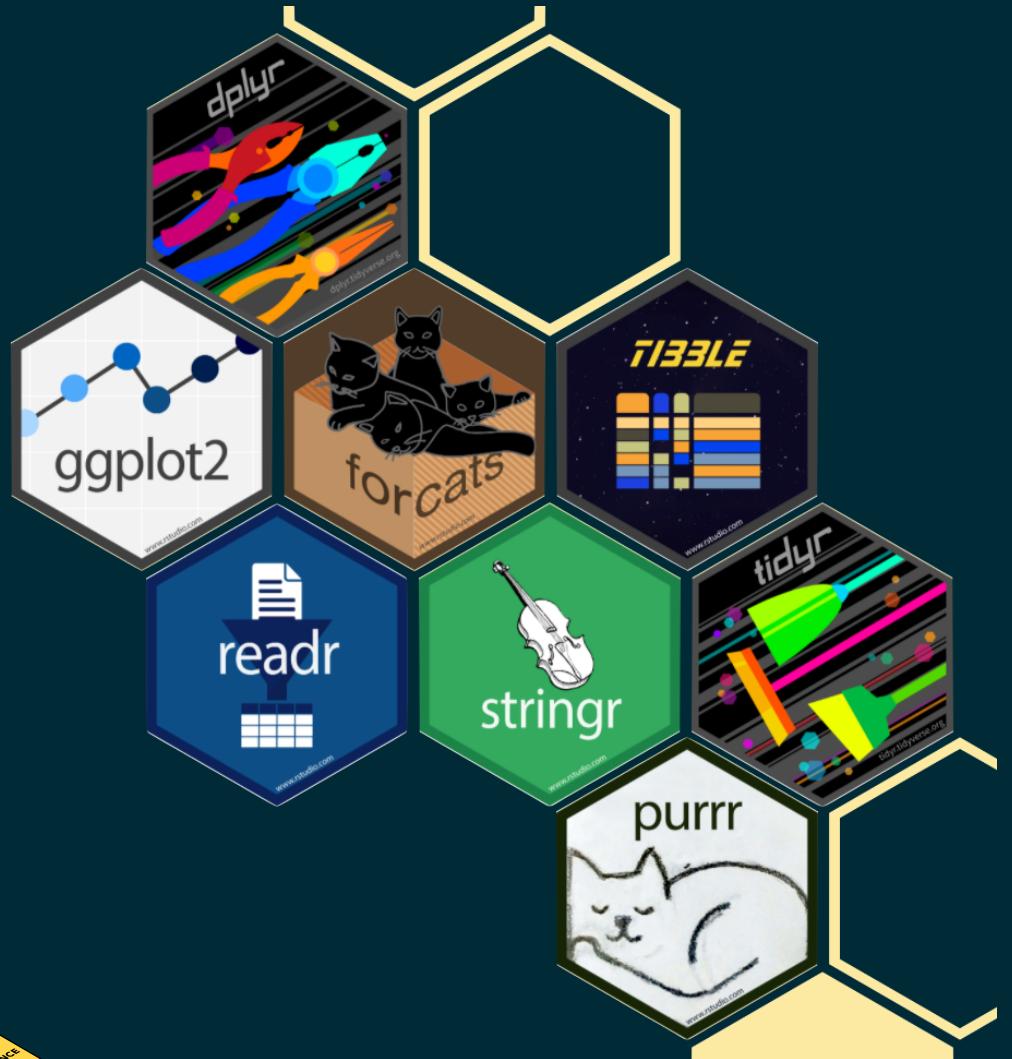
```
dataframe$var_name
```

- 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 open in the left pane and its rendered output in the right pane.

Annotations:

- knit:** A yellow arrow points to the "Knit" button in the RStudio toolbar.
- link:** A green arrow points to the URL link in the rendered output: <https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/>.
- code chunk:** A pink bracket highlights the code chunk starting at line 16: ````{r load-packages, message=FALSE}```
- yaml:** A red bracket highlights the YAML front matter at the top of the file.

