Data 607 Final Project Team Android

Matthew Roland, Jean Jimenez, Kelly Eng

2023-12-04

Clustering of European Country Data

Introduction

In this project, we explore the interesting association between European social values and Eurovision contest outcomes. By using skills learned in the semester, our focus is to cluster and visualize European survey data alongside Eurovision contest results. This approach aims to uncover potential correlations and insights into how cultural and social attitudes across various European regions might influence or reflect in the performances and results of the widely celebrated Eurovision Song Contest. By integrating and analyzing these diverse datasets, we seek to offer a unique perspective on the cultural dynamics of Europe as expressed through popular media and societal beliefs.

Importing the Data

We stored the European survey data on MongoDB because it was too large. A MongoDB database was constructed and data was imported to it. Afterwards, a connection was established to a MongoDB server, specifying the database and collection to be accessed.

We defined a specific set of columns to be retrieved. This was done by creating a vector of column names, which are fields within the MongoDB collection. These fields represent different attributes or questions from the survey data.

After defining the columns, a MongoDB projection query is constructed. This projection is a way to specify which fields in the documents should be included in the returned data. In this case, the query is set up to retrieve only the specific columns.

```r
connection_string <- "mongodb+srv://teamandroid:09sWUq5h7JMDioFk@cluster0.kk0rkpj.mongodb.net/"
collection <- mongo(collection="european_values", db="database", url=connection_string)

columns <- c("study", "wave", "uniqid", "year", "cntry_AN", "A001", "A002", "A003", "A004", "A005", "A0

#giturl <- "https://github.com/Mattr5541/DATA_607_Final_Project/raw/main/EVS_WVS_Joint_rData_v4_0.rdata

#data <- import(giturl)

projection <- paste0('{', paste0('"', columns, '": 1', collapse = ','), '}')
data <- collection$find(query = '{}', fields = projection)
```

Importing Eurovision Dataset

We imported the eurovision datasets two ways. Originally, the data was intended to be imported from a Google Sheet using the `read_sheet` function. This approach would have allowed for a direct import of data from a Google Sheet specified by its ID, making use of a package called `googlesheets4`. The specific sheet titled "All Songs Ever" from the Eurovision dataset was targeted.

However, due to difficulties in accessing the Google Sheet among the group an alternative method was adopted. The data was instead placed on GitHub.

```
##Google Drive Attmept
#eurovision_id="1jRFrSEQaLmYSFLujaaEMEE33rNUoOUMOlLGXngqaYLQ"
#eurovis_raw=read_sheet(eurovision_id, sheet="All Songs Ever")




eurogit <- "https://raw.githubusercontent.com/Mattr5541/DATA_607_Final_Project/main/Copy_of_Every_Eurovi
eurovis_raw <- import(eurogit)

eurogit <- "https://raw.githubusercontent.com/Mattr5541/DATA_607_Final_Project/main/Copy_of_Every_Eurovi
eurovis_raw <- import(eurogit)
```

Preprocessing the Data

From the European survey data, we selected many variables. These variables cover a wide range of topics, such as country abbreviations, various life values (like family, friends, work, religion), happiness and life satisfaction, religious and political affiliations, trust levels in different entities, demographic information, and socio-economic indicators.

Afterwards we filtered the data and applied regional classification. The classification divides the countries into regions like Western Europe and Southern Europe. We also dropped NA values.

```
#Selecting our variables of interest (go over this in case you want to add more variables or remove any
#Some notes: Only waves 5 & 7 are integrated into this dataset; wave 5 refers to the EVS data and wave

##Here are the variables I chose to include: cntry_AN = abbreviated names for countries; A001 = Importa

ews <- data[, !names(data) %in% "_id"]

#Belgium, Liechtenstein, Luxembourg, Monaco, Malta, San Marino were not included in the dataset; Note:
#Regions were defined in accordance with UN geoscheme classification (although I decided to add in Grea
ews <- ews %>% dplyr::filter(cntry_AN %in% c("AT", "FR", "DE", "NL", "CH", "GB", "AL", "AD", "BA", "HR"

ews <- ews %>% dplyr::mutate(region = ifelse(cntry_AN %in% c("AT", "FR", "DE", "NL", "CH", "GB"), "Weste

ews <- ews %>% dplyr::mutate(region = dplyr::if_else(cntry_AN == "US", "America", region))

##And now I can clean the data
#First, I'll start by removing all values that denote a participant's uncertainty or refusal to answer
ews <- ews %>% dplyr::mutate_all(~ ifelse(. %in% c(-1, -2, -3, -5), NA, .))
ews <- drop_na(ews)
```

Preprocessing Eurovision Data

From the eurovision data set, I created `semiclean_eurovis`. This subset focuses on specific columns: Country, Year, Language, Grand Final Points, and Grand Final Place. I filtered the data to include only the years from 2017 to 2022, focusing on Eurovision contests that correspond to the survey years.

I then process the different languages the songs are in to extract first language. I then classify them by language type.

```
#names(eurovis_raw)


semiclean_eurovis = eurovis_raw %>%
  select(Country, Year, Language, `Grand Final Points`,`Grand Final Place`) %>%
  filter(Year >= 2017, Year <= 2022)

semiclean_eurovis$`Grand Final Points`=as.numeric(unlist(semiclean_eurovis$`Grand Final Points`))
```

```
## Warning: NAs introduced by coercion
```

```
semiclean_eurovis$`Grand Final Place`=as.numeric(unlist(semiclean_eurovis$`Grand Final Place`))
```

```
## Warning: NAs introduced by coercion
```

```
names(semiclean_eurovis)=c("country","year","lang","pts","place")


#unique(semiclean_eurovis$lang)

sce1=semiclean_eurovis %>%
  mutate(first_lang=strsplit(lang, split=',')) %>%
  unnest(first_lang) %>%
  select(country, year, first_lang, pts,  place)


#turning Lang into Factor

clean_eurovis=sce1 %>%
  filter(complete.cases(.)) %>%
  mutate(language_category = case_when(
    first_lang %in%  c("Portuguese", "Italian", "French", " French","Spanish"," Italian") ~1,
    first_lang %in% c("English", "Icelandic"," Srnán Tongo"," English") ~2,
   first_lang %in%  c("Russian","Serbian", "Belarusian","Ukrainian","Slovene") ~3,
   first_lang %in%  c("Hungarian","Northern Sami"," Northern Sami") ~ 4,
    first_lang == "Albanian" ~ 5))

clean_eurovis=clean_eurovis %>%
  select(country, year, language_category, pts, place)
```

Initial Data Visualization and Exploration

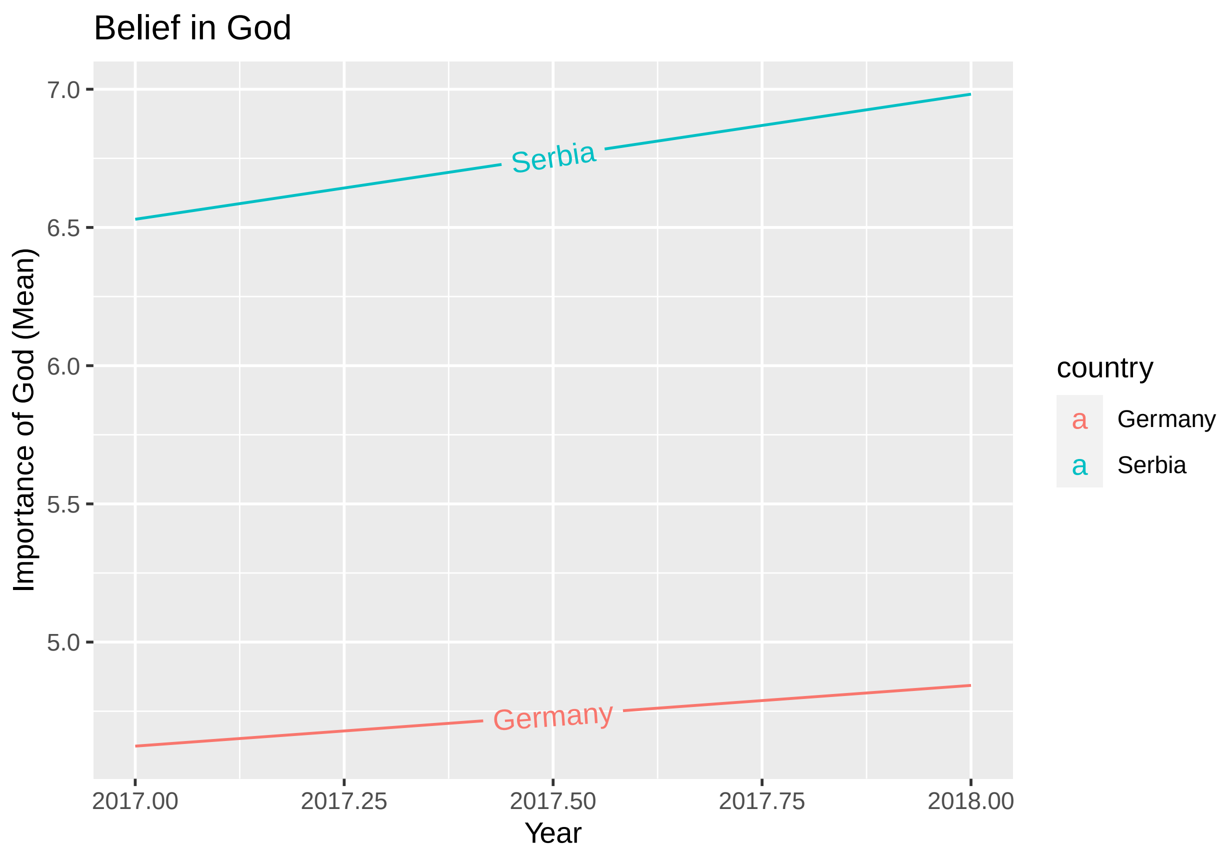
European Values Visualizations

Germany is the only country that shows up twice for 2 years for Western Europe, other countries only show up once for a year Serbia is the same case as Germany, but for Southern Europe How important God is in people's lives for these two countries increased

```
library(geomtextpath)
```

```
## Warning: package 'geomtextpath' was built under R version 4.3.2
```

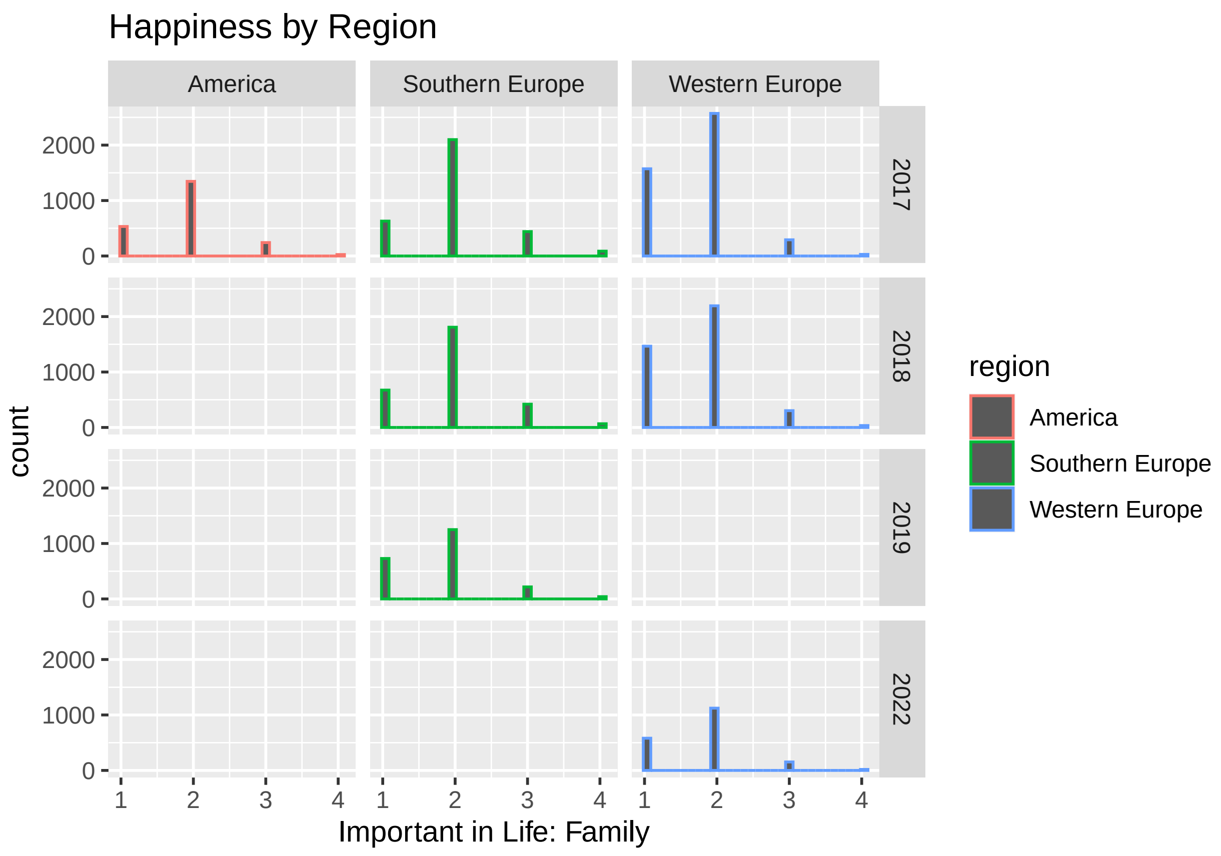
```
ews |>
  filter(cntry_AN %in% c("DE", "RS")) |>
  group_by(year, cntry_AN) |>
  mutate(mean = mean(F063), country = ifelse(cntry_AN == "DE", "Germany", "Serbia")) |>
  ggplot(aes(year, mean, col=country, label = country)) +
  geom_textpath() +
  labs(title = "Belief in God", x = "Year", y = "Importance of God (Mean)")
```



America only shows up for the year 2017, but by comparison, happiness levels is lower than both Southern & Western Europe

```
ews |>
  ggplot(aes(x=A008, col=region)) +
  geom_histogram() +
  labs(title="Happiness by Region", x="Important in Life: Family") +
  facet_grid(year~region)
```

