*Submitted to the Annals of the Department of Statistics and Operations Research*

By Nikola Rusinov ,

# PROBABILISTIC MODELING OF BASKETBALL EVENTS WITH MARKOV CHAINS

*University of Vienna*

This work presents an analysis within event sequences in basketball games using Play-by-Play data from the 2024-2025 Season of the **Second Austrian Bundesliga**. It explores the most common transitions between in-game events through probabilistic models along with discrete-time Markov chains. Two variants of the Markov model are developed and compared: a **first-order** model based on individual events, and a **second-order** model that incorporates combinations of two consecutive events. ***EvtRelFreq*** model was built for prediction of the next event on relative frequencies. The second developed model is a ***DurSample*** used to simulate the duration between consecutive events.

In addition, principal component analysis was applied to the transition matrices in order to visualize and cluster event sequences. The PCA-based exploration complements the Markov framework by highlighting patterns in event transitions that align closely with actual basketball strategies.

Simulations do validate all of the constructed models. Real game data can be compared to all of the models too. The potential of probabilistic models is demonstrated by the results, for describing and for predicting dynamics of basketball games and for laying the groundwork for tactical simulations.

**1. Introduction.** Basketball is a dynamic and fast-paced sport, where sequences of discrete events such as passes, shots, fouls, rebounds, and turnovers unfold in rapid succession. Modern basketball analytics increasingly rely on detailed *Play-by-Play* (PbP) data, which records each event in the game chronologically and with high granularity. Such data provide an opportunity to move beyond traditional box score statistics and instead model the temporal structure and dependencies between actions in a game.

This thesis focuses on modeling the behavior of a basketball team using Play-by-Play data from the entire season of 2024 - 2025. We explore how sequences of in-game events can be understood and predicted using probabilistic methods, specifically discrete-time Markov chains. The first-order Markov model assumes that the probability of a future event depends only on the current state (i.e., the last observed event), which allows for tractable analysis and simulation of possessions and transitions. On the other hand, the second-order Markov model describes better the sequential nature of the basketball game, as it constructs the probabilities of the next events, based on the previous two events.

Inspired by the methodology in (Vračar, Štrumbelj and Kononenko, 2016), we build and implement models such as the ***EvtRelFreq*** model, which predicts the next event based on empirical transition frequencies, and the ***DurSample*** model, which samples durations between successive events based on historical distributions. Aside from probabilistic modeling, this thesis addresses the issue of clustering event sequences, which provides an unsupervised perspective on basketball data.

Clustering is especially beneficial when attempting to find latent structures inside

the game flow. Clustering, which groups comparable event sequences or transition patterns, can expose recurring strategies, differentiate between offensive and defensive approaches, and identify team-specific behaviors. The merger of PCA and k-prototype allows us to investigate the structural organization of basketball possessions.

The ultimate goal of this work is to develop a framework that allows us to understand basketball possessions and to be able to simulate them realistically.

## 2. Data Structure and Preprocessing.

2.1. *Original Data set.* The Play-by-Play (PbP) data used in this work originate from official game reports from the "Austrian Basketball Second League". Specifically, from the entire 2024-2025 season. The Play-by-Play data was originally provided in PDF format, containing textual summaries of in-game events. These PDFs list events sequentially, split by quarters, and include information such as player actions, time remaining, and current score. To enable computational analysis, the PDFs were converted into machine-readable `.csv` files using a custom-built Python script.

Due to the nature of the PDF-to-CSV transformation, certain structural and contextual data was lost in the process. In particular, the information about which team played as home or away, the names of the competing teams, and the exact indicators for the start and end of games and quarters were not preserved in the CSV format.

The original dataset contains the following essential elements:

- Event time (relative to the quarter),
- Two columns with event description - one for the home and one for the away team (e.g., `Mister Muster (23) pts jump shot made`, `turnover`, `free throw missed`, etc.),
- The actual score at the time of the event,
- Score differential at the time of the event.

2.2. *Data set for the Simulation Problem.* To make the data suitable for probabilistic modeling and feature extraction, a series of preprocessing steps were performed. These include:

1. Removal of player names, jersey numbers, and non-possession-relevant events such as substitutions, jump balls, and assists.
2. Unification of event, the 124 unique labels into a fixed vocabulary of 22, with a prefix telling, if the team is home or away (e.g., `h 2pts made`, `a 3pts missed`, `h turnover`, `a foul`.
3. Mapping each event to its corresponding `GameID` and `Quarter`, and calculating elapsed time in seconds.

After completing all pre-processing steps, we obtained a clean event-based data set with a total of 50141 events (rows) and 5 variables (columns). Each row corresponds to a single action in the game. The data set totals 154 unique games

2.3. *Data set for the Clustering Problem.* The specific nature of the second problem requires some additional data processing. Our goal is to create a transition matrix for every single team; that's why we add 3 new columns. The factors **HomeTeam** and **AwayTeam** describe the two teams playing on every single row and whether the teams are guests or if they are the host. The last addition is the bivariate variable **homeAway**, which takes the value of 1 if the actor of the row-wise current event is performed by a team that is playing at its home venue, and 0 otherwise. This variable is dependent on the prefix of the current event. The data set we use for this problem ends up with dimensions of $50410 \times 8$

## 3. Methodology.

3.1. *Markov Process.* The theory of stochastic processes contains the theory of Markov processes as a special case. A stochastic process may be defined as a family of random variables.

$$\{X(t) : t \in T\}$$

A stochastic process may be of four types depending on the character of the parameter set $T$ and the random variable $X(t)$, both can be either *continuous* or *discrete*. The range of $X(t)$ is called the state space of the process. A stochastic process which has a discrete parameter space is called a stochastic chain. (*Dynkin[21]*).

The feature of a Markov process which distinguishes it from the others, is its dependency structure. We define $\{t_n\}$ be a sequence of values in $T$. We may assume $\{t_n\}$ is non- decreasing, then the conditional probability expression below specifies the Markov property of order $m$.

$$\begin{aligned} &P\big[X(t_n) \le x(t_n) \;\mid\; X(t_1) \le x(t_1), X(t_2) \le x(t_2), \ldots, X(t_{n-1}) \le x(t_{n-1})\big] \\ &= P\big[X(t_n) \le x(t_n) \;\mid\; X(t_{n-m}) \le x(t_{n-m}), X(t_{n-m-1}) \le x(t_{n-m-1}), \ldots, X(t_{n-1}) \le x(t_{n-1})\big] \end{aligned} \tag{3.1}$$

where $n \ge 2 \quad$ and $\quad 1 \le m \le n$.

If the stochastic process satisfies (3.1) and both the state space and the parameter space are discrete, the process is considered a discrete Markov chain of order $m$. In this scenario the definition of the **Markov property** becomes:

$$\begin{aligned} &P\big[X(t_n) = x(t_n) \;\mid\; X(t_1) = x(t_1), X(t_2) = x(t_2), \ldots, X(t_{n-1}) = x(t_{n-1})\big] \\ &= P\big[X(t_n) = x(t_n) \;\mid\; X(t_{n-m}) = x(t_{n-m}), X(t_{n-m-1}) = x(t_{n-m-1}), \ldots, X(t_{n-1})x(t_{n-1})\big] \end{aligned} \tag{3.2}$$

A **finite Markov Chain** is a stochastic chain, which satisfies (3.2) also fulfills the property, that for any $t \in T$, the range of $X(t)$ is a finite set.

According to Rodda (1969), for finite Markov chains of order one, the Markov property could be stated in words as, "The probability, that the process is in state

$x(t_n)$, given that it was in state $x(t_{n-1})$, etc, is the probability that the process occupies state $x(t_n)$ given at $t_{n-1}$ it was in state $x(t_{n-1})$."

