P-ISSN: 2394-1685
E-ISSN: 2394-1693
Impact Factor (ISRA): 5.38
IJPESH 2019; 6(4): 45-51

www.kheljournal.com
Received: 28-05-2019
Accepted: 30-06-2019

Martínez
Technical University of Cartagena, Business Management Department, Cartagena, Spain

Jose A
Technical University of Cartagena, Business Management Department, Cartagena, Spain

Correspondence
Martínez
Technical University of Cartagena, Business Management Department, Cartagena, Spain

A method to compute standard errors in per-minute performance metrics in basketball

Martínez and Jose A

Abstract
We provide a way to compute confidence intervals for per-minute based statistics of player performance in basketball, based on the Taylor-series method and the concept of propagation of errors. Using the Player Total Contribution index, we have illustrated the method for evaluating the performance of a NBA player. In addition, we provide a free to use Maxima code, to make easier its application for researchers and analysts. This study is the first in the sports analytics literature to provide a detailed method to compute confidence intervals for per-minute based performance variables.

Keywords: Basketball analytics, propagation of errors, uncertainty, productivity of players

1. Introduction
Sports analytics is a highly dynamic and growing research field [1, 2], and the contributions made to basketball have been pervasive in the last two decades [3, 4, 5, 6, 7]. In spite of several advanced metrics have been developed, box-score statistics continues to be considered as the first approximation to assess player performance, and the metrics based upon those box-score variables are still widely employed by managers, media and fans, because of its easiness [8].

Although box-score does not reflect per-minute stats at a glance, further analysis to compute the per-minute version of all of those variables is straightforward. In fact, the main basketball websites providing stats of professional basketball incorporates per-minute stats.

However, those stats summarizing the performance at the end of the season are provided without considering the standard errors of the estimates, which is always present when the games played by a player are lesser than the total team games. For example, in the NBA regular season, each team plays 82 games. If a player plays less than 82 games, all the variables representing performance have an associated uncertainty reflected by the standard errors of such estimates. If analysts do not take into account that uncertainty, comparison among players and rankings of top performers could be biased [9].

The aim of this research is to provide a way to compute standard errors of per-minute performance variables in basketball, based on the propagation error approach. In addition, we provide the code of a program written in Maxima to easily apply the method. All the methods are applied to the Player Total Contribution Index (PTC), which evaluates the productivity of players using box-score variables [10].

2. Materials and methods
We employed the Taylor-series approach to approximate the function $g = f(x_1, x_2 \dots x_k)$, which is composed of *k* variables of the box-score, in the proximity of the expected value of those variables. Using only the first term of the Taylor-series (1):

$$g = f(x_1, x_2 \dots x_k) = f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k) + \frac{\delta}{\delta x_1}[f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)](x_1 - \bar{x}_1) + \cdots + \frac{\delta}{\delta x_k}[f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)](x_k - \bar{x}_k) \quad (1)$$

As $\bar{g} = \bar{f}(x_1, x_2 \dots x_k) \approx f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)$, , then the variance $Var(w)$ of a new function *w* can be defined as [11] (2):

$$Var(w) = E[f(x_1, x_2 \dots x_k) - f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)]^2$$
$$= E\left[\frac{\delta}{\delta x_1}[f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)](x_1 - \bar{x}_1) + \dots + \frac{\delta}{\delta x_k}[f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)](x_k - \bar{x}_k)\right]^2 \quad (2)$$

Which can be expressed as (3):

$$Var(w) = \left[\sum_{i=1}^{k} \frac{\delta}{\delta x_i}[f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)]\frac{S_{x_i}^2}{n} + 2\sum_{i=1}^{k}\sum_{j=i+1}^{k} \frac{\delta}{\delta x_i}\frac{\delta}{\delta x_j}[f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)]\frac{S_{x_i x_j}}{n}\right]^2 \quad (3)$$

