UNIVERSITÀ DI PISA

Laboratory of Data Science

Project Assignment - Part 1, 2, 3

Gruppo 11
Cristiano Landi
Francesco Falleni

Anno Accademico 2021/2022

Contents

1 Introduzione

I record rappresentano dati relativi a vari match di tennis. I dati disponibili sono divisibili per:

- Tournament: ogni torneo contiene le informazioni sul nome, data e caratteristiche specifiche (superficie, partecipanti, livello, numero di spettatori, incassi)
- Player: identificati da un id, contengono le informazioni sul nome, mano dominante, altezza e paese di provenienza.
- Geography: informazioni che riguardano il paese (codice ISO3 o IOC e nome esteso in lingua inglese), continente e sigla.
- Match: ogni record contiene le statistiche del match, le informazioni sui giocatori e sul torneo.

I dati sono stati forniti in quattro file distinti: *tennis.csv*, *male.csv* , *female.csv* e *countries.csv*. Il file *tennis.csv* richiede di essere normalizzato per poter estrarre la lista dei tornei, le date ed i giocatori. I file *male.csv* e *female.csv* contengono rispettivamente la lista dei giocatori di sesso maschile efemminile. Infine, il file *countries.csv* contiene la lista dei paesi che saranno associati ai giocatori per indicarne la nazione di provenienza.

2 Parte 1

2.1 Assignment 0: Creazione del database

Prima di tutto, è stato creato il data warehouse (Figura 1) usando il software SQL Server Management Studio. Il tipo degli attributi è stato determinato analizzando i dati forniti. Ad esempio, per determinare la lunghezza massima del tipo string, si è cercato[1] il valore massimo presente nei dati per ogni attributo.

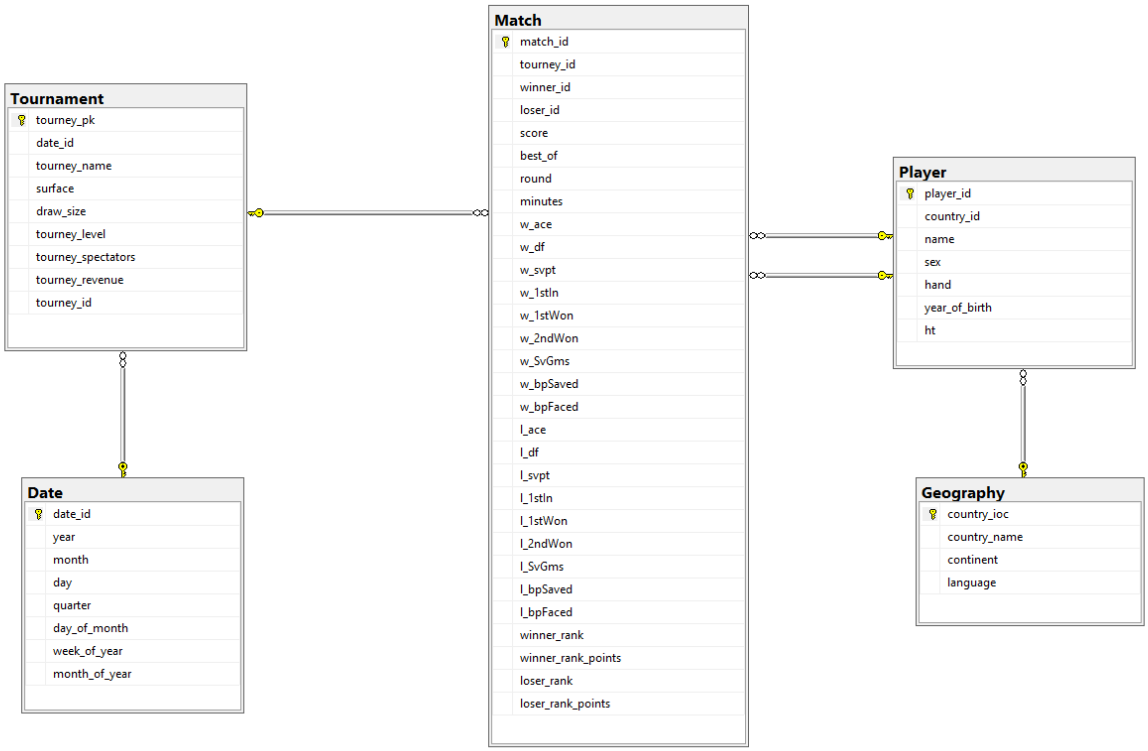


Figure 1: tennis db schema

[1] Si faccia riferimento al file "Missing values (Assignment 1.5).py"