In the context of this work, the above mentioned stochastic process

$$\{X(t) : t \in T\}$$

describes the sequence of events, derived from the Play-by-play data. Hence the state space ranges across finite and discrete set of possible events. This set includes all of the 22 unique events, acquired after the data preprocessing.

$$\mathcal{S} = \{a\_2\_made, h\_2\_made, a\_foul, \dots\}$$

3.2. *First-Order Markov Process.* When we discuss Markov chains, an ordered pair of states

$$(x(t_n), x(t_n + 1))$$

is called a **transition**. In such a transition the first element of the pair represents the state the process is in, at the beginning of an interval. A simple notation for the transition probabailites can be used.

$$P[s_j \mid s_i] = p_{ij}, \quad s_i, s_j \in \mathcal{S}, \quad i, j = 1, 2, \dots, S \tag{3.3}$$

$p_{ij}$ is the conditional probability that at $t_{n+1}$ the system occupied state $s_j$, given that at $t_n$ it occupied $s_i$. The transition probabilities determine the fundamental properties of the Markov chain. Those finite probabilites form an $S \times S$ matrix.

In Table 1 we can see the transition matrix of the dataset from the 2.Bundesliga, dependent on the first order Markov property. This matrix summarizes the dynamic of the game and reflects how likely one event is to follow another within the observed games.

| From | To | Count | Probability | Team | H/A |
|---|---|---|---|---|---|
| a 2 made | a 2 made | 29 | 0.0100 | Austrian Ballers | a |
| a 2 made | a 2 missed | 9 | 0.0031 | Austrian Ballers | a |
| a 2 made | a 3 made | 11 | 0.0038 | Austrian Ballers | a |
| a 2 made | a 3 missed | 10 | 0.0035 | Austrian Ballers | a |
| a 2 made | a foul | 475 | 0.1642 | Austrian Ballers | a |
| a 2 made | a jump ball won | 7 | 0.0024 | Austrian Ballers | a |
| a 2 made | a rebound defensive | 28 | 0.0097 | Austrian Ballers | a |
| a 2 made | a turnover | 7 | 0.0024 | Austrian Ballers | a |

TABLE 1
*Sample of the First Order Markov Transitional Probability Matrix*

The addition of the matrix' last two columns has significantly increased the information provided by the data frame. The column named "Team" provides the team who is acting during the sequence. With the last column we gain information about the fact of whether the team acting was on the receiving end or not. For every possible sequence there are two rows in the transition matrix. One considering that the acting team is home during the event and one more that interprets it as a guest. With these additions, one can derive special transition matrices unique for every team from the second Bundesliga, taking into consideration the venue.

3.3. *Second-Order Markov Process.* Another example of information, contained in the transition probability matrix are the **Second Order Transition probabilities.** If the system is curently in state $s_i$, then the conditional probability that it will proceed to state $s_j$ and afterwards to state $s_k$ is

$$P\big[s_j|s_i\big]P\big[s_k|s_i,s_j\big] = P\big[s_j|s_i\big]P\big[s_k|s_j\big] \tag{3.4}$$

Which is simply $p_{ij}p_{jk}$. Hence to get the value of $p_{ik}$, we need to sum over the index j accounts for all possible intermediate states.

$$p_{ik} = \sum_{j=1}^{S} p_{ij}p_{jk} \tag{3.5}$$

The conditional probability that the system will be at state $S_k$ two units late is the $(i,k) - th$ entry in the square of the first order transitional matrix.

| Prev1 | Prev2 | Next | Count | Probability | Team | H/A |
|---|---|---|---|---|---|---|
| h 2 made | a 2 made | h 3 missed | 8 | 0.2759 | Mistelbach Mustangs | h |
| h 2 made | a 2 missed | a rebound offensive | 9 | 0.2500 | Mistelbach Mustangs | h |
| h 2 made | a 2 missed | h rebound defensive | 27 | 0.7500 | Mistelbach Mustangs | h |
| h 2 made | a 3 made | a foul | 4 | 0.2500 | Mistelbach Mustangs | h |
| h 2 made | a 3 missed | a rebound offensive | 5 | 0.1852 | Mistelbach Mustangs | h |
| h 2 made | a 3 missed | h rebound defensive | 22 | 0.8148 | Mistelbach Mustangs | h |
| h 2 made | a foul | h foul | 10 | 0.8333 | Mistelbach Mustangs | h |
| h 2 made | a timeOut | a 2 missed | 5 | 0.3333 | Mistelbach Mustangs | h |
| h 2 made | a timeOut | h foul | 6 | 0.4000 | Mistelbach Mustangs | h |

TABLE 2
*Sample of the second order transitional Matrix*

In table 2 one can find the second order Transition Probability matrix. The first two columns describe the **"past"** two events on which the probability for the occurrence of the **"next"** one depends. In this work we would like to investigate if a framework with a second-order Markov chain would be more useful for the goal of realistically describing a basketball game. The large number of combinations of 3 variables that could be derived for building such a system would suggest a really broad analysis of a possession. The addition for the last two column, allows us to create team-specific transitional matrices. Analog to the first order case.

This structure is suitable for modeling event sequences where only the most recent event influences the next, and it requires fewer parameters.

3.4. *State Space of the Markov Model.* As mentioned previously, the state space of the Markov process is the range of the random variable $X(t) =: X$. In our case the state space consists of every possible end of possession play, derived from the play-by-play data set.

3.5. *Joint Distribution of States and Transitions.* In the context of Markov chains, the evolution of a sequence of events is governed by a probabilistic structure over sequences of states. Let $X_t$ denote the random variable representing the in-game event that occurs at time step $t$, and let $T$ be the total number of observed

events in a quarter (600 seconds). The sequence $X_0, X_1, \ldots, X_T$ thus represents the ordered progression of events in a part.

For a sequence of discrete states $X_0, X_1, \ldots, X_T$, the joint probability under a **first-order Markov assumption** is given by:

$$P(X_0, X_1, \ldots, X_T) = P(X_0) \prod_{t=1}^{T} P(X_t \mid X_{t-1})$$

This expression factorizes the joint distribution into a product of conditional probabilities that depend only on the immediately preceding state.

For the **second-order Markov chain**, the dependency extends to two previous states. The joint distribution then factorizes as:

$$P(X_0, X_1, \ldots, X_T) = P(X_0, X_1) \prod_{t=2}^{T} P(X_t \mid X_{t-1}, X_{t-2})$$

Here, the initial distribution $P(X_0, X_1)$ defines the starting condition, and each subsequent transition depends on the last two observed events. This structure allows the model to incorporate richer contextual dependencies, which may be particularly relevant in structured domains like basketball, where tactical sequences (e.g., rebound $\rightarrow$ fast break $\rightarrow$ 3pts shot) are common.

In practice, these conditional probabilities are estimated empirically from the observed Play-by-Play data.

3.6. *Modeling Event Sequences.* Each state in our Markov chain corresponds to a specific game event (e.g., `h_2pts made`, `a_turnover`), and the transition from one state to the next reflects the natural flow of a basketball game.

The **MEvt** model aims to estimate the probability of the next event given the current one. Formally, for a sequence of states $\{X_i\}_{i \in \mathbb{N}_0}$, we define:

$$P(X_{i+1} = y \mid X_i = x) = f(y \mid x)$$

As discussed in the previous section, this distribution is factorized into two components. In the **MEvt** framework, we focus on the first term, $f_{Y(\text{Evt})|X}$, which models the distribution over the next event type, given the current state. In its simplest form - the ***EvtRelFreq*** model - this probability is estimated using relative frequencies extracted from the observed data:

$$P(y \mid x) = \frac{C(x, y)}{\sum_k C(x, s_k)}$$

Here, $C(x, y)$ denotes the number of times event $y$ immediately follows event $x$ in the dataset. This model assumes the Markov property, meaning that the probability of the next event depends only on the current one, and not on earlier events.