Being $S_{x_i}^2$ the sample (n) variance of the k variable and $S_{x_i x_j}$ the sample covariance between two variables. The standard error of w is (4):

$$s.e.(w) = \sqrt{\left[\sum_{i=1}^{k} \frac{\delta}{\delta x_i}[f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)]\frac{S_{x_i}^2}{n} + 2\sum_{i=1}^{k}\sum_{j=i+1}^{k} \frac{\delta}{\delta x_i}\frac{\delta}{\delta x_j}[f(\bar{x}_1, \bar{x}_1 \dots \bar{x}_k)]\frac{S_{x_i x_j}}{n}\right]^2}\sqrt{\frac{N-n}{N-1}} \quad (4)$$

Where $\sqrt{\frac{N-n}{N-1}}$ is the finite population correction [12], which corrects standard errors in finite populations of size N.

Finally, a confidence interval around the mean of the function $\overline{G}$ can be built, using the following expression (5):

$$\bar{g} - Z_{1-\frac{\alpha}{2}}\, s.e.(w) \leq \overline{G} \leq \bar{g} + Z_{1-\frac{\alpha}{2}} \quad (5)$$

Being $\overline{G}$ the true population value and $Z_{1-\frac{\alpha}{2}}$ the standardized Normal cut point for a $100(1-\alpha)$ confidence level.

2.1 The PTC index

Player Total Contribution (PTC) [9] is an index comprising 11 variables of the box-score (6):

PTC = 1 Points made +.91 Blocks made + 58 Defensive rebounds + .92 Offensive rebounds + .86 Steals + .48 Assists + .23 Fouls drawn - .91 Missed field goals - .57 Missed free throws -.86 Turnovers – .23 Fouls made (6)

The per-minute version of this index is simply (7):

PTC_MP = PTC/MP (7)

Therefore, *PTC_MP* is an index comprising k=12 variables.

We can then apply expression (4) and (5) to compute the standard error of the estimate of the mean of this global performance index, and its 95% confidence interval.

2.2 The Maxima code

We have created a program written using the computer algebra system Maxima, which runs all the prior expressions. Users have only to provide the following inputs:

1. A txt raw data file for each player with the logs of each game: 12 variables with a required order.
2. The player name.
3. The population size N (number of all possible games: 82 for the NBA regular season)
4. The confidence level (usually 95%).

The code is provided in the Appendix.

2.3 Raw data

As an illustrative example, we employed the 72 games played by Luka Doncic, the player of the Dallas Mavericks NBA team, in the 2018/19 regular season. As Doncic was out 10 games because of several injuries, we considered 72 games as the sample of all the possible 82 games that Doncic could have played. Data was obtained from www.nba.com.

3. Results & Discussion

After applying the code showed in the Appendix, the results were the following (Table 1):

Table 1: Luka Doncic PTC_MP for the 2018/19 regular season

PTC_MP	95%CI_low	95% CI high
0.5727	0.5549	0.5906

Therefore, we have obtained an estimate of the mean value of his global contribution, using a per-minute performance measure, and the associated uncertainty of such estimation represented by a confidence interval.

This way of providing basketball metrics is more correct than the usual methods of simply reporting the punctual estimated mean without considering uncertainty, as all the stats websites of basketball do. By knowing the standard error, we could also compute the relative error of the estimation as a fraction of the estimated mean value. In the case of Luka Doncic, the relative error was 3.11%.

The size of relative errors can be considered as a criterion to decide if a player is included in a ranking or if the estimated performance of a player is reliable. High relative errors (for example above 10%) could indicate that the estimate is not reliable and such player has a great uncertainty in his performance.

Our procedure can also be applied with other measures of global performance in basketball as Tendex, Game Score or Win Score [1,3,4]. Analysts will decide what kind of productivity index is more complete. If other indexes are chosen, the weights will change (and they also have to be modified in the program). We clearly advocate for PTC, because of it is theoretically grounded and empirically validated [9].

In addition, an extension of our procedure and the application of our Maxima program can be done to any other per-minute performance measure in other sports, just changing the formula for the specific index and the weights (if necessary).

