# Basketball shot value & visualization

*Michael Lopez, Skidmore College*

### Overview

In this lab, we'll gain experience implementing logistic regression to estimate the probability of successful NBA shots. We'll also link to themes from our football unit - such as expected points added - and increase our visual literacy by sampling some of the `ggplot()` package in R. As one additional tool, I'll walk through a few examples of how we can **clean** what is originally a messy data set.

In R, there are a few ways to get data from the Internet. I use the `RCurl` package. Per usual, start by installing the package. You only have to do this once.

```
library(RCurl)
library(mosaic)
```

Next, we'll load this as the `url` using the `getURL` command, and load it into R.

```
url <- getURL("https://raw.githubusercontent.com/JunWorks/NBAstat/master/shot.csv")
nba.shot <- read.csv(text = url)
```

## Data Cleaning

The `nba.shot` data contain roughly 200,000 shots from the 2014-2015 season. This is awesome. However, much like real life, things are never as easy as they seem. Let's start by summarizing our data:

```
head(nba.shot)
summary(nba.shot)
```

1. What does each row refer to, and what are each of the columns?
2. Identify some issues that you see in the data set by looking at the output of the `summary()` command. For example, look for missing values or measurements that, from a basketball perspective, don't make sense.

Interestingly, a handful of shots are missing shot-clock information. There could be several explanations for this - an error in the data collection process, a broken shot clock, etc - and if we had more time, it may be worth exploring why this information is missing.

In the meantime, there are several ways of dealing with missing data - a whole book, in fact - but for today's purposes, we will make some assumptions and drop any rows with missing data. This will make our eventual analysis much easier.

```
nba.shot <- na.omit(nba.shot)
```

Note that dropping missing rows in this data set is more reasonable, given that the number of rows that we dropped accounts for less than 5% of the overall data.

There are some other issues that we will want to look at. The variable `PTS` contains the number of points that each shot was worth, and the variables `PTS_TYPE` and `SHOT_DIST` indicate the type of shot and distance.

```
tally(~PTS, data = nba.shot)
tally(PTS_TYPE ~ SHOT_DIST >= 22, data = nba.shot)
```

In the first table, we note that certain shots were counted as 4 or 6 points. That's not good, given that our data set contains only field goals, and not free throws (1 point) or touchdowns (wrong sport).

In the second table, there are roughly 1400 shots which are listed as three pointers, even though they were they were taken from a distnace of less than 22 feet. Given that the three-point line is at least 22 feet from the basket, it does not make sense to use these observations.

Here's one way of getting rid of the funny rows.

```
nrow(nba.shot)
nba.shot <- filter(nba.shot, PTS <4, SHOT_DIST>=22 |PTS_TYPE==2)
nrow(nba.shot)
```

2. How many rows were dropped using the above filtering?

## Expected points

All else being equal, what's the most efficient shot in the NBA?

Let's start by comparing the success rates of two-point shots to three-point shots.

```
tally(SHOT_RESULT ~ PTS_TYPE, data = nba.shot, format = "proportion")
```

3. Identify the expected point totals from all two-point shots and three-point shots in the 2014-15 season. Which one was preferred?

Let's look at certain players.

```
tally(SHOT_RESULT ~ PTS_TYPE,
      data = filter(nba.shot, playerName=="Stephen Curry"), format = "proportion")
tally(SHOT_RESULT ~ PTS_TYPE,
      data = filter(nba.shot, playerName=="Kevin Garnett"), format = "proportion")
```

4. For Curry and Garnett, calculate their expected point totals on two and three-point shots. What does that suggest about their optimal choices?

## Logistic Regression

Logistic regression will be another useful tool to (i)identify impacts of shooting success and (ii) allow us to judge which players have outperformed or underperformed expectations.

Here's one model.

```
fit.1 <- glm(SHOT_RESULT == "made" ~ SHOT_DIST + TOUCH_TIME +
             DRIBBLES + SHOT_CLOCK + CLOSE_DEF_DIST,
              data = nba.shot, family = "binomial")
```

5. Estimate the increased odds of a made shot taken with 1 more second left on the shot clock. Then, estimate the increased odds of a made shot with 10 more seconds on the shot clock.

We use the following to get each shot's expected points (given `fit.1()`), as well as the expected points added (`epa`) given the shot result.

```
nba.shot <- nba.shot %>%
  mutate(predicted.probs = fitted(fit.1),
  expected.pts = predicted.probs * PTS_TYPE,
         epa = PTS - expected.pts)
```

6. Look at the first row of the data set. Where do the `predicted.probs` (0.489), `expected.pts` (0.978), and `epa` (1.02) come from?

It's also possible to look at individual shots based on their expected points. For example, here are the six of the most difficult shots

```
head(arrange(nba.shot, expected.pts))
```

Interestingly, what do we notice about the data set's most difficult shot? It went in!

Here's a video (link).

Alternatively, here are the six shots worth the highest expected point totals.

```
head(arrange(nba.shot, -expected.pts))
```

In each of these examples, the shot is a three-pointer with more than an 80% chance of going in.

7. Looking at six *easy* shots above - can you tell where and why our logistic regression model went wrong?

## Shooter valuation using expected points added

We use a similar procedure to the one we developed with field goal kickers to estimate the cumulative expected points added from NBA shooters.

```
shot.group <- nba.shot %>%
  group_by(playerName) %>%
  summarise(total.epa = sum(epa), n.shots = length(epa)) %>%
  arrange(total.epa)
head(shot.group)
tail(shot.group)
```

For those of us who are basketball fans, these names match our expectations. Stephen Curry was worth nearly 300 points alone, relative to expectation, on his outstanding shooting.

### Visualizing expected points

Let's graph the metrics we just calculated. In this example, we'll make our first `ggplot()` graph of the semester (*Note*: `ggplot()` is its own package, but it also comes along when we load `mosaic`.

```
xyplot(total.epa ~ n.shots, data = shot.group)
p<- ggplot(shot.group, aes(n.shots, total.epa, label = playerName))
p + geom_text() + scale_y_continuous("Expected Points Added") +
  scale_x_continuous("Shot attempts") +
  ggtitle("Expected points added ~ number of shots, 2014-15 season") +
  theme_bw()

p + geom_point() + scale_y_continuous("Expected Points Added") +
  scale_x_continuous("Shot attempts") +
  ggtitle("Expected points added ~ number of shots, 2014-15 season") +
  theme_bw()
```

8. Describe the distribution of expected points added as a function of shot attempts. Why does the distribution fan out? Why does the distribution fan out on only one side?

## Final thought questions

9. Our model of shot probabilities is probably missing some other variables that effect success rates. What ones can you think of?

10. Returning to the issue from question (7). How does Krishna deal with this problem here?