Laboratory of Data Science

Gianni Andreozzi

Martina Trigilia

AA.2021/2022

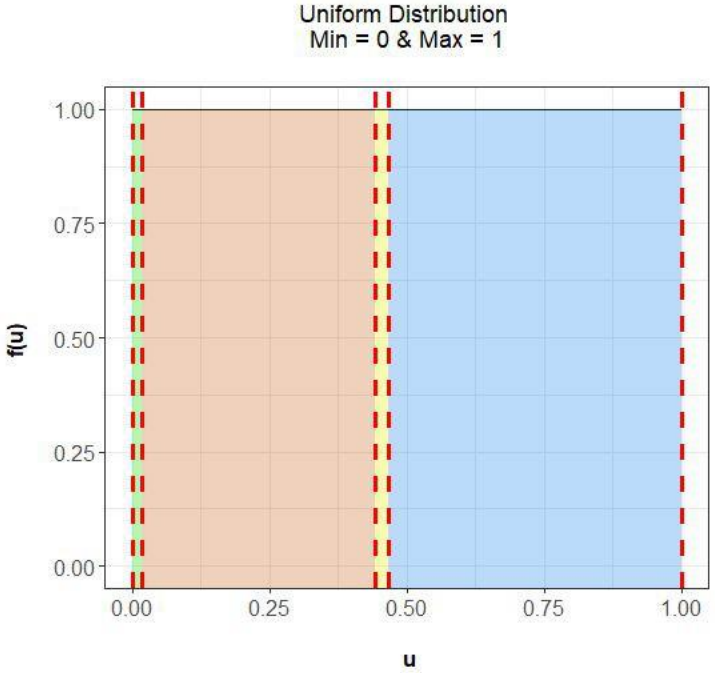
Part 1

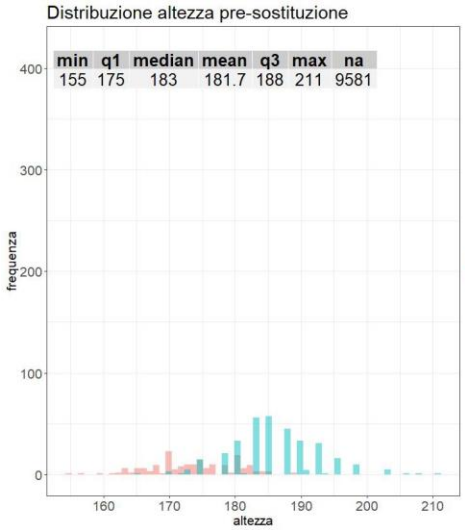
Pre Processing

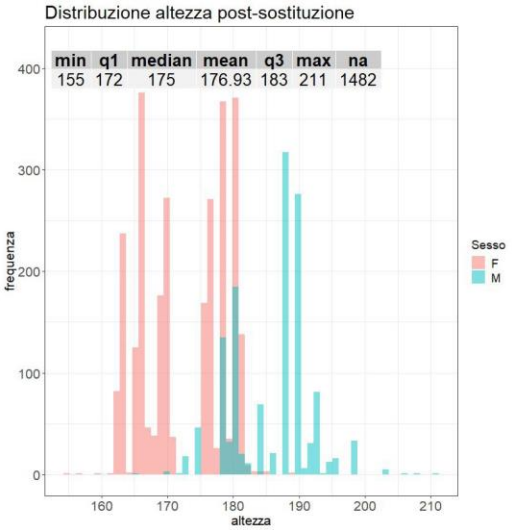
- **Correction of id errors**: different player_id but same person or same player_id but different names; same tourney_id associated with different names and different tourney level; same tourney_id and same match_num (match_id) but different players ⇒ Generation of new ids by *concatenation*.
- **Spelling errors in female_players.csv and male_players.csv**
- **IOC codes**:
  - Standardisation: e.g. DEU ⇨ GER
  - Spelling errors
  - Missing data ⇨ *https://en.wikipedia.org/wiki/List_of_IOC_country_codes*

Missing Values

Missing data were initially searched in other records for matching ids.







