Exploring the Eurovision Song Contest Results and Scoring

sergei dakov

2023-07-02

Intro

The Eurovision Song Contest (ESC) is a yearly event brought on with two main purposes:

1. unify Europe post World War II, and bring all countries together through art
2. popularize the newly emerging medium of television

Throughout the years the ESC made strides to achieve both those goals; the EBU, the governing body of the Eurovision, kept pushing advances in television broadcasting (color broadcasting, high definition), while Ukraines' victory in 2022 was possibly the most obvious example of EU citizens using the Eurovision as a display of public support to a single country.

Over the years, the competition changed the judging format multiple times, starting from closed jury voting, through public televotes, to a combination of both. The final major revision to the scoring system happened in 2016, under the new system each participating country would host a televote and a jury vote, each giving individual points. This system brought many peoples attention to the differences that often occur between the way the professional juries and general public rate the different songs and performances.

Using the data of all seven competitions from 2016 to 2023 (the competition was cancelled in 2020 due to the COVID-19 pandemic) we will explore these differences, and look for patterns in voting through the years.

we will be using the Tidyverse collection of libraries to streamline data processing- load in the library:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

The Data

Unfortunately, while there is a lot of data available on the Eurovision, there is no organized/easily accessible data that serves our purpose, thus we will scrape the data off wikipedia, to do so we will be using the Rvest package:

```
library(rvest)
```

```
##
## Attaching package: 'rvest'
```

```
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

Since the Wikipedia pages for the relevant competitions have multiple tables, we need to be able to quickly find and filter out only the relevant tables. The article pages for the competition are generally well structured, so we can automate the scraping process. In this analysis we will be focusing only on the finals of the competition in each year.

We define a function that allows us to discern if a given table might be relevant or not, one thing we can use is that all finals have 26 countries competing in them, while each semi-final has around 20, so we can use the number of rows of a table to know if a table pertains to a semi final or the finals. In addition, some pages list the locations of vignettes filmed for each country. Since we do not need this information, such tables can be filtered out

```
detect_relevant <- function(x) { #define function
  if (nrow(x)<=23){ #count the number of rows to filter tables too short to pertain to the finals
    return(FALSE)
  }
  if (any(str_detect(colnames(x),"Location"))){ # if a table has columns that pertain to location (like
    return(FALSE)
  }
  return(TRUE) #other tables are potential candidates
}
```

Using the Rvest package we can create a browser session, that will visit each page in order and collect the relevant tables, then we collect all the tables into one main dataframe

```
Eurovision_results <- tibble()  #initialize the main dataframe
relevant_years <- c(2016:2019,2021:2023) #define the relevant years, note 2020 is omitted due to cancel

for (i in relevant_years) {
current_url <- paste0("https://en.wikipedia.org/wiki/Eurovision_Song_Contest_",i) #create the url to vi
current_session <- session(current_url) #create session in given url

session_tables <- current_session %>% # collect all tables on the page
  html_nodes("table") %>% html_table()
session_tables <- session_tables[sapply(session_tables,detect_relevant)] #filter out only the potential


general_results <- session_tables[[3]] #the general results table contains each country, their final sc


extra_data <- session_tables[[2]] %>% select(Country,Language,`Songwriter(s)`) # this table contains in

detailed_breakdown_jury <- session_tables[[5]] # a detailed breakdown of how each jury voted
detailed_breakdown_public <- session_tables[[6]] # a detailed breakdown of how country voted in the tel
```

```
#due to the way the specific tables are formated the data reading has some faults that need to be corre
colnames(detailed_breakdown_public) <- detailed_breakdown_public[2,]  #some column names are read as va
colnames(detailed_breakdown_jury) <- detailed_breakdown_jury[2,]
colnames(general_results) <- str_remove(colnames(general_results),"\\[...") #removing annotations

if (is.character(general_results$Place)){ #in some cases, the editors elected to note a countries finis
general_results <- general_results %>% mutate(Place = parse_number(Place))
}

general_results <- general_results %>% mutate(Country = str_remove_all(Country,"\\[.\\]"))
extra_data <- extra_data %>% mutate(Language = str_remove_all(Language,"\\[.\\]"))

#main column names need to be identical to allow merging of the different tables
colnames(detailed_breakdown_public)[2] <- "Country"
colnames(detailed_breakdown_jury)[2] <- "Country"

#due to cell merging some headers are read as data, creating junk rows that need to be removed
detailed_breakdown_public <- detailed_breakdown_public %>%
  filter(!row_number() %in% c(1,2,3)) %>%
  select(-1) %>%
  mutate(n_vote = (ncol(.)-4))

detailed_breakdown_jury <- detailed_breakdown_jury %>%
  filter(!row_number() %in% c(1,2,3)) %>%
  select(-1) %>%
  select(-contains("Score")) %>%
  mutate(n_vote = (ncol(.)-1))
vote_breakdown <- detailed_breakdown_public %>% left_join(detailed_breakdown_jury,by="Country",suffix=c

#merge all the tables into one
adv_results <- general_results %>%
  left_join(extra_data,by="Country") %>%
  left_join(vote_breakdown,by="Country") %>%
  mutate(Year= i)
#add the table to the main dataframe
Eurovision_results <- Eurovision_results %>% bind_rows(adv_results)
# wait between queries, while the amount of queries made is quite small so it unlikely it would cause s
Sys.sleep(5)
}
```

Due to minor differences in how the articles for different years are formatted, some modifications need to be made to fix redundant columns or minor naming differences

```
#some fields have slightly different names in different years, as well as using both "Czech Republic" a
Eurovision_results <- Eurovision_results %>% select(-1) %>%
  unite("Jury_Score","Jury score","Jury vote score",sep = "",na.rm = T) %>%
  unite("Czech Republic_jury","Czech Republic_jury","Czechia_jury",sep="",na.rm = T) %>%
  unite("Czech Republic_public","Czech Republic_public","Czechia_public",sep="",na.rm = T) %>%
  mutate(Country =str_replace_all(Country,"Czechia","Czech Republic"))

#for ease of manipulation later, all spaces are removed from column names
colnames(Eurovision_results) <- str_replace_all(colnames(Eurovision_results)," ","_")
```

