Arthur Sweetman

STA 402

Final Project Report

December 9, 2019

Introduction

How do NBA players perform? This is a huge piece of information that has a million different parts to it. Overall, a player is supposed to help a team win. Furthermore, better players tend to help their team win a larger number of games. Sometimes, however, it can appear that a player is helping their team, when, in fact, they are having no effect and/or hurting their team. Now, I did not find a way to find this out, but the theoretical applications one would need to find the answer to this question can be used in the same fashion for what I did in my project. I decided to take a look at a data set of every single shot taken in the 2014-2015 NBA regular season. This data set has massive amounts of information that can be uncovered to tell you things about a player one would have never guessed before. Every single shot taken in the 2014-2015 NBA season has a ton of data that goes along with it: who took the shot? Did it go in? How far away were they from the basket? Who was the closest defender? How far away was the closest defender when the shot was taken? These are just a few of the more than twenty variables that have been recorded in this data set along with the shot. The data came from data.world.

I took just a couple of these variables when answering my research question: “How can I present certain aspects of this dataset in a useful way so that information about these aspects can be clearly and richly conveyed and eventually spark future analyses on this data?” I had to start by assessing the dataset and seeing which variables contained rich enough information to call for further action.

Description of Data

This dataset contains every shot taken in the 2014-2015 NBA regular season (128,069 observations, i.e. “shots”) and over 20 variables associated with that shot. Most of these variables are numeric and a few of them are Strings:

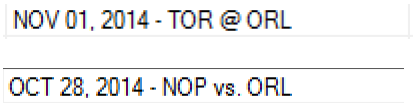
1. GAME_ID1 - int – unique numerical ID for each individual game
2. MATCHUP – string – shows teams playing each other and date of the matchup
3. LOCATION – string – says if the person who took this shot was Home or Away

4. W – string – indicates whether the person who took this shot was on the winning or losing team (W/L)
5. FINAL_MARGIN – int – shows the final margin of victory/defeat for the team of the player who took the shot
6. SHOT_NUMBER – int – the nth shot for that player in the game
7. PERIOD – int – which game period the game was in when the shot was taken
8. GAME_CLOCK – double – how much time was on the game clock when the shot was taken
9. SHOT_CLOCK – double – how much time was on the shot clock when the shot was taken
10. DRIBBLES – int – how many dribbles the player took before taking the shot
11. TOUCH_TIME – double – how long the player had the ball before taking the shot
12. SHOT_DIST – double – how far the player was from the basket when taking the shot
13. PTS_TYPE – int – says if the shot was a 2-point or 3-point attempt
14. SHOT_RESULT – string – indicates whether the shot was made or missed (“made” or “missed”)
15. CLOSEST_DEFENDER – string – the name of the defensive player who was closest the shooter when the shot was taken
16. CLOSEST_DEFENDER_PLAYER_ID – int – the numerical ID of the defensive player who was closest the shooter when the shot was taken
17. CLOSE_DEF_DIST – double – how far away the closest defender was from the shooter when the shot was taken
18. FGM – indicator – indicates whether the shot was made or missed (1 or 0)
19. PTS – int - how many points were awarded as a result of the shot (0, 2, or 3)
20. player_name – string – name of the player who took the shot
21. player_id – int – numerical ID of the player who took the shot

All these variables could have some use in a data analysis, so I did not remove any variables. I did, however, add certain temporary variables in the program “graphic.sas” and “shortthrees.sas” as well as certain permanent variables in the program “homeandaway.sas”. In “homeandaway.sas”, I added the variables, “home”, “away”, “team”, and “date”. Two of these variables, team and date, were necessary to add for use in the input of the macro variable “plotshots20142015nba” created in the program “grapic.sas”. The home and away variables were created for easy future access to the home and away teams in that game. Furthermore, the temporary variables created in “graphic.sas” were mostly used for assistance in plotting the points in the first graphical display. Variables such as “radius” and “theta” were used to assign polar coordinates to the data points, and “madex”, “madey”, “missedx”, and “missedy” were created to convert the polar coordinates back into cartesian coordinates so they

could be plotted on the basketball court, which is an x-y plane. An indicator variable was also created to randomly assign a location (left or right side of the 3-point line) for shots that were taken in “corners” of the 3-point line (the area where the 3-point line is a straight line instead of curved). Lastly, a “range” variable was created to classify a shot as being short-range, mid-range, or long-range. This was used in the FREQ plot. In the program “shortthrees.sas”, an indicator variable called “short3” was created to indicate whether a shot was classified as a three-point shot in the data set, but was shorter than 22 feet from the basket (more on this in the “results” section).