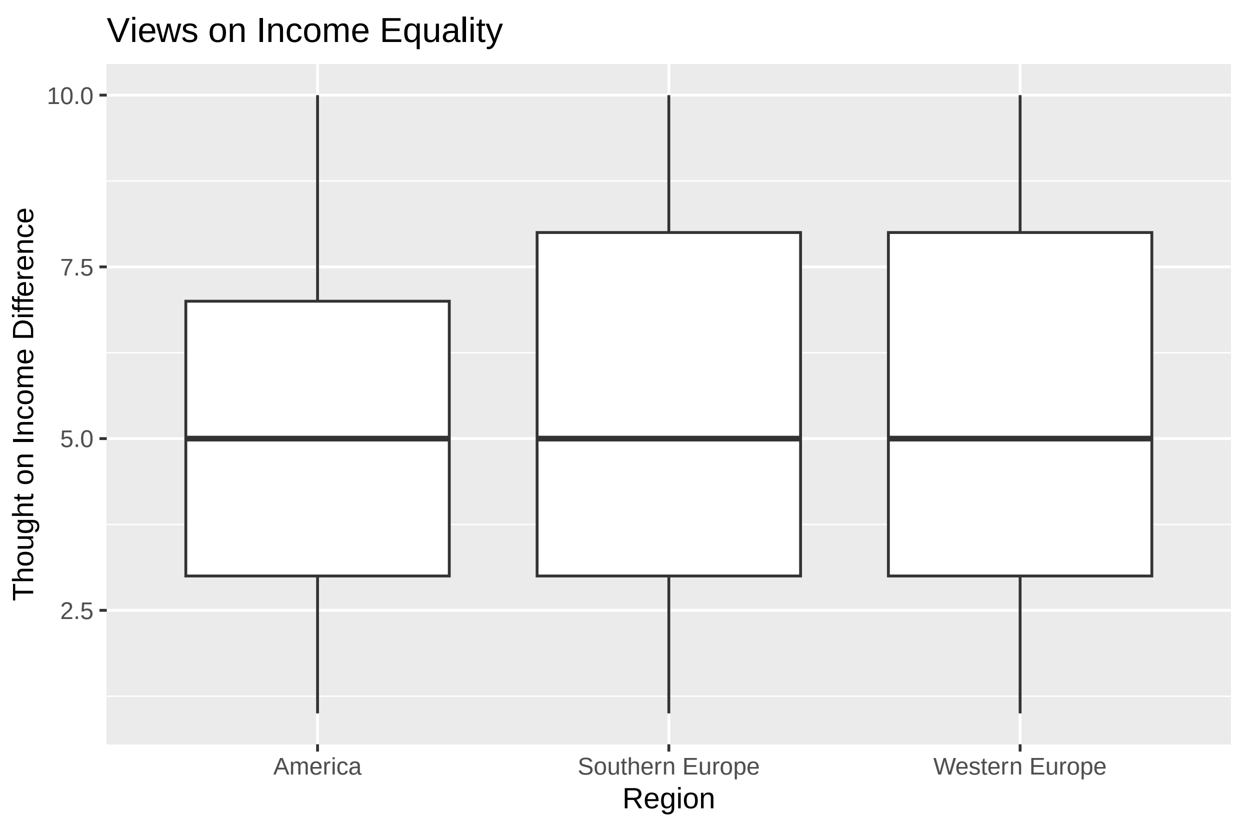
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
#X025A_01
```

Interquartile on Views on Income All 3 regions have the same median and first quartile, but it seems in Europe for the third quartile, income differences should be larger.

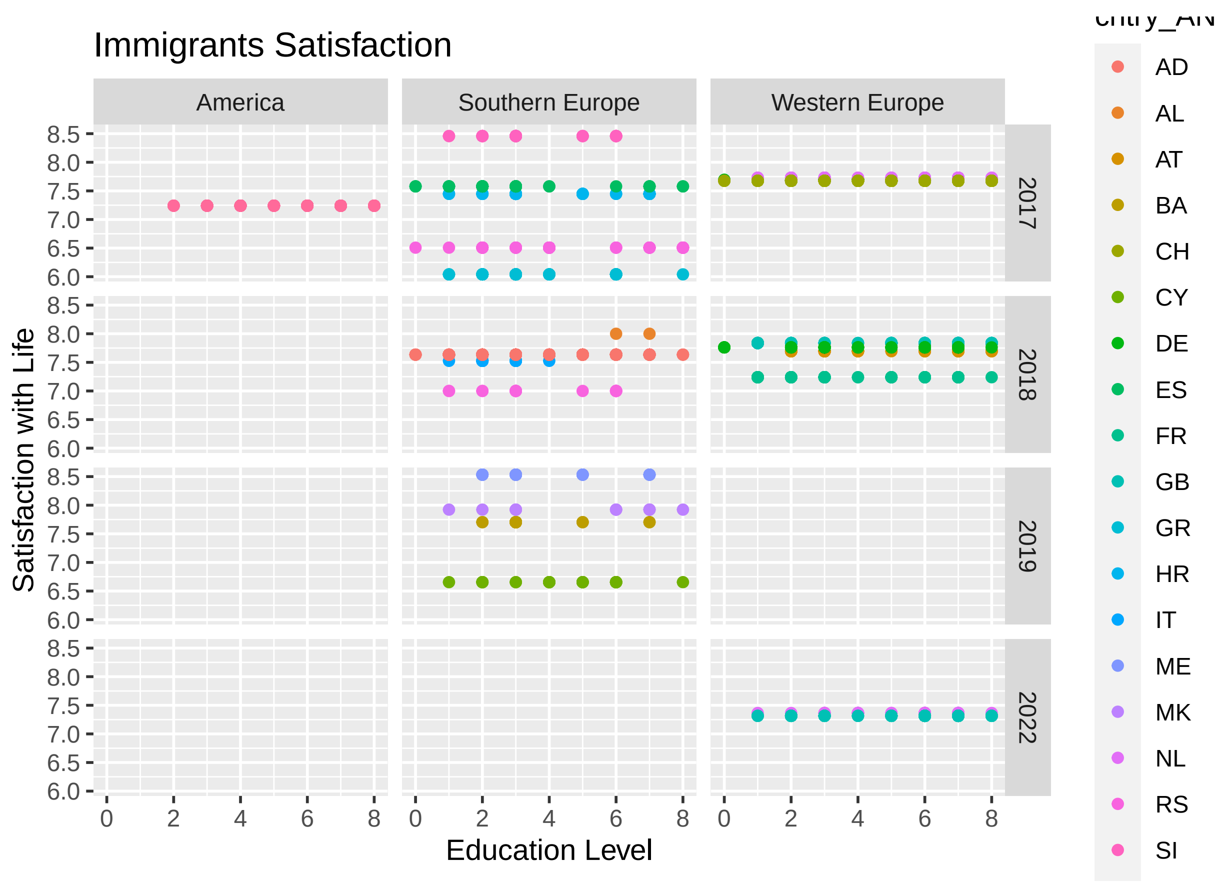
```
ews |>
  ggplot(aes(region, E035)) +
  geom_boxplot() +
  labs(title = "Views on Income Equality", x = "Region", y = "Thought on Income Difference", caption="O
```



On a scale of 1 to 10, 1 means more equal and 10 means income differences need to be larger

For immigrants, the Americans seems mostly satisfied regardless of education levels while in Europe it's more varied.

```
ews |>
  filter(G027A == 2) |>
  group_by(year, cntry_AN) |>
  mutate(satisfaction = mean(A170)) |>
  ggplot(aes(X025A_01, satisfaction, color=cntry_AN)) +
  geom_point() +
  facet_grid(year~region) +
  labs(x = "Education Level", y = "Satisfaction with Life", title = "Immigrants Satisfaction")
```



Eurovision Data

I created `most_gsb`, which targets countries that have won the top three places (first, second, or third) in each year of the contest. This is done by grouping the data by year, filtering to include only the top three places, and then re-grouping by country to count the total number of top-three finishes for each country.

After this grouping and filtering, the data is summarized to count the number of times each country has appeared in the top three places during the specified years. This summary is then arranged in descending order to highlight the countries with the most medals.

The next step involves creating a table visualization using the `flextable` package. The `flextable` function transforms the `most_gsb` data into a more visually appealing and understandable table format. The column names are made more descriptive: 'country' is relabeled as 'Nation' and 'count' as 'Number of Eurovision Medals'. This makes the table easier to interpret for readers.

Furthermore, an additional header row is added to the table to provide a clear title: "Table of Eurovision Medals by Country 2017-2022".

`ft_gsb2`, is a well-structured and informative table that visually represents the countries with the most top-three finishes in the Eurovision song contest over the specified years. This visualization is a useful tool for quickly understanding which countries have been most successful in recent Eurovisions, highlighting patterns and trends in the contest's outcomes.

```
#Visualizing Countries with most Gold, Silver, and bronze medals

most_gsb=semiclean_eurovis %>%
  group_by(year) %>%
```

```r
  filter(place==1| place ==2| place == 3) %>%
  ungroup()%>%
  group_by(country)%>%
  summarise(count = n()) %>%
  arrange(desc(count))

ft_gsb1= flextable(most_gsb)

ft_gsb2 = ft_gsb1 |>
  set_header_labels(country = "Nation", count = "Number of Eurovision Medals")  |>
  add_header_row(values = "Table of Eurovison Medals by Country 2017-2022", colwidths = 2)

ft_gsb2
```

```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

Table of Eurovison Medals by Country 2017-2022	
Nation	Number of Eurovision Medals
Italy	2
Austria	1
Bulgaria	1
Cyprus	1
France	1
Israel	1
Moldova	1
Netherlands	1
Portugal	1
Russia	1
Switzerland	1

After extracting the primary language, the data is regrouped by this first language and summarized to count the number of songs performed in each language. This summarization helps us see the frequency of each language's use in the contest. The results are arranged in descending order to highlight the most common languages.

The `flextable` package is again used to create a visually appealing table, `langs`, representing this language distribution data. The table's headers are renamed for clarity, and an explanatory header row is added, providing context to the table.

In addition to the table, a histogram is created using the `ggplot2`. This histogram, `eurolang_hist`, visualizes the same language distribution data.

```r
#Visualizing Language Distributions

eurovis_langs= semiclean_eurovis %>%
  group_by(lang) %>%
  mutate(first_lang=sapply(strsplit(lang, ", "), `[`, 1)) %>%
  ungroup()%>%
  group_by(first_lang) %>%
  summarise(count=n()) %>%
  arrange(desc(count))

total_songs=eurovis_langs %>%
  summarise(result=sum(count))

langs=flextable(eurovis_langs)|>
  set_header_labels(first_lang = "Song Performed First Language", count = "Number")  |>
  add_header_row(values = "Table of Eurovison Song First Language", colwidths = 2)

langs
```

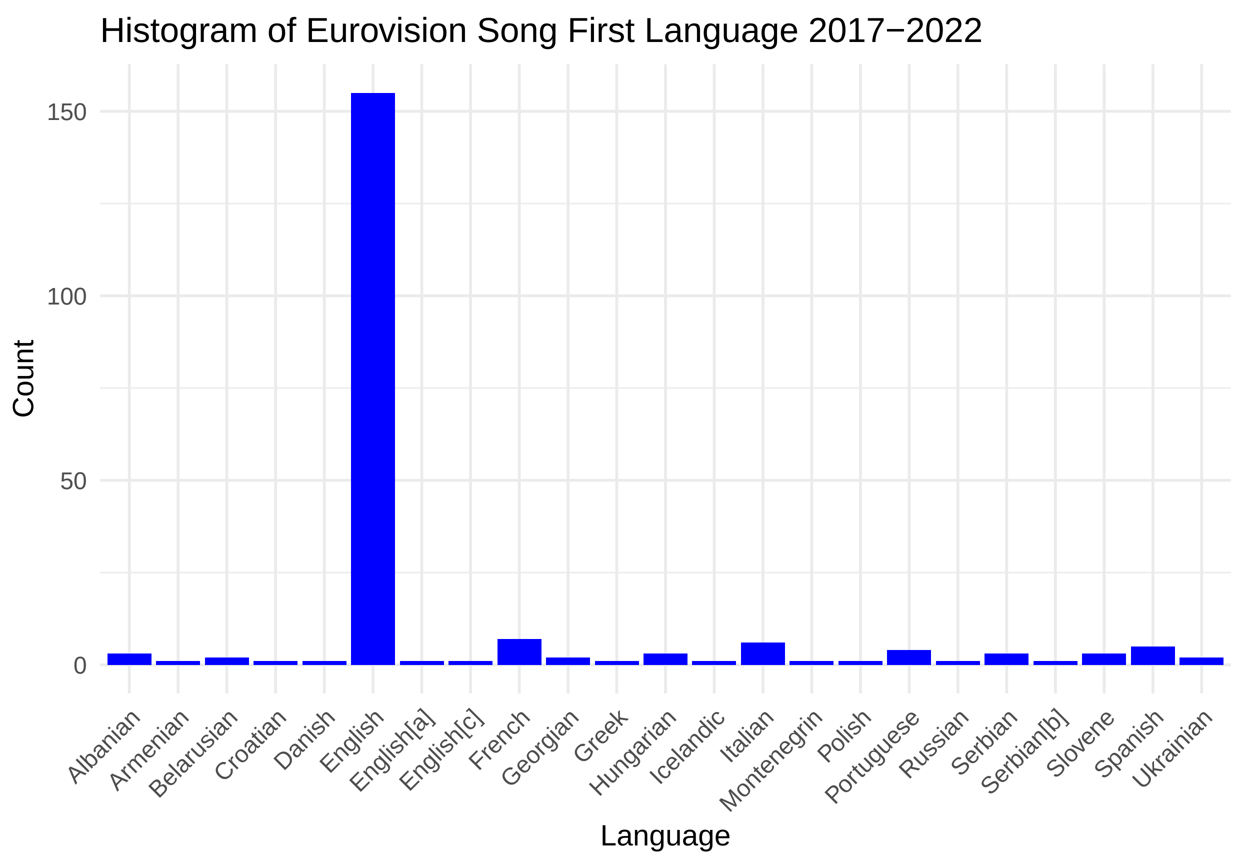
```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

Table of Eurovison Song First Language	
Song Performed First Language	Number
English	155
French	7
Italian	6
Spanish	5
Portuguese	4
Albanian	3
Hungarian	3
Serbian	3
Slovene	3
Belarusian	2
Georgian	2
Ukrainian	2

Table of Eurovison Song First Language

Song Performed First Language	Number
Armenian	1
Croatian	1
Danish	1
English[a]	1
English[c]	1
Greek	1
Icelandic	1
Montenegrin	1
Polish	1
Russian	1
Serbian[b]	1