```
# some numeric columns are read in as character due to previously mentioned errors, convert them back t
Eurovision_results <- Eurovision_results %>%
  relocate('Year','n_vote_jury','n_vote_public',.after = last_col()) %>%
  mutate(across(8:101,~parse_number(.x))) %>%
  mutate(across(8:101,~replace_na(.x,replace = 0))) %>%
  mutate(Song = str_remove_all(Song,"\"")) %>%
  rename(Songwriters = `Songwriter(s)`) %>%
  mutate(Songwriter_count = str_count(Songwriters,"[a-z][A-Z]")+1)
```

Initial Data Exploration

We can now begin to explore our data, to start we can look at hte top ten best scoring performances:

```
Eurovision_results %>%
  arrange(-Points) %>%
  head(10) %>%
  select(Country,Artist,Song,Points,Place,Year,number_voted=n_vote_public)
```

```
## # A tibble: 10 x 7
##    Country   Artist           Song             Points Place  Year number_voted
##    <chr>     <chr>            <chr>             <int> <dbl> <int>        <dbl>
##  1 Portugal  Salvador Sobral  Amar pelos dois     758     1  2017           42
##  2 Ukraine   Kalush Orchestra Stefania            631     1  2022           40
##  3 Bulgaria  Kristian Kostov  Beautiful Mess      615     2  2017           42
##  4 Sweden    Loreen           Tattoo              583     1  2023           38
##  5 Ukraine   Jamala           1944                534     1  2016           42
##  6 Israel    Netta            Toy                 529     1  2018           43
##  7 Finland   Käärijä          Cha Cha Cha         526     2  2023           38
##  8 Italy     Måneskin         Zitti e buoni       524     1  2021           39
##  9 Australia Dami Im          Sound of Silence    511     2  2016           42
## 10 France    Barbara Pravi    Voilà               499     2  2021           39
```

Of note here is that the Bulgarian entry from 2017 performed well despite not winning, and in fact has globally placed above multiple winners. Similarly Finland's entry from 2023 has outperformed the 2021 winner it is worth noting, too, that later years (2021 and on) have fewer countries participate in the vote, thus lowering the amount of possible points. Despite that, the Bulgarian entry has outperformed two winners with a similar voting size.

We can also arrange the data to see which are the best performing countries on average

```
Eurovision_results %>%
  select(Country,Place) %>%
  group_by(Country) %>%
  summarise(Times_paricipated=n(),mean_place = mean(Place)) %>%
  arrange(mean_place)
```

```
## # A tibble: 42 x 3
##    Country         Times_paricipated mean_place
##    <chr>                       <int>      <dbl>
##  1 Russia                          3       5
##  2 Italy                           7       5.71
##  3 Sweden                          7       5.86
```

```
##  4 North Macedonia                1       7
##  5 Bulgaria                       4       7.75
##  6 Ukraine                        6       9
##  7 Moldova                        5      10.2
##  8 Australia                      6      10.7
##  9 Norway                         6      10.7
## 10 Lithuania                      5      10.8
## # i 32 more rows
```

of note is that while Russia places the best on average in the years it participates, it has only participated in the finals 3 times out of 7, in part due to being barred from entry after the invasion of Ukraine in contrast, the other two best performing countries are Italy and Sweden who both took part in every one of the finals while maintaining strong placements from year to year

in fact, we can check the participation streak of each country and observe which qualify on the most consistent basis

```
Eurovision_results %>%
  select(Country,Place) %>%
  group_by(Country) %>%
  summarise(n=n(),mean_place = mean(Place)) %>%
  arrange(-n)
```

```
## # A tibble: 42 x 3
##    Country            n mean_place
##    <chr>          <int>      <dbl>
##  1 France             7      12.7
##  2 Germany            7      22.3
##  3 Italy              7       5.71
##  4 Spain              7      19.6
##  5 Sweden             7       5.86
##  6 United Kingdom     7      20.3
##  7 Australia          6      10.7
##  8 Cyprus             6      14.2
##  9 Israel             6      13.5
## 10 Netherlands        6      12.5
## # i 32 more rows
```

From this table we can see that Sweden is the only country not from the "big five" (France, Germany, Italy, Spain and the UK) to make an appearance in the finals of each year

Another thing to explore is the amount of credited song writers on each song:

```
Eurovision_results %>%
  arrange(-Songwriter_count) %>%
  select(Country,Song,Place,Year,Songwriter_count,Songwriters) %>%
  head(10)
```

```
## # A tibble: 10 x 6
##    Country    Song                   Place  Year Songwriter_count Songwriters
##    <chr>      <chr>                  <dbl> <int>            <dbl> <chr>
##  1 San Marino Adrenalina                22  2021               10 "Joy DebLinn~
##  2 Poland     Solo                      19  2023                7 "Maria Brobe~
```