The most difficult variable to work with was a String variable that was called “matchup”, and it included the date of the shot, and the two different teams that were playing each other for which the shot was taken in. One of the things someone can do with this data set is to analyze home-court advantage as well as cross-referencing information in this data set with another data set. Therefore, in order to prepare the data for something like this, it was necessary to separate the date, home team, and away team and put them into different variables. This became a little more complicated, however, when there turned out to be two different versions of the String, for example:



As you can see, one of the versions uses a “@” to separate the two teams while the other string uses a “vs” to separate the teams. This made it difficult to classify which team was home and which was away since one version implies the second team is home while the other version implies the first team is home.

Second, there were certain variables not in the data set that would have been convenient, although not particularly necessary, to have. One of these variables would have been coordinates of where the shot was taken on the court along with the distance from the basket rather than simply the distance from the basket. This would have been useful since it would have enabled someone to visualize where a certain player tends to shoot the ball, along with where that player shoots the ball with a higher success rate and where they shoot the ball with a lower success rate.

Strategy Employed

The strategy I employed in solving the problems above was to first separate the “matchup” variable into four separate variables: date, home team, and away team, and that team the player who took the shot was on. It was relatively simple to separate the String into its appropriate variables. I first had to split all the words and store them in an array. From there, I was able to use a simple IF/ELSE statement to dictate whether one

team was home or one team was away. After that, in order to find out which team the player was on, I used another IF/ELSE statement on the “location” variable to decide whether the player who took the shot was on the home team or the away team.

After I separated this variable, I decided to create certain graphics where one could visually see how far away a shot was taken from the basket and whether the shot was made or missed. I also wanted the user to be able to filter which data they wanted to see. I first had to figure out which resources from SAS I wanted to use to draw the basketball court and proceed to plot the points on the court signifying the distance away from the hoop the shot was taken. Like I explained before, this data set did not include exact coordinates of where the shot was taken, only the distance from which it was taken. Therefore, the exact positions of the shots that would end up in the graphic were not representative of the exact positions on the court in which the shot was taken, it only represents the distance from which the shot was taken.

I used the SG Annotation resource to draw the basketball court and plot the points on top of it. I first created a dataset for the basketball court annotation, called “courtanno.txt” (Figure 1.7). With this, I was able to draw the entire basketball court except for the curved portion of the three-point arc. Furthermore, in order to draw the two 3-point arcs, I had to use a do loop to create thousands of statements which each created a miniature line which was drawn according to the appropriate equation of the circular portions of each of the arcs (refer to “threepointarcs” DATA step in Code 1.2.1).

The next portion of this task was to proceed to plot each shot in the data set onto this court that I just drew. In order to usefully show the distances of the shots on the drawn court, I had to convert all the points into polar coordinates where the radius is the shot distance and the angle (“theta”) is randomly selected using a random-value generator. Once I did this, several small conflicts came in to play:

1. When a shot was taken from far away, how do I make sure the random-angle generator makes sure the shot gets plotted in a spot on the court rather than off the court?
2. Since there are areas around the three-point line that are closer than other points (i.e. in the corners), how do I make sure these three-point shots are not misrepresented as two-point shots in a different location on the court?

Both of these problems were able to be solved using trigonometry and involving the distance of the shot in the determinant for the boundaries of the random-angle generator (Code 1.2.3).

In addition to the basketball court display (Figures 1.1 and 1.4), I added a detailed histogram showing the distribution of the distances of shots (filtered by player, team, and/or game date) (Figures 1.2 and 1.5), as well as a frequency table which put

numerical values and proportions to the output of the data. The frequency table compares the frequency of made/missed shots to the distance of the shot (summed into “short”, “mid”, and “long” range) (Figures 1.3 and 1.6).

The last task was to “macro-ize” the entire program and make it so that the user can run the program quickly and easily as well as choose what data they want shown in the output.

