KickOff: a visual analytics system for analyzing football leagues in a given season

Andrea Napoletani[1] and Sergio Picca[2]

***Abstract*— In recent years, with the advent of technology, the world of football has changed radically. This change has affected the work of everyone in this world: coaches and their staff use the statistics to prepare matches knowing information about the opponents, journalists analyze statistics to improve the quality of their football commentaries and lots of supporters study this data just for statistical purposes. In this paper, we presented KickOff, a visual analytics system for analyzing football leagues in a given season. KickOff provides a web-based system organized in two views to support interactive visual analysis of an entire season of the most important football leagues: Italian Serie A, Spanish Primera Division, Franch League One, English Premier League and German Bundesliga. It also allows a comparison among players that belong to these leagues. Our source code is publicly available at `https://github.com/KickOff-VA/KickOff-VisualAnalytics`.**

I. INTRODUCTION

The growing impact of big data in the last years affect each task of everyone's life. Even the world of football is radically changed thanks to the use of new technologies and tools that allow being more precise and faster to accomplish some goals. Football results are strongly related to lots of parameters that often are represented by numeral values, the right analysis of these parameters can be an advantage towards opposing teams. For example, according to research about Italian Serie A [1], the performance of a team in the entire season are directly related to the number of conceded goals.

In this paper we present `KickOff`, a visual analytic system for evaluating statistics about the world of football. In particular, the system is designed to allow the user to analyze the performance of teams belonging to the most five important championships in Europe: *Italian Serie A*, *English Premier League*, *Spanish Primera Division*, *German Bundesliga* and *French League One*. In addition, `KickOff` allows comparisons of players in these leagues based on their statistics and their impact on the team's results in terms of minutes played and personal total score.

Our main goal is to allow journalists and supporters to obtain information from cross-correlation of data and derived metrics in a simple and fast way, in order to use it for improving working results or just for the statistical matter.

[1]Student of MSc Engineering in Computer Science, Università di Roma 'La Sapienza' Email: andrea.napoletani@gmail.com

[2]Student of MSc Engineering in Computer Science, Università di Roma 'La Sapienza' Email:sergiopicca39@gmail.com

II. DOMAIN AND ANALYTICAL TASKS

The first step of our work was the study of our domain of interest: the world of football. We started studying which are the most important factors that are useful to describe the performance of both teams and players. In this step, thanks to the help of the literature and existing tools like WyScout [2] and Transfermarkt [3] we identify the general requirements for an effective and efficient performance evaluation system.

A. Analytical Tasks

To accomplish this goal, the system was designed to address several visual analytical task categories for performance evaluation:

- **T1 Performance of the single player**
  the user may want to evaluate the skills of a single player in order to know which are his best aspects and in which he is lacking.
- **T2 Performance of the single team**
  the user may want to evaluate the performance of the single team during the year, see the progress of the team's score in the game weeks.
- **T3 Comparison among players**
  a comparison that a user may want to do is among single players belonging to the five major leagues. This comparison can be useful to analyze particular aspects of the player's performance and compare it in order to understand which player performs better in which situation. Each player is described by 38 attributes and a special one called *Overall* that summarizes the characteristics of the player in a unique value.
- **T4 Comparison among teams**
  another comparison might be the one among the teams belonging to the same league. Compare the performance of the teams during the year and cluster them based on the result of a dimensionality-reduction algorithm calculated on six attributes.

III. DATASET

To implement the views described above, we needed lots of details about teams and players' skills. At the moment of the creation of our system, we were no able to find a single open-source dataset that includes all the information that we need. For that reason, we decided to insert a preliminary step of data preprocessing in order to merge information coming from two different open-source datasets taken from Kaggle [4] and FigShare [5].