Code Structure

- Main function **create_tablecreate** ⇨ structure (header and *set of* ids) + reads the tennis table in a loop and fills the respective tables (match.csv, player.csv and tournament.csv).

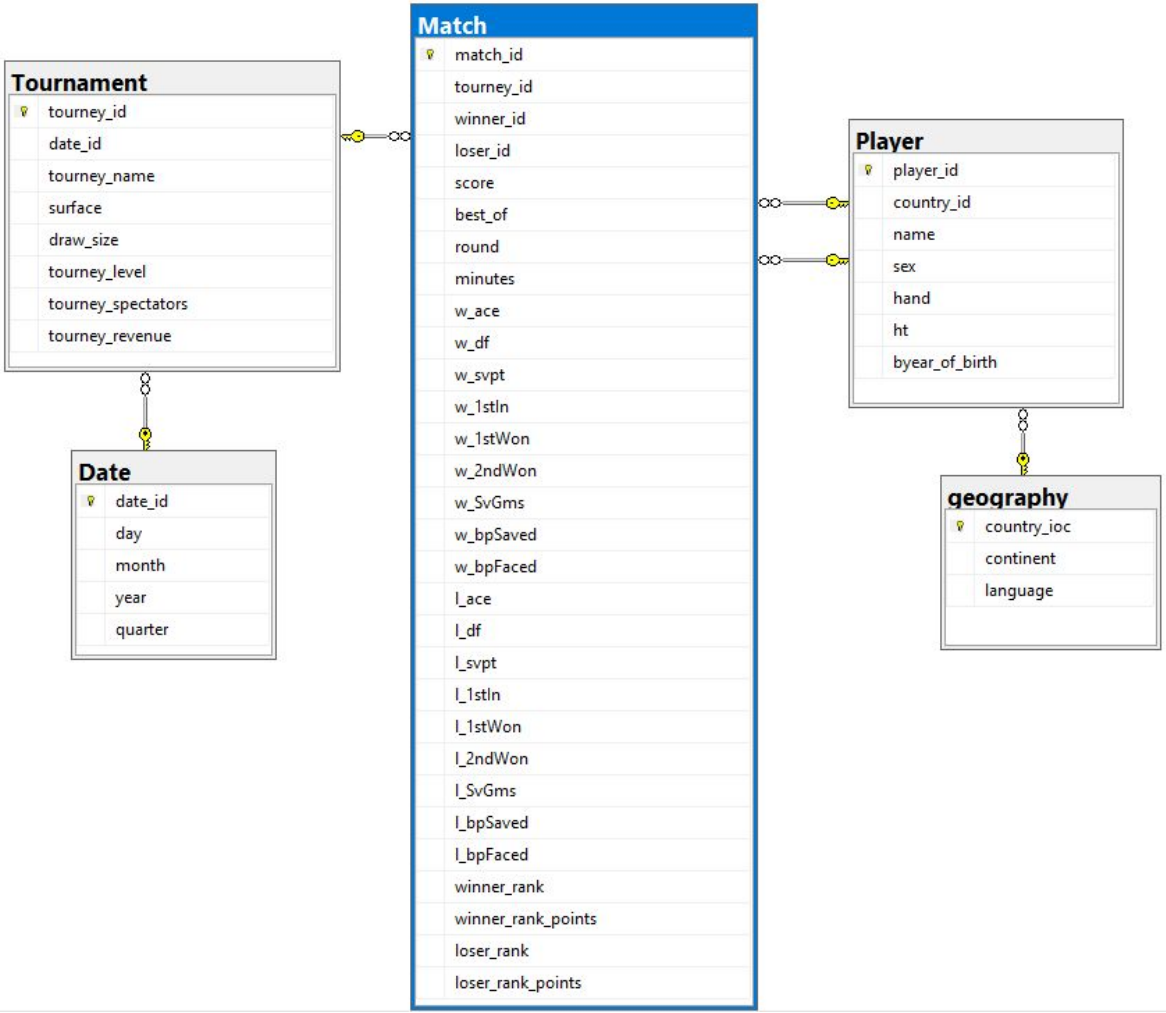
```
for row in tennis_db:
    create_match_row(row, match_from_tennis, file_match)

    create_tournament_row(row, tournament_from_tennis, tourney_ids, file_tournament)

    create_date_row(row, tourney_date_ids, file_date)

    create_player_row(row, player_winner_from_tennis, players_ids, sex_players, file_player, row["winner_id"]
                      , row["winner_name"], row["winner_age"])
    create_player_row(row, player_loser_from_tennis, players_ids, sex_players, file_player, row["loser_id"],
                      row["loser_name"], row["loser_age"])
```