Results

As I was creating this graphic and taking care of the problem that resulted in short three-pointers being misrepresented as two-pointers, I started to see a concerning trait of this specific data set. There is a significant number of observations where the shot distance was shorter than the closest point of the three-point line and the shot was said to be a three-pointer. This is technically impossible. Because of this feature, I wanted to know how often a particular shot like this showed up in the data. I ran PROC FREQ to count the number of times a shot was counted as a three-pointer and was taken less than 22 feet away from the basket compared to the total number of three-pointers taken (Figure 1.8 shows this output). The results showed that out of the 33,896 total three-pointers attempted in the 2014-2015 NBA season, a whopping 872 of those were taken less than 22 feet away from the basket according to this data set (2.57 percent of all three-pointers attempted). One could speculate why this could be. One reason could be human error when calling shots attempted close to the three-point line. Since referees are the ones who decide if a shot should count for three points or not, this could be a result of human error. However, the argument against this would be the fact that several of these shots in the data are not simply barely within 22 feet. Rather, these several are from very close to the basket; that is, less than 10 feet away. This fact suggests something else coming into play. The worst scenario is this data set being faulty in some way, where the data in here is not all 100 percent accurate.

In the output of this program, we can see a clear representation of the distances of shots on a basketball court, and whether or not they went in or not. To further the understanding of that display, a histogram clearly puts a visual to the density/frequencies of shots taken at any distance. Lastly, a frequency table provides information that the first two displays do not show, clear numeric values for the frequency those shot distances. In other words, this frequency table shows shot statistics on a particular player, team, and/or date of a game.

Discussion

The guiding research question of this project was, “How can I present certain aspects of this dataset in a useful way so that information about these aspects can be clearly and richly conveyed and eventually spark future analyses on this data?” In the output of this program, we see aspects of this dataset being presented in a clear and concise way, so there is a lot of well-rounded information being shown with only a few tables/charts. The last program, “shortthrees.sas” outputs one extra table; however, there is a clear use for the program “graphic.sas”, and “shortthrees.sas” does not belong to that same purpose. The output of “shortthrees.sas” shows us the frequency of three-point shots that were taken less than 22 feet from the basket, as it was explained in the results section above. The output of “shortthrees.sas” immediately provides a cause for further investigation and analysis, while the output of “graphic.sas” provides a tool for investigation of NBA players, teams, and games. This program can help people see trends in this data set that one would not have been able to see before. These observations seen after using this program could potentially lead to further investigation and analysis on either this particular data set or, in general, an NBA player, team, or game.

Code 1.1 **Code**

```
/* homeandaway.sas

Author:      Arthur Sweetman
Directory:   M:\STA 402\final-project
Purpose:     This file imports the data set shot_logs.csv and proceeds
             to record the date, home team, and away team based off the
             String varible "matchup".

*/

libname proj "M:\STA 402\final-project";

proc import datafile="M:\STA 402\final-project\shot_logs.csv"
      out=proj.shotlogs
      dbms=csv
      ;
run;

proc sort data=proj.shotlogs;
      by game_id1;
run;

data proj.shotlogs_splitted;

      set proj.shotlogs;
      format date mmddyy9.;
      date=input(matchup, anydtdte12.);

      array word[6] $;
      drop i;
      do i = 1 to 6;
            word[i] = scan(matchup, i);
      end;

      if word5 = "@" then
            do;
                  home=word6;
                  away=word4;
            end;
      else
            do;
                  home=word4;
                  away=word6;
            end;
      drop word1-word6;

      if location="H" then
            team = home;
      else
            team = away;

run;
```

