Statistical Methods and Stochastic Processes

COURSE PROJECT

From The Shadows to The Peak: A Statistical Analysis of Girona's Ascent Over Three Seasons

---

AUTHORS: Víctor Méndez Riosalido and Ricard Garcia Isern

DATE: 3-12-2023

COURSE: 2023-2024

INDEX

1.Context and goals

When tasked with conducting a statistical analysis, our primary aim was to choose a subject of statistical significance that also held personal relevance, allowing for a comprehensive interpretation of the data and its outcomes. After conscious research, we reached to the impressive performance of the Spanish football team, Girona FC, over the last few years.

The evolution of Girona FC serves as an exceptionally captivating and intriguing case study, undergoing a transition from descending from the maximum Spanish football league to ascending to the top of the same league, within a span of three years. In the 2018-19 season, the team faced relegation after finishing 18th at the bottom of the table, signifying a season marked by struggles and numerous losses. However, after three years, following its promotion to the first division, Girona achieved a 10th-place finish, in the exact middle of the leaderboard.

Furthermore, for the time we are performing this analysis, Girona occupies the 1st-place position in the top of the Spanish league. This scientific report aims to understand the variations in the team's results across different seasons, with a specific goal of identifying any potential correlation between the outcome of the match and the match immediately preceding it.

As mentioned before, since the main objective of this study is to predict the outcomes of matches involving our team of interest, Girona FC, we will conduct an analysis based on Markov Models sustained by the fact that the result of a match is closely related with the result of the previous one due to factors as motivation and shape of the team. More precisely, we will determine a transition probability matrix for each season. This technique will enable us to predict the results of matches for the remainder of the 2023-2024 season involving this team assuming the shape of the team is steady throughout the season.

1.1 Expected results before computations

The primary justification for choosing Girona FC over any other team is based on the evident disparities in the team's performance across different seasons. Our hypotheses are different depending on a specific season.

For the 2018-2019 season, we hypothesize that following any result in a game, the subsequent game is likely to result in a loss or a draw, considering the team's relegation during that season. Turning to the 2022-2023 season, the next match could produce any outcome, with a particular emphasis on draws. Lastly, for the 2023-2024 season, a win is anticipated. These assumptions are made just by looking at the results for each season.

Following this rationale, when we apply the log-likelihood ratio to a series of outcomes characterized by a lack of wins, we must infer that the series does not align with the expectations for the 2023-2024 season. This analytical approach aims to uncover statistical patterns in match outcomes that align with our hypothesized expectations for each respective season.

2.Data set description

To perform our statistical study we ended up with three different very interesting datasets from the FBRef website, one for each season, with a wide range of variables that could be used to perform multiple analyses depending on our main goal, more precisely, our data includes 17 categories.

2.1 Categorical data

We can divide our dataset into categorical data. This type of data is the one that represents non-numerical information grouped into categories or labels, subclassified as nominal (unordered) or ordinal (ordered). In our data set, these categories are the following:

- 'Resultado' → represents if the team wins (V), loses (D) or draws (E).
- *'Fecha'* → represents the date of the match, given in *YYYY-MM-DD* format.
- 'Hora' → represents the hour of the match, given in *HH:MM:SS* format.
- 'Comp' (Competition) → indicates the league or cup competition of the match.
- 'Ronda' → indicates the round of the match.
- 'Día' → indicates the day of the week when the match takes place.
- 'Sedes' → indicates whether the match was played at home (Local) or away (Visitante).
- 'Adversario' → represents the opposing team.
- 'Capitán' → indicates the captain of the team.
- 'Formación' → specifies the tactical formation used by the team.
- 'Árbirtro' → represents the name of the referee officiating the match.

2.2 Numerical data

We can divide our dataset into numerical data. This type of data is the one that refers to quantitative information represented by numbers. It can be grouped in discrete data (separate values) and continuous data (can take any value within a range). In our data set, these categories are the following:

- GF → indicates the number of goals scored by the team.
- GC → indicates the number of goals conceded by the team.
- xG → represents the expected number of goals for the team.
- xGA → represents the expected number of goals conceded by the team.
- 'Pos' (Position) → indicates the team's position in the european ranking
- 'Asistencia' → represents the attendance at the match.

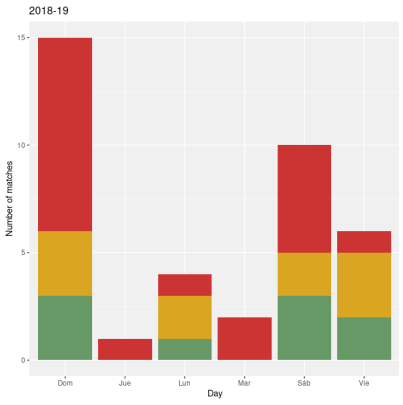
As commented before, we can do many different studies with all these categories. For our final analysis we are going to use only the information from the category '**Resultado'** which we are treating as a Markov chain. Before doing our final study, we will explore our different datasets.