```
##  3 Croatia     My Friend                 13  2017              6 "Jacques Hou~
##  4 Norway      Spirit in the Sky          6  2019              6 "Fred BuljoT~
##  5 Lithuania   Discoteque                 8  2021              6 "Mantas Bani~
##  6 Armenia     Snap                      20  2022              6 ".mw-parser-~
##  7 Moldova     Trenulețul                 7  2022              6 "Vasile Adva~
##  8 Sweden      Tattoo                     1  2023              6 "Peter Bostr~
##  9 Italy       No Degree of Separation   16  2016              5 "Federica Ab~
## 10 Malta       Walk on Water             12  2016              5 "Lisa Desmon~
```

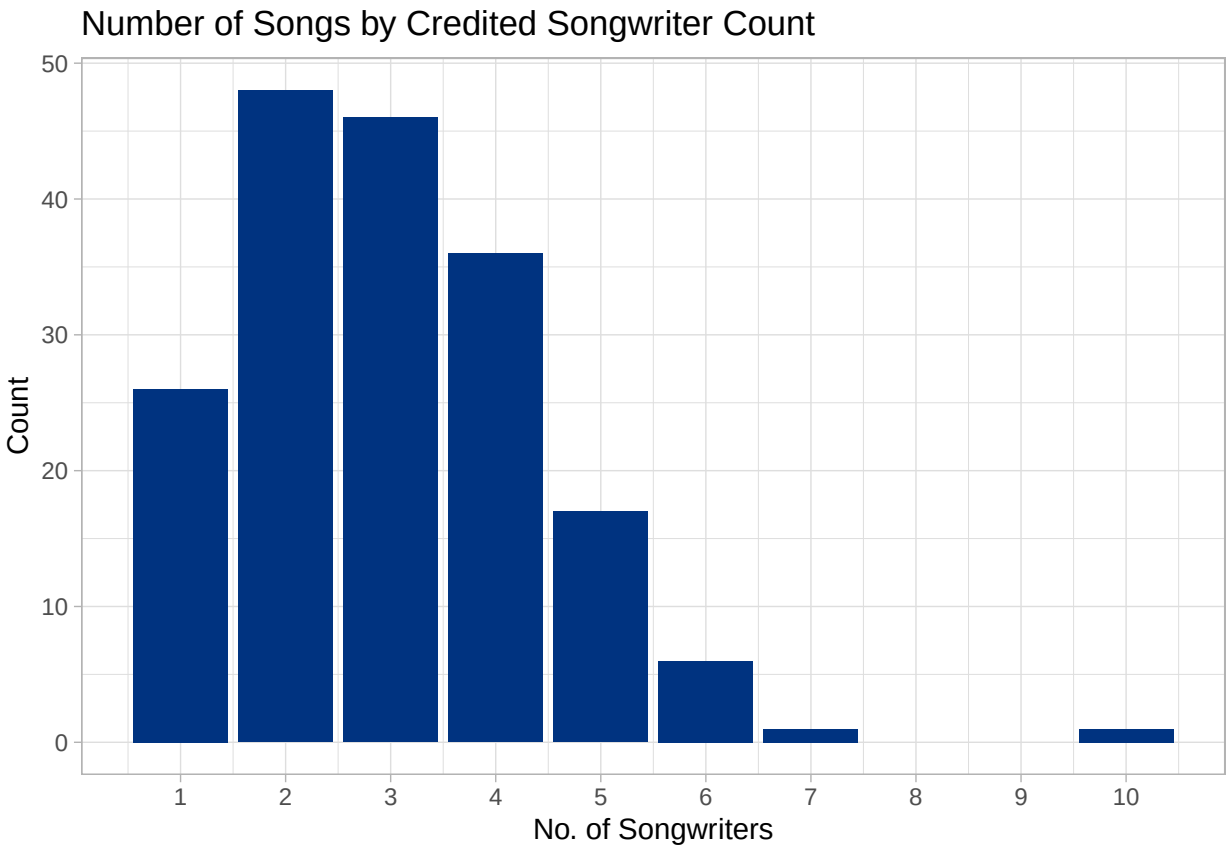
Of note is the fact that 'Adrenalina' had 10(!) credited writers, but the song failed to perform finishing 22nd.

To be able to better render the plots in the following sections we will use the ggplot library:

```
library(ggplot2)
```

We can check the general distribution of the amount of songwriters, to see how common is the "group writing" approach

```
Eurovision_results %>%
  select(Song,Songwriter_count,Place) %>%
  ggplot() +
  geom_bar(aes(x=Songwriter_count),fill=hsv(0.6,1,0.5)) +
  labs(x="No. of Songwriters",y="Count",title = "Number of Songs by Credited Songwriter Count") +
  theme_light() +
  scale_x_continuous(labels=1:10,breaks = 1:10)
```

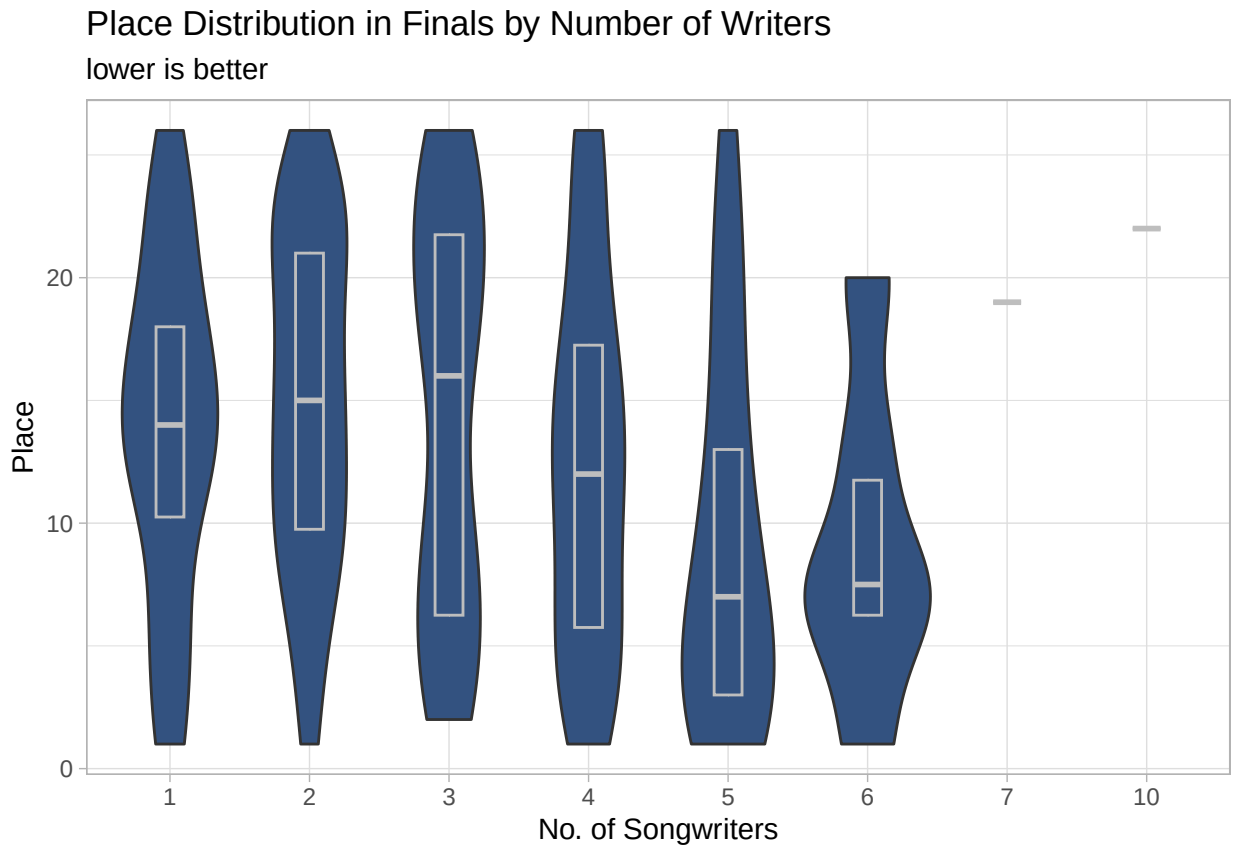


We can see that in general the most common numbers are two or three

a logical follow up question would be: do the additional writers help the song perform better, or do too many cooks spoil the broth? We can observe this information from the following plot - a nested violin and bar plot, the bar plot allows us to quickly see the median and quartiles for the data, while the violin gives an approximation of the distribution of values

```
Eurovision_results %>% select(Song,Songwriter_count,Place) %>% ggplot() + geom_violin(aes(factor(Songwr
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```



We can see that up to a point, adding more song writers does in fact match with a higher placing; but having more than 5 tends to lead to a worse placement. another thing, is that while songs with 3 writers are the most popular they tend to perform either well or poorly, with little in between. On the contrary, single writer songs tend to place more towards middle of the pack.

