An Application of the Mapper Algorithm to Sports Analytics in College Basketball

Towson University

Derek Margulies
Advisor: Dr. Christopher Cornwell

19 May 2020

Abstract

We use a novel method to analyze data from college basketball games. The data are obtained from the play-by-play logs from games played by Towson University's Men's Basketball team. After initial preprocessing, a set of data points is created, one for each of several time intervals during each game, and for each on-court player. A plus/minus statistic is also calculated and associated to each point. We visualize this data set with a network of nodes and edges, *i.e.,* a *graph*, by using the Mapper Algorithm (Singh, Mémoli, and Carlsson 2007). In our use of Mapper, we consider points with similar Four Factors stats as part of the same node. A metric is used to add in edges as Four Factors stats vary, and the variation of a plus/minus statistic is seen as you traverse the graph. The goal is to find Four Factors profiles that consistently appear together on the court and in conjunction with a very positive or a very negative plus/minus statistic. Such knowledge may be applied by the coaching staff to identify optimal lineups at key moments in games.

1 Introduction

Basketball analytics is a rapidly growing field within mathematics at both the collegiate and professional levels. In 2013, then-graduate student Drew Cannon compiled written reports for Butler University's basketball coaching staff, providing detailed analysis on opponents before each game [13]. Professional statisticians Wayne Winston and Jeff Sagarin advise the Dallas Mavericks, an NBA team, using a system that determines which lineups to use during games and which free agents to sign [4]. This project is designed to provide a similar advisory role to Towson University's Men's basketball coaching staff.

The source of analysis for this project is play-by-play data. The data are available at the Towson Tigers Men's basketball website [14] as text-based logs of events within each game. The play-by-play data are converted from a text-based format to a quantified data format using a collection of statistics called the Four Factors: the four game parameters that break down a player's offensive and defensive impact.

In basketball, offensive and defensive ratings provide summaries of how a team performs on a per-possession basis [3], [8]. The Four Factors—introduced by Dean Oliver, a professional statistician and assistant coach for the NBA's Washington Wizards—provide a breakdown of offensive and defensive ratings for teams [3], [8].

The technique of evaluating and analyzing Four Factors has been implemented across multiple decades of NBA seasons. Offensively, a team wants to minimize turnovers per possession and maximize the other three factors; defensively, a team wants to maximize turnovers per possession and minimize the other three factors. One goal of this project is to analyze college play-by-play data using the Four Factors, which was implemented at the professional level. Once Four Factors have been generated for players, they are visualized using the Mapper algorithm, a visualization tool used in topological data analysis.

This project is a novel application of topology to sports analytics because of the very few number of publications available on applying the Mapper algorithm to sports data. A literature search showed sparse examples of applying Mapper to sports analytics, namely professional hockey (NHL) and professional basketball (NBA) [1], [2]. The NHL study uses Mapper to analyze deficiencies in a team's composition and player trades & acquisitions [2]. The NBA study uses Mapper to identify 13 positions in basketball that more accurately reflect different play styles (as opposed to the traditional 5 positions) [1]. While both studies use Mapper, the goals of this project differ from those of the two previous studies, making the approach of this project unique.

The means for data visualization and analysis in this project is based on an application of the Mapper algorithm in precision medicine. In this study, researchers used Mapper to construct a precision model to characterize the complexity of Type 2 diabetes patient populations based on electronic medical records and genotype data [5]. This study found strong similarities among different subgroups of Type 2 diabetes from high-dimensional relationships that could not be observed by hand, such as laboratory tests [5]. Once patients were distinguished into subgroups, statistical methods were applied to clarify treatment options and sources of the disease for each subgroup [5]. Ultimately, three subgroups of Type 2 diabetes were found using the model, suggesting the idea that more than two types of diabetes exist [5]. Figure 1 shows the models generated from the study.

In the context of this project, similarities that are not apparent in one- or two-dimensions between different players and lineups that have historically made the team successful in a game are identified. The intention is not to consider high-performing players as a single type, but to consider the best lineup combinations to optimize the team's chances of winning games. These determinations are based on the Four Factors generated by each individual player for multiple games throughout a season.

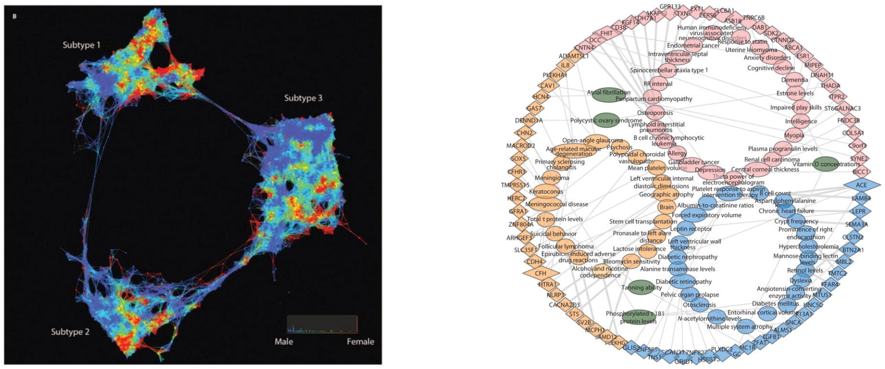


Figure 1: Left: Patient-patient network for 2,551 T2 diabetes patients. Right: Genotype-phenotype network for three subtypes of Type 2 diabetes, which are indicated by different colors. Ellipses represent phenotypes, parallelograms represent genes, and green ellipses represent shared phenotypes between different subgroups. All edges represent p-values; edge thickness describes significance of the p-value. From Li et al., 2015.