```r
eurolang_hist=ggplot(eurovis_langs, aes(x = first_lang, y = count)) +
  geom_bar(stat = "identity", fill = "blue") +
  theme_minimal() +
  labs(x = "Language", y = "Count", title = "Histogram of Eurovision Song First Language 2017-2022") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

eurolang_hist
```



This step calculates the total points each language has garnered in top positions across the specified years.

The results are then visualized in two ways. First, a table named `lang_pt_tbl` is created using the `flextable` package. Second, a histogram is generated using the `ggplot2` package.

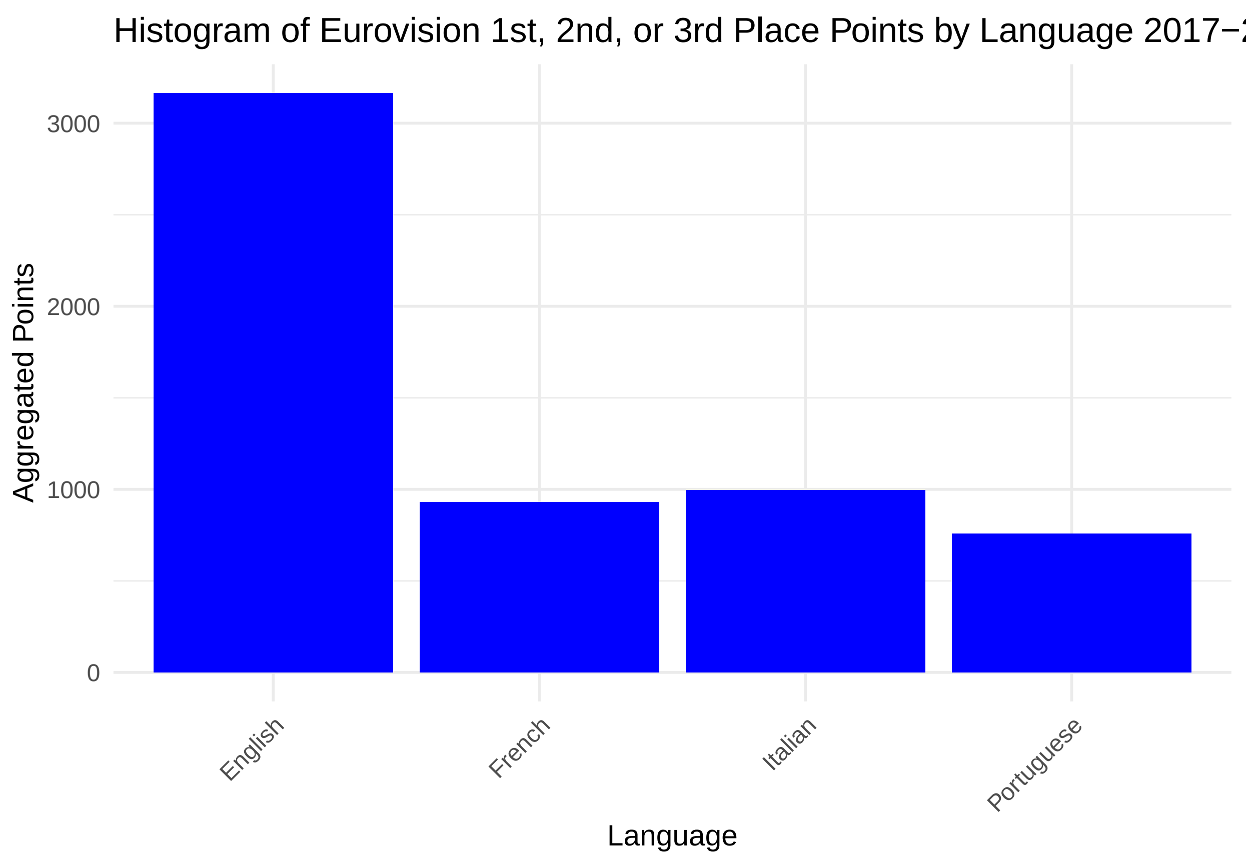
```
#Determining Winning Languages pts

lang_pts=semiclean_eurovis %>%
  group_by(year) %>%
  filter(place==1| place ==2| place == 3) %>%
  ungroup()%>%
  group_by(lang) %>%
  mutate(first_lang=sapply(strsplit(lang, ", "), `[`, 1)) %>%
  ungroup()%>%
  group_by(first_lang) %>%
  summarise(agg_pts=sum(pts, na.rm= TRUE))

lang_pt_tbl=flextable(lang_pts)|>
  set_header_labels(first_lang = "Song Performed First Language", agg_pts = "Aggregated Points")  |>
  add_header_row(values = "Table of Eurovision 1st, 2nd, or 3rd Place Points by Language 2017-2022", co

lang_pt_tbl
```

```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

Table of Eurovision 1st, 2nd, or 3rd Place Points by Language 2017-2022

Song Performed First Language	Aggregated Points
English	3,164
French	931
Italian	996
Portuguese	758

```
lang_pts_hist=ggplot(lang_pts, aes(x = first_lang, y = agg_pts)) +
  geom_bar(stat = "identity", fill = "blue") +
  theme_minimal() +
  labs(x = "Language", y = "Aggregated Points", title = "Histogram of Eurovision 1st, 2nd, or 3rd Place
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

lang_pts_hist
```



The dataset is grouped by both year and the first language, to observe the distribution of languages across different years. A summary is created for each language and year, counting the number of times each language appeared in the top three positions.

```
#Visualizing time distribution of Languages
lang_time=semiclean_eurovis %>%
  group_by(year) %>%
  filter(place==1| place ==2| place == 3) %>%
  ungroup()%>%
  group_by(lang) %>%
  mutate(first_lang=sapply(strsplit(lang, ", "), `[`, 1)) %>%
  ungroup()%>%
  group_by(year, first_lang) %>%
  summarise(count = n(), .groups = 'drop') %>%
  arrange(year, desc(count))


lang_time_tbl=flextable(lang_time)|>
  set_header_labels(year="Year", first_lang = "Song Performed First Language", count = "Count")  |>
  add_header_row(values = "Table of Eurovision 1st, 2nd, or 3rd Place Points by Language and Year 2017-

lang_time_tbl
```

```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

Table of Eurovision 1st, 2nd, or 3rd Place Points by Language and Year 2017-2022		
Year	Song Performed First Language	Count
2,017	English	2
2,017	Portuguese	1
2,018	English	3
2,019	English	2
2,019	Italian	1
2,021	French	2
2,021	Italian	1

The data is grouped by country. For each country, the total points (‘pts’) accumulated over the specified years are aggregated. This is done using the `sum` function, with the `na.rm = TRUE` parameter to ensure that missing values do not affect the calculation. T

To visualize this data, two methods are used: a table and a histogram.

```
#Visualizing Point Distributions by Country
eurovis_pts= semiclean_eurovis %>%
  group_by(country) %>%
  summarise(agg_pts=sum(pts, na.rm= TRUE)) %>%
  arrange(desc(agg_pts))


pts_tab=flextable(eurovis_pts)|>
  set_header_labels(country = "Country", agg_pts = "Aggregated Eurovision Points")  |>
  add_header_row(values = "Table of Aggregated Eurovison Points by Country 2017-2022", colwidths = 2)

pts_tab
```

```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

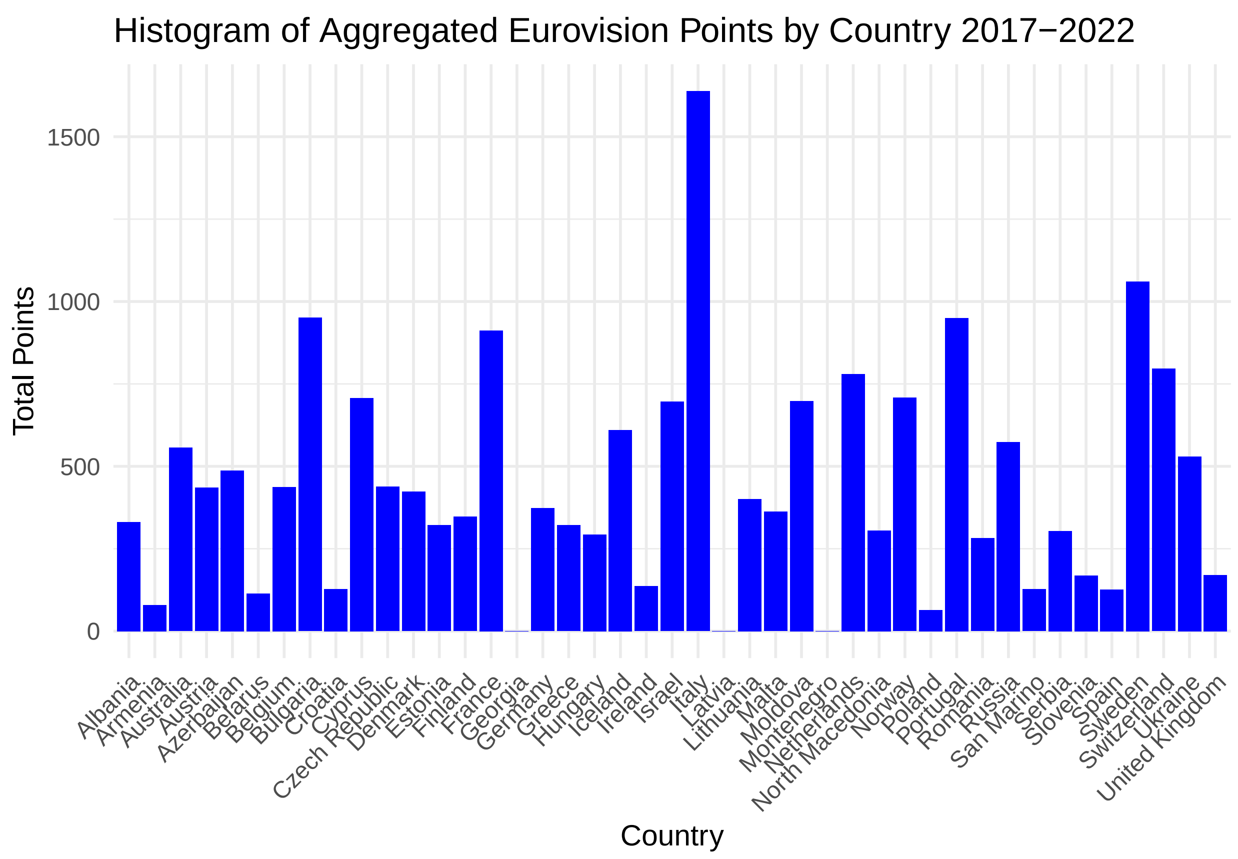
Table of Aggregated Eurovison Points by Country 2017-2022

Country	Aggregated Eurovision Points
Italy	1,638
Sweden	1,061
Bulgaria	951
Portugal	950
France	912
Switzerland	796
Netherlands	780
Norway	708
Cyprus	707
Moldova	698
Israel	696
Iceland	610
Russia	574
Australia	556
Ukraine	530
Azerbaijan	487
Czech Republic	438
Belgium	437
Austria	435
Denmark	423
Lithuania	401
Germany	373
Malta	362
Finland	347
Albania	331
Estonia	321
Greece	321
North Macedonia	305
Serbia	304
Hungary	293

Table of Aggregated Eurovison Points by Country 2017-2022

Country	Aggregated Eurovision Points
Romania	282
United Kingdom	170
Slovenia	169
Ireland	136
Croatia	128
San Marino	127
Spain	126
Belarus	114
Armenia	79
Poland	64
Georgia	0
Latvia	0
Montenegro	0

```r
europts_hist=ggplot(eurovis_pts, aes(x = country, y = agg_pts)) +
  geom_bar(stat = "identity", fill = "blue") +
  theme_minimal() +
  labs(x = "Country", y = "Total Points", title = "Histogram of Aggregated Eurovision Points by Country
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

europts_hist
```



Data Processing

Creating a Modified Data set Representing Mean Values by Country

We use `mutate` to rename a series of survey question codes (like A001, A002, etc.) to more descriptive variable names. This renaming makes the dataset more interpretable and user-friendly. For instance, 'A001' is renamed to 'Family', 'A002' to 'Friends', and so on, covering a wide range of topics like leisure, politics, work, religion, happiness, life satisfaction, and various other aspects related to personal beliefs, social trust, and environmental attitudes.

After these transformations, the `select` function is applied. This function is used to pick out only the relevant variables from the dataset.

```
#ews_group <- ews %>% group_by(year,cntry_AN ) %>% summarize(Family = mean(A001#), Friends = mean(A002)




ews_group <- ews  %>% mutate(Family =A001, Friends = A002, Leisure = A003, Politics = A004, Work = A005
   select(cntry_AN, year, Family , Friends , Leisure, Politics , Work, Religion , Happiness , Satisfacti
```

Adding Country Code to Eurovision Dataset

Here, I merge two datasets, one containing Eurovision data and another with country codes.

The purpose of this operation is to combine the Eurovision data with a standardized set of country codes, which is necessary to link it with our other data.

```
country_key=read.csv(url("https://gist.githubusercontent.com/tadast/8827699/raw/f5cac3d42d16b78348610fc4

country_key=country_key %>%
  select(Country, Alpha.2.code)

names(country_key)=c("country","cntry_AN")

eurovis_w_cc=clean_eurovis %>%
  right_join(country_key, by="country")
```

Merging Eurovision Data and Survey Data

```
# Checking column names and types in ews_group
#print(names(ews_group))
#str(ews_group)

# Checking column names and types in eurovis_w_cc
#print(names(eurovis_w_cc))
#str(eurovis_w_cc)

ews_group$year = as.integer(ews_group$year)
eurovis_w_cc$year = as.integer(eurovis_w_cc$year)


# Trim leading and trailing spaces in cntry_AN of eurovis_w_cc
eurovis_w_cc$cntry_AN <- trimws(eurovis_w_cc$cntry_AN)


joint_data = inner_join(ews_group, eurovis_w_cc, by = c("cntry_AN", "year"))
```

```
## Warning in inner_join(ews_group, eurovis_w_cc, by = c("cntry_AN", "year")): Detected an unexpected ma
## i Row 2696 of `x` matches multiple rows in `y`.
## i Row 38 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
joint_data$language_category=as.integer(joint_data$language_category)


joint_data=joint_data%>%
  ungroup()%>%
  filter(complete.cases(.)) %>%
  select(-cntry_AN)

#Alright, so character variables do not work with k-means clustering, so I have two options: 1) exclude

joint_data_mean <- joint_data %>% group_by(country) %>% summarize(Family = mean(Family), Friends = mean

joint_data_mean <- joint_data_mean %>% column_to_rownames(var = "country")
```

Exploratory Data Analysis

First, we define a function `summary_stats`. This function computes various summary statistics for a given numeric vector, including sample size, mean, standard deviation, variance, minimum and maximum values, and the interquartile range.

The function is then applied to several variables (`Friends`, `Leisure`, `Work`, and `Religion`) from the `joint_data` dataset. For each variable, the data is grouped by country, and the `summary_stats` function is applied to calculate the summary statistics. The results for each variable are stored in separate data frames (`friends_res`, `leisure_res`, `work_res`, `religion_res`) and are printed out for review.

Afterwards, we visualize dependent variables like points (`pts`) and placement (`place`) in the Eurovision contest by country. The first plot shows points by country, the second shows placement in the Eurovision by country, and the third plot compares points and placement, using a scatter plot to visualize the relationship between these variables across different countries.

Afterwards, we conduct a correlation analysis. It calculates a correlation matrix (`cor_matrix`) for numeric columns in the `joint_data` dataset. Although there is a commented-out code for visualizing the correlation matrix using `corrplot`, the analysis proceeds to identify the top 10 most strongly correlated pairs of variables, excluding self-correlations. This analysis reveals significant correlations, like a perfect correlation between 'Denomination' and 'Religious_Service_Attendance' and a strong correlation between 'Belive_in_Hell' and 'Believe_in_Heaven'.

