

Συστήματα Ανάλυσης και Διαχείρισης Μεγάλων Δεδομένων – Εργασία 1^η

Χατζόπουλος Ιάσων – p3312322 – ias.chatzopoulos@aueb.gr, JasonChatzopoulos@gmail.com

[Report link](#) , [Dashboard link](#)

Ζήτημα 1^ο

Αρχικά, εισήγαγα τα δεδομένα με τις παρακάτω εντολές στον staging πίνακα GAME_DATA. Ο πίνακας GAME_DATA περιέχει όλα τα δεδομένα του dataset.

create_database.sql
<pre>drop database if exists footballDW; go create database footballDW; go use footballDW go CREATE TABLE GAME_DATA (game_id INT, game_date DATE, home_club_id INT, home_club_name VARCHAR(100), home_club_goals INT, away_club_id INT, away_club_name VARCHAR(100), away_club_goals INT, winner INT, competition_id VARCHAR(10), competition_name VARCHAR(100), country_name VARCHAR(100), player_club_id INT, player_id INT, player_name VARCHAR(100), yellow_cards INT, red_cards INT, goals INT, assists INT, minutes_played INT, date_of_birth DATE, position VARCHAR(50)); BULK INSERT GAME_DATA FROM '/assignments/assignment_1/sql/data/footballData.txt' WITH (FIRSTROW = 2, FIELDTERMINATOR = ' ', ROWTERMINATOR = '\n');</pre>

Στην συνέχεια, αφού επιβεβαίωσα ότι τα δεδομένα είχαν περαστεί σωστά στον staging πίνακα, δημιούργησα τις διαστάσεις του Star Schema και εισήγαγα σε κάθε διάσταση τα κατάλληλα δεδομένα από το GAME_DATA.

Για τις διαστάσεις του Star Schema επέλεξα τις εξής:

- Διάσταση Ημερομηνίας (DIM_DATE)
- Διάσταση Ομάδας (DIM_CLUB)
- Διάσταση Παίκτη (DIM_PLAYER)
- Διάσταση Πρωταθλήματος (DIM_COMPETITION)

Για τον πίνακα των ομάδων, επέλεξα τα δεδομένα με το distinct και την ένωση των συνόλων των κλειδιών home teams και away teams επιλέγοντας έτσι όλες τις ομάδες μία φορά. Στη συνέχεια επέλεξα τους παίκτες με το distinct player_id. Με αυτό το τρόπο θα φαίνεται αν κάποιος παίκτης έχει αλλάξει ομάδα κατά τη διάρκεια του πρωταθλήματος.

create_dimensions.sql

```
CREATE TABLE DIM_DATE (  
  date_key INT PRIMARY KEY IDENTITY(1,1),  
  game_date DATE,  
    year as YEAR(game_date) PERSISTED,  
    month as MONTH (game_date) PERSISTED  
);  
  
CREATE TABLE DIM_CLUB (  
  club_id INT PRIMARY KEY,  
  club_name VARCHAR(100)  
);  
  
CREATE TABLE DIM_PLAYER (  
  player_id INT PRIMARY KEY,  
  player_name VARCHAR(100),  
  date_of_birth DATE,  
  position VARCHAR(50)  
);  
  
CREATE TABLE DIM_COMPETITION (  
  competition_id VARCHAR(10) PRIMARY KEY,  
  competition_name VARCHAR(100),  
  country_name VARCHAR(100)  
);  
  
-- Populate dimensions  
INSERT INTO DIM_DATE (game_date)  
SELECT DISTINCT game_date FROM GAME_DATA;  
  
INSERT INTO DIM_CLUB (club_id, club_name)  
SELECT DISTINCT home_club_id, home_club_name FROM GAME_DATA  
UNION  
SELECT DISTINCT away_club_id, away_club_name FROM GAME_DATA;  
  
INSERT INTO DIM_PLAYER (player_id, player_name, date_of_birth, position)  
SELECT DISTINCT player_id, player_name, date_of_birth, position FROM GAME_DATA;  
  
INSERT INTO DIM_COMPETITION (competition_id, competition_name, country_name)  
SELECT DISTINCT competition_id, competition_name, country_name FROM GAME_DATA;
```

Το Fact Table περιέχει στα κατηγορικά δεδομένα αναφορές στις διαστάσεις που ανέφερα παραπάνω με την χρήση των foreign keys τους. Η διάσταση των ομάδων λαμβάνει υπόψη του την εμφάνιση μίας ομάδας σαν γηπεδούχο ομάδα ή φιλοξενούμενη ομάδα. Τα ποσοτικά δεδομένα του fact table περιέχουν τα συνολικά goal του παίκτη, τις κίτρινες/κόκκινες κάρτες, τα συνολικά goal της γηπεδούχου ομάδας και της φιλοξενούμενης, την ομάδα που νίκησε κλπ.

Για την εισαγωγή των δεδομένων στο fact table έκανα τα παρακάτω insert κάνοντας join τις 4 διαστάσεις.

create_fact_table.sql

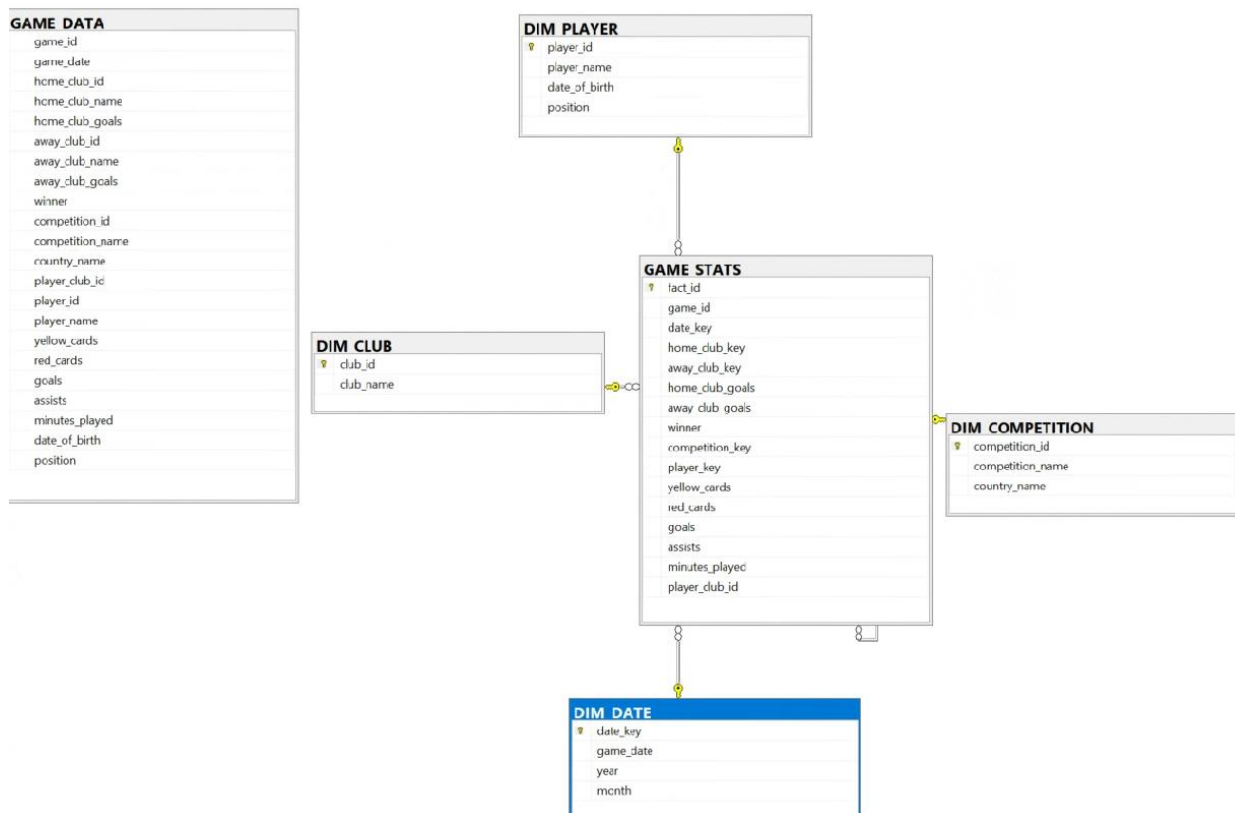
```
CREATE TABLE GAME_STATS (  
  fact_id INT PRIMARY KEY IDENTITY(1,1),  
  game_id INT,  
  date_key INT,  
  home_club_key INT,  
  away_club_key INT,  
  home_club_goals INT,  
  away_club_goals INT,  
  winner INT,  
  competition_key VARCHAR(10),  
  player_key INT,  
  yellow_cards INT,  
  red_cards INT,  
  goals INT,  
  assists INT,  
  minutes_played INT,  
  player_club_id INT,  
  FOREIGN KEY (date_key) REFERENCES DIM_DATE(date_key),  
  FOREIGN KEY (home_club_key) REFERENCES DIM_CLUB(club_id),  
  FOREIGN KEY (away_club_key) REFERENCES DIM_CLUB(club_id),  
  FOREIGN KEY (competition_key) REFERENCES DIM_COMPETITION(competition_id),  
  FOREIGN KEY (player_key) REFERENCES DIM_PLAYER(player_id)  
);  
  
INSERT INTO GAME_STATS (  
  game_id,  
  date_key,  
  home_club_key,  
  away_club_key,  
  home_club_goals,  
  away_club_goals,  
  winner,  
  competition_key,  
  player_key,  
  yellow_cards,  
  red_cards,  
  goals,  
  assists,  
  minutes_played,  
  player_club_id  
) SELECT  
  gd.game_id,  
  d.date_key,  
  hc.club_id AS home_club_key,
```

```

ac.club_id AS away_club_key,
gd.home_club_goals,
gd.away_club_goals,
gd.winner,
c.competition_id AS competition_key,
p.player_id AS player_key,
gd.yellow_cards,
gd.red_cards,
gd.goals,
gd.assists,
gd.minutes_played,
gd.player_club_id
FROM
    GAME_DATA gd
JOIN
    DIM_DATE d ON gd.game_date = d.game_date
JOIN
    DIM_CLUB hc ON gd.home_club_id = hc.club_id
JOIN
    DIM_CLUB ac ON gd.away_club_id = ac.club_id
JOIN
    DIM_COMPETITION c ON gd.competition_id = c.competition_id
JOIN
    DIM_PLAYER p ON gd.player_id = p.player_id;