2 Data Pre-Processing

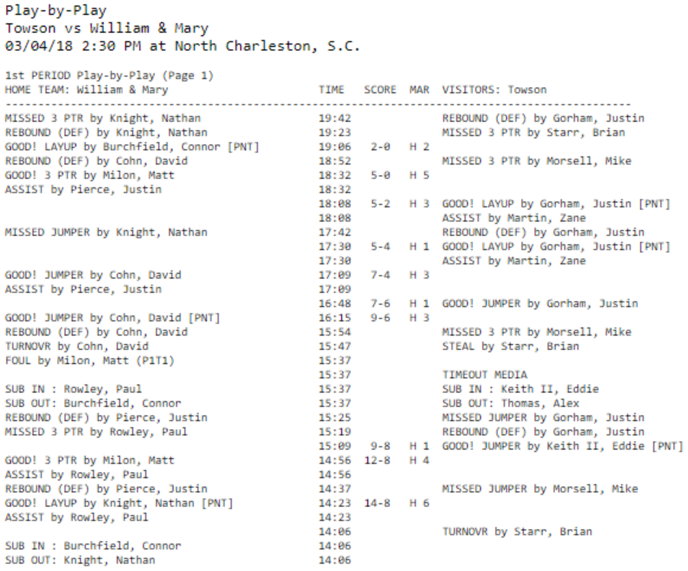
The code made to carry out data pre-processing can be found at the following Github repository:

https://github.com/dlymarg/tiger-bball

The play-by-play data come directly from the Towson Tigers basketball statistics webpage [14] and are downloaded as HTML files. All games are sorted by seasons and each game page consists of important information, such as the rosters for both teams, the box score, and—what is used extensively in this project—a play-by-play log. The play-by-play data are available in a raw, text-based table format that include events, players that committed events within plays, and a running track of the timestamps of different events (Figure 2).

A Python program parses each game webpage via BeautifulSoup [10]. The program locates the play-by-play log and the team rosters within each HTML file based on certain string patterns. Figure 2 shows the format of each play-by-play log. Two unique features of the play-by-play log are the long string of hyphens at the top of the play-by-play table and the change in font size from the header of the table to the events within the log. These patterns are used to tell the program where the play-by-play log starts. Once the program has located the start of the play-by-play log, it parses relevant information: it records information related to events that occurred within each play (Note: *events* are defined to be actions that players commit, such as field goals, whereas *plays* are defined to be series of events within the same timestamp, such as a missed field goal and a defensive rebound). The recorded information includes events such as field goal attempts, fouls, turnovers, rebounds, and substitutions & timeouts (if applicable); the players responsible for each event; and the time at which events occur. This information is also retrieved using string patterns, such as “GOOD!” and “MISSED” for shot attempts, “TURNOVR” for turnovers, and “REBOUND” for rebounds.

One important feature needed to distinguish different types of shots and rebounds is to examine what comes right after those keywords. This allows for the ability to assign point values to expressions that refer to scoring events. For example, 3 pointers, 2-point field goals, and free throws all have the string “GOOD!” and “MISSED” associated with them; to distinguish these types of shots from one another, a condition is applied that declares if “3 PTR” is seen next to either “GOOD!” or “MISSED”, record 3 points happening within the play. If not, then check to see if “FT SHOT” is seen next to either of the same two keywords. If so, record 1 point happening within the play, but if



```
Play-by-Play
Towson vs William & Mary
03/04/18 2:30 PM at North Charleston, S.C.

1st PERIOD Play-by-Play (Page 1)
HOME TEAM: William & Mary                       TIME   SCORE  MAR  VISITORS: Towson
------------------------------------------------------------------------------------------------
MISSED 3 PTR by Knight, Nathan                  19:42              REBOUND (DEF) by Gorham, Justin
REBOUND (DEF) by Knight, Nathan                 19:23              MISSED 3 PTR by Starr, Brian
GOOD! LAYUP by Burchfield, Connor [PNT]         19:06   2-0   H 2
REBOUND (DEF) by Cohn, David                    18:52              MISSED 3 PTR by Morsell, Mike
GOOD! 3 PTR by Milon, Matt                      18:32   5-0   H 5
ASSIST by Pierce, Justin                        18:32
                                                18:08   5-2   H 3  GOOD! LAYUP by Gorham, Justin [PNT]
                                                18:08              ASSIST by Martin, Zane
MISSED JUMPER by Knight, Nathan                 17:42              REBOUND (DEF) by Gorham, Justin
                                                17:30   5-4   H 1  GOOD! LAYUP by Gorham, Justin [PNT]
                                                17:30              ASSIST by Martin, Zane
GOOD! JUMPER by Cohn, David                     17:09   7-4   H 3
ASSIST by Pierce, Justin                        17:09
                                                16:48   7-6   H 1  GOOD! JUMPER by Gorham, Justin
GOOD! JUMPER by Cohn, David [PNT]               16:15   9-6   H 3
REBOUND (DEF) by Cohn, David                    15:54              MISSED 3 PTR by Morsell, Mike
TURNOVR by Cohn, David                          15:47              STEAL by Starr, Brian
FOUL by Milon, Matt (P1T1)                      15:37
                                                15:37              TIMEOUT MEDIA
SUB IN : Rowley, Paul                           15:37              SUB IN : Keith II, Eddie
SUB OUT: Burchfield, Connor                     15:37              SUB OUT: Thomas, Alex
REBOUND (DEF) by Pierce, Justin                 15:25              MISSED JUMPER by Gorham, Justin
MISSED 3 PTR by Rowley, Paul                    15:19              REBOUND (DEF) by Gorham, Justin
                                                15:09   9-8   H 1  GOOD! JUMPER by Keith II, Eddie [PNT]
GOOD! 3 PTR by Milon, Matt                      14:56  12-8   H 4
ASSIST by Rowley, Paul                          14:56
REBOUND (DEF) by Pierce, Justin                 14:37              MISSED JUMPER by Morsell, Mike
GOOD! LAYUP by Knight, Nathan [PNT]             14:23  14-8   H 6
ASSIST by Rowley, Paul                          14:23
                                                14:06              TURNOVR by Starr, Brian
SUB IN : Burchfield, Connor                     14:06
SUB OUT: Knight, Nathan                         14:06
```

Figure 2: Play-by-play data from a game in 2018.

not, then anything else should be considered 2 points. This approach also helps with accommodating for the variety of 2-point field goals, such as jump shots, layups, and dunks.