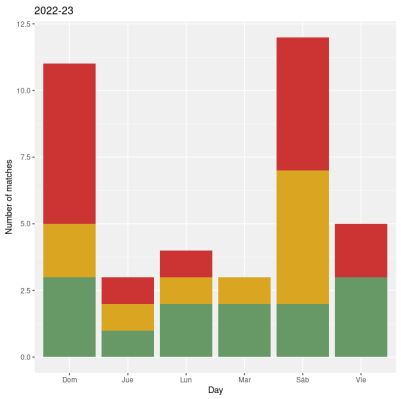
2.3 Exploratory analysis

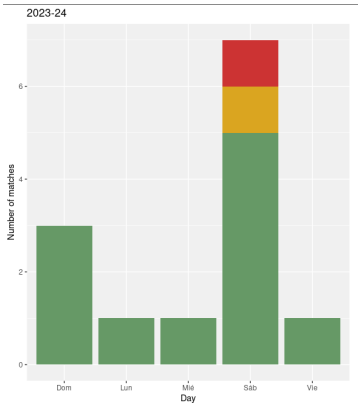
In initiating our analysis of the dataset, we implemented filters based on the competition and the prevailing season. We excluded matches not associated with the La Liga competition. Specifically, for the 23-24 season, all games initially denoted as "NA" were excluded from the analysis, signifying that they had not been played at the time of data collection.

We will now investigate the distribution of results across various interesting categories. It is crucial to note that the plots are arranged chronologically by season.

We have considered it fitting to combine the data related to the **day of the week** on which the match is played with the final **result**. This will allow us to analyze whether there is any discernible performance trend associated with this variable.



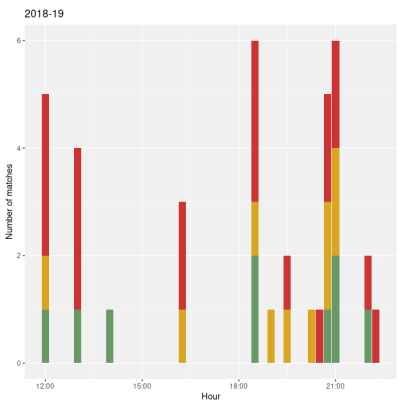


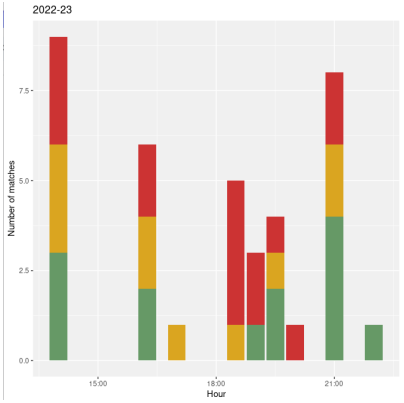


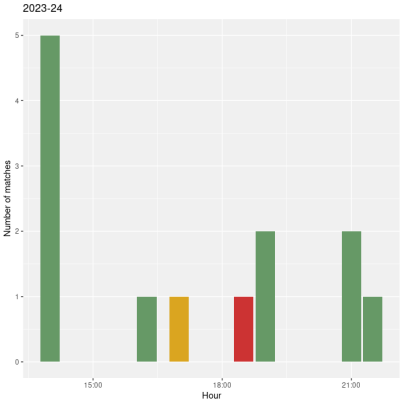
Upon visually inspecting the graphs, no clear performance pattern emerged. In the first season, there was a balanced distribution of victories, draws, and defeats across different days, taking into account the varying number of matches played. The second season followed a similar pattern, except for a noticeable shift on Saturdays, where there was a higher number of draws compared to other days. Particularly, in the last season, Saturday was distinctive as the only day with results differing from victories.

In summary, despite visual scrutiny, no apparent performance pattern was discerned throughout the seasons.

We have also integrated the data on match **results** with the **start time** of each match to visually explore if any significant patterns emerge.