4. Conclusions

We provide a way to compute confidence intervals for per-minute based statistics of player performance in basketball, based on the Taylor-series and the concept of propagation of errors. Using the Player Total Contribution index, we have illustrated the method for evaluating the performance of a NBA player. In addition, we provide a free to use Maxima code, to make easier its application for researchers and analysts.

Appendix 1: Maxima code (inputs are coloured in red):

```
/* Program developed by Jose A. Martinez (Technical University of Cartagena, Spain)
Users only have to provide the following input to the program:
1. A txt data file for each player with the logs of each game: 12 variables with a required order.
2. The player name.
3. The population size N (number of all possible games: 82 for the NBA regular season)
4. The confidence level (usually 95%). Do not include the "%" symbol, just the number.
*/
/* Read data from a .txt file.
Data has to be displayed in 12 columns:
PTS: points made (x1)
BLK: blocks made (x2)
DRB: deffensive rebounds (x3)
ORB: offensive rebounds (x4)
STL: steals (x5)
AS: assists (x6)
FD: fouls drawn (x7)
FGM: fields goal missed (x8)
FTM: free thrwo missed (x9)
TOV: turnovers (x10)
PF: personal fouls (x11)
MP: minutes played (y)
Columns have no header, only the box-score numbers */
data:read_matrix(file_search("Put The Source of The File"));
/* Put the player name*/
Player: "Put The Name Of The Player";
/* Load the Maxima packages*/
load(descriptive)$
load (distrib)$
/*Include the initial values*/
/* Total number of possible games(N=population)*/
N:82;
/* Set the desired confidence level (90;95;99...) and Z will be automatically computed*/
confidence_level: 95;
alpha: (100-confidence_level)/100, numer;
Z: quantile_normal((100-(100*alpha/2))/100,0,1), numer;
/* Weights of the PTC_m function*/
w1:1;
w2:0.91;
w3:0.58;
w4:0.92;
w5:0.86;
w6:0.48;
```

```
w7:0.23;
w8:-0.91;
w9:-0.57;
w10:-0.86;
w11:-0.23;
w12:1;
/* Equation of the PTC_m function*/
PTC_m: (w1*x1+w2*x2+w3*x3+w4*x4+w5*x5+w6*x6+w7*x7+w8*x8+w9*x9+w10*x10+w11*x11)/(w12*y);
/* Compute the 1x12 mean vector and the 12x12 covariance matrix (cm)*/
datatranspose:transpose(data);
meanvector:mean(data);
cm:cov1(data);
Varx1:cm [1, 1];
Varx2:cm [2, 2];
Varx3:cm [3, 3];
Varx4:cm [4, 4];
Varx5:cm [5, 5];
Varx6:cm [6, 6];
Varx7:cm [7, 7];
Varx8:cm [8, 8];
Varx9:cm [9, 9];
Varx10:cm [10, 10];
Varx11:cm [11, 11];
Vary:cm [12, 12];
covarx1x2:cm [1, 2];
covarx1x3:cm [1, 3];
covarx1x4:cm [1, 4];
covarx1x5:cm [1, 5];
covarx1x6:cm [1, 6];
covarx1x7:cm [1, 7];
covarx1x8:cm [1, 8];
covarx1x9:cm [1, 9];
covarx1x10:cm [1, 10];
covarx1x11:cm [1, 11];
covarx1y:cm [1, 12];
