

UNvotes_170702

Hanh Nguyen

7/2/2017

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##   filter, lag
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(unvotes)
```

```
## If you use data from the unvotes package, please cite the following:
##
## Erik Voeten "Data and Analyses of Voting in the UN General Assembly" Routledge Handbook of International
```

```
library(ggplot2)
library(tidyr)
library(purrr)
```

```
##
## Attaching package: 'purrr'
##
## The following objects are masked from 'package:dplyr':
##
##   contains, order_by
```

```
library(broom)
```

Erik Voeten “Data and Analyses of Voting in the UN General Assembly” Routledge Handbook of International Organization, edited by Bob Reinalda (published May 27, 2013).

Below is three datasets in the package and their following columns:

1. un_votes provides information on the voting history of the United Nations General Assembly. Contains one row for each country-vote pair.

- rcid: The roll call id; it is the primary key used to join with tables un_roll_calls and un_roll_call_issues
- vote: Vote result as a factor of yes/abstain/no (The original data included cases where a country was absent or was not yet a member. In this dataset these were filtered out to include only votes of Yes, Abstain, and No)
- country: Country name, by official English short name (ISO)

```
head(un_votes)
```

```
## # A tibble: 6 × 3
##   rcid    country  vote
##   <dbl>    <chr>  <fctr>
## 1     3      Egypt abstain
## 2     3   Honduras   yes
## 3     3  Costa Rica   yes
## 4     3 El Salvador   yes
```

```
## 5      3      France      no
## 6      3      Uruguay     yes
```

```
unique(un_votes$vote)
```

```
## [1] abstain yes      no
## Levels: abstain no yes
```

2. `un_roll_calls` provides information on each roll call vote of the United Nations General Assembly.

- `rcid`: The roll call id
- `session`: Session number. The United Nations holds one session per year; these started in 1946
- `importantvote`: Whether the vote was classified as important by the U.S. State Department report “Voting Practices in the United Nations”. These classifications began with session 39
- `date`: Date of the vote, as a Date vector
- `unres`: Resolution code
- `amend`: Whether the vote was on an amendment; coded only until 1985
- `para`: Whether the vote was only on a paragraph and not a resolution; coded only until 1985
- `short`: Short description
- `descr`: Longer description

```
head(un_roll_calls)
```

```
## # A tibble: 6 × 9
##   rcid session importantvote   date   unres amend para
##   <dbl>   <dbl>         <dbl> <date>   <chr> <dbl> <dbl>
## 1     3     1             0 1946-01-01 R/1/66     1     0
## 2     4     1             0 1946-01-02 R/1/79     0     0
## 3     5     1             0 1946-01-04 R/1/98     0     0
## 4     6     1             0 1946-01-04 R/1/107    0     0
## 5     7     1             0 1946-01-02 R/1/295    1     0
## 6     8     1             0 1946-01-05 R/1/297    1     0
## # ... with 2 more variables: short <chr>, descr <chr>
```

3. `un_roll_call_issues` provides issue (topic) classifications of roll call votes of the United Nations General Assembly, with one row for each pair of a roll call vote and an issue describing that vote. Many votes had no topic, and some have more than one.

- `rcid`: The roll call id; used to join with `un_votes` and `un_roll_calls`
- `short_name`: Two-letter issue codes
- `issue`: Descriptive issue name

```
head(un_roll_call_issues)
```

```
## # A tibble: 6 × 3
##   rcid short_name   issue
##   <dbl>   <chr>       <chr>
## 1    30      me Palestinian conflict
## 2    34      me Palestinian conflict
## 3    77      me Palestinian conflict
## 4   9002      me Palestinian conflict
## 5   9003      me Palestinian conflict
## 6   9004      me Palestinian conflict
```

Further details about the package and datasets can be found here <https://github.com/dgrtwo/unvotes> or by `??unvotes`

We want to know the voting pattern by each year and each country. Therefore, we will merge `un_votes` and `un_roll_calls` by `rcid`. Furthermore, we will create another field named `year` derived from the column `date`

```
df = merge(x=un_votes, y=un_roll_calls, by="rcid", all.x=TRUE)
df$year <- as.numeric(format(df$date,"%Y"))
head(df)
```