Language Barrier

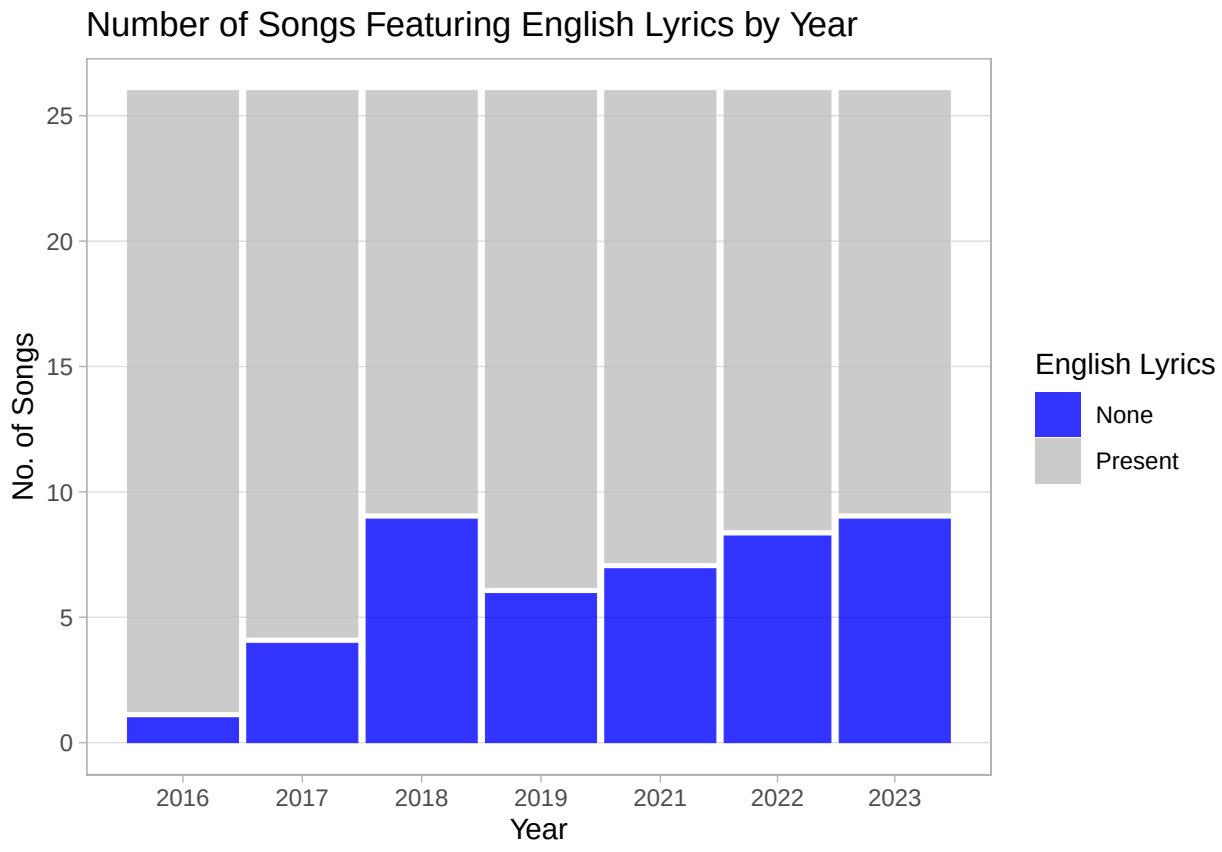
Many Eurovision viewers like the contest as it highlights the variety and diversity of cultures and nationalities of Europe, though many lament that most countries elect to send their songs in English, to make them more accessible and increase chances of winning.