```

Διαγραμματική απεικόνιση των πινάκων:



Ζήτημα 2°

2.1 Εμφάνιση όλων των αγώνων για το ελληνικό πρωτάθλημα

Για το ζήτημα 2.1 πραγματοποιήσα το παρακάτω query, επιλέγοντας την ημερομηνία, το όνομα της γηπεδούχου ομάδας, της φιλοξενούμενης ομάδας και τα συνολικά goal των δύο ομάδων από το Fact Table για κάθε εγγραφή όπου το όνομα του πρωταθλήματος αντιστοιχεί στο ελληνικό πρωτάθλημα (super-league-1).

2.1.sql

```
SELECT
    d.game_date,
    hc.club_name AS Home_Team,
    ac.club_name AS Away_Team,
    gs.home_club_goals,
    gs.away_club_goals
FROM
    GAME_STATS gs
JOIN
    DIM_DATE d ON gs.date_key = d.date_key
JOIN
    DIM_CLUB hc ON gs.home_club_key = hc.club_id
JOIN
    DIM_CLUB ac ON gs.away_club_key = ac.club_id
JOIN
    DIM_COMPETITION c ON gs.competition_key = c.competition_id
WHERE
    c.competition_name = 'super-league-1'
ORDER BY
    d.game_date ASC;
```

2.2 Εμφάνιση συνολικών goal, κίτρινων και κόκκινων καρτών

Για το ζήτημα 2.2 αθροίζω τα συνολικά goal και το πλήθος των καρτών για κάθε πρωτάθλημα και ομάδα με τη χρήση της συνάρτησης SUM για τις ποσοτικές τιμές και join για τις διαστάσεις των ομάδων και πρωταθλημάτων και group by με το πρωτάθλημα και τις ομάδες που παίζουν σε αυτό το πρωτάθλημα.

2.2.sql

```
SELECT
    c.competition_name AS 'League Name',
    cl.club_name AS 'Team Name',
    SUM(gs.goals) AS 'Total Goals',
    SUM(gs.yellow_cards) AS 'Total Yellow Cards',
    SUM(gs.red_cards) AS 'Total Red Cards'
FROM
    GAME_STATS gs
JOIN
    DIM_COMPETITION c ON gs.competition_key = c.competition_id
JOIN
    DIM_CLUB cl ON gs.player_club_id = cl.club_id
GROUP BY
    c.competition_name, cl.club_name
```

```
ORDER BY
c.competition_name, cl.club_name;
```

2.3 Συνολικές νίκες γηπεδούχων ομάδων, νίκες φιλοξενούμενων ομάδων και ισοπαλίες

Για το ζήτημα 2.3 επέλεξα τη χρήση μίας όψης καθώς με βοήθησε στην αναπαράσταση των δεδομένων για τα ερωτήματα του 3^{ου} ζητήματος. Στο select ψάχνω όλους τους αγώνες και κάνω χρήση της συνάρτησης count για να μετρήσω πόσες φορές έχει νικήσει/χάσει/βγει ισοπαλία η κάθε ομάδα, στη συνέχεια κάνω JOIN της διάστασης των πρωταθλημάτων και κάνω group by τα δεδομένα με βάση το όνομα της ομάδας.

2.3.sql

```
CREATE VIEW TOTAL_WIN_STATS AS
WITH WIN_STATS AS (
    SELECT
        c.competition_name AS 'League_Name',
        COUNT(DISTINCT CASE WHEN gs.winner = 1 THEN gs.game_id END) AS 'Home_Team_Wins',
        COUNT(DISTINCT CASE WHEN gs.winner = 2 THEN gs.game_id END) AS 'Away_Team_Wins',
        COUNT(DISTINCT CASE WHEN gs.winner = 0 THEN gs.game_id END) AS 'Draws'
    FROM
        GAME_STATS gs
    JOIN
        DIM_COMPETITION c ON gs.competition_key = c.competition_id
    GROUP BY
        c.competition_name
) SELECT
    g.League_Name AS 'League Name',
    g.Home_Team_Wins as 'Home Team Wins',
    g.Away_Team_Wins as 'Away Team Wins',
    g.DRAWS as 'Draws'
FROM
    WIN_STATS g;

SELECT * FROM TOTAL_WIN_STATS ORDER BY 'League Name';
```

Εναλλακτική λύση χωρίς τη χρήση όψεων, η λογική των queries παραμένει ίδια με πριν:

2.3.sql

```
SELECT
    c.competition_name AS 'League Name',
    COUNT(DISTINCT CASE WHEN gs.winner = 1 THEN gs.game_id END) AS 'Home Team Wins',
    COUNT(DISTINCT CASE WHEN gs.winner = 2 THEN gs.game_id END) AS 'Away Team Wins',
    COUNT(DISTINCT CASE WHEN gs.winner = 0 THEN gs.game_id END) AS 'Draws'
FROM
    GAME_STATS gs
JOIN
    DIM_COMPETITION c ON gs.competition_key = c.competition_id
GROUP BY
    c.competition_name
ORDER BY
    c.competition_name;
```

2.4 Συνολικός αριθμός goal που πέτυχε ο πρώτος σκόρερ της ομάδας για το ελληνικό πρωτάθλημα

Για το συγκεκριμένο ζήτημα δημιούργησα μία όψη προκειμένου να συγκεντρώσω τους top scorers. Οι top scorers του ελληνικού πρωταθλήματος, αφού χωριστούν σε partitions ανά ομάδα,

ταξινομούνται βάση των συνολικών goals εντός της ομάδας τους και η θέση τους έπειτα από την ταξινόμηση αποθηκεύεται στη στήλη Rank. Έπειτα μπορώ να κάνω queries σαν το παρακάτω όπου επιλέγω τους παίκτες που στην κατάταξη τους έχουν τον αριθμό 1.