2.2 Assignment 1: Integrazione dei dati

I dati contenuti nel csv fornito sono stati suddivisi in 5 file corrispondenti alle tabelle riportate nello schema in Figura 1.
È stata realizzata una funzione in grado di leggere un file contenente dati separati da uno specifico carattere o una serie di caratteri: in questo modo, con un unico metodo, si è in grado di processare file csv, tsv ed analoghi. La funzione genera una lista contenente, in ogni nodo, una riga, rappresentata come dizionario. Le chiavi del dizionario possono essere indicate al metodo tramite una lista passata come argomento o un booleano che, se true, indica alla funzione di trattare la prima riga del file come header.
Analogamente è stata realizzata una seconda funzione che, presa la lista dei dati, una seconda lista contenente solo i campi da salvare ed un path; salva un file csv contenente il sottoinsieme dei dati indicati con già i duplicati rimossi[2].
È stato necessario generare alcuni dati richiesti espressamente nell'assignment:

- Anno di nascita del giocatore: i dati originali contengono la data del match e l'età del giocatore (espressa come valore numerico con virgola), per calcolare l'anno di nascita è stato quindi sufficiente sottrarre alla data della partita l'età (convertita in giorni per una maggiore precisione).
- Lingua del giocatore: i dati forniti contengono solo il paese di provenienza del giocatore ed il codice ISO3 o IOC relativo. È stato prelevato da sito `geonames.org` la lista contenente la corrispondenza tra questi codice e la lingua parlata nel paese. Nel caso di più lingue ufficiali è stata scelta quella più parlata.

Nonostante i paesi siano identificati dal codice ISO3, IOC o entrambi, non è stato ritenuto opportuno uniformare la lista ad un unico codice in quanto, utilizzando l'attributo country_name al posto del country_ioc, è possibile eseguire tutte le query analitiche senza il problema di discriminare quale dei due codici è stato usato.
A partire dalla data del torneo sono state estratte informazioni aggiuntive come Quarter (1,2,3,4), day_of_week (Monday, Tuesday, ecc.) e week_of_year (1, 2, ..., 53).
Infine, sempre in questa fase, si sono generate le chiavi primarie per le tabelle, anche se successivamente, durante l'inserzione, è stato necessario generare una nuova chiave per match in quanto quella consigliata nell'assignment non risulta univoca. È necessario evidenziare che tale chiave non viene utilizzata per nessuna condizione di join e, di conseguenza, anche se generata non univoca nel momento di separazione del CSV principale, non crea nessuna perdita di informazione.

2.3 Missing values

Il dataset si presenta con alcuni attributi aventi valori mancanti, di seguito vengono riassunte le strategie adottate per rimpiazzare i valori nulli:

- *Match* 185.764 record totali
  - *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* 56% di valori mancanti. Dato l'elevato numero di valori mancanti, è stato deciso di eliminare questi record.
  - *minutes* 56% di valori mancanti. Il 99% di questi valori nulli appartengono agli stessi record del punto precedente e quindi sono già stati eliminati. I restanti valori sono stati sostituiti usando la funzione $ffill$ e $bfill$ di Pandas con criterio di raggruppamento il torneo (*tourney_id*)
  - *score* $< 1\%$ valori mancanti. Questi valori risultano collegati a quelli del punto precedente. Avendo eliminato i record al punto precedente, anche questi valori mancanti risultano trattati.

[2] I duplicati sono stati identificati trasformando la lista di dizionari in un set, ossia, si sono rimossi solo i record con tutti i valori esattamente uguali.

  - *winner_rank* e *winner_rank_points* 10% di valori mancanti. I valori mancanti rimanenti dall'eliminazione del punto precedente sono stati sostituiti usando la funzione ffill e bfill con criterio di raggruppamento l'id del vincitore (winner_id).

