International Seminar Participation Analysis

A. Introduction

1. Information about the organization:

ISUF's aim is the international and interdisciplinary sharing of ideas, methods and findings concerned with urban form. The International Seminar on Urban Form is the academic organization serving researchers and practitioners worldwide. The ISUF annual conference takes place over four to five days in mid-summer to early fall. Cities and universities all over the world have sponsored the conference, which attracts 200 to 400 scholars and practitioners. In addition to paper presentation and keynote speakers, guests enjoy tours and a banquet designed to honor the place.

2. Project Objective:

The aim of our project is to examine this dataset created by people who attended this conference, to make analyses on it and to clean the dataset. The cleaning step is the most important part of the project. Because in order to prevent inconsistency in the visualizations we will make, we need to clean our data well and make it ready for visualization.

3. Data Overview:

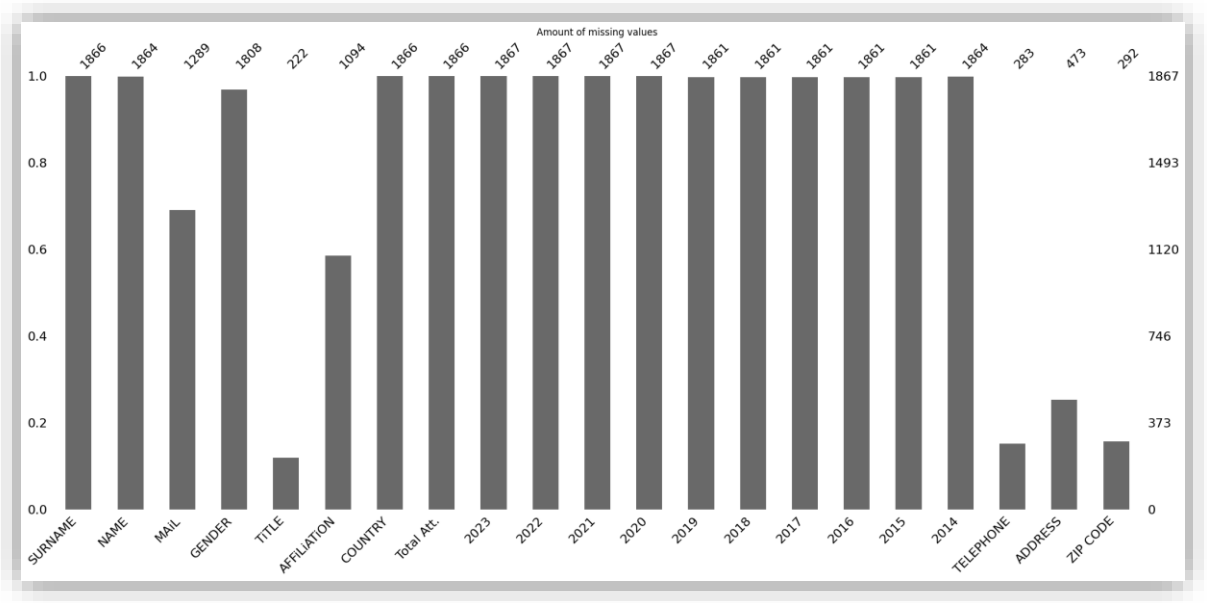
The dataset we received came in Excel format and has 1867 rows and 21 columns. Brief information about the columns:

- ***Surname***: Surname of participant
- ***Name***: Name of participant
- ***Mail***: Mail of participant
- ***Gender***: Gender of participant (Mail, Female, U)
- ***Title***: Title of participant (Dr. , Prof. , Untitled)
- ***Affiliation***: Affiliation of participant
- ***Country***: Participant's country
- ***Total Att***.: Total number of conference attendees
- ***2023*:** whether the participant participated this year or not
- …
- ***2014***: whether the participant participated this year or not
- ***Telephone***: Participant's telephone
- ***Address:*** Participant's address
- ***Zip Code:*** Participant's zip code

B. Data Preparation:

1. Data Cleaning:

- Handling missing values



There are many missing values in our dataset. Let's consider these missing values for each column.