2.4.sql

```
CREATE VIEW TOP_SCORRERS_GR AS
WITH GOAL_SCORRERS AS (
    SELECT
        p.player_name,
        cl.club_name,
        SUM(gs.goals) AS Total_Goals,
        RANK() OVER (PARTITION BY cl.club_name ORDER BY SUM(gs.goals) DESC) AS Rank
    FROM
        GAME_STATS gs
    JOIN
        DIM_COMPETITION c ON gs.competition_key = c.competition_id and
        c.competition_name = 'super-league-1'
    JOIN
        DIM_CLUB cl ON gs.player_club_id = cl.club_id
    JOIN
        DIM_PLAYER p ON gs.player_key = p.player_id
    GROUP BY
        p.player_name, cl.club_name
)
SELECT
    g.club_name AS 'Team Name',
    g.player_name AS 'Top Scorer',
    g.Total_Goals
FROM
    GOAL_SCORRERS g
WHERE
    g.Rank = 1;
```

2.5 Συνολικός αριθμός goal που σημειώθηκαν στο πρωτάθλημα της Αγγλίας ανά ομάδα, έτος και μήνα

Για το ζήτημα 2.5 και ομοίως με τα προηγούμενα 2 ζητήματα, επέλεξα την χρήση μίας όψης προκειμένου να οπτικοποιήσω με ευκολία τα δεδομένα με το power-bi. Παρακάτω βρίσκεται και η ισοδύναμη λύση χωρίς την χρήση όψης.

2.5.sql

```
CREATE VIEW PREMIER_LEAGUE_GOALS_VIEW AS
WITH PREMIER_LEAGUE_GOALS AS (
    SELECT
        cl.club_name AS 'Team',
        d.year AS 'Year',
        d.month AS 'Month',
        SUM(gs.goals) AS 'Total Goals'
    FROM
        GAME_STATS gs
    JOIN
        DIM_DATE d ON gs.date_key = d.date_key
    JOIN
        DIM_COMPETITION c ON gs.competition_key = c.competition_id
```

```

        JOIN
            DIM_CLUB cl ON gs.player_club_id = cl.club_id
        WHERE
            c.competition_name = 'premier-league'
        GROUP BY CUBE(
            cl.club_name,
            d.year,
            d.month)
    )
SELECT
    plg.Team,
    plg.Year,
    plg.Month,
    plg.[Total Goals]
FROM
    PREMIER_LEAGUE_GOALS plg;

```

-- εναλλακτική λύση χωρίς την χρήση όψης

```

SELECT
    cl.club_name AS 'Team',
    d.year AS 'Year',
    d.month AS 'Month',
    SUM(gs.goals) AS 'Total Goals'
FROM
    GAME_STATS gs
JOIN
    DIM_DATE d ON gs.date_key = d.date_key
JOIN
    DIM_COMPETITION c ON gs.competition_key = c.competition_id
JOIN
    DIM_CLUB cl ON gs.player_club_id = cl.club_id
WHERE
    c.competition_name = 'premier-league'
GROUP BY CUBE(
    cl.club_name,
    d.year,
    d.month)
ORDER BY
    cl.club_name, d.year, d.month;

```

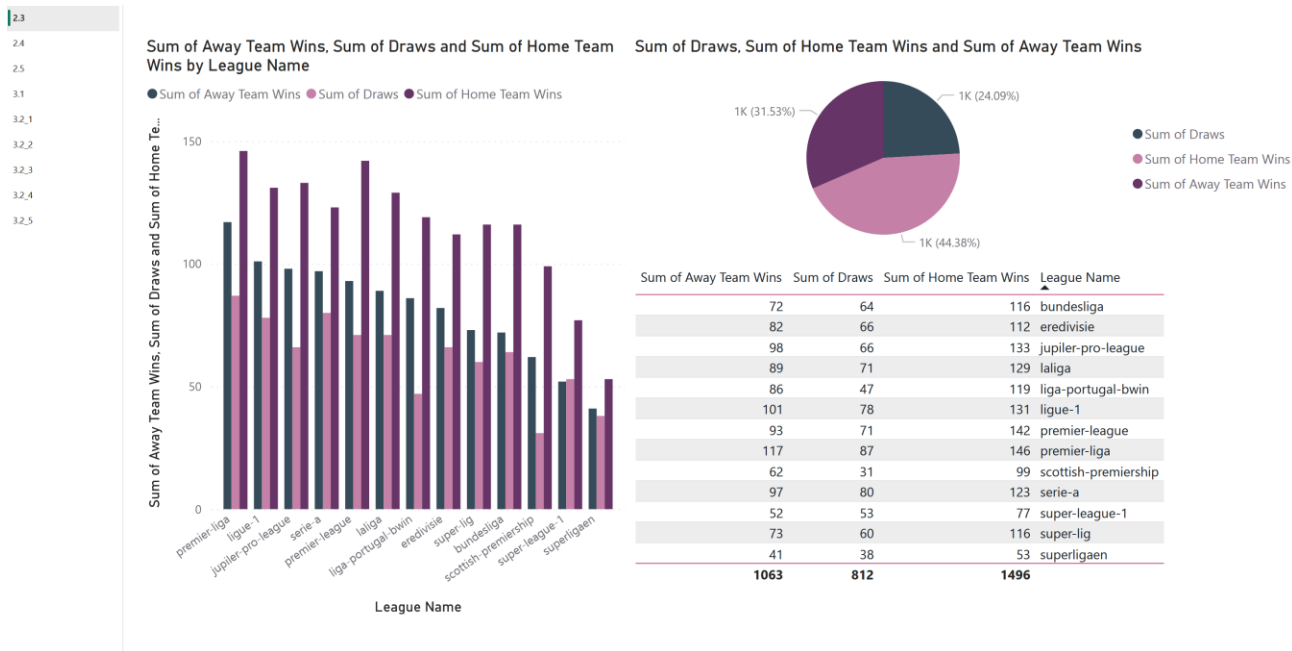
Ζήτημα 3^ο

3.1 Power BI

Το report για το power BI βρίσκεται εδώ: [Report link](#). Τα γραφήματα του ζητήματος 3.1 βρίσκονται στα φύλλα 2.3, 2.4 και 2.5.