R Markdown File Content (bechdel.Rmd):

```
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 ```{r load-packages, message=FALSE}  
17 library(fivethirtyeight)  
18 library(tidyverse)  
19 ...
```

Viewer Pane Content:

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

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?



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?

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 context and 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 finished report in a variety of formats including Microsoft 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 pre-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 `rmarkdown::render()` to render/knit at cmd line. Important args:

input - file to render	output_options - List of render options (as in YAML)	output_file	output_dir	params - list of params to use
------------------------	--	-------------	------------	--------------------------------

Embed code with knitr syntax

INLINE CODE
Insert with ``r <code>``. 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 `knitr::knit(r)``
````{r echo=TRUE}` `getRVersion()` `````

**GLOBAL OPTS**  
Set with `knitr::opts_chunk$set(...)`

Markdown Quick Reference  
Help -> Markdown Quick Reference

Link on Moodle and [HERE](#)

**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.com`



# 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

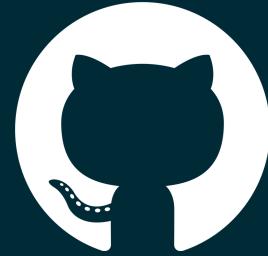


[datasciencebox.org](http://datasciencebox.org)

# Git and GitHub

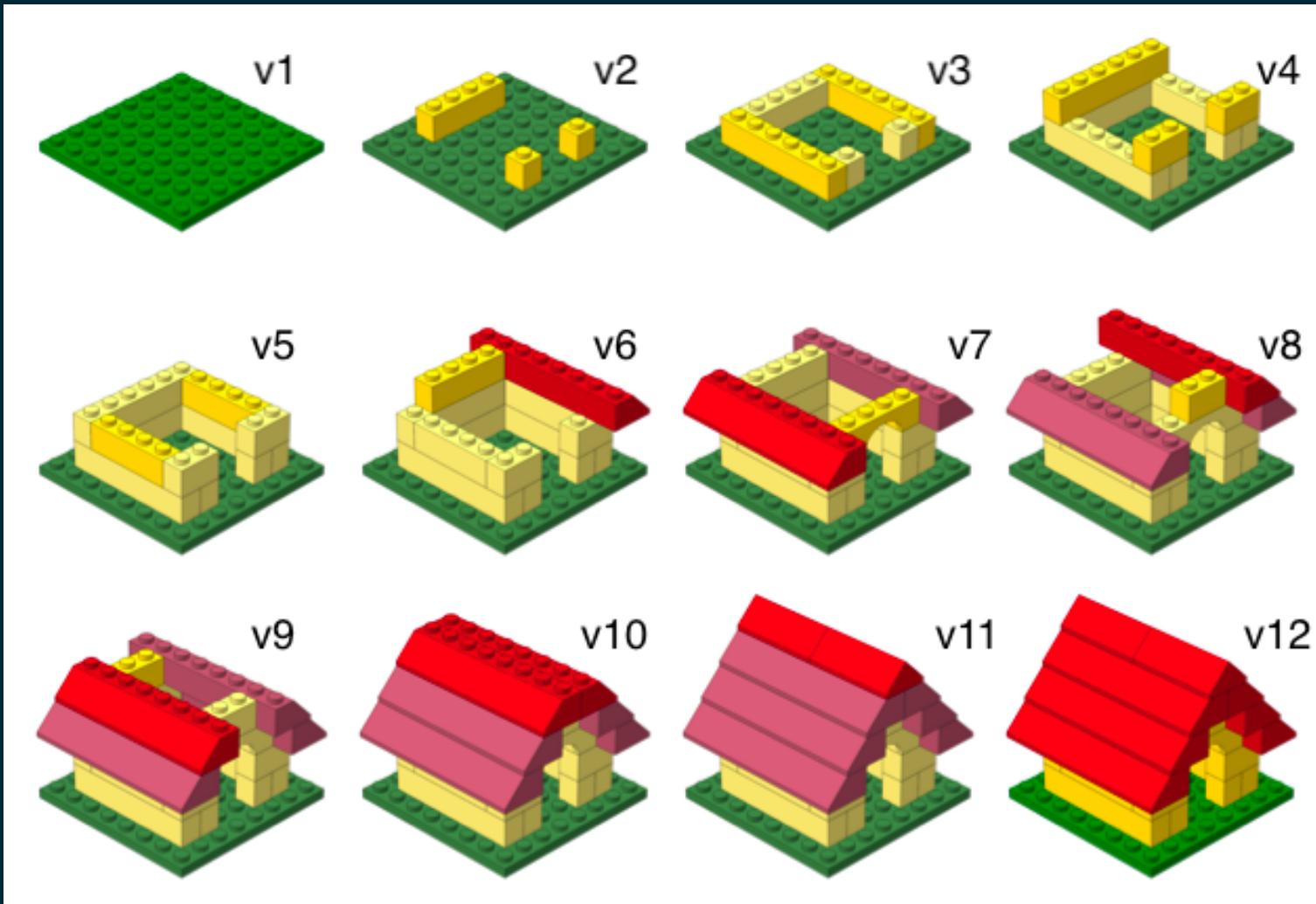