covarx2x3:cm [2, 3];
covarx2x4:cm [2, 4];
covarx2x5:cm [2, 5];
covarx2x6:cm [2, 6];
covarx2x7:cm [2, 7];
covarx2x8:cm [2, 8];
covarx2x9:cm [2, 9];
covarx2x10:cm [2, 10];
covarx2x11:cm [2, 11];
covarx2y:cm [2, 12];
covarx3x4:cm [3, 4];
covarx3x5:cm [3, 5];
covarx3x6:cm [3, 6];
covarx3x7:cm [3, 7];
covarx3x8:cm [3, 8];
covarx3x9:cm [3, 9];
covarx3x10:cm [3, 10];
covarx3x11:cm [3, 11];
covarx3y:cm [3, 12];
covarx4x5:cm [4, 5];
covarx4x6:cm [4, 6];
covarx4x7:cm [4, 7];
covarx4x8:cm [4, 8];
covarx4x9:cm [4, 9];
covarx4x10:cm [4, 10];
covarx4x11:cm [4, 11];
covarx4y:cm [4, 12];
covarx5x6:cm [5, 6];
covarx5x7:cm [5, 7];
covarx5x8:cm [5, 8];
covarx5x9:cm [5, 9];
covarx5x10:cm [5, 10];
covarx5x11:cm [5, 11];
covarx5y:cm [5, 12];
covarx6x7:cm [6, 7];
covarx6x8:cm [6, 8];
covarx6x9:cm [6, 9];
```

```
covarx6x10:cm [6, 10];
covarx6x11:cm [6, 11];
covarx6y:cm [6, 12];
covarx7x8:cm [7, 8];
covarx7x9:cm [7, 9];
covarx7x10:cm [7, 10];
covarx7x11:cm [7, 11];
covarx7y:cm [7, 12];
covarx8x9:cm [8, 9];
covarx8x10:cm [8, 10];
covarx8x11:cm [8, 11];
covarx8y:cm [8, 12];
covarx9x10:cm [9, 10];
covarx9x11:cm [9, 11];
covarx9y:cm [9, 12];
covarx10x11:cm [10, 11];
covarx10y:cm [10, 12];
covarx11y:cm [11, 12];
/* Sample size is automatically computed*/
n:length(datatranspose[1]);
/* The program initiates the calculus of the propagation of errors*/
varianzas1: (diff(PTC_m,x1,1))^2*Varx1/n;varianzas2:(diff(PTC_m,x2,1))^2*Varx2/n;varianzas3:(diff(PTC_m,x3,1))^2*
Varx3/n;varianzas4:(diff(PTC_m,x4,1))^2*Varx4/n;
varianzas5:(diff(PTC_m,x5,1))^2*Varx5/n;varianzas6:(diff(PTC_m,x6,1))^2*Varx6/n;varianzas7:(diff(PTC_m,x7,1))^2*
Varx7/n;varianzas8:(diff(PTC_m,x8,1))^2*Varx8/n;
varianzas9:(diff(PTC_m,x9,1))^2*Varx9/n;varianzas10:(diff(PTC_m,x10,1))^2*Varx10/n;varianzas11:(diff(PTC_m,x11,1))^
2*Varx11/n;varianzasy:(diff(PTC_m,y,1))^2*Vary/n;
covarx1_2:(diff(PTC_m,x1,1))*(diff(PTC_m,x2,1))*covarx1x2/n;
covarx1_3:(diff(PTC_m,x1,1))*(diff(PTC_m,x3,1))*covarx1x3/n;
covarx1_4:(diff(PTC_m,x1,1))*(diff(PTC_m,x4,1))*covarx1x4/n;
covarx1_5:(diff(PTC_m,x1,1))*(diff(PTC_m,x5,1))*covarx1x5/n;
covarx1_6:(diff(PTC_m,x1,1))*(diff(PTC_m,x6,1))*covarx1x6/n;
covarx1_7:(diff(PTC_m,x1,1))*(diff(PTC_m,x7,1))*covarx1x7/n;
covarx1_8:(diff(PTC_m,x1,1))*(diff(PTC_m,x8,1))*covarx1x8/n;
covarx1_9:(diff(PTC_m,x1,1))*(diff(PTC_m,x9,1))*covarx1x9/n;
covarx1_10:(diff(PTC_m,x1,1))*(diff(PTC_m,x10,1))*covarx1x10/n;
covarx1_11:(diff(PTC_m,x1,1))*(diff(PTC_m,x11,1))*covarx1x11/n;
covarx1_y:(diff(PTC_m,x1,1))*(diff(PTC_m,y,1))*covarx1y/n;
covarx2_3:(diff(PTC_m,x2,1))*(diff(PTC_m,x3,1))*covarx2x3/n;
covarx2_4:(diff(PTC_m,x2,1))*(diff(PTC_m,x4,1))*covarx2x4/n;
covarx2_5:(diff(PTC_m,x2,1))*(diff(PTC_m,x5,1))*covarx2x5/n;