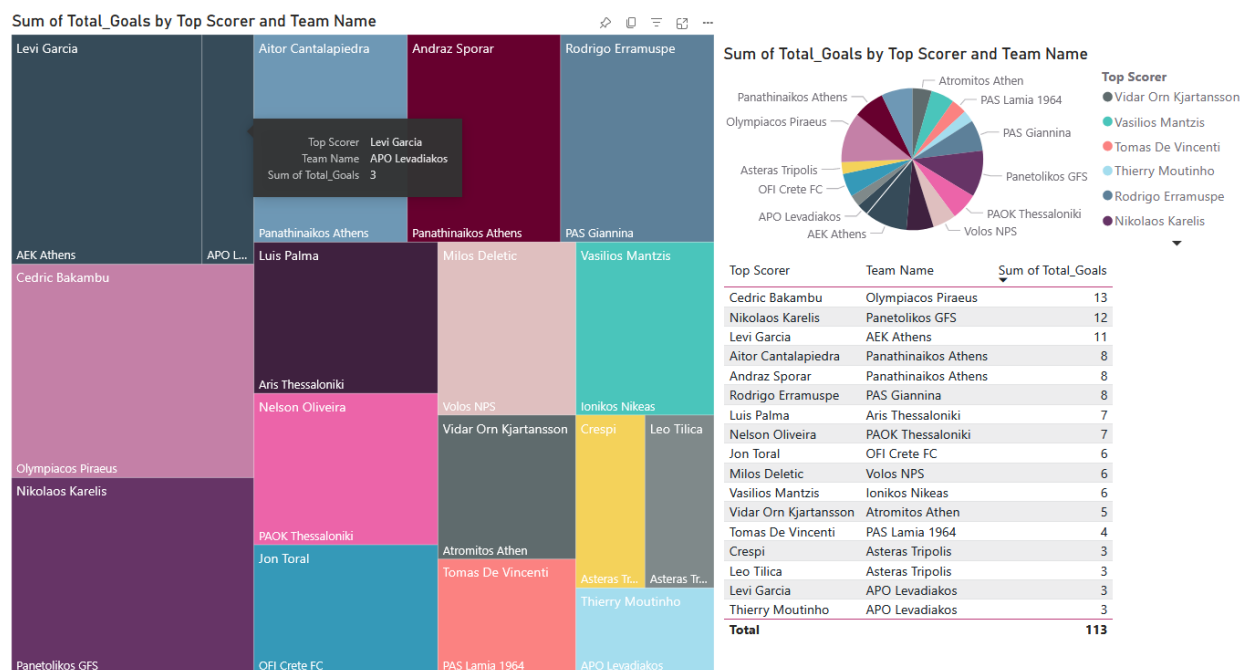
3.1.1 Γραφήματα για τα ερωτήματα 2.3 και 2.4

Για την παραγωγή των γραφημάτων επέλεξα ως είσοδο δεδομένων τις όψεις του ζητήματος [2.3](#), φτιάχνοντας τα παρακάτω γραφήματα. Επιπλέον τα γραφήματα είναι συνδεδεμένα οπότε αν ο χρήστης επιλέξει κάποιο τμήμα του γραφήματος, θα γίνουν highlight οι συνδεδεμένες τιμές.



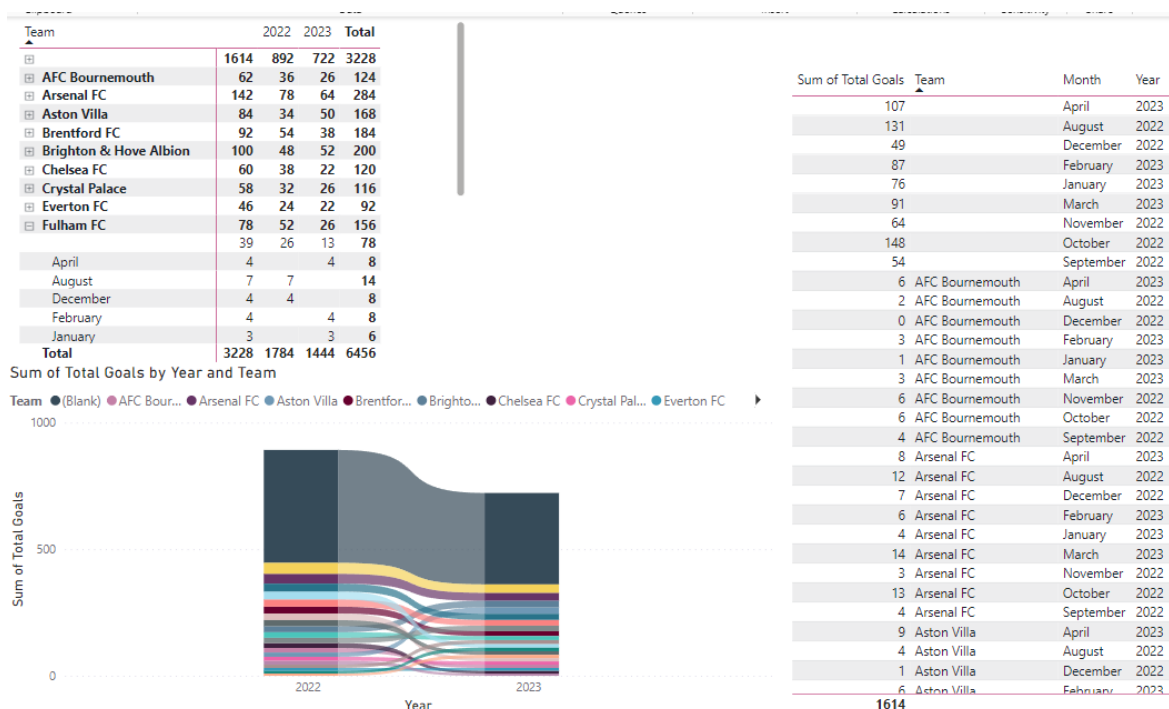
Αριστερά βρίσκονται οι συνολικές νίκες ήττες και ισοπαλίες ανά ομάδα και αναπαρίστανται σε ένα ιστόγραμμα. Δεξιά βρίσκονται συγκεντρωμένες σε ένα pie chart οι νίκες, ήττες και ισοπαλίες για κάθε ομάδα, ενώ κάτω δεξιά βρίσκεται ο αλφαβητικά ταξινομημένος πίνακας με τις αναλυτικές τιμές ανά πρωτάθλημα.

Για τα γραφήματα του 2.4 επέλεξα την όψη που ανέφερα στο 2.4. Σε αυτή την περίπτωση επέλεξα το Tree Map χρησιμοποιώντας για είσοδο δεδομένων τις στήλες της όψης. Τέλος έχω χρησιμοποιήσει ένα pie chart για να συγκρίνω τα συνολικά goal ανά top scorer καθώς και ένα πινακάκι με όλες το όνομα του παίχτη, της ομάδας και των συνολικών goal.



3.1.2 Γραφήματα για το ερώτημα 2.5

Για την αναπαράσταση του κύβου, χρησιμοποίησα τα παρακάτω γραφήματα με πηγή την όψη που περιέγραψα στο 2.5. Χρησιμοποίησα ένα matrix το οποίο περιέχει στη πρώτη στήλη το όνομα της ομάδας σε αλφαβητική σειρά και στις υπόλοιπες στήλες είναι τα αθροίσματα των goal ανά χρονική περίοδο (έτος, μήνα). Τα κελιά που είναι κενά προέρχονται από το group by cube και οι τιμές που είναι κενές αντικατοπτρίζουν τα aggregation για τα group by. Το matrix συγκεντρώνει τα goal για τις χρονικές περιόδους ανά ομάδα και δίνει την δυνατότητα στον χρήστη να επιλέξει όσους μήνες θέλει για κάθε ομάδα. Επιπλέον χρησιμοποίησα ένα ribbon chart για να αναπαραστήσω τα συνολικά goal ανά ομάδα, μήνα και χρονιά καθώς και ένα πινακάκι που περιέχει όλες τις αναλυτικές τιμές της όψης και όλα αλληλοεπιδρούν μεταξύ τους επιτρέποντας στο χρήστη να βρει και να αναπαραστήσει πληροφορίες με μεγάλη ευκολία.