```
##   rcid      country  vote session importantvote      date unres amend
## 1    3      Egypt abstain      1              0 1946-01-01 R/1/66     1
## 2    3    Honduras   yes      1              0 1946-01-01 R/1/66     1
## 3    3  Costa Rica   yes      1              0 1946-01-01 R/1/66     1
## 4    3 El Salvador   yes      1              0 1946-01-01 R/1/66     1
## 5    3      France   no       1              0 1946-01-01 R/1/66     1
## 6    3    Uruguay   yes      1              0 1946-01-01 R/1/66     1
##   para                                short
## 1    0 AMENDMENTS, RULES OF PROCEDURE
## 2    0 AMENDMENTS, RULES OF PROCEDURE
## 3    0 AMENDMENTS, RULES OF PROCEDURE
## 4    0 AMENDMENTS, RULES OF PROCEDURE
## 5    0 AMENDMENTS, RULES OF PROCEDURE
## 6    0 AMENDMENTS, RULES OF PROCEDURE
##
## 1 TO ADOPT A CUBAN AMENDMENT TO THE UK PROPOSAL REFERRING THE PROVISIONAL RULES OF PROCEDURE AND ANY
## 2 TO ADOPT A CUBAN AMENDMENT TO THE UK PROPOSAL REFERRING THE PROVISIONAL RULES OF PROCEDURE AND ANY
## 3 TO ADOPT A CUBAN AMENDMENT TO THE UK PROPOSAL REFERRING THE PROVISIONAL RULES OF PROCEDURE AND ANY
## 4 TO ADOPT A CUBAN AMENDMENT TO THE UK PROPOSAL REFERRING THE PROVISIONAL RULES OF PROCEDURE AND ANY
## 5 TO ADOPT A CUBAN AMENDMENT TO THE UK PROPOSAL REFERRING THE PROVISIONAL RULES OF PROCEDURE AND ANY
## 6 TO ADOPT A CUBAN AMENDMENT TO THE UK PROPOSAL REFERRING THE PROVISIONAL RULES OF PROCEDURE AND ANY
##   year
## 1 1946
## 2 1946
## 3 1946
## 4 1946
## 5 1946
## 6 1946
```

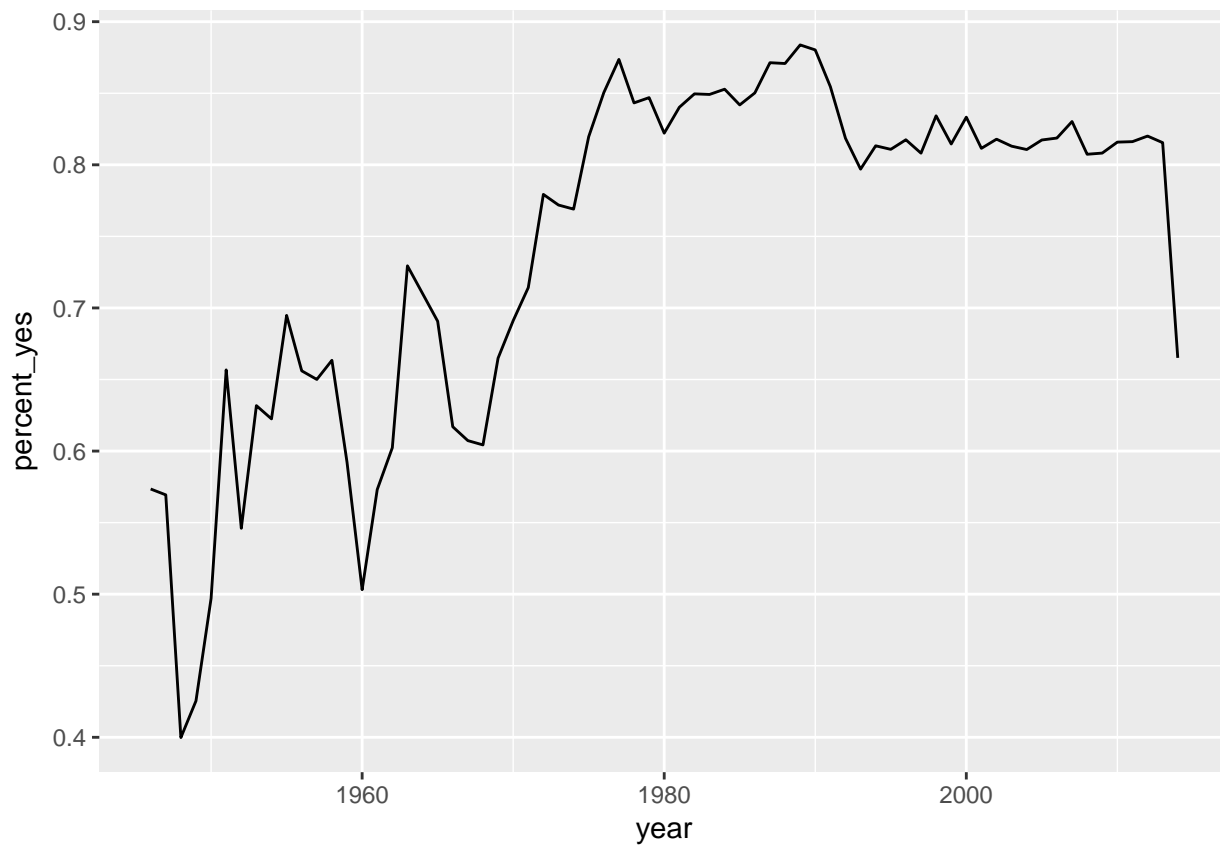
To see the voting pattern by year, we group the df by year using group_by() function

```
by_year = df %>%
  group_by(year) %>%
  summarize(total=n(), percent_yes = mean(vote=="yes"))
head(by_year)
```