A short description of the two datasets is shown above:

- **Soccer match event dataset** [6] is composed by 8 files containing information about *Coaches*, *Referees*, *Players*, *Teams*, *Competitions*, *Events*, *Matches* and *Players ranks*. All these files are in described in a JSON format so that, for example, a single team is described as:

```
{    "city": "Milano",
     "name": "Milan",
     "wyId": 3157,
     "officialName": "AC Milan",
          "area": {
               "name": "Italy",
               "id": "380",
               "alpha3code": "ITA",
               "alpha2code": "IT"
               },
     "type": "club"
}
```

  We decided to don't use *Coaches*, *Events* and *Referees* files that contain information that is not useful for our purpose.
  Let's see the content of the other files, describing their most important fields:

  - *Players* contains information about each player: personal information, identifier of the current club, role and the unique identifier;
  - *Teams* contains information about each team: name of the city, name of the team, the geographic area that belongs to, the type (club/national) and the unique identifier;
  - *Matches* contains information about each match of the year: the identifier of the competition, the week of the league that belongs to, the unique identifier, the winner and other detailed information;
  - *Competitions* contains information the leagues: the geographic area associated with the league, the name, the type (for club/international) and the unique identifier;
  - *PlayeRanks* contains information about the performance of a single player in a match: identifier of the match, the identifier of the player, the number of minutes played, the role in the match, the number of goals scored and the score obtained by the player in the match.

- **FIFA 19 complete player dataset** [7] is composed by one single file in CSV format with 18206 players and 88 attributes per player. Each row of the file represent a single player described by name, ID, age, link of the photo (if available), nationality, market value, club, preferred foot, other information and 60 attributes strictly related to player's football skills (ex: Crossing, Finishing, Passing ecc.)

A. *Preprocessing of data*

Starting from these two datasets we needed to merge information taken from both to create a single dataset containing only the data that we had to use for our visual views. We created different types of files with different structures based on the type of visualization that will use them, we can summarize them in two groups: data referred to **players** and referred to **teams**.

For the players we created a new CSV file that adds the information of players' skills taken from *FIFA19 complete player dataset* to the "base" information of a player taken from *Soccer match event dataset*.

For the teams we created 3 new files:

- the first CSV file contains information about the total score of a team in each game week: it has been computed as the sum of the scores of all players belonging to the team in the single game week, iterated on each game week;
- the second CSV file contains the result of the application of 3 different dimensionality-reduction algorithms applied on 6 attributes of each team:
  - *number of home goals*;
  - *number of away goals*;
  - *number of wins*;
  - *number of draws*;
  - *number of defeats*;
  - *total score of the team in the entire season*.

  The algorithms used in this step are **PCA** [8], **MDS** [9] and **tSNE** [10], each of them calculated on 2 components. The file also contains the results of **K-Means** [11] algorithm applied on the first component of each dimensionality-reduction algorithm to group the teams in 2, 3, 4 or 5 clusters. This step required particular attention for the right tune of the parameters of each algorithm, in order to obtain coherent results;
- the last file is a JSON format file and contains information about the similarity among teams belonging to the same championship. The file is structured as:
  - a group of *nodes* with one entry for each team of the championship;
  - a group of *links* with one entry for each couple of teams and a value that represents the similarity among them (3 values based on the results of the 3 dimensionality-reduction algorithms). This value is calculated as:

$$|valueFirstTeam - valueSecondTeam| \quad (1)$$

  An example of the file structure is shown below:

```
{
 "nodes": [
   {"name": "SPAL", "group": 0},
   {"name": "Milan", "group": 0},
    .
    .
    .
```

```
  ]


 "links": [
   {"source": "0", "target": "1",
   "value_MDS": 3.032360350838,
   "value_PCA": 2.527204736542,
   "value_tSNE": 13.708524},
    .
    .
    .
  ]
}
```

IV. Technologies

For the developing of our project we used three main technologies:

A. Python

Python [12] has been used for the entire step of data preprocessing. It has been used for all operation of *read from* and *write to* CSV and JSON files and also for the application of dimensionality-reduction algorithms. In particular, for the application of PCA, MDS and tSNE we used `scikit-learn` that provides us powerful and easy to apply functions for dimensionality reduction.

B. Javascript and D3.js

Javascript [13] and **D3.js** [14] have been used for the developing of the views that composes the system.

C. Node.js

Node.js [15] has been used to build our project. It allowed us better handling of connection and also an increment of the scalability of our project.

V. Views

KickOff is implemented as a web-based application composed of 2 main views accessible by the right side retractable menu (fig. 1):

- **players view** allows the user to select players to analyze his skills and abilities (T1). It also allows a comparison between players based on their skills (T3);
- **teams view** allows the user to analyze the trend of a single team during the season (T2) and also to compare different teams based on their results (T4).