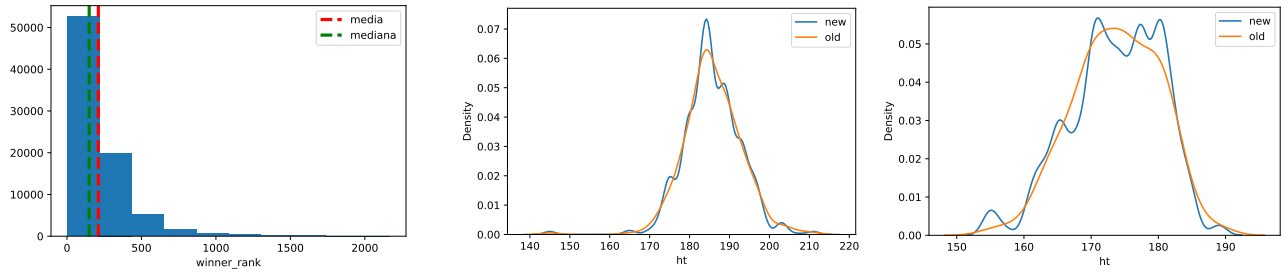
    Usando questo criterio sono però rimasti alcuni valori nulli, si è deciso di utilizzare la mediana dato che la distribuzione dei dati ha la forma di una esponenziale (Figura 2a).
  - *loser_rank* e *loser_rank_points* 19% di valori mancanti.

    I valori rimanenti dall'eliminazione precedente sono stati sostituiti usando la funzione ffill e bfill con criterio di raggruppamento l'id del perdente (loser_id).

    Anche in questo caso, sono rimasti alcuni valori mancanti che sono stati rimpiazzati usando la mediana.

- *Tournament* 4911 record totali
  - *surface* 1% di valori mancanti. Si è cercato di recuperare i valori mancanti utilizzando le funzioni $ffill$ e $bfill$ con criterio di raggruppamento il torneo (*tourney_name*). Dove non è stato possibile trovare un valore si è utilizzato il valore di maggioranza 'Hard'.

- *Player* record totali 10.074
  - *year_of_birth* 21% di valori mancanti. Non avendo un criterio di raggruppamento ragionevole per sostituire i valori mancanti, si è utilizzato le funzioni $ffill$ e $bfill$ che hanno mantenuto la distribuzione invariata.
  - *hand* $< 1\%$ di valori mancanti. I valori mancanti sono stati sostituiti con la categoria di maggioranza 'U'.
  - *ht* 95% valori mancanti. I valori mancanti dell'altezza sono stati rimpiazzati utilizzando le funzioni $ffill$ e $bfill$ con criterio di raggruppamento il sesso del giocatore.



(a) Distribuzione winner_rank (b) Altezza Male prima e dopo (c) Altezza Female prima e dopo

Figure 2

2.4 Assignment 2 - Caricamento dei dati

Dopo aver trattato i valori mancanti si è proseguito con l'inserimento dei record nel database.
L'inserimento delle 5 tabelle deve essere effettuato rispettando l'ordine imposto dai vincoli di chiave esterna, quindi per prime vengono inserite le tabelle Date e Geography, successivamente Player e Tournament ed infine la fact table Match.
Sono state realizzate due funzioni per inserire i record di una tabella nel database: la prima funzione utilizza il comando *execute* che effettua l'inserimento di un record per volta (fa una commit per ogni record). La seconda funzione utilizza il metodo *executemany* che effettua l'inserimento n record alla volta (fa una commit ogni n record), il numero di record n viene indicato con il parametro *chunk_size*.
Il processo di inserimento dei record non è immediato e potrebbero verificarsi degli errori in ogni momento (ad esempio di connessione).
Per evitare di dover eseguire nuovamente l'inserimento, viene controllato con una query il numero di record già presenti nella tabella e ripristinato l'inserimento a partire dall'ultimo record non inserito.

Durante la fase di inserimento della tabella Player è risultato che alcuni giocatori avevano un country_IOC non presente nella tabella Geography. Si è provato ad aggiungere i paesi mancanti utilizzando la lista fornita da `geonames.org` recuperandone una buona parte.

3 Parte 2

3.1 Assignment 0

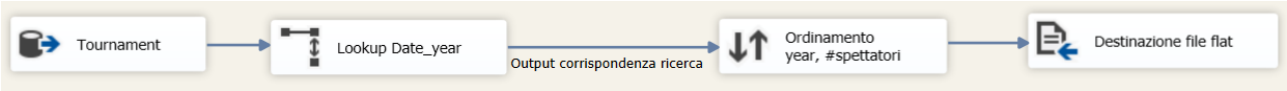


Figure 3: For every year, the tournaments ordered by spectators

In questo primo assignment è stato sufficiente caricare i soli attributi tourney_pk, tourney_name e date_id dalla tabella Tournament. In seguito è stato utilizzato l'operatore lookup per recuperare, tramite date_id, l'attributo year dalla tabella Date. Si è poi proseguito con l'ordinamento dei record per anno e numero di spettatori ed infine salvato sul file "assignment0.csv" il risultato.