```
## # A tibble: 6 × 3
##   year total percent_yes
##   <dbl> <int>      <dbl>
## 1  1946  2143    0.5734951
## 2  1947  2039    0.5693968
## 3  1948  3454    0.3998263
## 4  1949  5700    0.4254386
## 5  1950  2911    0.4970800
## 6  1951   402    0.6567164
```

The data frame by_year is actually a time series and by looking at the visualization, we can see a trend over time

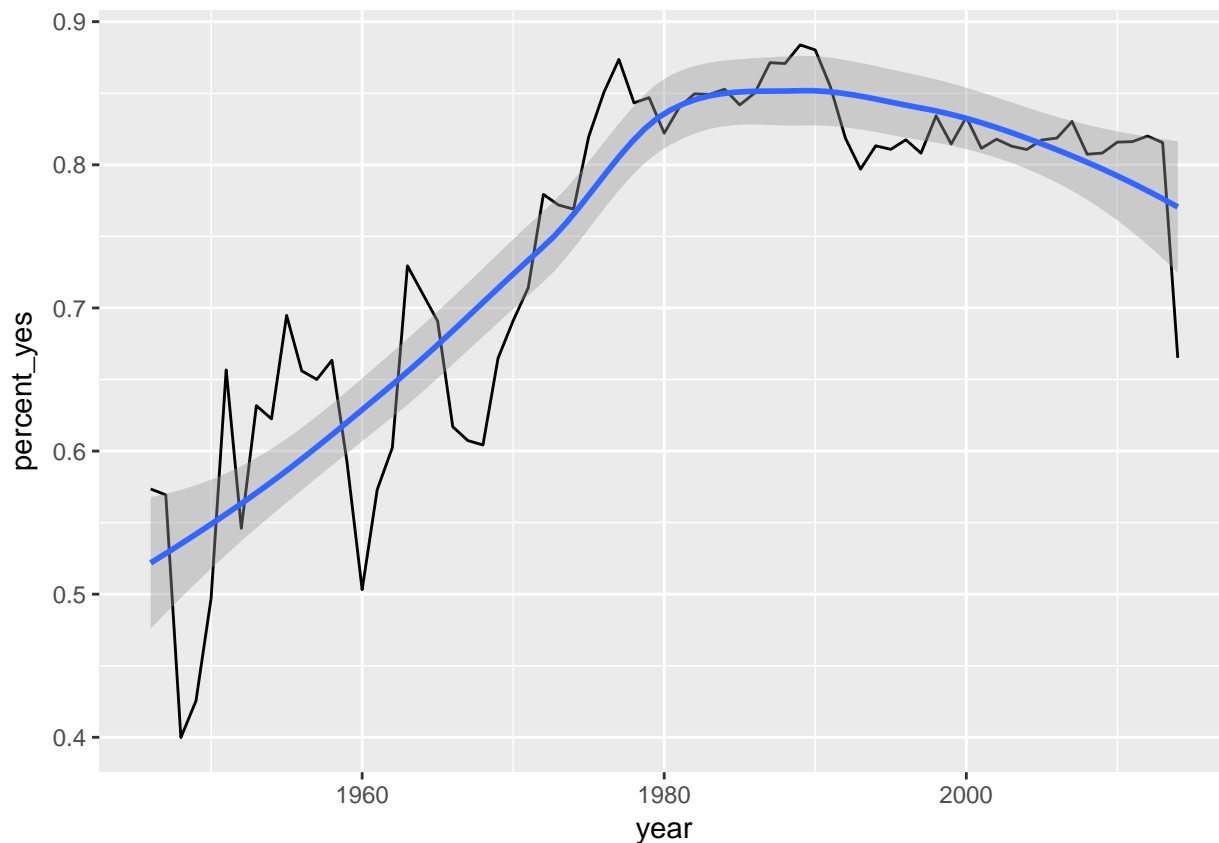
```
ggplot(by_year, aes(year, percent_yes)) +
  geom_line()
```



Adding the `geom_smooth()` function

```
ggplot(by_year, aes(year, percent_yes)) +  
  geom_line() +  
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```



A different perspective is to see voting patterns among countries.

```
by_country = df %>%
  group_by(country) %>%
  summarize(total=n(), percent_yes = mean(vote=="yes"))
head(by_country)
```

```
## # A tibble: 6 × 3
##       country total percent_yes
##       <chr> <int>      <dbl>
## 1 Afghanistan 4824  0.8381012
## 2 Albania      3363  0.7204877
## 3 Algeria      4374  0.8978052
## 4 Andorra      1410  0.6510638
## 5 Angola       2950  0.9223729
## 6 Antigua and Barbuda 2521  0.9170964
```

We sort the data frame by the number of votes and the % of “yes” votes in the ascending order

```
arrange(by_country, total)
```

```
## # A tibble: 200 × 3
##       country total percent_yes
##       <chr> <int>      <dbl>
## 1 Zanzibar     2  0.0000000
## 2 Kiribati     93  0.8172043
## 3 South Sudan  96  0.6979167
## 4 Montenegro  558  0.6433692
## 5 Tuvalu      576  0.8246528
```

```
## 6      Nauru      606  0.6089109
## 7 Timor-Leste  697  0.9670014
## 8      Tonga     775  0.7303226
## 9      Palau     777  0.3063063
## 10 Switzerland 857  0.6569428
## # ... with 190 more rows
```

```
arrange(by_country, percent_yes)
```

```
## # A tibble: 200 × 3
##               country total percent_yes
##               <chr> <int>      <dbl>
## 1             Zanzibar      2  0.0000000
## 2      United States  5237  0.2850869
## 3              Palau    777  0.3063063
## 4             Israel  4790  0.3503132
## 5  Federal Republic of Germany 2151  0.3984193
## 6  Micronesia, Federated States of 1341  0.4131245
## 7      United Kingdom  5218  0.4269835
## 8              France  5171  0.4320248
## 9      Marshall Islands  1468  0.4788828
## 10             Belgium  5238  0.4925544
## # ... with 190 more rows
```

We can recognize that the country that voted least frequently, Zanzibar, had only 2 votes in the entire dataset, thus its `percent_yes` is not meaningful. For this reason, we will exclude countries with fewer than 100 votes in total.

```
by_country %>%
  arrange(percent_yes) %>%
  filter(total >= 100)
```

```
## # A tibble: 197 × 3
##               country total percent_yes
##               <chr> <int>      <dbl>
## 1      United States  5237  0.2850869
## 2              Palau    777  0.3063063
## 3             Israel  4790  0.3503132
## 4  Federal Republic of Germany 2151  0.3984193
## 5  Micronesia, Federated States of 1341  0.4131245
## 6      United Kingdom  5218  0.4269835
## 7              France  5171  0.4320248
## 8      Marshall Islands  1468  0.4788828
## 9              Belgium  5238  0.4925544
## 10      Luxembourg  5169  0.5105436
## # ... with 187 more rows
```

Lastly, we want to summarize by both year and country, constructing a dataset that shows what fraction of the time each country votes “yes” in each year.

```
by_year_country = df %>%
  group_by(year, country) %>%
  summarize(total = n(),
            percent_yes = mean(vote == "yes"))
head(by_year_country)
```

```
## Source: local data frame [6 x 4]
```