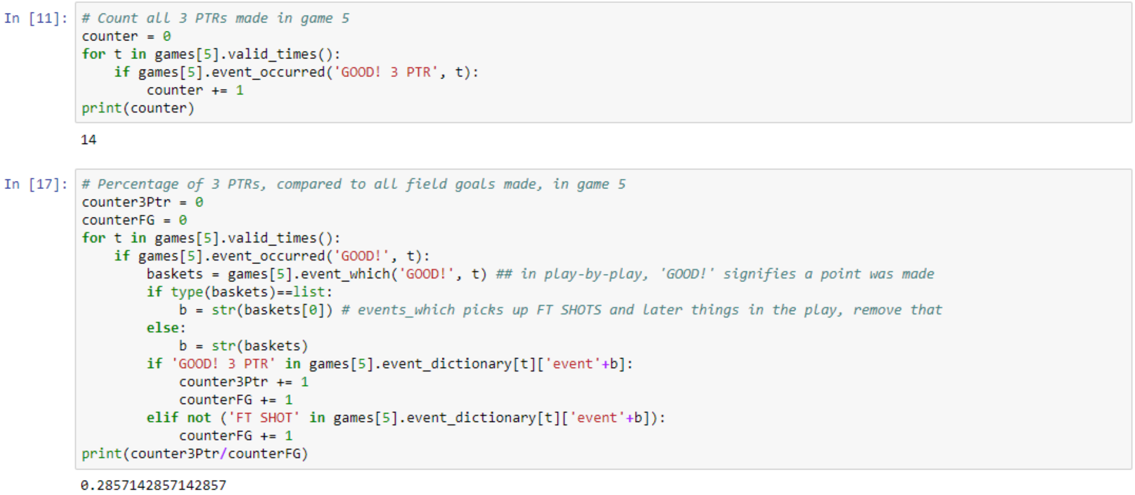
The program not only extracts the text in a play-by-play log, but it also performs two computations:

1. Ordering of events within a play
2. Calculation of points Towson scores and points opponent scores within a play.

The program classifies events based on the order in which they occur (it was determined that as many as 6 events can occur within a play) while keeping track of the player responsible for each event. It also calculates the number of points scored within a play by searching for keywords within the dictionary, such as "GOOD! JUMPER" (2 points) or "GOOD! 3 PTR" (3 points), and assigns the number of points scored in a play to the appropriate team based on which roster the scoring player belongs to. By collecting this information, the raw, text-based data can be converted to a data format that allows for computations. For example, the number of total 3 pointers and the percentage of 3 pointers compared to all field goals can be computed for a single game (Figure 3).

Like many real-world data, there were some difficulties in data pre-processing. One notable difficulty was in the method some substitutions were recorded. In Figure 4, the top two boxed-in regions show instances when Alex Thomas contributes to some plays; however, the bottom boxed-in region shows Alex Thomas being subbed in, which is a discrepancy. In other words, Alex Thomas cannot be subbed in if he was just on the court making plays shortly before the play-by-play log indicates he was subbed in. To avoid potential problems in the analysis, an approximation of the amount of time spent active in a specified time interval was computed to understand a player's involvement in plays. This approximation is made by dividing the number of times a player contributes to a play in a time interval by the number of timestamps in the same time interval. Computing this becomes useful when trying to consider data points that contribute towards the team's chances of winning either positively or negatively. The computation also allows for the ability to consider players who have a sufficiently high amount of activity in a time interval.

All the extracted data for a single game is stored in a Python dictionary with timestamps of plays as keys to the dictionary; each key consists of the sequence of events, players associated to each



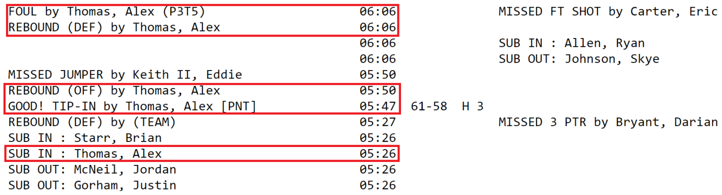
```
In [11]: # Count all 3 PTRs made in game 5
         counter = 0
         for t in games[5].valid_times():
             if games[5].event_occurred('GOOD! 3 PTR', t):
                 counter += 1
         print(counter)

         14

In [17]: # Percentage of 3 PTRs, compared to all field goals made, in game 5
         counter3Ptr = 0
         counterFG = 0
         for t in games[5].valid_times():
             if games[5].event_occurred('GOOD!', t):
                 baskets = games[5].event_which('GOOD!', t) ## in play-by-play, 'GOOD!' signifies a point was made
                 if type(baskets)==list:
                     b = str(baskets[0]) # events_which picks up FT SHOTS and later things in the play, remove that
                 else:
                     b = str(baskets)
                 if 'GOOD! 3 PTR' in games[5].event_dictionary[t]['event'+b]:
                     counter3Ptr += 1
                     counterFG += 1
                 elif not ('FT SHOT' in games[5].event_dictionary[t]['event'+b]):
                     counterFG += 1
         print(counter3Ptr/counterFG)

         0.2857142857142857
```

Figure 3: An example of data quantification. Computations are executed from the extracted text-based data, such as the number of three pointers made in a game and the percentage of three pointers in comparison to all field goals in a game. The first cell shows that, between both teams, game 5 had fourteen converted 3 pointers. The second cell shows that, between both teams, 28.6% of all successful field goals in game 5 were 3 pointers.



```
FOUL by Thomas, Alex (P3T5)            06:06              MISSED FT SHOT by Carter, Eric
REBOUND (DEF) by Thomas, Alex          06:06
                                       06:06              SUB IN : Allen, Ryan
                                       06:06              SUB OUT: Johnson, Skye
MISSED JUMPER by Keith II, Eddie       05:50
REBOUND (OFF) by Thomas, Alex          05:50
GOOD! TIP-IN by Thomas, Alex [PNT]     05:47  61-58  H 3
REBOUND (DEF) by (TEAM)                05:27              MISSED 3 PTR by Bryant, Darian
SUB IN : Starr, Brian                  05:26
SUB IN : Thomas, Alex                  05:26
SUB OUT: McNeil, Jordan                05:26
SUB OUT: Gorham, Justin                05:26
```

Figure 4: An example of messiness in the data. Alex Thomas is incorrectly recorded as being involved in plays shortly before being subbed in without being subbed out. From Towson Tigers.

event, substitutions & timeouts, and points scored by each team. This dictionary is referred to as the *event dictionary*. Once all the data from all games have been stored in their event dictionaries, they are converted to a suitable data format that allows for computations; this conversion is implemented with Dean Oliver's Four Factors.