covarx2_6:(diff(PTC_m,x2,1))*(diff(PTC_m,x6,1))*covarx2x6/n;
covarx2_6:(diff(PTC_m,x2,1))*(diff(PTC_m,x6,1))*covarx2x6/n;
covarx2_7:(diff(PTC_m,x2,1))*(diff(PTC_m,x7,1))*covarx2x7/n;
covarx2_8:(diff(PTC_m,x2,1))*(diff(PTC_m,x8,1))*covarx2x8/n;
covarx2_9:(diff(PTC_m,x2,1))*(diff(PTC_m,x9,1))*covarx2x9/n;
covarx2_10:(diff(PTC_m,x2,1))*(diff(PTC_m,x10,1))*covarx2x10/n;
covarx2_11:(diff(PTC_m,x2,1))*(diff(PTC_m,x11,1))*covarx2x11/n;
covarx2_y:(diff(PTC_m,x2,1))*(diff(PTC_m,y,1))*covarx2y/n;
covarx3_4:(diff(PTC_m,x3,1))*(diff(PTC_m,x4,1))*covarx3x4/n;
covarx3_5:(diff(PTC_m,x3,1))*(diff(PTC_m,x5,1))*covarx3x5/n;
covarx3_6:(diff(PTC_m,x3,1))*(diff(PTC_m,x6,1))*covarx3x6/n;
covarx3_7:(diff(PTC_m,x3,1))*(diff(PTC_m,x7,1))*covarx3x7/n;
covarx3_8:(diff(PTC_m,x3,1))*(diff(PTC_m,x8,1))*covarx3x8/n;
covarx3_9:(diff(PTC_m,x3,1))*(diff(PTC_m,x9,1))*covarx3x9/n;
covarx3_10:(diff(PTC_m,x3,1))*(diff(PTC_m,x10,1))*covarx3x10/n;
covarx3_11:(diff(PTC_m,x3,1))*(diff(PTC_m,x11,1))*covarx3x11/n;
covarx3_y:(diff(PTC_m,x3,1))*(diff(PTC_m,y,1))*covarx3y/n;
covarx4_5:(diff(PTC_m,x4,1))*(diff(PTC_m,x5,1))*covarx4x5/n;
covarx4_6:(diff(PTC_m,x4,1))*(diff(PTC_m,x6,1))*covarx4x6/n;
covarx4_7:(diff(PTC_m,x4,1))*(diff(PTC_m,x7,1))*covarx4x7/n;
covarx4_8:(diff(PTC_m,x4,1))*(diff(PTC_m,x8,1))*covarx4x8/n;
covarx4_9:(diff(PTC_m,x4,1))*(diff(PTC_m,x9,1))*covarx4x9/n;
covarx4_10:(diff(PTC_m,x4,1))*(diff(PTC_m,x10,1))*covarx4x10/n;
covarx4_11:(diff(PTC_m,x4,1))*(diff(PTC_m,x11,1))*covarx4x11/n;
covarx4_y:(diff(PTC_m,x4,1))*(diff(PTC_m,y,1))*covarx4y/n;
covarx5_6:(diff(PTC_m,x5,1))*(diff(PTC_m,x6,1))*covarx5x6/n;
covarx5_7:(diff(PTC_m,x5,1))*(diff(PTC_m,x7,1))*covarx5x7/n;
covarx5_8:(diff(PTC_m,x5,1))*(diff(PTC_m,x8,1))*covarx5x8/n;
covarx5_9:(diff(PTC_m,x5,1))*(diff(PTC_m,x9,1))*covarx5x9/n;
covarx5_10:(diff(PTC_m,x5,1))*(diff(PTC_m,x10,1))*covarx5x10/n;
covarx5_11:(diff(PTC_m,x5,1))*(diff(PTC_m,x11,1))*covarx5x11/n;
```

```
covarx5_y:(diff(PTC_m,x5,1))*(diff(PTC_m,y,1))*covarx5y/n;
covarx6_7:(diff(PTC_m,x6,1))*(diff(PTC_m,x7,1))*covarx6x7/n;
covarx6_8:(diff(PTC_m,x6,1))*(diff(PTC_m,x8,1))*covarx6x8/n;
covarx6_9:(diff(PTC_m,x6,1))*(diff(PTC_m,x9,1))*covarx6x9/n;
covarx6_10:(diff(PTC_m,x6,1))*(diff(PTC_m,x10,1))*covarx6x10/n;
covarx6_11:(diff(PTC_m,x6,1))*(diff(PTC_m,x11,1))*covarx6x11/n;
covarx6_y:(diff(PTC_m,x6,1))*(diff(PTC_m,y,1))*covarx6y/n;
covarx7_8:(diff(PTC_m,x7,1))*(diff(PTC_m,x8,1))*covarx7x8/n;
covarx7_9:(diff(PTC_m,x7,1))*(diff(PTC_m,x9,1))*covarx7x9/n;
covarx7_10:(diff(PTC_m,x7,1))*(diff(PTC_m,x10,1))*covarx7x10/n;
covarx7_11:(diff(PTC_m,x7,1))*(diff(PTC_m,x11,1))*covarx7x11/n;
covarx7_y:(diff(PTC_m,x7,1))*(diff(PTC_m,y,1))*covarx7y/n;
covarx8_9:(diff(PTC_m,x8,1))*(diff(PTC_m,x9,1))*covarx8x9/n;