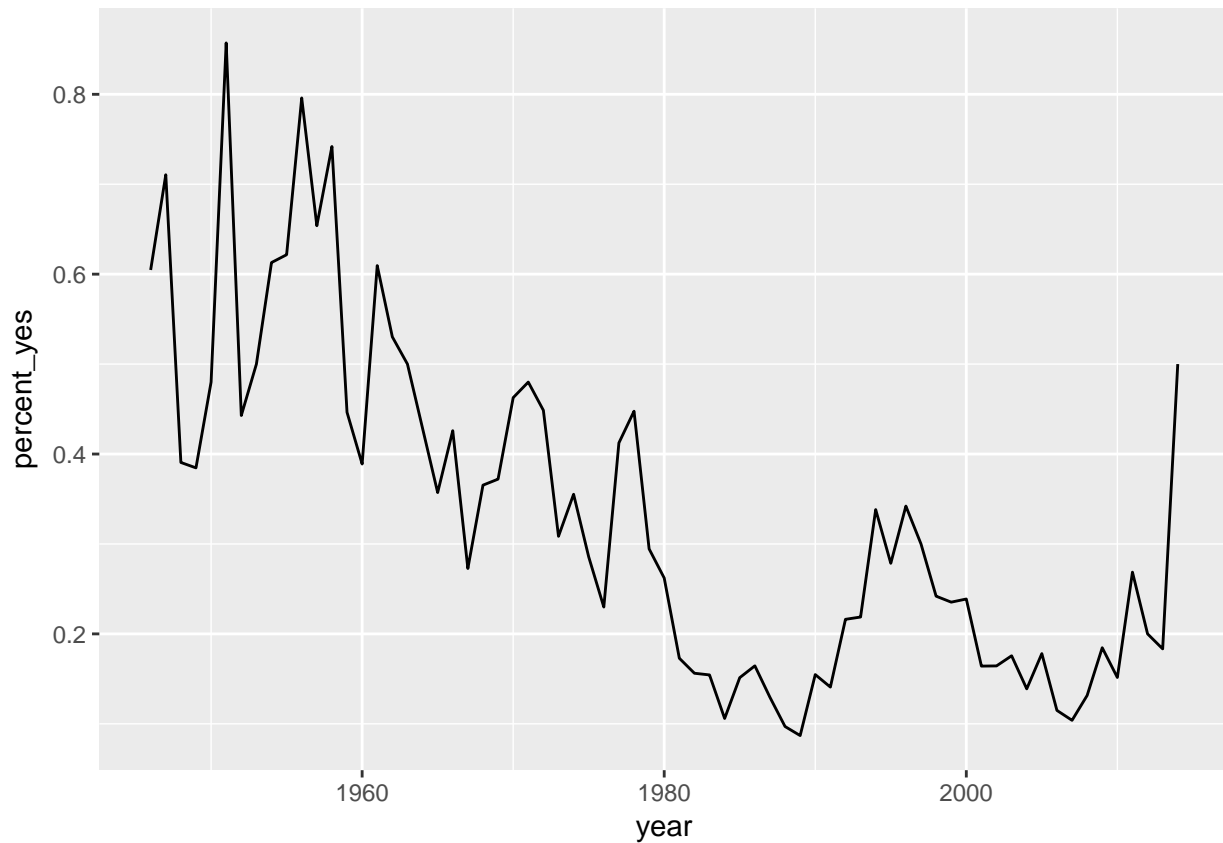
```
## Groups: year [1]
##
##   year                country total percent_yes
##   <dbl>                <chr> <int>         <dbl>
## 1  1946                Afghanistan    17    0.4117647
## 2  1946                Argentina     43    0.6976744
## 3  1946                Australia     43    0.5581395
## 4  1946                Belarus       43    0.4418605
## 5  1946                Belgium       43    0.6046512
## 6  1946 Bolivia, Plurinational State of 43    0.6976744
```

Looking at the US data

```
US_by_year = by_year_country %>%
  filter(country=="United States")
head(US_by_year)
```