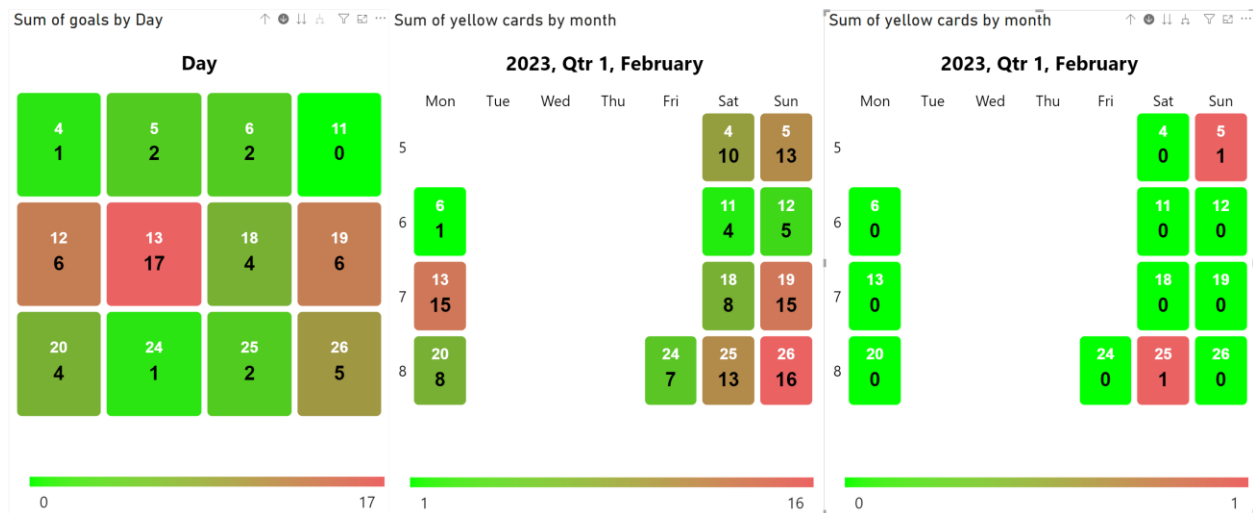
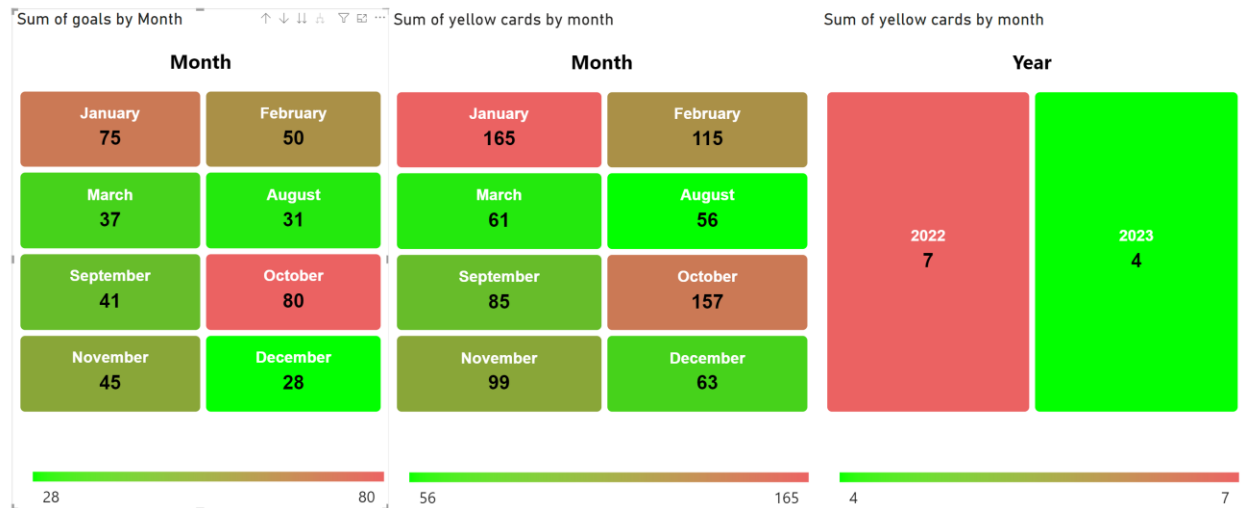


3.2 Εξερεύνηση δεδομένων

Για το επερώτημα 3.2 επέλεξα το ελληνικό πρωτάθλημα για να εξάγω και να οπτικοποιήσω διάφορα στατιστικά ως προς τις αποδόσεις και επιδόσεις των παικτών.

3.2_1

Για τα παρακάτω γραφήματα κατέβασα ένα εξωτερικό visualizer που εξειδικεύεται στην αναπαράσταση χρονικών δεδομένων με έξυπνους τρόπους σαν ημερολόγιο.. Το Calendar (by Datanaui) επιτρέπει στο χρήστη να κάνει drill down από έτος σε τρίμηνο, μήνα η και ημέρα διεξαγωγής του αγώνα. Με την επιλογή κάποιας ημερομηνίας, το εργαλείο έχει πλήρες integration με τα υπόλοιπα εργαλεία του Power BI.



Τα παραπάνω δεδομένα τα έχω λάβει απευθείας από το fact table και ισχύουν για όλα το ελληνικό πρωτάθλημα. Το φιλτράρισμα έγινε χρησιμοποιώντας τα filter για τα visuals και δημιουργήσα σχέσεις μεταξύ του fact table και των άλλων διαστάσεων.

country_name
is Greece

Filter type ⓘ

Basic filtering ▾

Search

☒ Greece 1

3.2_2

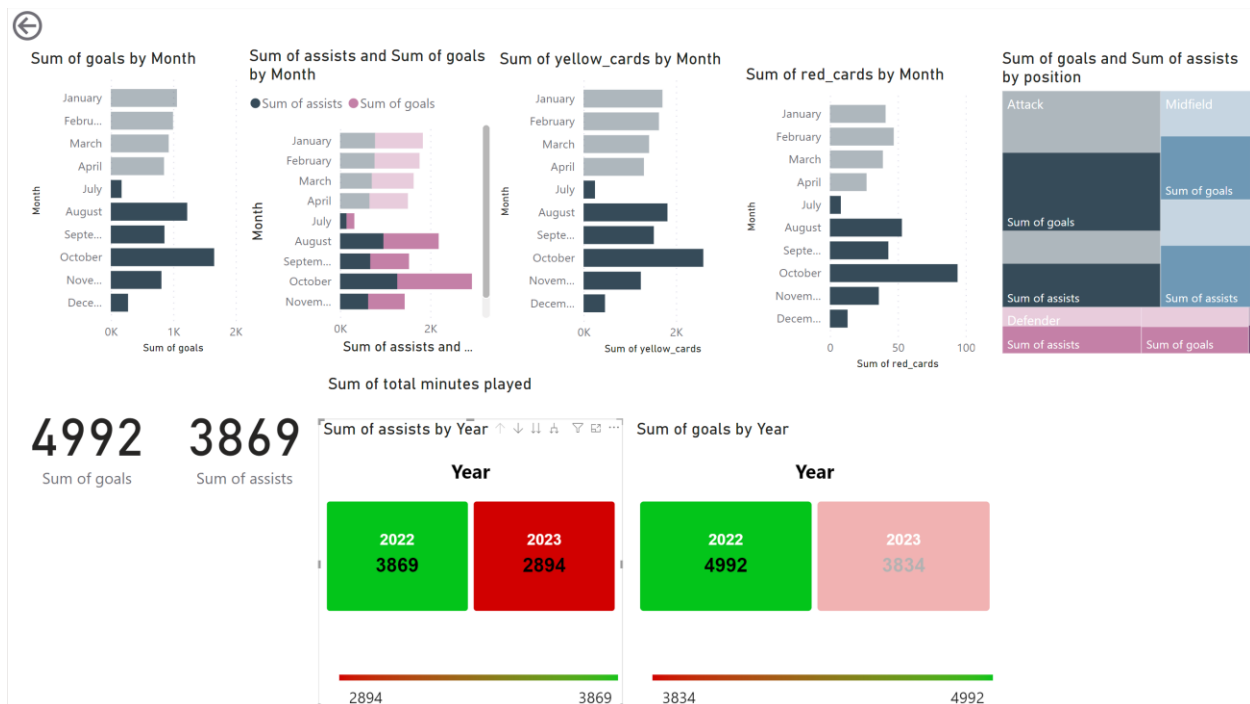
Στη συνέχεια δημιούργησα το παρακάτω φύλλο εξερευνώντας διάφορα ποσοτικά χαρ/κά συνολικά από όλους τους αγώνες από το fact table.

