STAT428 Group6 Project Written Report

Simulating NBA Match Results and Predicting NBA Playoff Teams

Zepeng Xiao, zepengx2
Shuogong, shuog2
Xiaoping Hua, xh3
Dongfan Li, dongfan2
Sixin Ma, sixinma2

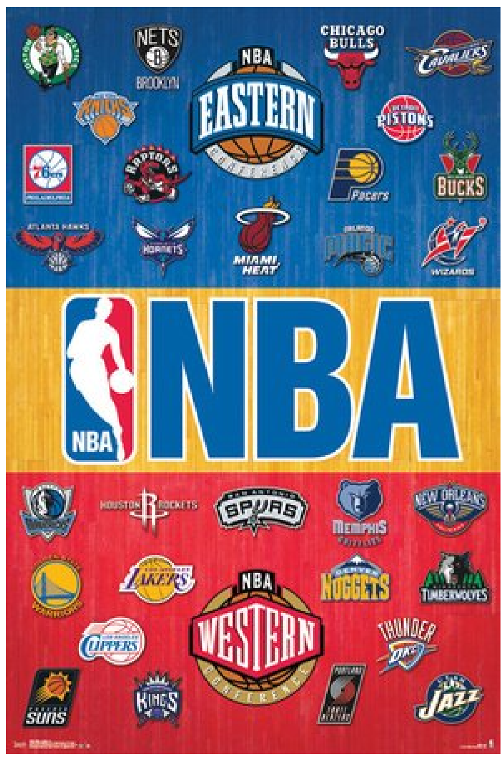
April 28, 2019

Abstract

The main purpose of our project is simulating match results and predicting NBA playoff teams for season 2017-18 based on this season's real data, and comparing our results with this season's real data to calculate our method's accuracy, since analyzing and predictiong NBA stats is popular currently. The method we used in our project includes **Random number generations**, **Permutation test** and **Ridge regression**.

We found that regarding the playoff teams for season 2017-18, the accuracy of our prediction is 87.5% (14/16) when comapring the actual result. The result can be regarded as accurate beacuse we used each team's former performance in this season to make predictions, which is reliable. But there are still some conditions that were not included in the model, like player's injury, team's stamina and so on.

Introduction



The National Basketball Association (NBA) is one of the most worldwide popular sporting events that is held every year in the world.

- It has two conferences, East and West, with 15 teams in each division.
- 30 teams fight in the league fight for a champion in a year-long season from October to June every year.
- Each season is split into regular season and playoffs.
- Each team has 82 matches to play in their regular season
- Teams in the same conference would play more often than those.
- Only the top 8 teams in each division are able to enter the playoffs of the season.
- Ranks are determined ascending by a special statistic called Games Behind:
  - When team a is the leading team of the conference.

$$TeamB'sgamesbehind = \frac{(TeamA'sWins - TeamA'sLosses) - (TeamB'sWins - TeamB'sLosses)}{2}$$

Thus which 16 teams would enter the playoffs every season is one of the biggest mystery in NBA every season.

Our goal, as introduced above, is simulating match results and predicting NBA playoff teams. In other words, we are going to predict which eight teams in each conference would get the lowest games behind by the end of the regular season.

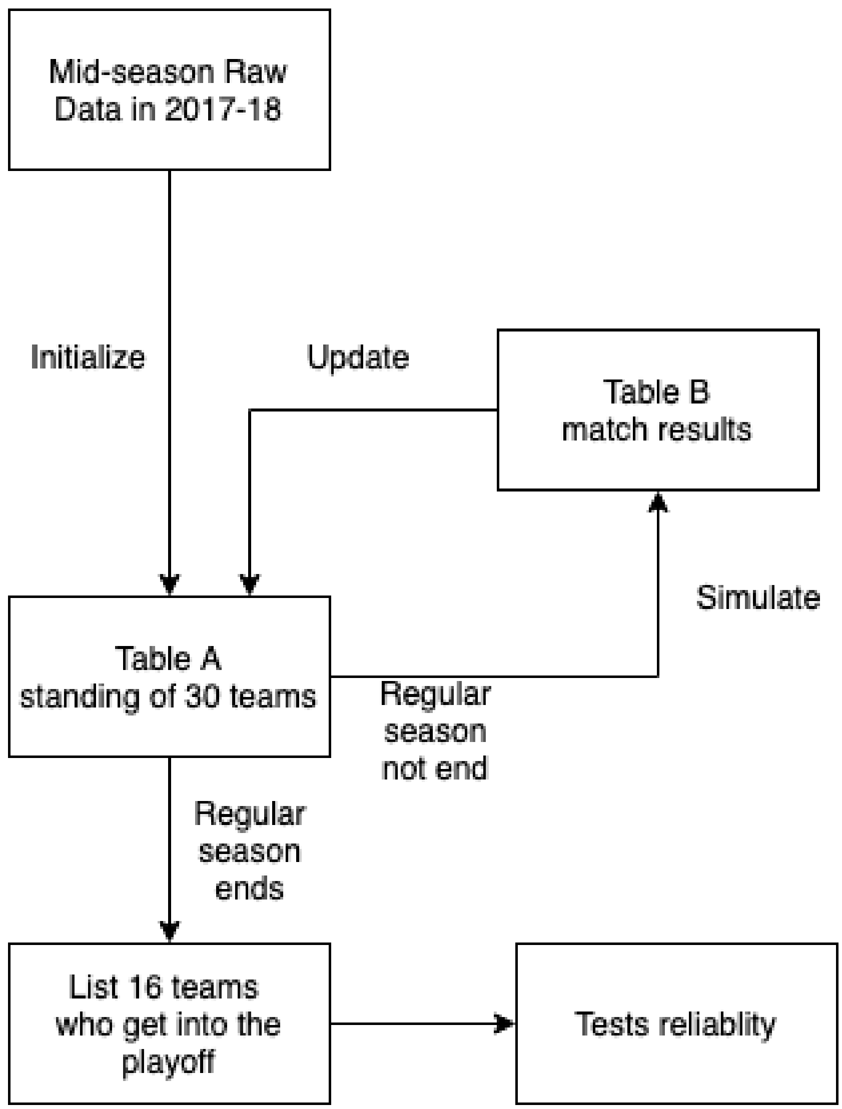
The focus is on the past 17-18 season. We will try to predict based on prior part of this season's real data, and comparing our results with this season's real data in the rest season to calculate our method's accuracy. If we found our prediction is reliable in this season, we have prepared data in the past 10 years for test and verification.