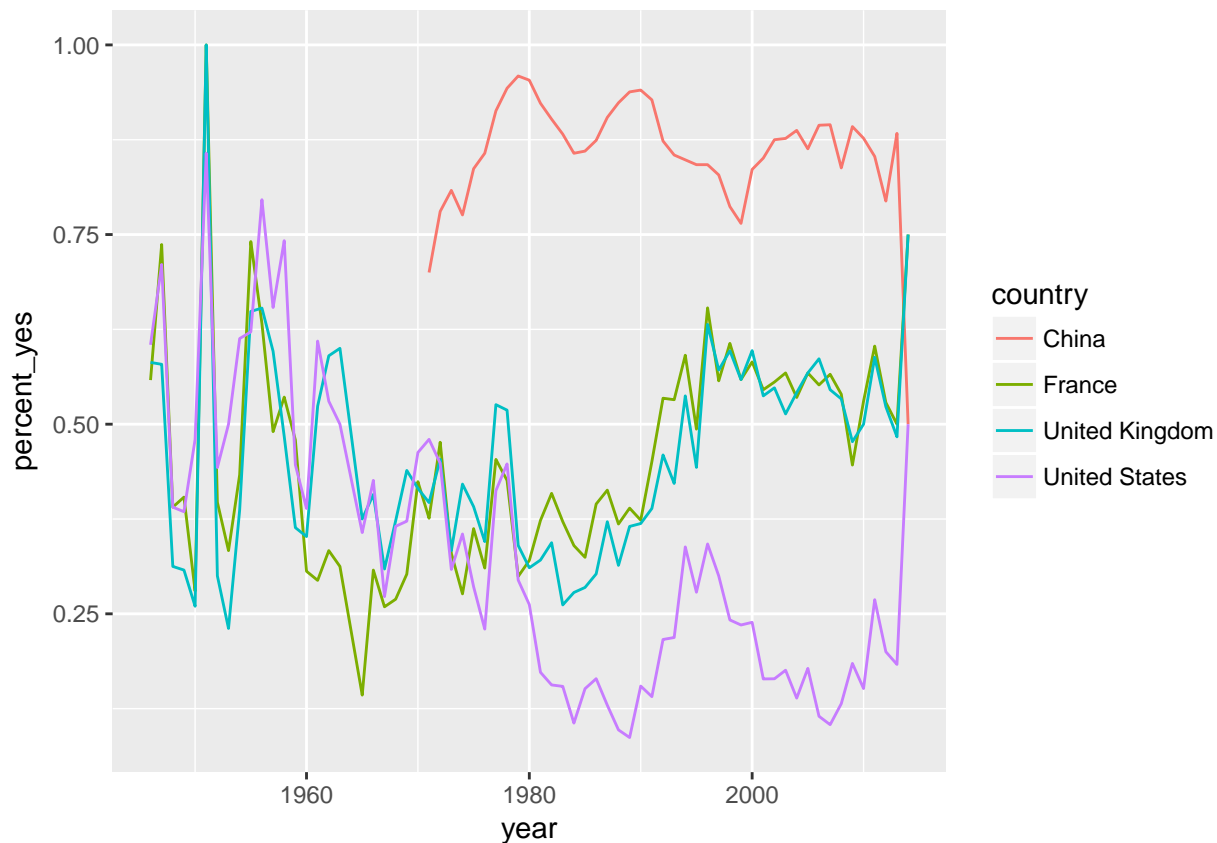
```
## Source: local data frame [6 x 4]
## Groups: year [6]
##
##   year      country total percent_yes
##   <dbl>      <chr> <int>         <dbl>
## 1  1946 United States    43    0.6046512
## 2  1947 United States    38    0.7105263
## 3  1948 United States    64    0.3906250
## 4  1949 United States   104    0.3846154
## 5  1950 United States    50    0.4800000
## 6  1951 United States     7    0.8571429
```

```
ggplot(US_by_year, aes(x=year,y=percent_yes)) +
  geom_line()
```



Plotting just one country at a time is interesting, but it'd be more insightful to compare trends between countries. For example, suppose we want to compare voting trends for the United States, the UK, France, and China.

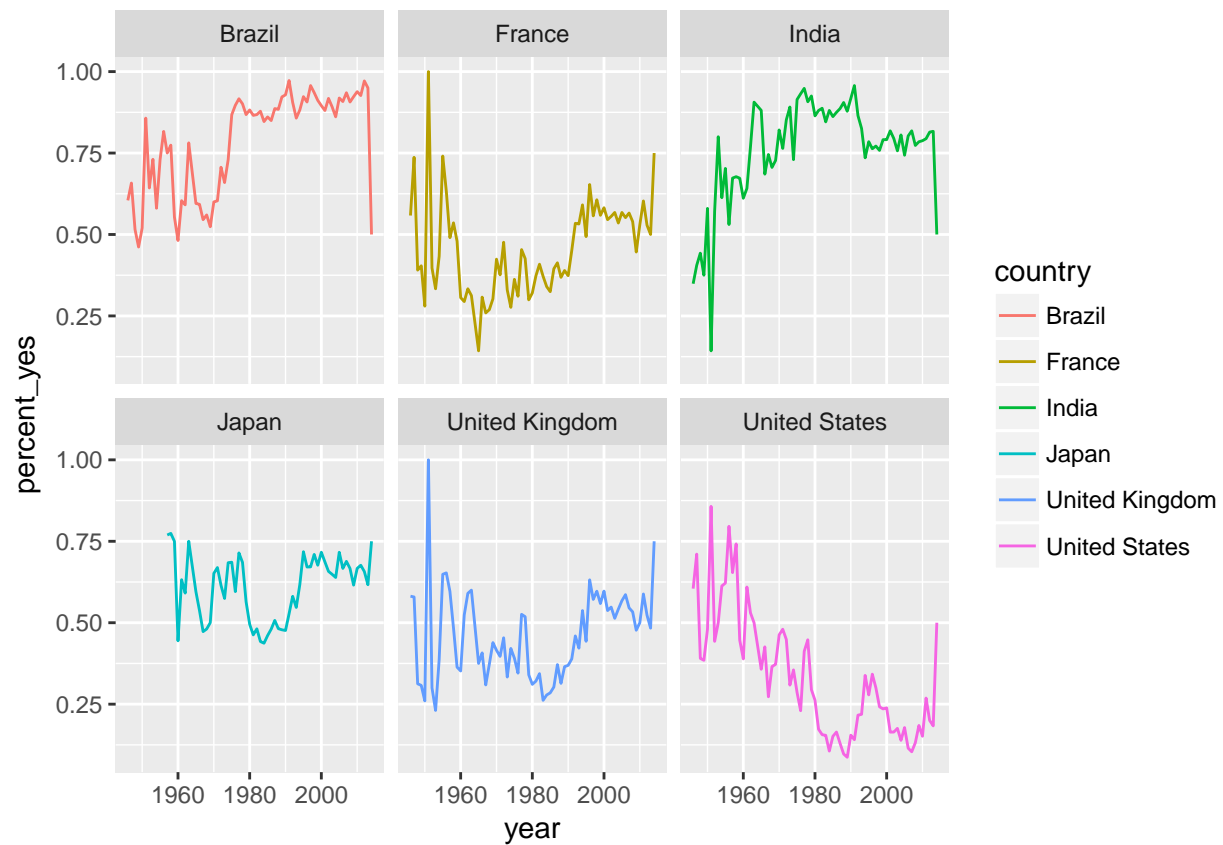
```
countries_4 <- c("United States", "United Kingdom",  
                "France", "China")  
countries_4_by_year = by_year_country %>%  
  filter(country %in% countries_4)  
  
ggplot(countries_4_by_year, aes(x=year, y=percent_yes, color=country)) +geom_line()
```

However, if we want to include more countries, this type of graph could be tough to read. The alternative way is faceting.