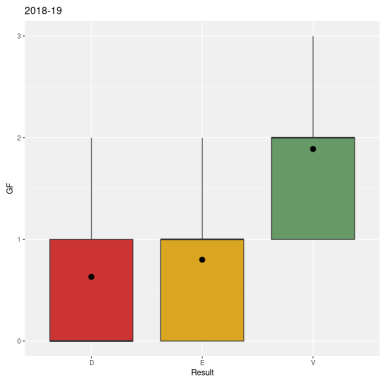


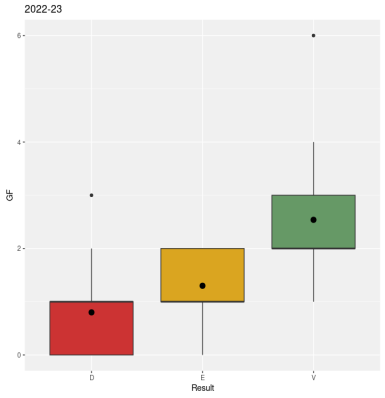


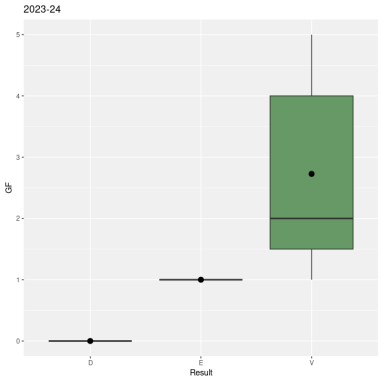
For the first season, we cannot observe any specific time where a discernible performance pattern or significant change is evident. The same holds true for the second season, with results remaining fairly balanced regardless of the match time. In the third season, however, we note a performance difference during the early afternoon hours, specifically between 17:00 and 18:00 hours, where a shift in the pattern is evident.

However, after visualizing the plots, we are unable to draw any conclusions regarding team performance based on the match start time.

Finally, we have merged the data on **goals in favor per match** with the results to analyze any differences.







For the first season, it can be observed that whether the Persia team or the team drew, the average goals were very similar, and if the team won, the average increased by one unit. In the second season, we can see that the average gradually increases depending on the outcome. Finally, for the third season, it is evident that the goal average is significantly higher when the result is a victory compared to other outcomes.

3.Techniques used to analyze the data

After carefully examining the available data and visualizing plots, we could not discern any discernible pattern. Despite our efforts to visually analyze the plots, the elusive nature of patterns prompted us to transition to the next phase of our study.

To carry out our statistical investigation, we are adopting models and tools designed to elucidate and predict the phenomenon at hand. The focal point of our study revolves around the performance outcomes of Girona FC, specifically, whether the team secures a victory, suffers a defeat, or concludes the match with a draw. With a clearly defined final objective for our project, we are strategically integrating the stochastic model known as the Markov Model.

To provide context, a stochastic model involves the analysis of a set of random variables that depend on some parameter and are correlated with time (t). Therefore, for each time value t, there exists a probability of the event occurring. The purpose of such models is to assess the likelihood of events that cannot be precisely predicted. Having explained the concept of a stochastic process, we can delve into Markov Models.

Markov models are optimal for analyzing systems that change randomly, where it is assumed that future states do not depend on past states. Such models can be employed to recognize patterns, make predictions, and derive statistics from sequential data. In our case, this data comprises the victories, defeats, and draws of the football team in question.

3.1 Markov chain model

This model studies the state of a system with a random variable that changes over time. Therefore, with this information, we can assume that the distribution of this variable depends only on the distribution of a previous state. The values taken by any of these random variables are called states, and with this model, we assume that the chains have a finite state space. In our case, there are 3 values {W, L, D}, corresponding to whether it is a win, loss, or draw, respectively.

A stochastic process is termed a Markov chain when the following Markov property is satisfied n times, and it holds for all states:

$$\Pr\{X_k = i | X_{k-1}, X_{k-2}, \ldots, X_1\} = \Pr\{X_k = i | X_{k-1}\}$$

The formula above tells us that the conditional probability of a Markov chain being in a future state (X_k), given the entire history of past states (X_{k-1}, X_{k-2}, ..., X_1), is equal to the conditional probability of being in that future state (X_k) given only the most recent state ($X_{k-1}1$). In summary, this property signifies that in a Markov chain, the current state contains all the necessary information to predict the future state, and knowledge of the past beyond the most recent state does not provide additional predictive power.

3.2 Transition probability matrix

The Markov chains are exclusively determined by the transition probability matrix. This matrix serves the primary purpose of quantifying the probabilities associated with transitioning from one state to another within the Markov process. In our specific context, the distinct states, previously mentioned, are Victory (V), defeat (D), or draw (E).

The transition probability matrix, conventionally denoted as P, is a square matrix where each entry Pij signifies the precise probability of transitioning from state i to state j in a single step. In more explicit terms, Pr(Xk=i|Xk-1).

Thanks to this matrix, subsequent to its application, we will be equipped to make predictions and conduct a statistical analysis of the behavior of our sports team, Girona FC.