In the upper part of each view there is also a setting bar that allows the user to tune parameters that will reflect on the components of the view, in particular: for the *players view* the user will be able to select the skills to be shown, for the *teams view* the user will able to select the championship, choose the number of clusters in which the teams will be grouped on according to the results of *K-Means* algorithm and the dimensionality-reduction algorithm to apply.

The system has a responsive layout that allows adapting the visualization of all the component of the views on different screen dimensions or different level of browser zooming.

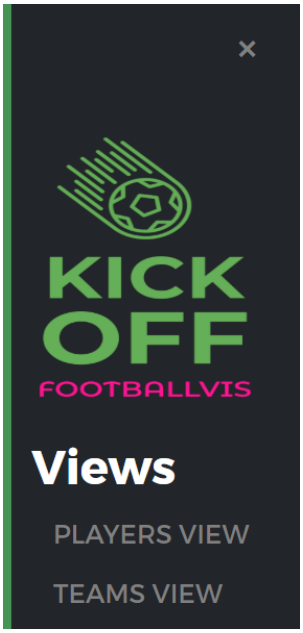

Fig. 1: Lateral menu.

A. Players view

This view implements two analytical tasks (T1 and T3) described in the *Section II*. It is logically divided in 2 steps:

- the **filtering step** is represented by the upper part of the page containing 2 components: *scatterplot* and *parallel coordinates*. These two component allow a first selection of the players that will be reflected in the second step;
- the **player-comparison step** is performed in the lower part of the page, in particular there are two components: a *selection* and a *visual component*. The *player-selection* part (fig 2), consisting of the group of players obtained in the filtering step, allows to select each player (displayed with a card) in order to make a further and detailed investigation in the visual component, by using a *barchart* and a *radar chart*.

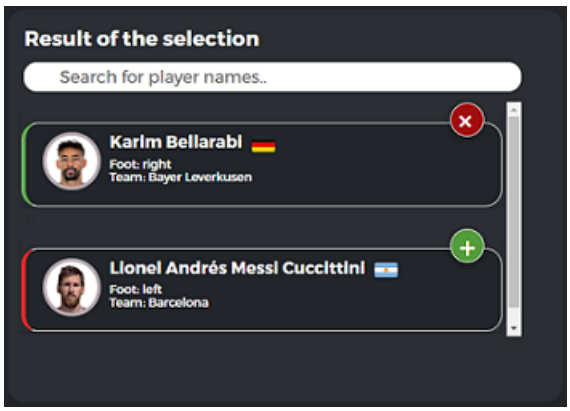

Fig. 2: Player Selection.

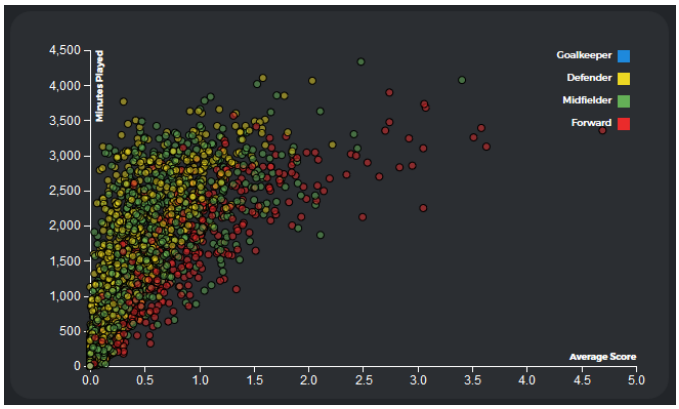

Fig. 3: Players Scatterplot.

1) ***Players Scatterplot****:* In this component (fig. 3) each player is represented by a dot whose colour depends on his role. The x-axis represents the *average score* of the player over the year and the y-axis the number of *minutes played* in the year.

This chart is useful to see the impact of a player during the year, the more the player is in the upper-right corner, the better his contribution to his team will be. On the contrary, the more his position is in the lower right corner, the less his contribution to his team will be.

2) ***Parallel Coordinates****:* In this component (fig. 4) each player is represented by a broken line whose colour depends on his role. The skills to be displayed can be selected from the dropdown list in the setting bar at the top of the page.

Thanks to this chart the user is able to see the values (approximately) of the players' selected skills.

