

# UNvotes

*Hanh Nguyen*

```
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)
library(fpc)
```

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
```

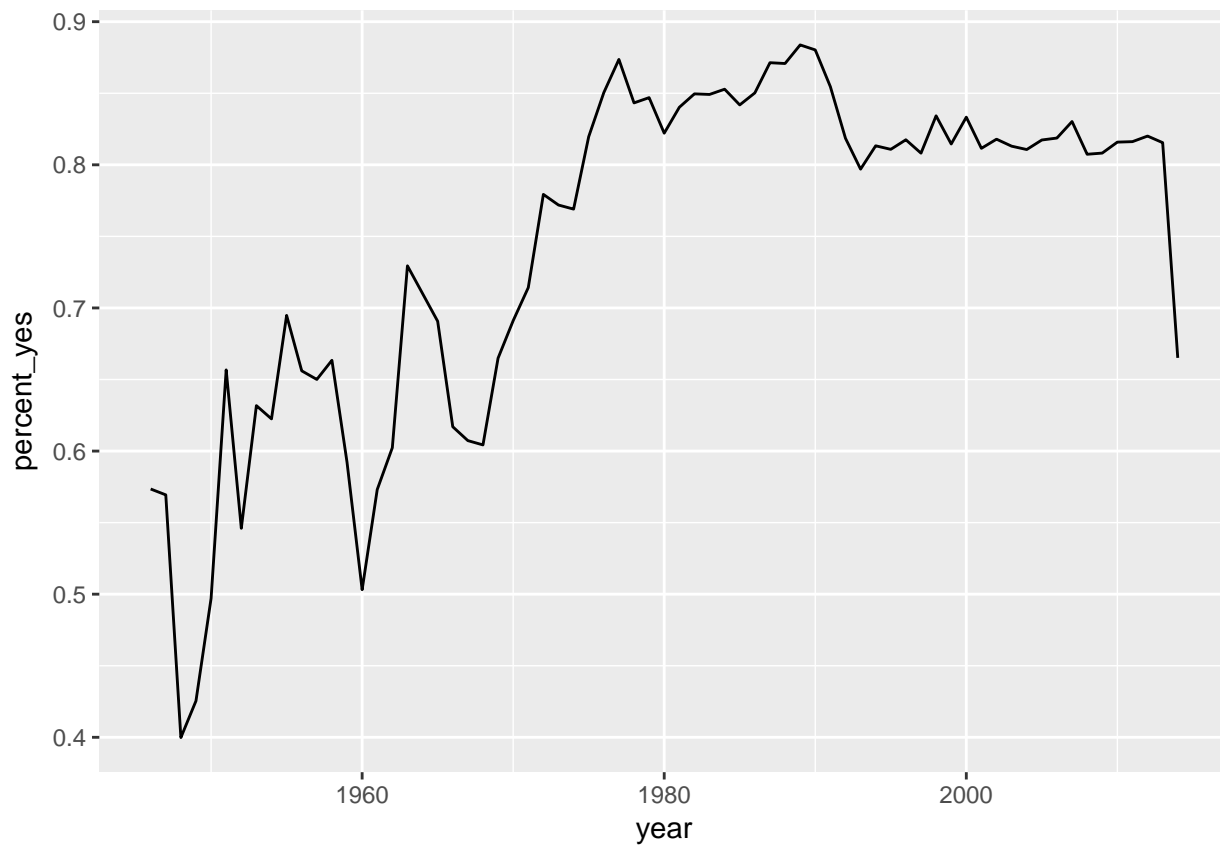
To see the voting pattern by year, we group the df by year using group\_by() function

```
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

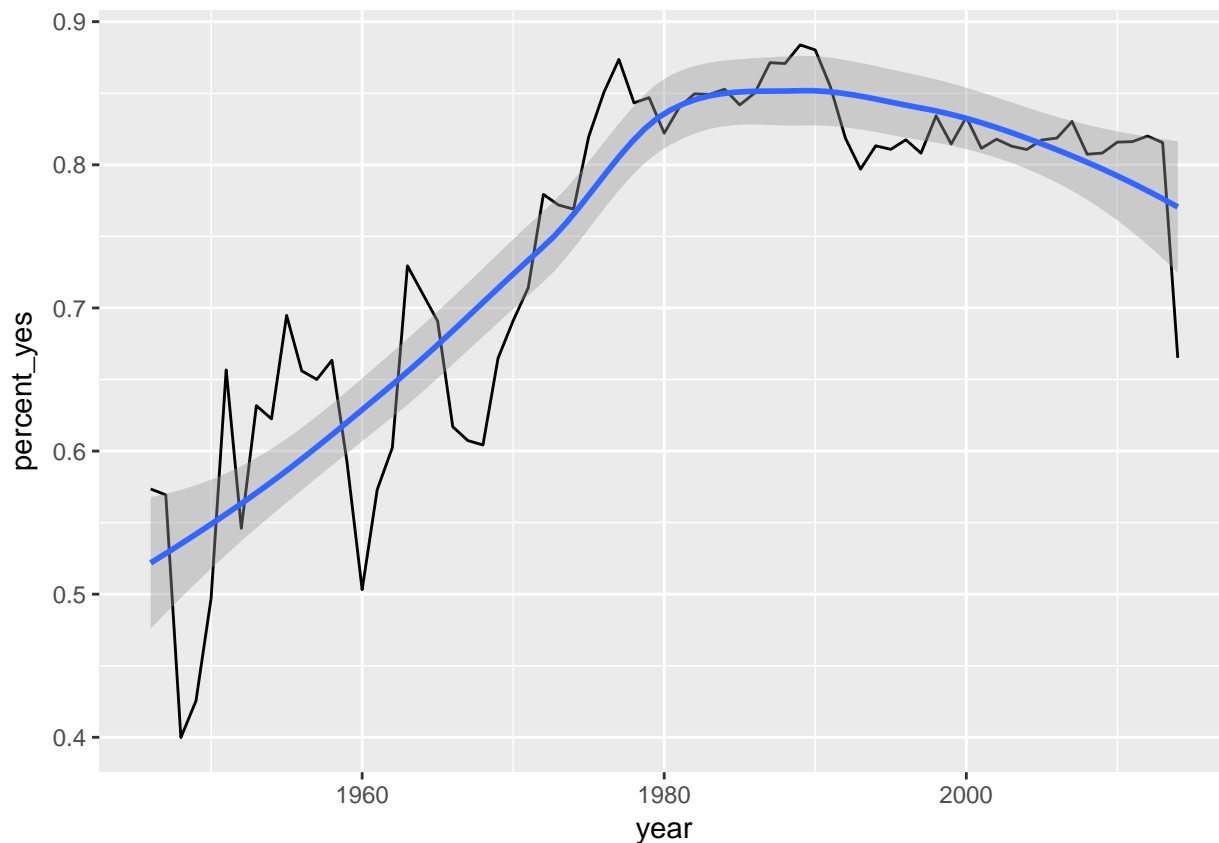
```
ggplot(by_year, aes(year, percent_yes)) +
  geom_line()
```