3 Methods

The code made to carry out data analysis can be found at the previously-mentioned Github repository:

https://github.com/dlymarg/tiger-bball

3.1 Play-By-Play Data Extraction & Quantification

Professional statistician Dean Oliver identified four factors that are important in breaking down a player's offensive and defensive rating in moments of a National Basketball Association (NBA) game. The four factors are computed for each on-court player over a specified time interval. These four factors are:

- Effective field goal percentage
- Turnovers per possession (TO/Poss)
- Offensive rebounding percentage
- Free throw rate

Free throw rate is a combination of two parameters: the number of times a player visits the foul line and the player's free throw shooting percentage. In reality, these factors can be thought of as five factors. One additional factor that is kept track of in this project is the field goals attempted. The purpose of keeping track of this statistic is to better understand the meaning of the effective field goal percentage. For example, suppose one player converts all 4 of his field goal attempts and another player converts his one and only field goal attempt. Although both scenarios indicate a 100% effective field goal percentage, a player that makes all 4 field goals attempted has a greater contribution to the team's score over a time interval than a player who made his one and only field goal attempt. All this information is stored in a 6-dimensional vector. For the purpose of performing computations on the data set in Python, the 6 factors are stored in a NumPy array (a NumPy array is Python's version of storing data in a vector-like structure and having the ability to perform vector operations) [7]. A single Four Factors vector is called a *Four Factors profile.*

To obtain these statistics, a Python program computes the Four Factors arrays for each on-court player from the data stored in the event dictionary over a specified time interval. The initial analysis generated Four Factors profiles over three ten-minute intervals with five-minute overlaps for each half (*i.e.,* 20 minutes left in the half to 10 minutes left in the half, 15 minutes to 5 minutes, and 10 minutes to 0 minutes). 10-minute intervals were chosen because of how nicely 10 minutes divides the time in a single half of collegiate basketball, which is played in two 20-minute halves. So, a single data point in the data set corresponds to a single Four Factors array generated for one player over a 10-minute interval. Later analysis was done over six-minute intervals with four-minute overlaps for each half, which gives 8 intervals for each half.

One other computation that is recorded is a variation of the plus/minus statistic. Plus/minus statistics are traditionally computed for each player and take the difference between the number of points a player's team scores and the number of points a player's team allows while that player is on the court. So, a negative plus/minus statistic for a player indicates that the team allows more points when that player is on the court; conversely, a positive plus/minus statistic for a player indicates that the team scores more points than it allows when that player is on the court. The variation of the plus/minus statistic used performs the same computations as a traditional plus/minus statistic, but imposes two conditions:

1. Point margins are only computed within a specified time interval
2. Point margins do not take the current score into account; it only keeps track of the number of points scored between the two teams within a specified time interval

These two conditions are important because finding which Four Factors profiles correspond to highly positive and highly negative plus/minus statistics inform the coaching staff of not only the kinds of plays that contribute to positive and negative outcomes, but also the players that are regularly associated with positive and negative outcomes. By only considering the points scored and allowed within a specified time interval, behaviors that lead to positive or negative outcomes are more easily identified regardless of the current score.

The Four Factors profiles coupled with the variation of the plus/minus statistic draw conclusions that inform the coaching staff of which players should be on the court in critical moments in games. This variation of the plus/minus statistic is crucial for the implementation of the Mapper algorithm.

3.2 Application of the Mapper Algorithm

The Mapper algorithm is a visualization tool used in topological data analysis that draws similarities in high-dimensional data. The algorithm constructs these similarities by reducing high-dimensional data to simplicial complexes (this project uses a graph for visualization) and capturing geometric and topological information from a data set.

The Mapper algorithm starts with an arbitrary data set X and uses a filter function $f\colon X \to \mathbb{R}$. As shown on $\mathbb{R}$ in Figure 5, the real line is divided into subintervals that cover the image of X under the function f. Each data point is assigned to its corresponding subinterval within $\mathbb{R}$ based on the filter function f; it is given a color based on its corresponding subinterval. Then, the algorithm looks at the preimage of the data points and identifies the number of data points and the number of clusters in a subinterval.

The number of data points in a cluster is used to determine the size of each node in the final Mapper graph and the number of clusters in a subinterval determine the number of nodes associated with that subinterval in the final Mapper graph. The clusters within each subinterval are determined by the density-based spatial clustering of applications with noise (DBSCAN) algorithm. In Figure 5, there appears to be 1 cluster in the yellow/top subinterval and 3 clusters in the teal/next subinterval. So, there should be 1 yellow node and 3 teal nodes, where the yellow node is the largest of the 4 mentioned nodes and one of the 3 teal nodes is considerably smaller due to the fewer number of data points in one of the clusters. Notice that in the overlap between the two subintervals, there appears to be 1 shared cluster. When this occurs, the graph gains an edge between the only yellow node and the smallest teal node. At this stage of the algorithm, there is a disconnected graph of order 4, size 1, and 3 components. This process repeats until the algorithm has generated a graph for all clusters in X across all subintervals within $\mathbb{R}$.

In the context of this project, X is represented by the Four Factors arrays, f is the variation of the plus/minus statistic, and the intervals represent subintervals of the range of plus/minus statistics encountered within the data set. Towson had plus/minus statistics as high as +20 and as low as -16. Given that the range of plus/minus statistics was 36, this allowed for a nice subinterval length of 9 points with an overlap of 3 points. Thus, there were 6 subintervals between +20 to -16.

3.3 The DBSCAN Algorithm

The DBSCAN algorithm is a density-based clustering algorithm, commonly implemented by the scikit-learn Python module [11], that identifies clusters within a data set based on two parameters: ε and MinPts. These parameters are important in determining whether a data point is a *core point*, a point that is *reachable* from a core point, or an *outlier/noise.*

A point is considered a *core point* if at least MinPts data points are within distance ε from it. A point is considered *reachable* if it is within distance ε of a core point, but either (1) does have at least MinPts data points within distance ε, or (2) does not have at least MinPts data points within distance ε. In (1), core points that are not the starting core point are also considered reachable; in (2), a point that is reachable (and is not itself a core point) is within distance ε of another core point, but has fewer than MinPts data points within distance ε of it. A point is considered an *outlier* if it is neither a core point nor reachable.

The metric used to measure the distance between data points can be determined by the user; the default setting is the Euclidean distance. DBSCAN has three advantages as a clustering algorithm: it identifies areas of high and low density within a data set, it distinguishes noise in a data set, and it does not require a pre-determined number of clusters (one of the parameters in the k-means clustering algorithm is the number of desired clusters).