Code Structure

- **Date.csv**: the date is transformed into datetime and the required attributes are extracted (the quarter via *get_quarter()*)
- **Player.csv**: *map_player()* creates a dictionary that maps the player's name to his gender; get_sex*()* gets the gender of the player by searching the dictionary; get_year_of_birth*()* extracts the year of birth via the tournament date and the player's years.
- **Geography.csv**: Creation of a dict associating the name of a country with its language (obtained thanks to coutry_list.csv). Reading country.csv and writing a table with the respective fields.

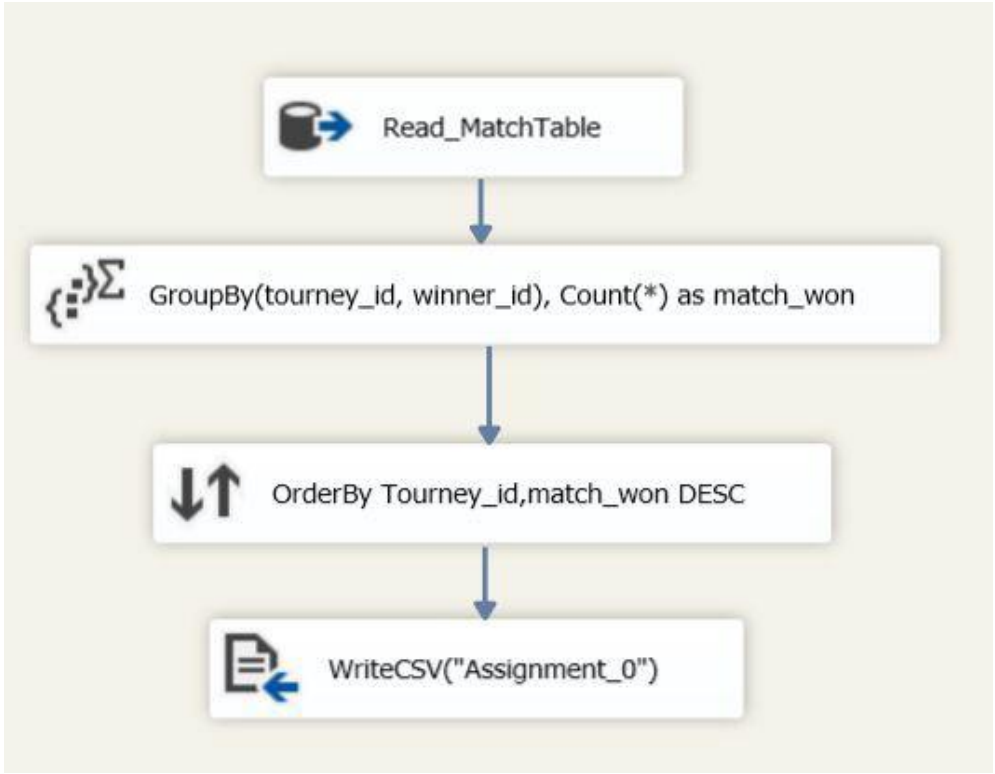
DW creation and data loading

- DW creation - SQL *Server Management Studio*
- Null values not allowed for primary keys
- Connection to the DB via the pyodbc library
- Constraints of primary and foreign keys: first the loading of the *Dimensions* Tables (commit immediately after loading each table)
- Loading the *Fact* Table: commit after each fifth of the table
- Connection closure