To implement this model, we created a shifted column `next_event` within our dataset, so it contained the subsequent event with each row. We constructed, during simulation, an empirical transition matrix via grouping all observed $(event, next\_event)$ pairs and counting their frequencies.

Realistic event sequences can be simulated thanks to this matrix. It captures all of the stochastic structure within basketball possessions since we sample iteratively for the next event that is based on one that is current.

Table 2 shows a subset for the transition matrix that focuses on away-team offensive events, which depicts the empirical transition probabilities that the ***EvtRelFreq*** model estimated. Each cell indicates the estimated probability for the row's event transitioning to the column's.

3.7. *Modeling Event Durations and Extended State Representation.* While the Markov chain models the probabilistic transitions between discrete basketball events, it does not inherently capture the time elapsed between those events. To address this, a separate model is used to describe the *duration* between transitions.

*Duration Modeling Framework..* Let $\mathcal{S}$ denote the state space of discrete basketball events, as defined previously. For every ordered pair of events $(s_i, s_j) \in \mathcal{S} \times \mathcal{S}$ representing a transition from event $s_i$ to event $s_j$, we associate a random variable:

$$D_{ij} \sim \mathcal{D}(s_i, s_j)$$

where $D_{ij}$ is the random variable representing the **duration** (in seconds) between event $s_i$ and event $s_j$, and $\mathcal{D}(s_i, s_j)$ denotes the empirical distribution of durations for this transition observed in the Play-by-Play data.

Let $t_k$ denote the game time (in seconds remaining) at the occurrence of the $k$-th event. Then, during simulation, the time update between successive events follows:

$$t_{k+1} = t_k - d_k \quad \text{where} \quad d_k \sim \mathcal{D}(s_k, s_{k+1})$$

Here:

- $t_k$ is the remaining time before the $k$-th event,
- $s_k$ is the $k$-th event,
- $s_{k+1}$ is the event sampled from the Markov transition model,
- $d_k$ is the sampled duration for the transition from $s_k$ to $s_{k+1}$.

In practice, $\mathcal{D}(s_i, s_j)$ is represented by a set of empirical samples collected for each observed event transition in the dataset. If no samples are available for a given pair, a fallback mechanism is used (e.g., drawing from a general or smoothed distribution).

3.8. *Simulating a Quarter.* Once both the ***EvtRelFreq*** and ***DurSample*** models have been constructed, we can simulate a full basketball quarter as a sequence of states. Each state consists of an event and its contextual information, and transitions are generated iteratively according to the Markov framework described earlier.

We implement a simulation procedure closely following Algorithm 1 from (Vračar, Štrumbelj and Kononenko, 2016), which is summarized below.

**Input:**

- $s$ – initial state (e.g., `h_jump ball won`, with starting time = 0),
- $M_{\text{Evt}}$ – event model based on empirical transition probabilities,

**Algorithm 1** Simulate a Basketball Quarter

**Require:** $s$ – initial state
$M_{\text{Evt}}$ – model for $f_Y(\text{Evt} \mid X)$
$M_{\text{Dur}}$ – model for $f_Y(\text{Dur} \mid X, Y_{\text{Evt}})$
**Ensure:** *outSeq* – a generated sequence of states
**function** SIMULATEQUARTER($s$, $M_{\text{Evt}}$, $M_{\text{Dur}}$)
    $outSeq \leftarrow \emptyset$
    $x \leftarrow s$
    **while** $x(\text{Time}) \geq 0$ **do**
        $y \leftarrow x$
        $y(\text{Evt}) \leftarrow$ sample from $M_{\text{Evt}}(x)$
        $dur \leftarrow$ sample from $M_{\text{Dur}}(x, y(\text{Evt}))$
        $y(\text{PtsDiff}) \leftarrow \text{update}(y(\text{PtsDiff}), y(\text{Evt}))$
        $y(\text{Time}) \leftarrow \text{update}(y(\text{Time}), dur)$
        $outSeq \leftarrow \text{append}(outSeq, y)$
        $x \leftarrow y$
    **end while**
    **return** *outSeq*
**end function**

- $M_{\text{Dur}}$ – duration model based on empirical time intervals.

**Output:** a simulated sequence of states representing one quarter.

This simulation procedure produces realistic sequences of play that follow the empirical structure of real data. It can be extended to full games by chaining four quarters with resets between them.

3.9. *Clustering.* Clustering algorithms are intended to group similar observations without relying on preexisting classifications. In the case of our dataset, this means that event sequences, state changes, and team-specific behaviors can be grouped into meaningful categories. This unsupervised approach enables us to detect hidden features in data that would not be obvious using basic frequency counts or transition matrices.Teams or players with similar styles of play can be grouped together, highlighting strategic similarities and differences across the league.

3.9.1. *Principle Component Analysis.* Jolliffe (2011) For each type of transition, we estimate its empirical probability by relative frequencies:

$$\hat{p}_{ij} = \frac{\text{count of transitions from } i \text{ to } j}{\text{total transitions from } i}.$$

in our problem there are multiple teams, that's why we compute the mean transition probability as

$$\bar{p}_{ij} = \frac{1}{T} \sum_{t=1}^{T} p_{ij}^{(t)},$$

where $p_{ij}^{(t)}$ is the transition probability for team $t$.

Thus, $\bar{p}_{ij}$ represents the average transition probability across all $T$ teams (or contexts).

Let $x \in \mathbb{R}^K$ be a feature vector, which has the value of $K$ taking every possible transition. In the first-order Markov model, we have every combination of pairs of the original state space, which have s values. Hence for the value of K we have:

$$K_1 o = S^2.$$

Analog we can derive the Value of K for the second order Markov Process. We need every possible combination of triplets, hence:

$$K_2 o = S^3.$$

The next step is to choose a matrix $U$, with orthonormal columns, such that the projection

$$UU'(p_{ij} - \bar{p}_{ij}), U \in \mathbb{R}^{T \times K}$$

approximates $x$ the best way possible.

*Feature dimension..* Let $S$ denote the number of distinct event types (states). In a first-order Markov representation we include all ordered pairs $(i \to j)$, hence the feature dimension is

$$K_1^o = S^2.$$

In a second-order Markov representation we include all ordered triples $(i, \ell \to j)$, hence

$$K_2^o = S^3.$$

In both cases, let $K \in \{K_1^o, K_2^o\}$ denote the resulting feature dimension, and $x \in \mathbb{R}^K$ the vectorized transition probabilities (e.g., $x = \mathrm{vec}(p_{ij})$ or $x = \mathrm{vec}(p_{i\ell j})$).

*Data matrix and centering..* Collect $n$ observations (e.g., teams, games, or contexts) as the rows of $X \in \mathbb{R}^{n \times K}$, and let

$$\mu = \frac{1}{n} \sum_{r=1}^{n} X_{r\cdot} \in \mathbb{R}^K, \qquad X_c = X - \mathbf{1}\, \mu^\top$$

be the column-mean vector and the centered data matrix.