For **Telephone**, **Address**, **zip code** and **affiliation** columns, we can remove these columns directly, they will not be useful for our further analysis.

Our missing value numbers for the remaining columns are as follows:

Surname	1
Name	3
Mail	578
Gender	59
Title	1645
Country	1
Total_Att	1
2023	0
2022	0
2021	0
2020	0
2019	6
2018	6
2017	6
2016	6
2015	6
2014	3

We have an incorrect row for the **Surname** column. and this row affects other columns. We remove this row directly. You can see this row from figure 1.

We are left with 2 rows with empty values for the **Name** column. When we examine these, we cannot make any inferences based on the other people in our data set. Therefore, we delete these 2 rows as well. You can see this 2 rows from figure 2.

There are many users whose **Mail** column is empty. People may not have mail, or they may have encountered errors during retrieval, so we thought it would be appropriate to replace these empty values with the word Unspecified. Similarly, let's write Unspecified instead of unspecified values in the **Title** column.

For the **Gender** column, values not already specified in the data set are written as U. We can also write U instead of empty values.

	Surname	Name	Mail	Gender	Title	Country	Total_Att	2023	2022	2021	2020	2019	2018	2017	2016	2015	2014
1866	NaN	NaN	NaN	NaN	NaN	NaN	NaN	235	312	228	236	276.0	212.0	389.0	142.0	248.0	290.0

Incorrect row for Surname. Figure 1

	Surname	Name	Mail	Gender	Title	Country	Total_Att	2023	2022	2021	2020	2019	2018	2017	2016
1231	xia	NaN	460105085@qq.com	U	NaN	Unidentified	1.0	0	0	0	0	0.0	0.0	0.0	1.0
1330	zhang	NaN	1102277260@qq.com	Male	NaN	China	4.0	0	2	1	0	0.0	0.0	1.0	0.0

Rows that have missing values for Name column. Figure 2

!! We have users who do not have any participation information in some years. Since they never participated in total, they will not be useful in our analysis. Therefore, we can delete these rows as well. You can see these deleted rows below. Apart from these, there are also users whose total participation number is 0. We have incorrect rows in these, so we remove them from our data set.

Name	Mail	Gender	Title	Country	Total_Att	2023	2022	2021	2020	2019	2018	2017	2016	2015	2014
francisco	falanizu@ucalgary.ca	U	Non-Titled	Canada	0.0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN
jeff	kunstadt@hotmail.com	male	Non-Titled	Unidentified	0.0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN
ana	Unspecified	U	Non-Titled	Portugal	0.0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	0.0
kattia	Unspecified	U	Non-Titled	France	0.0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	0.0
stephen	smwheeler@ucdavis.edu	U	Non-Titled	United States	0.0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN
cheol-jae,	Unspecified	Male	Non-Titled	South Korea	0.0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	0.0

2. Data Transformations:

- There were incorrectly written values in some columns. We fixed them by applying some transformations.
- Firstly, the **Gender** column contained values such as 'Female', 'Male', 'male' 'male ', 'U', 'M'. I corrected these. I changed the rows that said U to Unspecified.
- There were many incorrect values in the **Title** column, as you can see below. I corrected them and changed it to non-titled.

```
['Non-Titled' 'Dr.' 'Prof.' '53-523' '13002' 11120 '86170' '06200' 11000
 '89202450' '12327-490' '210096' '5616VD' '34696' 8034 '20146' 50009 46002
 '200092' '213299' '20000' '200082' '200063' '3200003' '6230' '78000'
 '28035' '121433' '08015' '81024' '999077' '06100' '40127' '76121'
 '620015' '117566' 'Hong Kong' '812 45' '01228-100 ' '300131' 210000
 '361021' '08014' '06680' '34794' '50018' 210018 '100081' '100871'
 '1700-351' '518055' 200000 '100085' '87.013-260' '16345' '7708'
 '305-8577' '518071' '30-864' '20133' '81104' '350000' 100084 '2000092'
 '45-758' '30.310-380' '1715' '400045' '300000' ' 200092' 'N179GJ'
 '31-515' 41092 '9040' '06 204 Nice Cedex 3' '34307' '06790' '06800'
 'NG9 2QZ' 33613 '20131' 'CA 94705' '50-137' '200030' '11000'
 '2200 Copenhagen N' '10125' '6954916' '300130' '08024 ' '4000' '201306'
 210096 '310003' '30 330 220' '4320 ' '1649-026' '4445-434' '1010' 1000
 '000000' '100000' '34377' '210018' '8037' 300000 '510641' '210008'
 '510000' '310023' '300072' 'zhaoxueying1999' '210093' '100025' '20132'
```