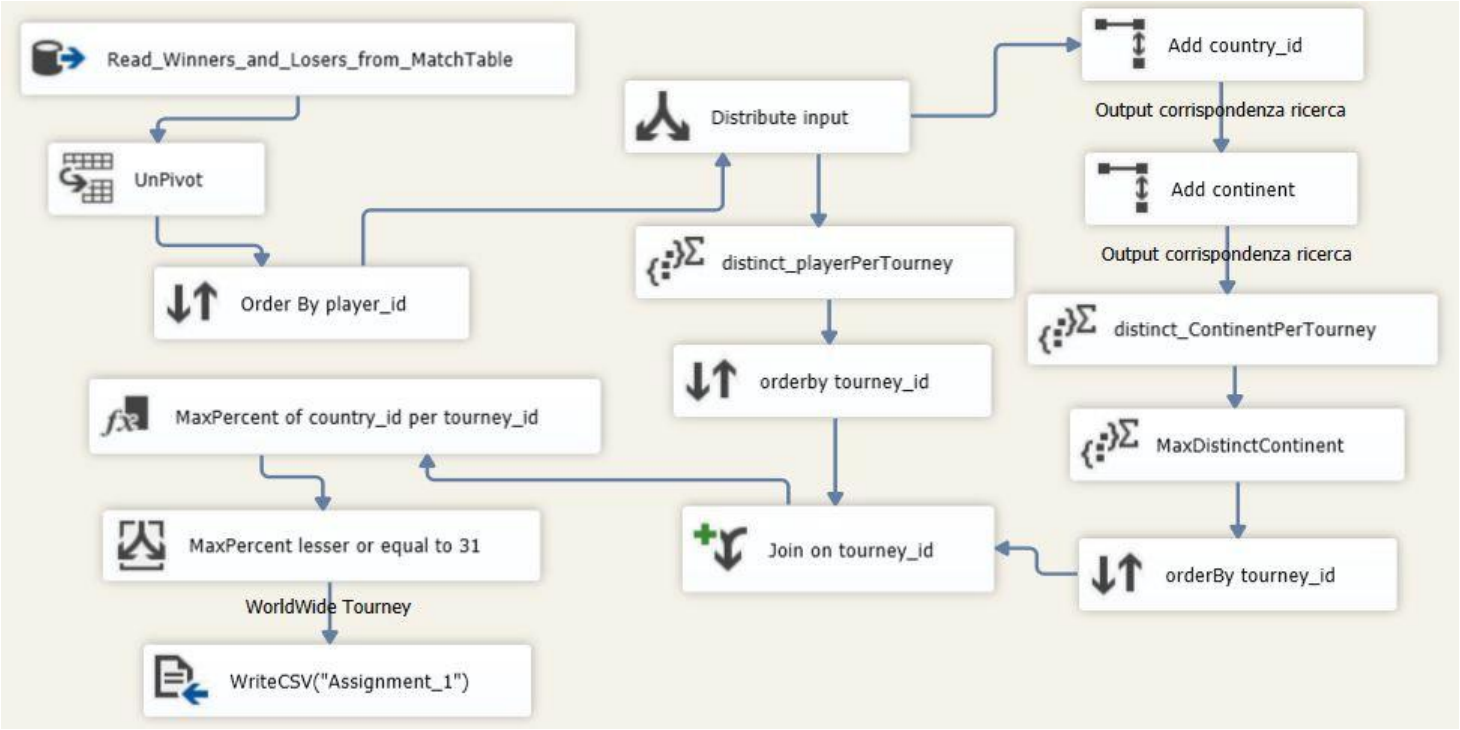

Tournament
tourney_id
date_id
tourney_name
surface
draw_size
tourney_level
tourney_spectators
tourney_revenue
Match
match_id
tourney_id
winner_id
loser_id
score
best_of
round
minutes
w_ace
w_df
w_svpt
w_1stIn
w_1stWon
w_2ndWon
w_SvGms
w_bpSaved
w_bpFaced
l_ace
l_df
l_svpt
l_1stIn
l_1stWon
l_2ndWon
l_SvGms
l_bpSaved
l_bpFaced
winner_rank
winner_rank_points
loser_rank
loser_rank_points
Player
player_id
country_id
name
sex
hand
ht
byear_of_birth
Date
date_id
day
month
year
quarter
geography
country_ioc
continent
language