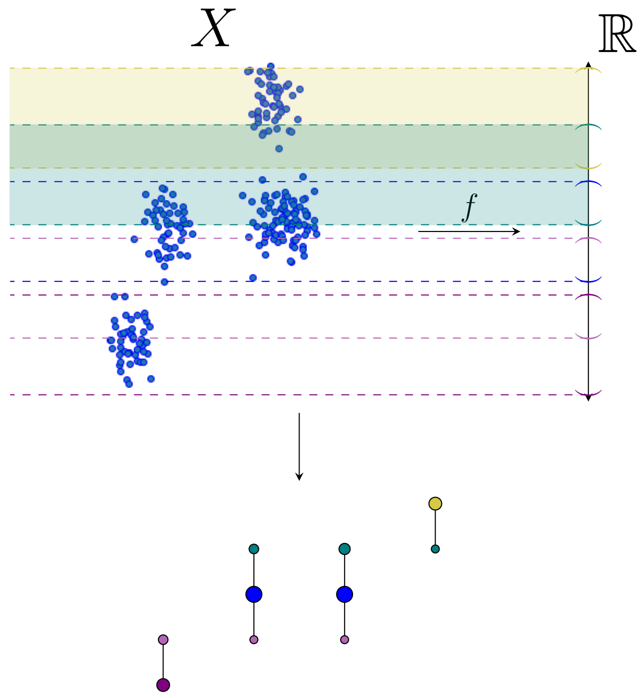


Figure 5: An overview of the Mapper algorithm on an arbitrary dataset, X, with filter function f. The output of the algorithm is a graph.

4 Results

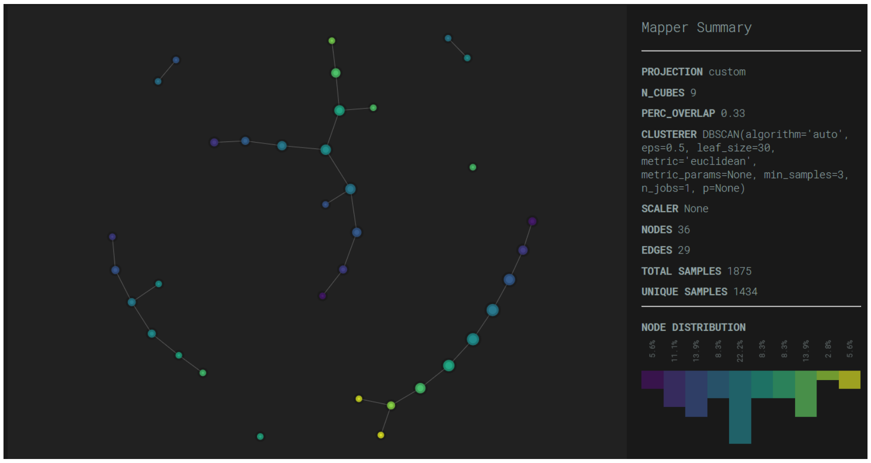
A Mapper graph was generated for the 2017-18 season using ten-minute intervals with five-minute overlaps for each half (Figure 6). The resulting Mapper graph shows multiple components with transitions from darker, more purple nodes to lighter, more yellow nodes. These nodes refer to Four Factors profiles that correspond to negative plus/minus statistics and positive plus/minus statistics, respectively. There are two questions that must be addressed when analyzing the Mapper graph:

1. What sorts of plays/play styles encourage a transition from negative plus/minus statistics to positive plus/minus statistics, and

2. What are the features that distinguish one component of the graph from another component?

These questions can be answered using linear regression analysis using the Four Factors. It is possible that linear regression analysis suggests an increase (or decrease) of a certain factor within the Four Factors leads to a more favorable plus/minus statistic. Moreover, different combinations of factors can distinguish components within the Mapper graph. The idea is that, for example, perhaps one component corresponds to greater field goal shooting efficiency while another component corresponds to fewer turnovers per possession.

Figure 7 shows linear regression analysis on the component at the bottom-right corner in the Mapper graph. The analysis suggests that turnovers per possession is the factor most closely associated with the component. Effective field goal percentage can also be considered as a factor closely associated with the component, but given the R^2 values of each factor, turnovers per possession would be a better indicator of the play style that should be emphasized in that component. The results from this analysis agree with how any ordinary person would assume a team should score more: minimize turnovers and convert more shots. However, linear regression does not become



```
Mapper Summary

PROJECTION custom
N_CUBES 9
PERC_OVERLAP 0.33
CLUSTERER DBSCAN(algorithm='auto',
eps=0.5, leaf_size=30,
metric='euclidean',
metric_params=None, min_samples=3,
n_jobs=1, p=None)
SCALER None
NODES 36
EDGES 29
TOTAL SAMPLES 1875
UNIQUE SAMPLES 1434

NODE DISTRIBUTION
```

Figure 6: The Mapper graph generated using the time intervals 20:00-10:00, 15:00-05:00, and 10:00-00:00 in each half. Each color corresponds to a particular plus/minus statistic: lighter, more yellow points are attributed to more positive plus/minus statistics while darker, more purple points are attributed to more negative plus/minus statistics.