Uniqu values for Title row

- There were some countries written incorrectly in the country column. For example, Brazil was written as Brasil, but it should have been Brazil. I corrected these.

- Additionally, I converted the participation numbers from float type to integer type.

3. Correcting data:

- The biggest problem in our dataset was that there were duplicate names and surnames due to some problems during the data collection phase. But these duplicates were not exactly the same, they were in the dataset with only a few letters different. For example, a person whose name and surname were olgun caliskan was in the dataset as olgun çaliskan in one more row. To fix this, I wrote an algorithm using Python's rapidfuzz library that would traverse the dataset and find pairs with more than 80% similarity in both name and surname. In this way, I caught about 30 pairs. And I cleaned them from the dataset.

- I will put the code version of this algorithm at the end of my report for those who are curious.

- I am putting some columns below as an example.

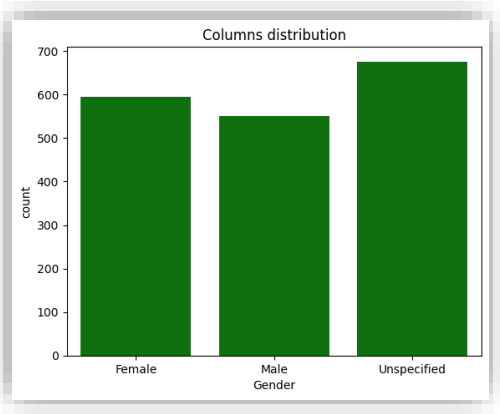
	dropped_index	dropped_name	dropped_surname	similar_to_index	similar_name	similar_surname
0	1495	md mustiafiz	al mamun	17	mustiafiz	al mamun
1	1447	inesa	alistratovaitė-kurtinaitienė	27	inesa	alistratovaite-kurtinaitiene
2	1800	olgu	caliskan	143	olgu	çaliskan
3	1725	chen	changyu	185	chen	chang
4	1622	carlos dias	coehlo	217	carlos dias	coelho
5	1671	jeffrey	cohen	220	jeffrey a.	cohen
6	1821	staël de alvarenga pereira	costa	231	stael de alvarenga pereira	costa
7	1395	dalia	dijokienė	284	dalia	dijokiene
8	327	marco	falsetti	326	marco	falsetti

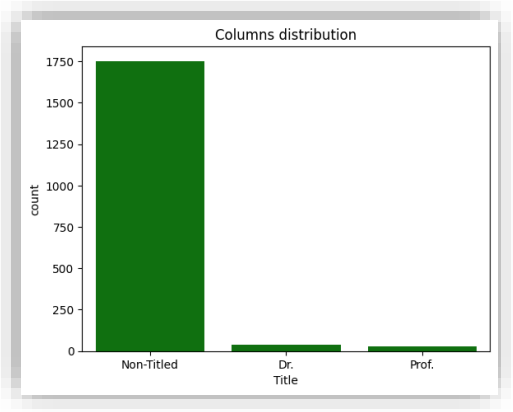
C. Data Analysis:

- This section includes the visualization steps and comments I have made.