Code 1.2.1

```
/* graphic.sas

Author:     Arthur Sweetman
Directory:  M:\STA 402\final-project
Purpose:    Create useful graphics illustrating the distribution
            of shot distances in the 2014-2015 NBA season (can be
            filtered by player name, team name, and date of the game).
*/

%macro plotshots20142015nba (playername=, teamname= , gamedate=);

libname proj "M:\STA 402\final-project";

/* This data step is used to draw the basketball court
      (not including the curved 3-point lines) */
data courtlines;
      infile "M:\STA 402\final-project\courtanno.txt" expandtabs pad;
      retain drawspace "datavalue" linecolor "black";
      input function $9. x1 y1 x2 y2 height width anchor $10.;
run;

* This data step is only used to draw each of the curved 3-point arcs;
data threepointarcs;
      retain drawspace "datavalue" linecolor "black";

      do;
            function="polyline"; x1=3; y1=14; x2=.; y2=.;
                  * starting point of lower 3-point line;
                  height=.; width=.; anchor="";
            output;
      end;
      do x1 = 3 to 47 by .01;
            function = "polycont";
            y1 = sqrt(564.0625-((x1-25)**2))+4.75;
                  * equation of lower 3-point line;
            x2=.; y2=.; height=.; width=.; anchor="";
            output;
      end;

      do;
            function="polyline"; x1=3; y1=80; x2=.; y2=.;
                  * starting point of upper 3-point line;
                  height=.; width=.; anchor="";
            output;
      end;
      do x1 = 3 to 47 by .01;
            function = "polycont";
            y1 = -sqrt(564.0625-((x1-25)**2))+89.25;
                  * equation of upper 3-point line;
            x2=.; y2=.; height=.; width=.; anchor="";
            output;
      end;

run;
```

Code 1.2.2

```
/* Combine the above data sets into one so the entire
      court annotation is in one data set */
data drawcourt;
      set courtlines threepointarcs;
run;

/* Set shot distances in terms of polar coordinates to use for plotting.
      This DATA step also filters the output based on the user input. */
data shotlogs_polar;
      set proj.shotlogs_splitted;

      /* In order to properly filter by date, we have to convert the input date
            into a SAS date format */
      %if &gamedate^= %then %do; %let fdate = input("&gamedate", anydtdte12.); %end;

      /* In this string of %IF/%ELSE %IF statements, we filter the dataset based on
            which inputs the user put in the macro */
      %if &playername^= & &teamname^= & &gamedate^= %then
            %do;
                  where player_name="&playername" and team="&teamname" and date=&fdate;
                  title "&playername's Shot Distribution on &gamedate";
            %end;
      %else %if &playername^= & &teamname^= %then
            %do;
                  where player_name="&playername" and team="&teamname";
                  title "&playername's Shot Distribution";
            %end;
      %else %if &playername^= & &gamedate^= %then
            %do;
                  where player_name="&playername" and date=&fdate;
                  title "&playername's Shot Distribution on &gamedate";
            %end;
      %else %if &teamname^= & &gamedate^= %then
            %do;
                  where team="&teamname" and date=&fdate;
                  title "&teamname's Shot Distribution on &gamedate";
            %end;

      %else %if &playername^= %then
            %do;
                  where player_name="&playername";
                  title "&playername's Shot Distribution";
            %end;
      %else %if &teamname^= %then
            %do;
                  where team="&teamname";
                  title "&teamname's Shot Distribution";
            %end;
      %else %if &gamedate^= %then
            %do;
                  where date=&fdate;
                  title "Shot Distribution of all players on &gamedate";
            %end;
```

Code 1.2.3

```
      /* Make sure all shots are plotted on the basketball court
            and can be seen in the graphic */
      radius = shot_dist;
      if radius > 25 then
            theta = rand('uniform', arcos(25/radius), 3.14-arcos(25/radius));
      else
            theta = rand('uniform', 0, 3.14);

      * Make sure shots taken from the corner 3-point line are plotted appropriately;
      if radius < 23.75 & radius >= 22 & pts_type = 3 then
            do;
                  indicator = rand('bernoulli', .5);
                  if indicator = 1 then
                        theta = rand('uniform', 0, arcos(22/radius));
                  else
                        theta = rand('uniform', 3.14-arcos(22/radius), 3.14);
            end;

      * Assign made and missed shots to different variables for plotting;
      if shot_result = "made" then
            do;
                  madex = radius*cos(theta)+25;
                  madey = radius*sin(theta)+4.75;
            end;
      else
            do;
                  missedx = radius*cos(theta)+25;
                  missedy = radius*sin(theta)+4.75;
            end;



      * Assign a shot as "close", "mid", or "long" range;
      if shot_dist < 12 then
            range = "Short";
      else if shot_dist < 22 then
            range = "Mid";
      else
            range = "Long";

      /* Assign labels to variables displayed in the output */
      label range="Distance" shot_result="Shot Result";

run;
```