Lastly, the code conducts a principal component analysis (PCA) on the correlation matrix. PCA is a technique used to reduce the dimensionality of the data while retaining as much variability as possible. The PCA results (`pca_result`) are summarized, and the transformed scores for the first 35 components are printed.

```r
# Custom function to get summary statistics
summary_stats <- function(x) {
  qs <- quantile(x, c(0.25, 0.5, 0.75))
  data.frame(
    sample_size = length(x),
    mean = mean(x),
    sd = sd(x),
    variance = var(x),
    minimum = min(x),
    maximum = max(x),
    interquartile_range = qs[3] - qs[1]
  )
}

friends_res <- joint_data |>
  group_by(country) |>
  summarize(across(Friends, summary_stats)) |>
  as_tibble()

leisure_res <- joint_data |>
  group_by(country) |>
  summarize(across(Leisure, summary_stats)) |>
  as_tibble()

work_res <- joint_data |>
  group_by(country) |>
  summarize(across(Work, summary_stats)) |>
  as_tibble()
```

```r
religion_res <- joint_data |>
  group_by(country) |>
  summarize(across(Religion, summary_stats)) |>
  as_tibble()

print.data.frame(religion_res)
```

```
##           country Religion.sample_size Religion.mean Religion.sd
## 1         Albania                  767      2.119948   0.9260974
## 2         Austria                  771      2.700389   1.0076267
## 3         Croatia                 1490      2.306040   0.9670341
## 4          Cyprus                  300      1.566667   0.7664946
## 5          France                 1001      2.900100   1.0440354
## 6         Germany                 1822      2.760703   0.9996337
## 7          Greece                  728      1.714286   0.9134268
## 8           Italy                  843      2.124555   0.9310695
## 9     Netherlands                 1336      2.917665   1.0059535
## 10         Serbia                  562      1.955516   0.8250619
## 11          Spain                  986      2.626775   1.1123000
## 12 United Kingdom                 1325      2.752453   1.0070764
##    Religion.variance Religion.minimum Religion.maximum
## 1          0.8576564                1                4
## 2          1.0153115                1                4
## 3          0.9351549                1                4
## 4          0.5875139                1                4
## 5          1.0900100                1                4
## 6          0.9992676                1                4
## 7          0.8343486                1                4
## 8          0.8668904                1                4
## 9          1.0119424                1                4
## 10         0.6807271                1                4
## 11         1.2372113                1                4
## 12         1.0142028                1                4
##    Religion.interquartile_range
## 1                             2
## 2                             2
## 3                             1
## 4                             1
## 5                             2
## 6                             2
## 7                             1
## 8                             2
## 9                             2
## 10                            1
## 11                            2
## 12                            2
```

```r
print.data.frame(work_res)
```

```
##           country Work.sample_size Work.mean   Work.sd Work.variance
## 1         Albania              767  1.190352 0.4486833     0.2013167
## 2         Austria              771  1.664073 0.8042055     0.6467465
## 3         Croatia             1490   1.536913 0.6297258     0.3965546
## 4          Cyprus              300   1.580000 0.8241343     0.6791973
## 5          France             1001   1.497502 0.6973118     0.4862438
## 6         Germany             1822   1.718441 0.8320068     0.6922352
## 7          Greece              728   1.359890 0.6290850     0.3957480
## 8           Italy              843   1.309609 0.5931038     0.3517722
## 9     Netherlands             1336   1.895958 0.8420317     0.7090174
## 10         Serbia              562   1.521352 0.7242228     0.5244987
## 11          Spain              986   1.320487 0.6001067     0.3601281
## 12 United Kingdom             1325   1.898113 1.0250802     1.0507895
##    Work.minimum Work.maximum Work.interquartile_range
## 1             1            4                        0
## 2             1            4                        1
## 3             1            4                        1
## 4             1            4                        1
## 5             1            4                        1
## 6             1            4                        1
## 7             1            4                        1
## 8             1            4                        1
## 9             1            4                        1
## 10            1            4                        1
## 11            1            4                        1
## 12            1            4                        1
```

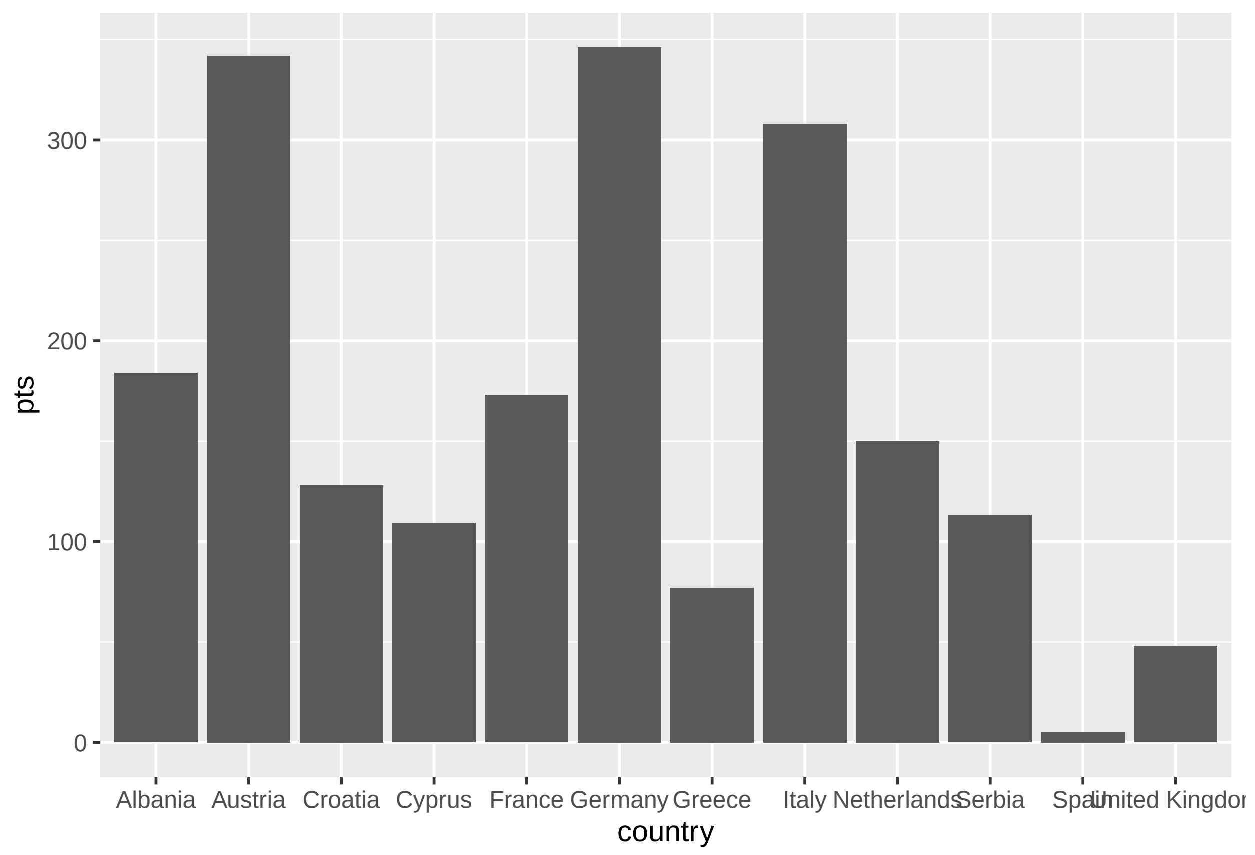
```r
print.data.frame(friends_res)
```

```
##           country Friends.sample_size Friends.mean Friends.sd Friends.variance
## 1         Albania                 767     1.753585  0.6386462        0.4078690
## 2         Austria                 771     1.409857  0.5703554        0.3253053
## 3         Croatia                1490     1.621477  0.5668908        0.3213652
## 4          Cyprus                 300     1.480000  0.6252224        0.3909030
## 5          France                1001     1.570430  0.6806502        0.4632847
## 6         Germany                1822     1.439078  0.5685988        0.3233046
## 7          Greece                 728     1.465659  0.6371591        0.4059718
## 8           Italy                 843     1.634638  0.5740515        0.3295351
## 9     Netherlands                1336     1.502246  0.5648859        0.3190961
## 10         Serbia                 562     1.483986  0.5510578        0.3036647
## 11          Spain                 986     1.421907  0.5704098        0.3253673
## 12 United Kingdom                1325     1.499623  0.5953304        0.3544183
##    Friends.minimum Friends.maximum Friends.interquartile_range
## 1                1               4                           1
## 2                1               4                           1
## 3                1               4                           1
## 4                1               4                           1
## 5                1               4                           1
## 6                1               4                           1
## 7                1               4                           1
## 8                1               4                           1
## 9                1               4                           1
## 10               1               3                           1
## 11               1               4                           1
## 12               1               4                           1
```

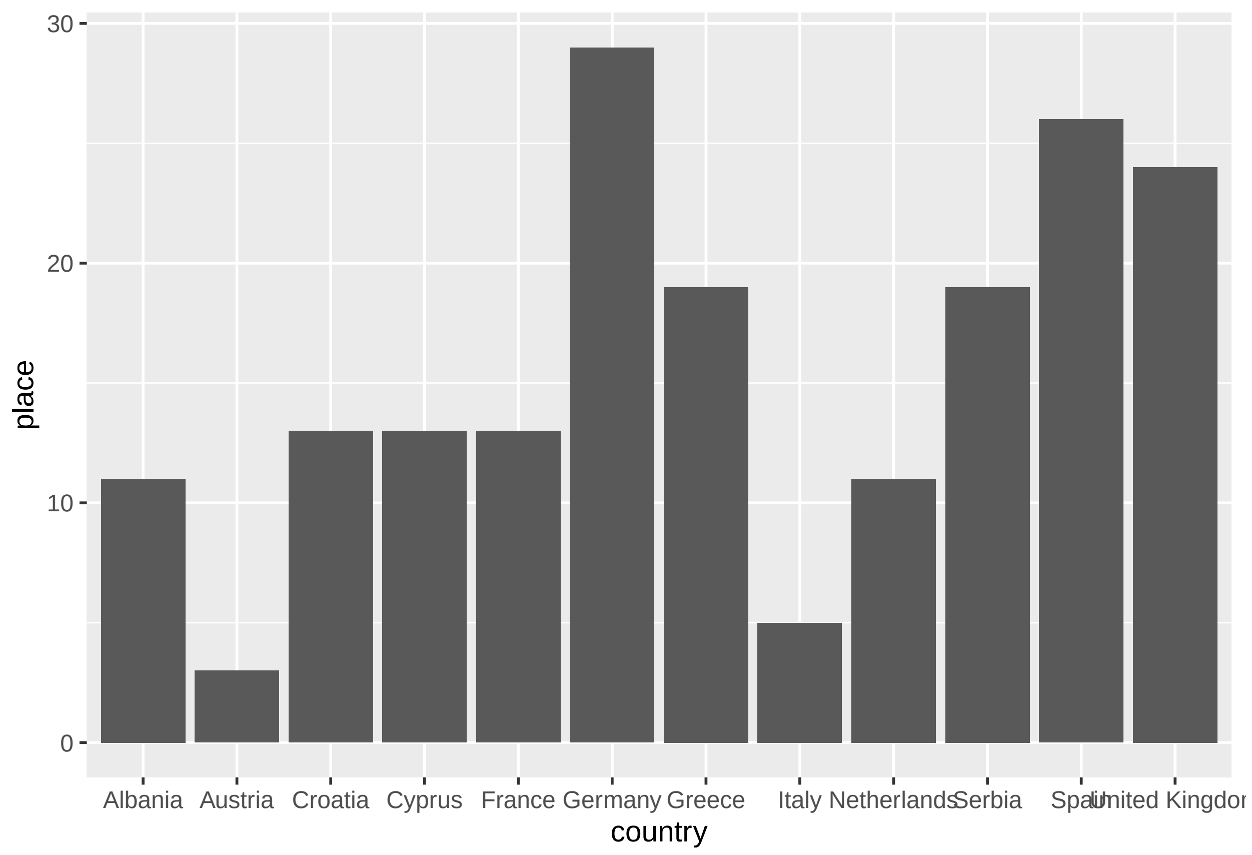
```r
print.data.frame(leisure_res)
```

```
##           country Leisure.sample_size Leisure.mean Leisure.sd Leisure.variance
## 1         Albania                 767     2.029987  0.6385263        0.4077158
## 2         Austria                 771     1.556420  0.6244953        0.3899944
## 3         Croatia                1490     1.789262  0.6109743        0.3732896
## 4          Cyprus                 300     1.623333  0.6705378        0.4496210
## 5          France                1001     1.772228  0.6663842        0.4440679
## 6         Germany                1822     1.661910  0.6009952        0.3611953
## 7          Greece                 728     1.722527  0.6797833        0.4621053
## 8           Italy                 843     1.715302  0.6122974        0.3749081
## 9     Netherlands                1336     1.508234  0.5740385        0.3295202
## 10         Serbia                 562     1.539146  0.6142090        0.3772527
## 11          Spain                 986     1.529412  0.6573652        0.4321290
## 12 United Kingdom                1325     1.607547  0.6468168        0.4183720
##    Leisure.minimum Leisure.maximum Leisure.interquartile_range
## 1                1               4                           0
## 2                1               4                           1
## 3                1               4                           1
## 4                1               4                           1
## 5                1               4                           1
## 6                1               4                           1
## 7                1               4                           1
## 8                1               4                           1
## 9                1               4                           1
## 10               1               4                           1
## 11               1               4                           1
## 12               1               4                           1
```