*Sample covariance and the PCA optimization..* Let the sample covariance be

$$S_X = \frac{1}{n-1} X_c^\top X_c \ \in \ \mathbb{R}^{K \times K}.$$

The first principal component (PC1) loading vector $u_1 \in \mathbb{R}^K$ solves

$$u_1 = \arg\max_{\|u\|=1} \ u^\top S_X u,$$

i.e., it maximizes the variance of the projected data $X_c u$. By the Rayleigh–Ritz theorem, $u_1$ is the eigenvector of $S_X$ associated with the largest eigenvalue $\lambda_1$:

$$S_X u_1 = \lambda_1 u_1, \qquad \lambda_1 = \max \mathrm{eig}(S_X).$$

The PC1 score vector (coordinates of the $n$ rows in the PC1 direction) is

$$z_1 \;=\; X_c u_1 \;\in\; \mathbb{R}^n.$$

The second principal component (PC2) loading vector $u_2$ solves

$$u_2 = \arg \max_{\|u\|=1,\; u^\top u_1 = 0} u^\top S_X u,$$

i.e., it maximizes variance subject to orthogonality to $u_1$. Hence $u_2$ is the eigenvector of $S_X$ associated with the second-largest eigenvalue $\lambda_2$:

$$S_X u_2 = \lambda_2 u_2, \qquad \lambda_2 = \text{second largest eigenvalue of } S_X,$$

with PC2 scores $z_2 = X_c u_2 \in \mathbb{R}^n$.

3.9.2. *K-Prototypes Clustering.* In order to cluster event transitions with both categorical and numerical attributes, we employed the **k-prototypes** algorithm. This method extends the classical k-means algorithm and the k-modes algorithm Huang (1998) by allowing the simultaneous use of categorical and continuous variables within the same clustering procedure .

The objective of the k-prototypes algorithm is to partition the data into $k$ clusters such that the similarity within clusters is maximized and the dissimilarity between clusters is minimized. This is achieved by minimizing the cost function

$$P(W, Q) = \sum_{l=1}^{k} \left( P_l^r + P_l^c \right),$$

where $W$ represents the assignment of objects to clusters and $Q$ denotes the set of cluster prototypes.

With term $P_l^r$ measures the total squared Euclidean distance between the numeric attributes of the objects assigned to cluster $l$ and the corresponding numeric part of the prototype. It ensures that numeric variables are grouped around their mean value, as in the classical k-means algorithm.

The term $P_l^c$ measures the dissimilarity for categorical attributes, expressed as the number of mismatches between the categorical values of the objects in cluster $l$ and the categorical mode of the prototype. The weight $\gamma$ balances the relative influence of numeric and categorical variables.

By minimizing $P(W, Q)$, the algorithm simultaneously reduces the variance of numeric attributes within clusters and the number of mismatches for categorical attributes. The resulting prototypes therefore represent the most typical combination of numeric and categorical values for each cluster.

## 4. Experiment.

4.1. *Simulation.* To evaluate the predictive quality of the ***EvtRelFreq*** model, we use the ***Mean Brier Score*** as the primary evaluation metric. The Brier Score is a proper scoring rule that measures the accuracy of probabilistic predictions. It quantifies the squared difference between the predicted probability distribution and the actual outcome. Suppose that on each of $n$ occasions an event, or in our case a play can occur in only one of $S$ possible classes or possession stopers. On occasion $i$, the forecast probabilities are shown as follows:

$$p_{i1}, p_{i2}, \ldots, p_{iS},$$

they denote the probability that the event will occur in classes $1, 2, \ldots, S$, respectively. The $S$ classes are chosen to be mutually exclusive and exhaustive, so that

$$\sum_{u=1}^{S} p_{iu} = 1, \qquad i = 1, 2, \ldots, n.$$

The second part of the formula is a marker for the effect of whether the event actually happened or not. The dichtonomous variable $o_k$ accepts the value of one, if the event actually occurs, and 0 if it doesn't. The difference between the actual result and the forecasted probability is then squared and averaged across all forecasting instances $K$. We are conducting a mean squared error estimation of the forecast.

$$\text{The Brier score is given by } := \frac{1}{S} \sum_{s=1}^{S} (p_s - o_s)^2$$

(BRIER, 1950)

In this work we want to learn about the reliance of the predictions. With a derived Brier score for each class separately, we gain understanding of the fact of which events our model predicts well and with which ones it faces difficulties. This gives us a solid base on which possibilities for future improvements can be easily located. It also avoids the masking effect that can occur in the overall average, where poor performance on rare events may be hidden by good performance on frequent ones.

TABLE 3
*Mean Brier Scores by Event Class*

| Class | Mean Brier | SD | Count |
|---|---|---|---|
| a_timeout | 0.052 | 0.004 | 73 |
| h_timeout | 0.051 | 0.004 | 73 |
| a_3pts made | 0.050 | 0.004 | 167 |
| a_foul | 0.050 | 0.003 | 154 |
| h_3pts made | 0.050 | 0.004 | 174 |
| h_foul | 0.049 | 0.003 | 171 |
| h_rebound offensive | 0.044 | 0.005 | 289 |
| h_turnover | 0.044 | 0.006 | 345 |
| a_rebound offensive | 0.043 | 0.003 | 257 |
| a_3pts missed | 0.042 | 0.005 | 349 |

Considering the class evaluation with Brier scores, we can confirm that our first-order model delivers sensible predictions. With values closer to 0, the metric indicates well-estimated predictions.

4.2. *Comparison of Brier Score Distributions: First- vs Second-Order Markov Models.* To evaluate the predictive accuracy of the Markov models, Brier scores were computed for individual transitions and visualized using stacked histograms. Figure 1 shows the Brier score distribution for the first-order Markov model (1oM), while Figure 2 illustrates the same for the second-order model (2oM).

*First-Order Markov Model (1oM).* The first-order model relies on single prior events for prediction. As a result, the number of unique transitions is limited, and the model produces a high number of evaluations. The histogram shows an extremely high density of transitions with Brier scores between 0 and 0.04. Most event types, including *rebound offensive*, *turnover*, and *made shot*, contribute to this bulk.

*Second-Order Markov Model (2oM).* In contrast, the second-order model considers pairs of prior events (Prev1 | Prev2), which expands the transition space significantly. This leads to a much lower total number of transitions with enough observations for reliable evaluation. The resulting histogram is more dispersed, with Brier scores ranging from 0 up to approximately 0.20. Although the counts are lower, the second-order model shows more event type diversity, including distinctions such as *made FT* and *missed FT*, indicating a more detailed description of game sequences.

*Conclusion..* The first-order model achieves low Brier scores across a large number of transitions but may oversimplify event dependencies. The second-order model, though evaluated on fewer data points, demonstrates stronger context sensitivity and modeling depth. The trade-off between coverage and context should be considered when selecting the appropriate model for simulation or predictive tasks.

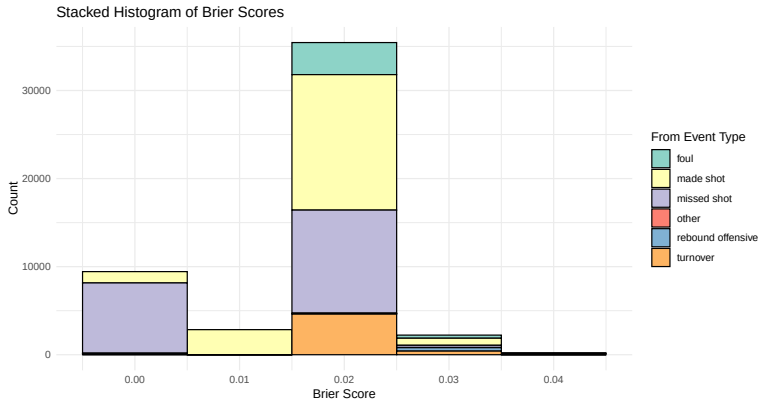


FIG 1. *Stacked histogram of Brier scores by event type – First-order Markov model.*