Code 1.2.4

```
/*****************************Plot data*******************************/

/* This plot uses the annotation dataset "drawcourt" to draw the basketball
      court and then plot the shots onto the court */
proc sgplot data=shotlogs_polar sganno=drawcourt aspect=2 noborder;
      xaxis min=0 max=50 display=none;
      yaxis min=0 max=94 display=none;
      scatter x=madex y=madey /
            transparency=0 markerattrs=(color=blue size=3)
            legendlabel="Made";
      scatter x=missedx y=missedy /
            transparency=0 markerattrs=(symbol=X color=red size=3)
            legendlabel="Missed";
run;

/* This plot outputs a histogram of all shots (filtered) according
      to their distance away from the basket. This is a visual to complement
      the above plot */
proc sgplot data=shotlogs_polar;
      histogram shot_dist / binwidth=1 boundary=lower;
      density shot_dist / type=kernel;
      refline 12 / axis=x label="Mid-Range";
      refline 22 / axis=x label="Long-Range";
      xaxis label="Shot Distance" values=(0 to 50 by 2);
      yaxis label="Shot Frequency (Percent)";
run;

/* This outputs a frequncy plot that helps the user see numerical
      values and proportions which correlate to the output of the
      above plots */
proc freq data=shotlogs_polar;
      table range*shot_result;
run;

%mend plotshots20142015nba;

/**************
DUE TO RESTRICTIONS IN THE DATA SET,
NAMES ARE CUT OFF AFTER 13 CHARACTERS,
ONLY TYPE A PLAYER'S NAME UP TO THEIR 13TH CHARACTER,
ALL LOWER CASE.
teamname MUST be a proper 3-letter uppercase abbreviation for an NBA team
gamedate MUST be between 10/28/14 - 03/04/15 and in the format mm/dd/yy
*/
%let rtfoutput = M:\STA 402\final-project\shotgraphic.rtf;
ods rtf bodytitle file="&rtfoutput";

%plotshots20142015nba(
playername=klay thompson,
/**********1234567890123*/
teamname=GSW,
gamedate=12/25/14
);

ods rtf close;
```

Here is the output of this rendition of the macro variable:

Figure 1.1

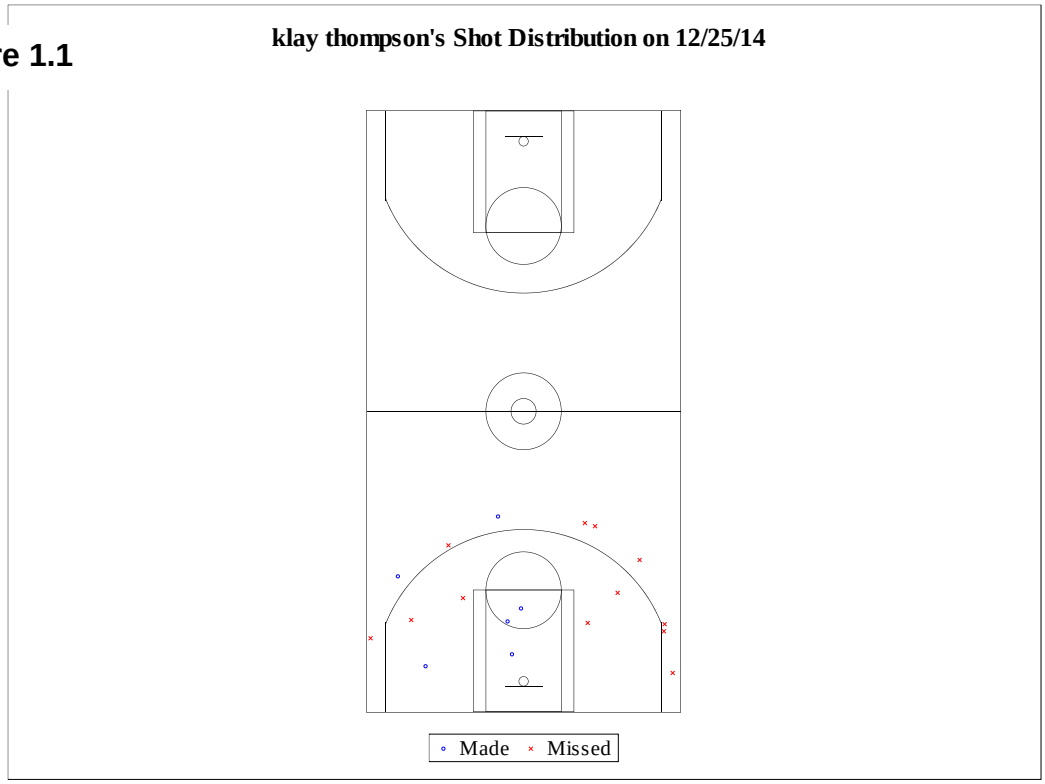


Figure 1.2

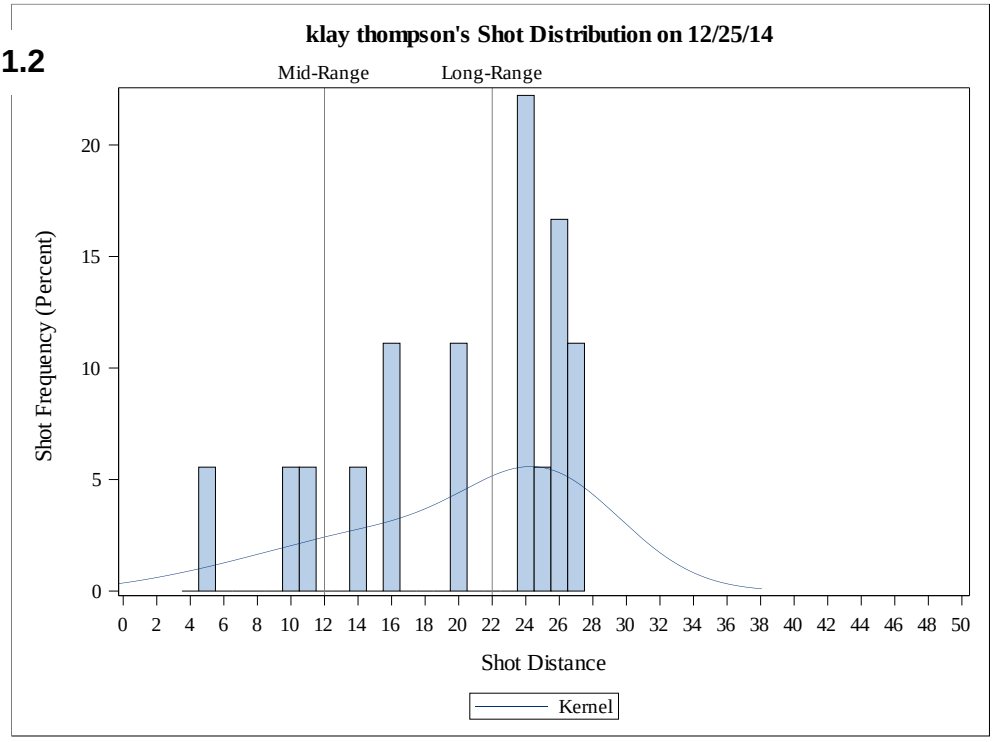


Figure 1.3 ***klay thompson's Shot Distribution on 12/25/14***

The FREQ Procedure

Table of range by SHOT_RESULT			
range(Distance)	**SHOT_RESULT(Shot Result)**		
Frequency Percent Row Pct Col Pct	**made**	**missed**	**Total**
Long	2 11.11 20.00 33.33	8 44.44 80.00 66.67	10 55.56
Mid	1 5.56 20.00 16.67	4 22.22 80.00 33.33	5 27.78
Short	3 16.67 100.00 50.00	0 0.00 0.00 0.00	3 16.67
Total	6 33.33	12 66.67	18 100.00

Here is a different rendition of the macro variable. This rendition shows Stephen Curry's season stats for the 2014-2015 NBA season:

Code 1.3

```
%plotshots20142015nba(
playername=stephen curry,
/**********1234567890123*/
teamname=,
gamedate=
);
```