Στην πρώτη σειρά και από τα αριστερά προς τα δεξιά απεικονίζω τα εξής:

1. Συνολικά goal ανά μήνα
2. Συνολικά goal και assist ανά μήνα
3. Συνολικές κίτρινες κάρτες
4. Συνολικές κόκκινες κάρτες
5. Συνολικά goal και assist ανά θέση

Στη δεύτερη σειρά και από τα αριστερά προς τα δεξιά απεικονίζω τα εξής:

1. Πινακάκι με τα συνολικά goal
2. Πινακάκι με τις συνολικές assist και αλληλοεπιδρούν με τα άλλα γραφήματα.
3. Ημερολόγιο για τις assist που δίνει την δυνατότητα στο χρήστη να κάνει drill down σε υποκατηγορίες (τετράμηνα, μήνες, βδομάδες και ημέρες)
4. Ημερολόγιο για τα goal με παρόμοιες δυνατότητες με το 3.



Για παράδειγμα, αν ο χρήστης κάνει drill down στις assist ανά μέρα θα δει όλες τις τιμές να αλλάζουν όπως παρακάτω:



Τα δεδομένα προέρχονται από το fact table με τις κατάλληλες σχέσεις με τις διαστάσεις να έχουν ληφθεί υπόψιν.

3.2_3

Σε αυτό το σημείο ήθελα να δω τι στατιστικά αποδόσεων μπορώ να εξάγω από το dataset. Έτσι, δημιούργησα την παρακάτω όψη προκειμένου να εξερευνήσω περεταίρω τα ποσοτικά δεδομένα του dataset για το ελληνικό πρωτάθλημα.

Έτσι, στην όψη PLAYER_STATISTICS_VIEW υπολογίζω κάποια στατιστικά κατά τη δημιουργία της όψης όπως τα goal/minute, goal participation και συνολική συμμετοχή στο score του αγώνα λαμβάνοντας υπόψη τυχόν διαιρέσεις με το 0, τα συνολικά goal, assists συνολικά. Επίσης με το παρακάτω script λαμβάνω υπόψιν και το γεγονός πως ένας παίχτης μπορεί να ανήκει στην γηπεδούχο ή την φιλοξενούμενη ομάδα κατά τον υπολογισμό των συνολικών goal.

3.2.sql

```
CREATE VIEW PLAYER_STATISTICS_VIEW AS

WITH PLAYER_STATS AS (

    SELECT

        p.player_name AS 'Player_Name',

        SUM(gs.goals) AS 'Player_Goals',

        SUM(gs.minutes_played) AS 'Minutes_Played',

        CASE

            WHEN SUM(gs.minutes_played) = 0 THEN 0
```

```

ELSE CAST(SUM(gs.goals) AS FLOAT) / SUM(gs.minutes_played)
END AS 'Goals_Per_Minute',
SUM(
CASE
WHEN
gs.home_club_key = cl.club_id
THEN
gs.assists
WHEN
gs.away_club_key = cl.club_id
THEN
gs.assists
ELSE
0
END) AS 'Player_Assists',
SUM(
CASE
WHEN
cl.club_id = gs.home_club_key
THEN
1
WHEN
cl.club_id = gs.away_club_key
THEN
1
ELSE
0
END) AS 'Games_Played',
(CAST(
SUM(
CASE
WHEN

```

```

                                cl.club_id = gs.home_club_key
                                THEN
                                gs.goals
                                WHEN
                                cl.club_id = gs.away_club_key
                                THEN
                                gs.goals
                                ELSE
                                0
                                END)AS FLOAT) / NULLIF(
SUM(
CASE
WHEN
cl.club_id = gs.home_club_key
THEN
gs.home_club_goals
WHEN
cl.club_id = gs.away_club_key
THEN
gs.away_club_goals
ELSE
0
END), 0)) AS 'Goal_Participation',
(CAST((
SUM(
CASE
WHEN
cl.club_id = gs.home_club_key
THEN
gs.goals
WHEN
cl.club_id = gs.away_club_key

```

```

                                THEN
                                    gs.goals
                                ELSE
                                    0
                                END) +
SUM(
    CASE
        WHEN
            cl.club_id = gs.home_club_key
        THEN
            gs.assists
        WHEN
            cl.club_id = gs.away_club_key
        THEN
            gs.assists
        ELSE
            0
        END)) AS FLOAT) / NULLIF(
SUM(
    CASE
        WHEN
            cl.club_id = gs.home_club_key
        THEN
            gs.home_club_goals
        WHEN
            cl.club_id = gs.away_club_key
        THEN
            gs.away_club_goals
        ELSE
            0
        END), 0)) AS 'Game_Participation',
SUM(gs.yellow_cards) AS 'Yellow_Cards',

```



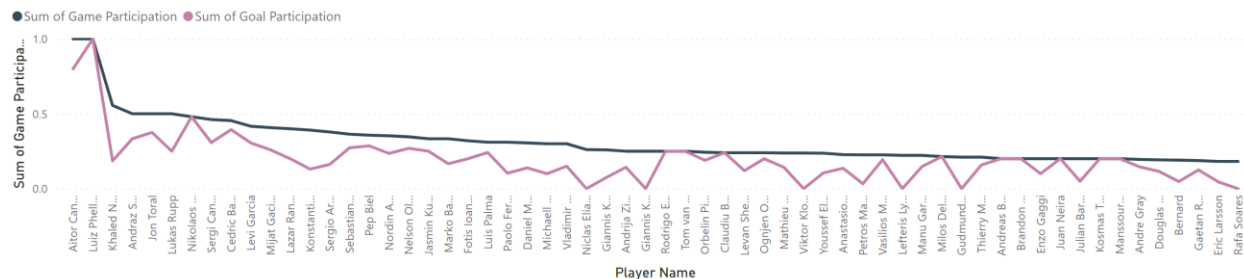
```

SUM(gs.red_cards) AS 'Red_Cards',
SUM(gs.home_club_goals) as 'home',
SUM(gs.away_club_goals) as 'away'
FROM
    GAME_STATS gs
JOIN
    DIM_COMPETITION c ON gs.competition_key = c.competition_id and country_name = 'Greece'
JOIN
    DIM_PLAYER p ON gs.player_key = p.player_id
JOIN
    DIM_CLUB cl ON gs.home_club_key = cl.club_id
GROUP BY
    p.player_name
)
SELECT
    ps.Player_Name AS 'Player Name',
    ps.Player_Goals as 'Player Goals',
    ps.Minutes_Played as 'Minutes Played',
    ps.Goals_Per_Minute as 'Goals per Minute',
    ps.Yellow_Cards as 'Yellow Cards',
    ps.Red_Cards as 'Red Cards',
    ps.Goal_Participation as 'Goal Participation',
    ps.Game_Participation as 'Game Participation'
FROM
    PLAYER_STATS ps;

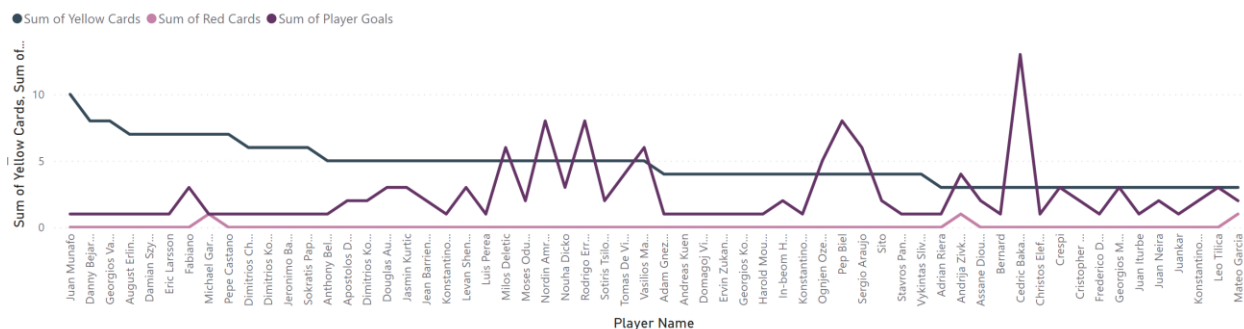
```

