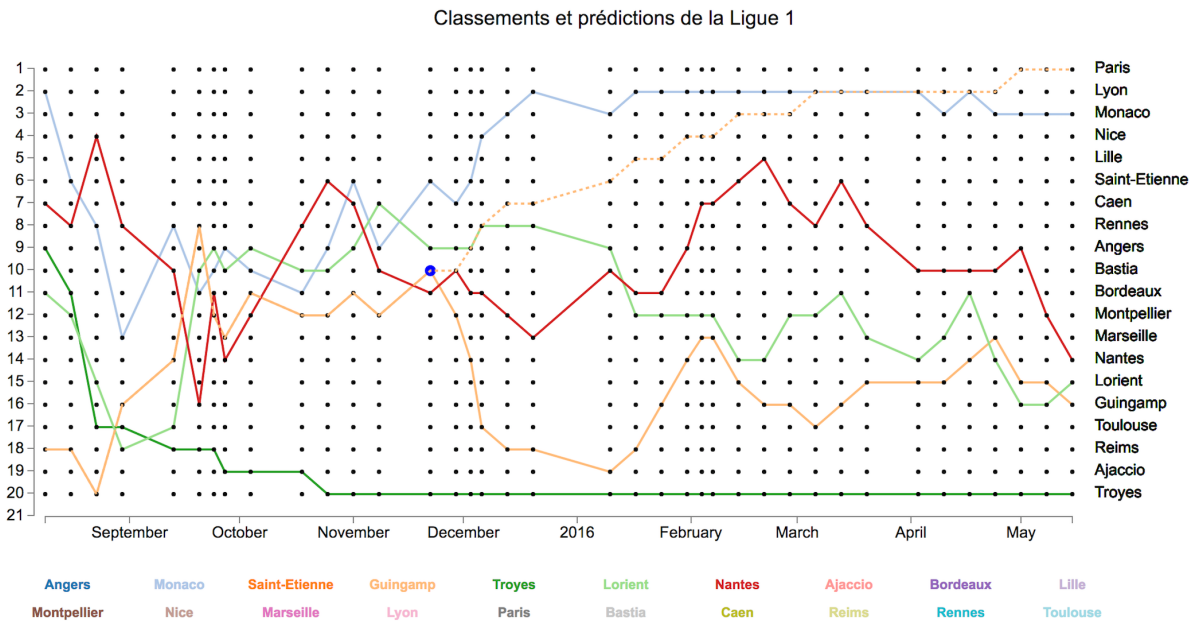
Championships ranking visualisation

Florian Robinet
Lyon 1 University
Data Science Degree

Orhan Yazar
Lyon 1 University
Data Science Degree



Abstract

This work is related with championships ranking visualization. It is a relatively generic model which could be adapted to many types of sports. The two main goals of this visualisation are to show the effective ranking of each team during a season and the ranking predictions linked. This visualization allows us to see the trend of a team and to make easy comparisons between several teams. Also, we can see easily the prediction ranking at each dates and for the rest of the season of every teams. It is a good tool to highlight the quality of the prediction by comparisons between the predicted and the effective ranking. Furthermore, the predictions are based on the historical ranking during the season so we could see the amelioration of prediction as the season progress. For the prediction we used different probabilistic and statistic models such as time series and exponential model. We decided to do this visualization because first we both like football and we were curious to know if we could be able to make prediction about football ranking and then compare it with the real evolution. We found a website which doest prediction only which lead us for the prediction but we wanted to represent this with a simple and clear linechart.

1 Introduction

With all the new technologies and the evolution of the mentality, we have to observe all the data around us, visualize and show all of them to understand them in all their forms. In fact, showing data can help us to understand them better and maybe discover somethings we didn't know. It can also permit us to explain difficult phenomenon in a simple way. As we said, we will focus here on the 2015-2016 season of "ligue 1" which is the french football first league.

Indeed, by the time, clubs, magazines and people in general like more and more the visualisation of ranking, goals and other data related to football. In fact, it's interesting to visualize this kind of data for people for example to bet, or for clubs to prepare matches, and for magazines to show results and evolution. What we have to know is that the visualization is intended for all public so it has to be simple and clear but in the same time provide enough information. There are many data to show in football but to make something easy to understand we have to focus on a subject. An interesting thing is to visualize ranking to see the evolution of one or more team all along the season. In fact, people may be curious to see how a club evolved during a season or clubs themselves.