To ascertain whether a Markov chain is more likely to conform to a model following one transition probability matrix or another model following a different transition probability matrix, the following formula can be employed:

$$\sum_i \log \frac{p_+(x_i|x_{i-1})}{p_-(x_i|x_{i-1})}$$

This formula implies a comparison of probabilities regarding whether a chain belongs to a transition probability matrix from a specific model or another. The nominator refers to the probability in a specific position of the transition probability matrix of one sequence while the denominator refers to the same position in the matrix of the other sequence. In our context an example with an order 1 Markov transition matrix would be the probability of winning a match after a defeat. The result can be analyzed as follows: if the final numerical outcome is positive, the chain is more likely to be associated with the model whose transition probability matrix is introduced in the numerator of the aforementioned formula. Conversely, if the result is negative, it suggests that the chain is more likely to be associated with the transition probability matrix in the denominator.

3.3 Steady state distribution

Once we are familiar with the transition probability matrix, we can grasp the concept of the steady state distribution. This distribution delineates the probability distribution of locating the system in each of its states as it evolves infinitely over time. In our specific scenario, the stationary distribution takes the form of a probability vector $\pi = [\pi V, \pi D, \pi E]$, where πV, πD, and πE represent the stationary probabilities of the team occupying the states of victory, defeat, and draw, respectively.

Upon reaching its steady state, the transition probabilities cease to change over time, resulting in the matrix remaining constant as time progresses. This state emerges when, after a substantial number of steps, the probability distribution converges to a constant state.

In essence, this matrix allows us to discern the long-term probabilities of the team under investigation being in each of its states, irrespective of the initial state.

3.4 Higher order Markov chains

To enhance the sophistication of our analysis, we extend our focus to higher order Markov chains. While the basic Markov chain model considers transitions between states based on the immediate preceding state, higher order Markov chains incorporate dependencies on multiple previous states.

In a higher order Markov chain of order *m* the probability of transitioning to a future state X_k depends on the sequence of the *m* most recent states. Mathematically, this can be expressed as:

$$Pr(Xk = i \mid Xk - 1, Xk - 2, \ldots, Xk - m)$$

Understanding higher order Markov chains allows us to capture more intricate dependencies within the sequential data of our football team's outcomes. It provides a detailed perspective on how past matches influence future states, beyond the limitations of a first-order Markov chain.

The order -*k* of a Markov chain represents the number of positions that will be taken into account for the transition probability matrix. The order 0 matrix shows the number of appearances each value of a Markov chain has, position 0. An order 1 matrix represents the probability of having a value of the chain immediately after another specific value, the position -1 . Then higher order matrices appeal to the probability of getting some value in the chain after a certain combination of values of length -*k*. The order 2 will look back 2 positions.

3.5 Comparison between different order Markov chains

In order to select the optimal order for Markov models, we turn to Bayesian Information Criteria (BIC) as a valuable metric for model comparison. BIC is a statistical measure that balances the goodness of fit and model complexity, providing a principled approach to select the most appropriate model order.

For each Markov model with a specific order *m* the BIC is calculated using the formula:

$$BIC = -2 * LL + m * log(n)$$

Where LL is the fitted log-likelihood of the model, n is the length of the chain and m is the number of parameters (n^{k+1} - n^k).

To compare different order Markov models, we compute the BIC for each model and select the model with the lowest BIC value. The lower BIC indicates a better trade-off between model fit and complexity, therefore we will have the best predictor for the match outcomes for Girona FC.

4.Analysis results

4.1 Import and data filtering

The initial phase involves importing data from the dataset, specifically from an Excel file. To accomplish this task, we utilized the R library 'readxl'. This process resulted in three distinct dataframes. For the 18-19 season, Girona participated in a total of 44 games, with only 38 in La Liga. In the 22-23 season, they played 30 games, once again with 38 in La Liga. In the current season, the team is anticipated to play a minimum of 40 games, although only 13 have been in La Liga thus far.

4.2 Count matrices

Count matrices diverge from transition state matrices by revealing the actual frequency or number of occurrences for each state within the chain. Unlike transition state matrices, which highlight probabilities of state transitions, count matrices emphasize providing a clear count of how often each state appears in the sequence.

