Welcome!

Data Science in a Box

datasciencebox.org

Modified by Tyler George

Hello world!

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.

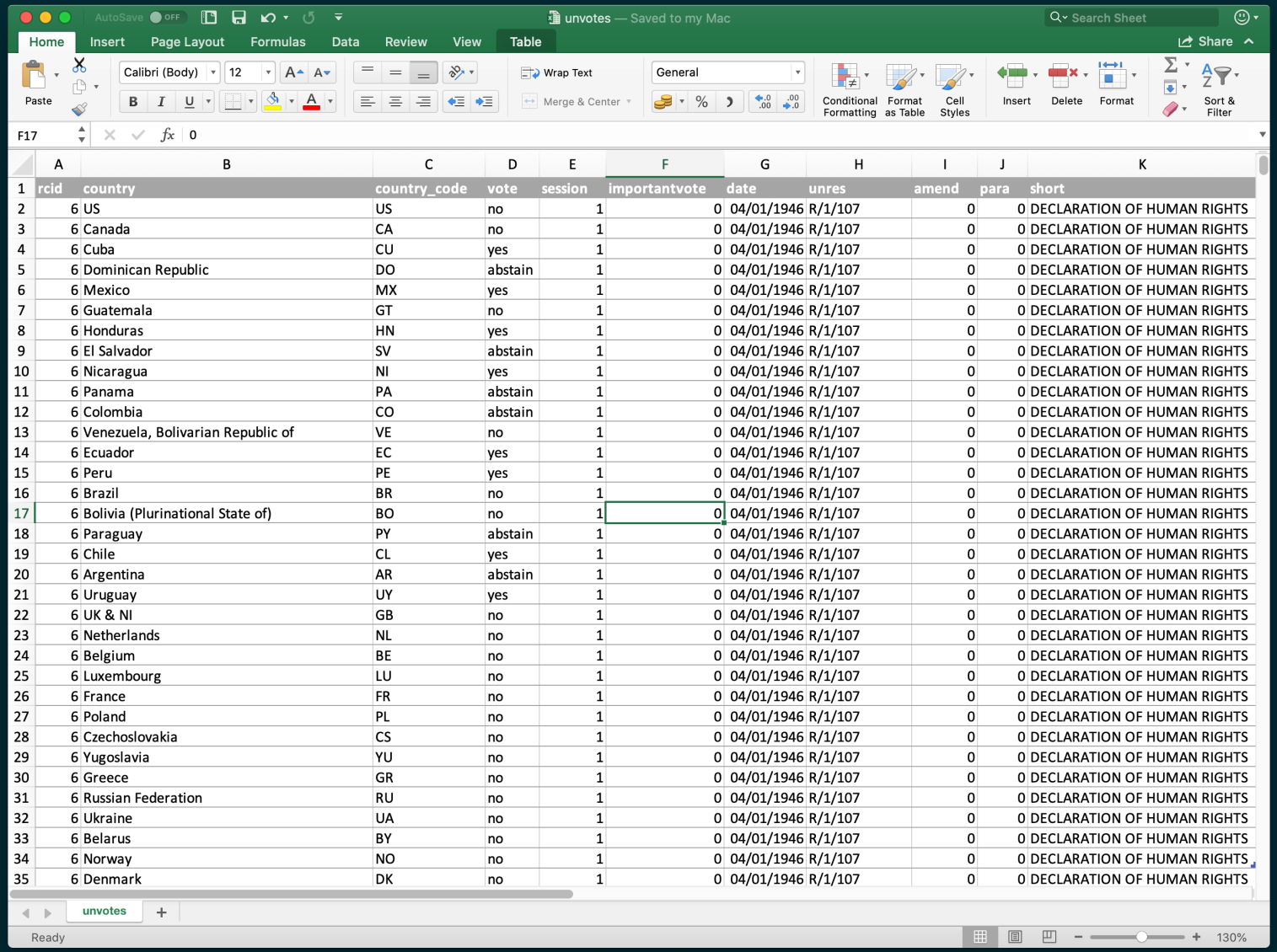
Q - What computing language will we learn?

A - R.

Q: Why not language X?

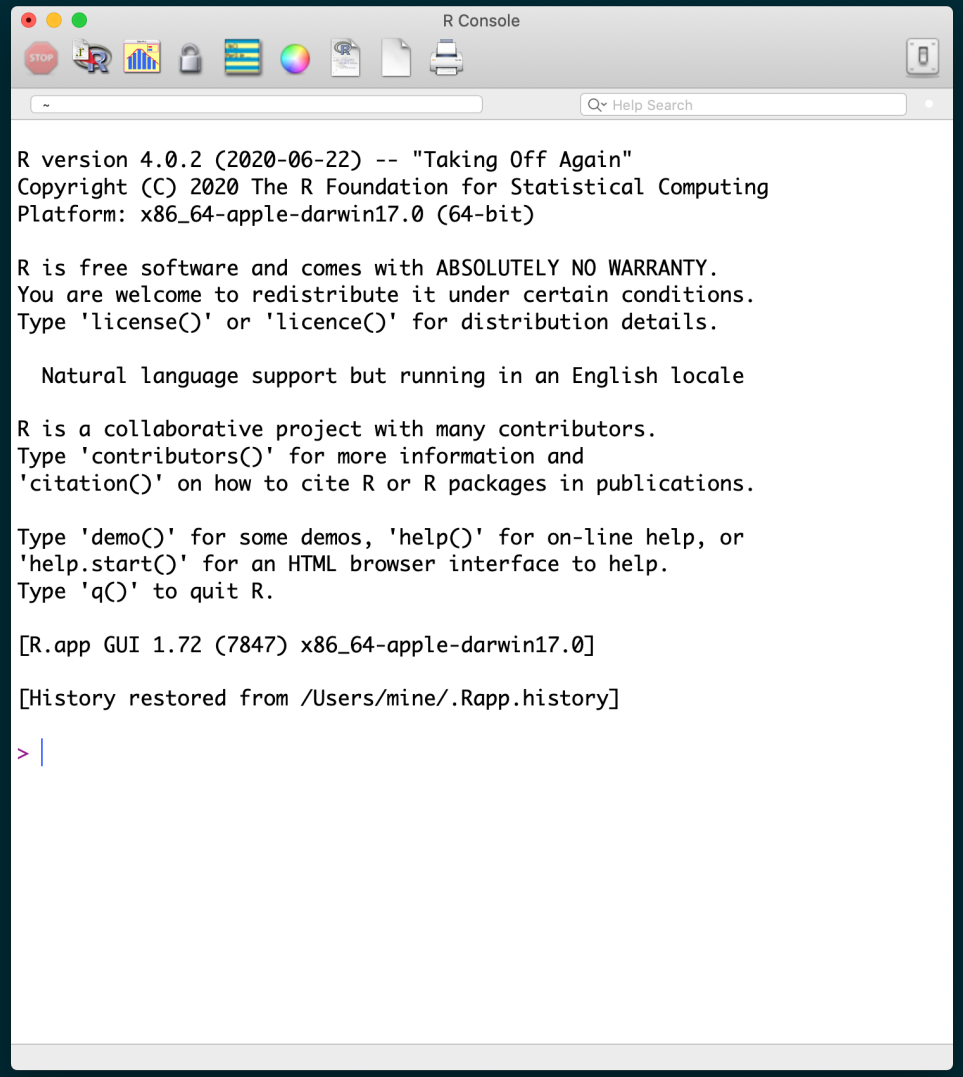
A: We can discuss that over  .

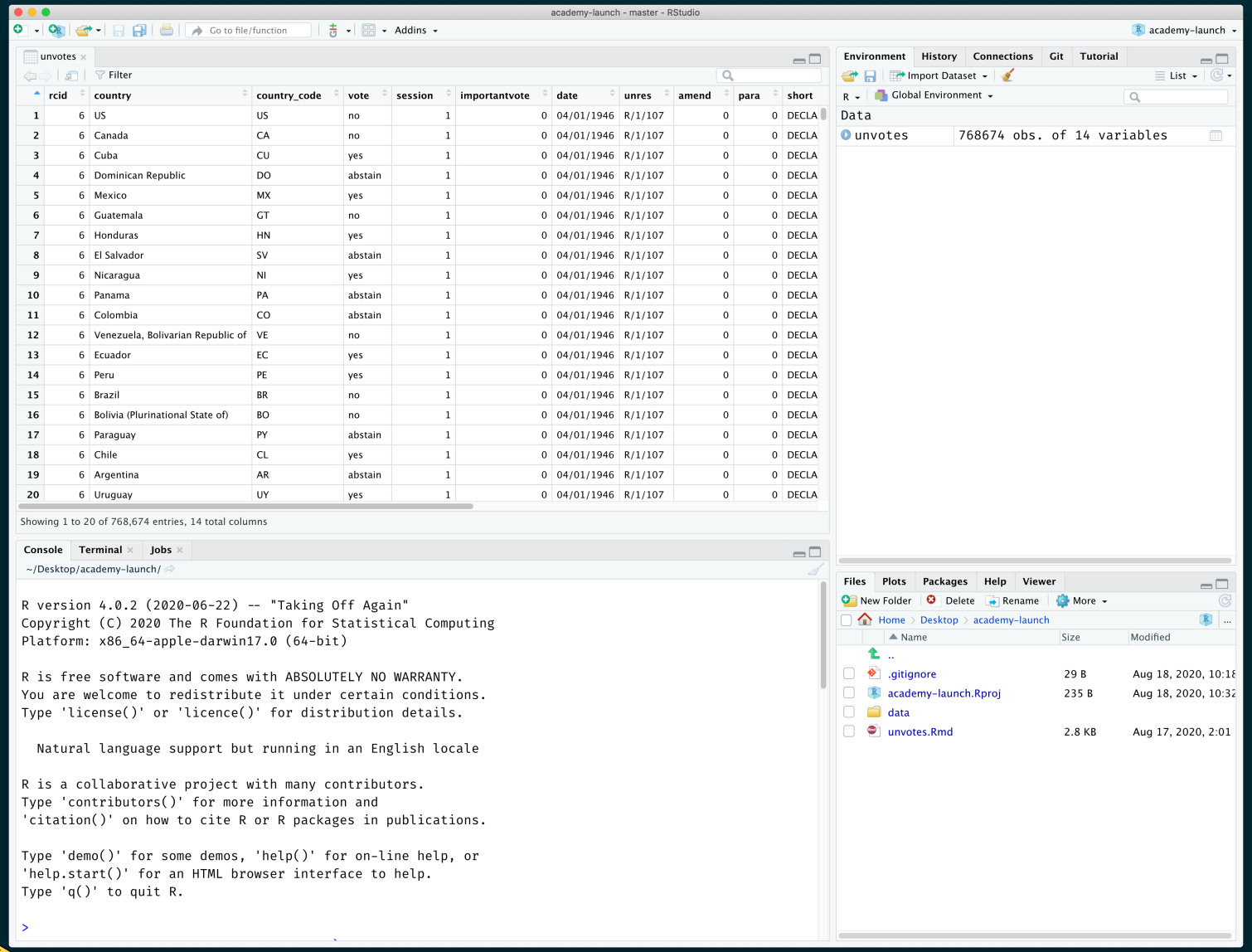
Software



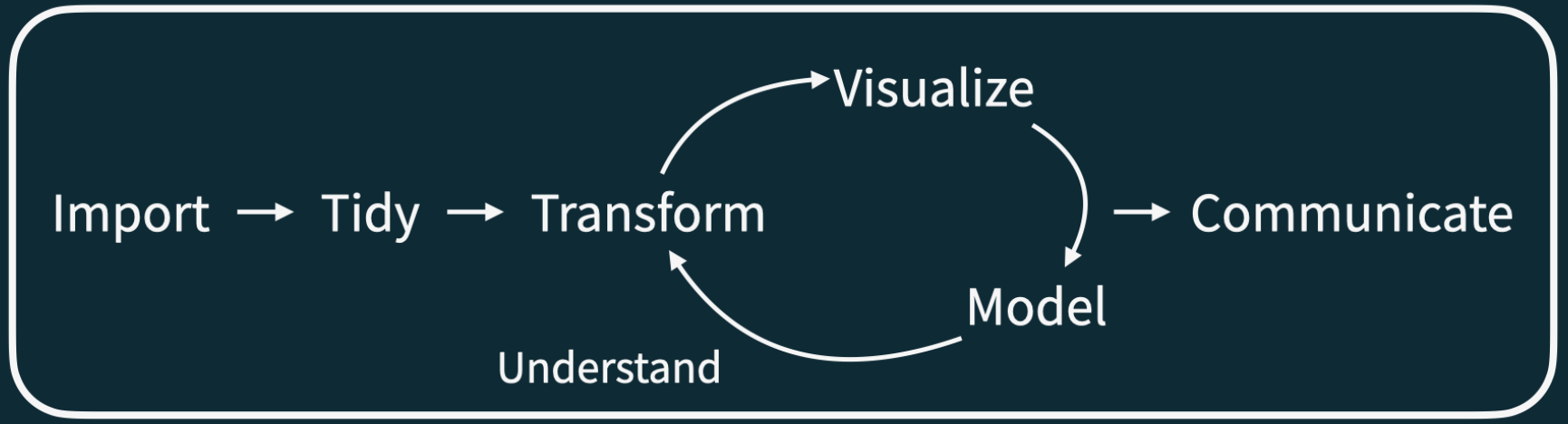
unvotes — Saved to my Mac

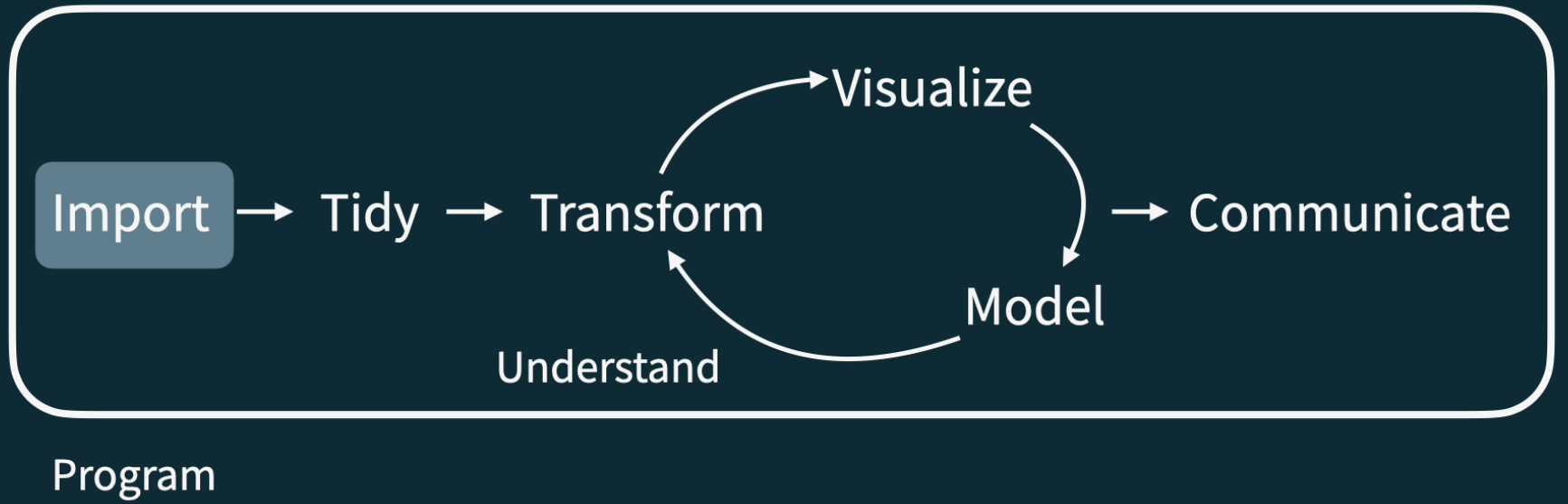
	A	B	C	D	E	F	G	H	I	J	K
1	rcid	country	country_code	vote	session	importantvote	date	unres	amend	para	short
2	6	US	US	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
3	6	Canada	CA	no	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF HUMAN RIGHTS
4	6	Cuba	CU	yes	1	0	04/01/1946	R/1/107	0	0	DECLARATION OF 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