1. General Attendance Statistics:

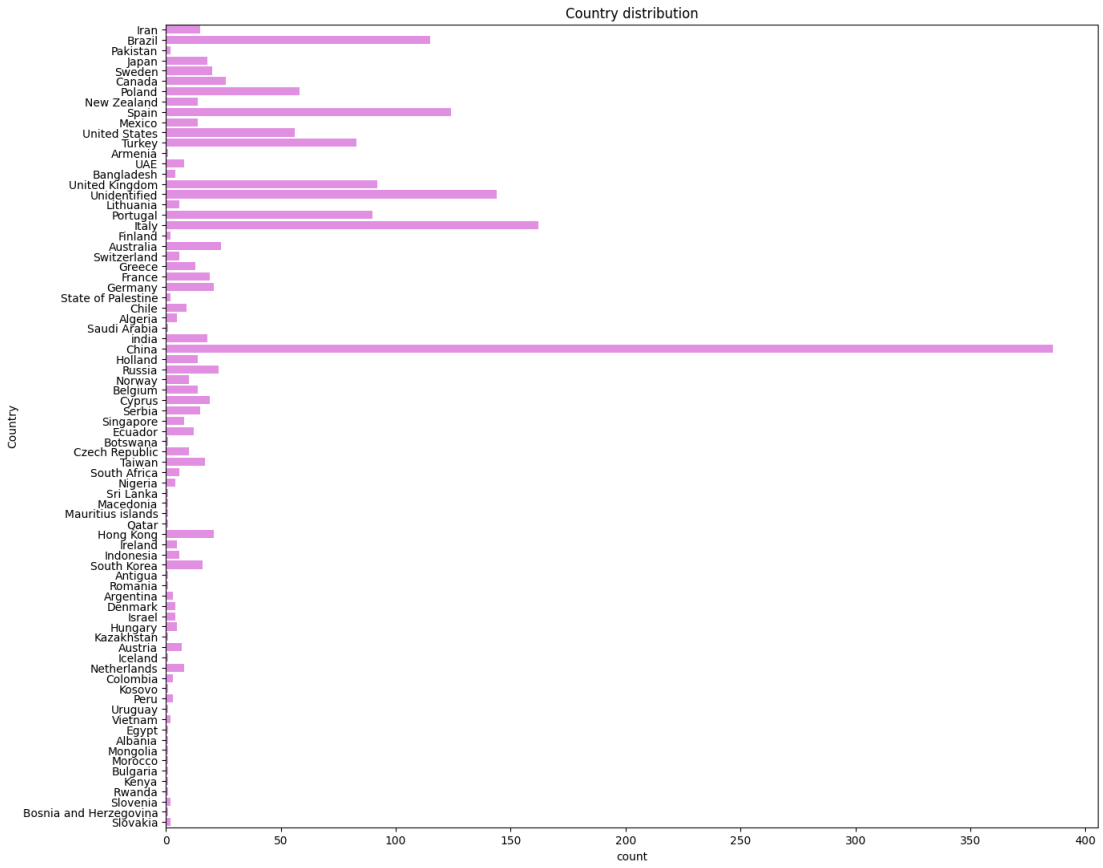
- First of all, if we look at the distributions in our data set, there is a smooth and balanced distribution in terms of gender.





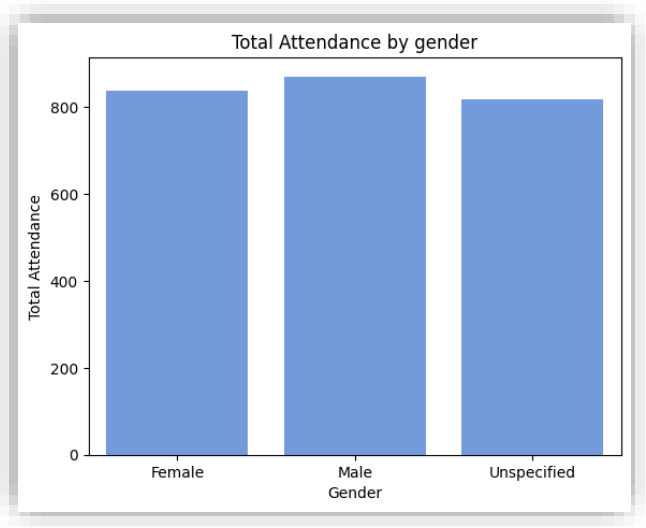
- If we look at the Title column, we can talk about an unbalanced distribution here. But in this case, it is normal, because the number of professors and doctors who participated is expected to be less than the non-titled ones.