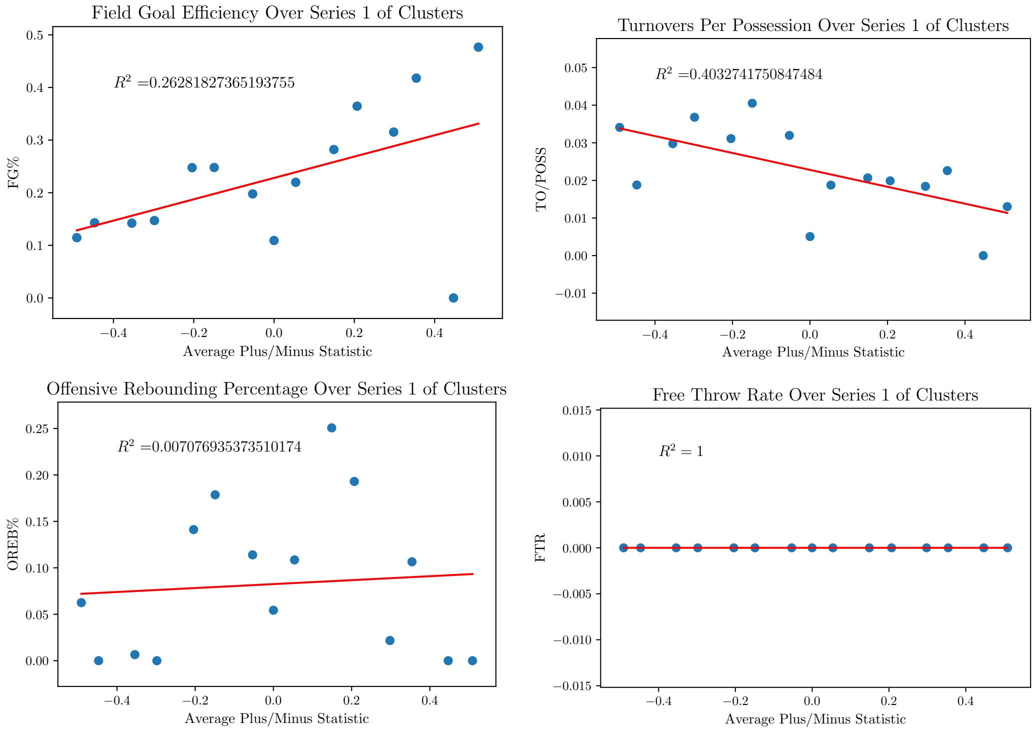


Figure 7: Linear regression analysis on the component at the bottom-right corner of the Mapper graph. The x-axis looks at the average plus/minus statistic starting at the purple node and ends at the green node of degree 3.

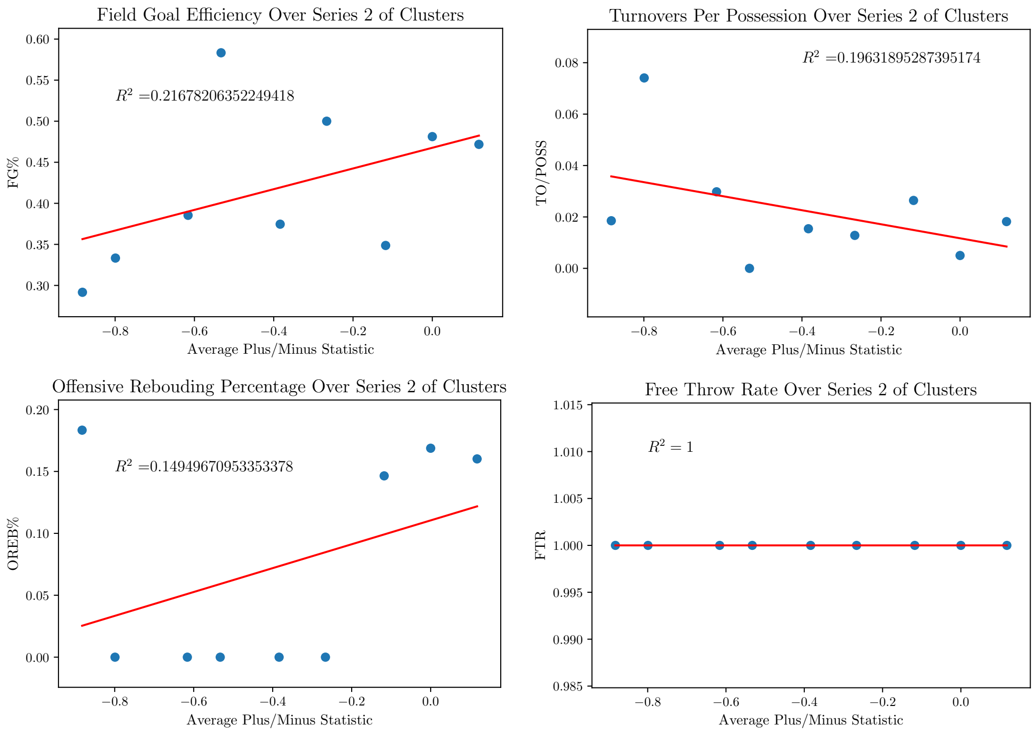


Figure 8: Linear regression analysis on an induced subgraph of the component at the center of the Mapper graph. The analysis specifically pertains to the nodes that run from the center of the entire graph to the teal node of degree 3 adjacent to two darker deal nodes and one lighter green node.

as informative of what play styles the coaching staff should encourage in some components of the Mapper graph.

Figure 8 shows linear regression analysis on an induced subgraph of the component at the center of the Mapper graph (see caption for exact location). From this analysis, it is not exactly clear what combination of factors are associated with the induced subgraph. Although there is a similar trend in effective field goal percentage and turnovers per possession as shown in Figure 7, the R^2 values of both factors are low. Thus, a different approach to analysis of the Mapper graph using the Four Factors must be taken. An alternative approach can be made by converting the Mapper graph to a NetworkX graph.

NetworkX is a module in Python that allows for the creation, manipulation, and analysis of structure, dynamics, and functions of networks (*i.e.*, graphs) [6]. This module gives the ability to show different methods of viewing how to position nodes and edges in a graph. The conversion from a Mapper graph to a NetworkX graph allows for more expansive analysis. Figure 9 shows the NetworkX version of the Mapper graph. Note that the layout of the NetworkX graph is identical to that of the Mapper graph, but does not parameters shown in the Mapper graph, such as node size and node color.

5 Future Work

To provide more accurate analysis of play-by-play data, other seasons' data must be extracted and quantified. Some adjustments need to be made to accommodate for differences in formatting in different seasons. Figures 10 and 11 are from play-by-play data in 2013 and 2020, respectively.