Part 2

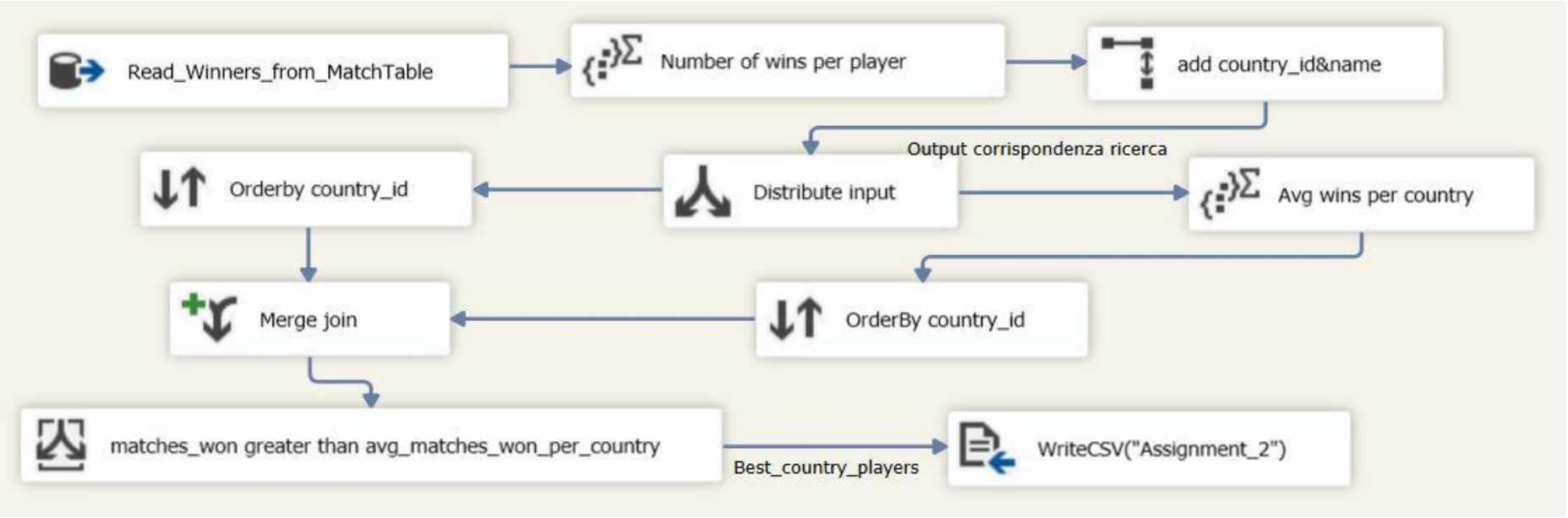
Assignment 0: *For every tournament, the players ordered by number of matches won.*



Assignment 1: *A tournament is said to be "worldwide" if no more than 30% of the participants come from the same continent. List all the worldwide tournaments.*



Assignment 2: *For each country, list all the players that won more matches than the average number of won matches for all players of the same country.*