3.2 Assignment 1

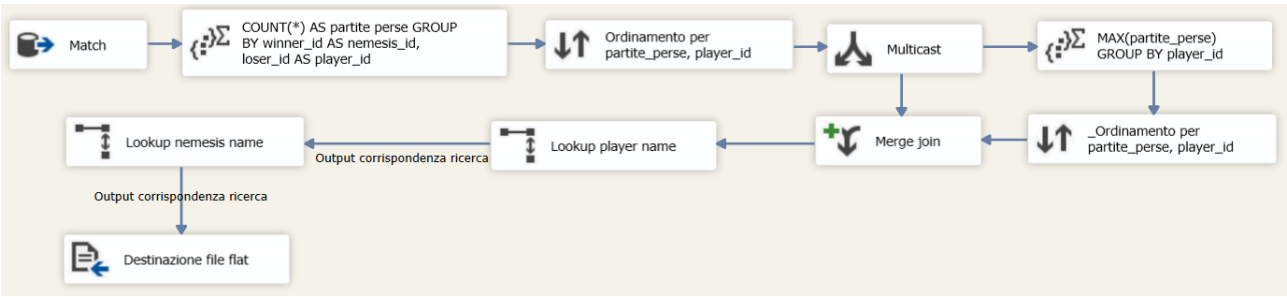


Figure 4: For any given player, his or her "nemesis" is the player against whom he or she lost most matches. List every player with the respective nemesis and the number of matches lost.

In questo assignment si è proceduto dividendo l'esercizio in più sotto-problemi:

1. Calcolare, per ogni coppia (vincitore, perdente) il numero di partite perse: dopo aver caricato la fact table Match, si è calcolato quante volte la coppia (vincitore, perdente) fosse presente nella tabella. Inoltre, si è sfruttato tale operatore per rinominare gli attributi winner_id e loser_id rispettivamente in nemesis_id e player_id.

2. Selezione dell'avversario con cui un dato giocatore ha perso più frequentemente: in questa fase, l'operazione più difficile è stata quella di usare l'operatore di aggregazione MAX per individuare la nemesi senza perdere le informazioni su di essa con i soli operatori di Visual Studio. Per fare ciò è stato necessario usare l'operatore multicast per preservare come una sorta di "vista temporanea" una copia del risultato del punto precedente. Sul primo ramo in output del multicast è stato poi selezionato il numero massimo di partite perse per ogni player_id e, successivamente, è stata fatta la join tra l'output dell'aggregazione ed il secondo ramo del multicast al fine di ottenere l'ennupla <MAX(partite_perse, player_id, nemesis_id>. Si noti come la merge join abbia richiesto che entrambi i rami dei dati in input fossero prima ordinati dal relativo operatore, anche se in verità sarebbe stato sufficiente il solo ordinamento prima del multicast.

3. Recupero dei nomi dei giocatori e delle nemesi coinvolte e salvataggio del risultato: l'ultimo step di questo flusso di dati è stato quello di recuperare il nome del giocatore e della nemesi tramite l'uso di due operatori di lookup ed infine salvare sul file "assignment1.csv" il risultato.