We have looked for a model that could show us multiple predictions while keeping something clear and readable. The issue is highlighted during this work was:

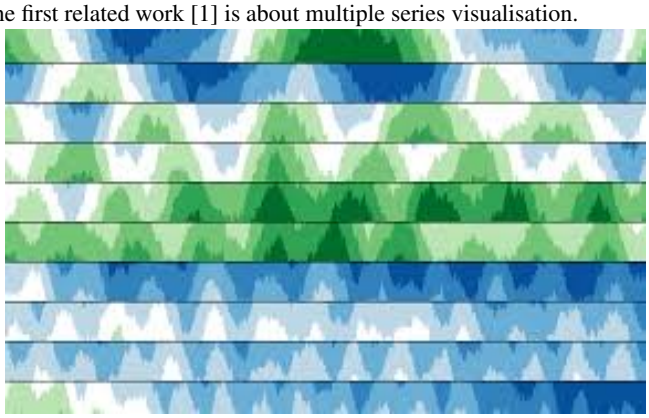
How highlight championships prediction ranking in a clear way ?

We will try to answer this question by seeing the related work which have influenced us, then the building of the visualisation and finally the issues that we have encountered and the possible improvements.

2 Related Work

2.1 Interactive horizon graph

The first related work [1] is about multiple series visualisation.



Interactive Horizon Graph

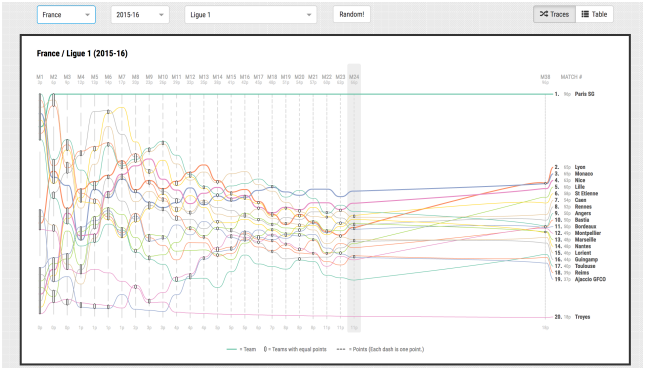
- Interactive Horizon Graphs is a merge between Reduced Line Chart graphs and Horizon graphs. The realization of this type of graph is done in three steps: the coloration, the slicing and the superposition. Two main interactive techniques are implemented in this visualisation: the control of the baseline position and possibility of zooming on the values.

- We have retained a main thing of this visualisation which is that it is really difficult to study multiple series in line charts. In this visualisation technique there is a coloration by intensity which make the differentiation easier. In that sense, we put in our visualisation a selector for each team to draw the curves, it makes easier the readability.

- We have not retained the two main interactive techniques of the interactive horizon techniques. The aspect of coloration was barely impossible to implement in our visualisation. In that case we would have separate the curve for each team which is not really readable, furthermore, for values between 1 and 20 it is not really efficient. As well, we thought that the baseline spanning is not effective for our type of data.

2.2 Standings Tracer

The second related work [2] is more similar to our work and is focus on soccer championships ranking.



Standings Tracer

- The standings tracer is a really good visualisation which offers lots of information. First, you have the championships soccer ranking for a large number of countries and seasons. The ranking is associated to the points of the team and the curve drawing is associated to that. Second, you have the result of each matches during the season. Third, you could see a linear interpolation of the ranking for each day of the season. Finally, there is a form of classification for the similar teams.

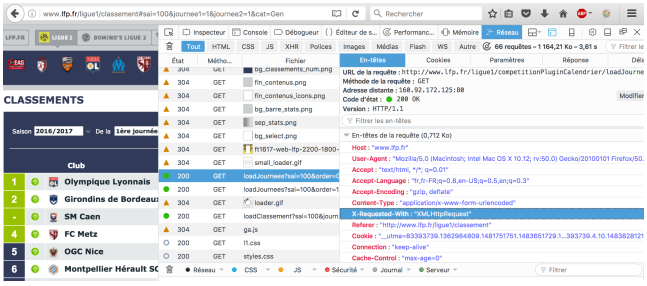
- We have tried to retain lots of things in this visualisation due to its quality. The first thing is the global appearance of the visualisation. It is a visualisation of the championship ranking with the day of the championship as the x-axis and the ranking as the y-axis. Also, the coloration of each team curve and the name of each team on the right has been kept.

- Due to the complexity of this visualisation it was not possible to retain all the elements. We have not included the result of the matches because it was too difficult to make it readable. We did not retain the classification method and the linear interpolation. It is two elements that we could have implemented to give more sense to our visualisation but there were technical difficulties for realizing it.