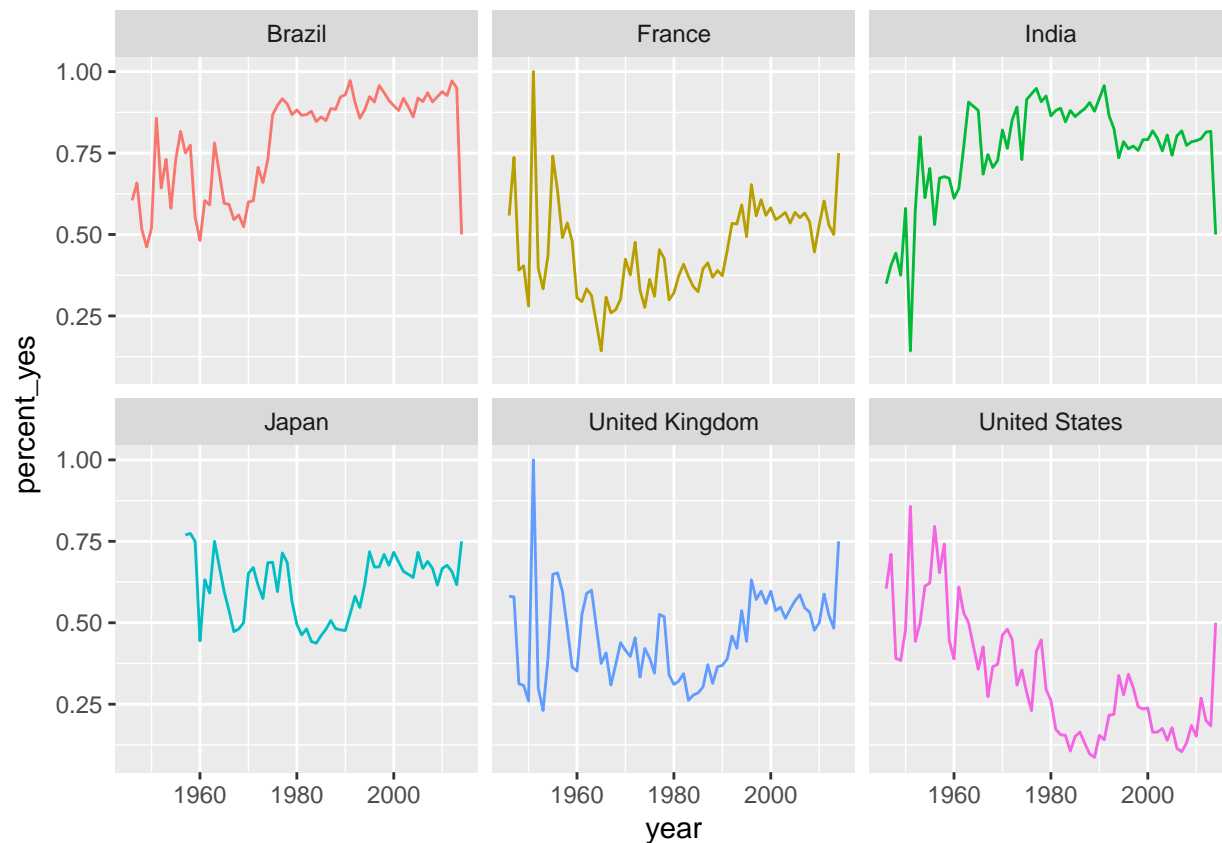
```
countries_6 <- c("United States", "United Kingdom",
                 "France", "Japan", "Brazil", "India")
countries_6_by_year = by_year_country %>%
  filter(country %in% countries_6)

ggplot(countries_6_by_year, aes(x=year,y=percent_yes,color=country)) + geom_line() + facet_wrap(~ count
```



The legend seems redundant so we remove it

```
ggplot(countries_6_by_year, aes(x=year,y=percent_yes,color=country)) + geom_line() + facet_wrap(~ count
```



Optional: Feel free to add countries that you're interested in!

MODELING

Linear regression on the United States

```
US_fit = lm(percent_yes ~ year, data= US_by_year)
summary(US_fit)

##
## Call:
## lm(formula = percent_yes ~ year, data = US_by_year)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20064 -0.08413 -0.01884  0.07237  0.40291
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.1619466   1.5138661    9.355 1.02e-13 ***
## year        -0.0069835   0.0007644   -9.135 2.50e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1259 on 66 degrees of freedom
## Multiple R-squared:  0.5584, Adjusted R-squared:  0.5517
```

```
## F-statistic: 83.46 on 1 and 66 DF, p-value: 2.502e-13
```

Fit models for 4 countries

```
UK_by_year = by_year_country %>%
  filter(country=="United Kingdom")
UK_fit = lm(percent_yes ~ year, data= UK_by_year)

France_by_year = by_year_country %>%
  filter(country=="France")
France_fit = lm(percent_yes ~ year, data= France_by_year)

China_by_year = by_year_country %>%
  filter(country=="China")
China_fit = lm(percent_yes ~ year, data= China_by_year)
```

Tidy models and combine them together

```
US_tidied = tidy(US_fit)
UK_tidied = tidy(UK_fit)
France_tidied = tidy(France_fit)
China_tidied = tidy(China_fit)
```