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]
```

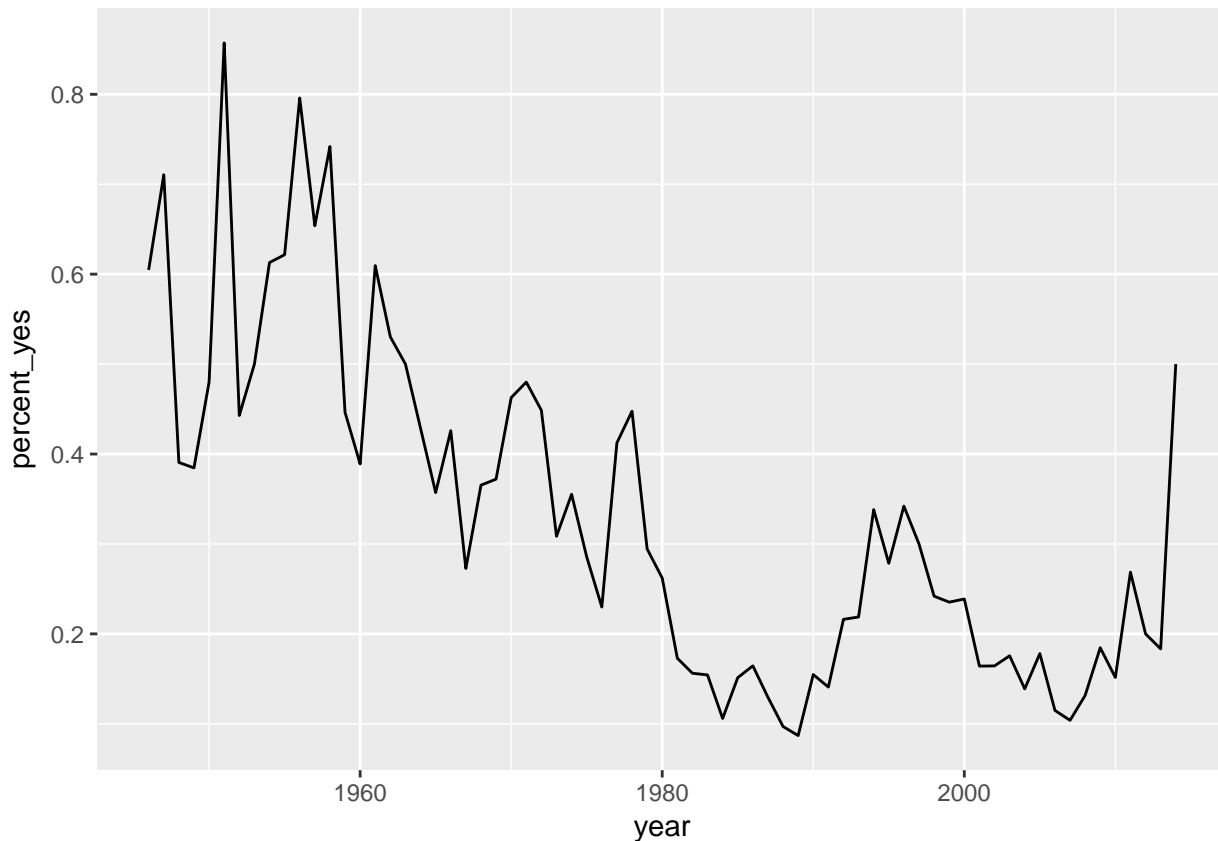
```
## Groups: year [1]
##
##   year                country total percent_yes
##   <dbl>                <chr> <int>         <dbl>
## 1  1946                Afghanistan    17    0.4117647
## 2  1946                Argentina     43    0.6976744
## 3  1946                Australia     43    0.5581395
## 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. Here we're interested in 8 most powerful countries in 2017 and see how their historical voting behaviors are. According to <https://www.usnews.com/news/best-countries/power-full-list>, the Power subranking is based on an equally weighted average of scores from five country attributes that related to a country's power: a leader, economically influential, politically influential, strong international alliances and strong military alliances. The Power subranking score had a 7 percent weight in the overall Best Countries ranking.

Those countries are the United States, Russia, China, the UK, Germany, France, Japan and Israel.

Before we get their data, we need to find the exact names used. I will demonstrate one example, suppose we look for Russia.

```
subset(un_votes, country=="Russia")
```

```
## # A tibble: 0 × 3
## # ... with 3 variables: rcid <dbl>, country <chr>, vote <fctr>
```

It seems that Russia is not the exact name used in the dataset!

```
russia_find = grepl("Russia", un_votes$country)
russia_df = un_votes[russia_find,]
head(russia_df)
```

```
## # A tibble: 6 × 3
##   rcid      country  vote
##   <dbl>    <chr> <fctr>
## 1     3 Russian Federation    no
## 2     4 Russian Federation    yes
## 3     5 Russian Federation    yes
## 4     6 Russian Federation    no
```