3 Project Description

3.1 Database built up

We took our data on the website www.lfp.fr with a web-scrapping python code. The code allows us to collect the league 1 championships ranking day by day for each seasons since 1930.
The website doesnt write the data in the html code of the webpage but obtains it from a server with a JavaScript code. So the first step was to find where and how the data are loaded on the page. We found this information by the study of the headers.



Header of the GET request to obtain the data

The download of the data is done with the software Curl using the right header information such as X-Requested-With which must be an XML Http request. The data are written under an html format so it is necessary to parse it. An excellent package to do that is Beautiful Soup. We could access to each values of a table by a loop on the tr and td html tag. The data are stocked in a python dictionary with the teams as keys value. Finally, the dictionary is exported in text files with regex treatment for finalise the parsing.

```
v=np.r_[5:50:1]
for int in v:
    vv=str(int)
    vec=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38]
    dico = {}
    for i in vec:
        a = str(i)
        url1 = 'http://www.lfp.fr/ligue1/competitionPluginClassement/loadClassement?sai='+vv+'&journee1=1&journee2='+a+'&cat=Gen'
        buffer = StringIO()
        c = pycurl.Curl()
        c.setopt(c.URL,url1)
        c.setopt(pycurl.HTTPHEADER, ['X-Requested-With: XMLHttpRequest', 'Content-Type: application/x-www-form-urlencoded'])
        c.setopt(c.WRITEDATA, buffer)
        c.perform()
        c.close()
        bodi = buffer.getvalue()
        time.sleep(1)
        soup = BeautifulSoup(bodi, 'html.parser')
        lignes = soup.findAll('tr')
        for l in lignes:
            if (len(l.findAll('td')) == 11):
                team = l.findAll('td')[2].text.replace(" ", "").replace("\n", "").replace("\t", "").encode('ascii',
                                                                                                            'ignore') + \
                       l.findAll('td')[0].text
                teamf = re.sub(r'(([1-9].*|\-))$', r',\1', team)
                teamff = teamf.split(',')
                if teamff[0] in dico.keys():
                    dico[teamff[0]].append([teamff[1]])
                else:
                    dico[teamff[0]] = [[teamff[1]]]
```

Web-scrapping python code

3.2 The prediction methods

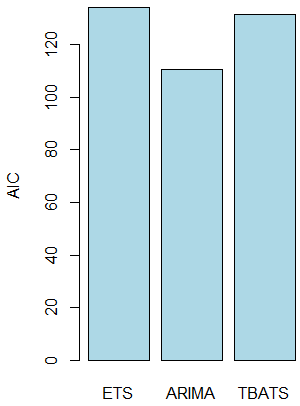
There are many methods to predict data like linear models, discriminant methods, time series and many others. Here we'll use the time series methods because we thought it was the most coherent and adapted methods and because it's difficult and it takes more time to make this kind of forecast with other method. In fact, we have data by week all along the year and we want to predict for example the next 13 week's score knowing the first 25 weeks, it means that we want to forecast data knowing what happened before. In the time series methods, we can choose different model to fit and predict our data.

Time series decompose variables into two components: Trend and Seasonal, and then forecast data. Using time series, we have 3 data modeling methods:

- exponential model
- autoregressive integrated moving averages model
- TBATS model

The exponential model apply an exponential smooth before prediction (smoothing parameters are estimated using maximum likelihood estimation).
The auto ARIMA model provides seasonal and non-seasonal ARIMA model estimation including covariates $(y_t = \mu + {}_1y_{(t-1)} + {}_2y_{(t-2)} + {}_3y_{(t-3)} + \varepsilon_t)$.
The TBATS mode is a multiple seasonal model used for annual seasonality, or weekly seasonality.

It's interesting to first work with the three methods and then try to choose the best one comparing the AIC's (Akaike information criterion: is a measure of the quality of a statistical model proposed by Hirotugu Akaike in 1973). The best model is the one with the lowest AIC. We tried it for many forecast and we found the best model for our forecast. The best model is not always the same but most of the time the best one is the AIC model. As you can see in the picture below, the best model to use here is the autoregressive integrated moving averages model:



AIC plot for models

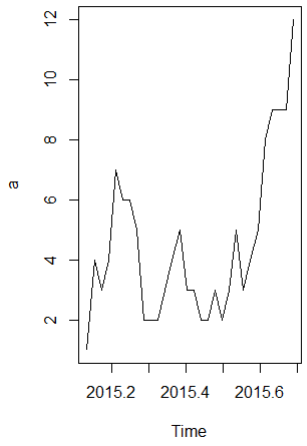
Now the thing is to tell the machine that we can't go below 1 and above 20. So we have to make a code to adapt those limit. The code below is to do this for all the prediction we need, ie 760 predictions.