covarx8_10:(diff(PTC_m,x8,1))*(diff(PTC_m,x10,1))*covarx8x10/n;
covarx8_11:(diff(PTC_m,x8,1))*(diff(PTC_m,x11,1))*covarx8x11/n;
covarx8_y:(diff(PTC_m,x8,1))*(diff(PTC_m,y,1))*covarx8y/n;
covarx9_10:(diff(PTC_m,x9,1))*(diff(PTC_m,x10,1))*covarx9x10/n;
covarx9_11:(diff(PTC_m,x9,1))*(diff(PTC_m,x11,1))*covarx9x11/n;
covarx9_y:(diff(PTC_m,x9,1))*(diff(PTC_m,y,1))*covarx9y/n;
covarx10_11:(diff(PTC_m,x10,1))*(diff(PTC_m,x11,1))*covarx10x11/n;
covarx10_y:(diff(PTC_m,x10,1))*(diff(PTC_m,y,1))*covarx10y/n;
covarx11_y:(diff(PTC_m,x11,1))*(diff(PTC_m,y,1))*covarx11y/n;
/* The program continues computing the error variance*/
covariance_part:2*(covarx1_2+covarx1_3+covarx1_4+covarx1_5+
covarx1_6+covarx1_7+covarx1_8+covarx1_9+covarx1_10+covarx1_11+covarx1_y+
covarx2_3+covarx2_4+covarx2_5+covarx2_6+covarx2_7+covarx2_8+covarx2_9+covarx2_10+
covarx2_11+covarx2_y+covarx3_4+covarx3_5+covarx3_6+covarx3_7+covarx3_8+
covarx3_9+covarx3_10+covarx3_11+covarx3_y+covarx4_5+covarx4_6+covarx4_7+covarx4_8+
covarx4_9+covarx4_10+covarx4_11+covarx4_y+covarx5_6+covarx5_7+covarx5_8+covarx5_9+
covarx5_10+covarx5_11+covarx5_y+covarx6_7+covarx6_8+covarx6_9+covarx6_10+covarx6_11+
covarx6_y+covarx7_8+covarx7_9+covarx7_10+covarx7_11+covarx7_y+covarx8_9+covarx8_10+
covarx8_11+covarx8_y+covarx9_10+covarx9_11+covarx9_y+covarx10_11+covarx10_y+covarx11_y);
variance_part:varianzas1+varianzas2+varianzas3+varianzas4+varianzas5+varianzas6+varianzas7+
varianzas8+varianzas9+varianzas10+varianzas11+varianzasy;
/* The standard error is computed without the finite population correction (fpc)*/
standarderror:(variance_part+covariance_part)^0.5, numer;
/* The standard error is computed wit the finite population correction (fpc)*/
standarderrorfpc:standarderror*(((N-n)/(N-1))^0.5), numer;
/* The standard error function is evaluated at the mean values of the 12 variables)*/
standarderrorfinal: ev(standarderrorfpc,
x1=meanvector [1],
x2=meanvector [2],
x3=meanvector [3],
x4=meanvector [4],
x5=meanvector [5],
x6=meanvector [6],
x7=meanvector [7],
x8=meanvector [8],
x9=meanvector [9],
x10=meanvector [10],
x11=meanvector [11],
y=meanvector [12]);
/* The final information is provided. Now we have the Player Total Contribution per minute at the
end of the season with a relative error and a confidence interval*/
Player:Player;
PTC_MP:(w1*meanvector[1]+w2*meanvector[2]+w3*meanvector[3]+w4*meanvector[4]+
w5*meanvector[5]+w6*meanvector[6]+w7*meanvector[7]+w8*meanvector[8]+w9*meanvector[9]+
w10*meanvector[10]+w11*meanvector[11])/(w12*meanvector[12]), numer;
Relative_error:100*(Z*standarderrorfinal)/PTC_MP;
IC95low:PTC_MP-(Z*standarderrorfinal), numer;
IC95high:PTC_MP+(Z*standarderrorfinal), numer;
/* End of the program*/
```

