Week 6.2 - Using Summary Statistics to Examine the Hot Hand

Using Summary Statistics to Examine the "Hot Hand"

Import useful libraries and the updated shot log data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
Shotlog = read.csv("~/Google Drive/Sports Analytics Moocs/MOOC 1 - Foundations of sports analytics/Week
Player_Stats = read.csv("~/Google Drive/Sports Analytics Moocs/MOOC 1 - Foundations of sports analytics
Player_Shots = read.csv("~/Google Drive/Sports Analytics Moocs/MOOC 1 - Foundations of sports analytics
Player_Game = read.csv("~/Google Drive/Sports Analytics Moocs/MOOC 1 - Foundations of sports analytics/
head(Shotlog)
```

```
##   team_previous_shot player_position home_game location_x
## 1                                 SF       Yes         97
## 2             MISSED              SF       Yes        279
## 3             MISSED              SF       Yes         58
## 4             SCORED              SF       Yes        691
## 5             MISSED              SF       Yes        691
## 6             MISSED              SF       Yes        679
##   opponent_previous_shot home_team           shot_type points away_team
## 1                 SCORED       ATL    Pullup Jump Shot      2       WAS
## 2                 SCORED       ATL           Jump Shot      3       WAS
## 3                 SCORED       ATL  Cutting Layup Shot      2       WAS
## 4                 MISSED       ATL    Pullup Jump Shot      3       WAS
## 5                 MISSED       ATL    Pullup Jump Shot      2       WAS
## 6                 MISSED       ATL Step Back Jump Shot      3       WAS
##   location_y                      time       date  shoot_player
## 1        405 0 days 00:01:09.000000000 2016-10-27 Kent Bazemore
## 2        130 0 days 00:03:11.000000000 2016-10-27 Kent Bazemore
## 3        275 0 days 00:09:53.000000000 2016-10-27 Kent Bazemore
## 4        100 0 days 00:04:50.000000000 2016-10-27 Kent Bazemore
## 5        181 0 days 00:06:29.000000000 2016-10-27 Kent Bazemore
## 6        109 0 days 00:07:46.000000000 2016-10-27 Kent Bazemore
##   time_from_last_shot quarter current_shot_outcome current_shot_hit
## 1                  NA       1               MISSED                0
## 2                   4       1               MISSED                0
## 3                  30       2               MISSED                0
## 4                  39       3               SCORED                1
## 5                  20       3               MISSED                0
## 6                  21       3               MISSED                0
##   lag_shot_hit average_hit shot_count shot_per_game
## 1            0   0.4085873        722             7
## 2            0   0.4085873        722             7
## 3            0   0.4085873        722             7
## 4            0   0.4085873        722             7
## 5            1   0.4085873        722             7
## 6            0   0.4085873        722             7
```

Conditional Probability

We can first calculate the conditional probability of making a shot in the current period conditional on making the previous shot.

$$Conditional\ Probability = \frac{Probability\ of\ Making\ Consecutive\ Shots}{Probability\ of\ Making\ Previous\ Shot}$$

We will need to create a variable that indicates a player made consecutive shots.

```r
Shotlog$conse_shot_hit = ifelse(Shotlog$current_shot_hit == 1 &
                                  Shotlog$lag_shot_hit == 1, 1, 0)
head(Shotlog)
```