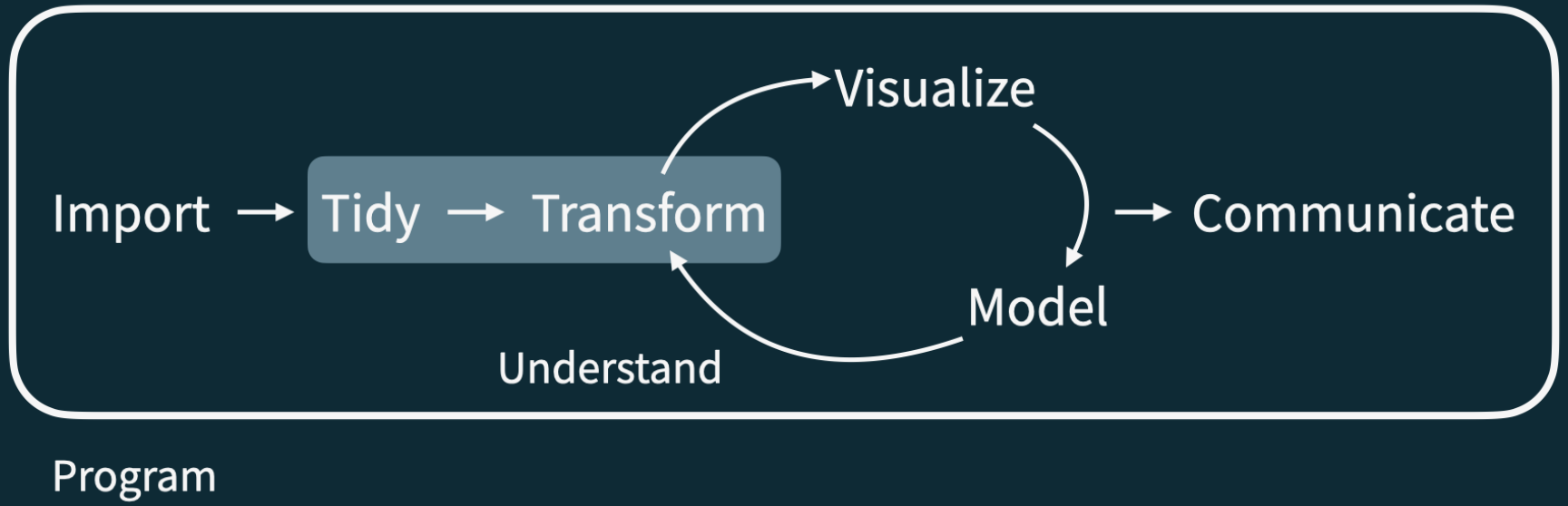

R Console
~
Help Search
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type '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]
>

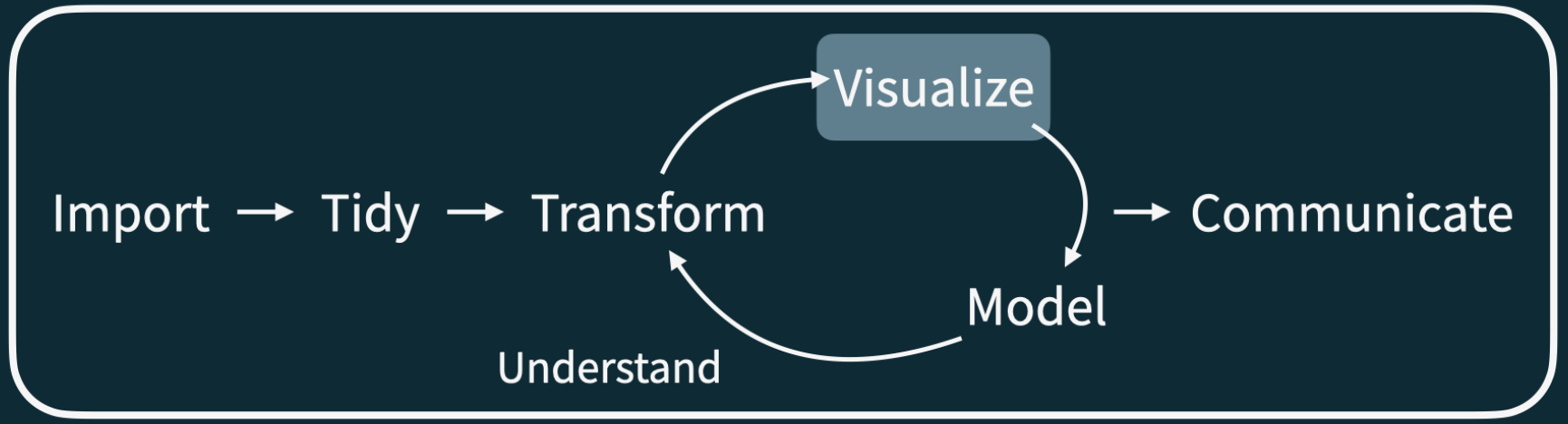

academy-launch - master - RStudio
Go to file/function
Addins
academy-launch
unvotes
Filter
Environment
History
Connections
Git
Tutorial
Import Dataset
List
R
Global Environment
Data
unvotes
768674 obs. of 14 variables
Showing 1 to 20 of 768,674 entries, 14 total columns
Console
Terminal
Jobs
~/Desktop/academy-launch/
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.
>
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Desktop
academy-launch
Name
Size
Modified
..
.gitignore
29 B
academy-launch.Rproj
235 B
data
unvotes.Rmd
2.8 KB
Aug 17, 2020, 2:01

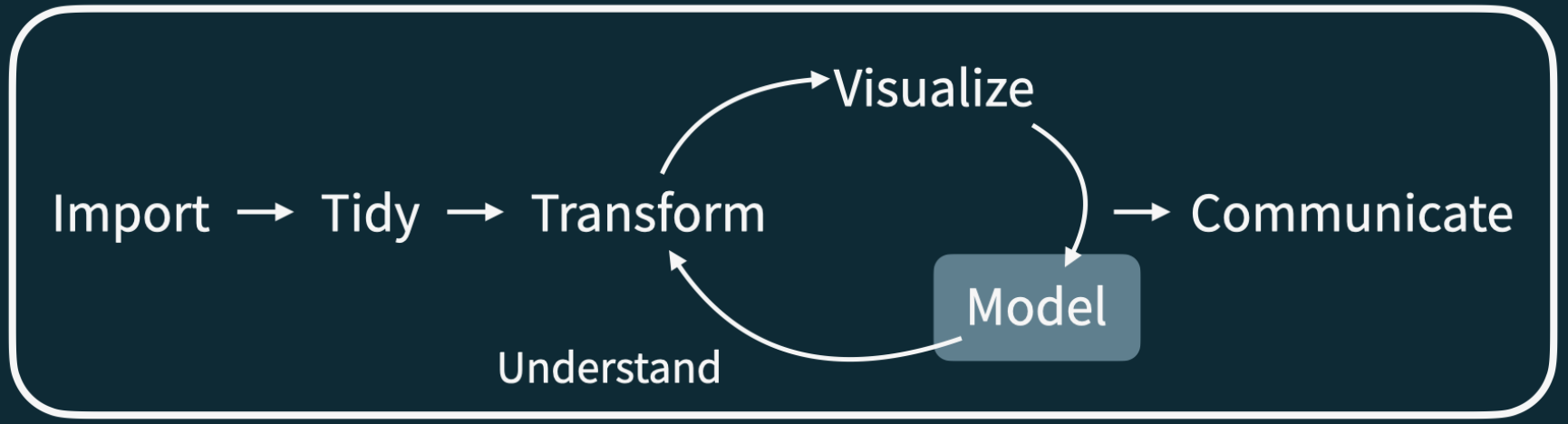
Data science life cycle

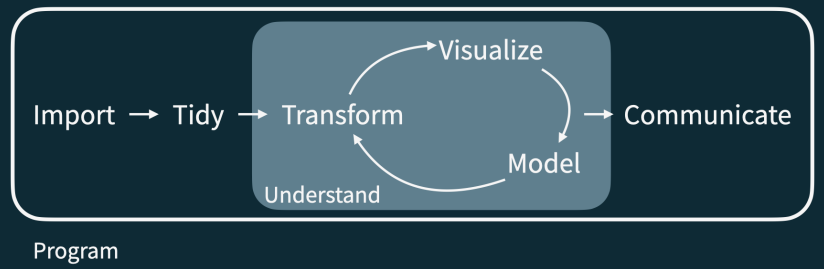

Visualize
Import
Tidy
Transform
Communicate
Model
Understand
Program

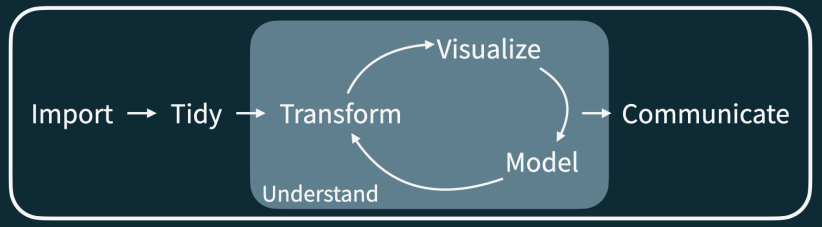

Visualize
Import
Tidy
Transform
Communicate
Model
Understand
Program


Visualize
Import
Tidy
Transform
Communicate
Model
Understand
Program


Visualize
Import
Tidy
Transform
Communicate
Model
Understand
Program


Visualize
Import
Tidy
Transform
Communicate
Model
Understand
Program


Import
Tidy
Transform
Visualize
Model
Understand
Communicate
Program



Google Trend Index

Trend index

Travel

Date range

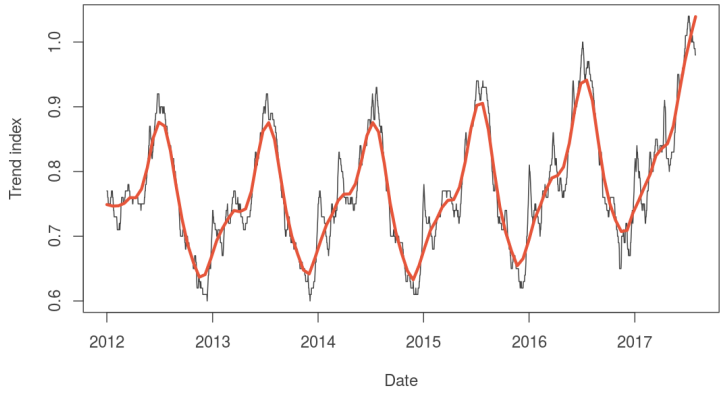
2012-01-01 to 2017-07-31

Overlay smooth trend line

Smoother span:

0.05

Higher values give more smoothness.



The Google Travel Index tracks queries related to airlines, hotels, beach, southwest, las vegas, flights, etc. The index is set to 1.0 on January 1, 2004 and is calculated only for US search traffic.

Source: Google Domestic Trends

Google Trend Index

Trend index

Travel

Date range

2012-01-01 to 2017-07-31

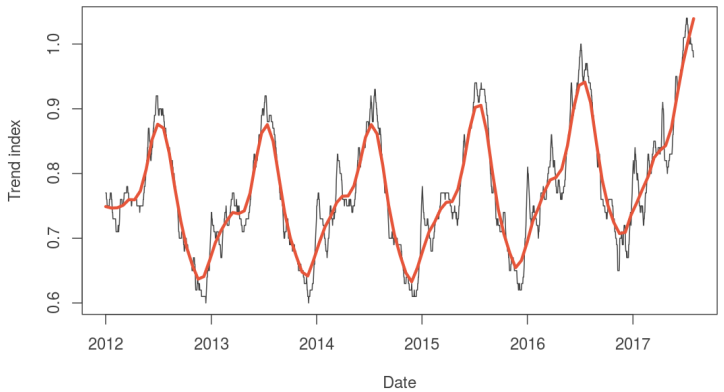
Overlay smooth trend line

Smoother span:

0.05 1

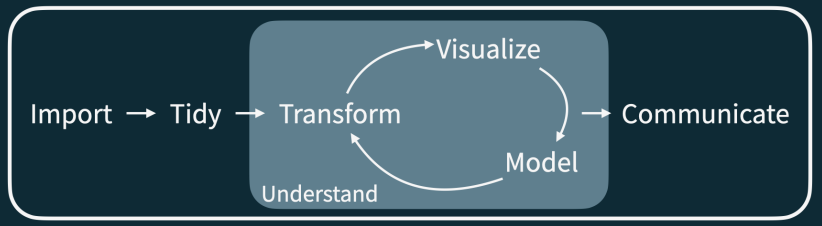
0.01 0.11 0.21 0.31 0.41 0.51 0.61 0.71 0.81 0.91 1

Higher values give more smoothness.



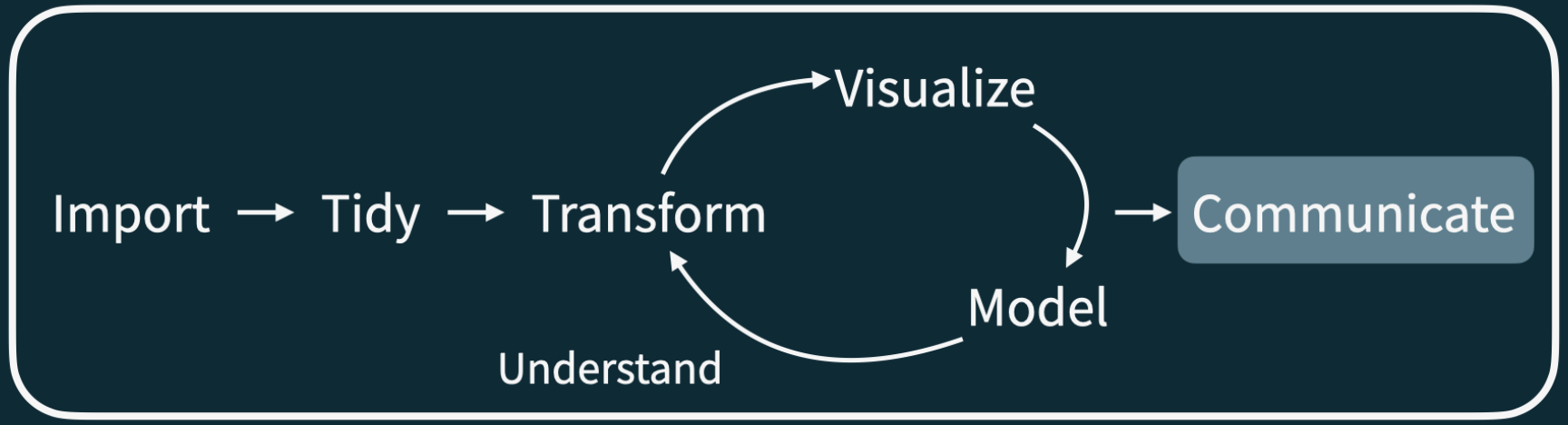
The Google Travel Index tracks queries related to airlines, hotels, beach, southwest, las vegas, flights, etc. The index is set to 1.0 on January 1, 2004 and is calculated only for US search traffic.