Analysis of Resolution Type

A different angle is to look at the types of resolutions. There are 6 issue types as below

```
unique(un_roll_call_issues$issue)
```

```
## [1] "Palestinian conflict"
## [2] "Nuclear weapons and nuclear material"
## [3] "Arms control and disarmament"
## [4] "Human rights"
## [5] "Colonialism"
## [6] "Economic development"
```

We want to know if countries have any preference or particular voting pattern for any issue. First, we join two datasets `un_votes` and `un_roll_call_issues` using `rcid`

```
head(un_votes)
```

```
## # A tibble: 6 × 3
##   rcid    country  vote
##   <dbl>    <chr>  <fctr>
## 1     3      Egypt abstain
## 2     3   Honduras    yes
## 3     3  Costa Rica    yes
## 4     3 El Salvador    yes
## 5     3    France     no
## 6     3   Uruguay    yes
```

```
head(un_roll_call_issues)
```

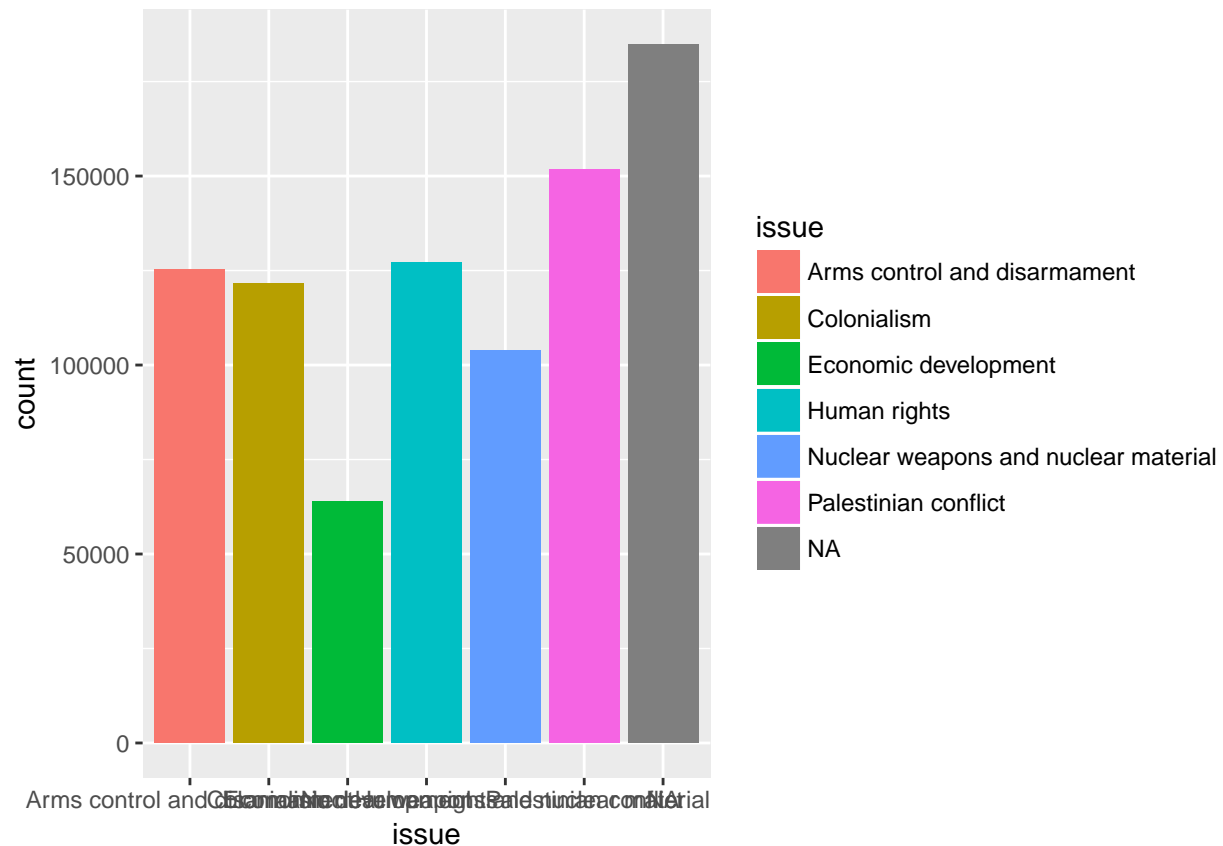
```
## # A tibble: 6 × 3
##   rcid short_name      issue
##   <dbl>    <chr>      <chr>
## 1    30      me Palestinian conflict
## 2    34      me Palestinian conflict
## 3    77      me Palestinian conflict
## 4   9002      me Palestinian conflict
## 5   9003      me Palestinian conflict
## 6   9004      me Palestinian conflict

df2 = merge(x=un_votes, y=un_roll_call_issues, by="rcid", all.x=TRUE)
head(df2)
```

```
##   rcid   country  vote short_name issue
## 1    3     Egypt abstain    <NA>  <NA>
## 2    3  Honduras   yes    <NA>  <NA>
## 3    3  Costa Rica   yes    <NA>  <NA>
## 4    3 El Salvador   yes    <NA>  <NA>
## 5    3    France    no    <NA>  <NA>
## 6    3   Uruguay   yes    <NA>  <NA>
```