Συνεπώς, κατάφερα να αναπαραστήσω την επίδοση των παιχτών με τα παρακάτω γραφήματα. Στο πρώτο γράφημα φαίνεται το goal participation και συνολικό participation στα goal του αγώνα για τους παίκτες για τους οποίους τα participation είναι μεγαλύτερα του 0. Αυτό το πέτυχα με την χρήση φίλτρων. Στο δεύτερο γράφημα απεικονίζονται οι συνολικές κίτρινες και κόκκινες κάρτες, που έχει πάρει ένας παίκτης με γκρι και ροζ αντίστοιχα, ενώ με το μωβ φαίνονται τα συνολικά goal.

Sum of Game Participation and Sum of Goal Participation by Player Name

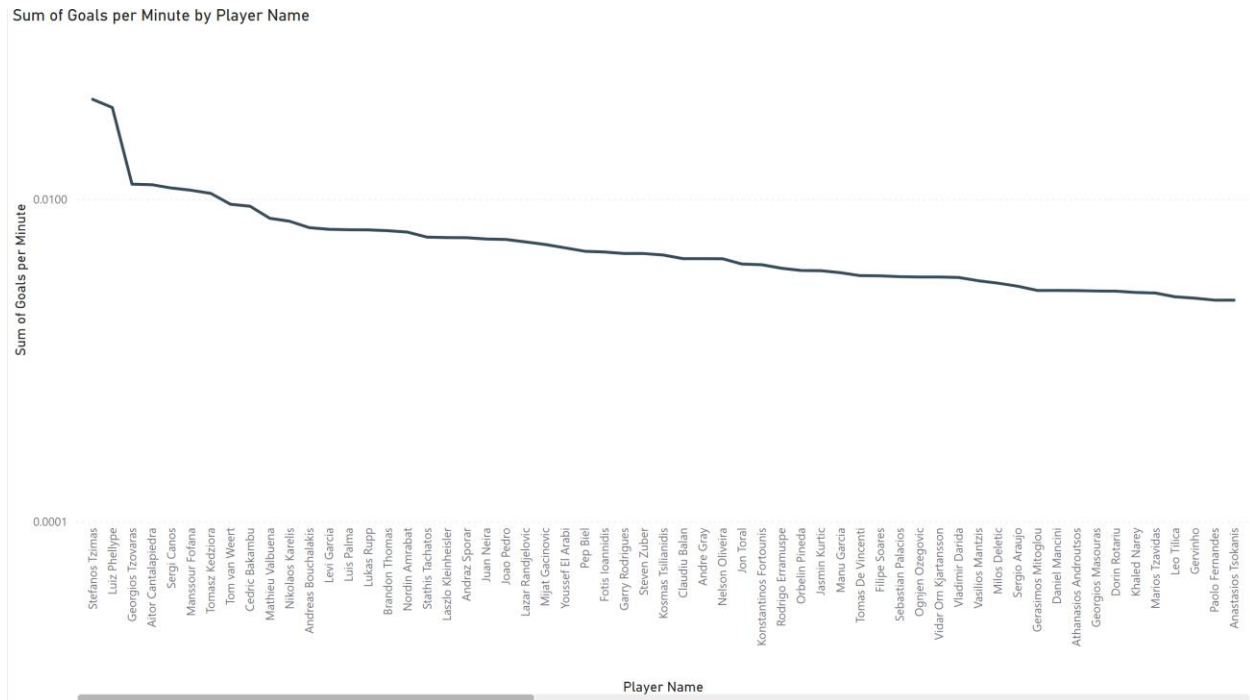


Sum of Yellow, Red Cards and Goals by Player Name



3.2_4

Εδώ αξιοποίησα τα average στατιστικά που υπολόγισα στην παραπάνω όψη. Έτσι στο παρακάτω γράφημα απεικονίζεται το goal/minute στατιστικό για τους παίκτες σε λογαριθμική κλίμακα. Την λογαριθμική κλίμακα την ενεργοποίησα αφού έβαλα φίλτρα ώστε να εμφανιστούν μόνο οι παίκτες με συνολικά γκολ/λεπτό > 0.0 .

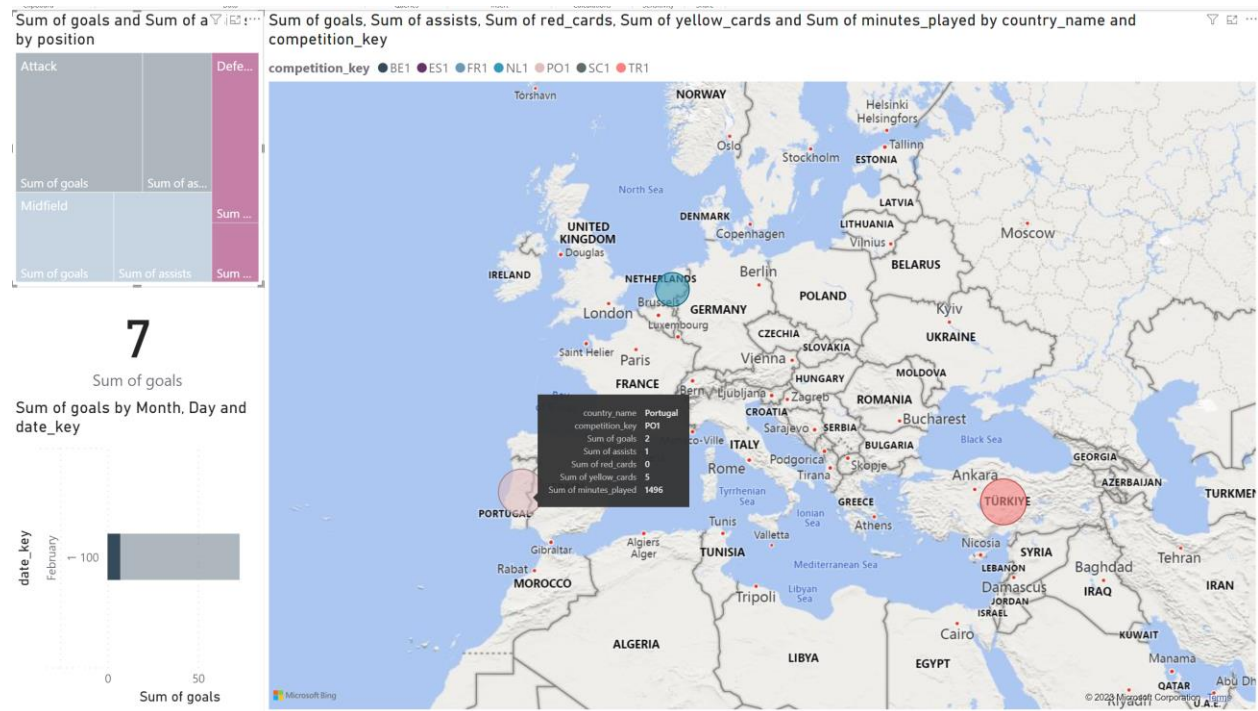


3.2_5

Σε αυτό το σημείο χρησιμοποίησα τη λειτουργία του map visualizer και συνέδεσα τις χώρες που διεξάχθηκαν με βάση το όνομα της χώρας που έχει δηλωθεί στο τα πρωταθλήματα και άθροισα τα ποσοτικά χαρ/κά όπως τα goal, assist, total play time κλπ. Στο πλάι χρησιμοποίησα κάποια ήδη υπάρχοντα visualizers για καλύτερη οπτικοποίηση των τιμών.