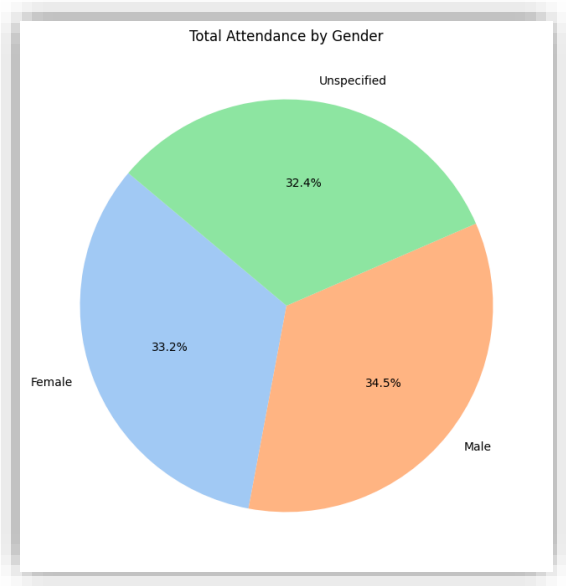
- If we look at the situation of the countries, we see that advanced modern societies such as China, Brazil, America, Italy and Spain stand out here.



2. Gender-Based Attendance Analysis:

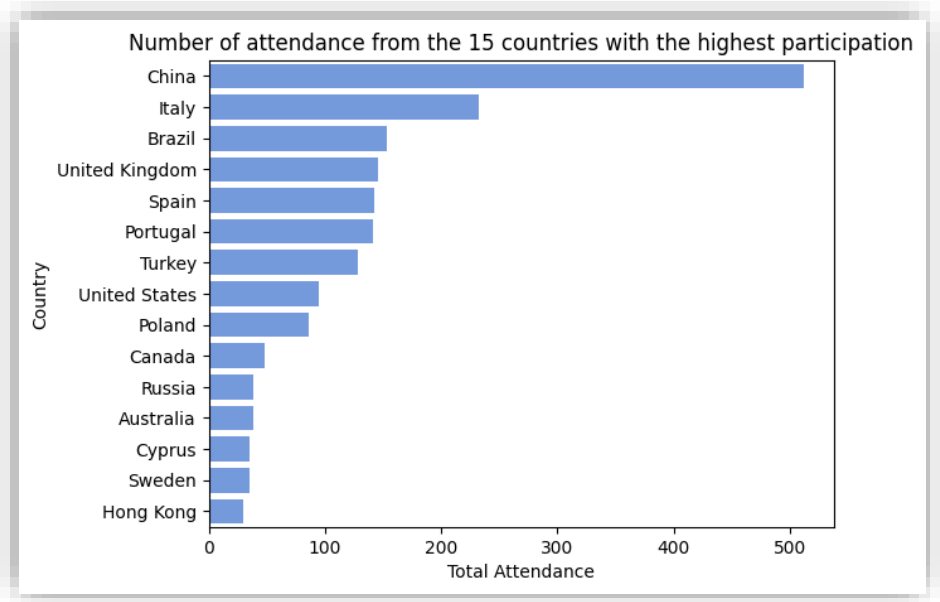
- When we look at the total participation numbers based on gender, we can say that it is balanced again.
- You can also see this in a pie chart