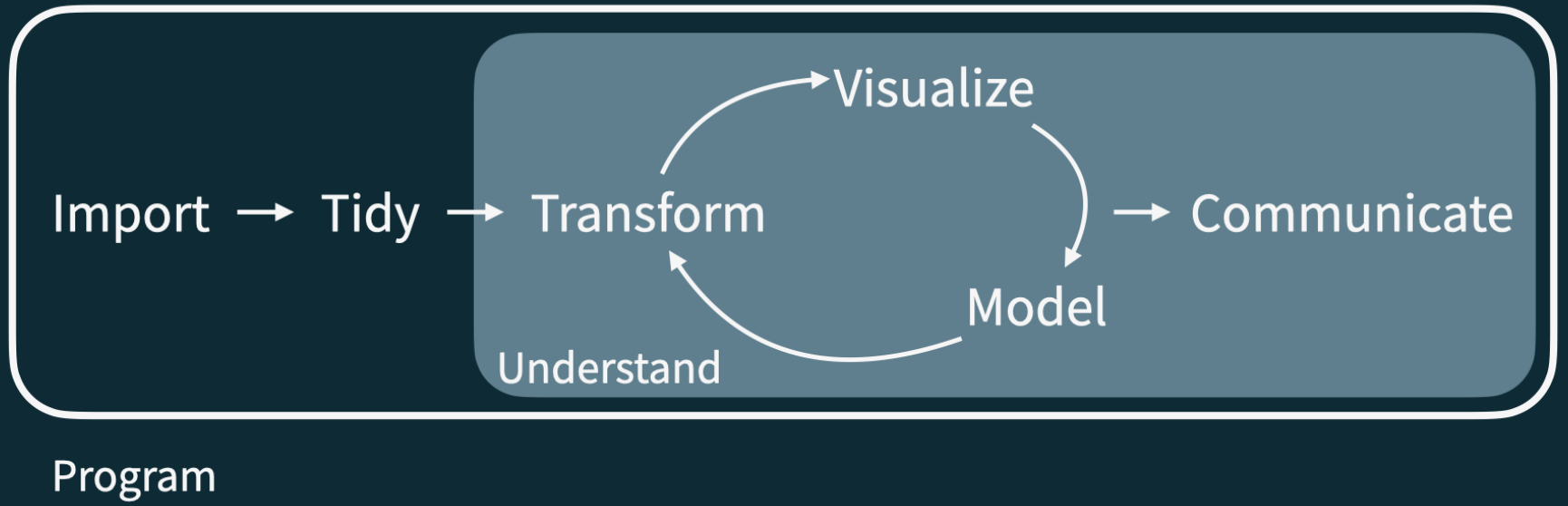
Source: Google Domestic Trends

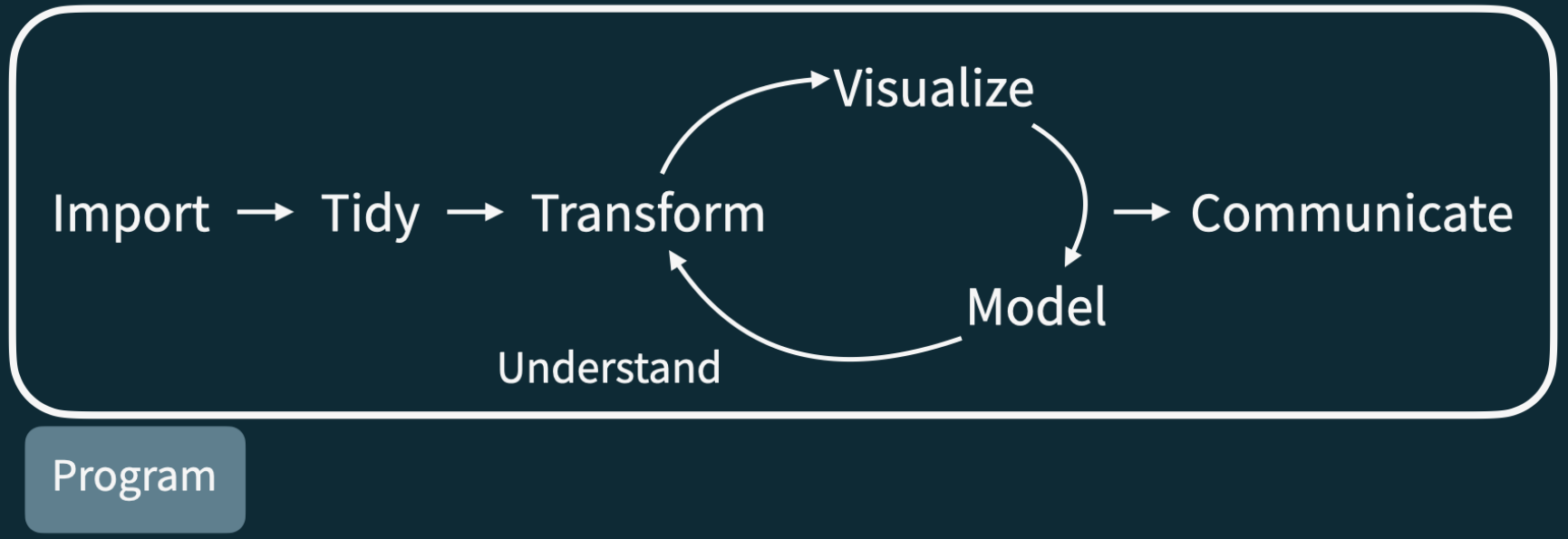


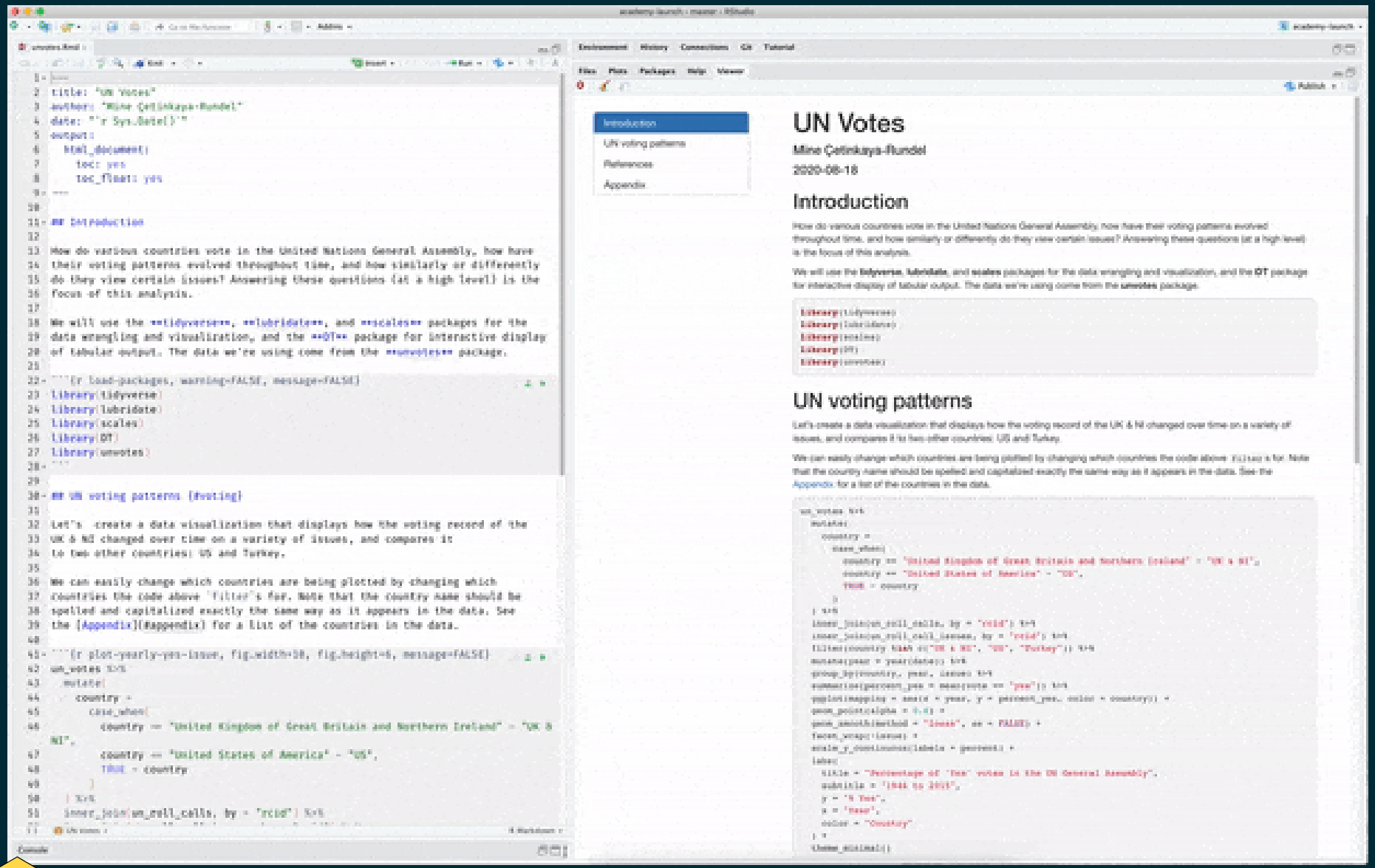
Program

```
## # 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 ...               ...
```


Visualize
Import
Tidy
Transform
Communicate
Model
Understand
Program

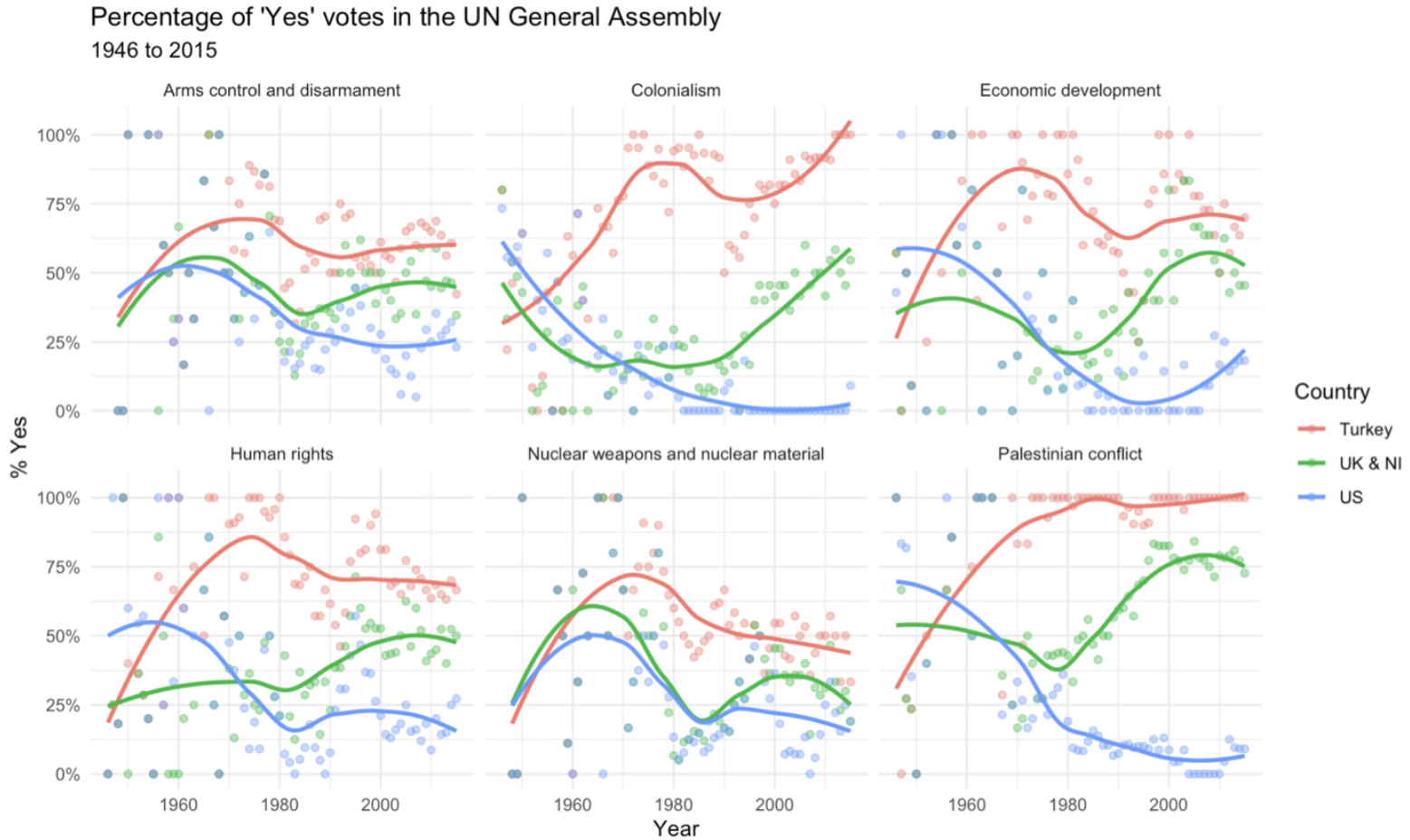

Import
Tidy
Transform
Visualize
Model
Understand
Communicate
Program

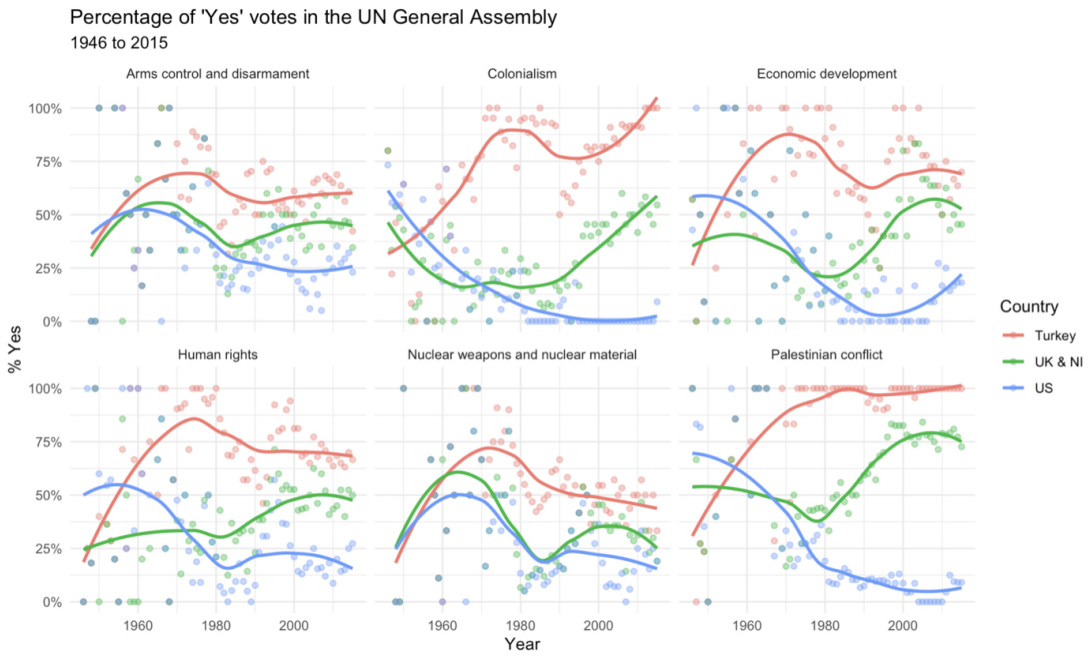

Visualize
Import
Tidy
Transform
Communicate
Model
Understand
Program


title: "UN Votes"
author: "Mine Çetinkaya-Rundel"
date: "`r Sys.Date()`"
output:
  html_document:
    toc: yes
    toc_float: yes