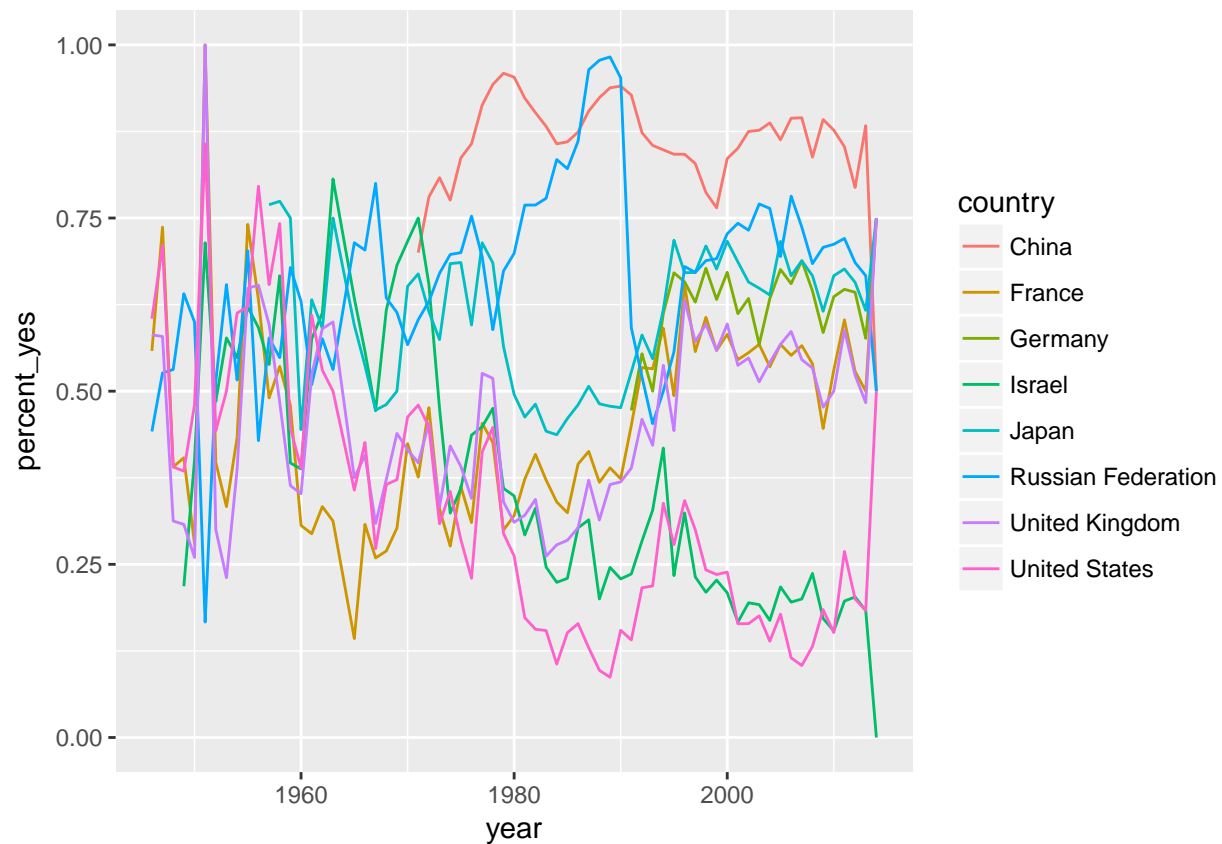


```
## 5      7 Russian Federation      no
## 6      8 Russian Federation      yes
```

So the right name is Russian Federation! Running the same commands, we will be able to figure out the names for these 8 countries

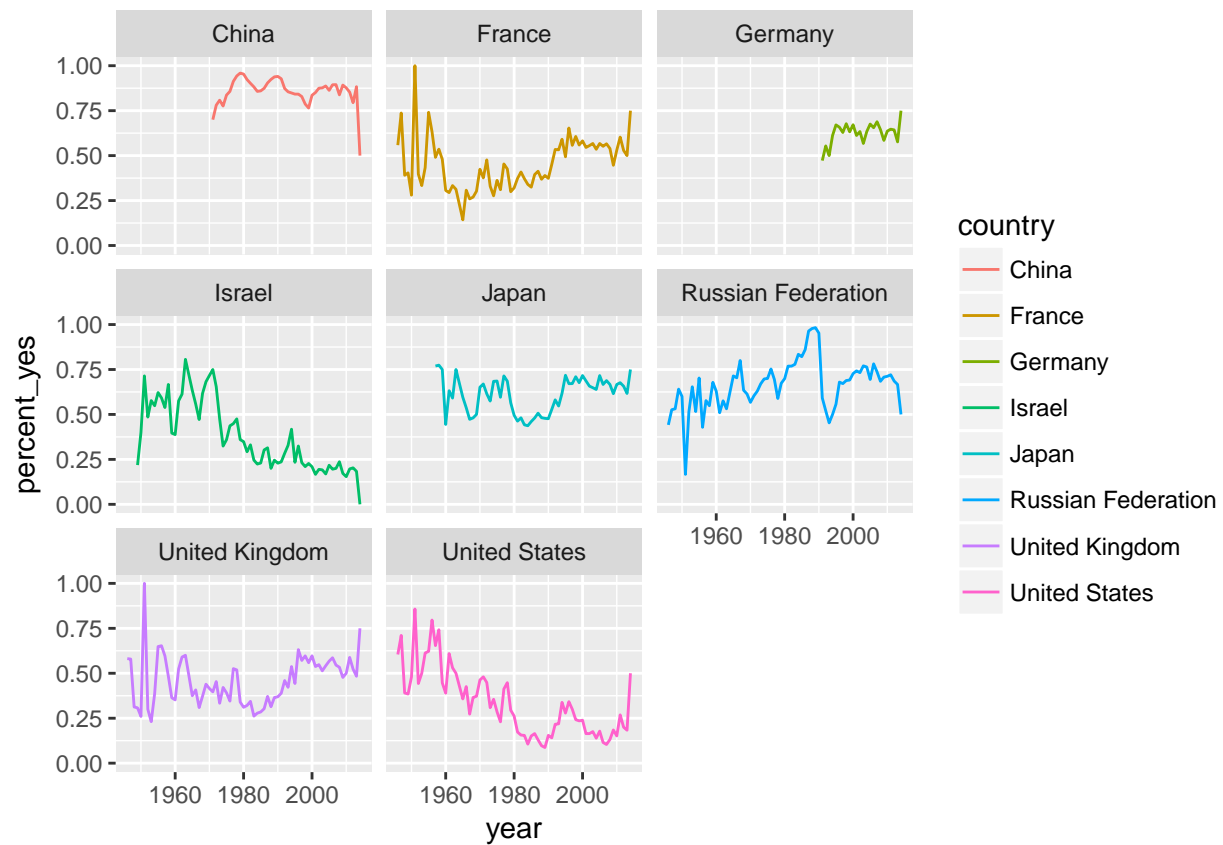
```
countries_8 <- c("United States", "Russian Federation", "China", "United Kingdom", "Germany", "France", "Israel")
countries_8_by_year = by_year_country %>%
  filter(country %in% countries_8)
```

```
ggplot(countries_8_by_year, aes(x=year, y=percent_yes, color=country)) + geom_line()
```