Here, we present the count matrices for the **2018-19 season** considering various values of k, where k represents the number of matches considered:

K=0

D	E	V
19	10	9

K=1

	D	E	V
D	11	3	4
E	5	2	3
V	3	4	2

- K=2

	D	E	V
D:D	7	1	2
D:E	2	2	1
D:V	2	0	1
E:D	1	1	1
E:E	1	1	0
E:V	2	0	2
V:D	2	1	1
V:E	1	1	1
V:V	0	2	0

Here, we present the count matrices for the **2022-23 season:**

- K=0

D	E	V
15	10	13

- K=1

	D	E	V
D	5	5	4
E	3	2	5
V	6	3	4

- K=2

	D	E	V
D:D	1	2	1
D:E	1	2	0
D:V	3	1	2
E:D	2	1	2
E:E	0	0	2
E:V	1	1	1
V:D	2	0	2
V:E	3	1	1
V:V	1	2	1

Here, we present the count matrices for the **2023-24 season:**

K=0

D	E	V
1	1	11

K=1

	D	E	V
D	0	0	1
E	0	0	1
V	1	0	9

K=2

	D	V
D:D	0	0
D:E	0	0
D:V	0	1
V:D	0	1
V:E	0	1
V:V	1	7

4.3 Transition probability matrices

In this section of the report, we discuss the construction of transition probability matrices.

Here, we showcase the transition probability matrices for the **2018-19 season**, considering various values of k, where k represents the number of matches considered:

K=0

D	E	V
0.5000000	0.2631579	0.2368421

K=1

	D	E	V
D	0.6111111	0.1666667	0.2222222
E	0.5000000	0.2000000	0.3000000
V	0.3333333	0.4444444	0.2222222

K=2

	D	E	V
D:D	0.7000000	0.1000000	0.2000000
D:E	0.4000000	0.4000000	0.2000000
D:V	0.6666667	0.0000000	0.3333333
E:D	0.3333333	0.3333333	0.3333333
E:E	0.5000000	0.5000000	0.0000000
E:V	0.5000000	0.0000000	0.5000000
V:D	0.5000000	0.2500000	0.2500000
V:E	0.3333333	0.3333333	0.3333333
V:V	0.0000000	1.0000000	0.0000000

Here, we present the transition probability matrices for the **2022-23 season:**

K=0

D	E	V
0.3947368	0.2631579	0.3421053

K=1

	D	E	V
D	0.4166667	0.4166667	0.3333333
E	0.3000000	0.2000000	0.5000000
V	0.4285714	0.2142857	0.2857143

K=2

	D	E	V
D:D	0.2500000	0.5000000	0.2500000
D:E	0.3333333	0.6666667	0.0000000
D:V	0.5000000	0.1666667	0.3333333
E:D	0.4000000	0.2000000	0.4000000
E:E	0.0000000	0.0000000	1.0000000
E:V	0.3333333	0.3333333	0.3333333
V:D	0.5000000	0.0000000	0.5000000
V:E	0.6000000	0.2000000	0.2000000
V:V	0.2500000	0.5000000	0.2500000

Here, we present the transition probability matrices for the **2023-24 season:**

K=0

D	E	V
0.07692308	0.07692308	0.84615385

K=1

	D	E	V
D	0	0	1.0
E	0	0	1.0
V	0.1	0	0.9

K=2

	D	V
D:D	*	*
D:E	*	*
D:V	0.000	1.000
V:D	0.000	1.000
V:E	0.000	1.000
V:V	0.125	0.875

In the matrix, where k=2, several values are observed as *. This is attributed to the absence of corresponding data for the given cases throughout the season.

4.4 BIC Statistics

To compute the Bayesian Information Criterion (BIC), we require the count matrix, the probability matrix, the length of the chain, and the number of parameters. Both matrices have already been obtained in preceding steps. The length of the chain is determined as the number of values minus the order of the Markov chain. The number of parameters is calculated as the total number of elements in the matrix minus the number of rows.

For the **2018-19 season**, we have obtained the following Bayesian Information Criterion (BIC) values for the different orders of Markov Chains.

n0 = 38; m0 = 3-1
BIC0 = 86.24129

n1 = 38-1; m1 = 9-3
BIC 1 = 94.97165

n2 = 38-2; m2 = 27-9
BIC2 = 124.7269

In this case, we observe that the order that best fits our model is 0, as the BIC for this order is the smallest value among the other BIC values.

For the **2022-23 season**, we have obtained the following Bayesian Information Criterion (BIC) values for the different orders of Markov Chains.

n0 = 38; m0 = 3-1
BIC0 =89.74983

n1 = 38-1; m1 = 9-3
BIC1 = 97.98919

n2 = 38-2; m2 = 27-9
BIC2 = 129.2836

In this case, we can also observe that the order that best fits our model is 0, as the BIC for this order is the smallest value among the other BIC values.