To see how predictive ability differs among event categories, we utilize a stacked histogram of Brier Scores organized by the previous event. Each bar reflects the distribution of Brier Scores for transitions caused by a single event type, allowing us to compare prediction quality across contexts. The x-axis shows the Score values ranging from 0 (perfect prediction) to 1 (completely incorrect), allowing us to

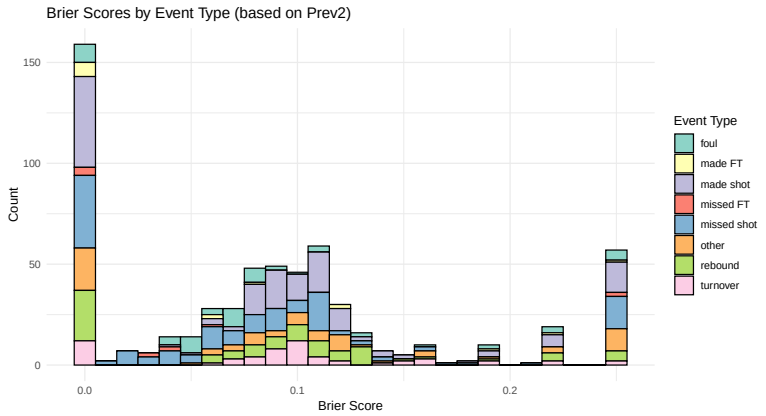


FIG 2. *Stacked histogram of Brier scores by event type – Second-order Markov model.*

assess the accuracy of individual transition predictions based on their error . This picture shows which sorts of occurrences produce more accurate forecasts (lower Brier Scores) and which are associated with greater uncertainty.

Each color represents a category of the `current state` event, including made and missed shots, fouls, turnovers, and other transition types. The two obvious outliers explain the results after a shot, one could either be scored, or missed followed by a rebound. The distribution of the events across the $x-axis$ implies the variety of ways in which a basketball possession could unfold.

The same idea is being applied to the second-order Markov model. The higher amount of combinations logically leads to more columns, describing a similar-to-normal distribution shape. with the exception of the values at the origin of the x-axis. There one can find situations that could happen after specific, really specific scenarios. For example, a player can collect a defensive rebound only when the opponent has thrown the ball and has missed. Also, rule-specific situations could be found there, the single free throw rule is a clear example. A player is awarded a single free throw shot if he has been fouled during his shot, and also he scored. (`a_2pts made` $\rightarrow$ `h_foul`) $\Rightarrow$ `a_free throw 1 of 1 made`

4.3. *Scoring Probability* . To evaluate how well the simulation model captures realistic scoring behavior, we analyze the probability of scoring as a function of the current point difference. Figure 9 shows the scoring probability on the y-axis and the point difference n the x-axis, discretized into groups of 5-point intervals.

Three curves are displayed: the red line represents empirical probabilities, while the green dashed line corresponds to the results of 500 simulated games generated using the ***EvtRelFreq*** and ***DurSample*** models of the first-order Markov model. Another 500 games have been simulated using the second-order Markov framework, the results are reprsented by the blue line.

The behaviour of the simulated probabilites seems comparable to real life. The most intense part of a basketball game is when the score is equal, so the game is up

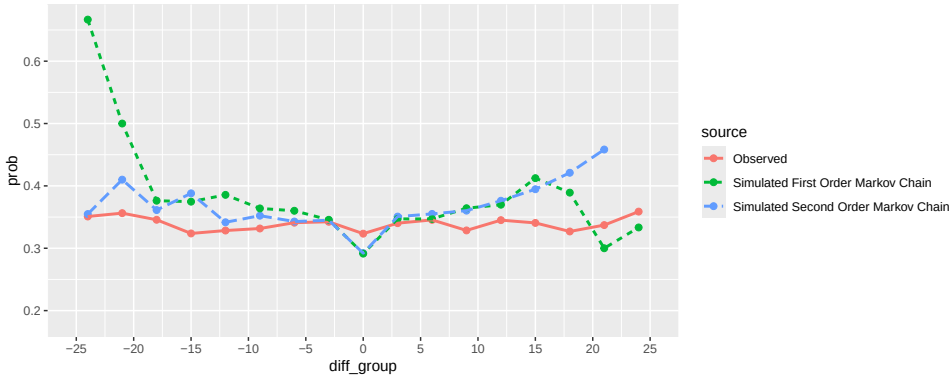


FIG 3. *Scoring Probability by Points Difference*

for grabs for both teams, that results in careful play with strong deffence. Logicaly, that leads to low scoring probabilities, there the simulation takes its minimum. All of the lines describe similar behaviour, all of them having symmetry around the 0.

To assess the predictive performance of the proposed probabilistic models, a residual analysis was conducted comparing empirical scoring probabilities with model-based estimates. Specifically, scoring probabilities were aggregated by score difference groups, and differences were visualized using residual plots.

Three pairwise comparisons were made:

1. **Observed vs. First-Order Markov Model**: This comparison evaluates the fit of the first-order model to the real Play-by-Play data.
2. **Observed vs. Second-Order Markov Model**: This comparison reflects the improvement (or lack thereof) when the Markov model accounts for two prior events.
3. **First-Order vs. Second-Order Markov Model**: This direct model-to-model comparison reveals where the two approaches diverge in terms of predicted scoring probabilities, potentially highlighting systematic patterns that only the higher-order model captures.

In each case, the residual is computed as the difference between the scoring probability estimated by one model and that from the other (or the observed data). Formally, for a given score difference group $d$, the residual is defined as:

$$\text{Residual}_d = \hat{p}_{1,d} - \hat{p}_{2,d}$$

where $\hat{p}_{1,d}$ and $\hat{p}_{2,d}$ are the estimated scoring probabilities for group $d$ under two different sources (e.g., observed vs. model, or model vs. model).

Residuals are then plotted across point difference groups to visualize systematic under- or overestimation in different game contexts. Both plots against the observed data demonstrate small residuals, indicating appropriate fit. The second order Markov suggests a slightly better fit, due to its slightly shorter range of