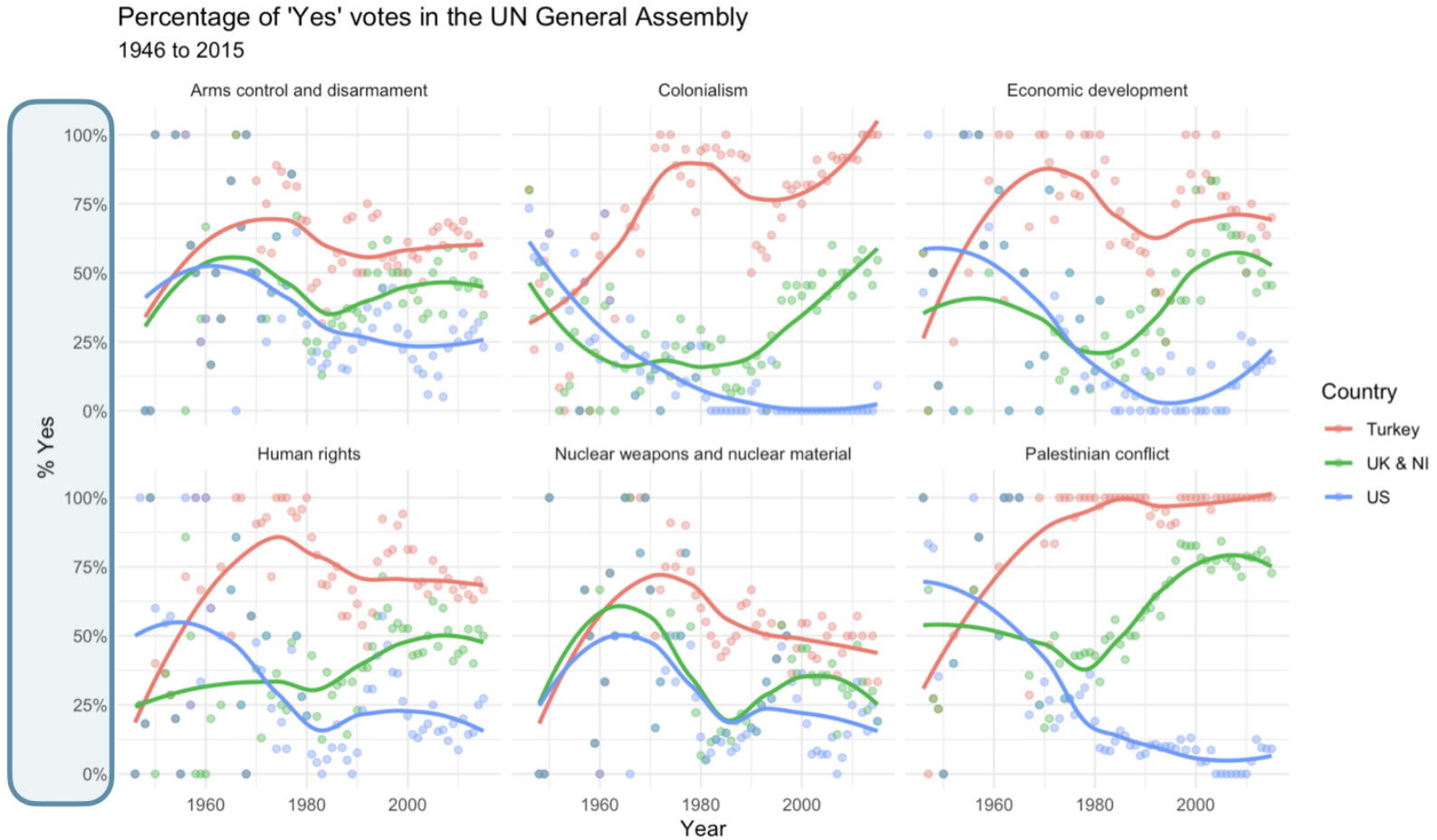
Introduction
How do various countries vote in the United Nations General Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do they view certain issues? Answering these questions (at a high level) is the focus of this analysis.
We will use the **tidyverse**, **lubridate**, and **scales** packages for the data wrangling and visualization, and the **DT** package for interactive display of tabular output. The data we're using come from the **unvotes** package.
```{r load-packages, warning=FALSE, message=FALSE}
library(tidyverse)
library(lubridate)
library(scales)
library(DT)
library(unvotes)
```
UN voting patterns {#voting}
Let's create a data visualization that displays how the voting record of the UK & NI changed over time on a variety of issues, and compares it to two other countries: US and Turkey.
We can easily change which countries are being plotted by changing which countries the code above `filter`s for. Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. See the [Appendix](#appendix) for a list of the countries in the data.
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
  mutate(