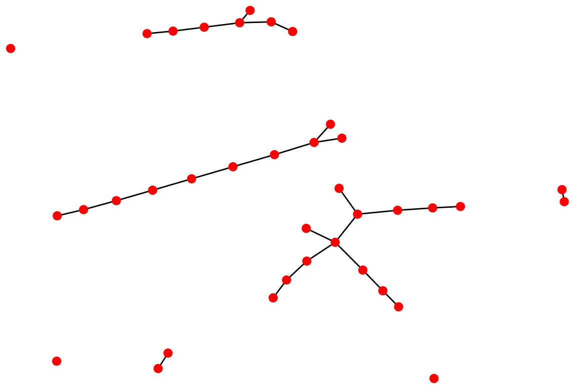
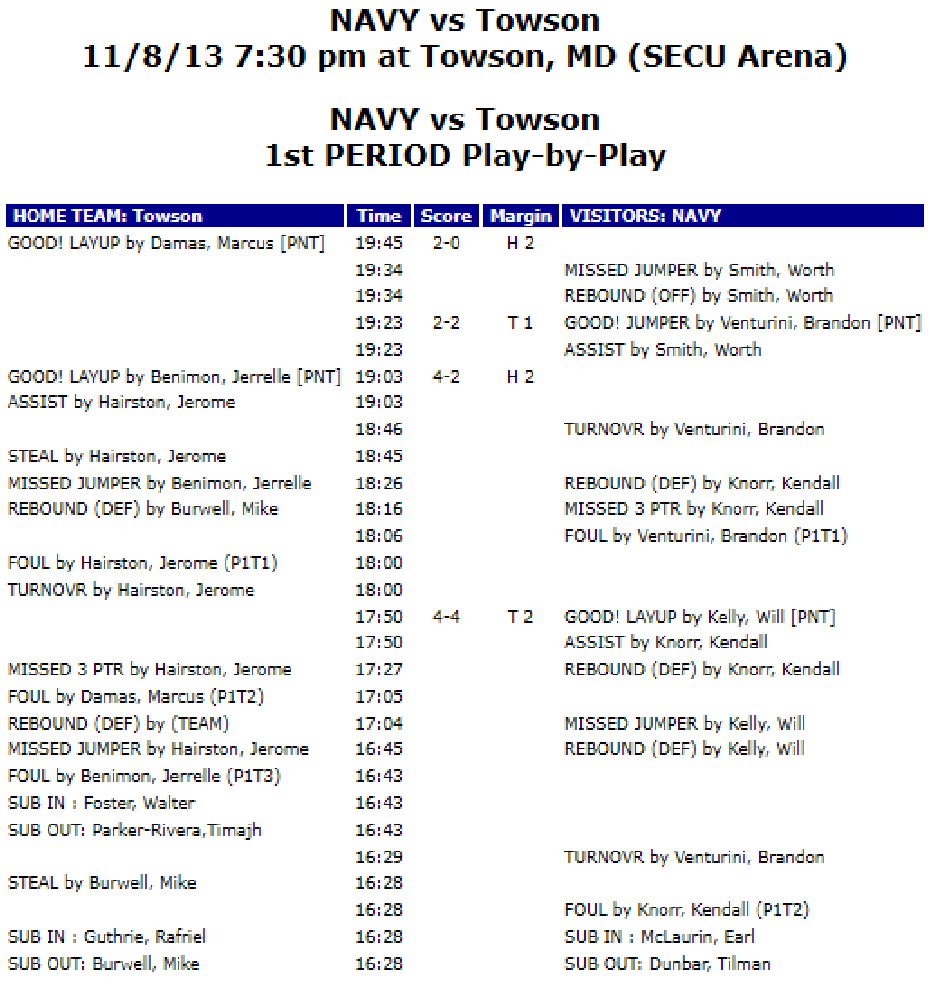


Figure 9: The NetworkX version of the Mapper graph. Note the graphs have identical layouts.

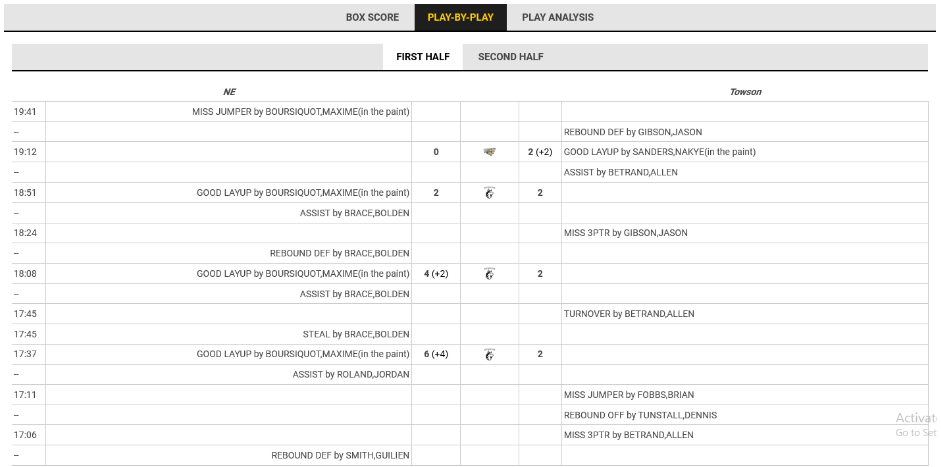


NAVY vs Towson
11/8/13 7:30 pm at Towson, MD (SECU Arena)

NAVY vs Towson
1st PERIOD Play-by-Play

HOME TEAM: Towson	Time	Score	Margin	VISITORS: NAVY
GOOD! LAYUP by Damas, Marcus [PNT]	19:45	2-0	H 2	
	19:34			MISSED JUMPER by Smith, Worth
	19:34			REBOUND (OFF) by Smith, Worth
	19:23	2-2	T 1	GOOD! JUMPER by Venturini, Brandon [PNT]
	19:23			ASSIST by Smith, Worth
GOOD! LAYUP by Benimon, Jerrelle [PNT]	19:03	4-2	H 2	
ASSIST by Hairston, Jerome	19:03			
	18:46			TURNOVR by Venturini, Brandon
STEAL by Hairston, Jerome	18:45			
MISSED JUMPER by Benimon, Jerrelle	18:26			REBOUND (DEF) by Knorr, Kendall
REBOUND (DEF) by Burwell, Mike	18:16			MISSED 3 PTR by Knorr, Kendall
	18:06			FOUL by Venturini, Brandon (P1T1)
FOUL by Hairston, Jerome (P1T1)	18:00			
TURNOVR by Hairston, Jerome	18:00			
	17:50	4-4	T 2	GOOD! LAYUP by Kelly, Will [PNT]
	17:50			ASSIST by Knorr, Kendall
MISSED 3 PTR by Hairston, Jerome	17:27			REBOUND (DEF) by Knorr, Kendall
FOUL by Damas, Marcus (P1T2)	17:05			
REBOUND (DEF) by (TEAM)	17:04			MISSED JUMPER by Kelly, Will
MISSED JUMPER by Hairston, Jerome	16:45			REBOUND (DEF) by Kelly, Will
FOUL by Benimon, Jerrelle (P1T3)	16:43			
SUB IN : Foster, Walter	16:43			
SUB OUT: Parker-Rivera,Timajh	16:43			
	16:29			TURNOVR by Venturini, Brandon
STEAL by Burwell, Mike	16:28			
	16:28			FOUL by Knorr, Kendall (P1T2)
SUB IN : Guthrie, Rafriel	16:28			SUB IN : McLaurin, Earl
SUB OUT: Burwell, Mike	16:28			SUB OUT: Dunbar, Tilman

