Clustering NBA Players Based on In-Game Statistics

Tyler Kim [*1]

Abstract

The advent of data collection and game tracking in modern professional sports offers potential for increasingly sophisticated analysis of gameplay. This paper focused on developing NBA player groupings that are more informative than the traditional enumeration of 5 positions. Using detailed statistics — including groupings by play-type (pick and roll, isolation) and more — applying the K-means clustering algorithm identifies functional roles on the basketball court.

1. Introduction

1.1. Background

There are traditionally five positions in basketball: point guards, shooting guards, small forwards, power forwards, and centers. Each position provides a unique framework for a player's role on the court and their contribution to the game. For instance, point guards typically manage ball handling and initiate offensive plays by passing to teammates, whereas power forwards often play near the basket, focusing on layups and rebounds. Much of this positioning and delegation of roles is attributable to physical qualities such as size and athleticism — taller players are typically forwards who guard the basket, get rebounds, and score layups; athletic players are typically explosive and coordinated guards, who handle the ball and create scoring opportunities.

However, modern basketball displays significant variation in play style and capability within each position. A notable example of this inter-positional variance is the "stretch four," a power forward with exceptional three-point shooting skills. These players "stretch" the defense by forcing them to cover more open court. This new archetype emerged due to the increased emphasis on three-point shooting in contemporary basketball. Understanding these evolving roles is crucial for team management and strategy formulation. Identifying player types more precisely can enhance team composition, improve player development strategies, and optimize game tactics to counter specific opponents.

This has prompted me to reconsider the traditional categorization of NBA players. This paper looks to use use statistics to identify player types more effectively than the traditional set of five positions allows. To do so, we use in-game data from `nba.com/stats` to cluster players based on similarity. Due to the large dimensionality of the dataset, we project the data onto a lower dimensional representation in order to visualize the data and its clusters.

2. Related Work

Player clustering in basketball analytics has evolved significantly to address the limitations of traditional "box score" statistics, which often fail to capture the complex skill sets of modern NBA players. Alagappan's seminal work at the 2012 MIT Sloan Sports Analytics Conference introduced the expansion of traditional positions into thirteen detailed player roles using topological data analysis, challenging the traditional five-position framework and enriching the understanding of player contributions (**?**). Building on this, Chang et al. utilized larger datasets and advanced statistical methods such as PCA and k-means clustering to refine player categorization, which helped visualize NBA players as a network clustered from their statistical performance (**?**).

Chen, Zhang, and Xu applied clustering techniques to categorize players into specific offensive roles using play-type data from the Chinese Basketball Association, finding significant correlations between these roles and team performance (**?**). This highlighted the value of detailed role analysis for strategic team management.

This study enhances the literature by integrating various statistics, including clutch stats, hustle plays, and advanced shooting metrics. This approach improves the granularity and accuracy of player clustering by using comprehensive datasets, providing new insights into player contributions that can influence team management and player evaluation strategies.

[*]Equal contribution [1]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA. Correspondence to: Tyler Kim <tylerkim@mit.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

3. Data

3.1. Description

Our data is collected from the https://www.nba.com/stats/players/traditional page. I collect player stats from every season since the 2015-16 season. In our analysis, I only use data from the 2023-24 season. Potential extensions to this project could look into cluster evolution over time, or how players progressed throughout their career.

Our data contains 628 features for 572 players during the 2023-24 season. All statistics are measured on a per-game basis. A comprehensive explanation of the statistics captured follows:

- **Traditional Statistics**: Common statistics like points, rebounds, assists, etc.
- **Tracking Statistics**: Tracks player performance based on in-game moves. This includes catch-and-shoot opportunities, drives to the basket, pull-up shots, and more.
- **Clutch Statistics**: Statistics measured during "clutch time", which is "defined as the final five minutes of the fourth quarter or overtime when the score is within five points".[1]
- **Play Type Statistics**: Tracks player performance based on common set plays. Some examples include pick-and-roll, isolation plays, and post-up plays.
- **Box Out Statistics**: Player statistics to measure how often players box, and rebounding outcomes when they box out — both on offense and defense.
- **Hustle Statistics**: Measures player's "hustle", including the number of charges they draw, loose balls they recover, and more.

A schema is provided below in Figure 1 for visualization purposes.