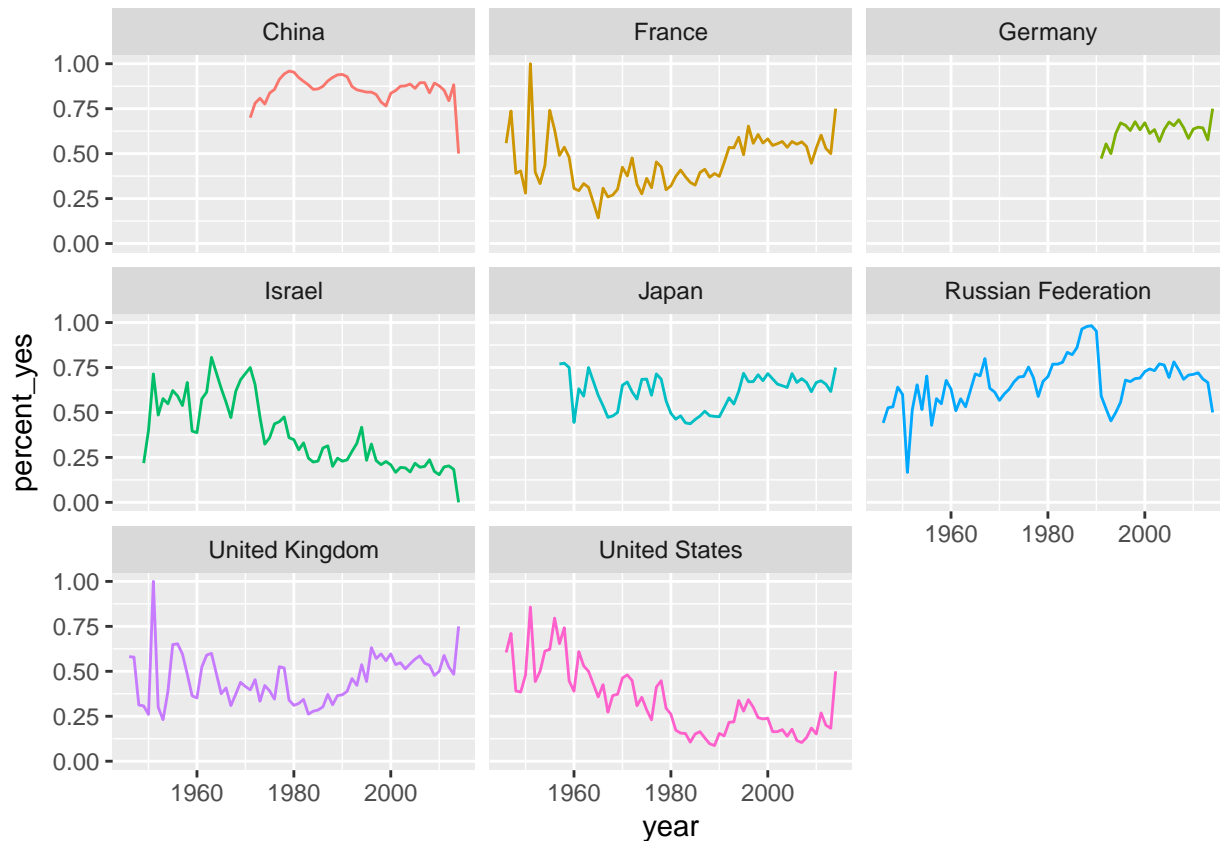
However, this type of graph could be tough to read. The alternative way is faceting.

```
ggplot(countries_8_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_8_by_year, aes(x=year,y=percent_yes,color=country)) + geom_line() + facet_wrap(~ count
```



China consistently had very high percents of “yes” votes except a drop in 2014. Japan had a bit lower percents fluctuating around .5 and .75. The United States started at a similar level but steadily was voting more “no” or “abstain” and its “yes” percents of votes even lowered to .12 during 1980s. Israel’s voting pattern also had a similar movement to the United States’s.

Optional: Feel free to explore countries that you’re interested in!

## 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)
```

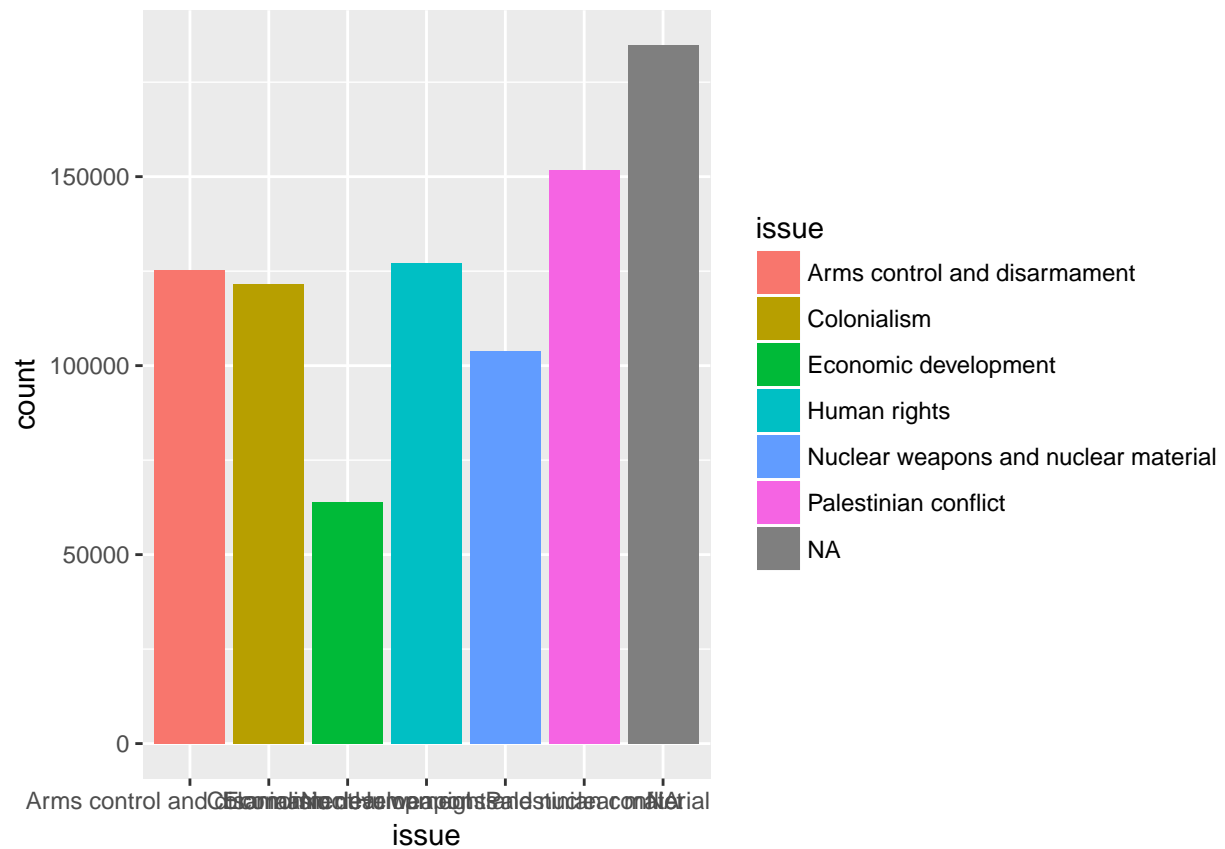
```
## # A tibble: 6 × 3
##   rcid short_name      issue
##   <dbl>      <chr>      <chr>
## 1    30      me Palestinian conflict
## 2    34      me Palestinian conflict
## 3    77      me Palestinian conflict
## 4  9002      me Palestinian conflict
## 5  9003      me Palestinian conflict
## 6  9004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