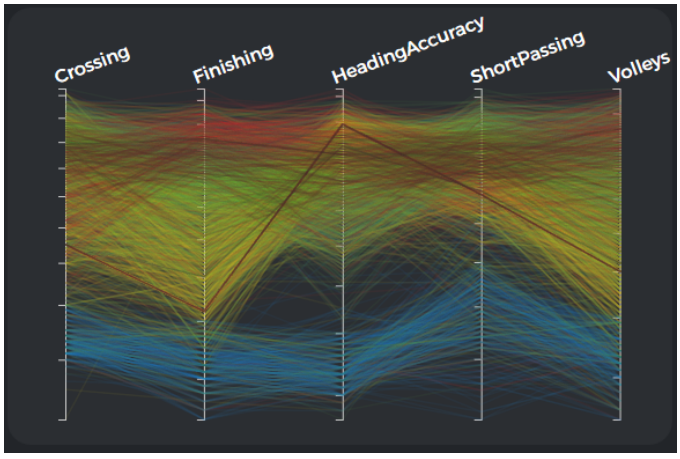

Fig. 4: Parallel Coordinates.

3) ***Barchart****:* In this component (fig. 5) each player is represented by a vertical bar where the height depends on the *overall* value of the player. The colours of the bars are assigned according to a colour scale starting from intense orange to intense blue, passing through different shades of these colours (it depends on the number of selected players).

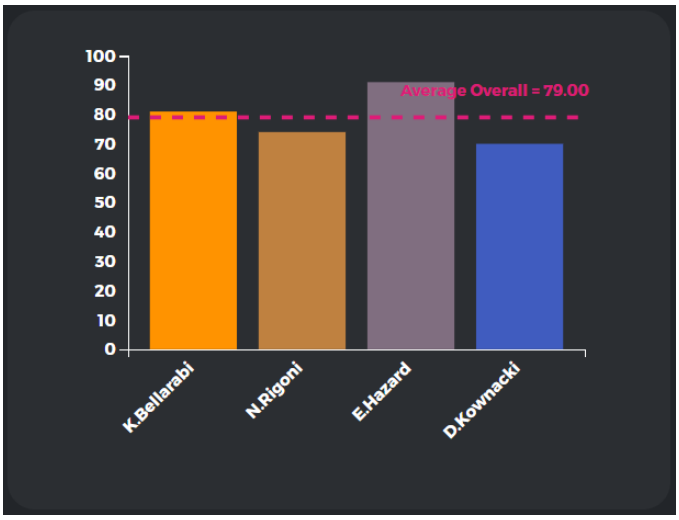

Fig. 5: Players Barchart.

This chart allows a direct comparison among players based only on the *overall* value.

4) ***Radar Chart****:* In this last component (fig. 6) each player is represented by a polygon where each vertex represents the exact value of a skill. The colour of each player's polygon is the same as the barchart one and the thickness of the polygonal outline encodes the value of the player's *overall*. The skills to be displayed can be selected from the dropdown list in the setting bar at the top of the page.

This chart allows a direct comparison among players based on their selected skills' values.

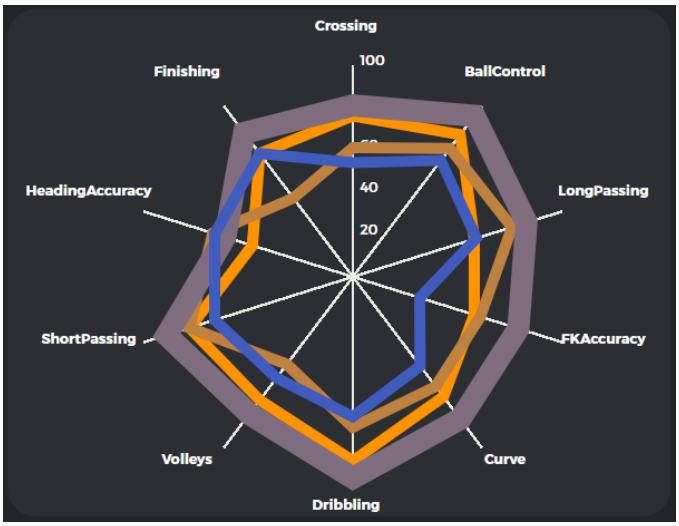

Fig. 6: Radar Chart.

5) ***Interactions among components****:* The components of the *players view* are connected through interactions, it means that operation done on one of them triggers a change on one (or more) other components of the view.