5. Acknowledgments

This study is the result of the activity carried out under the program Groups of Excellence of the region of Murcia, the Fundación Séneca, Science and Technology Agency of the region of Murcia project 19884/GERM/15.

6. References

1. Berri DJ, Schmidt MB, Brook SL. The wages of wins: Taking measure of the many myths in modern sport. Palo Alto, CA: Stanford University Press, 2006.
2. Winston WL. Mathletics. New Yersey: Princeton University Press Berri, D.J. Who is most valuable? Measuring the player's

production of wins in the National Basketball Association. Managerial and Decision Economics. 2009; 20(8):411-427.

3. Berri DJ. A simple measure of Worker productivity in the National Basketball Association. In Humphreys, B and Howard, D. (Eds): The Business of Sport.; Westport, Conn. Praeger, 2008, 1-40.
4. Berri DJ. Measuring performance in the National Basketball Association. In Stephen Shmanske, S. and Kahane, L. (Eds): The Handbook of Sports Economics. Oxford University, 2012.
5. Hollinger J. Pro basketball forecast: 2005-2006. Potomac Books Inc, 2005.
6. Martínez JA. Factors determining production (FDP) in basketball. Economics and Business Letters. 2012; 1(1):21-29.
7. McFarlane P. Evaluating NBA end-of-game decision-making. Journal of Sports Analytics. 2019; 5:17-22.
8. Martínez JA, Martínez L. Un método probabilístico para las clasificaciones estadísticas de jugadores en baloncesto. Revista Internacional de Ciencias del Deporte. 2010; 18(6):13-36.
9. Martínez JA. A more robust estimation and extension of factors determining production (FDP) of basketball players. International Journal of Physical Education, Sports and Health, 2019.
10. Ku HH. Notes on the use of propagation error formulas. Journal of Research of the National Bureau of Standards-C. Engineering and Instrumentation. 1966; 70(4):263-273.
11. Levy PS, Lemeshow S. Sampling of populations: Methods and applications (3ª ed). Wiley series in probability and statistics. Survey Methodology Section, 1999.