For the **2023-24 season**, we have obtained the following Bayesian Information Criterion (BIC) values for the different orders of Markov Chains. Before delving into the details, it is pertinent to mention that we have encountered certain complications related to the calculation of BIC values for this specific season. In the transition probability matrix of order 1, there are values that are zero due to the absence of corresponding situations. This poses challenges when calculating the BIC, as the logarithm is undefined, and the formula cannot be applied correctly. To address this issue, we have replaced all zero values with an infinitesimally small value (1e-10). Furthermore, Girona FC has only drawn one match throughout the entire season, the very first one; therefore, the Markov chain of order 2 lacks meaningful significance.

n0 = 13; m0 = 3-1
BIC0 = 60.10408

n1 = 13-1; m1 = 9-3
BIC1 = 23.89601

2 = 13-2; m2 = 27-9
BIC2 = /

In this case, we observe that the order that best fits our model is 1, as the BIC for this order is smaller than the BIC for order 0.

4.5 Origin of a given chain

We opted to analyze a random sequence of results, denoted as 'VDEED,' to determine its probable association with a particular season. Based on our projections, since the sequence includes only one victory, it is unlikely to correspond to the 2023-24 season. Consequently, we encountered the need to address the issue of adding 1e-10 to the formula when calculating the logarithm of the transition matrix, especially when the value equals 0.

```
## s18-19 // s23.24 = 66.18718

## s22-23 // s23-24 = 66.84396

## s18-19 // s22.23 = -0.6567795
```

5.Conclusions and discussion

Thanks to various matrices, such as the count matrix and the transition probability matrix, we can observe that in the 2018-19 season, there was a significant number of defeats. Using the first and second-order count matrices, we can identify that these losses often originated from a preceding defeat. For other result sequences, we observe more similar values among different event sequences.

In the second season, 2022-23, we note that all results are evenly distributed, and there is no discernible pattern repeating throughout this season.

For the 2023-24 season, a drastic performance shift is evident. We observe that the majority of matches end in victory, and consequently, there are only transitions from victory to victory when analyzing the different matrices.

When analyzing the BIC statistics to determine the best model for predicting match outcomes, particularly predicting a chain, we have found that the optimal order is 0 for the 2018-19 and 2022-23 seasons. This result indicates that previous results do not determine the outcome of the next match, contrary to our initial hypothesis. However, we remain firm in the belief that streaks and the factors mentioned in the introduction do influence the team's performance. Due to the small size of our result chains, we have not been able to confirm our initial hypothesis.

For the last season, the best order to describe our model is 1, but this is highly influenced by the significant number of consecutive victories achieved by the team, GIRONA FC.

Regarding the final analysis of our diagnostic, which concerns predicting the season to which a certain result sequence belongs, we have reached a significant result that aligns with the theory explained earlier. This suggests that a sequence with few victories does not belong to the last season. When comparing between the other two seasons, the result indicates that it is part of the 2022-23 season, thus aligning with our approach, as it has a distribution more similar to the 2022-23 season than the 2018-19 season.

To improve the work and the obtained results, we firmly believe that a larger sample size is necessary. Unfortunately, the Spanish football league has a limited number of match days per season, making it challenging to exclusively compare one season from one year to another. For example, additional studies could be conducted with longer periods, such as analyzing data over five seasons. However, in doing so, we must consider many more factors when analyzing, such as the team's roster, the

opponent's roster, the league they are in, salaries, the number of competitions the team participates in, the coach, among others.

6.Bibliography

- Theory

https://en.wikipedia.org/wiki/Markov_model

https://medium.com/modelos-estoc%C3%A1sticos-teor%C3%ADa-de-colas-y-modelos-de/modelos-estoc%C3%A1sticos-teor%C3%ADa-de-colas-y-modelos-de-markov-18c0100ad077

https://www.sciencedirect.com/topics/mathematics/step-transition-probability

https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120607B.pdf

- Data

https://fbref.com/es/equipos/9024a00a/2023-2024/Estadisticas-de-Girona

https://fbref.com/es/equipos/9024a00a/2022-2023/Estadisticas-de-Girona

https://fbref.com/es/equipos/9024a00a/2018-2019/Estadisticas-de-Girona

7.Appendix

#Libraries

```
library(readxl)
library(ggplot2)
library(seqinr)
```

#Importing and filtering datasets

```
s18.19 <- read_excel('./18-19.xlsx')
length(s18.19$Fecha)
s18.19 <- s18.19[s18.19$Comp == 'La Liga',]
length(s18.19$Fecha)

s22.23 <- read_excel('./22-23.xlsx')
length(s22.23$Fecha)
s22.23 <- s22.23[s22.23$Comp == 'La Liga',]
length(s22.23$Fecha)

s23.24 <- read_excel('./23-24.xlsx')
length(s23.24$Fecha)
s23.24 <- na.omit(s23.24[s23.24$Comp == 'La Liga',])
length(s23.24$Fecha)
```