[1]https://www.nba.com/news/stats-breakdown-coming-through-in-the-clutch

Figure 1. Star Schema Diagram for NBA Data

3.2. Pre-processing

The most important pre-processing step I took was handling null values. In traditional statistics, every player has statistics present. However, in some of the more advanced stat tables, there were many missing values. These missing values were not random — the data for "Isolation" play types has a disclaimer that "Minimum of 10 min/game and 10 possessions per play type to qualify". I deemed that players who did not qualify to be tracked should be imputed with 0 since that is essentially the interpretation of their absence in the data. Intuitively, this makes sense — most NBA centers do not handle the ball often, and thus their true statistics in play types like "isolation" are essentially 0. This idea generalizes to all players who are missing stats in-game scenarios that they are unlikely to participate in.

Aside from null value handling, some of our methods require additional pre-processing, such as double centering in MDS, or standardizing in PCA. I address the particular pre-processing steps for each analysis in the Methods [4] section.

4. Methods

4.1. Multidimensional Scaling

Multidimensional Scaling (MDS) is a statistical technique for projecting high-dimensional data into low-dimensional space. MDS focuses on preserving the distance relationships among the original data points, meaning that items that are similar to each other in the high-dimensional space remain close in the reduced space, and those that are different are placed further apart.

Specifically, MDS starts by computing a matrix of distances D between each pair of points in the data set. This matrix represents the pairwise dissimilarities between data points.

For n items, D is an $n \times n$ symmetric matrix where the element d_{ij} is the distance between item i and item j.

The next step involves transforming the distance matrix D into a matrix B through double centering:

$$B = -\frac{1}{2} J D^2 J$$

where D^2 denotes the element-wise square of D, and J is the centering matrix defined as $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, with I being the identity matrix and $\mathbf{1}$ a vector of all ones. MDS then performs an eigenvalue decomposition on B to obtain eigenvalues λ_i and eigenvectors v_i.

The final configuration of points in a lower-dimensional space is obtained by selecting the top k eigenvalues and their corresponding eigenvectors. The coordinate matrix X is given by:

$$X = V_k \Lambda_k^{1/2}$$

where V_k is the matrix of the k largest eigenvectors and Λ_k is the diagonal matrix of the corresponding eigenvalues. The rows of X represent the coordinates of the original items in the new k-dimensional space.

See Figure 2 for our MDS results.

4.2. Principal Component Analysis

Principal Component Analysis (PCA) is an alternative statistical technique for dimensionality reduction. Specifically, PCA works by transforming the data into a new coordinate system defined by the directions which maximizes variance.

First, the data is standardized the data to have a mean of zero and a standard deviation of one, ensuring equal treatment of all variables.

$$\mathbf{Z} = \frac{\mathbf{X} - \mu}{\sigma}$$

Next, a covariance matrix is calculated from the standardized data to capture the variance and covariance between features.

$$\mathbf{C} = \frac{1}{n-1}\mathbf{Z}^T\mathbf{Z}$$

Then, the covariance matrix is decomposed into its eigenvectors and eigenvalues, which define the new axes and their importance, respectively.

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$$

Lastly, principal components are selected based on the magnitude of their eigenvalues, and the original data is projected onto these components to reduce dimensions.

$$\mathbf{P} = \mathbf{Z}\mathbf{V}_k$$

See Figure 4 for our PCA results.

4.3. K-Means Clustering

K-means clustering is an iterative algorithm that creates clusters in data given k, a constant representing the number of clusters desired.

Algorithm 1 K-means Clustering

```
k = number of clusters
Randomly initialize k centroids in data
while clusters are still updating do
    Reassign points to clusters to nearest centroid
    Re-compute the centroid based on the new cluster
    if No updates then
        return Cluster assignments
    end if
end while
```

I use this algorithm to construct our clustering assignments. This particular algorithm is optimal to use in our case, as it lets us control the overall number of clusters I group players into. Given the real-world implication of a player cluster representing players of a similar production level, we'd like to have ultimate control over the granularity of the clustering assignments. Additionally, an unsupervised learning algorithm since we are relying on the algorithm itself to build the clusters, and do not have labeled data. Figure 2 and 4 each display the result of k-means clustering for $k = 6$.

4.4. Multinomial Logistic Regression

Multinomial logistic regression (MLR) is an extension of logistic regression. In MLR, the problem formulation is slightly different — I first choose a baseline class to serve as reference to the other classes. Then, model the log-odds function as a linear combination of regression coefficients and features. For a given baseline class K, MLR computes $n-1$ regression estimates for each output class k, not including the baseline K:

$$ln(\frac{\mathbb{P}(y_i = k)}{\mathbb{P}(y_i = K)}) = \beta_k X_i$$

where y_i is the class of data point i. From this equation, we can see that coefficient estimates apply only to comparisons between the given class k and the baseline class K. Thus, for positive values of β_{ik}, an increase in feature i makes it more likely that X would be assigned to k than K, while a negative β_{ik} makes K more likely.