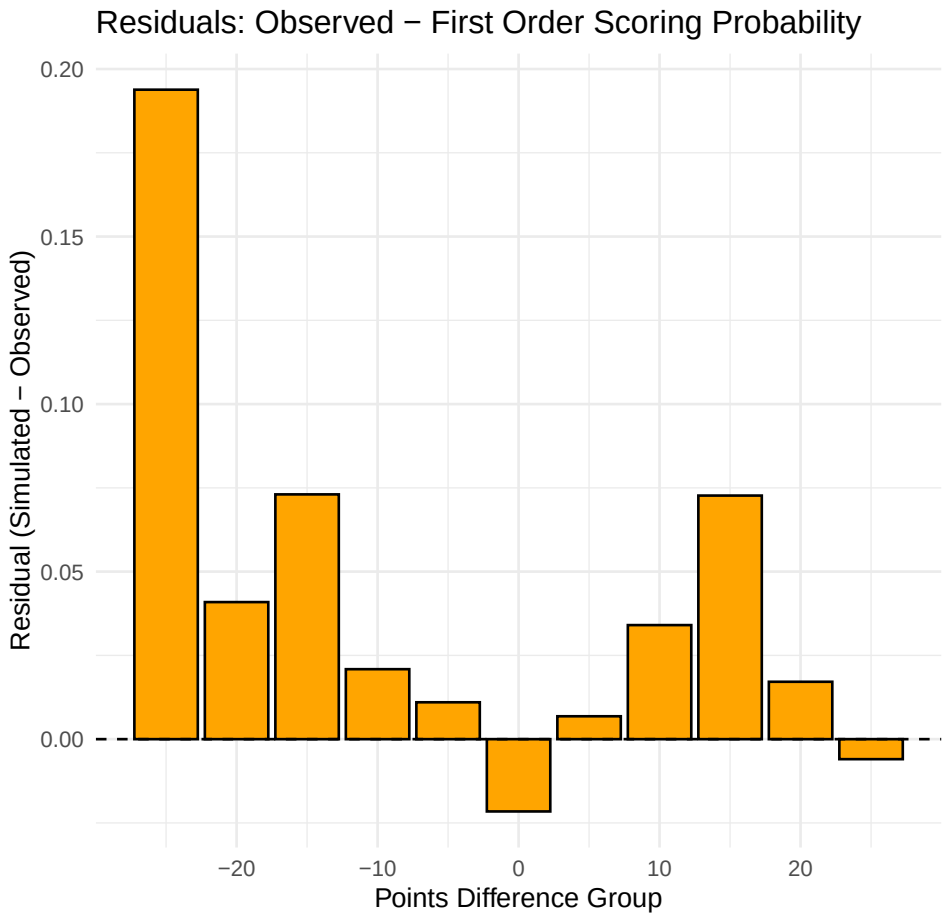


FIG 4. *Observed vs. First Order*

Residual difference. The second-order Markov chain, due to its higher complexity, manages to simulate realistic scoring probabilities even at the extremas. Keeping the residual difference below 0.04. Such a direct comparison between the two column diagrams could confirm our initial thought that a more complex, nuanced second-order Markov model may lead to overall better simulations. Even though the second-order estimates better the scoring probabilities, the first-order chain also brings solid results. With a residual difference of **maximum** .075 at the extrema, the one-transition model delivers a solid pipeline for realistic simulation of basketball games.

The Simulated against Simulated residual plot, points out really small residual differences, raising up on the extremas.

Both models show reliable fit.

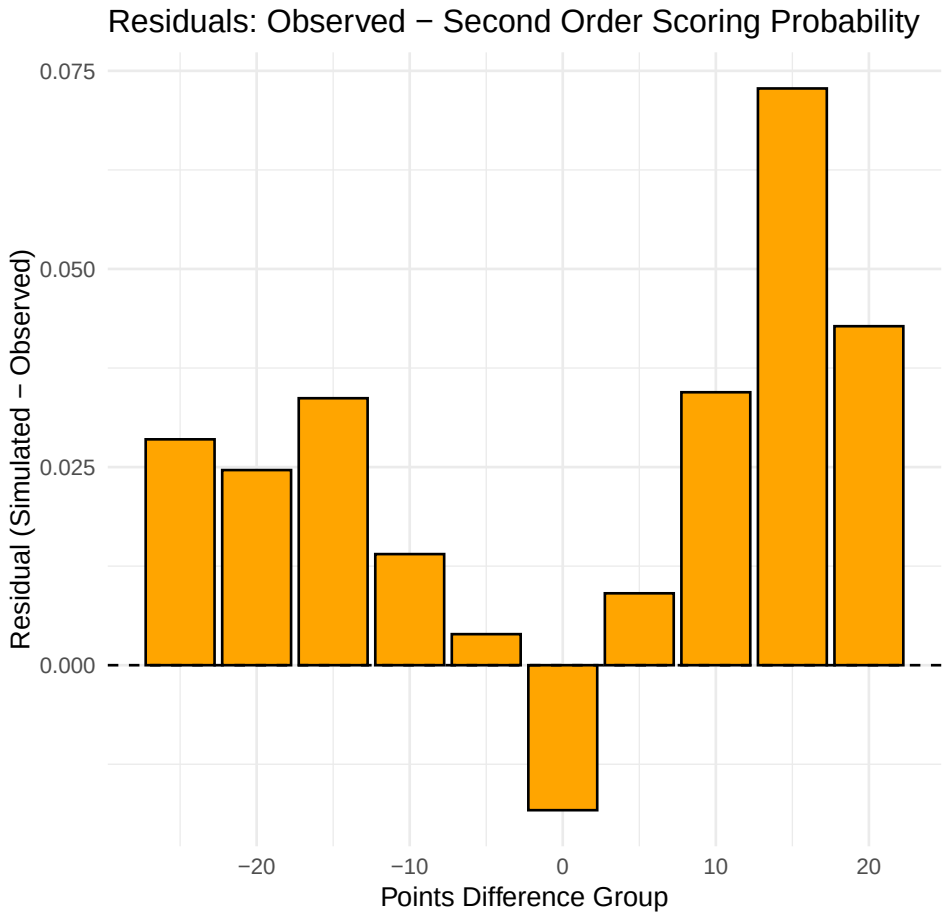


FIG 5. *Observed vs. Second Order*

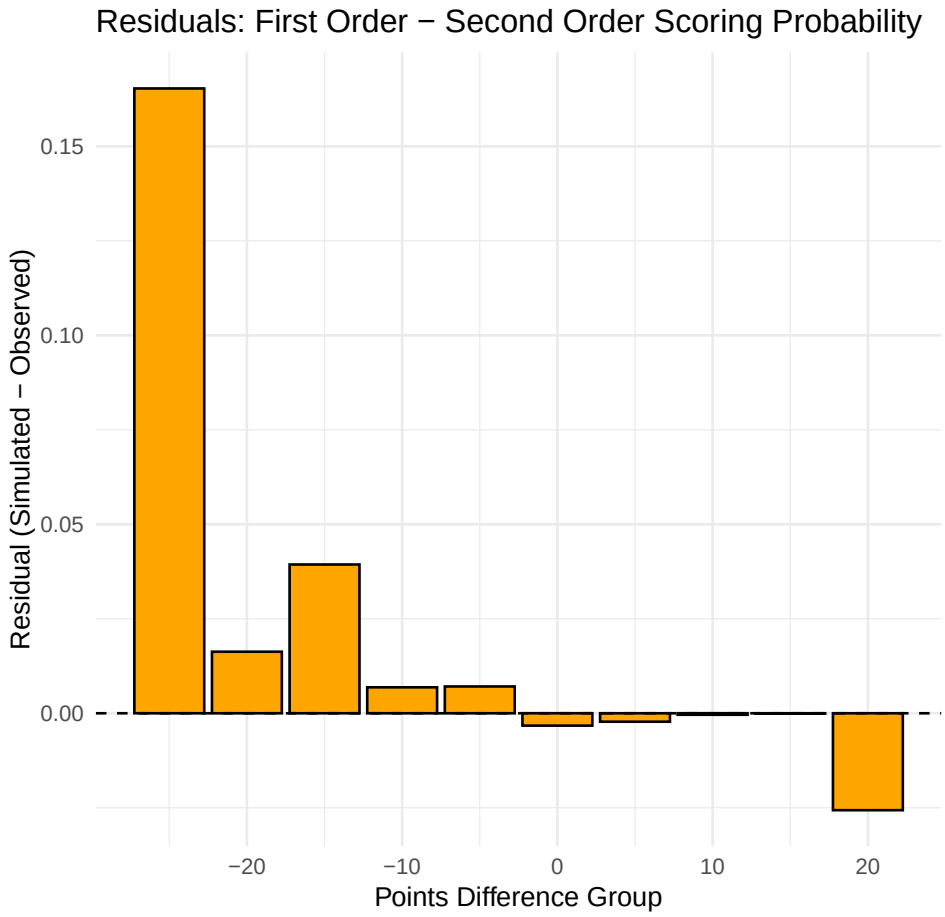


FIG 6. *First Order vs Second Order*

4.4. *Clustering of the Transition Matrices.* The statistical analysis of basketball play-by-play data requires high-dimensional structures since each game is defined by a large range of event types and transition probabilities. Such data frequently contain significant correlations between characteristics, obscuring underlying structures and making direct interpretation challenging. After creating the team-specific transitional matrices and binding them together, we end up with two data frames, one for the first-order Markov chain and one for the second-order. The first order consists of 6740 entries, and the second order table of almost 7400. The analysis of both of them would benefit highly from an extensive dimensionality reduction. In this work we go through and have 2 transition matrices for every one of 15 teams. We have separated the games when the teams were away, from where they are at home, so we can get the most out of our dataset. To understand more about the underlying connections between the teams, we proceed with a dimensionality reduction, via the PCA algorithm, followed by a K-Prototypes clustering procedure. Every step in the preprocessing and the analysis is being repeated and adapted, also for the higher-order Markov process. We want to investigate a specific scenario. How the possession will unfold depends on the tactics of the team, the skills of the players, and the preparation of their opponents. To investigate this case, we subset our big matrix to one that, as initial states, takes **(h rebound deffensive)** for an home team POV and **(a rebound defensive)** analogous.