```
##   team_previous_shot player_position home_game location_x
## 1                                 SF       Yes         97
## 2             MISSED              SF       Yes        279
## 3             MISSED              SF       Yes         58
## 4             SCORED              SF       Yes        691
## 5             MISSED              SF       Yes        691
## 6             MISSED              SF       Yes        679
##   opponent_previous_shot home_team           shot_type points away_team
## 1                 SCORED       ATL    Pullup Jump Shot      2       WAS
## 2                 SCORED       ATL           Jump Shot      3       WAS
## 3                 SCORED       ATL  Cutting Layup Shot      2       WAS
## 4                 MISSED       ATL    Pullup Jump Shot      3       WAS
## 5                 MISSED       ATL    Pullup Jump Shot      2       WAS
## 6                 MISSED       ATL Step Back Jump Shot      3       WAS
##   location_y                      time       date  shoot_player
## 1        405 0 days 00:01:09.000000000 2016-10-27 Kent Bazemore
## 2        130 0 days 00:03:11.000000000 2016-10-27 Kent Bazemore
## 3        275 0 days 00:09:53.000000000 2016-10-27 Kent Bazemore
## 4        100 0 days 00:04:50.000000000 2016-10-27 Kent Bazemore
## 5        181 0 days 00:06:29.000000000 2016-10-27 Kent Bazemore
## 6        109 0 days 00:07:46.000000000 2016-10-27 Kent Bazemore
##   time_from_last_shot quarter current_shot_outcome current_shot_hit
## 1                  NA       1                MISSED              0
## 2                   4       1                MISSED              0
## 3                  30       2                MISSED              0
## 4                  39       3                SCORED              1
## 5                  20       3                MISSED              0
## 6                  21       3                MISSED              0
##   lag_shot_hit average_hit shot_count shot_per_game conse_shot_hit
## 1            0   0.4085873        722             7              0
## 2            0   0.4085873        722             7              0
## 3            0   0.4085873        722             7              0
## 4            0   0.4085873        722             7              0
## 5            1   0.4085873        722             7              0
## 6            0   0.4085873        722             7              0
```

We can create a player level dataframe. The average of the variable "conse_shot_hit" would be the joint probability of making current and previous shots. We will also calculate the average of "lag_shot_hit" to indicate the probability of making the previous shot.

```
Player_Prob = Shotlog %>% group_by(shoot_player) %>%
  summarise(conse_shot_hit = mean(conse_shot_hit),
            average_lag_hit = mean(lag_shot_hit)) %>%
  ungroup()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(Player_Prob)
```

```
## # A tibble: 6 x 3
##   shoot_player  conse_shot_hit average_lag_hit
##   <chr>                  <dbl>           <dbl>
## 1 A.J. Hammons           0.185           0.407
## 2 Aaron Brooks           0.181           0.416
## 3 Aaron Gordon           0.213           0.458
## 4 Adreian Payne          0.154           0.410
## 5 Al Horford             0.216           0.471
## 6 Al Jefferson           0.217           0.507
```

Calculate conditional probability for each player

We can calculate the conditional probability by dividing the joint probability by the probability of making the previous shot.

```
Player_Prob$conditional_prob = Player_Prob$conse_shot_hit/Player_Prob$average_lag_hit
head(Player_Prob)
```

```
## # A tibble: 6 x 4
##   shoot_player  conse_shot_hit average_lag_hit conditional_prob
##   <chr>                  <dbl>           <dbl>            <dbl>
## 1 A.J. Hammons           0.185           0.407            0.455
## 2 Aaron Brooks           0.181           0.416            0.434
## 3 Aaron Gordon           0.213           0.458            0.465
## 4 Adreian Payne          0.154           0.410            0.375
## 5 Al Horford             0.216           0.471            0.458
## 6 Al Jefferson           0.217           0.507            0.427
```

We can merge the "Player_Prob" data frame with the "Player_Stats" data frame we created earlier to compare the conditional probability and the unconditional probability. If the two probabilities are the same, or almost the same, then we fail to find evidence that the making the current shot depends on making the previous shot.

```
Player_Stats = left_join(Player_Prob, Player_Stats, by= 'shoot_player')
head(Player_Stats, 10)
```

```
## # A tibble: 10 x 5
##    shoot_player    conse_shot_hit average_lag_hit conditional_prob average_hit
##    <chr>                    <dbl>           <dbl>            <dbl>       <dbl>
##  1 A.J. Hammons             0.185           0.407            0.455       0.405
##  2 Aaron Brooks             0.181           0.416            0.434       0.403
##  3 Aaron Gordon             0.213           0.458            0.465       0.455
##  4 Adreian Payne            0.154           0.410            0.375       0.426
##  5 Al Horford               0.216           0.471            0.458       0.473
##  6 Al Jefferson             0.217           0.507            0.427       0.499
##  7 Al-Farouq Aminu          0.160           0.390            0.411       0.393
##  8 Alan Anderson            0.115           0.385            0.3         0.375
##  9 Alan Williams            0.28            0.524            0.534       0.517
## 10 Alec Burks               0.159           0.415            0.384       0.399
```

Let's first take a quick look at the missing data "Player_Stats" data frame.

```
sapply(Player_Stats, function(x) sum(is.na(x)))
```

```
##     shoot_player   conse_shot_hit  average_lag_hit conditional_prob
##                0                0                0                8
##      average_hit
##                0
```

Note that when we created the "conditional_prob" variable, some observations may have missing value since the "average_lag_shot" variable may contain zero value. We will delete these observations with missing values in conditional probability.