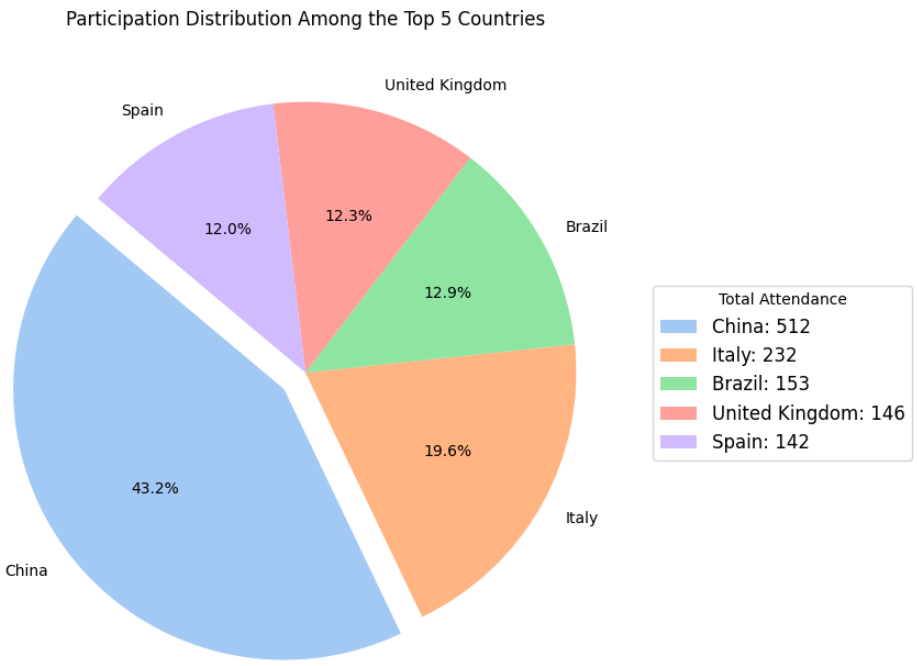


3. Region-Based Attendance Analysis:

- Below you can see the top 15 countries that attended the conference the most.

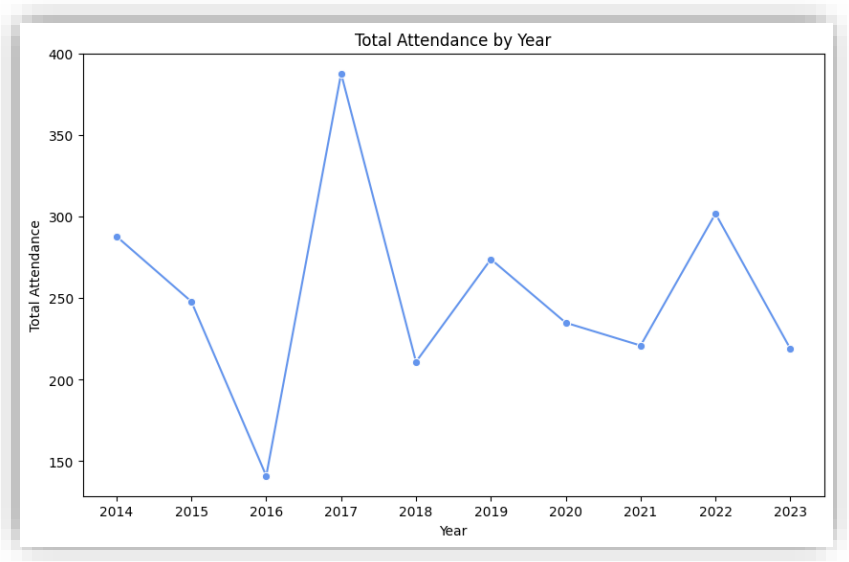


!! And again, you can see the countries that attended the most with a pie chart and see how much China was present in this conference.