    country =
      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") %>%
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) %>%
  summarize(percent_yes = mean(vote == "yes")) %>%
  ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
  facet_wrap(~issue) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "Percentage of 'Yes' votes in the UN General Assembly",
    subtitle = "1946 to 2015",
    y = "% Yes",
    x = "Year",
    color = "Country"
  ) +
  theme_minimal()
```

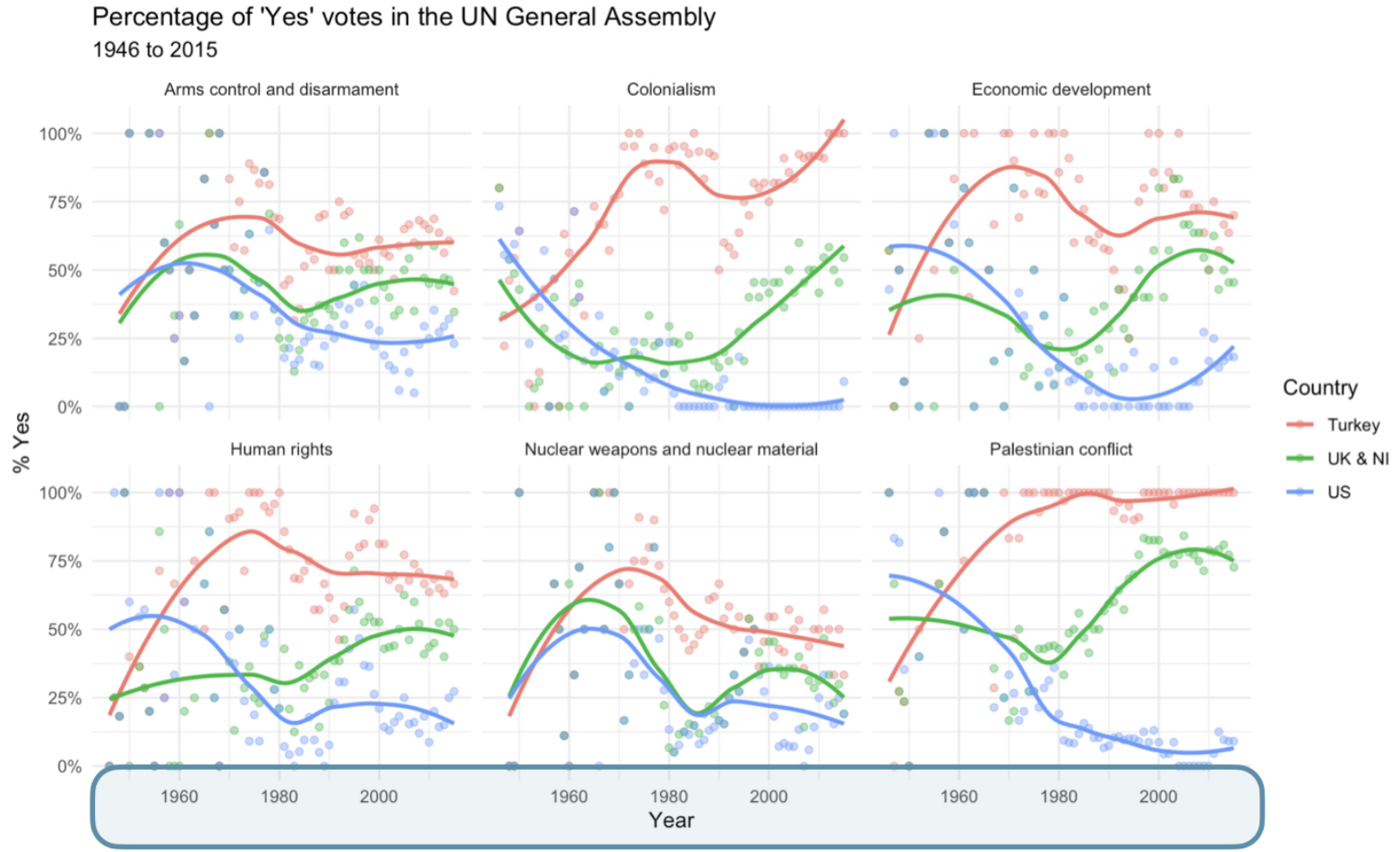
Let's dive in!

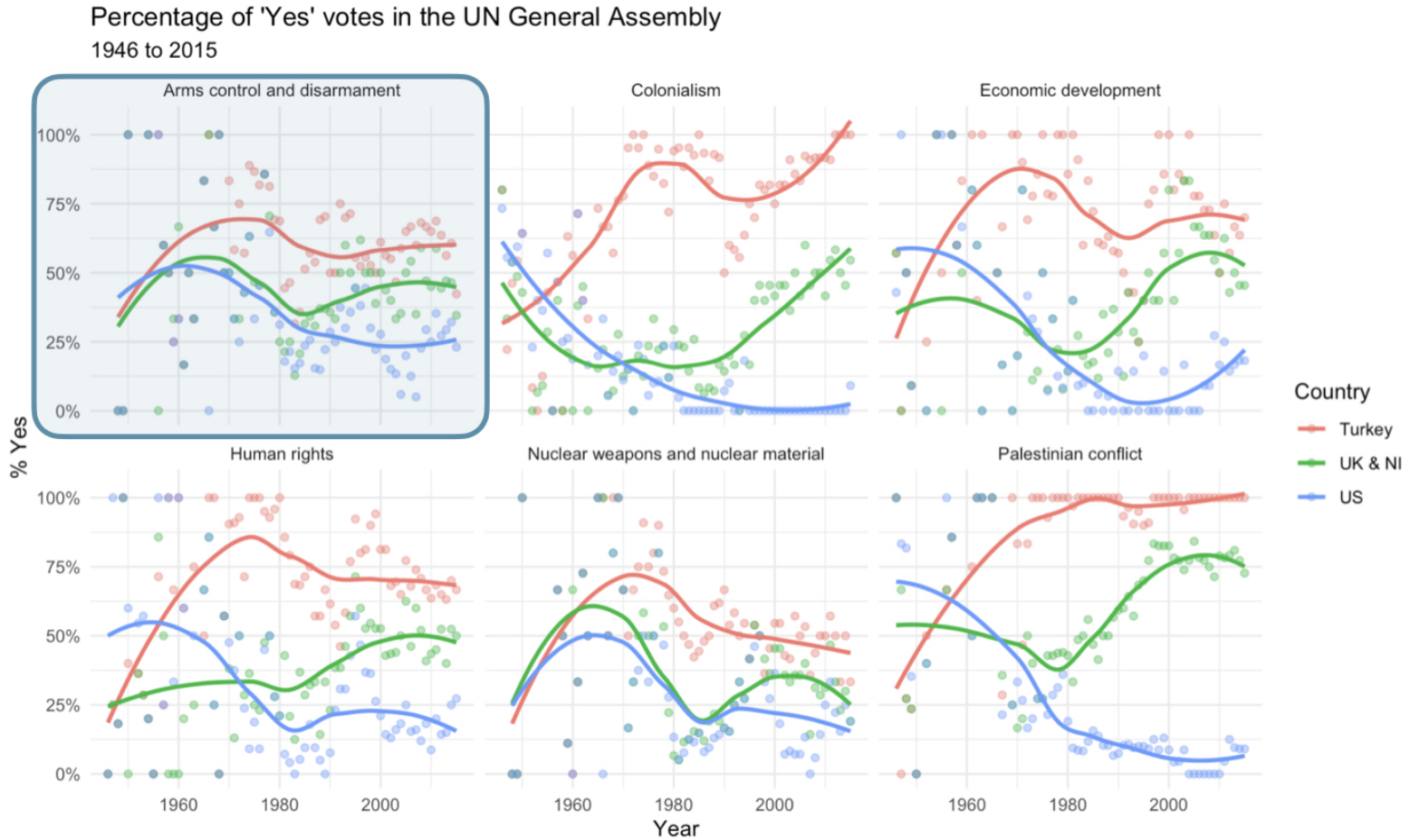

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US

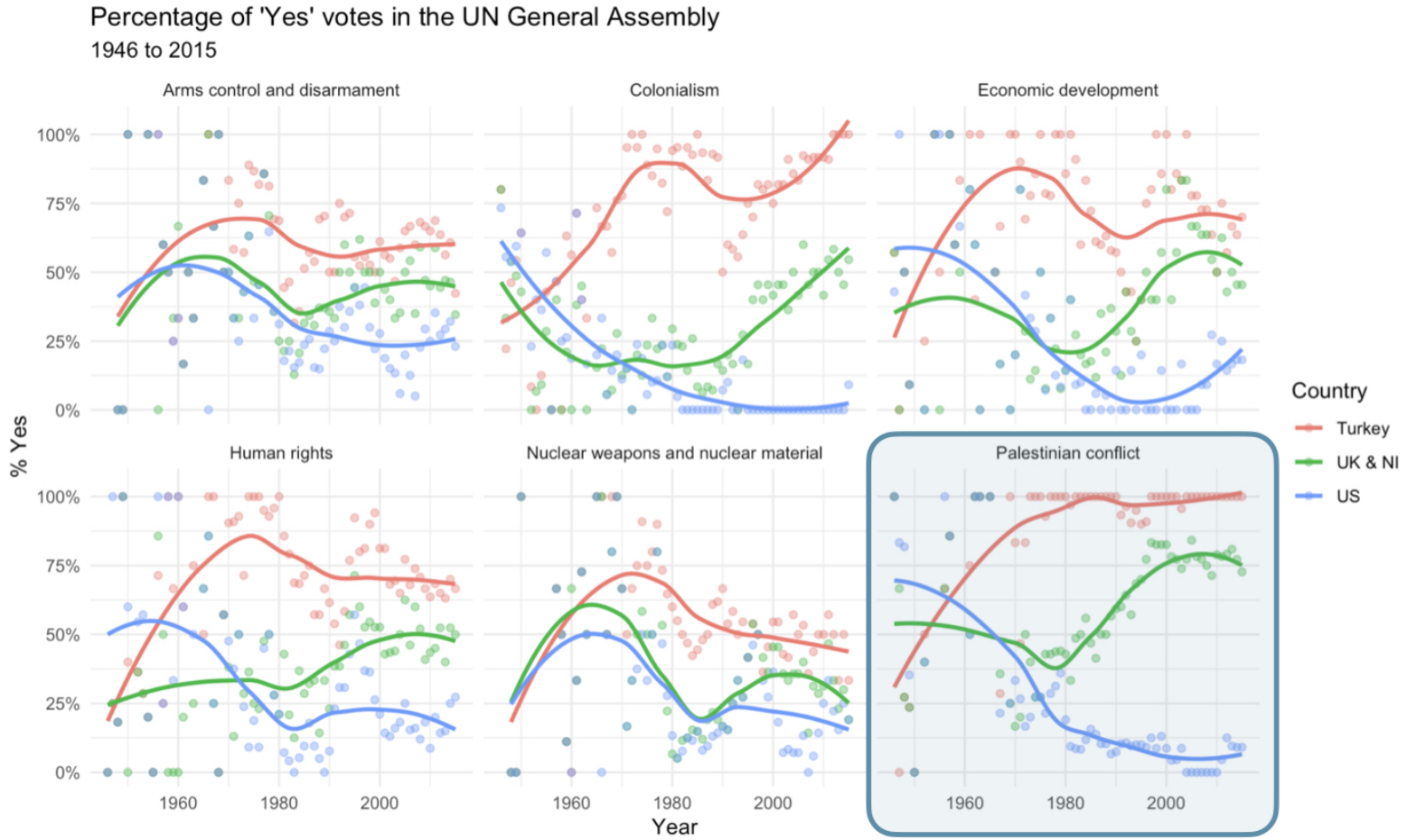


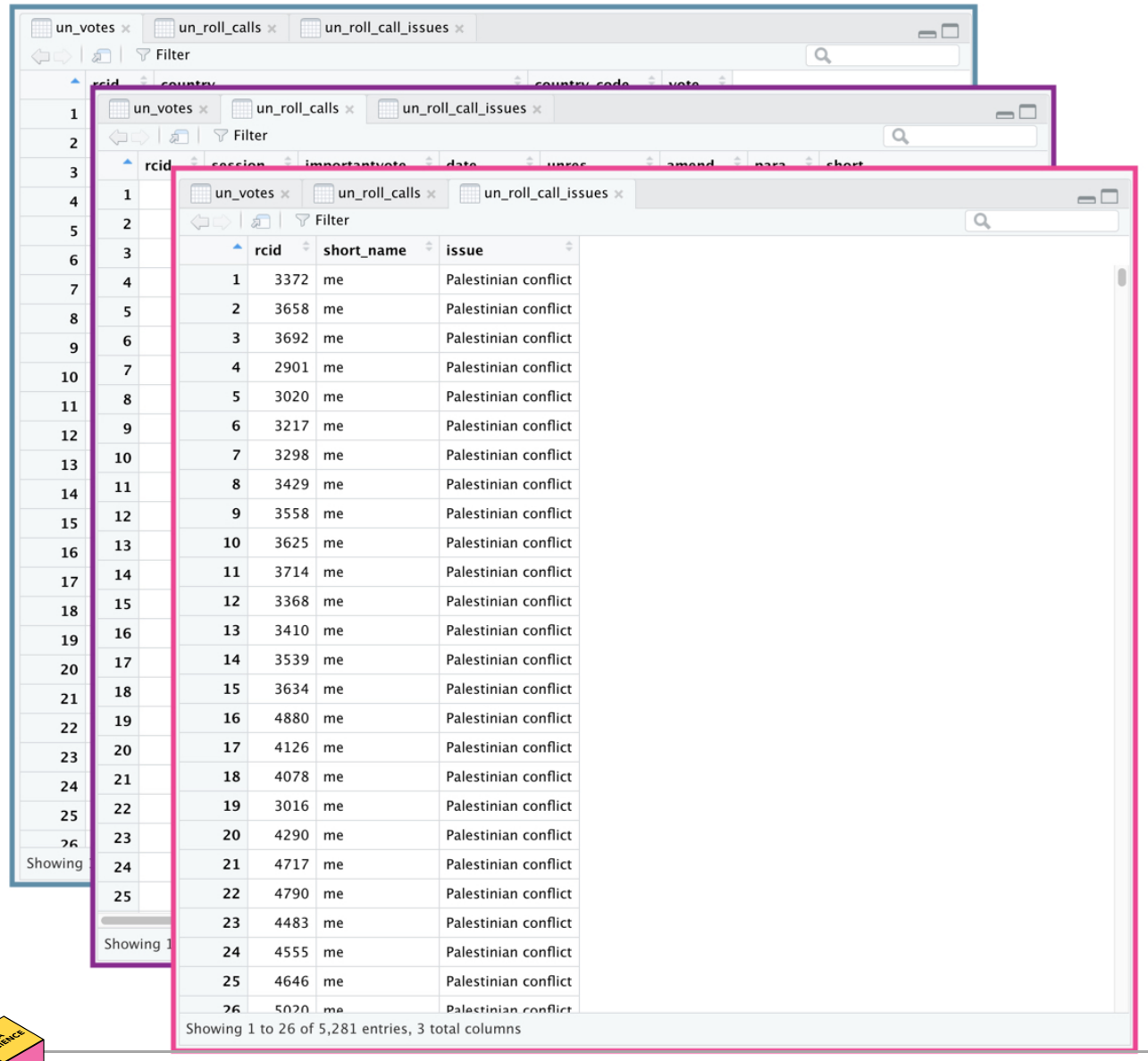
- Where can we find the issues for each visualization?
- Which countries are being visualized?
- What is our response? What is on our y-axis?
- What is our predictors? What is on the x-axis?

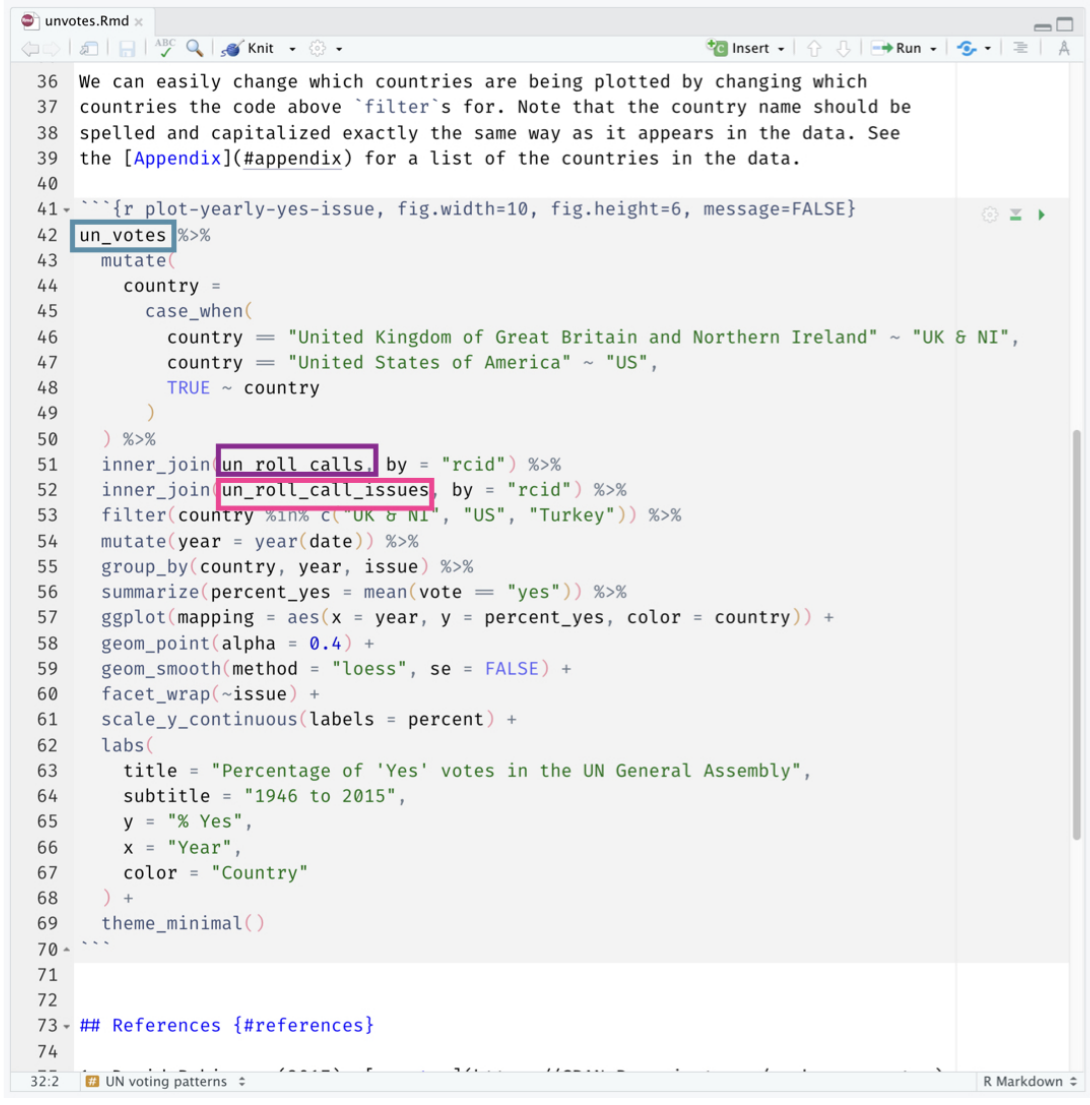

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US

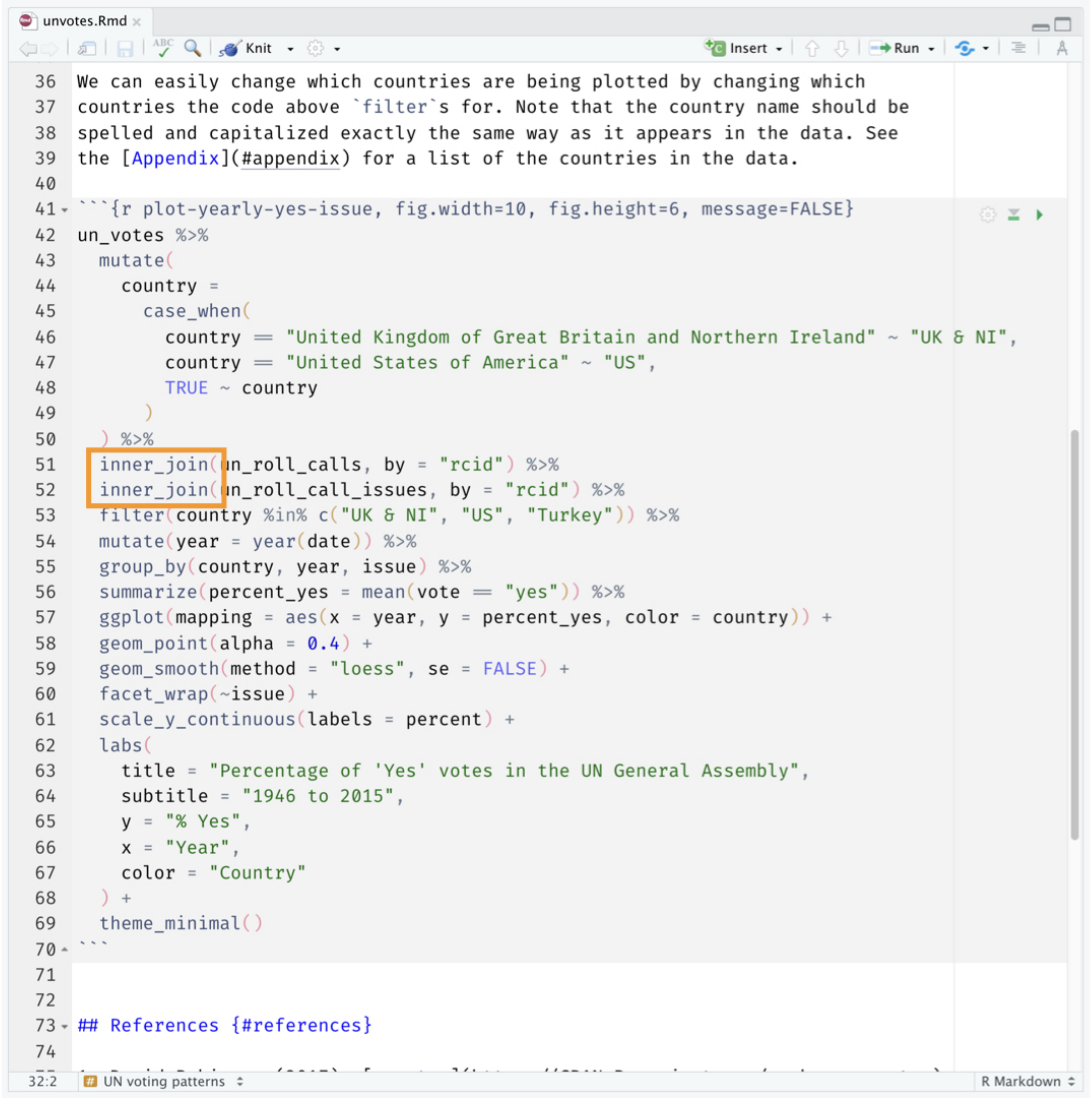

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US

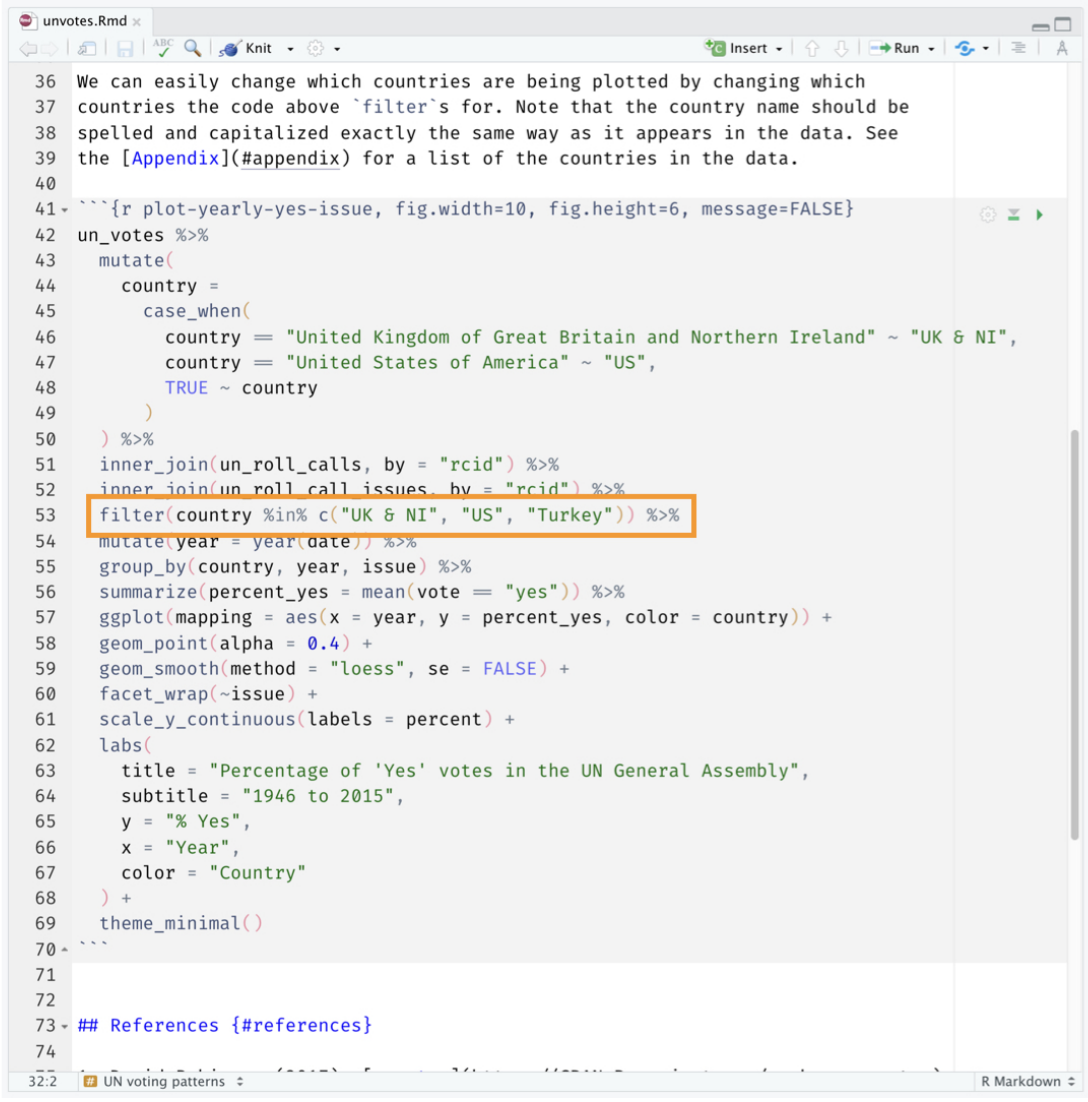