Dean Oliver turned an interesting page in basketball, development history with his work on the "Four Factors" -Oliver (2004).He created a linear model, trying to explain how to win of basketball.He discovered that over 70 Percent of the variance of winning, is covered by four key statistics. We want to investigate 2 of his four factors with the team-specific transition matrices.

To investigate the separation of play-by-play events after clustering, we combined the results of the `k-prototypes` algorithm with a principal component analysis (PCA).

1. The categorical variables (`From`, `To` and `venue`) were transformed into dummy variables using the `model.matrix()` function. The numerical variables (`count` and `Probability`) were appended to this matrix.
2. PCA was applied to the combined matrix in order to reduce the high-dimensional representation into two principal components (PC1 and PC2). These components capture the maximum possible variance while enabling two-dimensional visualization.
3. The cluster membership obtained from `k-prototypes` was added to the dataset as a categorical variable. In addition, a variable `tier` was created, which groups teams into performance tiers based on their number of wins.

4.5. *Offensive Rebound.* The offensive rebound is special because it allows the attacking team to take another possession, without having the risk of being scored on. Our goal here is to investigate the behavior of the B2L teams, specifically how successful are they with securing the offensive rebound, after a missed shot.

As covariables in the clustering and visualization procedure we included the

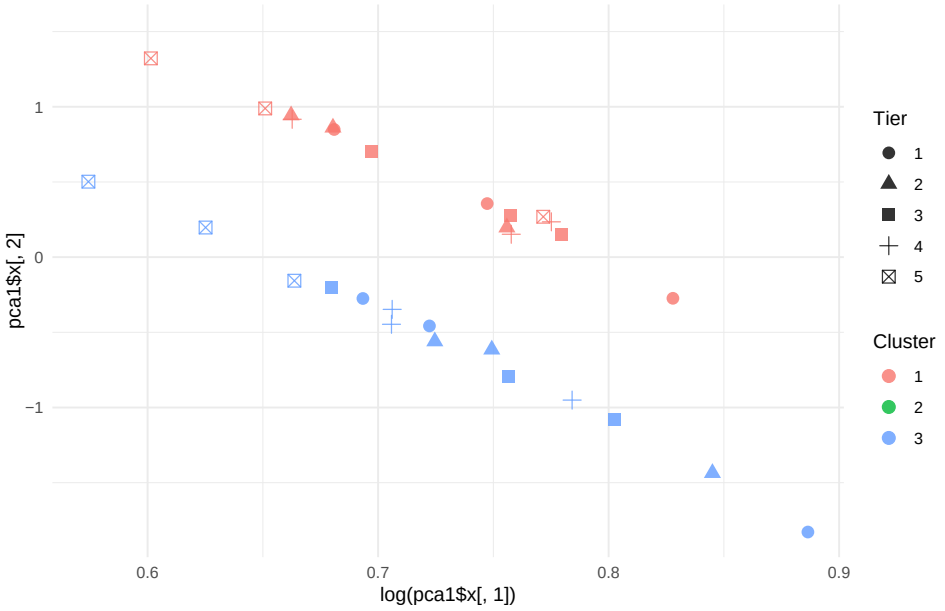


FIG 7. *First Order Offensive Rebound:* ***log(PCA1)*** *vs PCA2*

following:

(a) **Initial state (shot missed):** Indicates whether the possession started with a missed shot. This variable further distinguishes whether the attempt was taken from inside the three-point line or from behind it. In addition, the home/away context of the team is recorded.

(b) **Next state (offensive rebound):** Captures whether the missed shot was followed by an offensive rebound. This covariable also specifies whether the rebound was obtained by the home or the away team.

(c) **Transition probability:** The estimated probability of observing this exact transition (from missed shot to offensive rebound) for the given team. This value reflects the relative likelihood of the transition within the team's overall play-by-play dynamics.

(d) **Transition count:** The number of times the specified transition was observed in the dataset for the given team. This absolute frequency complements the probability by quantifying the empirical occurrence.

From the scatter plot of the first two principal components, we can see two well-separated clusters. Two linear groups parallel to each other. As points we have chosen the "Tier." In every tier we put 3 teams, dependent on their ranking from last season. In the first tier we can find the 3 teams with the most amount of wins; similarly, in the last one we find the worst teams for the 2024-2025 season. When we closely investigate the characters, it's easy

to see that if a team lands in the top left corner, it's more likely for it to be on the bottom of the table. When we follow the icons with the direction to the diagonal, we see gradient improvement. On the downright corner we see a bunch of high-tier teams. This may suggest that the team's success is well explained by their probability of getting an offensive rebound after missing a shot. To understand the clustering procedure, one can take a look at the following two tables. After understanding them, we can argue that the cluster algorithm grouped the probabilities with respect to the venue the game was played at. This is a well-known idea in sports, called "home court advantage." There is a belief that teams play differently away and at home. The second cluster is lost after logarithmizing the x-axis, due to the whole group has been full with outliers.

TABLE 4
*Shot vs Cluster*

| | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| a 2 missed | 0 | 0 | 15 |
| a 3 missed | 0 | 15 | 0 |
| h 2 missed | 15 | 0 | 0 |
| h 3 missed | 15 | 0 | 0 |

TABLE 5
*Rebound vs Cluster*

| | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| a rebound offensive | 0 | 0 | 30 |
| h rebound offensive | 15 | 15 | 0 |

4.5.1. *Second-Order Markov Test.* In order to focus on a specific sequence of play-by-play events, we constructed a subset of the full transition matrix that represents a second-order Markov process. The logic of the subsetting is as follows:

- We first select all sequences where the event two steps before the current state (`Prev1`) corresponds to a missed field goal attempt. This can be either a missed two-point shot (`2 missed`) or a missed three-point shot (`3 missed`).
- The immediately preceding event (`Prev2`) is required to be an offensive rebound by the same team. This ensures that we are analyzing possessions where a missed shot is followed by the same team regaining control of the ball.
- The variable `venue` is included to distinguish between home (`h`) and away (`a`) sequences, so that both contexts are considered separately.
- For the next event (`Next`), we retain only cases where the follow-up after the offensive rebound is another field goal attempt by the same team, whether made or missed. Specifically, we keep `2 made`, `3 made`, `2 missed`, and `3 missed`.