Figure 10: Play-by-play data from 2013



	NE				*Towson*
19:41	MISS JUMPER by BOURSIQUOT,MAXIME(in the paint)				
--					REBOUND DEF by GIBSON,JASON
19:12		**0**		**2 (+2)**	GOOD LAYUP by SANDERS,NAKYE(in the paint)
--					ASSIST by BETRAND,ALLEN
18:51	GOOD LAYUP by BOURSIQUOT,MAXIME(in the paint)	**2**		**2**	
--	ASSIST by BRACE,BOLDEN				
18:24					MISS 3PTR by GIBSON,JASON
--	REBOUND DEF by BRACE,BOLDEN				
18:08	GOOD LAYUP by BOURSIQUOT,MAXIME(in the paint)	**4 (+2)**		**2**	
--	ASSIST by BRACE,BOLDEN				
17:45					TURNOVER by BETRAND,ALLEN
17:45	STEAL by BRACE,BOLDEN				
17:37	GOOD LAYUP by BOURSIQUOT,MAXIME(in the paint)	**6 (+4)**		**2**	
--	ASSIST by ROLAND,JORDAN				
17:11					MISS JUMPER by FOBBS,BRIAN
--					REBOUND OFF by TUNSTALL,DENNIS
17:06					MISS 3PTR by BETRAND,ALLEN
--	REBOUND DEF by SMITH,GUILIEN				

Figure 11: Play-by-play data from 2020

The 2013 play-by-play data look somewhat similar to the 2017-18 play-by-play data, but the table headers in the 2013 data look different and the font size of the header is significantly greater than that of the 2017-18 data. The 2013 data are stored in HTML-based tables, which makes data extraction easier compared to the 2017 data, which was completely text-based (both the data and the table). The 2020 play-by-play data records the same information as both the 2013 and 2017-18 data, but stores it in an entirely different format. Timestamps are in a different location on the table, point margins are recorded differently, and there are images of teams in the middle of the table. Once the differences in formatting have been accounted for, play-by-play analysis using the Mapper algorithm can be implemented.

One area of further analysis is the idea of placing weights on different factors. In the paper co-authored by Dean Oliver, different factors have different levels of impact on a player's Four Factors profile [3]. Based on analysis conducted in NBA games, effective field goal percentage was identified as the factor that has the greatest overall impact in a player's Four Factors profile (*i.e.*, the greatest weight), while free throw rate was identified as the factor that has the least overall impact in a player's Four Factors profile [3].

Another approach to take in future work is to alter parameters in different stages of the analysis. Analysis on the same 2017-18 data set with six-minute time intervals has begun, but the analysis has not yet gone in as great depth as the analysis using ten-minute time intervals. Analysis on two-man lineups based on their shared Four Factors profile, as opposed to individuals and their Four Factors profiles, and how they affect the team's plus/minus statistic is another, more informative approach to how the coaching staff can construct optimal lineup combinations. A Mapper graph can be generated for all possible two-man lineups to understand which two players appear to contribute positively and negatively to the team's plus/minus statistic from their shared Four Factors profile.

Some brief analysis on the play-by-play data showed that there were three common groupings of three play types, two of which were consistently prevalent in the data set. Further investigation would unveil what those play types are and what the coaching staff could do to improve their chances of winning games.

6 Acknowledgments

I would like to thank Towson University's Jess & Mildred Fisher College of Science & Mathematics Endowment and the Department of Mathematics for funding this project. I would also like to thank Towson University's Department of Mathematics for giving the opportunity to pursue a departmental honors degree. Finally, and most importantly, I would like to thank my advisor Dr. Christopher Cornwell for teaching and guiding me over the course of this research project. Without his mentorship, none of this would be possible.

References

[1] Ayasdi. (3 April 2012). *From 5 to 13: Redefining the Positions in Basketball.* [Video]. YouTube. https://www.youtube.com/watch?v=cdwP_sSaRCQ.

[2] Goldfarb, D. (2014). An Application of Topological Data Analysis to Hockey Analytics. *ArXiv, abs/1409.7635.*

[3] Kubatko, J., Oliver, D., Pelton, K., & Rosenbaum, D. T. (2007). A Starting Point for Analyzing Basketball Statistics. *Journal of Quantitative Analytics in Sports, 3*(3).

[4] Leonhardt, D. (2003). Pro Basketball; Mavericks' New Math May Be an Added Edge. *The New York Times.* Retrieved from https://www.nytimes.com/2003/04/27/sports/pro-basketball-mavericks-new-math-may-be-an-added-edge.html

[5] Li, L., Cheng, W. Y., Glicksberg, B. S., Gottesman, O., Tamler, R. Chen, R., ... & Dudley, J.T. (2015). Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science translational medicine, 7*(311), 311ra174.

[6] NetworkX Developers. (2020). *Overview of NetworkX.* NetworkX. https://networkx.github.io/documentation/stable/

[7] NumPy Developers. (2020). *NumPy Documentation.* NumPy. https://numpy.org/doc/

[8] Oliver, D. (2004). *Basketball On Paper: Rules and Tools for Performance Analysis.* Lincoln, NE: University of Nebraska Press

[9] The Pandas Development Team. (2020). *Pandas Documentation.* Pandas. https://pandas.pydata.org/pandas-docs/stable/

[10] Richardson, Leonard. (2020). *Beautiful Soup Documentation.* Crummy. https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[11] Scikit-Learn Developers. (2020). *sklearn.cluster.dbscan.* Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.dbscan.html

[12] Singh, G., Mémoli, F., & Carlsson, G. (2007). Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. *SPBG.* 91-100

[13] Thamel, P. (2013). Butler has found secret weapon in statistical guru Drew Cannon. *Sports Illustrated.* Retrieved from https://www.si.com/college-basketball/2013/03/20/drew-cannon-butler

[14] Towson University Athletics. (2020). *Men's Basketball Stats.* Towson University Athletics. https://towsontigers.com/sports/2014/8/5/MBB_0805140803.aspx?path=mbball