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US

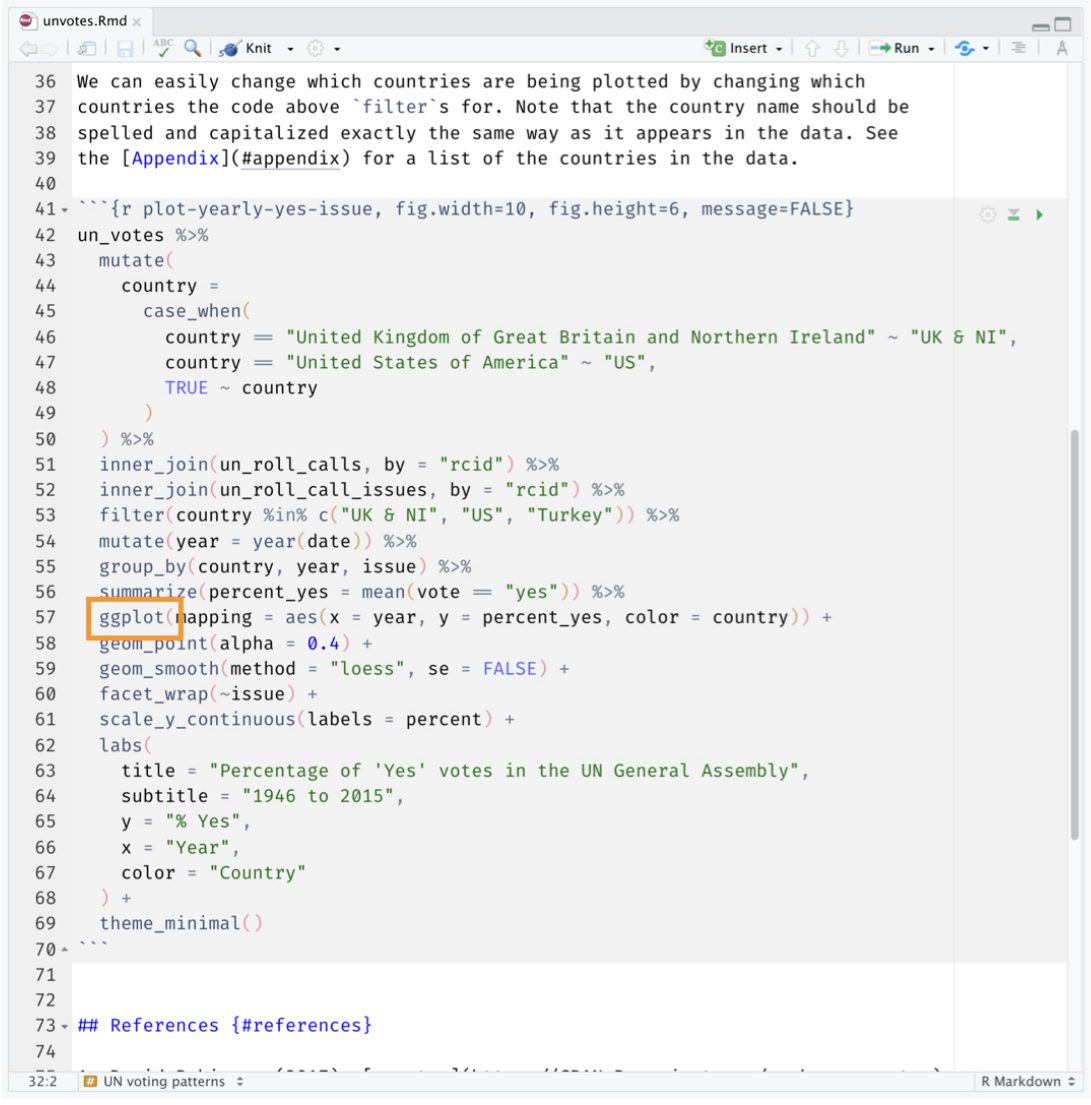

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
Human rights
Nuclear weapons and nuclear material
Palestinian conflict
% Yes
100%
75%
50%
25%
0%
1960
1980
2000
Year
Country
Turkey
UK & NI
US


un_votes
un_roll_calls
un_roll_call_issues
Filter
rcid
country
country_code
vote
session
importantvote
date
unres
amend
para
short
short_name
issue
Palestinian conflict
Showing 1 to 26 of 5,281 entries, 3 total columns


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


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


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


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

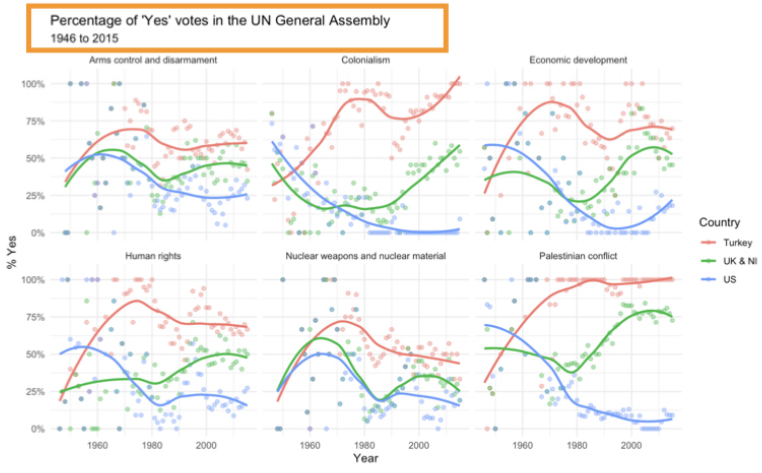
unvotes.Rmd

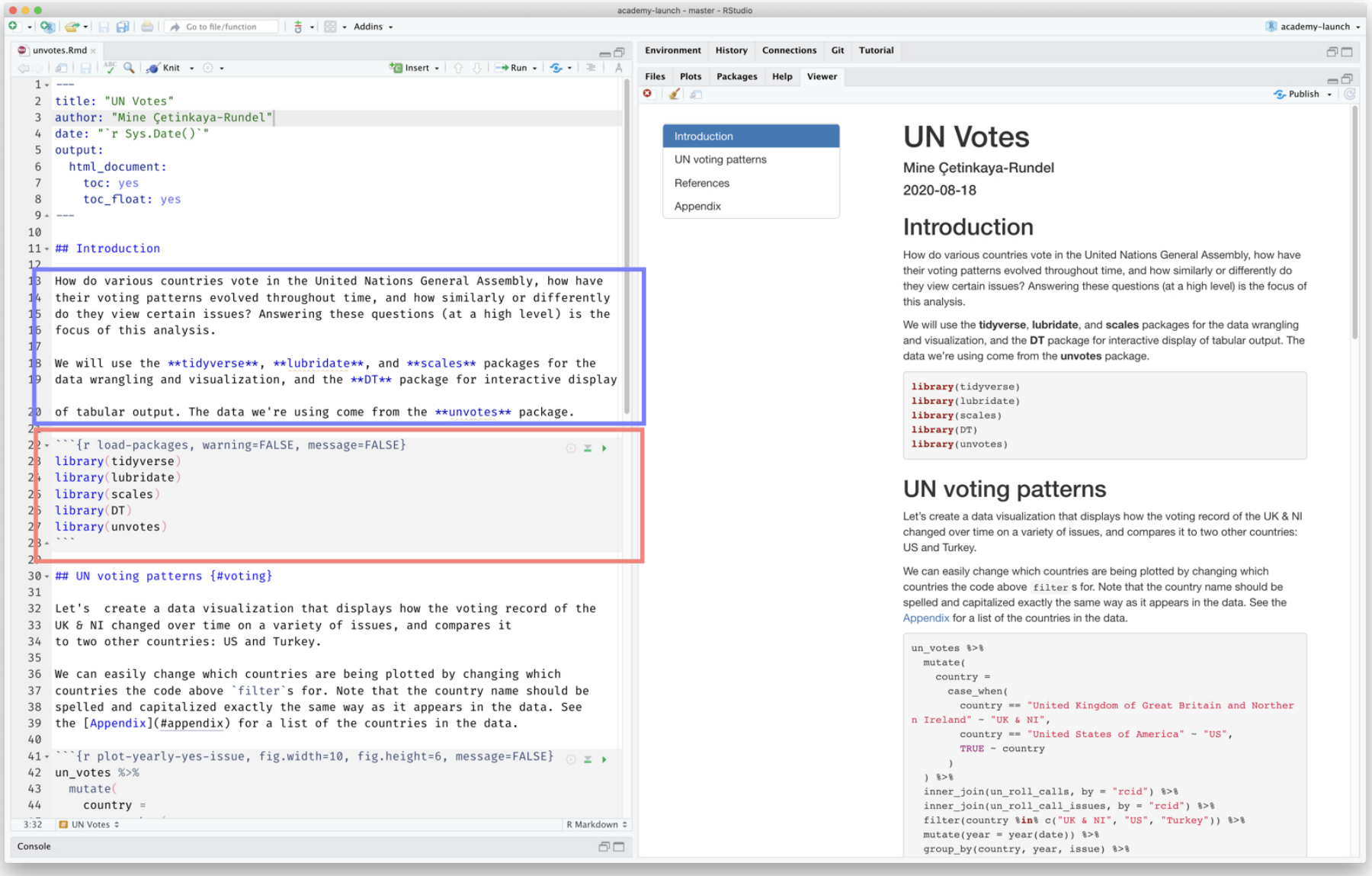
```
We can easily change which countries are being plotted by changing which
countries the code above `filter`s for. Note that the country name should be
spelled and capitalized exactly the same way as it appears in the data. See
the [Appendix](#appendix) for a list of the countries in the data.