3.3 Assignment 2

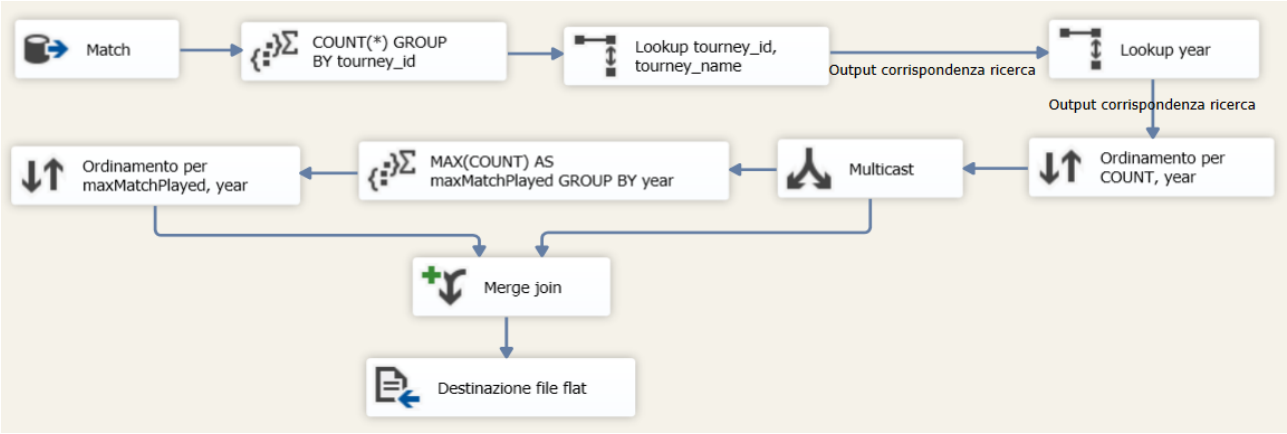


Figure 5: For each year, the name of the tournament with more matches played.

Anche in questo assignment si è proceduto dividendo l'esercizio in più sotto-problemi:

1. Recupero dei dati necessari e prima aggregazione: è stato necessario, oltre che caricare con il relativo operatore la tabella Match, recuperare dalla tabella Tournament e Date gli attributi tourney_id, tourney_name e year tramite l'operatore di lookup. Si è proceduto poi al calcolo del numero di match giocati per ogni torneo per ogni anno. Successivamente si è osservato come l'attributo tourney_id della tabella Match identificasse univocamente un torneo nella tabella Tournament e, di conseguenza, anche l'anno in cui si è svolto tale torneo. Quindi si è deciso di spostare l'operatore di aggregazione subito dopo il caricamento della tabella Match.
2. Selezione del tournment con più match giocati per ogni anno: la strategia adottata è analoga a quella vista nel punto 2 dell'assignment precedente. È stata salvata una copia dei dati in output dal punto precedente per poi sfruttarla con l'operatore merge join per recuperare i dati del torneo una volta selezionato il massimo numero di match giocati per ogni anno.
3. Salvataggio del risultato: infine si è salvato il risultato del punto precedente nel file "assignment2.csv" con il relativo operatore.

4 Parte 3

4.1 Assignment 0 - Cubo

Il cubo è stato creato a partire dal database SQL definendo le gerarchie per le dimensioni Date e Geography:

- *DayMonthQuarterYear* in Date: Date Id (Day) → Month → Quarter → Year
- *CountryContinent* in Geography: Country → Continent sia per Winner che per Loser.

Successivamente, si sono impostate le misure winner rank, winner rank points, loser rank, loser rank points necessarie per gli assignment successivi. Si sono aggiunte le metriche avgWinnerRankPoints, avgLoserRankPoints e avgMinutes definite come la media per numero di match. Infine, si sono incluse nel cubo alcune delle restanti misure della tabella Match e Tournament in modo da poterle utilizzare per realizzare la dashboard dell'assignment 5.

4.2 Assignment 1,2,3 - Query MDX

1. *Show the player that lost the most matches for each country*: utilizzando l'operatore $GENERATE()$ in combinazione con $TOPCOUNT()$ si è cercato per ogni paese ([Loser - Country].[Country Name].[Country Name]) il giocatore con il valore di [Measures].[Match count] più alto.

2. *For each tournament, show the loser with the lowest total loser rank points*: usando l'operatore $GENERATE()$ in combinazione con $BOTTOMCOUNT()$ si è cercato in ogni torneo ([Tournament].[Tourney Pk].[Tourney Pk]) il giocatore con il minor valore di [Measures].[Loser Rank Points] all'interno del torneo stesso.

3. emphFor each tournament, show the loser with the highest ratio between his loser rank points and the average winner rank points of that tournament: anche in questo caso, usando l'operatore $GENERATE()$ in combinazione con $TOPCOUNT()$ si è cercato in ogni torneo ([Tournament].[Tourney Pk].[Tourney Pk]) il giocatore con il valore del ratio più alto. Il ratio è stato calcolato grazie ad un nuovo membro dividendo il Loser Rank Points di ogni giocatore con la media di winner rank points del torneo considerato.