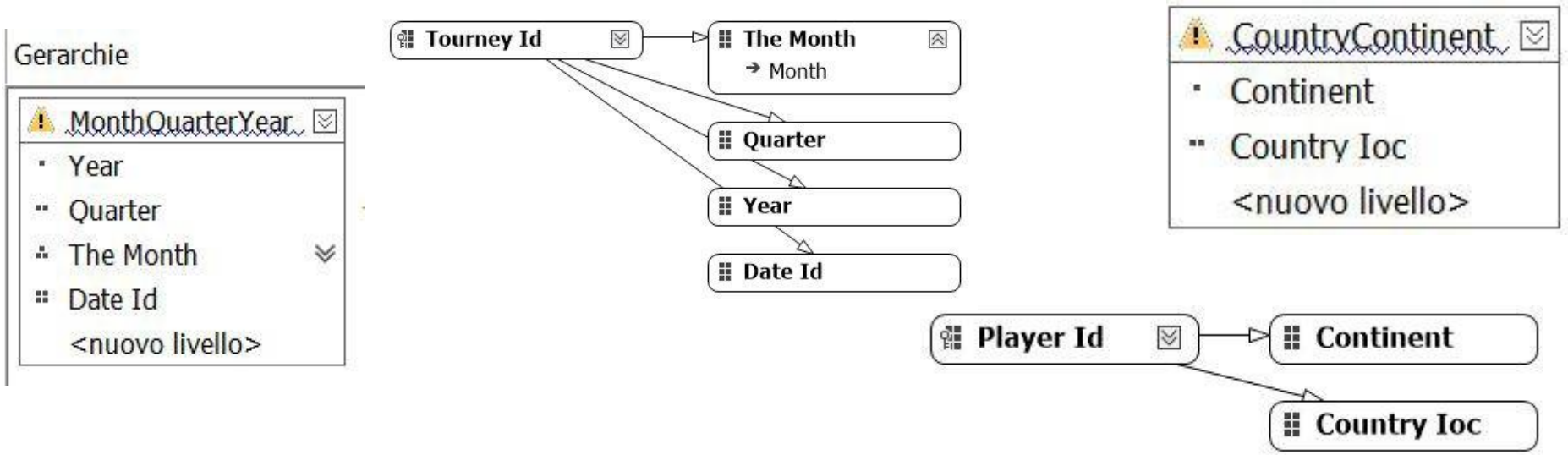
Part 3

Assignment 0: *Build a datacube from the data of the tables in your database, defining the appropriate hierarchies for time and geography. Use the rank and rank points of the winner and loser as measure.*

Creating the dimensions (Player and Tournament) and the cube.

Definition of hierarchies:

- CountryContinent (Continent → Country_ioc) within the Player dimension
- ***MonthQuarterYear*** (Year → Quarter → The Month → Date Id) within the Tournament dimension. The Month represents the month of the year and has been derived from Month *('New Named Calculation')* and the functional dependency The Month → Month has been defined.



Assignment 1: *Show the percentage increase in winner rank points with respect to the previous year for each winner*

```
WITH MEMBER abs_incr AS
([Tournament].[MonthQuarterYear].currentmember.lag(1), [Measures].[Winner Rank Points] )

MEMBER diff AS
[Measures].[Winner Rank Points] - abs_incr

MEMBER perc_incr AS
CASE WHEN abs_incr = 0 THEN '-' ELSE diff/abs_incr END,
FORMAT_STRING = 'percent'.

SELECT {[Measures].[Winner Rank Points], perc_incr} ON COLUMNS,
NONEMPTY (([Winner].[Name].[Name], [Tournament].[MonthQuarterYear].[Year])) ON ROWS
FROM [Group_17];
```