```
Player_Stats = Player_Stats %>% filter(!is.na(conditional_prob))
```

We can first check which players have the highest conditional probability, i.e., more likely to have hot hand.

Let's sort the data by conditional probability.

```
Player_Stats = Player_Stats %>% arrange(desc(conditional_prob))
head(Player_Stats, 10)
```

```
## # A tibble: 10 x 5
##    shoot_player     conse_shot_hit average_lag_hit conditional_prob average_hit
##    <chr>                     <dbl>           <dbl>            <dbl>       <dbl>
##  1 Axel Toupane              0.5             0.667            0.75        0.556
##  2 Lucas Nogueira            0.505           0.676            0.747       0.660
##  3 Salah Mejri               0.451           0.622            0.725       0.642
##  4 Chris McCullough          0.389           0.556            0.7         0.5
##  5 DeAndre Jordan            0.490           0.704            0.696       0.714
##  6 Deyonta Davis             0.36            0.52             0.692       0.511
##  7 Montrezl Harrell          0.436           0.644            0.677       0.652
##  8 JaVale McGee              0.437           0.652            0.671       0.652
##  9 Nick Collison             0.429           0.643            0.667       0.609
## 10 Walter Tavares            0.667           1                0.667       0.8
```

Comparing the "conditional_prob" variable and the "average_hit" variable, some players have a slightly higher conditional probability but some also have a lower conditional probability.

We can sort the data by the value of difference between conditional and unconditional probabilities.

```
Player_Stats$diff_prob = Player_Stats$conditional_prob - Player_Stats$average_hit
Player_Stats = left_join(Player_Stats, Player_Shots, by = 'shoot_player')
Player_Stats = Player_Stats %>% arrange(desc(diff_prob))
head(Player_Stats)
```

```
## # A tibble: 6 x 7
##   shoot_player conse_shot_hit average_lag_hit conditional_prob average_hit
##   <chr>                 <dbl>           <dbl>            <dbl>       <dbl>
## 1 Lamar Patte~         0.0909           0.182             0.5        0.2
## 2 Diamond Sto~         0.125            0.25              0.5        0.231
## 3 Kyle Wiltjer         0.167            0.333             0.5        0.286
## 4 Marcus Geor~         0.25             0.5               0.5        0.286
## 5 Chris McCul~         0.389            0.556             0.7        0.5
## 6 Axel Toupane         0.5              0.667             0.75       0.556
## # ... with 2 more variables: diff_prob <dbl>, shot_count <int>
```

Comparing the "conditional_prob" variable and the "average_hit" variable, some players have a slightly higher conditional probability but some also have a lower conditional probability. We can sort the data by the value of difference between conditional and unconditional probabilities. We can see that Lamar Patterson has the highest difference between the two probabilities, at 30%. But we could also see that the sample size for Patterson is pretty small. For Damjan Rudez and Miles Plumlee, we have about 90 observations and their differences in the probabilities are about 0.16.

T-test for statistical significance on the difference

More rigorously, we can use a t-test to test if the players' probability of hitting the goal is statistically significantly different than their conditional probability.

We need to choose a significance level before we perform the test. If the test produces a p-value less than the chosen significance level, then we say that there is a statistically significant difference between the two probabilities; otherwise, we fail to find evidence to support that the two probabilities are statistically significantly different from each other.

The most commonly used significance level is 0.05.

To perform a t-test, we will use the "t.test()" function.

```
t.test(Player_Stats$conditional_prob,Player_Stats$average_hit)
```

```
##
##  Welch Two Sample t-test
##
## data:  Player_Stats$conditional_prob and Player_Stats$average_hit
## t = -1.4416, df = 871.96, p-value = 0.1498
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.021497691  0.003290576
## sample estimates:
## mean of x mean of y
## 0.4381706 0.4472742
```

The first number is the t-statistics and the second number is the p-value.

Note that the p value for the t test is about 0.12, which is higher than the conventional significance level 0.05. Thus the conditional probability is not statistically significantly different than the average success rate. In other words, in the analysis of conditional probability, we fail to find evidence to support the "hot hand".

Autocorrelation Coefficient

We can calculate the autocorrelation coefficient by calculating the correlation coefficient between the "current_shot_hit" variable and the "lag_shot_hit" variable.