4.3 Assignment 4 - Geographical distribution of winner and loser rank points

Per visualizzare la distribuzioni di winner rank points e loser rank points si sono utilizzate due coppie di componenti mappa e istogramma a colonne raggruppate rispettivamente per winner e loser. Per impostare la località della mappa si è utilizzato la gerarchia *CountryContinent* (in un caso quella relativa al winner e al loser), per la dimensione si utilizzato la misura rank points in maniera analoga. Non è stato possibile rappresentare con un'unica coppia mappa-istogramma sia i dati del winner che del loser in quanto sarebbe stato necessario rielaborare l'intero design del database e del cubo per poter rappresentare come tabella dei fatti il match per singolo giocatore (allo stato attuale il fatto è relativo al singolo match e quindi a due giocatori, i quali potrebbero avere nazione di origine diversa) Gli istogrammi mettono a confronto i valori di winner rank points e loser rank points per ogni continente e per ogni paese.

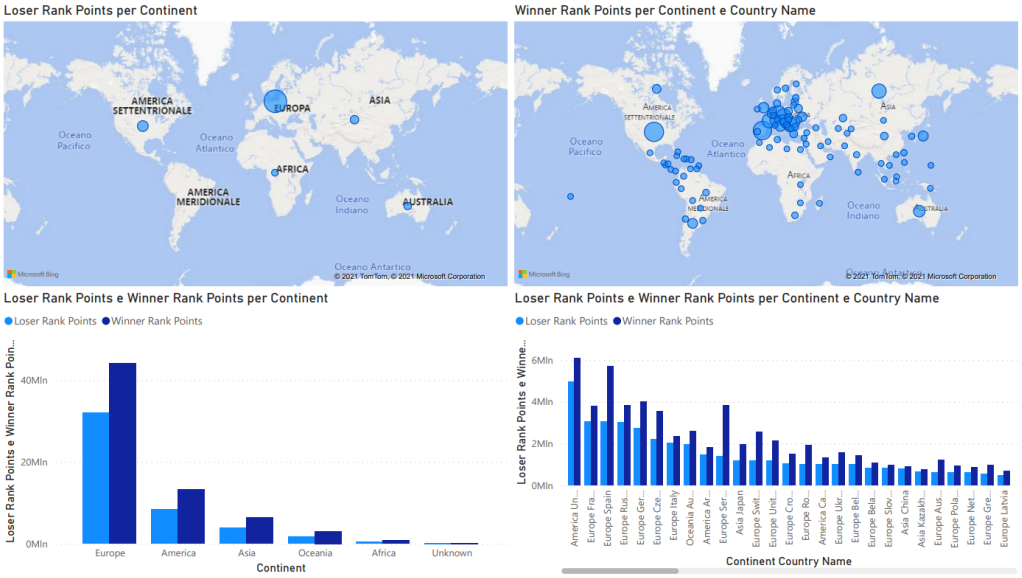


Figure 6: Distribuzione geografica di *winnerRankPoints* e *loserRankPoints*

Per meglio comprendere la distribuzione dei due attributi si è ritenuto necessario visualizzare i valori medi, in quanto, se in un paese si sono giocati più match allora la somma totale dei punteggi sarà naturalmente più grande. L'unica differenza a livello di realizzazione di questa seconda dashboard sta nell'utilizzo delle metriche *avgWinnerRankPoints* e *avgLoserRankPoints* per la dimensione.