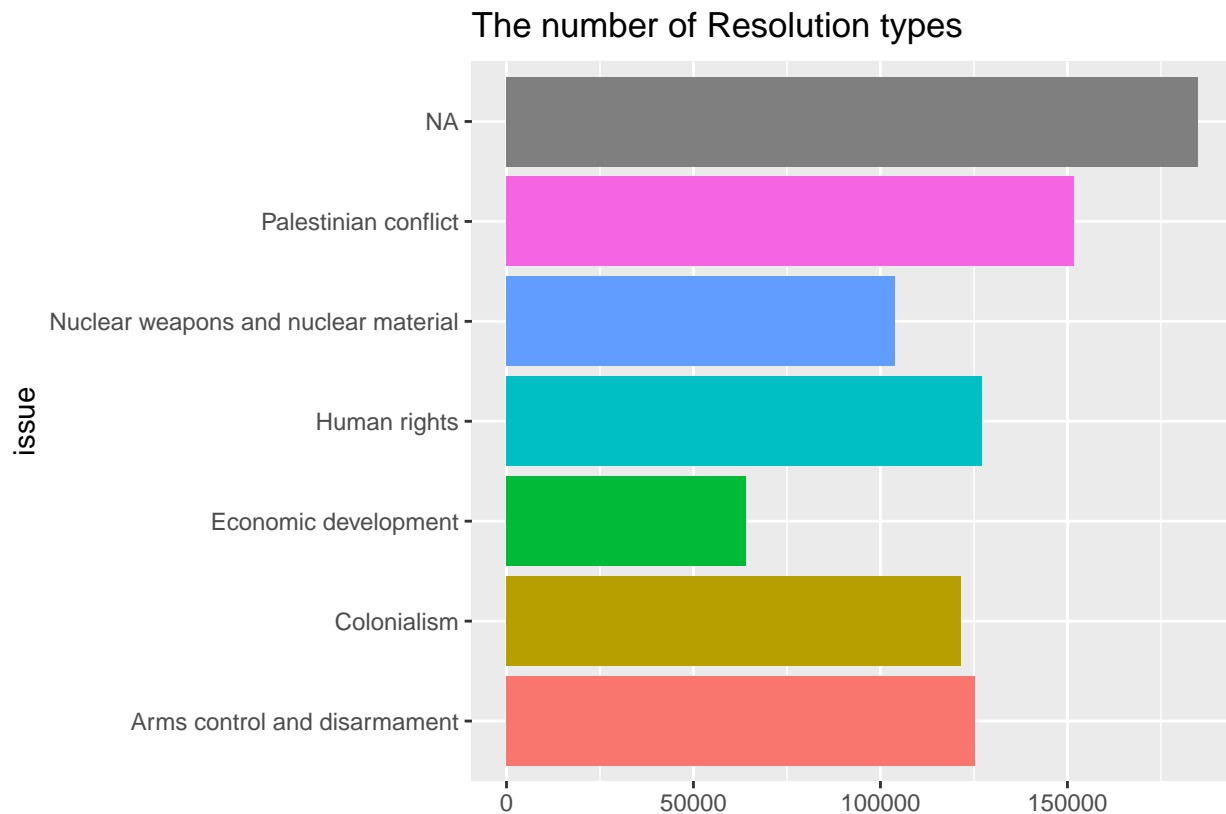
Plotting the data frame to see the number of resolutions by issue

```
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

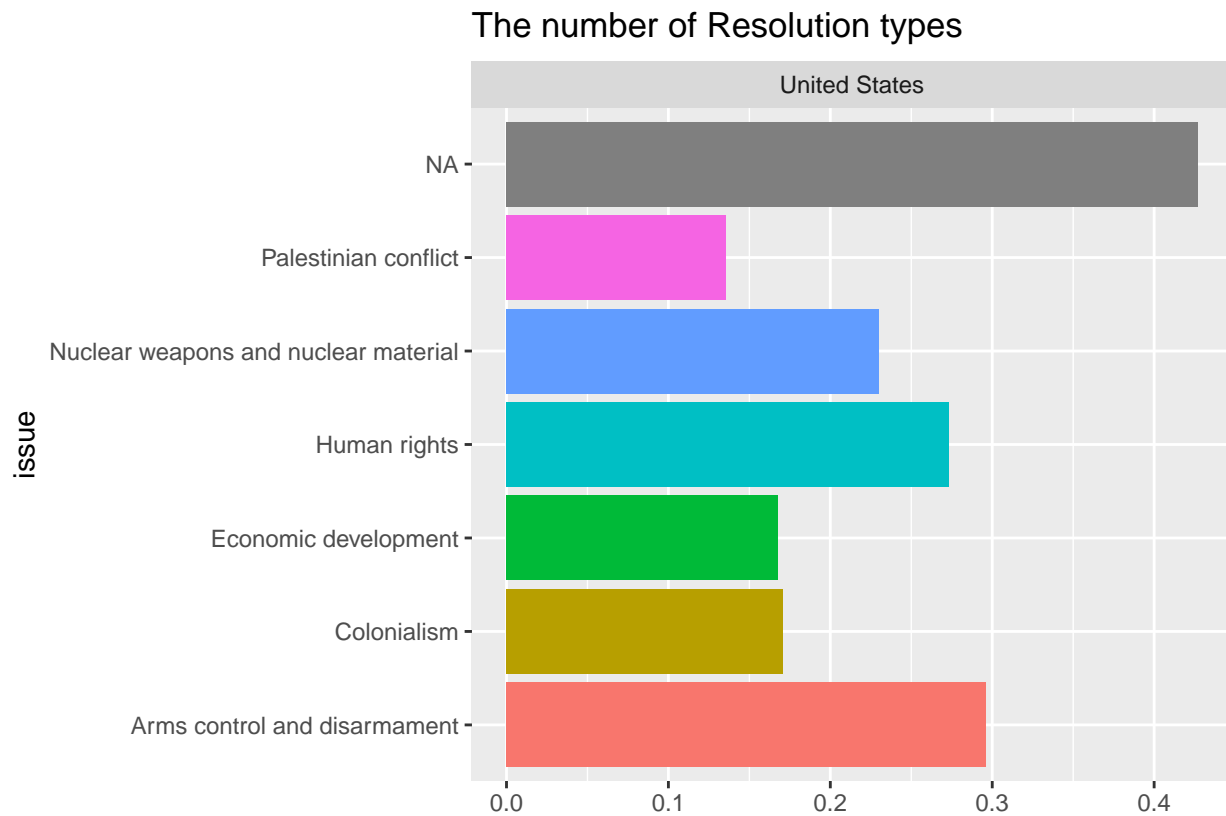
```
US_by_issue = by_issue_country %>%
  filter(country=="United States")
```