Listing 1: code R for limit

```
a=ts(data,start=c(2015,08,22),frequency=365.25/7)
for(i in 1:length(a)){
  if(a[i]==20){
    a[i]=19.5
  }
}
a.transform=log(a/(20-a))
m_a.transform = auto.arima(a.transform)
f_a.transform = forecast(m_a.transform, h=10)
f_a = f_a.transform
f_a$lower <- 19*exp(f_a.transform$lower)/(1+exp(f_a↩
    .transform$lower)) + 1
f_a$mean <- 19*exp(f_a.transform$mean)/(1+exp(f_a.↩
    transform$mean)) + 1
f_a$upper <- 19*exp(f_a.transform$upper)/(1+exp(f_a↩
    .transform$upper)) + 1
```

When we transform the time series to set the limit, we use a log function but the problem is that when $a = 20, 20 - a = 0$ and we can't apply log to 0 so we need a kind of preprocessing. Every time we have 20, we change it into 19.5 to avoid 0 and stay close o the real data.

Now let's see how can we display the prediction. The two figures below show the plot of the ranking of Angers for the first 30 days and the figure below show the ranking for the first 30 days plus the prediction for the 8 last days with the 80



Ranking of Angers for the first 30 days

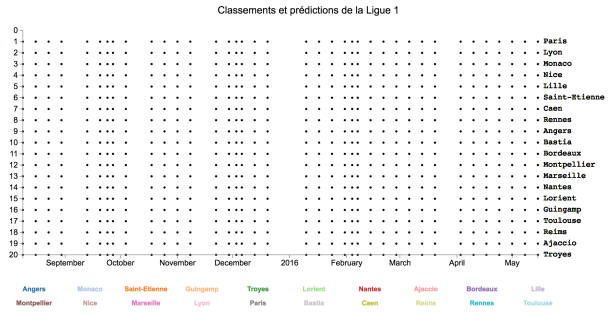
=200]prediction

Ranking and prediction with confidence interval

3.3 The visualisation

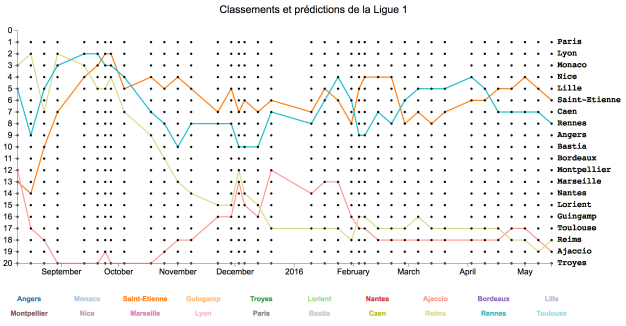
The data are stocked under a csv format. It contains for each teams the effective and the predicted ranking day by day. There are 20 teams, and for each team 38 days, the rank of all those days and the points at each day. We also have the prediction of all days but the first because we have nothing to predict it in our data and the AIC for each prediction's model. We have 34 200 data. We are using the function d3. csv () to parse the data and the function d3. nest () to classify them by team. Also, we have inserted the championship total points for each team and day of the season. A vector of colour is defined to attribute a unique colour to each team.

The x-axis is scaled on the date of the matches. We have made the choice to keep the date because like that we could see if the teams have played recently or not. Concerning the y-axis, we have switch the ranking to put the one up and the twenty down. It gives us a better readability of the ranking, instinctively we expect the one to be up.



The visualisation backgroud

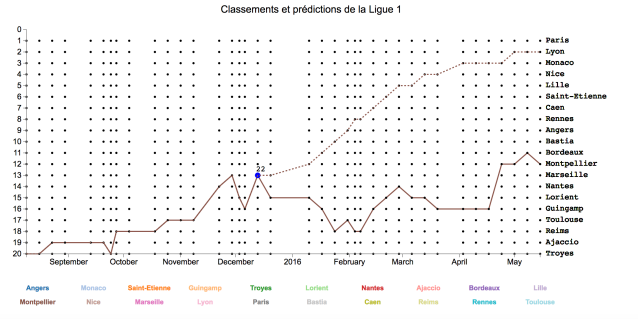
The lines are drawn on the by a loop on the data, we call the attributes with a function associated to each column. We do not success to make a loop to call the attribute. Each lines are associated to an id to recognize them. The lines concerning the effective ranking are continuous and the predictions are dotted.