#Generating the plots

```
ggplot(s18.19, aes(x = Hora, fill = Resultado)) +
  geom_bar(show.legend = FALSE) +
  labs(title = "2018-19",
       x = "Hour",
       y = "Number of matches") +
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))

ggplot(s22.23, aes(x = Hora, fill = Resultado)) +
  geom_bar(show.legend = FALSE) +
  labs(title = "2022-23",
       x = "Hour",
       y = "Number of matches") +
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))
```

```
ggplot(s23.24, aes(x = Hora, fill = Resultado)) +
  geom_bar(show.legend = FALSE) +
  labs(title = "2023-24",
       x = "Hour",
       y = "Number of matches") +
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))
```

```
ggplot(s18.19, aes(x = Día, fill = Resultado)) +
  geom_bar(show.legend = FALSE) +
  labs(title = "2018-19",
       x = "Day",
       y = "Number of matches") +
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))

ggplot(s22.23, aes(x = Día, fill = Resultado)) +
  geom_bar(show.legend = FALSE) +
  labs(title = "2022-23",
       x = "Day",
       y = "Number of matches") +
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))

ggplot(s23.24, aes(x = Día, fill = Resultado)) +
  geom_bar(show.legend = FALSE) +
  labs(title = "2023-24",
       x = "Day",
       y = "Number of matches") +
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))
```

```
ggplot(s18.19, aes(x = Resultado, y = GF, fill = Resultado)) +
  geom_boxplot(show.legend = FALSE) +
  stat_summary(fun = mean, geom = 'point',  size = 3, color = "black",
show.legend = FALSE) +
  labs(title = "2018-19",
       x = "Result",
       y = "GF") +
```

```
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))

ggplot(s22.23, aes(x = Resultado, y = GF, fill = Resultado)) +
  geom_boxplot(show.legend = FALSE) +
  stat_summary(fun = mean, geom = 'point',  size = 3, color = "black",
show.legend = FALSE) +
  labs(title = "2022-23",
       x = "Result",
       y = "GF") +
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))

ggplot(s23.24, aes(x = Resultado, y = GF, fill = Resultado)) +
  geom_boxplot(show.legend = FALSE) +
  stat_summary(fun = mean, geom = 'point',  size = 3, color = "black",
show.legend = FALSE) +
  labs(title = "2023-24",
       x = "Result",
       y = "GF") +
  scale_fill_manual(values = c("V" = "#669966", "E" = "#DAA520", "D" =
"#CC3333")) +
  theme(panel.background = element_rect(fill = "#F0F0F0"))
```