```
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
```

Making the plot

```
US_by_issue %>%
  ggplot(aes(x=issue,y=percent_yes)) +
  geom_bar(stat="identity",aes(fill=issue)) +
  coord_flip() +
  theme(legend.position="none") +
  ggtitle("The number of Resolution types") +
  ylab("") +
  facet_wrap(~ country)
```



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

Again, we want to see 8 countries at one view for comparison.

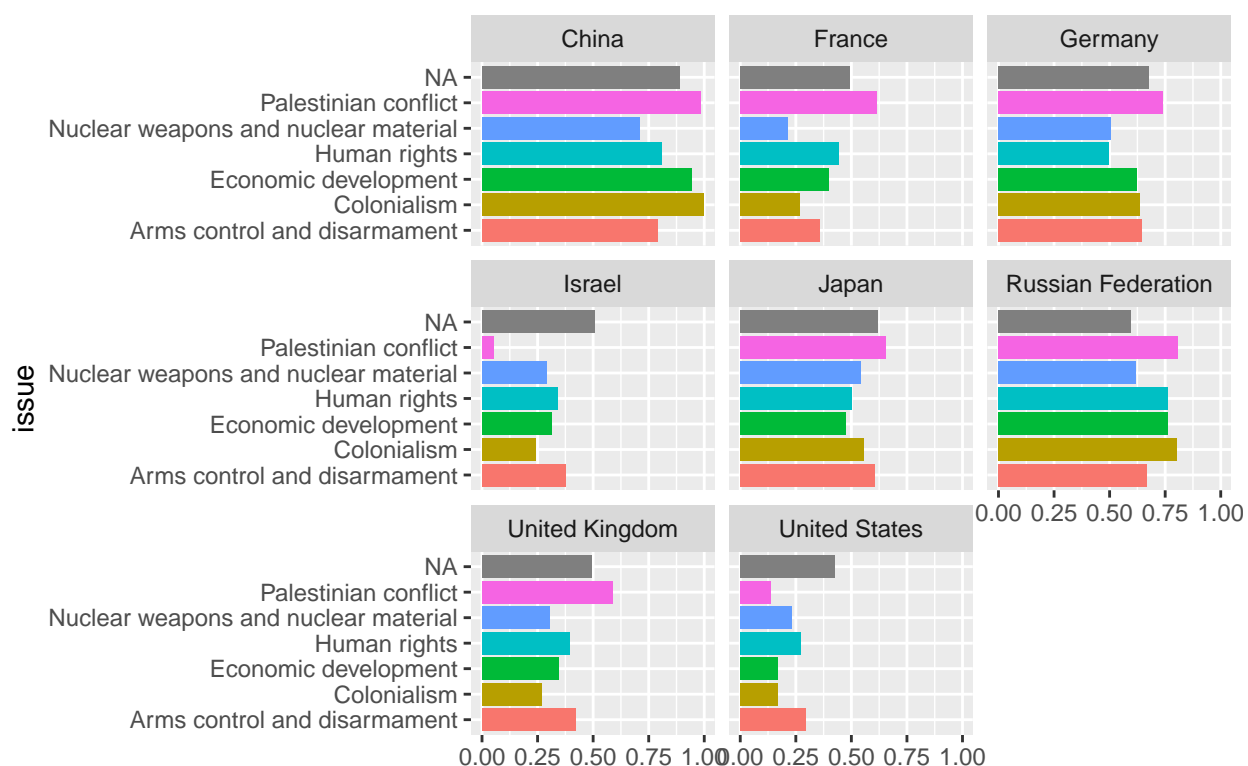
```
countries_8 <- c("United States", "Russian Federation", "China", "United Kingdom", "Germany", "France", "United States")

countries_8_by_issue = by_issue_country %>%
  filter(country %in% countries_8)

countries_8_by_issue %>%
  ggplot(aes(x=issue, y=percent_yes)) +
  geom_bar(stat="identity", aes(fill=issue)) +
  coord_flip() +
  theme(legend.position="none") +
  ggtitle("The number of Resolution types") +
  ylab("") +
  facet_wrap(~ country)
```



## The number of Resolution types



We want to use k-means clustering to subset countries into smaller groups of different voting behaviors. First, we remove countries that have small number of votes because the percentage might not be meaningful with small sample sizes.

```
# remove countries with total votes less than 10 for each period each issue
```

```
by_issue_country = by_issue_country %>%
```

```
  filter(total >= 100)
```

```
dff1 = subset(by_issue_country, select=-total)
```

```
dff1$issue[dff1$issue=="Arms control and disarmament"] = "Arms"
```

```
dff1$issue[dff1$issue=="Human rights"] = "Human"
```

```
dff1$issue[dff1$issue=="Nuclear weapons and nuclear material"] = "Nuclear"
```

```
dff1$issue[dff1$issue=="Palestinian conflict"] = "Palestinian"
```

```
dff1$issue[dff1$issue=="Economic development"] = "Economic"
```

```
dff1$issue[is.na(dff1$issue)] = "Other"
```

```
dff2 = dff1 %>%
```

```
  spread(issue, percent_yes)
```

```
head(dff2)
```

```
## # A tibble: 6 × 8
```

```
##       country      Arms Colonialism Economic      Human      Nuclear
##       <chr>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Afghanistan 0.8729352 0.9055300 0.9158654 0.8583851 0.9044684
## 2 Albania      0.6594059 0.8357558 0.7949640 0.7031746 0.5951327
## 3 Algeria      0.8522293 0.9645293 0.9604938 0.8549811 0.8836141
## 4 Andorra      0.6246154 0.7727273 0.7086614 0.5267857 0.4880000
## 5 Angola       0.9018613 0.9775281 0.9867987 0.8319605 0.9196787
## 6 Antigua and Barbuda 0.9448399 0.9406780 0.9488189 0.8458498 0.9287305
## # ... with 2 more variables: Other <dbl>, Palestinian <dbl>
```