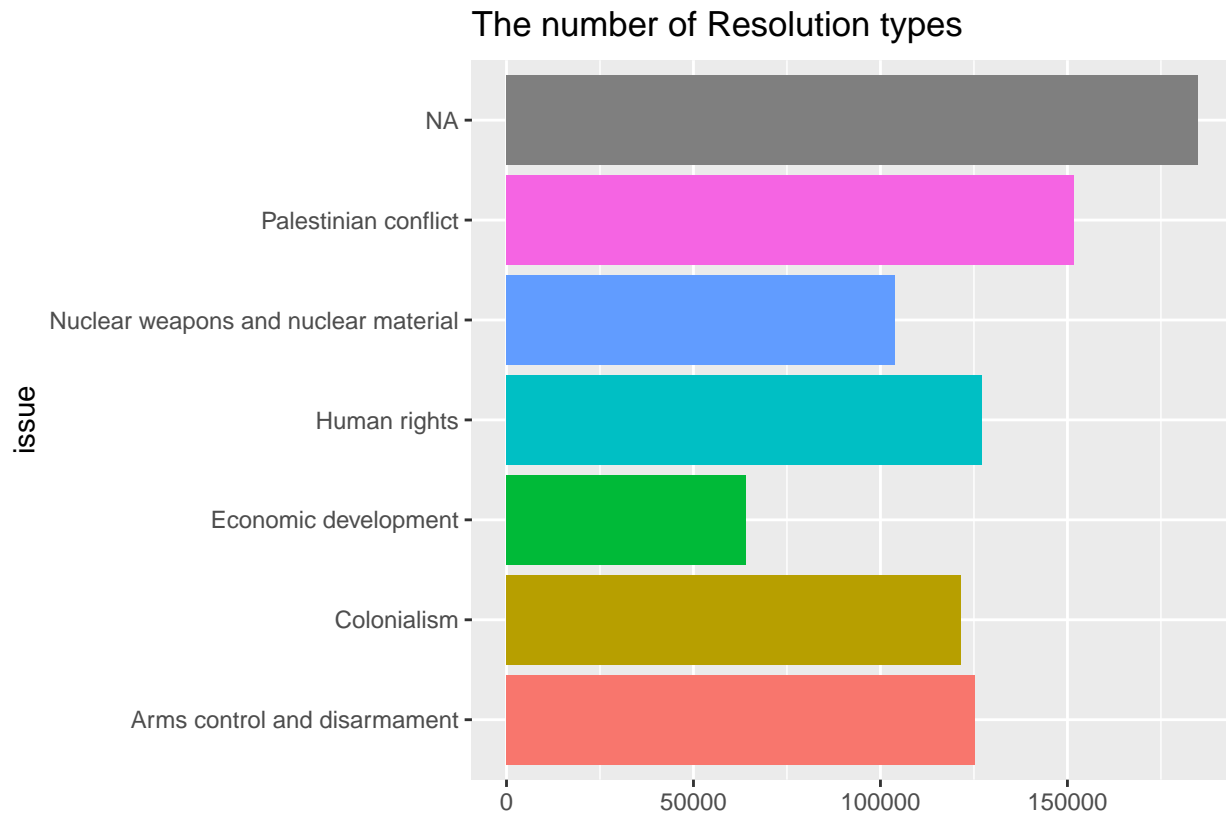
Plotting the data frame to see the number of resolutions by issue

```
df2 %>%
  ggplot(aes(x=issue)) +
  geom_bar(aes(fill=issue))
```



Adding features to enhance the look

```
df2 %>%
  ggplot(aes(x=issue)) +
  geom_bar(aes(fill=issue)) +
  coord_flip() +
  theme(legend.position="none") +
  ggtitle("The number of Resolution types") +
  ylab("")
```



From the chart, Palestinian conflict is the major concern of United Nation during 1946-2014, following by Human rights, Arms control and disarmament and Colonialism. Interestingly, economic development gets the least attention.

Another way to get a similar result is to group by the dataset by issue as follows:

```
by_issue = df2 %>%
  group_by(issue) %>%
  summarize(total = n(),
             percent_yes = mean(vote == "yes"))
head(by_issue)
```

```
## # A tibble: 6 × 3
##           issue total percent_yes
##           <chr> <int>      <dbl>
## 1 Arms control and disarmament 125332  0.8296046
## 2 Colonialism 121523  0.7952486
## 3 Economic development  63915  0.8253931
## 4 Human rights 127195  0.7495814
## 5 Nuclear weapons and nuclear material 103804  0.8096123
## 6 Palestinian conflict 151624  0.8379412
```

All resolutions but Human Rights has 80% and above consensus (“yes” votes). Human Rights Resolutions have 75% votes with “yes”.

Now we want to look at the data not only by issue but also by country

```
by_issue_country = df2 %>%
  group_by(issue, country) %>%
  summarize(total = n(),
             percent_yes = mean(vote == "yes"))
head(by_issue_country)
```

```
## Source: local data frame [6 x 4]
## Groups: issue [1]
##
##           issue          country total percent_yes
##           <chr>         <chr> <int>      <dbl>
## 1 Arms control and disarmament Afghanistan    787    0.8729352
## 2 Arms control and disarmament Albania         505    0.6594059
## 3 Arms control and disarmament Algeria         785    0.8522293
## 4 Arms control and disarmament Andorra         325    0.6246154
## 5 Arms control and disarmament Angola          591    0.9018613
## 6 Arms control and disarmament Antigua and Barbuda 562    0.9448399
```

Let's take US as an example to see the country's voting pattern on different issues

```
US_by_issue = by_issue_country %>%
  filter(country=="United States")
US_by_issue
```

```
## Source: local data frame [7 x 4]
## Groups: issue [7]
##
##           issue          country total percent_yes
##           <chr>         <chr> <int>      <dbl>
## 1 Arms control and disarmament United States    834    0.2961631
## 2 Colonialism United States    955    0.1706806
## 3 Economic development United States    448    0.1674107
## 4 Human rights United States    871    0.2732491
## 5 Nuclear weapons and nuclear material United States    696    0.2298851
## 6 Palestinian conflict United States   1026    0.1354776
## 7 <NA> United States   1516    0.4267810
```

The United States seems to disagree with most resolutions, lowest “yes” voting is to Palestinian conflict.

Sources:

<https://github.com/dgrtwo/unvotes>

<https://www.kaggle.com/karimkardous/vote-dynamics/code>