We can plot the number of songs utilizing English in their lyrics throughout the years with a mosaic plot to do this we will use the ggmosaic add-on to the ggplot library

```
library(ggmosaic)

Eurovision_results %>%
  select(Song,Language,Country,Place,Year) %>%
  mutate(is_eng = str_detect(Language,"English")) %>%
  ggplot() +
  geom_mosaic(aes(x=product(is_eng,Year),fill=is_eng)) +
  labs(x="Year",y= "No. of Songs",title = "Number of Songs Featuring English Lyrics by Year",fill="Engli
  theme_light() +
  scale_y_continuous(breaks=(seq(0,26,by=5))/26,labels = seq(0,26,by=5)) +
  theme(panel.grid.minor =  element_blank(),panel.grid.major.x = element_blank()) +
  scale_fill_manual(values=c("blue","grey"),labels= c("None","Present"))
```

```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## i Please use `unite()` instead.
## i The deprecated feature was likely used in the ggmosaic package.
##   Please report the issue at <https://github.com/haleyjeppson/ggmosaic>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

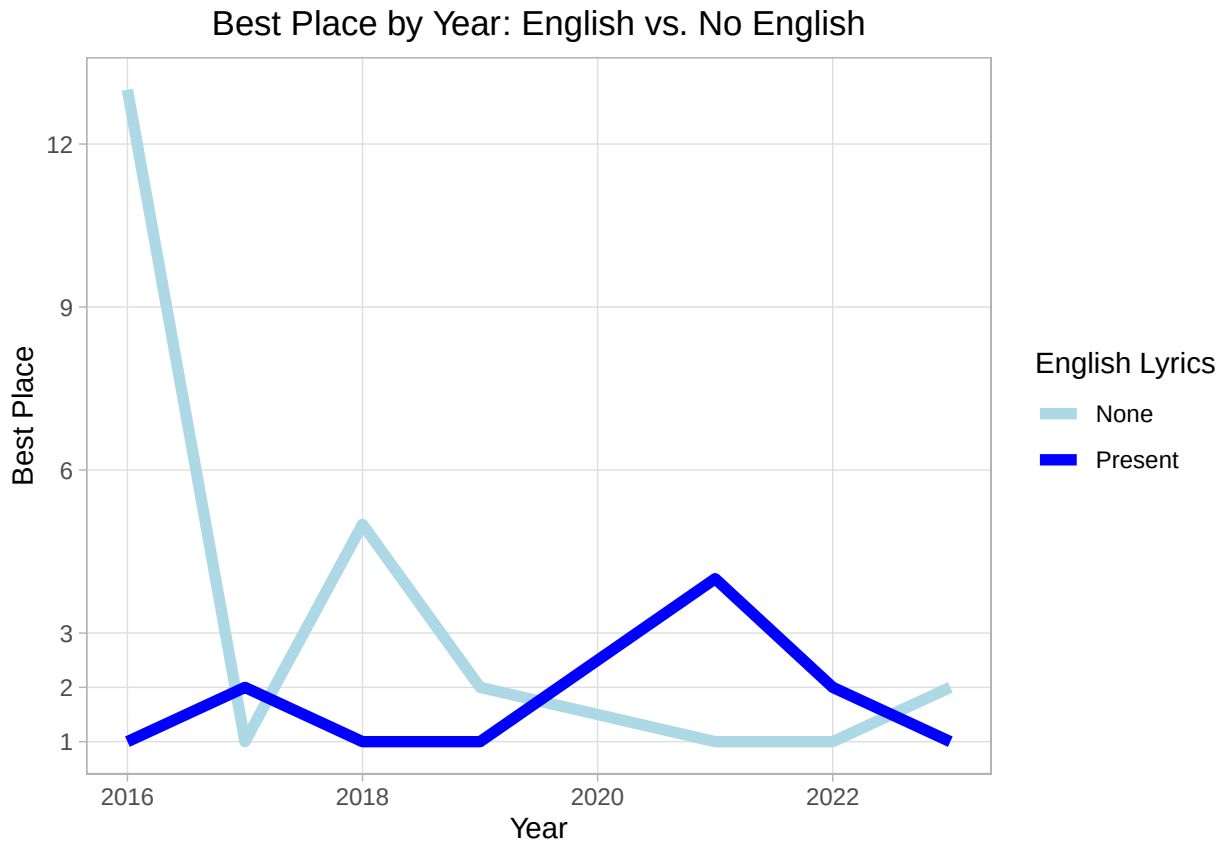


In general there is an upwards trend in songs performed in the native languages of countries, such a trend could be explained by many factors, many of them demographic and Sociopolitical. One additional such factor could be a followup reaction to good placements of previous contestants. We can check this by plotting the performance of English language songs against all other languages

First, we can compare the best placement throughout the years

```
Eurovision_results %>%
  select(Song,Language,Country,Place,Year) %>%
  mutate(is_eng = str_detect(Language,"English")) %>%
  group_by(Year,is_eng) %>%
  summarise(best_place = min(Place),avg_place = mean(Place)) %>%
  ggplot() +
  geom_line(aes(x=Year,y=best_place,group=is_eng,colour=is_eng),lwd=2,linetype=1)+
  labs(x="Year",y= "Best Place",title = "Best Place by Year: English vs. No English",color="English Lyri
  theme_light() +
  scale_y_continuous(breaks=c(1,2,seq(0,15,by=3)),labels = c(1,2,seq(0,15,by=3))) +
  theme(panel.grid.minor =  element_blank(),plot.title = element_text(hjust=0.5)) +
  scale_color_manual(values=c("Light Blue","blue"),labels= c("None","Present"))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