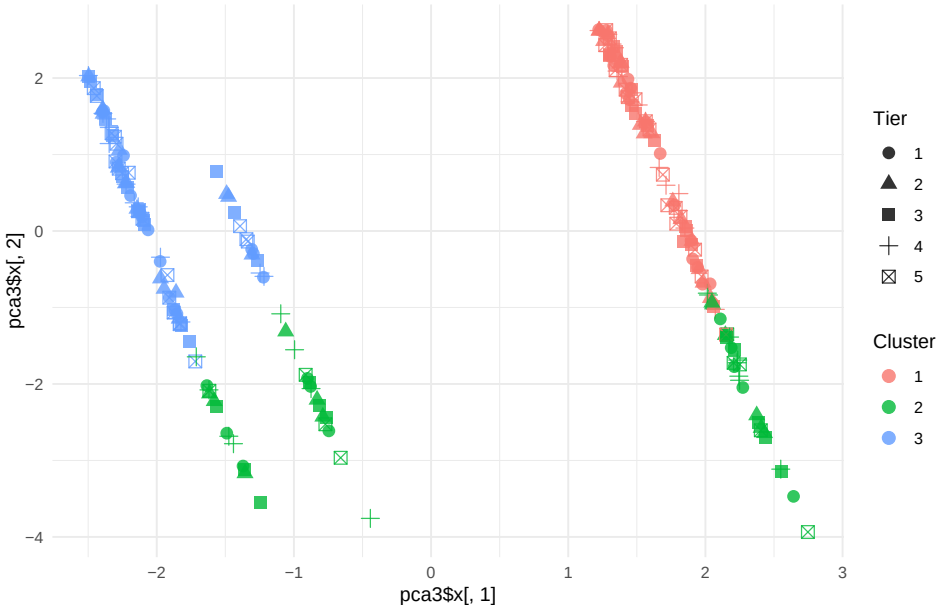


FIG 8. *Second Order Offensive Rebound: PCA1 vs PCA2*

In this case we can see all 3 clusters. But the big variety of combinations, which the second-order Markov provides, is making the process of interpreting harder compared to the first-order Markov process. We can still watch the same diagonal downward shape of the point clouds. Interesting to see is that here the parallel lines are not detected by the algorithm as clusters. For low values on the PC2 axis, a data point will always be allocated to the second cluster. After the point exceeds some threshold on the vertical axis, then it could be further clustered by another centroid.

4.6. *Turnovers Leading to Immediate Baskets.* An important aspect of basketball analytics is the impact of turnovers on scoring. A turnover represents a lost possession, and in many cases it allows the opposing team to quickly convert the mistake into easy points, often in the form of a fast break. From a modeling perspective, these situations are of particular interest, since they represent high-value transitions.

To study this phenomenon, we extracted from the transition matrix all sequences where a turnover was immediately followed by a made two-point basket by the opposing team.

Once again we have logarithmized due to the number of outliers.Almost perfect separation could be seen in 9.In the first quadrant are allocated the higher-tier teams, and diagonally from this clout , the weaker teams are to be found. Similar to before, the results confirm the prejudice that the better teams are more efficient on the fast break after a mistake. Here the clustering

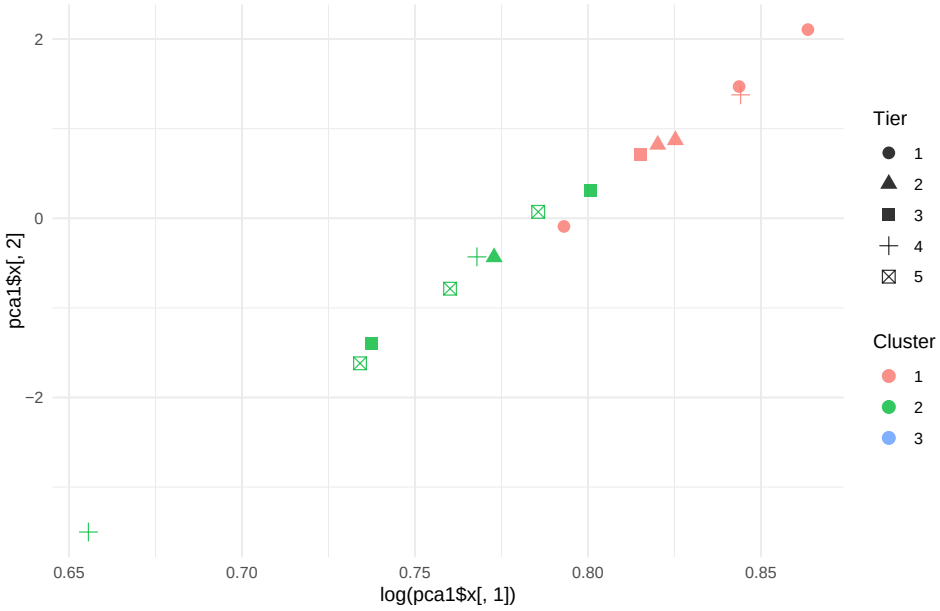


FIG 9. *First Order Turnover to Layup:* ***log(PCA1)*** *vs PCA2*

may be derived mostly from the level of the teams. Higher success probabilities lead to more successful teams.

4.6.1. *Conclusion.* The analysis based on the first-order Markov chain combined with PCA and clustering turned out to be both accurate and highly interpretable. The principal components captured the main dynamics of play-by-play transitions, and the resulting clusters aligned well with intuitive basketball concepts. The first-order approach provides a realistic estimation of the ranking of a team.
By contrast, the second-order Markov extension did not provide additional explanatory power for our purposes. Although it formally captures more context by considering the two most recent events, the resulting state space becomes much larger and sparser. This complexity makes the PCA loadings and clustering structure harder to interpret . Consequently, the second-order approach appears less useful for practical analysis compared to the clear and robust results of the first-order model.

**5. Discussion .** The results demonstrate that even simple non-parametric models such as ***EvtRelFreq*** and ***DurSample*** can reproduce an interpretable version of basketball gameplay, including realistic scoring patterns and state transitions. However, the analysis also reveals limitations in predictive sharpness and calibration.
Residual analysis and scoring probability comparisons reveal that while the

first-order model achieves reliable baseline performance, the second-order model offers more sensitive predictions. The histogram and residual plots show that the second-order model more closely matches observed scoring patterns, especially in balanced or decisive game situations. The final probability-to-score plot confirms that both models approximate the general scoring trend across different score margins, yet the second-order model does so with greater fidelity in most regions. This comes at the cost of increased model complexity and sparser data coverage.

From the perspective of dimensionality reduction and clustering, the first-order Markov chain combined with PCA turned out to be both accurate and highly interpretable. The principal components captured the main dynamics of play-by-play transitions, and the resulting clusters aligned well with intuitive basketball concepts. In this sense, the first-order approach provides a realistic approximation of game flow while remaining simple enough to interpret and to connect with tactical insights.

By contrast, the second-order Markov extension, although theoretically deeper and capable of capturing short-term memory, did not yield additional explanatory value in the PCA and clustering context. The expansion of the state space led to sparsity and interpretability issues. In summary, the first-order model strikes a practical balance between realism and interpretability, whereas the second-order model highlights the potential of higher- order stochastic modeling but suffers from complexity and sparse data. Together, these findings suggest that first-order models are well suited for exploratory and interpretive analysis, while second-order models may be more appropriate for specialized predictive tasks and tactical simulations.

## References.


BRIER, G. W. (1950). VERIFICATION OF FORECASTS EXPRESSED IN TERMS OF PROBABILITY. *Monthly Weather Review* **78** 1 - 3.

Huang, J. Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery* **2** 283-304.

Jolliffe, I. (2011). Principal component analysis. In *International encyclopedia of statistical science* 1094–1096. Springer.

Oliver, D. (2004). *Basketball on paper : rules and tools for performance analysis*, 1st ed. ed. Brassey's, Inc.,, Washington, D.C. :.

Rodda, B. E. (1969). *An analytic system for the statistical analysis of Markov chains*. Tulane University.

Vračar, P., Štrumbelj, E. and Kononenko, I. (2016). Modeling basketball play-by-play data. *Expert Systems with Applications* **44** 58-66.