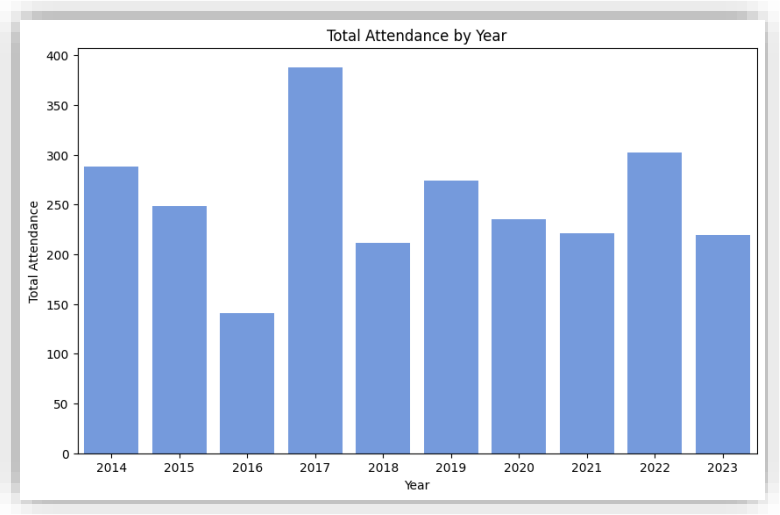


4. Time Series Analysis:

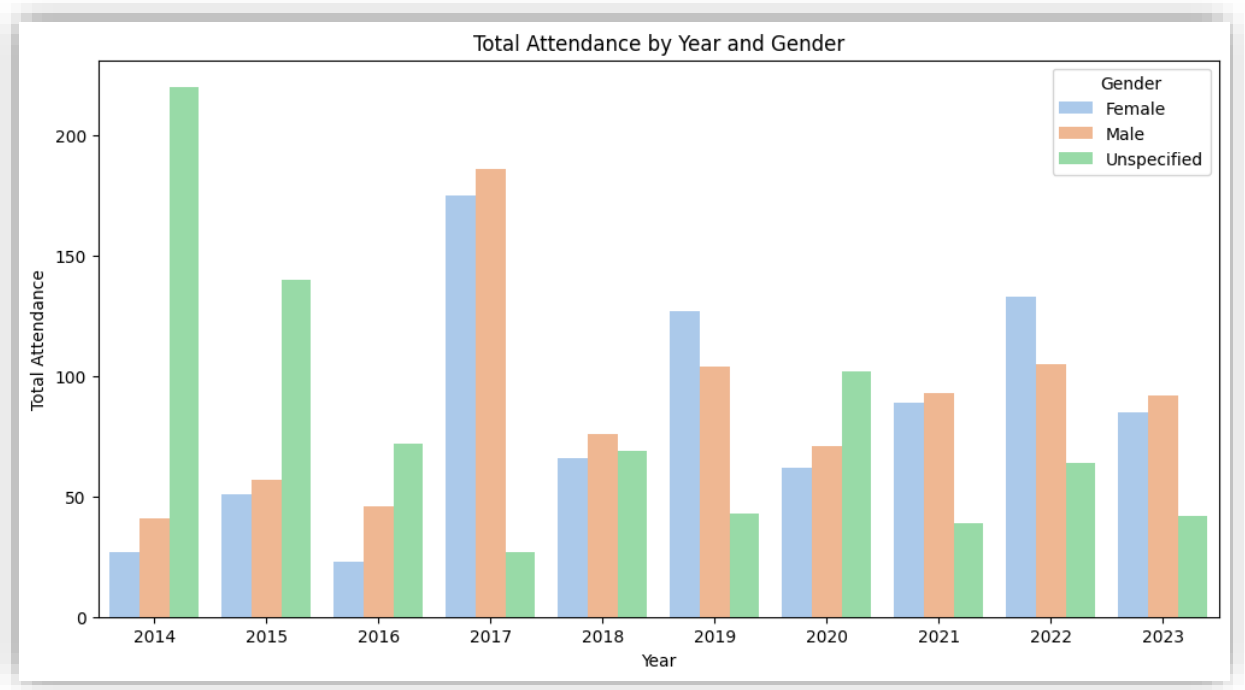
- This section will include year-based analyses and visualizations.
- As you can see in the graph above, a record attendance was broken for the conference in 2017. The conference is held in a different country every year, so this is likely due to this. And you can also see that attendance has decreased throughout 2020 and 2021, which can be said to be due to the coronavirus outbreak in those years.



- Again, you can see this with a bar chart.