Parameters in MLR are usually estimated through Maximum A Posteriori (MAP) estimation, where we maximize the posterior distribution with respect to β. The posterior is given by $f(\beta|X)$. Rearranging using Bayes Rule, MAP

estimation can be expressed as

$$\hat{\beta}_{MAP} =_{\beta} f(X|\beta)g(\beta)$$

where $g(\beta)$ is the prior distribution over β. These coefficients are estimated by quasi-Newton algorithms like L-BFGS, or gradient methods.

I use the regression outputs to infer the relative predictive power of certain features in assigning points to clusters. While these regression estimates don't necessarily extend within clusters, the coefficients of a cluster can be compared to the coefficients of other clusters *compared to the same baseline* in order to interpret which features impact which clusters.

For instance, if cluster 1 has a positive coefficient for feature i that is much larger than clusters 2, 3, and 4, then I can say that feature i is more predictive of cluster 1 than the others since they were compared to the same baseline. Additionally, we can also uncover information about the baseline cluster by re-running the regression with a different baseline cluster and comparing results. Tables 1 and 2 show our regression results, for a visual example.

5. Results

Figure 2 shows the results of running MDS to obtain a 3-dimensional representation of the data and applying K-Means clustering to it ($k = 6$). Figure 3 is an MDS stress plot, which visualizes the relationship between the dimensionality of the data and the stress value. The stress value decreases significantly at 3 dimensions, suggesting that 3 dimensions capture a substantial amount of the data's structure while minimizing dissimilarities between distances between points in the lower dimensional space and the original high-dimensional data. In other words, the stress plot indicates that 3 dimensions effectively represent the data's structure with minimal loss of information.

Initially, I planned on exploring a wider range of dimensions for clustering after dimensionality reduction with MDS. However, applying clustering to higher-dimensional embeddings yielded significantly different assignments compared to 3 dimensions. This finding, along with the substantial decrease in stress observed in the MDS stress plot at 3 dimensions, suggests that using more dimensions might not be necessary for our data.

Figure 4 shows the results of running PCA to obtain a 3-dimensional representation of the data as well as applying K-Means clustering ($k = 6$). Figure 5 is a PCA Scree plot, which visualizes variance explained by each principal component in PCA.

We can see that little of the variance in the overall data is explained by the first 3 principal components — 52.7%, to be exact. Therefore, I use MDS instead, since it is a distance-based metric.

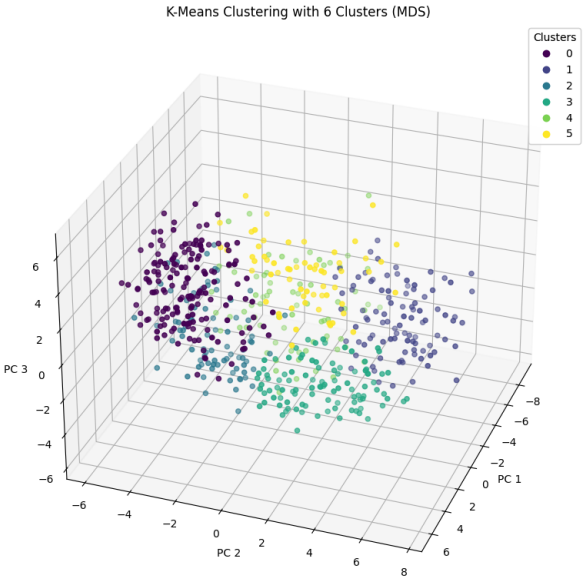

Figure 2. Cluster assignments in reduced dimensions (MDS)