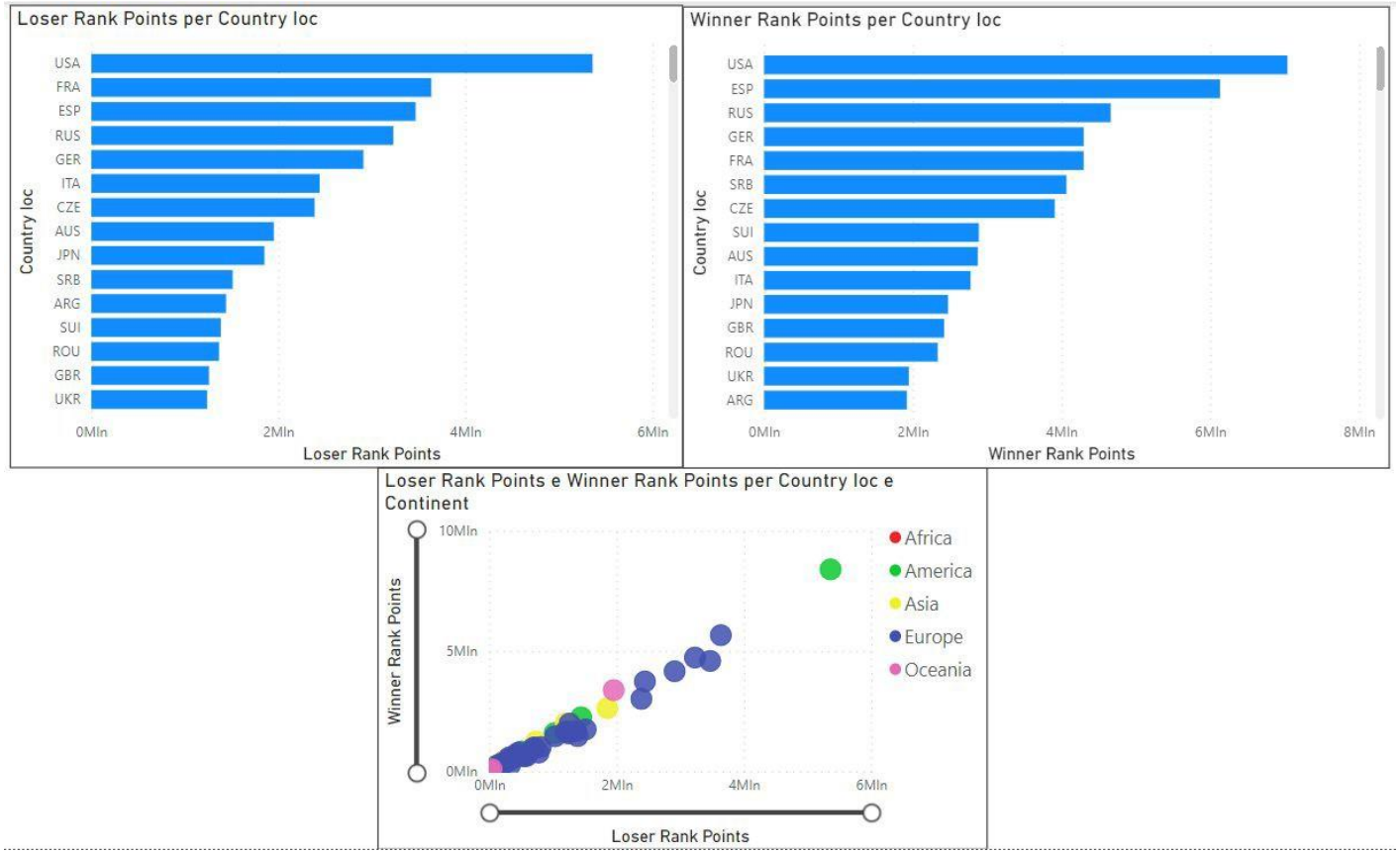
Assignment 2: *For each country show the total winner rank points in percentage with respect to the total winner rank points of the corresponding continent.*

```
WITH MEMBER perc_country AS
[Measures].[Winner Rank Points]/([Winner].[CountryContinent].currentmember.parent,
[Measures].[Winner Rank Points]),
FORMAT_STRING = 'percent'.

SELECT {[Measures].[Winner Rank Points], perc_country } ON COLUMNS,
NONEMPTY (([Winner].[Continent].[Continent], [Winner].[CountryContinent].[Country Ioc])) ON ROWS
FROM [Group_17];
```

Assignment 3: *Show the losers having a total loser rank points greater than 10% of the totals loser rank points in each continent by continent and year.*

```
WITH MEMBER points_contyear AS ([Loser].[Name].currentmember.parent, [Measures].[Loser
Rank Points])

MEMBER ratio AS [Measures].[Loser Rank Points]/points_contyear ,
FORMAT_STRING = 'Percent'.

SELECT {[Measures].[Loser Rank Points], ratio, points_contyear } ON COLUMNS ,
NONEMPTY(
FILTER(([Tournament].[MonthQuarterYear].[Year], [Loser].[CountryContinent].[Continent],
[Loser].[Name].[Name]),
        ratio > 0.10)
) ON ROWS
FROM [Group_17];
```

Assignment 4: *Create a dashboard that shows the geographical distribution of winner rank points and loser rank points.*



Assignment 5: *Create a dashboard of your choosing, that you deem interesting w.r.t. the data available in your cube*