Figure 1.4

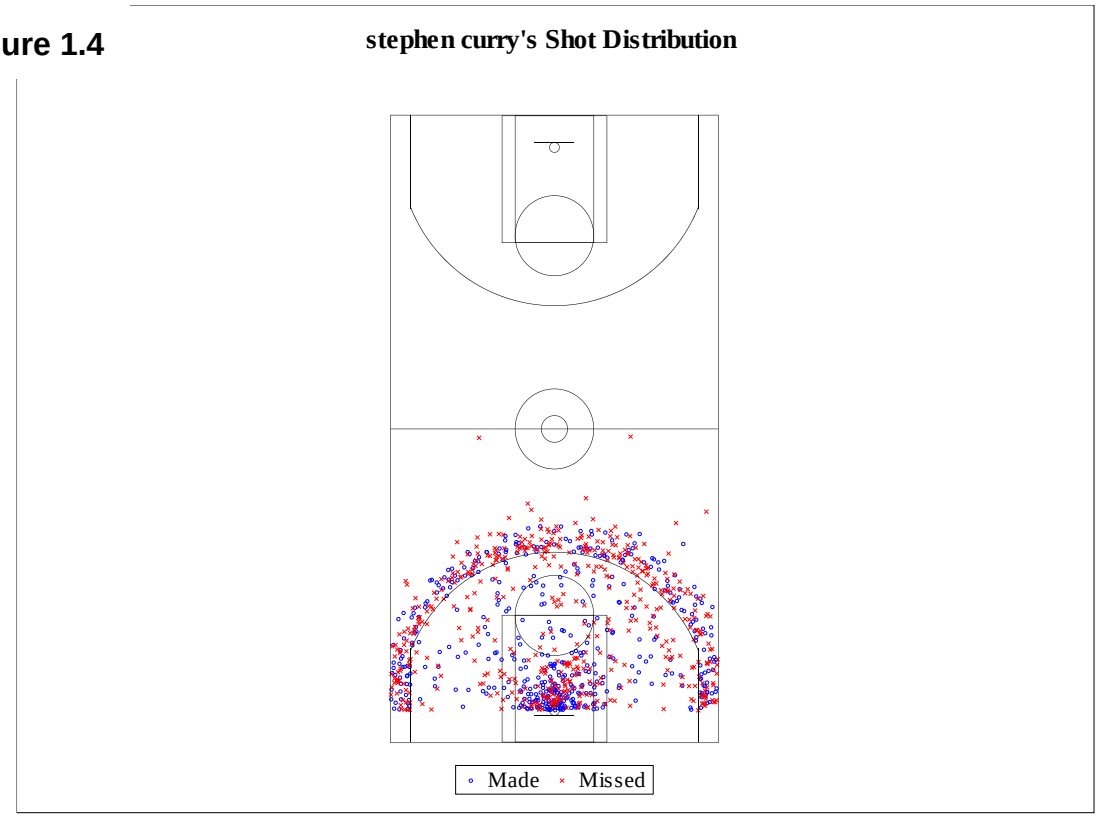


Figure 1.5

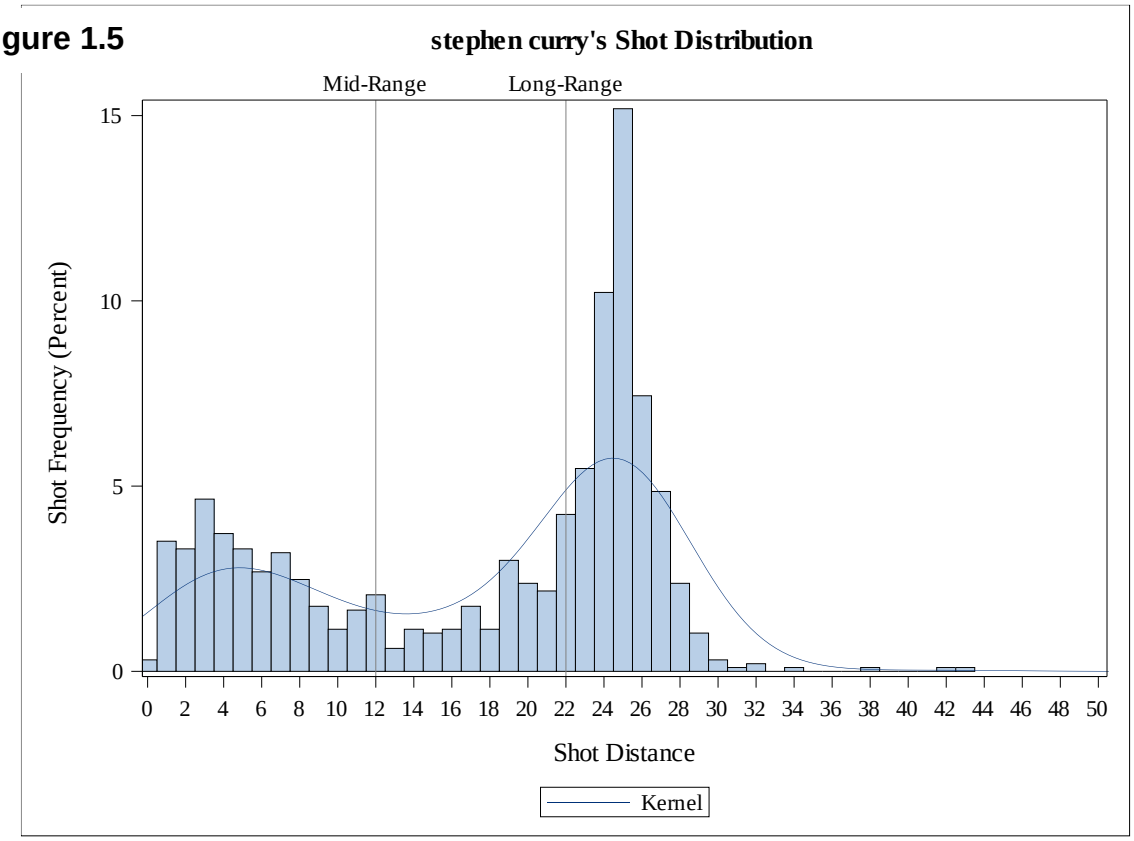


Figure 1.6 *stephen curry's Shot Distribution*

The FREQ Procedure

Table of range by SHOT_RESULT			
range(Distance)	**SHOT_RESULT(Shot Result)**		
Frequency Percent Row Pct Col Pct	**made**	**missed**	**Total**
Long	196 20.25 40.25 41.70	291 30.06 59.75 58.43	487 50.31
Mid	82 8.47 50.00 17.45	82 8.47 50.00 16.47	164 16.94
Short	192 19.83 60.57 40.85	125 12.91 39.43 25.10	317 32.75
Total	470 48.55	498 51.45	968 100.00

This is the file “courtanno.txt”, found in the directory “M:\STA 402\final-project”.
Used in the first DATA step (“courtlines”) in the “plotshots20142015nba” macro variable.

Figure 1.7

```
rectangle 25 47 . . 94 100 .
line      0 47 50 47 . . .
oval      25 47 . . 4 8 .
oval      25 47 . . 12 24 .
line      3 0 3 14 . . .
line      47 0 47 14 . . .
line      3 94 3 80 . . .
line      47 94 47 80 . . .
line      22 4 28 4 . . .
line      22 90 28 90 . . .
oval      25 4.75 . . 1.5 3 .
oval      25 89.25 . . 1.5 3 .
rectangle 17 0 . . 19 32 bottomleft
rectangle 17 94 . . 19 32 topleft
rectangle 19 0 . . 19 24 bottomleft
rectangle 19 94 . . 19 24 topleft
oval      25 19 . . 12 24 .
oval      25 76 . . 12 24 .