```
summary(dff2)
```

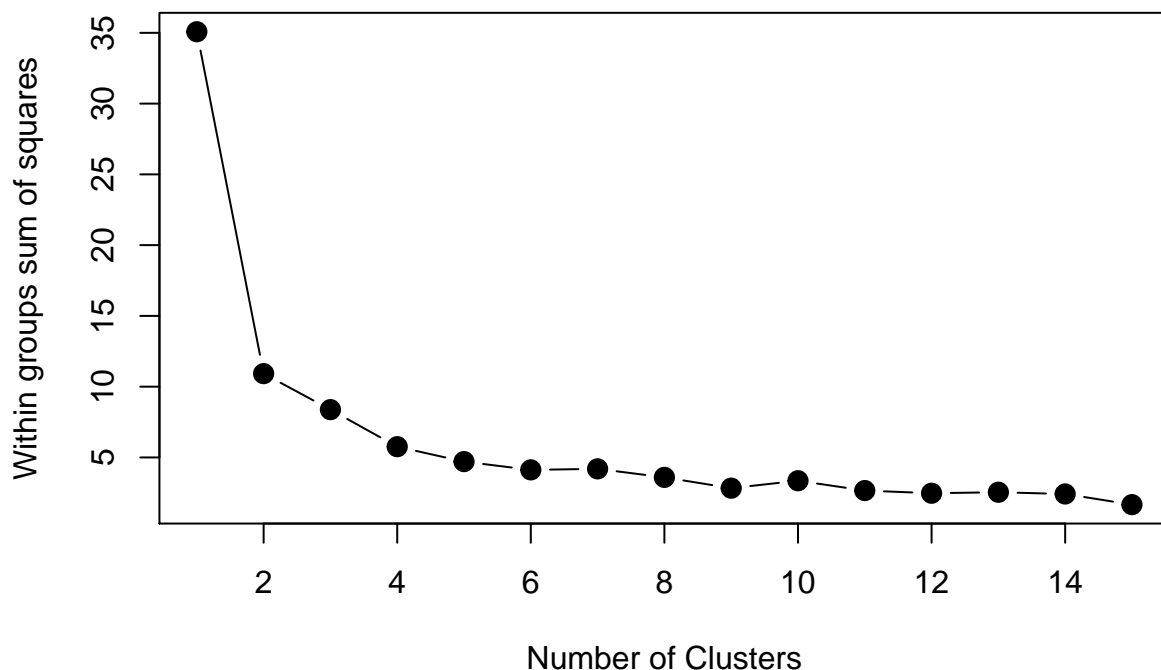
```
##      country      Arms      Colonialism      Economic
## Length:197      Min.   :0.2962      Min.   :0.1707      Min.   :0.1674
## Class :character 1st Qu.:0.7025      1st Qu.:0.7316      1st Qu.:0.7390
## Mode  :character Median :0.8998      Median :0.8711      Median :0.9023
##                      Mean  :0.8230      Mean  :0.8053      Mean  :0.8276
##                      3rd Qu.:0.9422      3rd Qu.:0.9355      3rd Qu.:0.9416
##                      Max.   :0.9924      Max.   :1.0000      Max.   :1.0000
##                      NA's    :2                      NA's    :11
##      Human      Nuclear      Other      Palestinian
## Min.   :0.2732      Min.   :0.2016      Min.   :0.4268      Min.   :0.01762
## 1st Qu.:0.6674      1st Qu.:0.6411      1st Qu.:0.7079      1st Qu.:0.75235
## Median :0.8071      Median :0.9048      Median :0.8159      Median :0.88741
## Mean   :0.7398      Mean   :0.7951      Mean   :0.7924      Mean   :0.82981
## 3rd Qu.:0.8478      3rd Qu.:0.9463      3rd Qu.:0.8824      3rd Qu.:0.95610
## Max.   :0.9863      Max.   :0.9930      Max.   :0.9744      Max.   :1.00000
##                      NA's    :3                      NA's    :8                      NA's    :1
```

```
dff2 =na.omit(dff2)
```

We will choose the number of clusters by computing Within sum of squares in clusters.

```
wss <- (nrow(dff2[, -1]) - 1) * sum(apply(dff2[, -1], 2, var))
for (i in 2:15) wss[i] <- sum(kmeans(dff2[, -1], centers=i)$withinss)
plot(wss, xlim = c(1, 15), type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares",
     main="Assessing the Optimal Number of Clusters with the Elbow Method",
     pch=20, cex=2)
```

## Assessing the Optimal Number of Clusters with the Elbow Method



It seems like 2 clusters is a good option.