```
cor(Shotlog$current_shot_hit, Shotlog$lag_shot_hit)
```

```
## [1] 0.0002267794
```

As we can see, though the autocorrelation coefficient is positive, the magnitude is very small and close to zero. Since some players may have "hot hand", and hence strong correlation between outcomes of adjacent shots, while some may not. We can also calculate autocorrelation coefficient for each player.

```
Autocorr_Hit = Shotlog %>% group_by(shoot_player) %>%
  summarise(autocorr = cor(current_shot_hit, lag_shot_hit)) %>% ungroup()
```

```
## Warning in cor(current_shot_hit, lag_shot_hit): the standard deviation is zero

## Warning in cor(current_shot_hit, lag_shot_hit): the standard deviation is zero

## Warning in cor(current_shot_hit, lag_shot_hit): the standard deviation is zero

## Warning in cor(current_shot_hit, lag_shot_hit): the standard deviation is zero

## Warning in cor(current_shot_hit, lag_shot_hit): the standard deviation is zero

## Warning in cor(current_shot_hit, lag_shot_hit): the standard deviation is zero

## Warning in cor(current_shot_hit, lag_shot_hit): the standard deviation is zero

## Warning in cor(current_shot_hit, lag_shot_hit): the standard deviation is zero
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(Autocorr_Hit, 10)
```

```
## # A tibble: 10 x 2
##    shoot_player    autocorr
##    <chr>              <dbl>
##  1 A.J. Hammons      0.145
##  2 Aaron Brooks      0.0170
##  3 Aaron Gordon      0.0134
##  4 Adreian Payne    -0.187
##  5 Al Horford       -0.0214
##  6 Al Jefferson     -0.133
##  7 Al-Farouq Aminu   0.0186
##  8 Alan Anderson    -0.0767
##  9 Alan Williams     0.0292
## 10 Alec Burks       -0.0214
```

How informative the autocorrelation coefficient also depends on the number of shots per game for each player. Let's add the number of shots and the number of shots per game to the autocorrelation matrix and sort the data by the size of autocorrelation coefficient.

```
Player_Game_Shot = Player_Game %>% group_by(shoot_player) %>%
  summarise(avg_shot_game = mean(shot_per_game)) %>%
  ungroup()
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
head(Player_Game_Shot)
```

```
## # A tibble: 6 x 2
##   shoot_player   avg_shot_game
##   <chr>                  <dbl>
## 1 A.J. Hammons            2.8
## 2 Aaron Brooks            4.84
## 3 Aaron Gordon           10.8
## 4 Aaron Harrison          1
## 5 Adreian Payne           3.6
## 6 Al Horford             11.8
```

```
Autocorr_Hit = left_join(Autocorr_Hit, Player_Game_Shot,
                         by = 'shoot_player')
Autocorr_Hit = Autocorr_Hit %>% arrange(desc(autocorr))
head(Autocorr_Hit)
```

```
## # A tibble: 6 x 3
##   shoot_player        autocorr avg_shot_game
##   <chr>                  <dbl>         <dbl>
## 1 Kyle Wiltjer           0.632          1.75
## 2 Lamar Patterson        0.389          3.75
## 3 Diamond Stone          0.333          2.6
## 4 Johnny O'Bryant III    0.302          3
## 5 Chris Andersen         0.25           2.2
## 6 Axel Toupane           0.250          3
```

We will merge the Player_Game_Shot dataframe to the Player_Shots dataframe since both dataframes are measured in player level and both contain information on the number of shots.

```
Player_Shots = left_join(Player_Shots, Player_Game_Shot,
                         by = 'shoot_player')
head(Player_Shots)
```

```
##     shoot_player shot_count avg_shot_game
## 1   A.J. Hammons         42       2.80000
## 2   Aaron Brooks        300       4.83871
## 3   Aaron Gordon        864      10.80000
## 4 Aaron Harrison          4       1.00000
## 5  Adreian Payne         54       3.60000
## 6     Al Horford        801      11.77941
```

Save updated data

```
write.csv(Shotlog, 'Shotlog2.csv', row.names=FALSE)
write.csv(Player_Stats, 'Player_Stats2.csv', row.names=FALSE)
write.csv(Player_Shots, 'Player_Shots2.csv', row.names=FALSE)
```