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

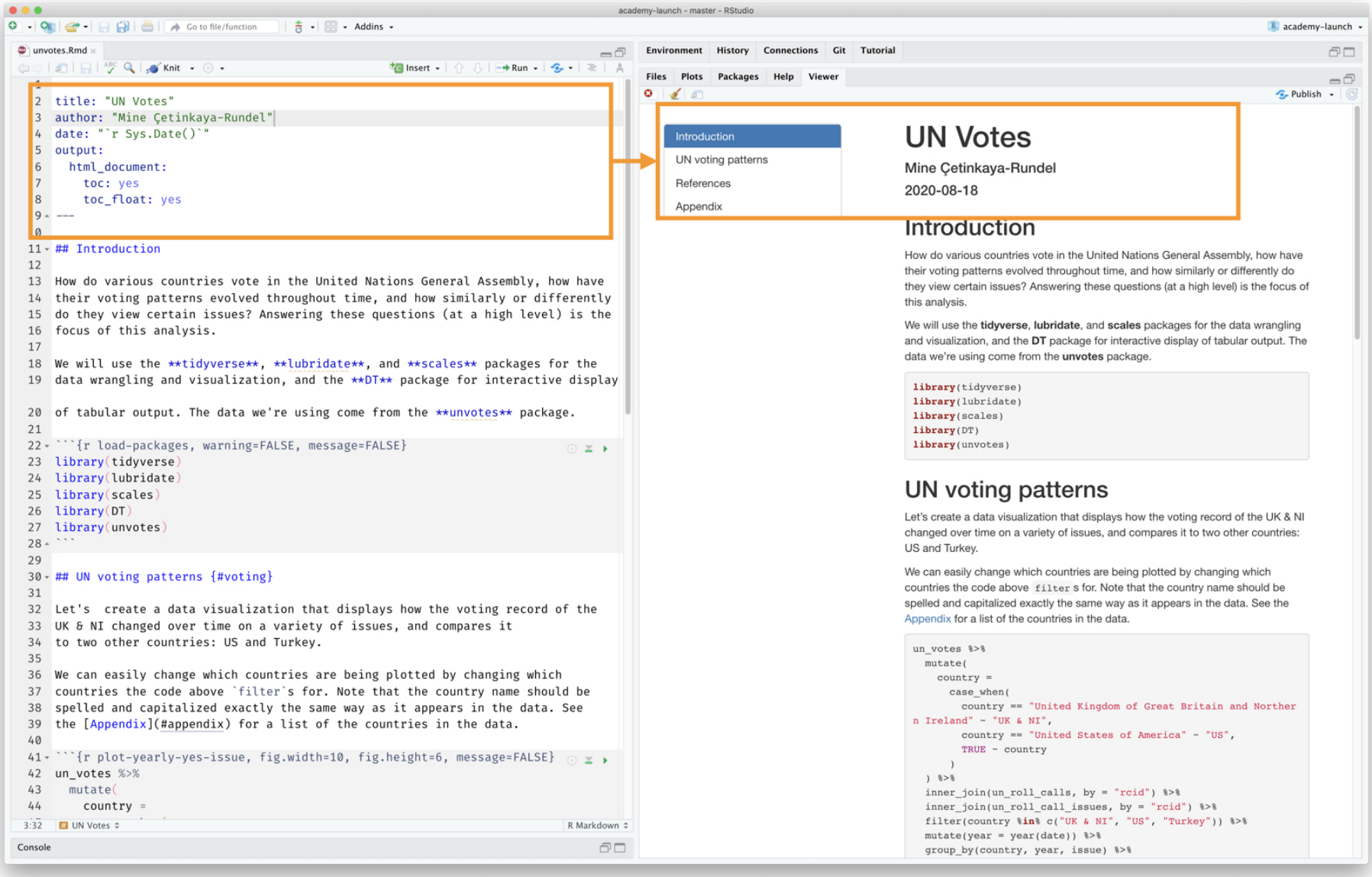
References {#references}
```

32:2 UN voting patterns R Markdown



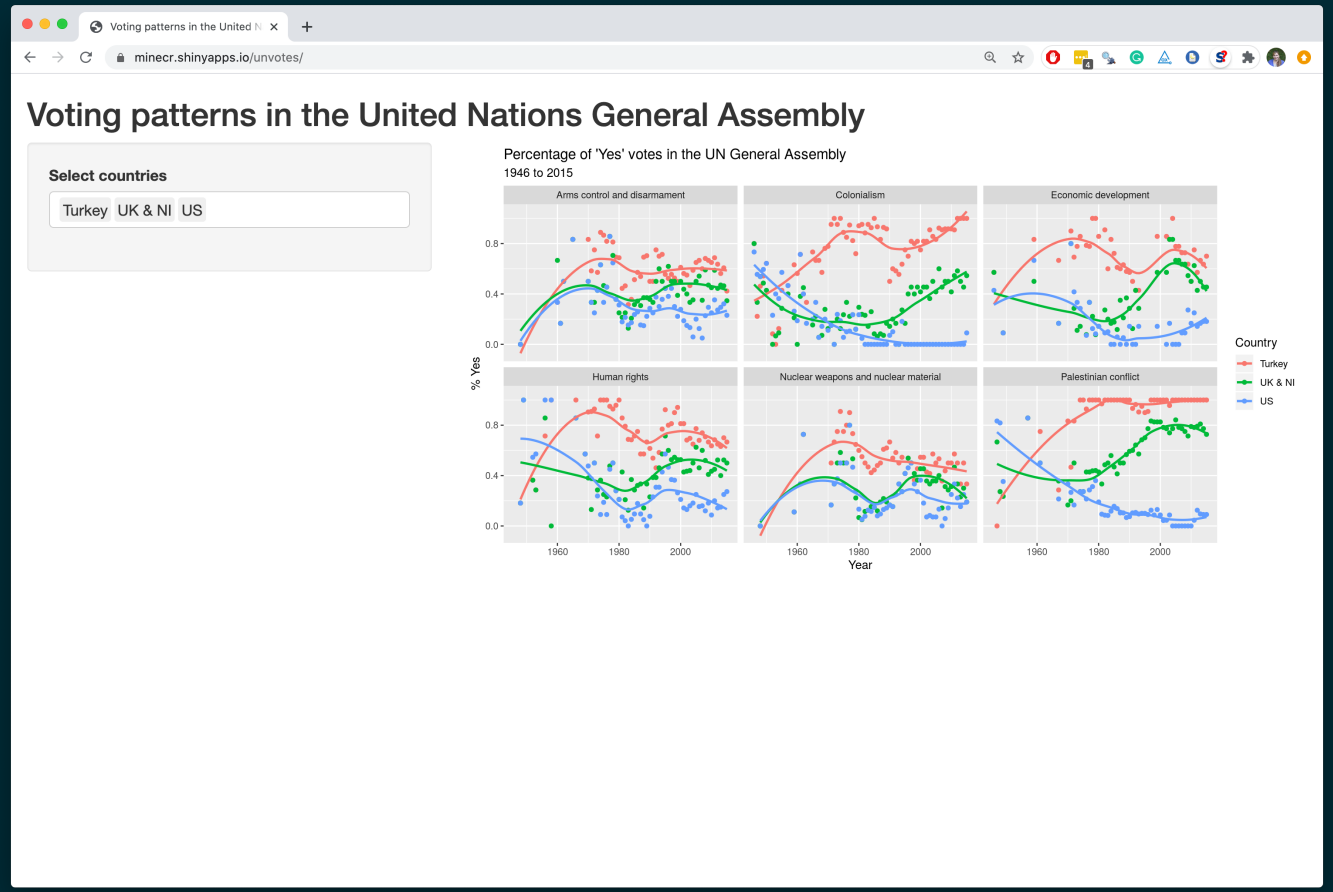

academy-launch - master - RStudio
unvotes.Rmd
Knit
Insert
Run
---
title: "UN Votes"
author: "Mine Çetinkaya-Rundel"
date: "`r Sys.Date()`"
output:
html_document:
toc: yes
toc_float: yes

Introduction
How do various countries vote in the United Nations General Assembly, how have
their voting patterns evolved throughout time, and how similarly or differently
do they view certain issues? Answering these questions (at a high level) is the
focus of this analysis.
We will use the **tidyverse**, **lubridate**, and **scales** packages for the
data wrangling and visualization, and the **DT** package for interactive display
of tabular output. The data we're using come from the **unvotes** package.
```{r load-packages, warning=FALSE, message=FALSE}
library(tidyverse)
library(lubridate)
library(scales)
library(DT)
library(unvotes)
```
UN voting patterns {#voting}
Let's create a data visualization that displays how the voting record of the
UK & NI changed over time on a variety of issues, and compares it
to two other countries: US and Turkey.
We can easily change which countries are being plotted by changing which
countries the code above `filter`s for. Note that the country name should be
spelled and capitalized exactly the same way as it appears in the data. See
the [Appendix](#appendix) for a list of the countries in the data.
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
mutate(
country =
3:32
UN Votes
R Markdown
Console
Environment
History
Connections
Git
Tutorial
Files
Plots
Packages
Help
Viewer
Publish
Introduction
UN voting patterns
References
Appendix
UN Votes
Mine Çetinkaya-Rundel
2020-08-18
Introduction
How do various countries vote in the United Nations General Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do they view certain issues? Answering these questions (at a high level) is the focus of this analysis.
We will use the tidyverse, lubridate, and scales packages for the data wrangling and visualization, and the DT package for interactive display of tabular output. The data we're using come from the unvotes package.
library(tidyverse)
library(lubridate)
library(scales)
library(DT)
library(unvotes)
UN voting patterns
Let's create a data visualization that displays how the voting record of the UK & NI changed over time on a variety of issues, and compares it to two other countries: US and Turkey.
We can easily change which countries are being plotted by changing which countries the code above 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 Norther
n 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
Knit
Insert
Run
title: "UN Votes"
author: "Mine Çetinkaya-Rundel"
date: "`r Sys.Date()`"
output:
html_document:
toc: yes
toc_float: yes

Introduction
How do various countries vote in the United Nations General Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do they view certain issues? Answering these questions (at a high level) is the focus of this analysis.
We will use the **tidyverse**, **lubridate**, and **scales** packages for the data wrangling and visualization, and the **DT** package for interactive display of tabular output. The data we're using come from the **unvotes** package.
```{r load-packages, warning=FALSE, message=FALSE}
library(tidyverse)
library(lubridate)
library(scales)
library(DT)
library(unvotes)
```
UN voting patterns {#voting}
Let's create a data visualization that displays how the voting record of the UK & NI changed over time on a variety of issues, and compares it to two other countries: US and Turkey.
We can easily change which countries are being plotted by changing which countries the code above `filter`s for. Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. See the [Appendix](#appendix) for a list of the countries in the data.
```{r plot-yearly-yes-issue, fig.width=10, fig.height=6, message=FALSE}
un_votes %>%
mutate(
country =
3:32
UN Votes
R Markdown
Console
Environment
History
Connections
Git
Tutorial
Files
Plots
Packages
Help
Viewer
Publish
Introduction
UN voting patterns
References
Appendix
UN Votes
Mine Çetinkaya-Rundel
2020-08-18
Introduction
How do various countries vote in the United Nations General Assembly, how have their voting patterns evolved throughout time, and how similarly or differently do they view certain issues? Answering these questions (at a high level) is the focus of this analysis.
We will use the tidyverse, lubridate, and scales packages for the data wrangling and visualization, and the DT package for interactive display of tabular output. The data we're using come from the unvotes package.
library(tidyverse)
library(lubridate)
library(scales)
library(DT)
library(unvotes)
UN voting patterns
Let's create a data visualization that displays how the voting record of the UK & NI changed over time on a variety of issues, and compares it to two other countries: US and Turkey.
We can easily change which countries are being plotted by changing which countries the code above 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 $\rightarrow$ R
- Literate programming (code, narrative, output in one place) $\rightarrow$ R Markdown
- Version control $\rightarrow$ 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

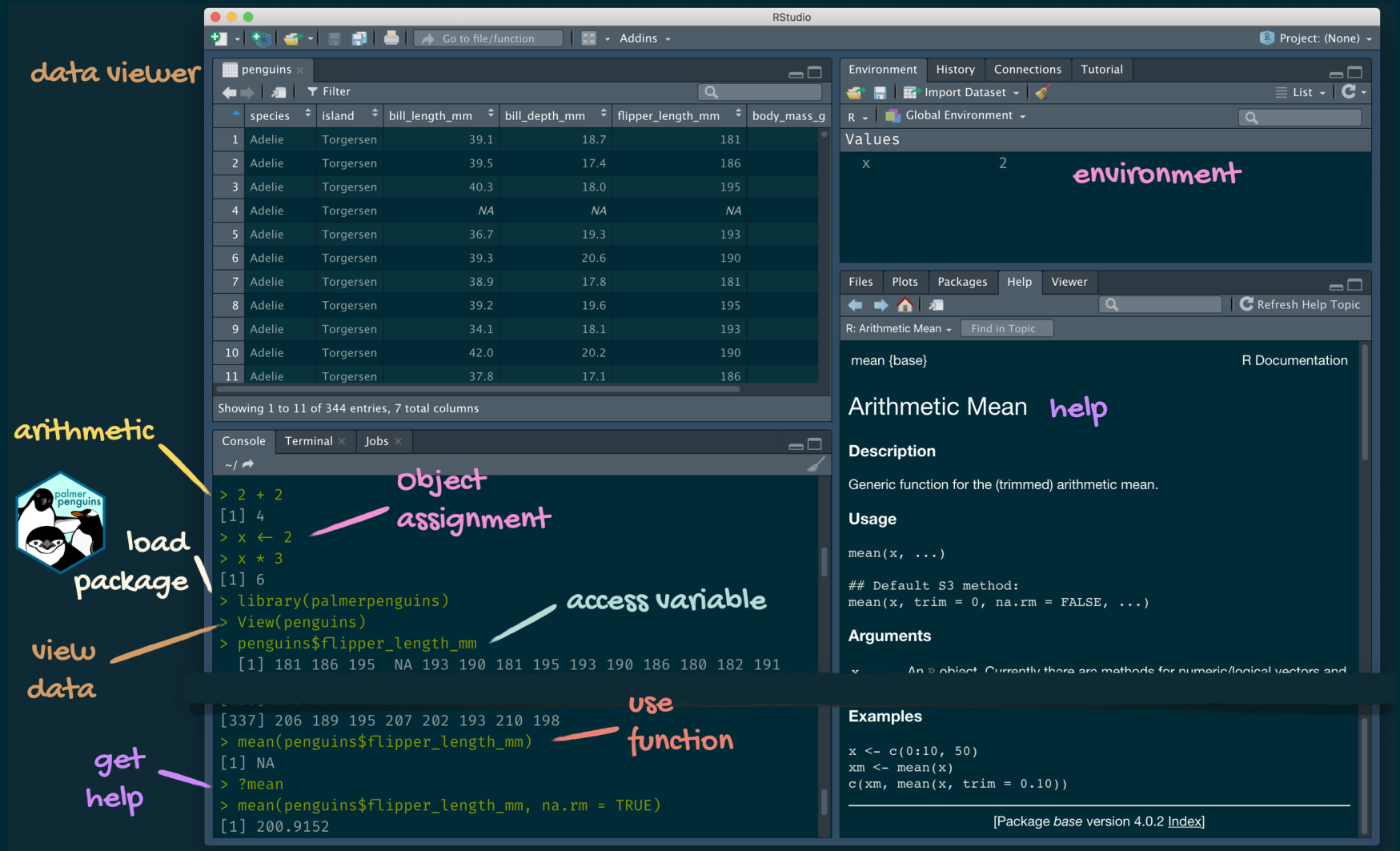
R packages

- **Packages** are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data[1]
- 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



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

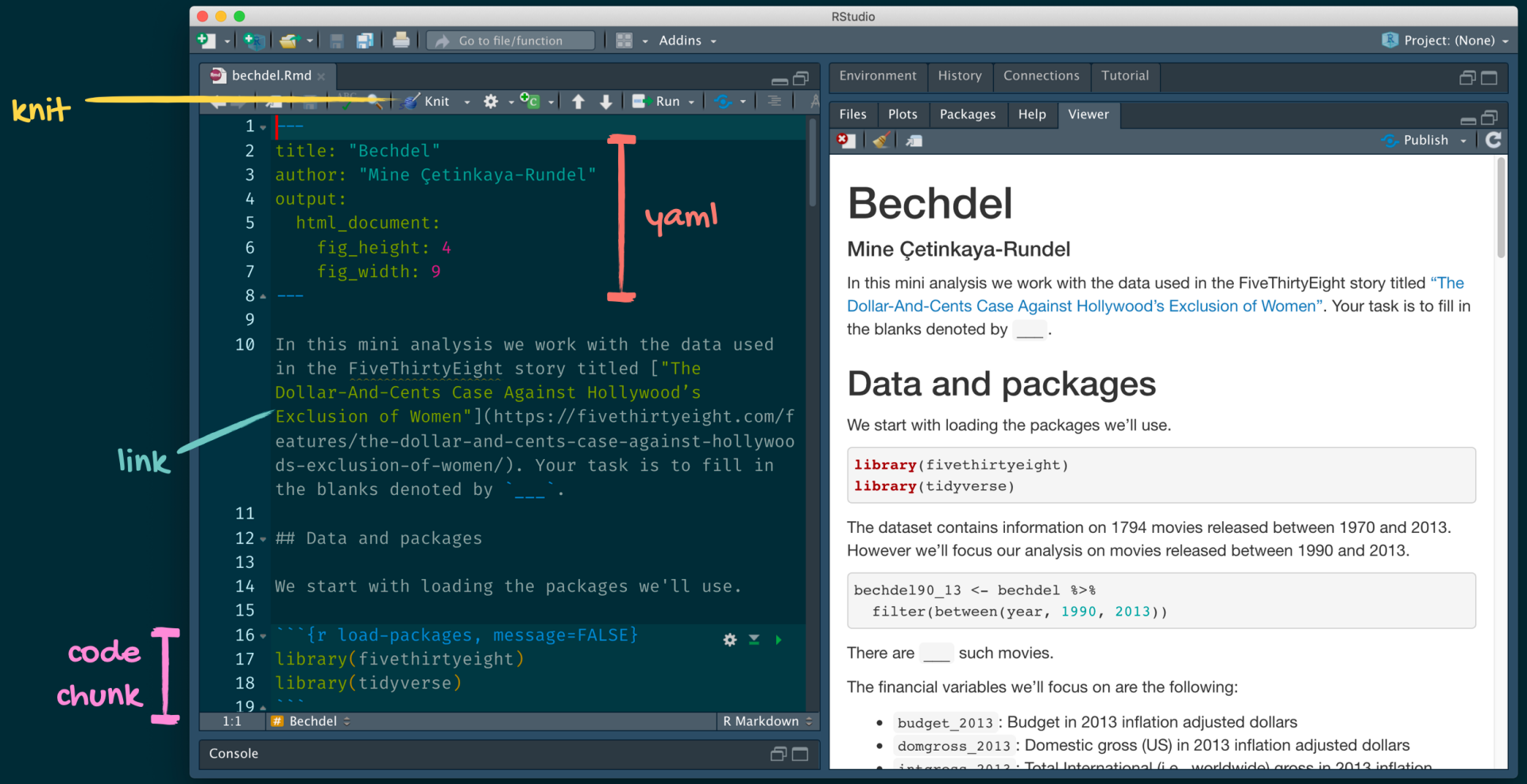
- **rmarkdown** and the various packages that support it enable R users to write their code and prose in reproducible computational documents
- We will generally refer to R Markdown documents (with `.Rmd` extension), e.g. *"Do this in your R Markdown document"* and rarely discuss loading the rmarkdown 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