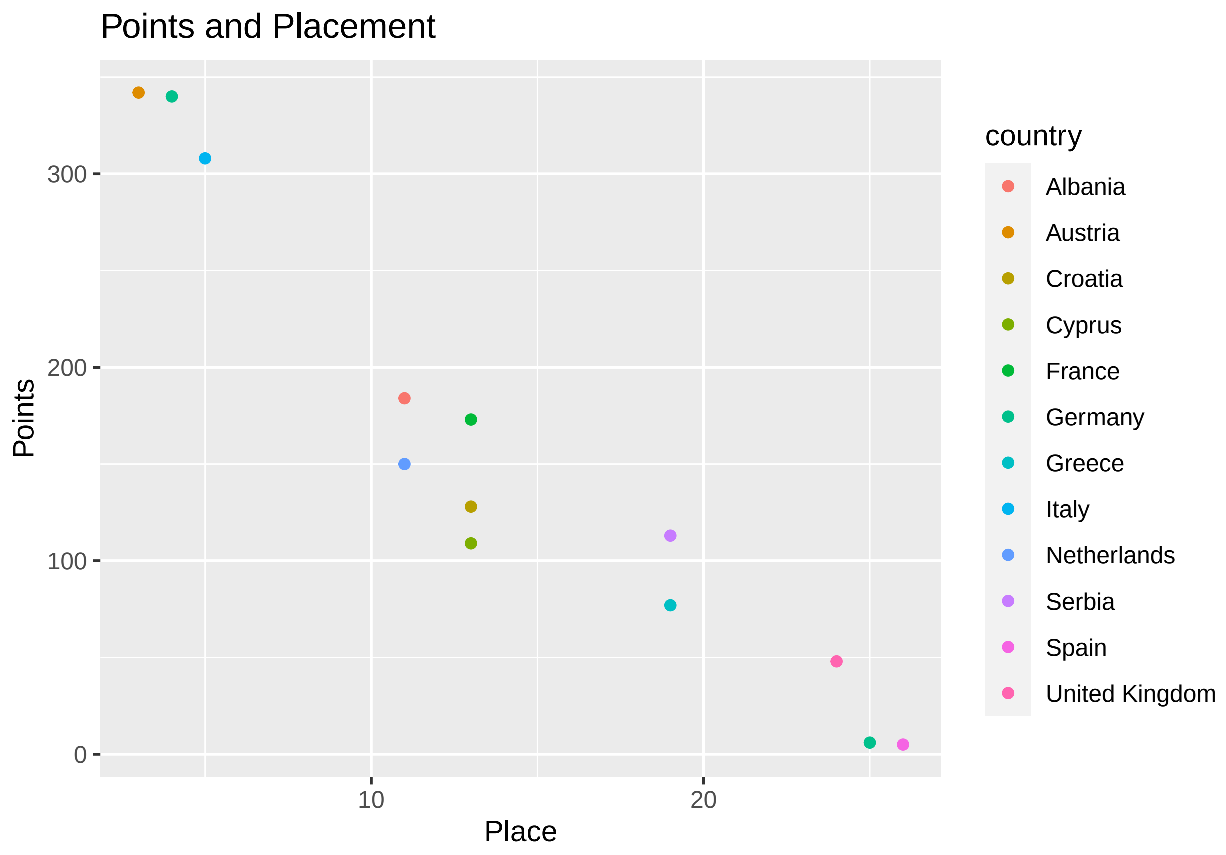
```r
# Dependent Variables

# Points by Country
joint_data |>
  distinct(country, pts, .keep_all = TRUE) |>
  ggplot(aes(country, pts)) +
  geom_col()
```



```r
# Placement in Eurovision by Country
joint_data |>
  distinct(country, place, .keep_all = TRUE) |>
  ggplot(aes(country, place)) +
  geom_col()
```



```r
joint_data |>
  distinct(country, place, pts, .keep_all = TRUE) |>
  ggplot(aes(x = place, y = pts, color = country)) +
  geom_point() +
  labs(x = "Place", y = "Points", title = "Points and Placement")
```



```r
numeric_columns <- sapply(joint_data, is.numeric)
cor_matrix <- cor(joint_data[, numeric_columns])

# The results are very messy and numbers don't fit properly that's why the below lines are commented ou
# corrplot(cor_matrix, method="number")
# corrplot(cor_matrix, method="color")

pca_result <- princomp(cor_matrix, cor=TRUE)
summary(pca_result)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     3.4109506 2.1663852 1.79269204 1.58457448 1.29504516
## Proportion of Variance 0.3061733 0.1235059 0.08457223 0.06607569 0.04413532
## Cumulative Proportion  0.3061733 0.4296792 0.51425141 0.58032710 0.62446242
##                            Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     1.20754460 1.11090338 1.04972998 1.01269937 0.94687639
## Proportion of Variance 0.03837274 0.03247648 0.02899824 0.02698842 0.02359408
## Cumulative Proportion  0.66283515 0.69531163 0.72430987 0.75129829 0.77489237
##                           Comp.11   Comp.12    Comp.13    Comp.14    Comp.15
## Standard deviation     0.93960584 0.8728406 0.85616604 0.79806630 0.77739047
## Proportion of Variance 0.02323314 0.0200487 0.01929001 0.01676078 0.01590358
## Cumulative Proportion  0.79812550 0.8181742 0.83746422 0.85422500 0.87012858
##                           Comp.16    Comp.17    Comp.18    Comp.19    Comp.20
## Standard deviation     0.74023009 0.73180542 0.66705226 0.65228834 0.63627212
## Proportion of Variance 0.01441949 0.01409314 0.01170944 0.01119684 0.01065374
## Cumulative Proportion  0.88454807 0.89864120 0.91035064 0.92154749 0.93220123
##                           Comp.21     Comp.22     Comp.23     Comp.24
## Standard deviation     0.60031454 0.572687147 0.559848032 0.542667298
## Proportion of Variance 0.00948362 0.008630804 0.008248153 0.007749679
## Cumulative Proportion  0.94168485 0.950315655 0.958563808 0.966313487
##                            Comp.25     Comp.26     Comp.27    Comp.28
## Standard deviation     0.519746691 0.485592947 0.442898406 0.42760823
## Proportion of Variance 0.007108858 0.006205277 0.005162079 0.00481181
## Cumulative Proportion  0.973422345 0.979627622 0.984789701 0.98960151
##                            Comp.29     Comp.30     Comp.31     Comp.32
## Standard deviation     0.355999442 0.298572379 0.258204595 0.240164809
## Proportion of Variance 0.003335147 0.002345933 0.001754463 0.001517872
## Cumulative Proportion  0.992936659 0.995282592 0.997037056 0.998554928
##                             Comp.33      Comp.34      Comp.35      Comp.36
## Standard deviation     0.1592253410 0.1256070427 0.1117999462 3.582856e-02
## Proportion of Variance 0.0006671766 0.0004151876 0.0003289271 3.378121e-05
## Cumulative Proportion  0.9992221041 0.9996372917 0.9999662188 1.000000e+00
##                        Comp.37 Comp.38
## Standard deviation           0       0
## Proportion of Variance       0       0
## Cumulative Proportion        1       1
```

```r
transformed_scores <- pca_result$scores[, 1:35]
print(transformed_scores)
```

```
##                                    Comp.1      Comp.2      Comp.3      Comp.4
## year                            1.0608969  0.42412453  4.02515229  1.61097110
## Family                         -2.6947787  1.38226099 -0.11435288 -2.38719843
## Friends                        -0.1356264  3.22108601 -0.43917204 -0.53436271
## Leisure                         0.8492334  2.55477748 -0.07043226  0.10831693
## Politics                        1.5536267  3.32694123 -1.06910667  1.18561231
## Work                           -2.4874032 -0.46593785  0.29316030 -1.64001547
## Religion                       -7.2522207  1.48237035  0.44264037  0.74272880
## Happiness                       0.6272919  3.86959919 -0.42197801 -1.40441082
## Satisfaction                   -1.7586534 -4.27336118  0.56996765  1.93529491
## Religious_Children              4.1560602 -1.11546355 -0.78207028 -0.84247161
## Organized_Religion              1.8060070 -3.43334956  1.00509219 -1.24072819
## Cultural_Activities            -1.7935872 -3.55127225  0.59536576 -0.55645680
## Political_Party_Affiliation    -1.0125809 -1.92039779  0.78144495 -0.72770535
## Trust_People                    2.7036990  3.89699293 -0.23035420  0.33693481
## Protect_Environment_vs_Economy  0.5880318  1.63938017 -0.15007618  0.51176638
## Trust_Family                   -1.4417021  2.45521980 -0.04203925 -2.74522241
## Trust_Religion                  2.0785161  3.85166730  0.65778368 -0.18354028
## Materialism                    -4.0438160 -2.52595023  0.69935836 -0.07182142
## Future_Change_Less_Work         2.3286882  1.26532605  0.09939557  1.67754502
## Income_Inequality              -0.5629820 -0.75280870  0.58911230  1.52772742
## Business_Ownership             -1.6033639 -0.91890396 -1.23765505 -2.05768774
## Competition                    -1.8696087 -0.49165574 -0.70803150 -3.17570640
## Confidence_Church              -6.3404112  1.46640913  0.25129428  0.72986495
## Democracy_Religion              3.2781475  0.43264836 -1.24788164  0.13364627
## Deonomnation                    4.4929774 -0.66051717 -0.99381699  2.19135157
## Religious_Service_Attendance    4.4929774 -0.66051717 -0.99381699  2.19135157
## Pray                            0.9776760  0.01952371  3.91814671 -0.29067318
## Pray_Outside_Service           -6.0823449  0.35121524 -2.80828518  0.65681921
## Religious                      -6.8500166  0.94480659  0.35651014  0.48822091
## Belivie_in_God                  6.0008684 -1.03568215 -0.47652476 -0.10019224
## Belive_in_Afterlife             3.9327757 -2.29722209 -0.47766219 -1.94501360
## Belive_in_Hell                  4.9777367 -1.16869818 -0.85177900 -1.66981008
## Believe_in_Heaven               5.1343965 -1.83147520 -0.70086932 -1.85088655
## Job_Priority_to_Immigrants     -3.6414015 -2.94110717 -0.07708842  0.23238267
## Impact_of_Immigration          -1.8769926 -3.05503105 -0.92964414  2.60113102
## language_category               1.7417031  1.06125612  0.19371163  4.18838587
## pts                             0.2813345  0.15842261  6.01200438 -0.36207749
## place                          -1.6151542 -0.70467682 -5.66750360  0.73592905
##                                     Comp.5       Comp.6        Comp.7
## year                            0.06620205  0.236335064  1.7122292019
## Family                          1.02570793 -1.407438647 -1.0008375869
## Friends                         1.29628981 -3.377231298  0.0240046557
## Leisure                         1.60073674 -3.311336497 -0.0002122818
## Politics                       -1.21572126 -1.250969754  1.3124327738
## Work                            0.55069943 -1.331850205  0.1402494232
## Religion                       -0.64021990  0.245470472  0.2561572984
## Happiness                       2.42993828  0.996286075 -0.7879166796
## Satisfaction                   -2.85910374 -1.041454989  0.7466655794
## Religious_Children             -0.59759010  0.034790291  0.1686779653
## Organized_Religion              0.48150768 -0.513129882 -1.0760330695
## Cultural_Activities             0.75213513  0.334110456 -1.8993110298
## Political_Party_Affiliation     1.31993743  1.544716937 -2.1779884442
## Trust_People                   -1.13365041  1.399211082 -0.1414322346
## Protect_Environment_vs_Economy -2.20876070  0.613964883 -0.8699432276
## Trust_Family                    0.33211836 -0.054793020 -1.1327292177
## Trust_Religion                 -0.92053790  1.853034490 -0.2466719417
## Materialism                     0.56213904  0.006477605  0.0574224703
## Future_Change_Less_Work        -1.67705023 -0.394525134 -0.9128862158
## Income_Inequality              -1.82847230 -0.334382717 -3.1676930377
## Business_Ownership              0.61524457  1.429558963  2.8888768690
## Competition                     0.03630128  1.337284013  2.0614096184
## Confidence_Church              -0.55349864  0.687462894 -0.0902965037
## Democracy_Religion             -0.79966513  0.848580762  1.0796095605
## Deonomnation                    1.98039026  0.655645592  0.1831961309
## Religious_Service_Attendance    1.98039026  0.655645592  0.1831961309
## Pray                            0.98787966  2.135707806 -0.2286923525
## Pray_Outside_Service           -1.73897780 -0.897398908  0.3326994116
## Religious                      -0.40038918  1.059118915  0.3224958043
## Belivie_in_God                 -0.08770541 -0.508564628 -0.0627213156
## Belive_in_Afterlife            -1.14908289 -0.587680939  0.0849576136
## Belive_in_Hell                 -1.09456364 -0.234825912  0.2677797963
## Believe_in_Heaven              -1.21598820 -0.518062030  0.1664918010
## Job_Priority_to_Immigrants      1.11435812 -0.541824660  0.0722669848
## Impact_of_Immigration           1.99428646 -0.706671810  1.3600290860
## language_category               1.60487213  0.268541875  0.0053300526
## pts                            -0.72726178 -0.745513989  0.9940655100
## place                           0.11710457  1.415711255 -0.6248785994
##                                     Comp.8      Comp.9      Comp.10     Comp.11
## year                            1.01105350  0.62189310 -0.005225484  0.99270230
## Family                         -0.23274593 -1.11048831  0.757090039  1.34737829
## Friends                        -0.23338746  0.59741306 -0.542371995 -1.47873276
## Leisure                          -0.14857204  1.06472789 -0.975665972 -1.92673840
## Politics                         -0.41563974 -1.23444661 -0.717007950  0.07858743
## Work                             -2.49362997 -2.06392417 -1.560264971  1.05159308
## Religion                          0.13555711 -0.29127744 -0.430733779 -0.04209463
## Happiness                         0.79560036  0.76268109  0.802505748  0.44826859
## Satisfaction                     -0.83433282 -0.51451721 -1.161419617 -0.39915301
## Religious_Children               -0.02061824  0.21338191 -0.331096933  0.12132051
## Organized_Religion               -0.90175552 -0.27804103 -0.193263441  0.22131319
## Cultural_Activities              -0.68065087  0.26846145 -0.384449982  0.04236521
## Political_Party_Affiliation      -0.61479845  3.40701431 -2.011225197 -0.14321219
## Trust_People                      0.53441203 -0.08995883 -0.752262017 -0.08548165
## Protect_Environment_vs_Economy -3.27469225  0.97274805  1.523298290  1.36259694
## Trust_Family                      1.48993510 -0.68138218  1.126272922  1.71708305
## Trust_Religion                    0.11837769 -0.69297283 -1.318412481 -0.48239349
## Materialism                       1.57410710 -0.51910934 -0.535463968  0.28918980
## Future_Change_Less_Work          -0.36365611  1.91773805  1.234173679 -0.10002984
## Income_Inequality                 1.17569064 -1.38345407  2.096667612 -2.83059844
## Business_Ownership               -1.43074690  1.18830530  0.919075967 -1.90081742
## Competition                      -0.91093328 -0.21531114  1.923928258 -1.41494105
## Confidence_Church                 0.10011999 -0.19684189 -0.278180701  0.08408680
## Democracy_Religion                1.98999842  0.30807274 -1.109214405  0.03137941
## Deonomnation                     -0.71055533 -1.09329406  0.493948510  0.24855191
## Religious_Service_Attendance     -0.71055533 -1.09329406  0.493948510  0.24855191
## Pray                             -0.15629849 -1.64855168 -0.844569555 -1.24943568
## Pray_Outside_Service              0.45040793  1.17980338  0.280704015  1.09661243
## Religious                         0.43070735  0.01592450 -0.427475275 -0.06100263
## Belivie_in_God                    0.04967788 -0.03248554  0.171348098  0.20693737
## Belive_in_Afterlife               0.87576660  0.02654801 -0.053104838  0.44489726
## Belive_in_Hell                    0.92905713  0.22358907 -0.188344915  0.28931677
## Believe_in_Heaven                 0.64847721  0.06364000 -0.080050151  0.34303450
## Job_Priority_to_Immigrants        0.69771997 -0.42560529  0.916035871 -0.08054823
## Impact_of_Immigration             1.48584985  0.69561173  0.896114257  0.67799097
## language_category                -0.94896775  0.16291718  0.537606985  0.68224155
## pts                               0.29694596  0.54365236  0.677668143  0.42776617
## place                             0.29307468 -0.66916753 -0.950583273 -0.25858602
##                                     Comp.12       Comp.13     Comp.14
## year                              0.171201183  1.7125086635  0.13514538
## Family                           -1.965602023  0.7540548346 -0.16334195
## Friends                          -0.455977856  0.2909620664 -0.97672510
## Leisure                           0.902118723 -0.1375352213 -0.86218836
## Politics                          0.062042369 -1.3768579442  0.47389109
## Work                              1.096435657  1.7527751144  2.23566708
## Religion                          0.196541151 -0.5524369837  0.09582394
## Happiness                         1.773038841 -0.7691990488  1.42160474
## Satisfaction                     -1.503719360  0.1610899005 -0.82449905
## Religious_Children                0.219213470  0.1657810795 -0.15442533
## Organized_Religion                0.017860107 -0.7788609483  0.05994509
## Cultural_Activities              -0.558286499 -1.8210139943  0.51007185
## Political_Party_Affiliation      -0.713965967  1.3520761805 -0.22273861
## Trust_People                     -0.495843761 -0.0123351285  0.16891647
## Protect_Environment_vs_Economy  2.122708579  0.7392503236 -1.85654954
## Trust_Family                     -1.869008889  0.4597464822 -1.10028516
## Trust_Religion                   -0.617499026 -0.2601877117 -0.43096827
## Materialism                       0.745777327 -0.8958857358 -0.25008606
## Future_Change_Less_Work        -0.777691567 -1.4273133898  1.03406961
## Income_Inequality               0.250583145  1.7947415110  1.15361216
## Business_Ownership             -1.373149402  0.0188063555  1.25810262
## Competition                     0.112006233  0.3742674772 -0.75852103
## Confidence_Church              -0.003372218 -0.9686279490  0.01618293
## Democracy_Religion              0.366155416  1.5745798788 -0.29932986
## Deonomnation                   -0.702533581 -0.1181016069 -0.55702232
## Religious_Service_Attendance   -0.702533581 -0.1181016069 -0.55702232
## Pray                            0.640035292 -0.4543403433 -0.70577706
## Pray_Outside_Service           -0.131607666 -0.0016300988  0.53806031
## Religious                       0.327025427 -0.1439658256 -0.09942516
## Belivie_in_God                 -0.039246402 -0.1911382702  0.20338546
## Belive_in_Afterlife             0.581516410 -0.5584724125  0.27936596
## Belive_in_Hell                  0.418231795 -0.2932217422  0.22442019
## Believe_in_Heaven               0.404449340 -0.3267107976  0.27725654
## Job_Priority_to_Immigrants      1.275557368 -0.6819327532 -1.41675362
## Impact_of_Immigration           0.430816987  0.8033162239  0.22592090
## language_category              -0.636034357  0.0002864778  0.85053170
## pts                            -0.032777745 -0.3704419255  0.31870231
## place                           0.465535080  0.3040688684 -0.24501752
##                                    Comp.15     Comp.16      Comp.17     Comp.18
## year                            1.02694609  0.20832389 -0.002928109  0.18598291
## Family                         -1.36228169  0.15785953  0.409738397 -2.15750758
## Friends                         0.12375053  0.11262910 -0.567368504  0.32205478
## Leisure                         0.36925587  0.33729939  0.397672900  0.41075011
## Politics                        0.58186478 -0.12028953 -0.231389316 -1.77016668
## Work                           -0.09035245 -0.49010453  0.869332435  1.07825088
## Religion                       -0.13719043 -0.10724660 -0.257777598 -0.30631540
## Happiness                      -0.21129745 -0.03710157 -0.417281415 -0.18682628
## Satisfaction                   -0.44132632 -0.10561451  0.037964381  0.21209198
## Religious_Children             -0.10791710  0.06349360  0.532547801 -0.33181614
## Organized_Religion              0.41249026  0.14619292 -0.512131082  0.74323001
## Cultural_Activities             2.92598135  1.46448325  0.869545133 -0.59872570
## Political_Party_Affiliation    -0.44445390 -1.30871815 -0.567302111 -0.72038300
## Trust_People                    0.09151297 -0.04853936  0.058324661  0.74655381
## Protect_Environment_vs_Economy -0.15506586  1.35961411 -0.478239868 -0.25800349
## Trust_Family                    0.50009285  0.71813382  0.306450546  1.57740086
## Trust_Religion                  0.30520726 -0.25153606  0.008243622  0.46449980
## Materialism                    -0.77433448  0.24967877 -1.702070283  0.59097341
## Future_Change_Less_Work        -1.24220968 -0.77138049  2.280252018  0.76433454
## Income_Inequality               0.22227965  0.15533029 -0.676645067 -0.25600025
## Business_Ownership             -1.07698905  2.12971541 -0.579918602  0.38769135
## Competition                     1.71287352 -2.41057153  0.414872012 -0.26259005
## Confidence_Church              -0.27245490 -0.36279904 -0.512561060  0.03759238
## Democracy_Religion              0.64421979  1.05822042  1.089708106 -0.50189780
## Deonomnation                    0.03652022 -0.54241490 -0.706071082  0.22157975
## Religious_Service_Attendance    0.03652022 -0.54241490 -0.706071082  0.22157975
## Pray                           -1.12718926  0.46187623  0.919201937 -0.40373400
## Pray_Outside_Service            0.70377390 -0.53591595 -0.794149331  0.37476232
## Religious                      -0.12165102 -0.06598738  0.223372914 -0.20217059
## Belivie_in_God                 -0.07023833 -0.01673891 -0.373174117  0.10307084
## Belive_in_Afterlife            -0.40333907 -0.28888804 -0.568847489  0.00629040
## Belive_in_Hell                 -0.37140204 -0.26534857 -0.251444704 -0.38487122
## Believe_in_Heaven              -0.36696143 -0.21089389 -0.387491992 -0.19750921
## Job_Priority_to_Immigrants     -0.96187374 -0.42042627  1.397944093  0.42156343
## Impact_of_Immigration          -0.02038564  0.39091776  0.788272954 -0.51419517
## language_category               0.14752222 -0.18454191 -0.255304528  0.03946574
## pts                             0.08532925 -0.18420628 -0.444265233 -0.07544444
## place                          -0.16722685  0.25790988  0.388988660  0.21843797
##                                     Comp.19     Comp.20     Comp.21
## year                            0.2516526727  0.28431756  0.15144248
## Family                          1.0638061433  0.32082744 -0.46962811
## Friends                         0.8681012125  1.40457937  0.08409526
## Leisure                        -0.0984556349 -1.17805213  0.18311574
## Politics                       -1.7482218225 -1.06680336  0.10311335
## Work                           -0.0895358041  0.02007308  0.56807964
## Religion                       -0.1044422610 -0.06690287  0.09989933
## Happiness                      -0.0007837731  0.02739997 -0.44831275
## Satisfaction                    0.0464931772 -0.06166470 -0.07413891
## Religious_Children              0.0562464842 -0.47247159 -1.03980349
## Organized_Religion             -0.1101664904 -0.65211892 -2.68910006
## Cultural_Activities             0.4443030438  0.70531234  0.83507909
## Political_Party_Affiliation    -1.0739224992 -0.35108360  0.41760435
## Trust_People                    0.0682337816  0.96819076 -0.28689493
## Protect_Environment_vs_Economy  0.2421338073 -0.02064270  0.24164457
## Trust_Family                   -1.2339352062 -1.11042860  0.61416371
## Trust_Religion                  0.0718103194  1.23442659 -0.75073432
## Materialism                     1.6607381772 -1.08859907  0.54366652
## Future_Change_Less_Work         0.9610008869 -0.76075627  0.33892702
## Income_Inequality              -0.4350438513 -0.07223882  0.01953620
## Business_Ownership             -0.6124542090  0.12229762  0.09039504
## Competition                     0.7469736343 -0.45566722  0.02266248
## Confidence_Church              -0.1316811263  0.06660683  0.24379390
## Democracy_Religion              0.5639921358 -0.71760775 -0.22949099
## Deonomnation                   -0.0058610566 -0.17753405  0.44534105
## Religious_Service_Attendance   -0.0058610566 -0.17753405  0.44534105
## Pray                            0.1844540028 -0.36667467  0.29509836
## Pray_Outside_Service           -0.0726882880  0.27054822 -0.44925049
## Religious                       0.1694502831 -0.12806374 -0.23063074
## Belivie_in_God                 -0.1837940991  0.11828791  0.11256608
## Belive_in_Afterlife             0.0136957117  0.49464896  0.64462801
## Belive_in_Hell                 -0.0351825871  0.50967579  0.63116039
## Believe_in_Heaven              -0.0396617957  0.45248835  0.50088702
## Job_Priority_to_Immigrants     -1.6225808445  1.44069713 -0.12888835
## Impact_of_Immigration          -0.1771314605  0.26674604 -0.40756564
## language_category               0.2567053302  0.21950859 -0.13691955
## pts                            -0.2564800166  0.07647784 -0.18504644
## place                           0.3680930786 -0.07826628 -0.10583589
##                                   Comp.22     Comp.23       Comp.24
## year                            0.005845127  0.24802237  6.219493e-01
## Family                         -0.416437437 -0.46412786  7.971595e-01
## Friends                         1.351347924  1.20952879 -8.651877e-01
## Leisure                        -1.655820476 -1.00953915  5.868632e-01
## Politics                        1.307385841 -0.24014538 -5.942843e-05
## Work                            0.255083557 -0.05362043 -2.034437e-02
## Religion                       -0.407857574  0.43830375 -1.062471e-01
## Happiness                      -0.390950256  0.66099994 -4.989565e-02
## Satisfaction                   -0.566042754  0.16129617 -2.688641e-01
## Religious_Children              0.107208937  0.36364810 -8.948284e-01
## Organized_Religion              0.242968259  0.26035625  2.040710e-01
## Cultural_Activities             0.033679815 -0.02235911  1.560827e-01
## Political_Party_Affiliation     0.531428093 -0.13757745  3.602027e-02
## Trust_People                    0.468302135 -0.80836293  6.578867e-01
## Protect_Environment_vs_Economy  0.175656861 -0.24348567 -1.411751e-01
## Trust_Family                    0.092707819 -0.04084287 -7.811029e-01
## Trust_Religion                 -0.307274994 -1.20041579 -3.458603e-01
## Materialism                     1.347876682 -1.10491319  3.547020e-01
## Future_Change_Less_Work         0.545217952  0.27788567  1.439140e-01
## Income_Inequality               0.110332293 -0.04717490  4.802153e-02
## Business_Ownership              0.077852348 -0.10738735  3.040565e-01
## Competition                     0.230372736 -0.45506770 -1.412653e-01
## Confidence_Church              -0.496736667  0.47606084 -1.868973e-01
## Democracy_Religion              0.448985622  0.75985722  9.594209e-01
## Deonomnation                   -0.340259200  0.77190764  4.714175e-01
## Religious_Service_Attendance   -0.340259200  0.77190764  4.714175e-01
## Pray                           -0.138148608  0.37869765 -6.713823e-01
## Pray_Outside_Service           -0.466009950  0.32966843  4.906652e-01
## Religious                      -0.544888018  0.62769716 -3.825068e-01
## Belivie_in_God                 -0.007971578 -0.16143657  1.913157e-01
## Belive_in_Afterlife            -0.475084175 -0.02640041 -1.547521e-01
## Belive_in_Hell                 -0.477843283 -0.01142470 -4.286249e-01
## Believe_in_Heaven              -0.443907570 -0.09933958 -2.667951e-01
## Job_Priority_to_Immigrants      0.649629349 -0.19351457  1.194014e+00
## Impact_of_Immigration           0.015651682 -0.85136961 -1.566963e+00
## language_category              -0.019221362 -0.97694240 -4.684320e-01
## pts                            -0.281309970  0.33623449  1.311753e-01
## place                          -0.221509963  0.18337553 -7.896863e-02
##                                     Comp.25     Comp.26     Comp.27
## year                            0.316950292  0.25902686  0.78543981
## Family                         -0.038558674 -0.02107459  0.02106832
## Friends                        -0.068768138  0.17348077  0.17459261
## Leisure                        -0.010121991 -0.19149053  0.03935693
## Politics                        0.016729891  0.27815517  0.09702793
## Work                            0.089880006 -0.02971067 -0.32148857
## Religion                       -0.198174901  0.21452270  0.08329326
## Happiness                      -0.051868834 -0.16073949 -0.35660613
## Satisfaction                   -0.319190622 -0.25856738 -0.22815443
## Religious_Children              1.342968761 -1.98309390 -0.17088690
## Organized_Religion             -0.600747129  0.73903793  0.46787889
## Cultural_Activities             0.088535841 -0.24037805 -0.15524124
## Political_Party_Affiliation    -0.050846425  0.08843578 -0.07430415
## Trust_People                   -1.807200263 -1.34938340 -0.02946637
## Protect_Environment_vs_Economy -0.133019467  0.13935153 -0.16733163
## Trust_Family                    0.012718651  0.03227329  0.21504349
## Trust_Religion                  1.387600825  1.04655590 -0.82835200
## Materialism                     0.485031056 -0.23111863 -0.23518166
## Future_Change_Less_Work         0.152427338  0.48951854 -0.14086963
## Income_Inequality               0.190540278 -0.04571790 -0.06323719
## Business_Ownership              0.234894542  0.04197558  0.03373426
## Competition                    -0.285090848  0.01498925  0.12812691
## Confidence_Church              -0.389717281 -0.15272168 -0.20159509
## Democracy_Religion              0.021819772  0.40501598 -0.24382566
## Deonomnation                    0.142661414 -0.05373104 -0.48220577
## Religious_Service_Attendance    0.142661414 -0.05373104 -0.48220577
## Pray                           -0.533619911  0.14428058  0.28547406
## Pray_Outside_Service            0.265053018 -0.04576509 -0.24356837
## Religious                       0.261338861 -0.15119637  0.51847062
## Belivie_in_God                 -0.505114887  0.21064064 -0.47415805
## Belive_in_Afterlife            -0.216385002  0.36096797  0.39459972
## Belive_in_Hell                  0.170997046  0.03725063  0.42629481
## Believe_in_Heaven              -0.072058582  0.23549285  0.46402689
## Job_Priority_to_Immigrants      0.554425365 -0.22236876  0.21746152
## Impact_of_Immigration          -0.859884717  0.33392706 -0.93757494
## language_category               0.409573321 -0.21239891  1.45776559
## pts                             0.006760995 -0.18731046 -0.51862105
## place                          -0.153201012  0.34559889  0.54521894
##                                    Comp.28     Comp.29       Comp.30
## year                            0.766024916  1.02193287  0.1818045247
## Family                          0.111539274  0.11495394  0.0586428466
## Friends                        -0.078412426 -0.02070433  0.0162738078
## Leisure                         0.195032326 -0.02907846  0.0238970884
## Politics                        0.227172821  0.27776317 -0.0905019315
## Work                            0.040011548 -0.19926507 -0.0337925886
## Religion                        0.049216878 -0.19964833  0.0998094054
## Happiness                      -0.900835776  0.69900328 -0.6068798585
## Satisfaction                   -1.000255734  0.64710723 -0.6599282972
## Religious_Children              0.247155645  0.11039550  0.4182817325
## Organized_Religion              0.399956562 -0.21974891 -0.0567769440
## Cultural_Activities             0.047596439  0.04192372 -0.0501290166
## Political_Party_Affiliation     0.024643294 -0.01775389 -0.0009029612
## Trust_People                    0.430255376 -0.07151167 -0.1836861091
## Protect_Environment_vs_Economy  0.067253854 -0.06338621 -0.0644841050
## Trust_Family                   -0.202311175  0.02501834 -0.0685539079
## Trust_Religion                  0.058630714  0.02328740 -0.0233636164
## Materialism                     0.028302383  0.06220739 -0.1297776737
## Future_Change_Less_Work         0.380587481  0.10850248 -0.0003548520
## Income_Inequality               0.130708831 -0.03735655 -0.0185335961
## Business_Ownership              0.128631733 -0.01075075  0.0274904012
## Competition                    -0.270244546  0.01822507 -0.0065831641
## Confidence_Church               0.046394354 -0.52893156  0.7732029762
## Democracy_Religion             -0.802715311 -0.91669428 -0.0365912507
## Deonomnation                    0.435427819 -0.15877464 -0.1374487889
## Religious_Service_Attendance    0.435427819 -0.15877464 -0.1374487889
## Pray                           -0.147528142  0.18761775  0.1968561951
## Pray_Outside_Service           -0.074996574  0.05823524  0.1282105546
## Religious                       0.530053284 -0.21351353 -0.4050468472
## Belivie_in_God                 -0.843414107  0.44347893  0.7576158576
## Belive_in_Afterlife             0.003283352  0.04294207  0.5736007392
## Belive_in_Hell                  0.412535572 -0.45390038 -0.5810848243
## Believe_in_Heaven               0.240003629 -0.23873899 -0.3049080048
## Job_Priority_to_Immigrants     -0.243924799 -0.06276380 -0.0490172244
## Impact_of_Immigration           0.457564692 -0.08853263 -0.0393463613
## language_category              -1.033074726 -0.57104513  0.0647399693
## pts                            -0.340103680 -0.29925460  0.0804434115
## place                           0.044406401  0.67753396  0.2842712024
##                                    Comp.31       Comp.32      Comp.33
## year                             0.171943056  0.3037784356  0.053319825
## Family                           0.004439022  0.0355232501 -0.022615411
## Friends                         -0.006359899 -0.0194085902 -0.019545664
## Leisure                         -0.009825593  0.0201229245 -0.011596788
## Politics                        -0.080087739  0.0350704146 -0.081404369
## Work                            -0.003955989  0.0046980868 -0.024078172
## Religion                         0.023028507 -0.2674400147  0.764262141
## Happiness                       -0.105298694  0.3132949083  0.027694374
## Satisfaction                    -0.108064467  0.3356780970  0.002073140
## Religious_Children              -0.011499753  0.0369815437  0.128958772
## Organized_Religion               0.039973082  0.1226735609  0.031973763
## Cultural_Activities             -0.005992582  0.0002717217 -0.002260176
## Political_Party_Affiliation     -0.037397400 -0.0208849078  0.007991197
## Trust_People                    -0.131027599 -0.0712979253  0.023776560
## Protect_Environment_vs_Economy -0.040476383  0.0342760451 -0.003005478
## Trust_Family                    -0.006389729 -0.0007404710  0.017317715
## Trust_Religion                  -0.072903805  0.0553337974  0.002935538
## Materialism                      0.044996678 -0.0087697590 -0.013706748
## Future_Change_Less_Work         -0.012199053  0.0356506716  0.021560315
## Income_Inequality               -0.022422357  0.0378628890 -0.006864318
## Business_Ownership              -0.015384838  0.0256822036  0.003841368
## Competition                      0.013415782  0.0483705442  0.016412344
## Confidence_Church                0.397149148  0.9136847427 -0.199740901
## Democracy_Religion              -0.115091244  0.1217619091  0.013537388
## Deonomnation                    -0.110606934 -0.0315774529  0.021904925
## Religious_Service_Attendance   -0.110606934 -0.0315774529  0.021904925
## Pray                             0.100215578 -0.3358847447 -0.006848585
## Pray_Outside_Service             0.089176359 -0.5072416173 -0.051755727
## Religious                       -0.304641684 -0.3920170958 -0.511099483
## Belivie_in_God                   0.499976167 -0.4276655821 -0.160859643
## Belive_in_Afterlife             -1.167042827  0.1269364116  0.025227913
## Belive_in_Hell                   0.577951518  0.0541346694  0.046020109
## Believe_in_Heaven                0.382456788 -0.0029955957  0.014574032
## Job_Priority_to_Immigrants       0.030242800  0.0485926088  0.007019340
## Impact_of_Immigration           -0.068792654  0.0290055083  0.008026989
## language_category               -0.047632790 -0.0276534711  0.010314992
## pts                              0.028035189 -0.4066965399 -0.107341216
## place                            0.190701275 -0.1875337233 -0.047924989
##                                      Comp.34      Comp.35
## year                             0.0281091717  0.025197552
## Family                          -0.0100992471  0.002241341
## Friends                         -0.0083532190 -0.010629342
## Leisure                         -0.0102997345 -0.004324040
## Politics                        -0.0365663481 -0.018730530
## Work                             0.0013371476  0.011279216
## Religion                         0.2143601701  0.100320683
## Happiness                        0.0147473200 -0.006015231
## Satisfaction                     0.0099633022  0.017353030
## Religious_Children               0.0362574216 -0.011310107
## Organized_Religion              -0.0050409323  0.039981493
## Cultural_Activities              0.0014700044 -0.002742558
## Political_Party_Affiliation     -0.0003920068 -0.010693623
## Trust_People                     0.0020134560 -0.002261627
## Protect_Environment_vs_Economy  0.0011114971  0.009152273
## Trust_Family                       0.0001354260  0.001234551
## Trust_Religion                     0.0179922047 -0.004517645
## Materialism                        0.0071993781  0.005806982
## Future_Change_Less_Work            0.0098522275  0.003025221
## Income_Inequality                  0.0007658728  0.003499097
## Business_Ownership                -0.0055749331  0.003453457
## Competition                        0.0047571597  0.006733967
## Confidence_Church                 -0.0133169757 -0.020629482
## Democracy_Religion                 0.0157281698 -0.011473200
## Deonomnation                       0.0020058151 -0.012560729
## Religious_Service_Attendance       0.0020058151 -0.012560729
## Pray                              -0.2344259011 -0.113514809
## Pray_Outside_Service              -0.2884388139 -0.138540187
## Religious                          0.3163031437  0.151700639
## Belivie_in_God                     0.2989034927  0.233020851
## Belive_in_Afterlife               -0.0585540891  0.058693207
## Belive_in_Hell                    -0.2682582653  0.354810161
## Believe_in_Heaven                  0.3003262051 -0.454740246
## Job_Priority_to_Immigrants         0.0178507267  0.007594477
## Impact_of_Immigration             -0.0084899764 -0.017776085
## language_category                 -0.0258532323 -0.014153838
## pts                               -0.1704557464 -0.093638776
## place                             -0.1590757067 -0.074285414
```

```r
# Don't want the column to be correlated with itself for obvious reasons
diag(cor_matrix) <- NA

# Flatten the upper triangle of the correlation matrix (excluding diagonal)
upper_triangle <- cor_matrix[upper.tri(cor_matrix, diag = FALSE)]

# Find the indices of the top 10 correlations
top_indices <- order(upper_triangle, decreasing = TRUE)[1:10]

# Extract the top 10 most strongly correlated pairs
top_correlated_pairs <- data.frame(
  variable1 = rownames(cor_matrix)[row(cor_matrix)[upper.tri(cor_matrix, diag = FALSE)][top_indices]],
  variable2 = colnames(cor_matrix)[col(cor_matrix)[upper.tri(cor_matrix, diag = FALSE)][top_indices]],
  correlation = upper_triangle[top_indices]
)

# Denomination & Religious_Service_Attendance have perfect correlation
# Belive_in_Hell & Believe_in_Heaven have the second strongest correlation at 0.7733818

print(top_correlated_pairs)
```

```
##              variable1                    variable2 correlation
## 1         Deonomnation Religious_Service_Attendance   1.0000000
## 2       Belive_in_Hell            Believe_in_Heaven   0.7733818
## 3  Belive_in_Afterlife            Believe_in_Heaven   0.6324739
## 4             Religion                    Religious   0.6251510
## 5             Religion            Confidence_Church   0.6119951
## 6  Belive_in_Afterlife               Belive_in_Hell   0.5509635
## 7       Belivie_in_God            Believe_in_Heaven   0.5120930
## 8    Confidence_Church                    Religious   0.5007874
## 9                  year                          pts    0.4747207
## 10             Religion         Pray_Outside_Service    0.4641389
```