- Git is a version control system -- like “Track Changes” features from Microsoft Word, on steroids
- It's not the only version control system, but it's a very popular one

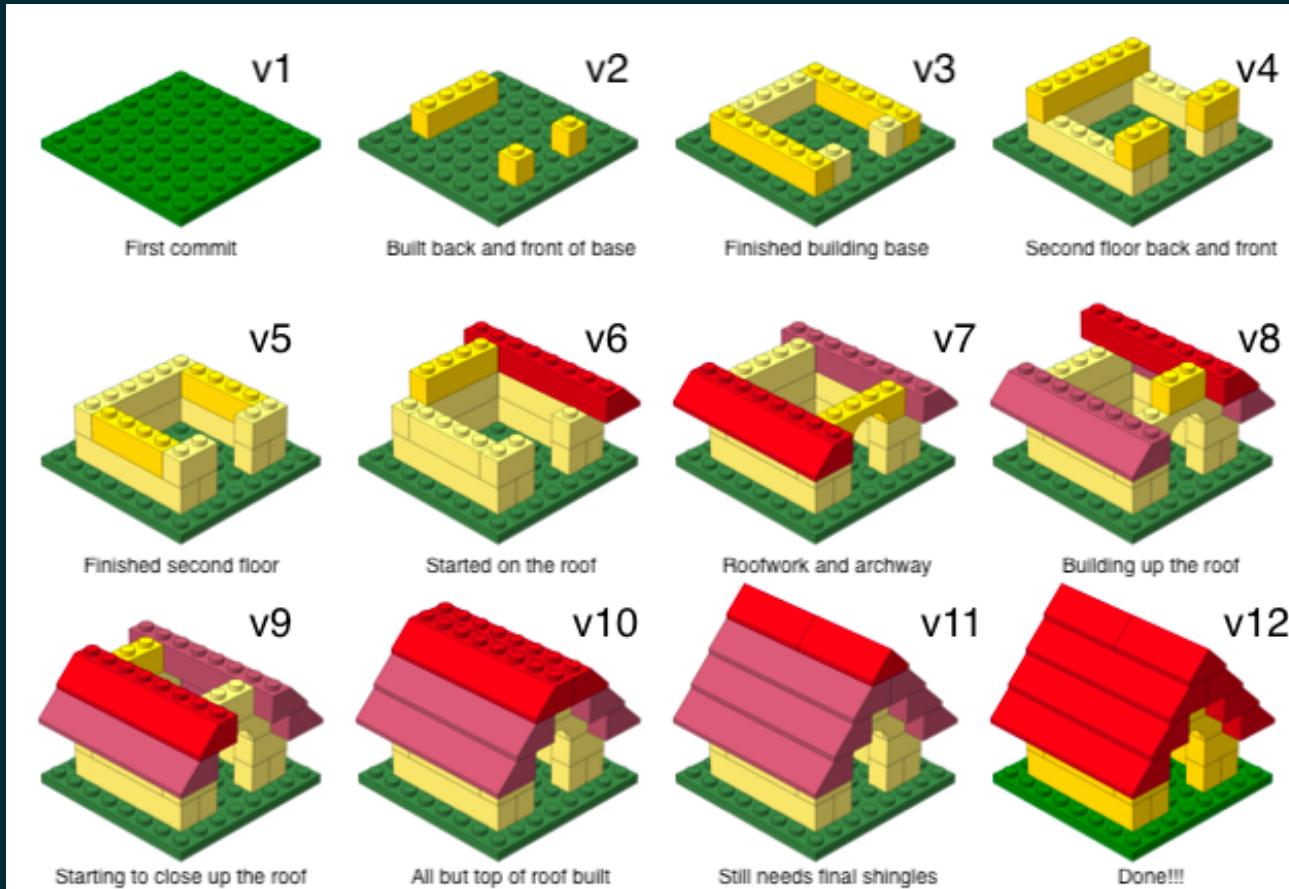


- GitHub is the home for your Git-based projects on the internet -- like DropBox but much, much better
- We will use GitHub as a platform for web hosting and collaboration (and as our course management system!)

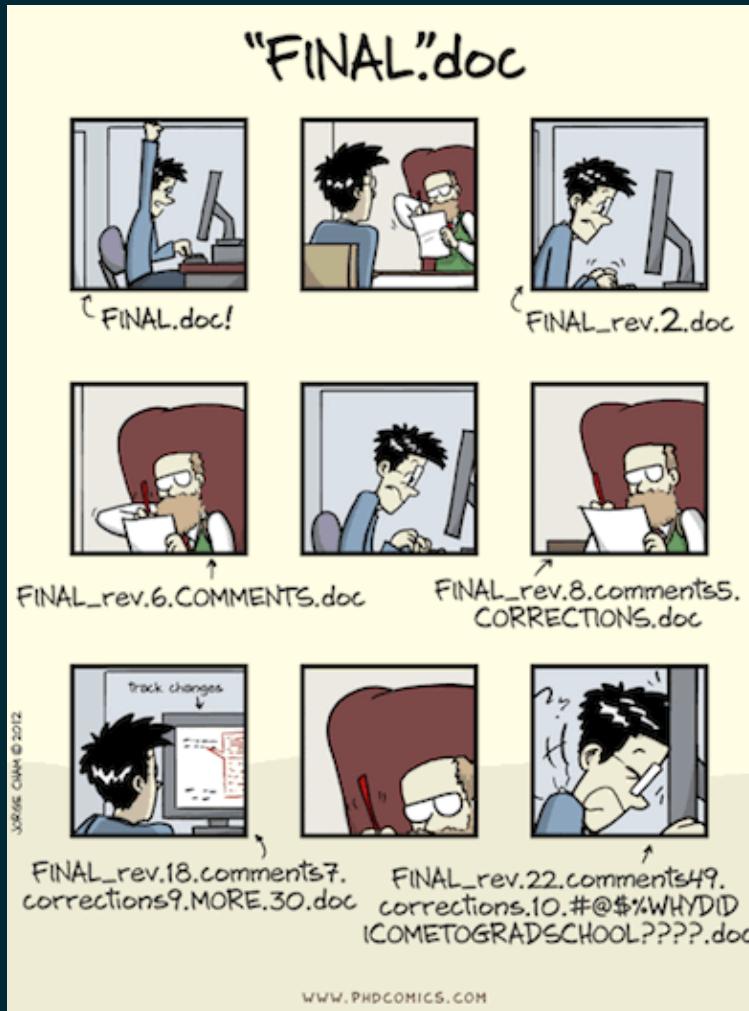
# Versioning



# 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.



# 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.



# 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](http://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!†*

<sup>†</sup>Just like a real data scientist!