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
```

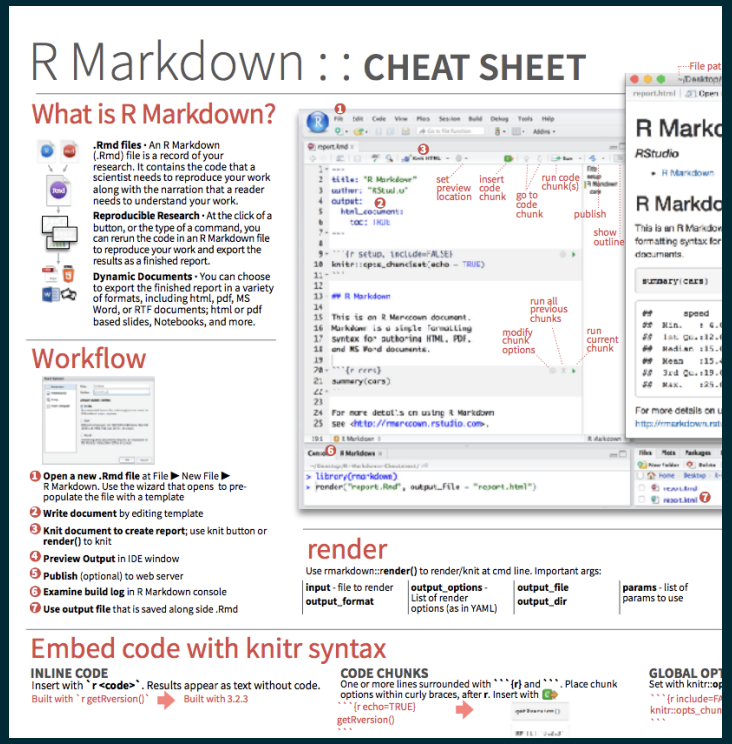
> What happens? Why the error?

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?

.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 your work.

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 html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

Workflow

1. **Open a new .Rmd file** at File ▶ New File ▶ R Markdown. Use the wizard that opens to pre-populate the file with a template
2. **Write document** by editing template
3. **Knit document to create report**; use knit button or **render()** to knit
4. **Preview Output** in IDE window
5. **Publish** (optional) to web server
6. **Examine build log** in R Markdown console
7. **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_format**	**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

```{r echo=TRUE}
getRversion()
```

GLOBAL OPT

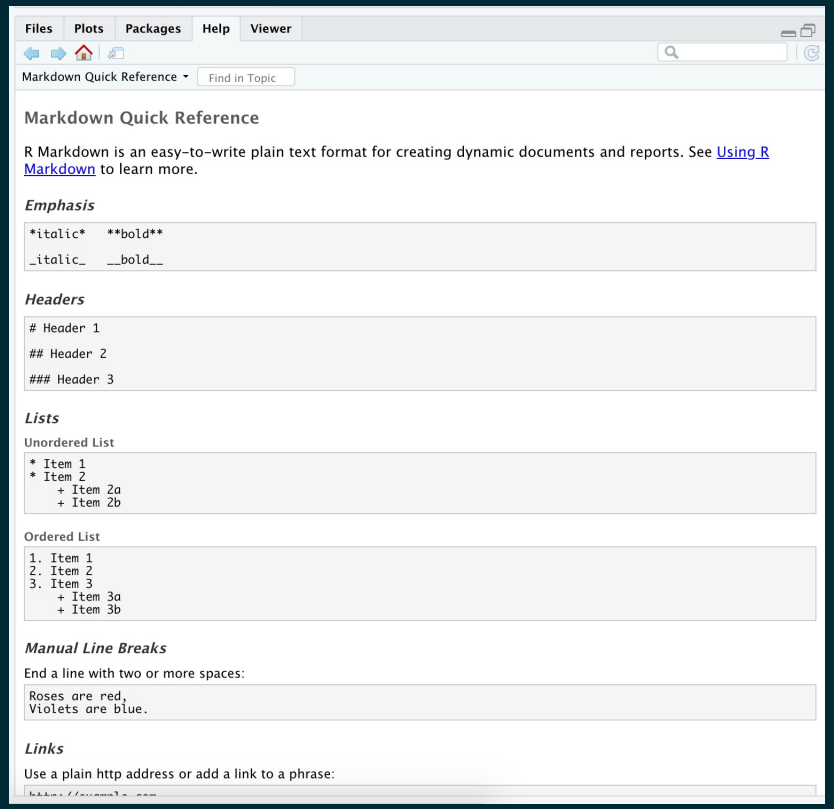
Set with knitr::op

```{r include=FA
knitr::opts_chun
```

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:

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

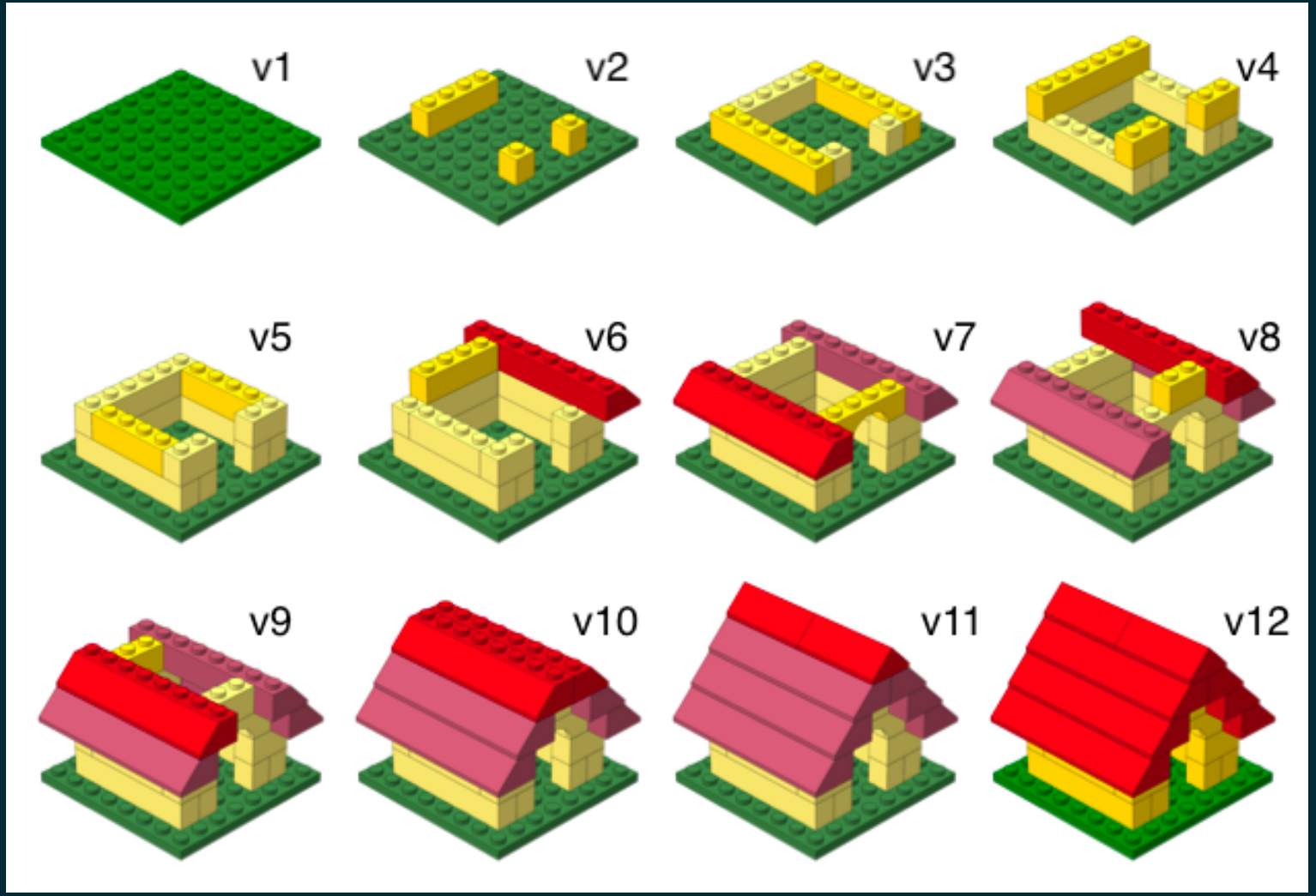
Git and GitHub

Git and GitHub

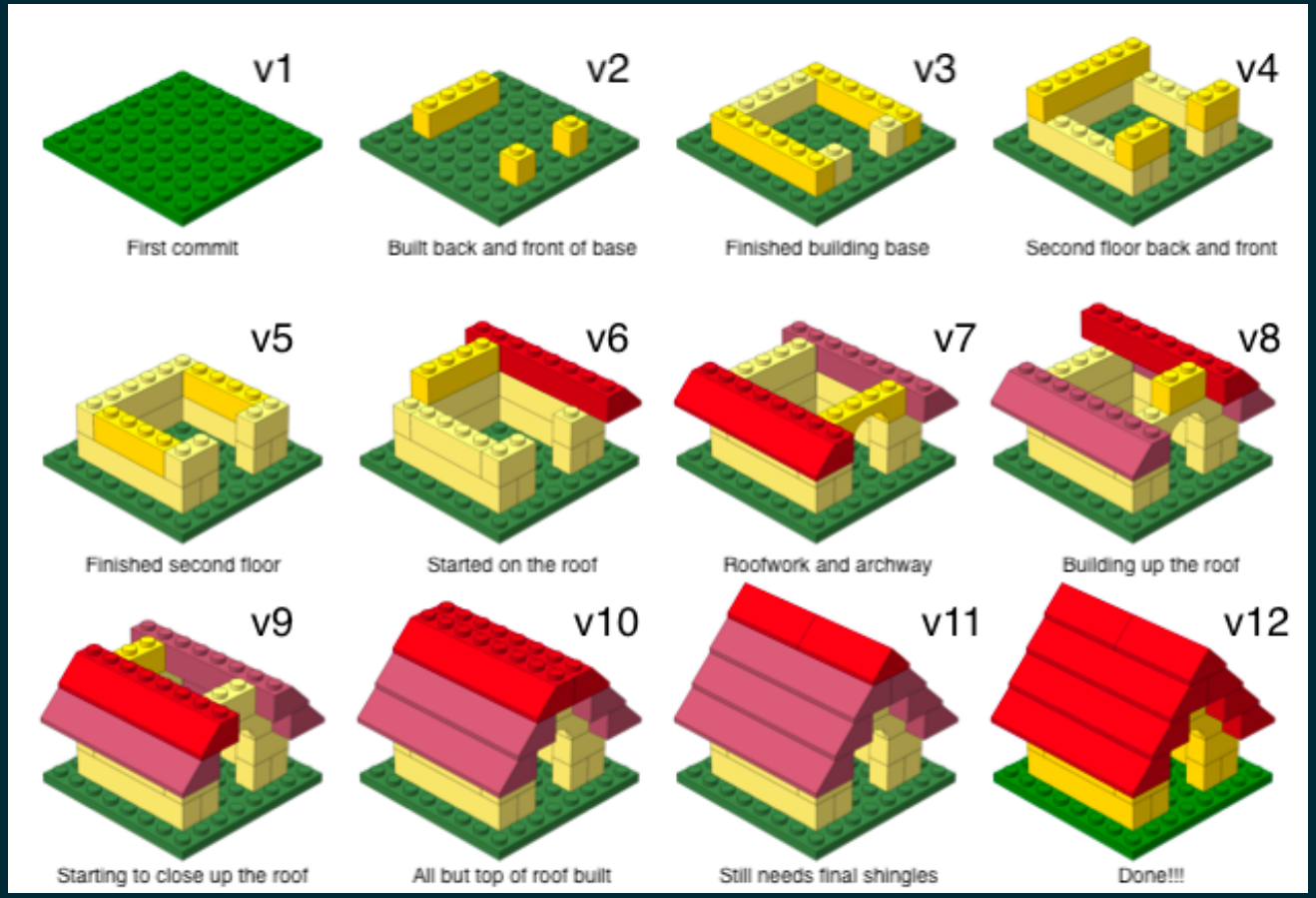
- Git is a version control system -- like “Track Changes” features from Microsoft Word, on steroids
- It's not the only version control system, but it's a very popular one

- 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!)

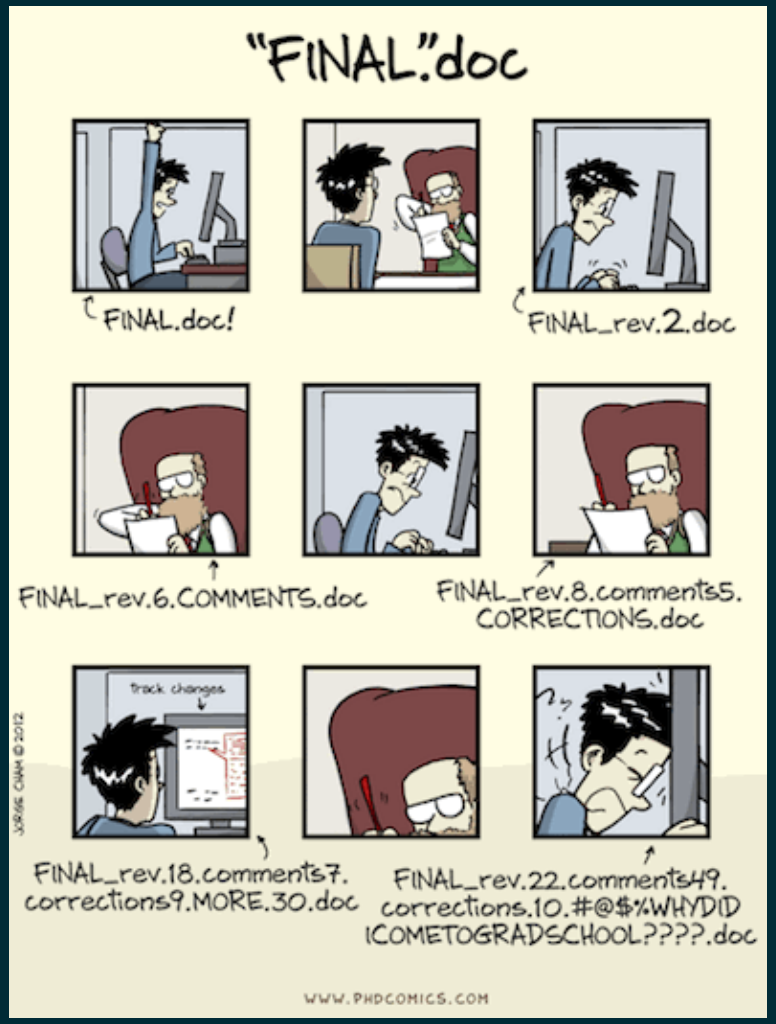
Versioning



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.

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. 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!