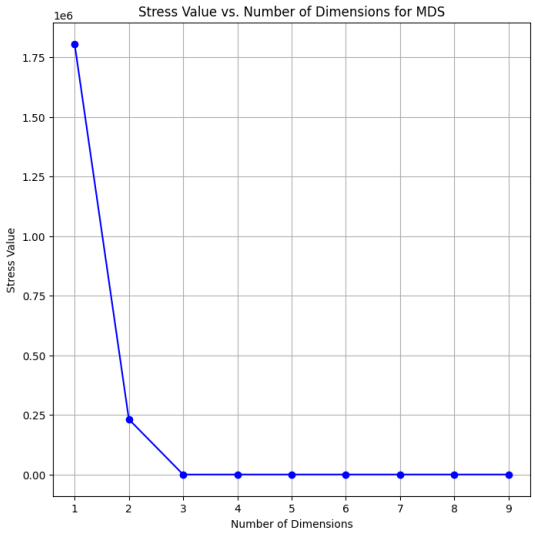

Figure 3. MDS Stress plot

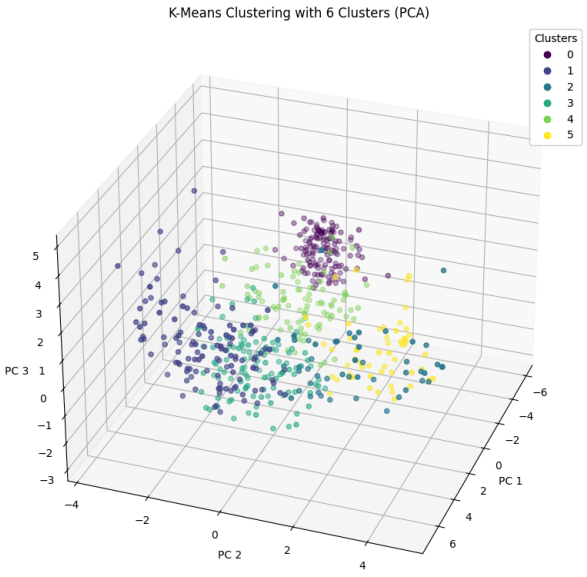

Figure 4. Cluster assignments in reduced dimensions (PCA)

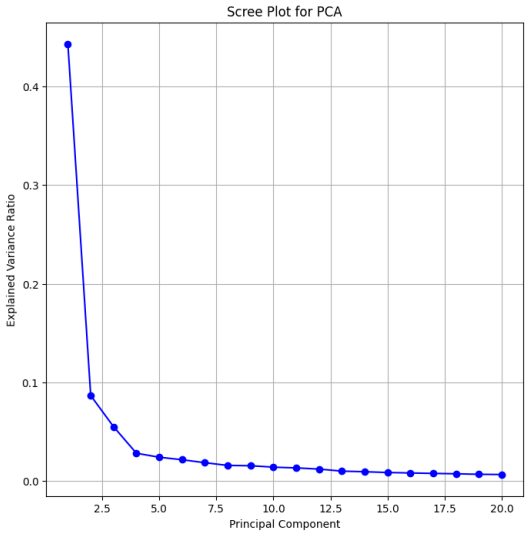

Figure 5. PCA Scree plot

5.1. Interpretation

I use the regression tables in the appendix to interpret the meaning of our clusters. Tables 1 and 2 in the appendix represent the MLR regression using Cluster 0 and Cluster 5 as the baseline classes, respectively. Standard errors for coefficient estimates are given in parentheses. P-values are indicated by "*" icons, with a detailed guide at the bottom of the table. I compute adjusted P-values using Benjamini-Hochberg adjustment, in order to control the Type 1 error rate due to the quantity of features I regress on.

The interpretation of coefficients is that a one-unit increase in feature value would correspond to an increase (or decrease, if the coefficient is negative) of β_i in the log odds for predicting cluster i over baseline.

I give commentary on the various clusters found by our model and their relevant features, according to the MLR.

- **Clusters 0 and 3**: Both of these clusters contain NBA players who receive little play time and do not record impactful stats — the canonical "bench warmer". One interesting finding to note is the coefficient of .884 for the "Minutes" feature of cluster 3 relative to cluster 0, which is statistically significant at the .001 level.[2] In addition to playing more minutes, cluster 3 can be characterized as scoring more points while recording less turnovers than cluster 0 (significant at the .05 level).

  In addition to bench players, cluster 3 contains high-production players who did not record many games due to injury (Ja Morant, Zach LaVine, Marcus Smart, Robert Williams III). No such players appear in cluster 0, indicating two "levels" of bench players.

- **Cluster 1**: Across baselines, cluster 1 is associated with higher three point percentage and free throw percentage across all clusters (statistically significant at the .01 and .05 levels). This indicates that cluster 1 are good shooters. They also have the lowest relative coefficient[3] on turnovers, indicating that could be a guard-type player expected to make smart passes. The top 3 leaders in 3-point percentage (Grayson Allen, Luke Kennard, Mike Conley) are present in this cluster, but positions range from point guard to power forward (Taurean Prince).