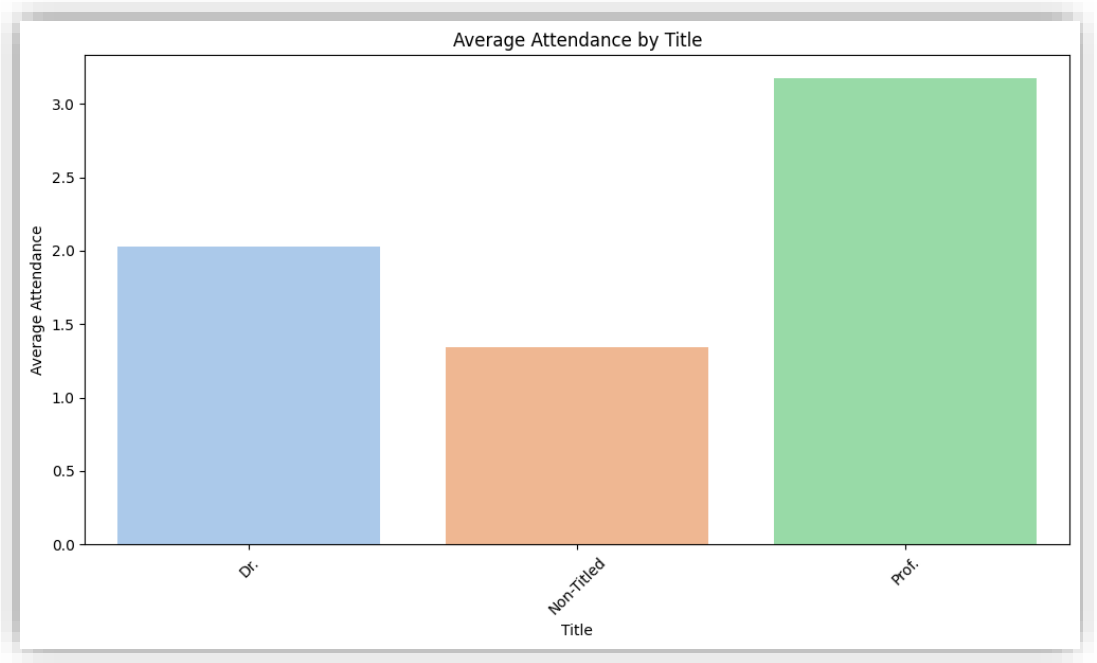


- I would also like to present you a chart divided by gender. You can see the participation rates of genders by year.



5. Title-Based Attendance Analysis:

- When we look at the average participation numbers on titles in this section, we see that, as expected, people with the titles of prof and dr actually show more participation on average than those without titles.



D) Code Snippet:

```python
import pandas as pd
from rapidfuzz import fuzz

threshold = 80

to_drop = []
dropped_rows_info = []

def is_valid_index(index, df):
    return index >= 0 and index < len(df)

for i in range(len(df)):
    for j in range(i + 1, len(df)):
        if not (is_valid_index(i, df) and is_valid_index(j, df)):
            continue

        try:
            name_similarity = fuzz.ratio(df.loc[i, 'Name'], df.loc[j, 'Name'])
            surname_similarity = fuzz.ratio(df.loc[i, 'Surname'], df.loc[j, 'Surname'])

            if name_similarity > threshold and surname_similarity > threshold:
                to_drop.append(j)
                dropped_rows_info.append({
                    'dropped_index': j,
                    'dropped_name': df.loc[j, 'Name'],
                    'dropped_surname': df.loc[j, 'Surname'],
                    'similar_to_index': i,
                    'similar_name': df.loc[i, 'Name'],
                    'similar_surname': df.loc[i, 'Surname']
                })
        except KeyError as e:
            pass

df_dropped_info = pd.DataFrame(dropped_rows_info)

df_cleaned = df.drop(to_drop).reset_index(drop=True)
```

- This code is designed to identify and remove duplicate rows in a DataFrame by comparing the similarity of names and surnames between different rows. Using the **fuzz.ratio** function from the **rapidfuzz** library, it calculates the similarity scores between the names and surnames in two different rows. If both the name and surname similarity scores exceed a predefined threshold (set to 80), the row is considered a potential duplicate and is marked for removal.

- The indices of rows identified as duplicates are stored in a list (**to_drop**), and detailed information about each dropped row, including its index, name, surname, and the index of the similar row, is recorded in another list (**dropped_rows_info**). This information is then converted into a DataFrame (**df_dropped_info**). Finally, the code removes the identified duplicate rows from the original DataFrame and resets the index, resulting in a cleaned DataFrame (**df_cleaned**).