The effective rankings

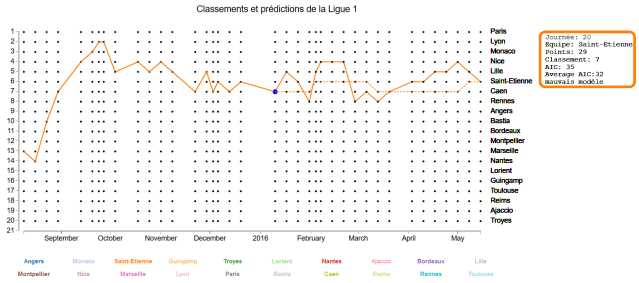
Each dots represent the state for team at a day of the championships. We have decided to make them interactive and to put them as a key element of the visualisation. We you put your pointer under it, that display the total championships points of the team at this date. Furthermore, it displays the prediction for this date. When you move your pointer it disappears to not overload the visualisation. On the right of the visualisation we wrote the name of each team ranking for the last day of the championships.

The legend is put at the bottom of the visualisation. It is an interactive legend which allows to draw the ranking of a team by clicking. In the same way you could make it disappear with a second click or superimpose several curves. It was really important to have this functionality to keep a clear visibility of each curves.



The prediction rankings

Everytime you pass your mouse on a point, it display the prediction based on all the days before from this point to the end.



Informations for each point

Passing the mouse to a point also display informations about the point, the day, the team, the team's rank, and the model's performance. The performance is measured with the AIC.

We add a last thing to our visualization. As you can see, we put the teams name at the right too. It has two functions, first, the teams are classed by order of the rank of the last day. Second, when you click on a team, it opens a website where you can find information about the teams.

4 Discussion

For the visualization, there are many things we could have add or improve. First, we could smooth the lines to make it prettier but this kind of lines shows better variations. We tried to make a style effect for the curve with the function interpolate and the argument basis. This idea was inspired by the visualisation standings tracer, this functionality improves the visual aspect of the curves. The issue was that the curves must pass throw the dot so the points must be interpolated too and we did not success this task. We also wanted to add a slider to choose the year we cant to display but it was a bit difficult due to the number of our data. With 1 season only we have 760*39 data. if we add another year or 2 or 3 other year, it will be useless because it will take too much time to make the visualization work.

Another issue that we encounter concerns the JavaScript code. To draw a curve, we used a combination of a function which was returning the arguments and a function which was drawing effectively the curves. With this manner, we did not succeed a loop, so we wrote a function for each day of the championships (38 in total). This is not the right way to proceed but it was difficult to find a loop which was functioning.

When we wanted to add the url to get more information of each team, we tried to add a second function to the team name at the bottom, but it was difficult as we already add a "click on" to display lines. We could maybe find something else to avoid this.

For the prediction, we used a time series method but as we said there are many and many methods to forecast data. Here we predicted data with only one method using only 1 model. To go further, we can first try all the three models to predict our data using time series and for each prediction, use the most appropriated method to improve our forecast because as we said, some prediction are better with the exponential model or with the TBATS model so if we have more time, we can, for each prediction, use all three models, compare AIC's and choose the one with the lowest AIC instead of choosing the same model for all the prediction. This step can already grandly improve the prediction. Then, instead of using only time series method, we can try to use other methods like linear regression, discriminant analysis, and so on.

Here we tried to forecast the rank just using the rank before but we can build more complex model adding the budget of the club, the average of goal, results of passed years and many others to enrich the model and improve the prediction. We can add all the variable we want, even if there seem to be useless because when we build our model, we can ask for the model performance and for the best variable combination. The problem is that there are too much possibilities for prediction methods, models and the choice of variable and if we want to find the best combination we need more time and particularly to be organized because all those tests create an enormous quantity of data which we need then to treat and use. Moreover, when we built up our times series, we had to do a preprocessing before the forecast, to avoid NaN, and Infinity arguments which distorted a bit our data and so the predictions. We can try another way to avoid this preprocessing or maybe find another preprocessing to set the limit for the predictions.

5 Conclusion

As a conclusion, we can say that this visualization can be a support for who wants to go further. For example, first build a better and more complex prediction model and add more options for the visualization. We made something simple to use because our visualization is for everybody. Every football amateur can check how is evolving or how evolved the championship. A team can check his own evolution during the season and compare with other team.

6 acknowledgments

For this project, we wish to thank mr.Vuillemot who helped us develop our project leading us for the visualization. We also wish to thanks mrs.Mercadier who helped us for the time series prediction.

References

[1] J. D. F. Charles Perin, Frederic Vernier. Interactive horizon graphs, 2015.

[2] K. Wongsuphasawat. Standings tracer, 2016.