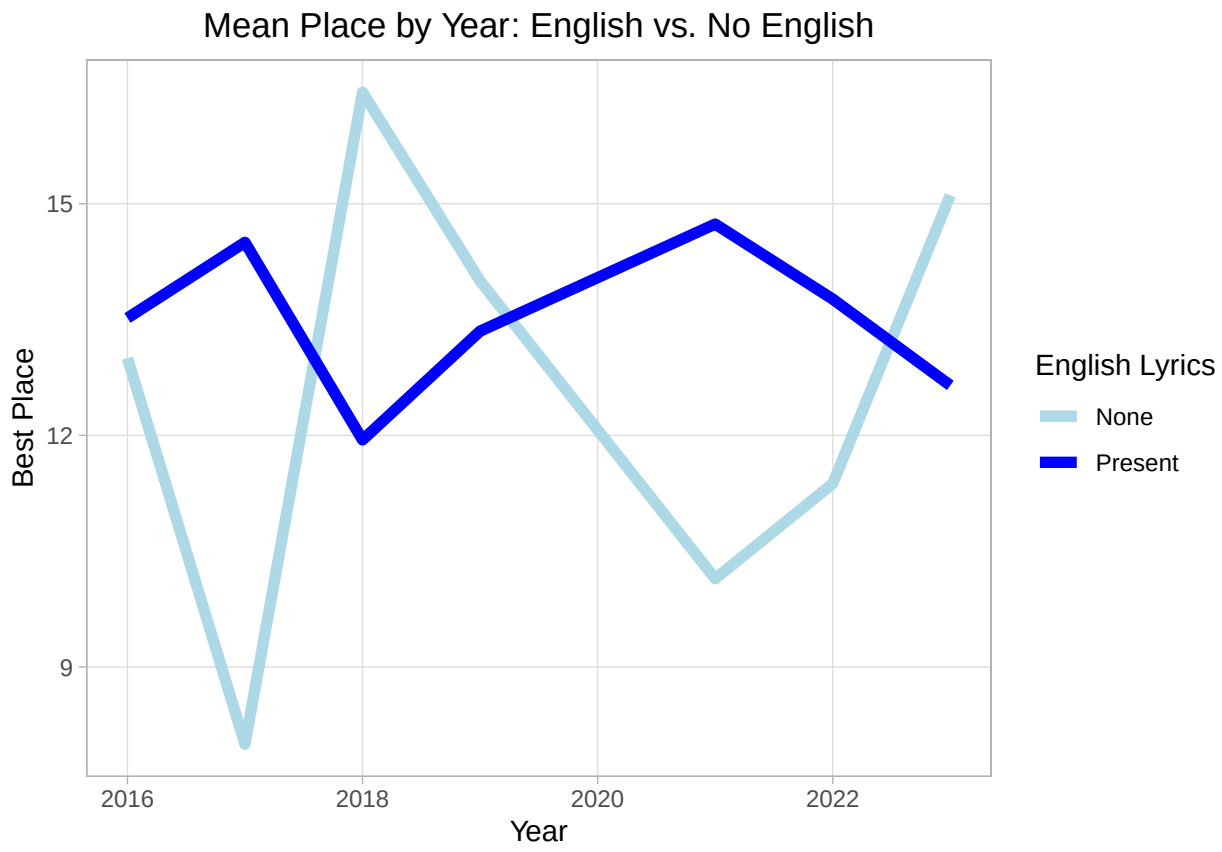


We can see a trend of non-English songs performing better in the later years, as well as 2017, possibly leading to more countries willing to send songs in their native languages. Judging the general performance based only on the best performer is somewhat problematic, so we can also plot the average placement

```
Eurovision_results %>% select(Song,Language,Country,Place,Year) %>% mutate(is_eng = str_detect(Language
  geom_line(aes(x=Year,y=avg_place,group=is_eng,colour=is_eng),lwd=2,linetype=1)+
  labs(x="Year",y= "Best Place",title = "Mean Place by Year: English vs. No English",color="English Lyri
```

```
  theme_light() +
  scale_y_continuous(breaks=c(1,2,seq(0,15,by=3)),labels = c(1,2,seq(0,15,by=3))) +
  theme(panel.grid.minor =  element_blank(),plot.title = element_text(hjust=0.5)) +
  scale_color_manual(values=c("Light Blue","blue"),labels= c("None","Present"))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```



Once again, we see as similar trend. non-English songs performed well in 2019-2022 as well as 2017, correlating with the increase in non-English songs the following years. There is also a noticeable poor performance of such songs in 2018, which matches the decline in number of songs in 2019

###The Jury Dilemma

Ever since the change in scoring format, a common point of contention was the difference in scoring between the juries and the public televote, this situation came to a boil in 2023 as many people felt Sweden's Loreen was unjustified in winning, and only won due to jury favoritism (this issue was compounded by the fact most people voted for Kaarja from Finland)

we can explore this concept of "Jury favorite" vs "fan favorite" throughout the competitions, by mapping for each year which country got the most points, their final place and the percent of maximum available points they received, sorted per year. (note - the percentage is of the maximum points a single country is eligible to receive, that being the (number of voting countries-1) X 12, as each country is barred from voting for its' reperesentative)

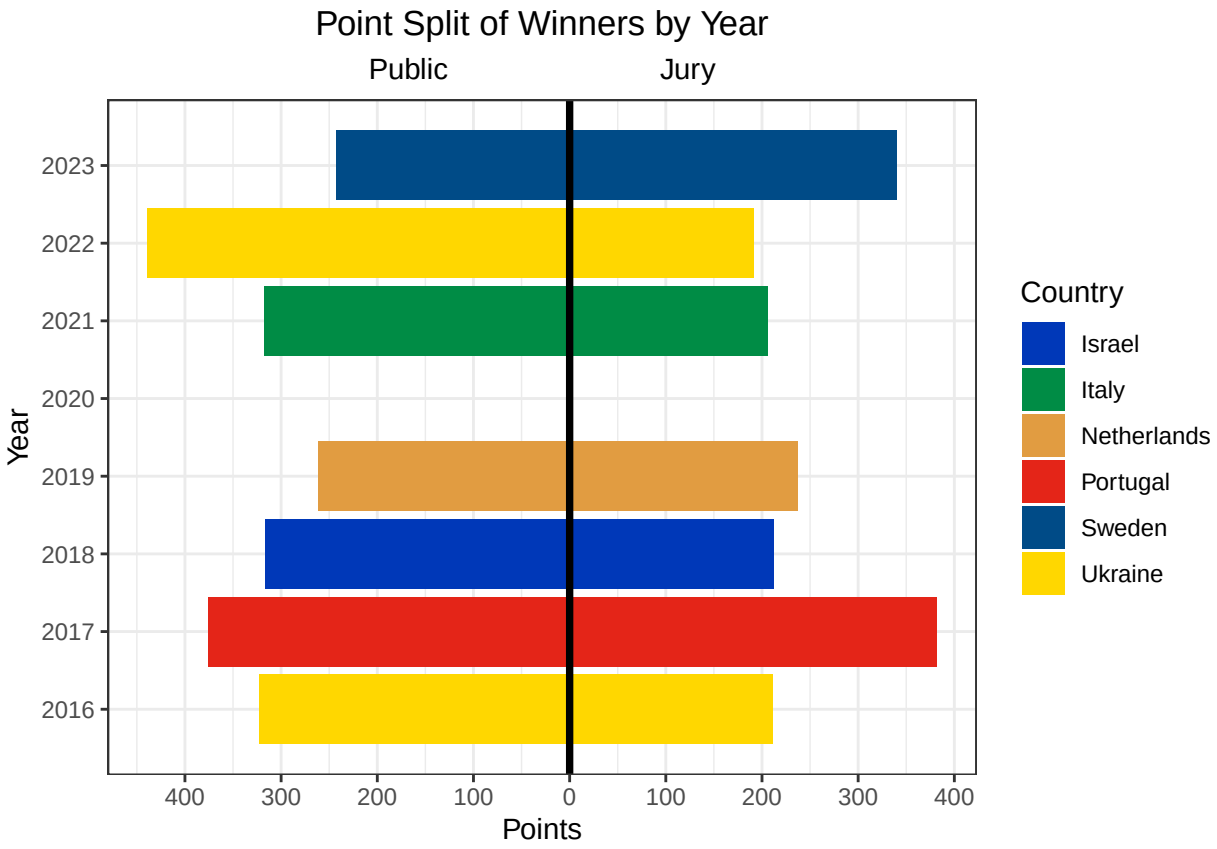
```
Eurovision_results %>%
  select(Song,Language,Country,Place,Year,Jury_Score,Televoting_score,n_vote_jury,n_vote_public) %>%
  group_by(Year) %>%
  summarise(Jury_favorite = Country[which.max(Jury_Score)],
            )
```

```
## # A tibble: 7 x 9
##    Year Jury_favorite   Jury_Points Jury_Place Jury_percent crowd_favorite
##   <int> <chr>                 <dbl>      <dbl>        <dbl> <chr>
## 1  2016 Australia               320          2        0.650 Russia
## 2  2017 Portugal                382          1        0.776 Portugal
## 3  2018 Austria                 271          3        0.538 Israel
## 4  2019 North Macedonia         247          7        0.515 Norway
## 5  2021 Switzerland             267          3        0.586 Italy
## 6  2022 United Kingdom          283          2        0.605 Ukraine
## 7  2023 Sweden                  340          1        0.787 Finland
## # i 3 more variables: crowd_Points <dbl>, crowd_place <dbl>,
## #   crowd_percent <dbl>
```

The most important thing to note here is that since 2017 up until 2023 the fan favorite outperformed the jury favorite in all of the competitions (in aggregate score), as well as the fact that in 4 of the 6 competitions before 2023 the fan favorite won the competition overall. Possibly compounding the matter is the fact that in 2023 Finland has received the second highest percentage of televote points, only behind Ukraine. This is while Sweden received the highest percentage of jury points.

We can further see this discrepancy when we compare the jury/public split of each previous winner