Performing K-Means Clustering on the Standardized Dataset

Standardization:

Doing

$$X_{\text{standardized}} = \frac{X - \mu}{\sigma}$$

```
#stand_jointdf = joint_data %>%
  #mutate(across(-c(year, country), ~ (scale(.))))

stand_jointdf = joint_data_mean %>% scale()
```

First, we remove the 'country' column from the `stand_jointdf` dataset to create a modified version (`stand_jointdf_mod`).

K-means clustering is performed on `stand_jointdf` with two centers (or clusters) specified, and the process is repeated 25 times (`nstart = 25`) to ensure stable results. `k_means_results` is displayed using the `str` function, and key information like the centroids of the clusters (`centers`) and the size of each cluster (`size`) are printed.

Next, we want to figure out the optimal number of clusters (k value) for the dataset. We made a function, `wss`, to calculate the total within-cluster sum of squares (WSS) for different values of k. The WSS is a measure of the compactness of the clusters and is used to assess the quality of clustering. Lower WSS values generally indicate better clustering. The function is applied to a range of k values from 1 to 10, and the resulting WSS values are plotted against the number of clusters. This plot helps in identifying the "elbow point," which indicates the optimal k value. Based on the bend in the graph, 4 might be a more optimal choice for the number of clusters.

Then, k-means clustering is performed again on the `stand_jointdf` dataset, this time with 4 clusters. The final clustering result (`final`) is printed and visualized using `fviz_cluster`.

```
#stand_jointdf_mod <- stand_jointdf %>% select(-c(country))

#stand_jointdf_mod <- stand_jointdf

k_means_results <- kmeans(stand_jointdf, centers = 2, nstart = 25)
str(k_means_results)
```

```
## List of 9
##  $ cluster     : Named int [1:12] 2 1 2 2 1 1 2 2 1 2 ...
##   ..- attr(*, "names")= chr [1:12] "Albania" "Austria" "Croatia" "Cyprus" ...
##  $ centers     : num [1:2, 1:37] 0.678 -0.678 -0.485 0.485 -0.441 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:37] "Family" "Friends" "Leisure" "Politics" ...
##  $ totss       : num 407
##  $ withinss    : num [1:2] 96.2 158.6
##  $ tot.withinss: num 255
##  $ betweenss   : num 152
##  $ size        : int [1:2] 6 6
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

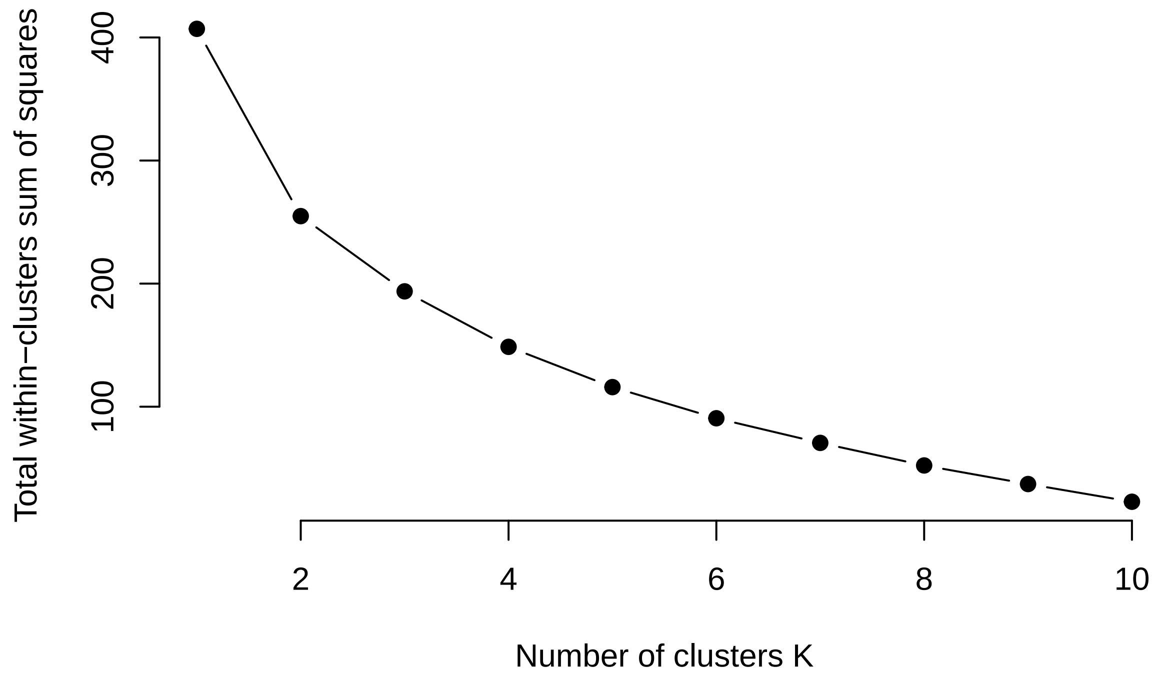
```r
print(k_means_results$centers)  # Cluster centroids
```