Let's see these interactions in detail:

- the **players scatterplot** component implements a lasso

function to select groups of players. This selection will be reflected on both *parallel coordinates* and *focused players* components. It also allows the user to know the complete name, the average score and the number of minutes played of a player opening a tooltip when the mouse is over the dot;

- the **parallel coordinates** component allows the brush functionality. By clicking and dragging along any axis, you can specify a filter for that dimension. Also this filtering will be reflected on both *players scatterplot* and *focused players* components;
- the **focused players** box shows the cards of the players selected in the *filtering step*. Adding one player (clicking on the "+" button) will draw his bar in the *players barchart* and will add the relative polygon in the *radar chart*;
- the **players barchart** allows to focus a single player by moving the mouse over his bar, it will also highlight the relative polygon in the *radar chart*;

B. Teams View

This view implements two analytical tasks (T2 and T4) described in the *Section II*. It is divided in three visualization: the *teams scatterplot* and the *matrix* giving an overview of the teams' performance, the *teams barchart* allowing direct comparison between two teams.

*1) **Teams Scatterplot**:* In this component (fig. 7) each team is represented by a dot whose colour depends on the cluster it belongs to. The position of the dot in the chart depends on the results of the dimensionality-reduction algorithm (2 components) applied on the team's attributes as explained in *section III*. For this reason, both x-axis and y-axis are meaningless but the user can obtain information about teams seeing their position: the closer they are, the more their performances are similar.

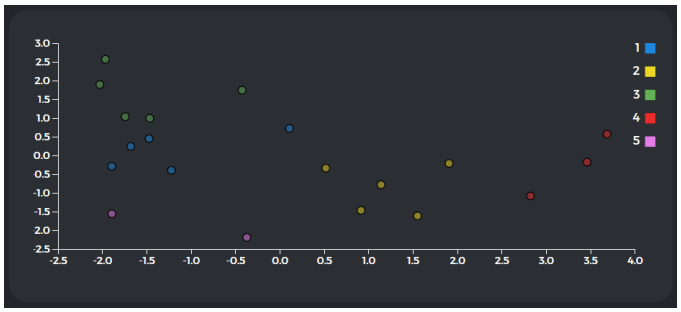

Fig. 7: Teams Scatterplot.

*2) **Matrix**:* In this component the teams are represented on both rows and columns. The cell encodes the information about the *dissimilarity* between the teams identified by that row and column using a 9 levels green color scale [16].

This chart (fig. 8) allows a comparison among teams' performance, in particular, for each couple of teams the higher is the intensity of the green, the more they have different performance, on the contrary, the lower is the intensity, the more the teams will have similar performance.

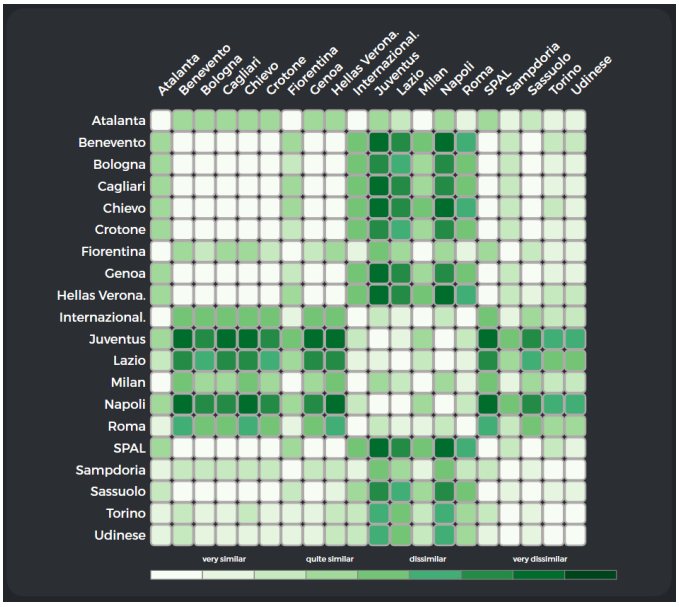

Fig. 8: Matrix.

*3) **Teams Barchart**:* In this component, each bar represents the total score of a team in a game week. If 2 teams are selected will be shown two adjacent bars, one for each team, for the half of the total number of game weeks, otherwise, if just 1 team is selected will be shown just one bar for each game week. The second half of game weeks results is accessible by clicking on the "Second round" button.