- **Cluster 2**: Characterized by the highest relative coefficients on minutes and points (significant to the .001 and .01 levels respectively), this cluster contains the highest offensively producing players of the NBA. MVP candidate guards Luka Doncic, Shai Gilgeous-Alexander are present, alongside high producing forward-type players like Jaren Jackson Jr, Julius Randle, and LeBron James.

- **Cluster 4**: This cluster has positive coefficients on games played and minutes (both staistically significant at the .001 level), a large positive coefficient on rebounds (.01 level), and is the only cluster with a positive coefficient on field goal percentage (.01 level).

  Cluster 4 are high performing forward-type players. They play a lot because are consistent scorers from the

[2]Unless otherwise noted, coefficient values will refer to the coefficient in Table 1 with baseline cluster 0.

[3]by which I mean the coefficient compared to the coefficients for other clusters on the same feature relative to the same baseline

field who get rebounds. The high field goal percentage but low three point percentage indicates 2-point shots closer to the basket, which typically have higher success percentages. While many of these prototypical NBA forwards are in this cluster like Giannis Antetokounmpo and Bam Adebayo, there are also many guard-like players who are known for scoring around the rim and getting rebounds: Josh Hart, Russell Westbrook, and Josh Giddey. Giddey and Hart lead all shooting guards in rebounds for the 2024 season.

- **Cluster 5**: This cluster has a positive coefficient on games played (.001 level) but the coefficient on minutes relative to cluster 0 is not statistically different from zero. Their coefficient on turnovers (.001 level) is almost as low as the guards, but they have an even higher coefficient on rebounds (.01 level) than cluster 4.

  These players are backup or low-usage forward-type players. They play in many games, but their minutes are similar to bench players, and get many rebounds. They are not scoring factors, but they also do not turn the ball over much — indicating that they do not get many opportunities with the ball (and as such, less opportunities for a turnover). Characteristic players include Al Horford, Mo Bamba, and Tristan Thompson.

6. Potential Applications

- **Player Acquisition**: The construction of a basketball team is fluid — trades happening during the season as well as free agency in the off-season means that rosters are not fixed for most of the year. These clusters offer a framework to identify players who address a roster need. For instance, a coach who notices that games are often lost because the 2nd team is out-rebounded might target a player in Cluster 5 to provide back-up forward support.

- **Line-up Setting**: The clusters could aid coaches in setting game lineups. The clusters can be used in 2 ways in this scenario — defensively and offensively. For instance, if another team is substituting in players from cluster 1, a coach would want to sub-in his best perimeter defenders and draw up plays to designed to contest 3-point shots.

7. Conclusion

While our proposed clusters rely on traditional player positions for their interpretability, our clusters do identify out-of-position players who exhibit play styles very different from their assigned 1-5 position. From our analysis, players cluster into groups that can be best described as 3-point scoring guards (cluster 1), offensive gamechangers (cluster 2), tier 1/2 big men (cluster 4/5), and bench players (clusters 0 and 3). Our clustering assignments do not appear to be a suitable replacement for traditional positioning, but they do represent a grouping of players based on in-game production as opposed to physical factors, shedding light on some of the functional roles that players on the court are expected to take.

8. Appendix

Table 1. Multinomial Logistic Regression Results: Baseline Cluster = 0 (Standard errors in parentheses)