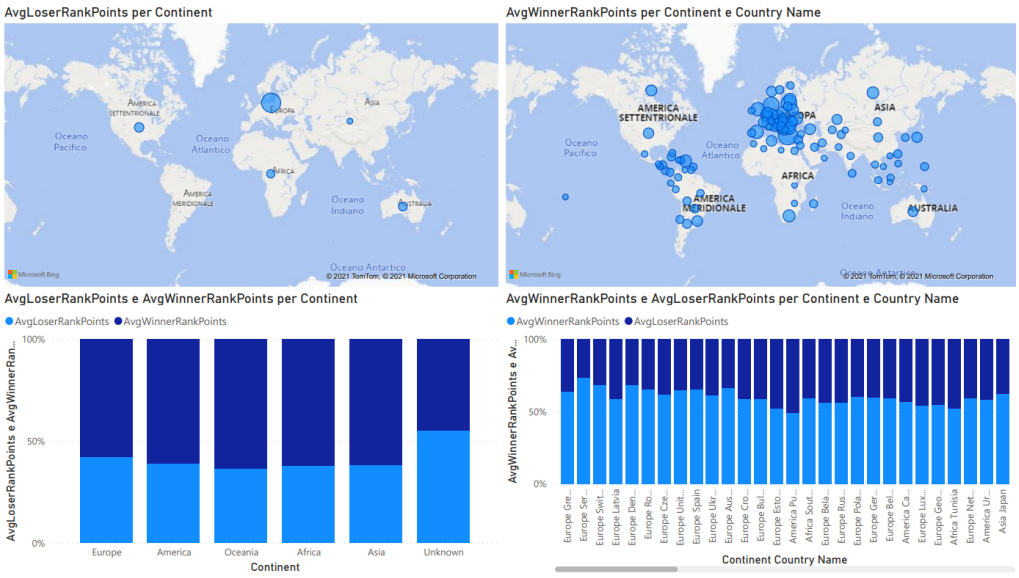


Figure 7: Distribuzione geografica di *avgWinnerRankPoints* e *avgLoserRankPoints*

4.4 Assignment 5 - Analisi Tornei e numero di Match

Dato che le dashboard precedenti permettono di esplorare la dimensione Geography mediante la gerarchia CountryContinent, per realizzare quest'ultima dashboard si è deciso di utilizzare le dimensioni restanti: Tournament, Date e Match. La Dashboard realizzata permette di visualizzare interattivamente, mediante un pannello di selezione, la distribuzione dei tornei e del numero di match giocati nel tempo. Le misure utilizzate sono Tourney Spectators, Tourney Revenue, avgWinnerRankPoints e Match Count. La Dashboard si compone di:

1. Pannello di selezione: permette di filtrare i dati per Year, Quarter, Month (Date), per Best Of, Score (Match) e per Tourney Name (Tournament).

2. Grafico a dispersione Tornei: mostra i tornei sulla base di Tourney Spectators e avgWinnerRankPoints, la dimensione di ogni punto è data da Match Count. Inoltre, ogni torneo è colorato in base al suo livello.

3. Istogramma Revenue Tornei: mostra i Revenue totali dei tornei selezionati. Sull'asse x vengono mostrati i Tourney Level, ogni barra si divide in base a Surface.

4. Grafico a linee di Match Count: mostra il numero di match giocati per ogni Round nel tempo. É possibile espandere la gerarchia di Date fino al livello del giorno.

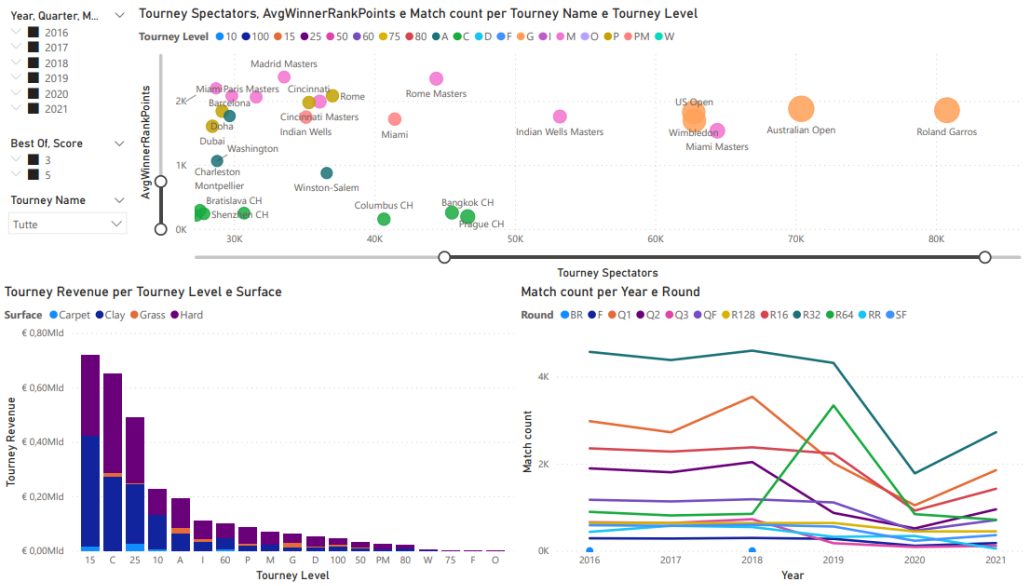


Figure 8: Analisi Tornei e numero di Match