```
##       Family    Friends    Leisure   Politics       Work   Religion  Happiness
## 1  0.6783419 -0.4853335 -0.4411384 -0.7205388  0.5517621  0.8638409 -0.7684754
## 2 -0.6783419  0.4853335  0.4411384  0.7205388 -0.5517621 -0.8638409  0.7684754
##   Satisfaction Religious_Children Organized_Religion Cultural_Activities
## 1    0.6343732         -0.7298506          0.3474591           0.5009127
## 2   -0.6343732          0.7298506         -0.3474591          -0.5009127
##   Political_Party_Affiliation Trust_People Protect_Environment_vs_Economy
## 1                  -0.2226048   -0.8102748                     -0.2578536
## 2                   0.2226048    0.8102748                      0.2578536
##   Trust_Family Trust_Religion Materialism Future_Change_Less_Work
## 1    0.4024436     -0.6810859   0.7633451              -0.7965517
## 2   -0.4024436      0.6810859  -0.7633451               0.7965517
##   Income_Inequality Business_Ownership Competition Confidence_Church
## 1         0.2935565          0.3751764   0.4524243          0.785582
## 2        -0.2935565         -0.3751764  -0.4524243         -0.785582
##   Democracy_Religion Deonomnation Religious_Service_Attendance       Pray
## 1         -0.4662486   -0.5106538                   -0.5106538 -0.2697225
## 2          0.4662486    0.5106538                    0.5106538  0.2697225
##   Pray_Outside_Service  Religious Belivie_in_God Belive_in_Afterlife
## 1            0.4990746  0.8847768     -0.8585003          -0.3069426
## 2           -0.4990746 -0.8847768      0.8585003           0.3069426
##   Belive_in_Hell Believe_in_Heaven Job_Priority_to_Immigrants
## 1     -0.7776405        -0.6800779                  0.8632198
## 2      0.7776405         0.6800779                 -0.8632198
##   Impact_of_Immigration language_category         pts      place
## 1              0.226156        -0.3137419 -0.02569552  0.1414246
## 2             -0.226156         0.3137419  0.02569552 -0.1414246
```

```r
print(k_means_results$size)     # Size of each cluster
```

```
## [1] 6 6
```

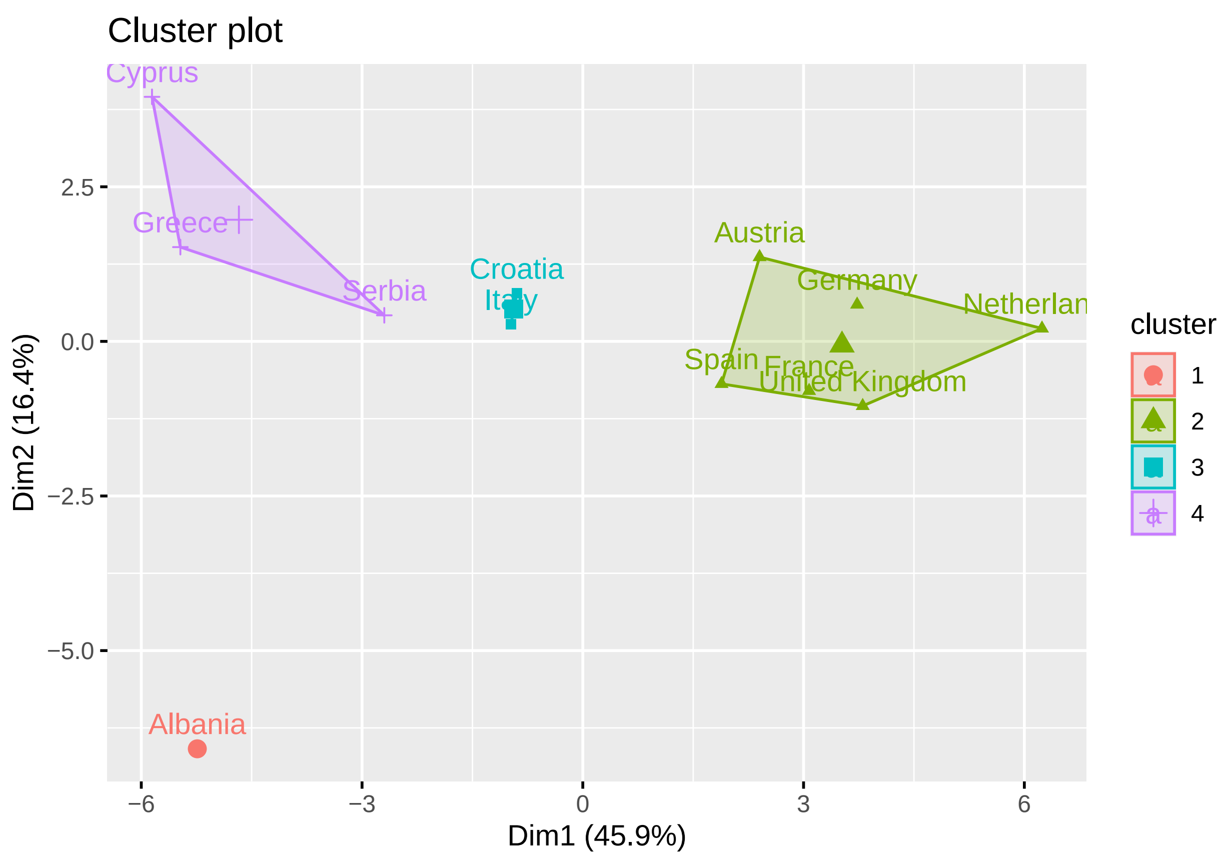
```r
#Now I will use a function to calculate the optimal k value for this analysis
set.seed(123)
wss <- function(k) {
  kmeans(stand_jointdf, k, nstart = 25)$tot.withinss
}

k_values <- 1:10

wss_values <- map_dbl(k_values, wss)

plot(k_values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")
```



```
## Based on the "bend" in the graph, 4 may be a more optimal choice for our k

final <- kmeans(stand_jointdf, centers = 4, nstart = 25)
print(final)
```

```
## K-means clustering with 4 clusters of sizes 1, 6, 2, 3
##
## Cluster means:
##       Family    Friends    Leisure   Politics       Work   Religion  Happiness
## 1 -1.4331419  2.2471822  2.4226562  1.9701819 -1.5517114 -0.5330377  1.4641209
## 2  0.6783419 -0.4853335 -0.4411384 -0.7205388  0.5517621  0.8638409 -0.7684754
## 3 -0.1005815  1.0209682  0.5470994  0.3301241 -0.5211973 -0.3301244  0.4463967
## 4 -0.8119154 -0.4590392 -0.2900081  0.5642675 -0.2388223 -1.3299197  0.7513127
##   Satisfaction Religious_Children Organized_Religion Cultural_Activities
## 1   -0.4454065         -0.1261935         -1.1847809          -1.3705191
## 2    0.6343732         -0.7298506          0.3474591           0.5009127
## 3    0.1907027          0.3104482         -0.2150797          -0.4888034
## 4   -1.2474126          1.2948002         -0.1566049          -0.2191168
##   Political_Party_Affiliation Trust_People Protect_Environment_vs_Economy
## 1                  -0.9652418    1.3437286                      0.7305360
## 2                  -0.2226048   -0.8102748                     -0.2578536
## 3                  -0.4102023    0.2974441                     -0.9122281
## 4                   1.0404250    0.9743440                      0.8803473
##   Trust_Family Trust_Religion Materialism Future_Change_Less_Work
## 1   -1.4251807      0.4573337 -0.87312368               2.1725935
## 2    0.4024436     -0.6810859  0.76334510              -0.7965517
## 3    0.5241569     -0.2478594  0.05991546              0.4108790
## 4   -0.6792650      1.3749668 -1.27559262              0.5949862
##   Income_Inequality Business_Ownership Competition Confidence_Church
## 1        1.29076059         -1.3221952 -2.08048262       -0.39677438
## 2        0.29355649          0.3751764  0.45242430        0.78558201
## 3       -0.04521454          0.3578836 -0.24766704       -0.03554635
## 4       -0.98722350         -0.5482101 -0.04624304       -1.41520833
##   Democracy_Religion Deonomnation Religious_Service_Attendance       Pray
## 1          0.2808224    2.1490414                    2.1490414 -0.5416790
## 2         -0.4662486   -0.5106538                   -0.5106538 -0.2697225
## 3          0.3546402   -0.7701735                   -0.7701735 -0.5416790
## 4          0.6024629    0.8184094                    0.8184094  1.0811241
##   Pray_Outside_Service  Religious Belivie_in_God Belive_in_Afterlife
## 1           -0.1164480 -0.8528834      1.2609504          -1.8470706
## 2            0.4990746  0.8847768     -0.8585003          -0.3069426
## 3            0.2685401 -0.7509570      0.5256994           1.0779286
## 4           -1.1383599 -0.9846212      0.9462175           0.5109563
##   Belive_in_Hell Believe_in_Heaven Job_Priority_to_Immigrants
## 1     -0.4241039        -0.8946086                 -1.1406432
## 2     -0.7776405        -0.6800779                  0.8632198
## 3      0.9798777         1.1191024                 -0.5269143
## 4      1.0433972         0.9122904                 -0.9949491
##   Impact_of_Immigration language_category         pts      place
## 1             2.1582902         2.7452420  0.34243312 -0.4820080
## 2             0.2261560        -0.3137419 -0.02569552  0.1414246
## 3            -0.5157146        -0.7843548  0.69170053 -0.7739366
## 4            -0.8279324         0.2353065 -0.52388702  0.3937778
##
## Clustering vector:
##        Albania        Austria        Croatia         Cyprus         France
##              1              2              3              4              2
##        Germany         Greece          Italy    Netherlands         Serbia
##              2              4              3              2              4
##          Spain United Kingdom
##              2              2
##
## Within cluster sum of squares by cluster:
## [1]  0.00000 96.21666 11.99057 40.41851
##  (between_SS / total_SS =  63.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
fviz_cluster(final, data = stand_jointdf)
```



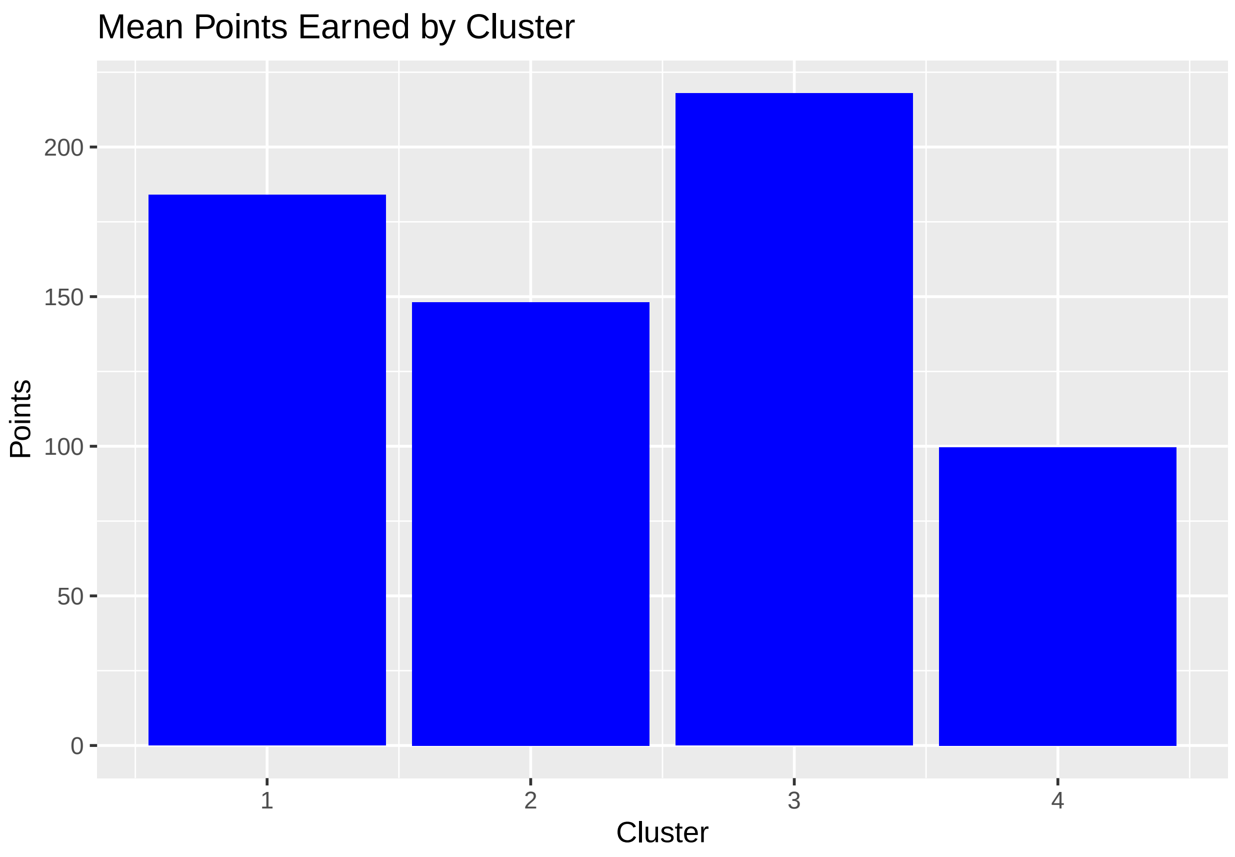
With that, we are left with four clusters, wherein cluster 1 contains observations for Albania; cluster 2 contains observations for Croatia and Italy; cluster 3 contains observations for Serbia, Cyprus, and Greece; Cluster 4 contains observations for Austria, Germany, the Netherlands, Spain, France, and the United Kingdom

```
joint_data_cluster <- joint_data_mean %>% mutate(Cluster = final$cluster) %>% group_by(Cluster) %>% summ

flextable(joint_data_cluster)
```

```
## Warning: fonts used in ‘flextable‘ are ignored because the ‘pdflatex‘ engine is
## used and not ‘xelatex‘ or ‘lualatex‘. You can avoid this warning by using the
## ‘set_flextable_defaults(fonts_ignore=TRUE)‘ command or use a compatible engine
## by defining ‘latex_engine: xelatex‘ in the YAML header of the R Markdown
## document.
```

Cluster	pts	place
1	184.00000	11.00000
2	148.16392	15.27113
3	218.00000	9.00000
4	99.66667	17.00000

```
joint_data_cluster %>% ggplot(aes(Cluster, pts)) + geom_bar(stat = "identity", fill = "blue") + ggtitle
```



Based on this descriptive cluster analysis, it would seem that countries in the second cluster (Croatia and Italy) are more likely to score the highest number of points, and thus, place higher

Conclusion

We successfully merged and analyzed European survey data with Eurovision contest outcomes, offering a interesting perspective on how cultural and social values may influence or reflect in a popular international event. The use of k-means clustering provided valuable insights, segmenting countries into distinct groups based on their sociocultural attributes and Eurovision contest performances.

The analysis highlighted intriguing trends, such as the unique position of countries like Croatia and Italy in securing higher points in Eurovision, possibly linked to their distinct cultural values.

Future Directions

This project lays the groundwork for several interesting avenues of research. One potential area is the exploration of longitudinal changes in cultural values and their impact on Eurovision outcomes over a more extended period. Additionally, integrating individual-level survey responses or detailed Eurovision audience voting patterns, could offer deeper insights.

Another interesting direction could be the application of more sophisticated machine learning techniques, like hierarchical clustering or neural networks, to better capture the nuances of cultural influence.

Another thing we could have done was conduct multiple K-means test after performing PCA dimension reduction and comparing the results.

Resources

https://uc-r.github.io/kmeans_clustering#:~:text=clustering%20algorithms%20%26%20visualization-,Data%20Preparation,scaled)%20to%20make%20variables%20comparable.