Feature	Cluster Assignment				
	1	2	3	4	5
Intercept	-33.691^{***}	-36.921^{***}	-5.051	-33.304^{***}	-35.905^{***}
	(0.324)	(0.248)	(6.338)	(0.272)	(0.776)
Defensive Rating	-0.026	-0.039	-0.083	-0.159^{*}	0.061
	(0.064)	(0.077)	(0.060)	(0.074)	(0.053)
Age	-0.403^{***}	-0.515^{***}	-0.326^{**}	-0.308^{*}	-0.086
	(0.122)	(0.141)	(0.108)	(0.141)	(0.100)
Games Played	0.452^{***}	0.380^{***}	0.001	0.431^{***}	0.410^{***}
	(0.062)	(0.064)	(0.037)	(0.065)	(0.058)
Minutes	1.043^{***}	1.358^{***}	0.884^{***}	0.989^{***}	0.272
	(0.259)	(0.282)	(0.242)	(0.273)	(0.244)
Points	1.325^{**}	1.485^{**}	1.017^{*}	1.234^{*}	0.725
	(0.499)	(0.500)	(0.456)	(0.497)	(0.503)
Field Goal %	-0.067	-0.162	0.017	0.257^{**}	0.096
	(0.105)	(0.136)	(0.052)	(0.094)	(0.073)
3-Point %	0.179^{**}	0.115	0.054	0.027	0.037
	(0.060)	(0.100)	(0.033)	(0.059)	(0.049)
Free Throw %	0.127^{*}	0.118	0.038	0.045	0.025
	(0.046)	(0.061)	(0.025)	(0.055)	(0.030)
Turnovers	-4.876^{***}	-3.204^{**}	-3.103^{**}	-2.929^{*}	-4.412^{***}
	(1.344)	(1.319)	(1.089)	(1.357)	(1.399)
Rebounds	-0.422	0.473	0.855	2.024^{**}	2.445^{**}
	(0.818)	(0.818)	(0.718)	(0.797)	(0.798)
Assists	0.737	1.377	1.187	0.111	-0.314
	(0.845)	(0.850)	(0.734)	(0.940)	(0.950)
Steals	2.906	3.400	5.146	0.559	2.227
	(2.941)	(3.057)	(2.614)	(3.222)	(2.848)
Blocks	2.024	0.419	1.457	3.000	5.079
	(3.253)	(3.296)	(2.431)	(3.193)	(3.136)
$+/-$	-0.100	-0.534^{*}	-0.441^{*}	-0.593^{*}	0.095
	(0.290)	(0.301)	(0.251)	(0.312)	(0.287)

Note: $^{***}p < 0.001$,** $p < 0.01$,* $p < 0.05$

Table 2. Multinomial Logistic Regression: Baseline Cluster = 5 (Standard errors in parentheses)

Feature	Cluster Assignment				
	0	1	2	3	4
Intercept	42.330^{***}	-3.367^{***}	-16.866^{***}	35.204^{***}	-5.101^{***}
	(3.357)	(0.209)	(0.094)	(3.500)	(0.078)
Defensive Rating	-0.117^{*}	-0.033	0.050	-0.181^{**}	-0.159^{*}
	(0.059)	(0.051)	(0.069)	(0.056)	(0.064)
Age	0.079	-0.312^{***}	-0.412^{***}	-0.232^{**}	-0.207
	(0.104)	(0.087)	(0.111)	(0.094)	(0.113)
Games Played	-0.432^{***}	0.046	-0.027	-0.428^{***}	0.034
	(0.059)	(0.038)	(0.045)	(0.053)	(0.039)
Minutes	-0.267	0.763^{***}	1.080^{***}	0.597^{**}	0.714^{***}
	(0.257)	(0.185)	(0.211)	(0.199)	(0.197)
Points	-0.810	0.564^{*}	0.702^{**}	0.187	0.458
	(0.512)	(0.270)	(0.273)	(0.302)	(0.260)
Field Goal %	-0.088	-0.171	-0.257^{*}	-0.073	0.166^{*}
	(0.079)	(0.092)	(0.127)	(0.074)	(0.077)
3-Point %	-0.027	0.144^{***}	0.052	0.029	-0.012
	(0.052)	(0.038)	(0.092)	(0.051)	(0.039)
Free Throw %	-0.018	0.097^{*}	0.084	0.014	0.016
	(0.032)	(0.042)	(0.058)	(0.033)	(0.052)
Turnovers	4.586^{***}	-0.450	1.259	1.483	1.588
	(1.430)	(1.087)	(1.105)	(1.145)	(1.110)
Rebounds	-2.633^{**}	-2.876^{***}	-1.994^{***}	-1.581^{**}	-0.427
	(0.825)	(0.488)	(0.485)	(0.537)	(0.365)
Assists	0.265	1.095	1.706^{*}	1.510^{*}	0.409
	(0.961)	(0.676)	(0.708)	(0.707)	(0.764)
Steals	-2.075	0.626	1.230	3.265	-1.616
	(3.014)	(1.786)	(1.996)	(1.874)	(2.037)
Blocks	-4.842	-2.932	-4.397^{**}	-3.683	-2.091
	(3.251)	(1.660)	(1.786)	(2.684)	(1.295)
$+/-$	-0.175	-0.146	-0.522^{*}	-0.561^{*}	-0.644^{**}
	(0.302)	(0.218)	(0.238)	(0.242)	(0.246)

Note: $^{***}p < 0.001$,** $p < 0.01$,* $p < 0.05$