```
Eurovision_results %>%
  filter(Place==1) %>%
  select(Country,Points,Jury_Score,Televoting_score,Year) %>%
  ggplot() +
  geom_col(aes(x=Year,y=Jury_Score,fill=Country))+
  geom_col(aes(x=Year,y=-Televoting_score,fill=Country))+
  geom_hline(aes(yintercept=0),colour="Black",lwd=1.3)+
  labs(x="Year",y= "Points",title = "Point Split of Winners by Year",subtitle="Public
  theme_bw() +
  scale_y_continuous(breaks=seq(-400,400,by=100),labels = c(seq(400,100,by=-100),seq(0,400,by=100))) +
  scale_x_continuous(breaks = 2016:2023, labels=2016:2023) +
  theme(panel.grid.minor.y = element_blank(),plot.subtitle = element_text(hjust=0.5),plot.title = eleme
  scale_fill_manual(values=c("#0038b8","#008C45","#e19c41","#E42518","#004B87","#ffd700"))+
  coord_flip()
```



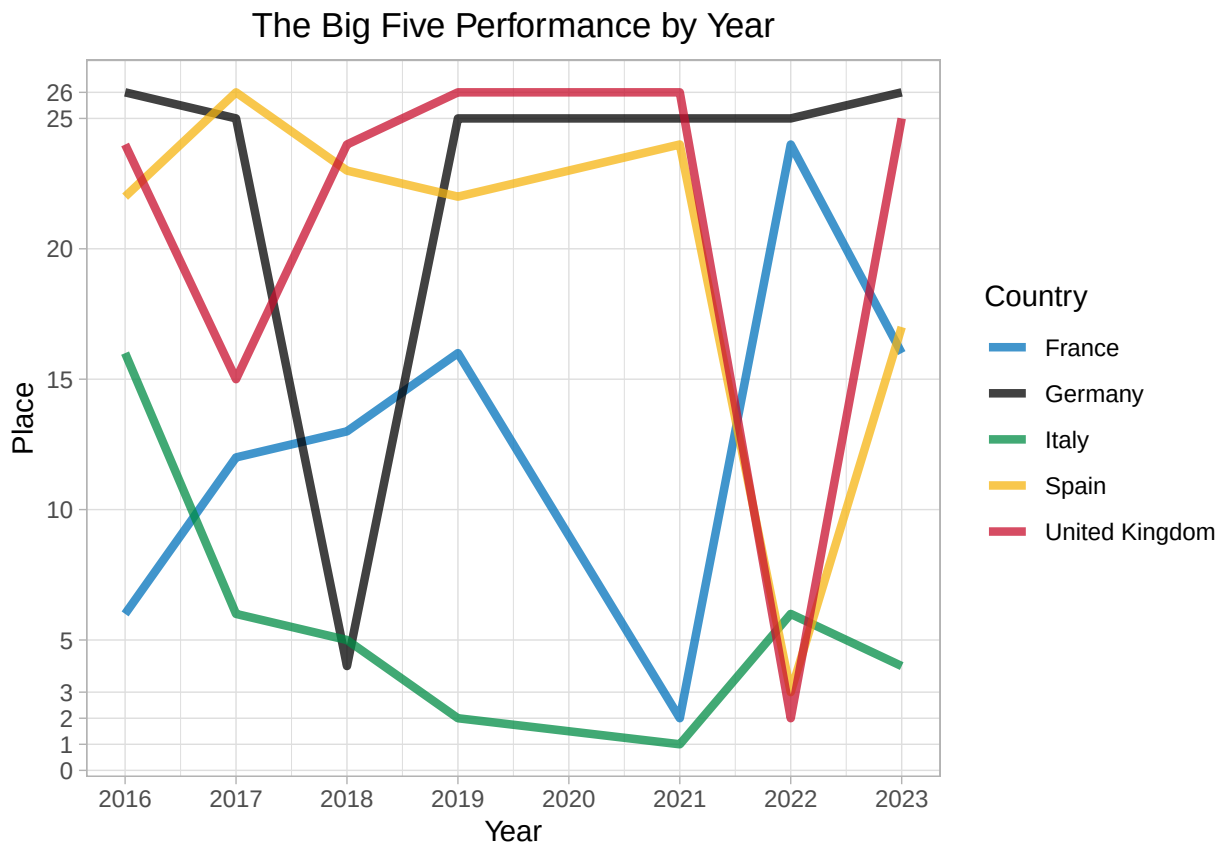
The Big Five

As gratitude for their major financial investment in Eurovision throughout history France,Germany, Italy, Spain, and the UK were given special status as the "Big Five", giving those countries direct qualification to the finals. Many pundits and Eurovision fans spoke against the establishment and continued use of the "big five" due to continued poor performance, potentially blocking more promising countries from qualifying from the semifinals

We can plot the performance of these countries over the year range to check he validity of these claims

```
big_five <-c("France","Germany","Italy","Spain","United Kingdom")

Eurovision_results %>% filter(Country %in% big_five) %>% select(Country,Place,Year) %>% ggplot() +
  geom_line(aes(x=Year,y=Place,group=Country,colour=Country),lwd=1.5,alpha=0.75) +
  labs(x="Year",y= "Place",title = "The Big Five Performance by Year",color="Country") +
  theme_light() +
  scale_y_continuous(breaks=c(1,2,3,seq(0,26,by=5),26),labels = c(1,2,3,seq(0,26,by=5),26)) +
  scale_x_continuous(breaks = 2016:2023, labels=2016:2023) +
  theme(panel.grid.minor.y = element_blank(),plot.subtitle = element_text(hjust=0.5),plot.title = eleme
  scale_color_manual(values=c("#0072bb","Black","#008C45","#f6b511","#C8102E"))
```



We can see that despite individual good performances, only Italy has consistently performed well in the competition with the UK and Germany struggling the most

Block Voting

Another hotly debated topic is the alleged political block-voting that plagues the Eurovision. Block-voting being the tendency of neighboring countries to vote for their "friend" countries regardless of the actual performances.

If we total the number of points each country gave to each other country, we can find these 'best friend' countries:

```
public_support <-Eurovision_results %>% group_by(Country) %>% summarize(across(ends_with("public"),~sum
jury_support <- Eurovision_results %>% group_by(Country) %>% summarize(across(ends_with("jury"),~sum(.x

public_support %>%
  select(-n_vote_public) %>%
  pivot_longer(cols=ends_with("public"),names_to = "other_country") %>%
  mutate(other_country = str_remove(other_country,"_public")) %>%
  mutate(Country = str_replace_all(Country," ","_")) %>%
  rename(giving_country = other_country,receiving_Country = Country,total_points=value) %>%
  filter(total_points>0) %>%
  arrange(-total_points)
```

```
## # A tibble: 1,348 x 3
##    receiving_Country giving_country total_points
```

```
##    <chr>             <chr>                <dbl>
##  1 Italy             Malta                   77
##  2 Cyprus            Greece                  72
##  3 Italy             Albania                 72
##  4 Sweden            Denmark                 69
##  5 Italy             Switzerland             65
##  6 Ukraine           Czech_Republic          65
##  7 Serbia            Slovenia                64
##  8 Serbia            Croatia                 63
##  9 Ukraine           Poland                  62
## 10 Moldova           Romania                 60
## # i 1,338 more rows
```

For the televote, we can see multiple commonly called out blocks (such as Malta to Italy, Greece to Cyprus, Romania to Moldova) dominate the top 10 positions, while some slightly less expected patterns occurs as well (Albania to Italy, Czech Republic to Ukraine)

```
jury_support %>%
  select(-n_vote_jury) %>%
  pivot_longer(cols=ends_with("jury"),names_to = "other_country") %>%
  mutate(other_country = str_remove_all(other_country,"_jury")) %>%
  mutate(Country = str_replace_all(Country," ","_")) %>%
  rename(giving_country = other_country,receiving_Country = Country,total_points=value) %>%
  filter(total_points>0) %>%
  arrange(-total_points)
```

```
## # A tibble: 1,355 x 3
##    receiving_Country giving_country total_points
##    <chr>             <chr>                <dbl>
##  1 Sweden            Finland                 73
##  2 Cyprus            Greece                  60
##  3 Italy             Malta                   59
##  4 Sweden            Czech_Republic          59
##  5 Sweden            Estonia                 56
##  6 Italy             Albania                 55
##  7 Italy             San_Marino              54
##  8 Sweden            Denmark                 53
##  9 Sweden            Norway                  52
## 10 Sweden            Germany                 51
## # i 1,345 more rows
```

For the Jury vote some of the same patterns show (Greece to Cyprus, and Albania to Italy). Though another interesting pattern appears, as Sweden is the recipient in 6 of the top 10 occasions, meaning juries tend to generally favor Sweden

Of course, counting the total points would tend to favor countries that participate more times as they wouldhave more opportunities to receive points. Instead we can look at the average number of points given each year:

```
public_support <-Eurovision_results %>%
  select(-n_vote_public)%>%
  group_by(Country) %>%
  summarize(across(ends_with("public"),~mean(.x)))
```

```
jury_support <- Eurovision_results %>%
  select(-n_vote_jury)%>% group_by(Country) %>%
  summarize(across(ends_with("jury"),~mean(.x)))

public_support  %>%
  pivot_longer(cols=ends_with("public"),names_to = "other_country") %>%
  mutate(other_country = str_remove(other_country,"_public")) %>%
  mutate(Country = str_replace_all(Country," ","_")) %>%
  rename(giving_country = other_country,receiving_Country = Country,mean_points=value) %>%
  filter(mean_points>0) %>%
  arrange(-mean_points)
```

```
## # A tibble: 1,348 x 3
##    receiving_Country giving_country mean_points
##    <chr>             <chr>                <dbl>
##  1 Cyprus            Greece                12
##  2 Latvia            Lithuania             12
##  3 Moldova           Romania               12
##  4 North_Macedonia   Serbia                12
##  5 North_Macedonia   Slovenia              12
##  6 Russia            Moldova               12
##  7 Greece            Cyprus                11.5
##  8 Hungary           Serbia                11.3
##  9 Russia            Latvia                11.3
## 10 Armenia           Georgia               11
## # i 1,338 more rows
```

In this case some of the "political" patterns start emerging more strongly, as the top 10 includes mostly neighbouring countries or countries with strong historical ties

```
jury_support %>%
  pivot_longer(cols=ends_with("jury"),names_to = "other_country") %>%
  mutate(other_country = str_remove_all(other_country,"_jury")) %>%
  mutate(Country = str_replace_all(Country," ","_")) %>%
  rename(giving_country = other_country,receiving_Country = Country,mean_points=value) %>%
  filter(mean_points>0) %>%
  arrange(-mean_points)
```

```
## # A tibble: 1,355 x 3
##    receiving_Country giving_country mean_points
##    <chr>             <chr>                <dbl>
##  1 Georgia           United_Kingdom        12
##  2 Greece            Cyprus                12
##  3 North_Macedonia   Austria               12
##  4 North_Macedonia   Switzerland           12
##  5 North_Macedonia   Moldova               12
##  6 North_Macedonia   United_Kingdom        12
##  7 North_Macedonia   Serbia                12
##  8 North_Macedonia   Albania               12
##  9 Russia            Azerbaijan            12
## 10 Sweden            Finland               10.4
## # i 1,345 more rows
```

Here, we can see some of those patterns reappear (such is the case with cyprus and Greece). The major issue with this method is that it heavily favors countries that have participated only once or twice, one way to resolve this issue could be treating each non-compete year as '0' points,but this could drag us into a pitfall of assuming the 0 points would hold even if the vote had been possible, thus interfering with the very thing we would like to check

Some possible explanations for these patterns could be: - Immigration from one country to another creating a large subset of population likely to vote for their country of origin - Similar cultures leading to similar tastes in music, this could mostly affect songs with a more ethnic sound or ones performed not English - On the jury side, Sweden is a major powerhouse in the music scene in general, and the Eurovision in particular. In fact, many countries hire Swedish writers and composers to assist in writing their contestant's songs

Cocnlusion

To summarize- while there are patterns that can be noticed in the structure of the results, for the main part the competition results are hard to predict before the contestants are revealed. While certain predictions could be reliable sight-unseen (the Cyprus-Greece relationship, or Sweden performing strongly with the jury), it is quite likely that this unpredictability, along with the rising trend in style and language variability, are both part of what makes the Eurovision song contest one of the most viewed televised events every year.