This chart (fig. 9) allows the user to directly compare the trend of 1 or 2 teams during the season and analyze which game week had the best performance.

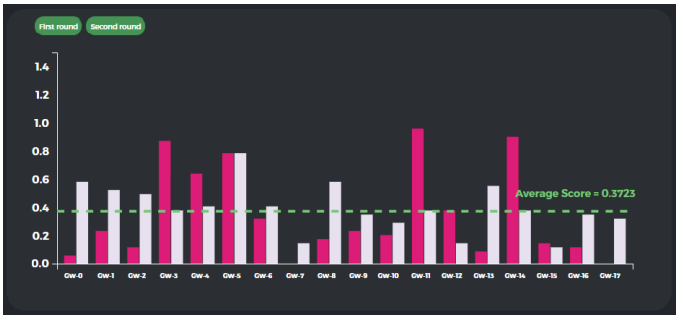

Fig. 9: Teams Barchart.

*4) **Interactions among components**:* The components of the *teams view* are connected through interactions, it means that operation done on one of them triggers a change on one (or more) other components of the view.

Let's see these interactions in detail:

- the **teams scatterplot** component implements a lasso function to select groups of teams. This selection will be reflected on the *matrix* highlighting the corresponding name of selected teams. It also allows the user to know the complete name of a team opening a tooltip when the mouse is over the dot;
- the **matrix** component allows exploiting the mouse

over function on a cell to individuate the position of the corresponding teams on the *teams scatterplot*. Is possible to select 1 or 2 teams by clicking on a cell, it will reflect the selection on the *teams barchart* updating the bars. It also allows the user to know the value of the *dissimilarity* opening a tooltip when the mouse is over the cell;

- the **teams barchart** component allows to exploit the mouse over function on a bar to highlights the corresponding team on both the *teams scatterplot* and *matrix* components. It also allows the user to know the complete name of the team and the corresponding score value opening a tooltip when the mouse is over the bar of a game week.

VI. Conclusion and Future Work

In this paper, we presented KickOff, a visual analytics system for analyzing football leagues in a given season. KickOff provides a web-based system organized in two views to support interactive visual analysis of an entire season of the most important football leagues: Italian Serie A, Spanish Primera Division, Franch League One, English Premier League and German Bundesliga.

To optimize and improve the system, we plan to add data about more football seasons, allow the possibility to compare teams from minor leagues and add information related to competitions for National Teams. Other possible improvements are to allow the comparison of players based on different (and selectable) attributes and add the possibility to analyze the performance of a player in a single match.

VII. Source Code and Demo

The source code of KickOff project, included the part of data preprocessing, is available at https://github.com/KickOff-VA/KickOff-VisualAnalytics.

A demo of the system is available at https://kickoff.azurewebsites.net/

References

[1] G. D. Carolis, "Vuoi vincere lo scudetto? il miglior attacco è la difesa," *Corriere*, 2020.
[2] S. S. Company, "Wyscout," *https://wyscout.com/*.
[3] M. Seidel, "Transfermarket," *https://www.transfermarkt.it/*.
[4] Google, "Kaggle, find and use datasets or complete tasks.," *https://www.kaggle.com/datasets*.
[5] DigitalScience, "Figshare," *https://figshare.com/*.
[6] L. Pappalardo and E. Massucco, "Soccer match even dataset," *https://doi.org/10.6084/m9.figshare.c.4415000.v5*, 2019.
[7] K. Gadiya, "Fifa 19 complete player dataset," *https://www.kaggle.com/karangadiya/fifa19*, 2019.
[8] "Principal component analysis," *https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html*.
[9] "Multidimensional scaling," *https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html*.
[10] "t-distributed stochastic neighbor embedding," *https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html*.
[11] "Introduction to k-means clustering," *https://blogs.oracle.com/datascience/introduction-to-k-means-clustering*.
[12] "Python," *https://www.python.org/*.
[13] "Javascript," *https://www.javascript.com/*.
[14] "D3.js data-driven documents," *https://d3js.org/*.
[15] "Node.js," *https://nodejs.org/en/*.
[16] "Colorbrewer 2.0," *https://colorbrewer2.org/#type=sequential&scheme=BuGn&n=9*.