 # Building the transition matrices and Computing the BIC statistics

```
s18.19_results <- s18.19$Resultado

s18.19_results_0c <- count(s18.19_results, 1, alphabet = c('V', 'E',
'D'))
s18.19_results_0c
s18.19_results_0p <- s18.19_results_0c/length(s18.19_results)
s18.19_results_0p
n <- length(s18.19_results)
m <- 3 - 1
BIC0 <- -2*sum(mapply(function(x, y) x * log(y), s18.19_results_0c,
s18.19_results_0p))+ m*log(n)
BIC0

a <- count(s18.19_results, 2, alphabet = c('V', 'E', 'D'))
s18.19_results_1c <- matrix(a, 3, 3, byrow=TRUE, dimnames =
list(c("D","E","V"),c("D","E","V")))
s18.19_results_1c
s18.19_results_1p <-
s18.19_results_1c[,]/(s18.19_results_1c[,1]+s18.19_results_1c[,2]+s18.19
_results_1c[,3])
s18.19_results_1p
n <- length(s18.19_results) - 1
m <- 9 - 3
BIC1 <- -2*sum(mapply(function(x, y) x * log(y), s18.19_results_1c,
s18.19_results_1p))+ m*log(n)
BIC1

n <- length(s18.19_results)
xt=factor(s18.19_results[3:n])
xt_1=factor(s18.19_results[2:(n-1)])
xt_2=factor(s18.19_results[1:(n-2)])
s18.19_results_2c <- table(xt_1:xt_2,xt)
s18.19_results_2c
s18.19_results_2p <-
s18.19_results_2c[,]/(s18.19_results_2c[,1]+s18.19_results_2c[,2]+s18.19
_results_2c[,3])
s18.19_results_2p
n <- n -2
m <- 27 - 9
BIC2 <- -2*sum(mapply(function(x, y) x * log(y), s18.19_results_2c,
s18.19_results_2p + 1e-10))+ m*log(n)
BIC2
```

```
s22.23_results <- s22.23$Resultado

s22.23_results_0c <- count(s22.23_results, 1, alphabet = c('V', 'E',
'D'))
s22.23_results_0c
s22.23_results_0p <- s22.23_results_0c/length(s22.23_results)
s22.23_results_0p
n <- length(s22.23_results)
m <- 3 - 1
BIC0 <- -2*sum(mapply(function(x, y) x * log(y), s22.23_results_0c,
s22.23_results_0p))+ m*log(n)
BIC0


a <- count(s22.23_results, 2, alphabet = c('V', 'E', 'D'))
s22.23_results_1c <- matrix(a, 3, 3, byrow=TRUE, dimnames =
list(c("D","E","V"),c("D","E","V")))
s22.23_results_1c
s22.23_results_1p <-
s22.23_results_1c[,]/(s22.23_results_1c[,1]+s18.19_results_1c[,2]+s22.23
_results_1c[,3])
s22.23_results_1p
n <- length(s22.23_results) - 1
m <- 9 - 3
BIC1 <- -2*sum(mapply(function(x, y) x * log(y), s22.23_results_1c,
s22.23_results_1p))+ m*log(n)
BIC1

n <- length(s22.23_results)
xt=factor(s22.23_results[3:n])
xt_1=factor(s22.23_results[2:(n-1)])
xt_2=factor(s22.23_results[1:(n-2)])
s22.23_results_2c <- table(xt_1:xt_2,xt)
s22.23_results_2c
s22.23_results_2p <-
s22.23_results_2c[,]/(s22.23_results_2c[,1]+s22.23_results_2c[,2]+s22.23
_results_2c[,3])
s22.23_results_2p
n <- n - 2
m <- 27 - 9
BIC2 <- -2*sum(mapply(function(x, y) x * log(y), s22.23_results_2c,
s22.23_results_2p + 1e-10))+ m*log(n)
BIC2
```

```
s23.24_results <- s23.24$Resultado

s23.24_results_0c <- count(s23.24_results, 1, alphabet = c('V', 'E',
'D'))
s23.24_results_0c
s23.24_results_0p <- s23.24_results_0c/length(s23.24_results)
s23.24_results_0p
n <- length(s23.24_results)
par <- 3 - 1
BIC0 <- -2*sum(mapply(function(x, y) x * log(y), s23.24_results_0c,
s23.24_results_0p))+ m*log(n)
BIC0

a <- count(s23.24_results, 2, alphabet = c('V', 'E', 'D'))
s23.24_results_1c <- matrix(a, 3, 3, byrow=TRUE, dimnames =
list(c("D","E","V"),c("D","E","V")))
s23.24_results_1c
s23.24_results_1p <-
s23.24_results_1c[,]/(s23.24_results_1c[,1]+s23.24_results_1c[,2]+s23.24
_results_1c[,3])
s23.24_results_1p
n <- length(s23.24_results) - 1
m <- 9 - 3
BIC1 <- -2*sum(mapply(function(x, y) x * log(y), s23.24_results_1c,
s23.24_results_1p + 1e-10))+ m*log(n)
BIC1

n <- length(s23.24_results)
xt=factor(s23.24_results[3:n])
xt_1=factor(s23.24_results[2:(n-1)])
xt_2=factor(s23.24_results[1:(n-2)])
s23.24_results_2c <- table(xt_1:xt_2,xt)
s23.24_results_2c
s23.24_results_2p <-
s23.24_results_2c[,]/(s23.24_results_2c[,1]+s23.24_results_2c[,2])
s23.24_results_2p
n <- length(s23.24_results) - 2
m <- 27 - 9
```

#Computing the Log-Likelihood comparison method

```
sequence <- c("V", "D", "E", "E", "D")
v <- c('D'=1,'E'=2,'V'=3)
n <- length(random_sequence) - 1
log_likeilhood <- c()
for (i in seq(n)){
  from <- v[random_sequence[i]]
  to <- v[random_sequence[i+1]]
  log_likeilhood[i] <-
log(s18.19_results_1p[from,to]/(s23.24_results_1p[from,to]+ 1e-10))
}
sum(log_likeilhood)


log_likeilhood <- c()
for (i in seq(n)){
  from <- v[random_sequence[i]]
  to <- v[random_sequence[i+1]]
  log_likeilhood[i] <-
log(s22.23_results_1p[from,to]/(s23.24_results_1p[from,to]+ 1e-10))
}
sum(log_likeilhood)


log_likeilhood <- c()
for (i in seq(n)){
  from <- v[random_sequence[i]]
  to <- v[random_sequence[i+1]]
  log_likeilhood[i] <-
log(s18.19_results_1p[from,to]/(s22.23_results_1p[from,to]+ 1e-10))
}
sum(log_likeilhood)
```