```

Code 1.4 **More Code**

```
/* shortthrees.sas

Author:     Arthur Sweetman
Directory:  M:\STA 402\final-project
Purpose:    Investigate the proportion of shots that were closer than
            22 feet away and were counted as 3-pointers
*/
libname proj "M:\STA 402\final-project";
data shortthrees;
      set proj.shotlogs_splitted;
      if pts_type = 3 & shot_dist < 22 then
            short3 = 1;
      else short3 = 0;
run;

title 'Proportion of "Short" Threes Compared to all Threes Attempted';
ods rtf bodytitle file="M:\STA 402\final-project\shortthrees.rtf";
proc freq data=shortthrees;
      table short3*pts_type;
run;
ods rtf close;
title;
```

Figure 1.8

Proportion of "Short" Threes Compared to all Threes Attempted

The FREQ Procedure

Table of short3 by PTS_TYPE			
short3	**PTS_TYPE**		
Frequency Percent Row Pct Col Pct	**2**	**3**	**Total**
0	94173 73.53 74.04 100.00	33024 25.79 25.96 97.43	127197 99.32
1	0 0.00 0.00 0.00	872 0.68 100.00 2.57	872 0.68
Total	94173 73.53	33896 26.47	128069 100.00

I have manually highlighted the key number in this output. This 2.57 means that approximately 2.57% (872/33896) of three-point attempts in this data set are also said to be less than 22 feet away according to this data set.