```

set.seed(20)
kmc <- kmeans(dff2[, -1], centers=2, iter.max=1000)
kmc

## K-means clustering with 2 clusters of sizes 48, 138
##
## Cluster means:
##      Arms Colonialism Economic      Human      Nuclear      Other
## 1 0.6066172    0.5706019 0.5685406 0.5158260 0.5052855 0.6320169
## 2 0.9023907    0.8938900 0.9177192 0.8272374 0.9036598 0.8497849
##   Palestinian
## 1    0.6566774
## 2    0.9041908
##
## Clustering vector:
##   [1] 2 1 2 1 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2
##  [36] 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 1 2 1 1 2 2 1 2 1 2 1 2 2
##  [71] 2 2 2 2 2 1 1 2 2 2 2 1 1 1 2 1 2 2 2 2 1 2 2 2 1 2 2 2 2 1 1 1 1 2 2
## [106] 2 2 2 2 1 2 2 2 1 1 1 2 2 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 1 1 2 2 1
## [141] 2 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 1
## [176] 1 2 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 5.952613 4.968272
## (between_SS / total_SS =  68.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"

```

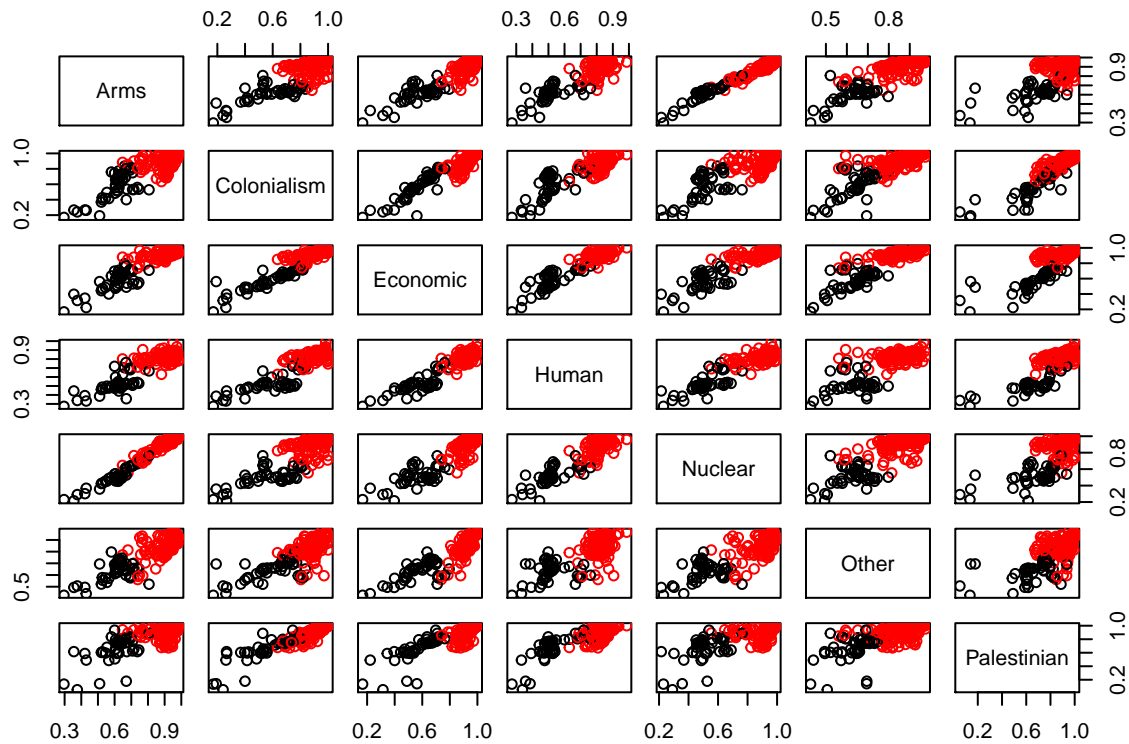
k-means clustering offers two clusters. There are 48 countries in group 1 and 138 countries in group 2. It shows that group 2 is pro- UN resolutions while group 1 appears to disagree more with the resolutions. Looking at cluster means, group 2 consistently voted “yes” much more than group 1 across all kinds of resolutions.

Visualizing clusters

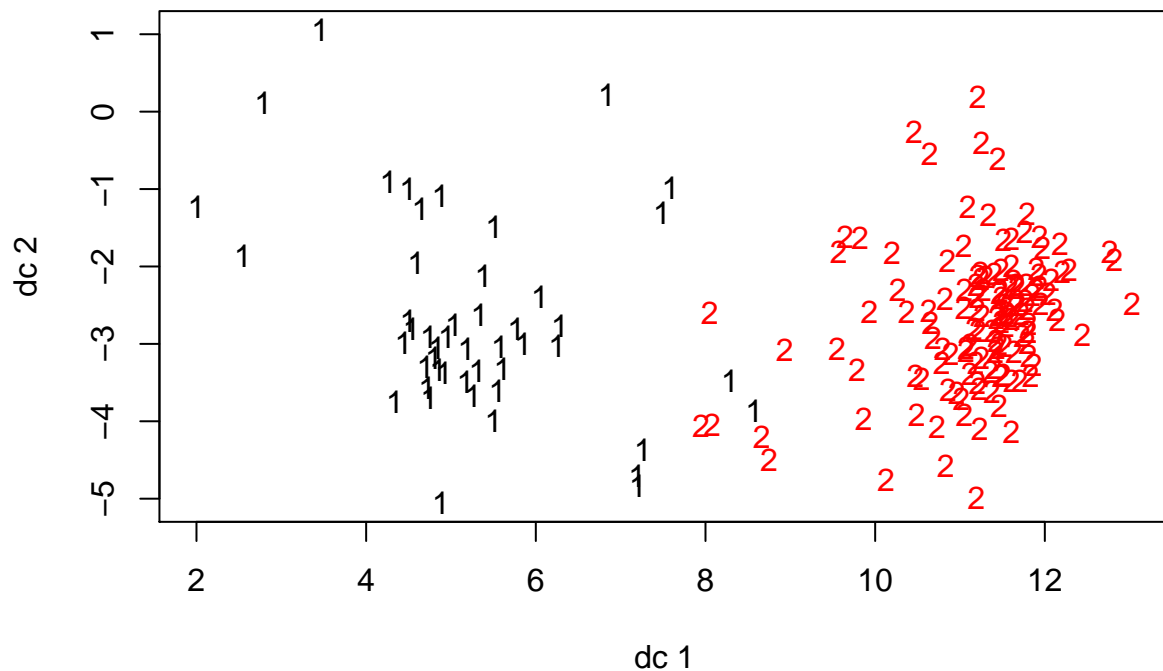
```

with(dff2, pairs(dff2[, -1], col=c(1:2)[kmc$cluster]))

```



```
plotcluster(dff2[,-1], kmc$cluster)
```



Back to 8 countries that we're interested in, we need to figure which country belongs to which cluster.

```
# append cluster assignment
```

```
dff2_cluster <- data.frame(dff2, kmc$cluster)
```

```
head(dff2_cluster)
```

```
##          country      Arms Colonialism  Economic    Human    Nuclear
## 1      Afghanistan 0.8729352   0.9055300 0.9158654 0.8583851 0.9044684
```

```
## 2          Albania 0.6594059 0.8357558 0.7949640 0.7031746 0.5951327
## 3          Algeria 0.8522293 0.9645293 0.9604938 0.8549811 0.8836141
## 4          Andorra 0.6246154 0.7727273 0.7086614 0.5267857 0.4880000
## 5          Angola 0.9018613 0.9775281 0.9867987 0.8319605 0.9196787
## 6 Antigua and Barbuda 0.9448399 0.9406780 0.9488189 0.8458498 0.9287305
##          Other Palestinian kmc.cluster
## 1 0.7192362 0.9427027 2
## 2 0.6517533 0.8220859 1
## 3 0.8651786 0.9830688 2
## 4 0.7078652 0.7774936 1
## 5 0.9334443 0.9713877 2
## 6 0.9365427 0.8853503 2
```

```
dim(dff2_cluster)
```

```
## [1] 186 9
```

```
countries_8_cluster = dff2_cluster %>%
  filter(country %in% countries_8)
countries_8_cluster
```

```
##          country      Arms Colonialism Economic      Human      Nuclear
## 1          China 0.7914764 0.9965870 0.9418283 0.8065434 0.7098540
## 2          France 0.3555018 0.2663755 0.3968610 0.4449541 0.2151163
## 3          Germany 0.6443149 0.6381323 0.6241611 0.4960212 0.5072464
## 4          Israel 0.3762255 0.2416953 0.3150358 0.3384419 0.2908012
## 5          Japan 0.6048780 0.5577157 0.4740566 0.5012255 0.5437318
## 6 Russian Federation 0.6678657 0.8018868 0.7632743 0.7609195 0.6187050
## 7          United Kingdom 0.4216867 0.2675906 0.3458980 0.3926941 0.3015873
## 8          United States 0.2961631 0.1706806 0.1674107 0.2732491 0.2298851
##          Other Palestinian kmc.cluster
## 1 0.8894231 0.98474946 2
## 2 0.4936793 0.61635833 1
## 3 0.6777251 0.73868313 1
## 4 0.5064541 0.05226131 1
## 5 0.6181102 0.65310275 1
## 6 0.5982850 0.80859375 1
## 7 0.4947090 0.58909445 1
## 8 0.4267810 0.13547758 1
```

The result shows that China is classified in group 2 while the rest in group 1. This again confirms the charts above. Out of 8 countries, only China seems to agree with United Nations resolutions as it voted “yes” around 90% of the times. While the remaining 7 countries have more votes against United Nations resolutions.

Sources:

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

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

<https://stats.stackexchange.com/questions/31083/how-to-produce-a-pretty-plot-of-the-results-of-k-means-cluster-analysis>