Έτσι μπορούμε να κάνουμε έξυπνα drill downs και να εξαγάγουμε διάφορα στατιστικά όπως το πόσα goal έβαλαν αμυντικοί παίκτες μία συγκεκριμένη μέρα.



Power BI Dashboard

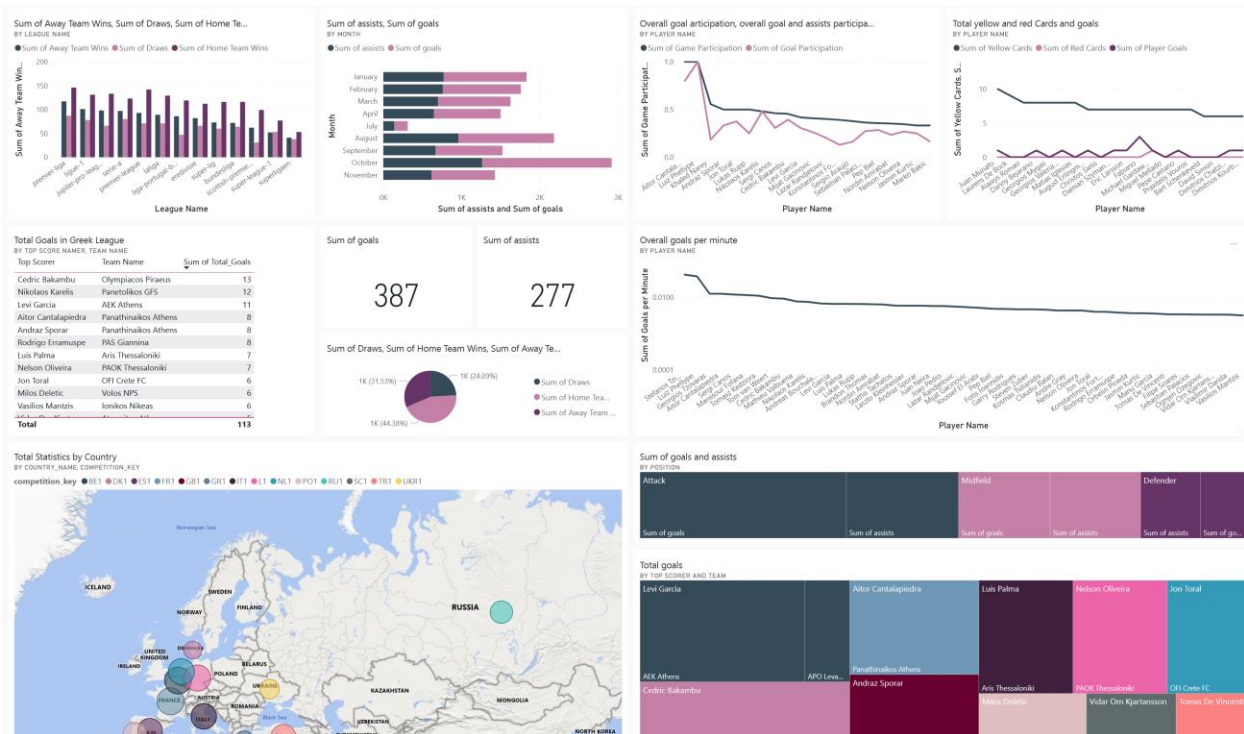
Για το dashboard πήρα τα πιο σημαντικά γραφήματα από τα παραπάνω κεφάλαια σχετικά με τις επιδόσεις των ομάδων και των παιχτών σε σχέση με την πάροδο του χρόνου. Παρότι φαίνεται πυκνό σε πληροφορία, έχω λάβει υπόψιν [τον τρόπο με τον οποίο διαβάζουν συνήθως οι άνθρωποι μια σελίδα\(f-shape reading\)](#) και συνεπώς οι πιο σημαντικές πληροφορίες που είναι καθαρές βρίσκονται στο πάνω αριστερά και προς το κέντρο ενώ τα περίπλοκα γραφήματα βρίσκονται στα άκρα.

Στην πρώτη σειρά αναπαριστώ τις συνολικές νίκες ήττες και ισοπαλίες ανά πρωτάθλημα, τα συνολικά goal και assist για όλα τα πρωταθλήματα, την συνολική συμμετοχή στα goal του παιχνιδιού, την συνολική συμμετοχή συνολικά goal και assist και τις συνολικές κίτρινες και κόκκινες κάρτες.

Στην δεύτερη σειρά απεικονίζω τους top scorer ανά ομάδα και τα συνολικά goal τους, 2 πινακάκια με τα συνολικά goals και assists για το ελληνικό πρωτάθλημα, ένα pie-chart με τις συνολικές νίκες, ήττες και ισοπαλίες και ένα γράφημα με την απόδοση goal/συνολικό χρόνο αγώνων.

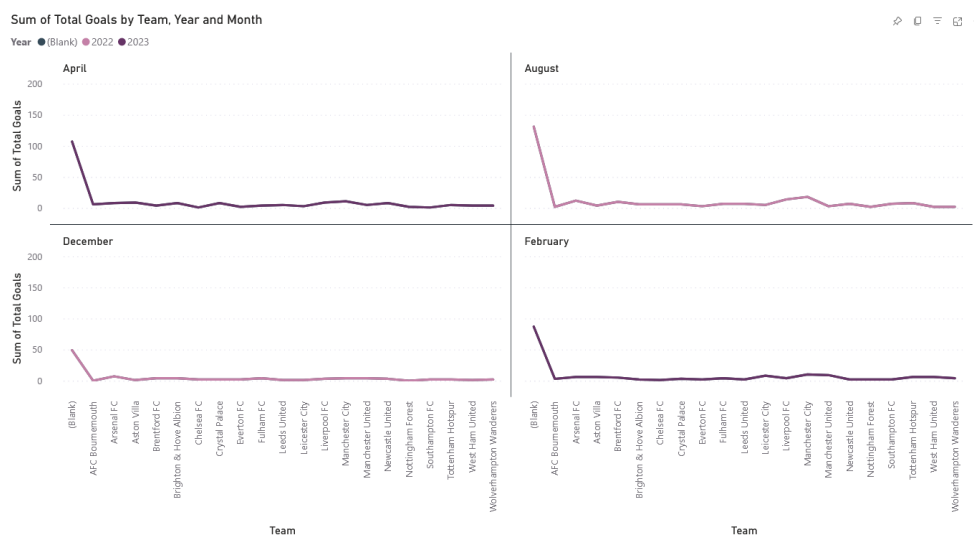
Στην τρίτη σειρά έχω τον χάρτη από το φύλλο 3.2_5, ένα tree graph με τα συνολικά goal και assist ανά θέση και το tree graph από το 2.4.

[Dashboard link](#)



Παράρτημα

Παρακάτω θα βρείτε μία εναλλακτική αναπαράσταση του κύβου αθροίζοντας τα goal για κάθε μήνα και ομάδα. Τα γραφήματα μπορείτε να το βρείτε στο 2.5(alt) φύλλο του report.



F-Shape pattern reading: https://en.wikipedia.org/wiki/Screen_reading