Methods

Processing Data

We made two tables A and B to store the standing data of all teams (in A) and the match data of each matches (in B).

Then use the "current" standing data in A to simulate upcoming matches in B and use the following match data to regenerate new standing data in A



Raw data used:

From 2017-18_standing.csv

```r
str(standing_data[,c(1,2,13,14,15,16,17,18,20,21)])
```

```
## 'data.frame':    5040 obs. of  10 variables:
##  $ stDate  : Factor w/ 168 levels "2017-10-17","2017-10-18",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ teamAbbr: Factor w/ 30 levels "ATL","BKN","BOS",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ homeWin : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ homeLoss: int  0 0 0 0 0 0 0 0 0 1 ...
##  $ awayWin : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ awayLoss: int  0 0 1 0 0 0 0 0 0 0 ...
##  $ confWin : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ confLoss: int  0 0 1 0 0 0 0 0 0 1 ...
##  $ lastTen : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ gamePlay: int  0 0 1 0 0 1 0 0 0 1 ...
```

From 2017-18_teamBoxScore.csv

```r
str(match_data[,c(1,10,11,13,14,17,73)])
```

```
## 'data.frame':    2460 obs. of  7 variables:
##  $ gmDate  : Factor w/ 168 levels "2017-10-17","2017-10-18",..: 1 1 1 1 2 2 2 2 2 2 ...
##  $ teamAbbr: Factor w/ 30 levels "ATL","BKN","BOS",..: 3 6 11 10 4 9 2 12 16 22 ...
##  $ teamConf: Factor w/ 2 levels "East","West": 1 1 2 2 1 1 1 1 1 1 ...
##  $ teamLoc : Factor w/ 2 levels "Away","Home": 1 2 1 2 1 2 1 2 1 2 ...
##  $ teamRslt: Factor w/ 2 levels "Loss","Win": 1 2 2 1 1 2 1 2 1 2 ...
##  $ teamPTS : int  99 102 122 121 90 102 131 140 109 116 ...
##  $ opptPTS : int  102 99 121 122 102 90 140 131 116 109 ...
```

Initial table A:

```r
str(tablea180311)
```

```
## 'data.frame':    30 obs. of  26 variables:
##  $ teamname     : Factor w/ 30 levels "ATL","BKN","BOS",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ confname     : Factor w/ 2 levels "East","West": 1 1 1 1 1 1 2 2 1 2 ...
##  $ homerate     : num  0.429 0.353 0.657 0.528 0.455 ...
##  $ awayrate     : num  0.156 0.273 0.719 0.323 0.242 ...
##  $ last10       : num  0.2 0.2 0.6 0.5 0.3 0.4 0.3 0.6 0.3 0.7 ...
##  $ homescorewin : num  105 105 104 108 103 ...
##  $ homescorelost: num  108.2 108.4 99.7 107.2 105.9 ...
##  $ awayscorewin : num  102 107 104 105 104 ...
##  $ awayscorelost: num  109 112 101 108 113 ...
##  $ confrate     : num  0.209 0.359 0.674 0.425 0.45 ...
##  $ numberdayoff : num  2 3 3 1 2 2 1 2 2 2 ...
##  $ lastgame     : Factor w/ 2 levels "Away","Home": 1 1 1 2 1 1 2 2 2 1 ...
##  $ totalmatch   : int  67 67 67 67 66 66 67 67 66 67 ...
##  $ confmatch    : int  43 39 43 40 40 41 45 45 45 41 ...
##  $ homematch    : int  35 34 35 36 33 33 36 36 35 33 ...
##  $ awaymatch    : int  32 33 32 31 33 33 31 31 31 34 ...
##  $ l1           : num  1 1 1 2 2 1 1 2 2 1 ...
##  $ l2           : num  1 2 2 1 1 1 2 2 1 1 ...
##  $ l3           : num  1 1 2 1 2 2 2 1 1 2 ...
##  $ l4           : num  2 1 1 1 1 2 1 1 1 2 ...
##  $ l5           : num  1 1 2 1 2 1 1 2 1 2 ...
##  $ l6           : num  2 1 2 1 1 1 1 2 2 2 ...
```

```
##  $ l7           : num  1 2 2 2 1 2 2 1 1 2 ...
##  $ l8           : num  1 1 2 2 1 1 1 1 1 2 ...
##  $ l9           : num  1 1 1 2 1 2 1 2 1 2 ...
##  $ l10          : num  1 1 1 2 1 1 1 2 2 1 ...
```

Columns' meaning:

- "Teamname": Name of team
- "Confname": East or West, which defines which conference the team belongs to.
- "Homerate": winrate at home
- "Awayrate": winrate away
- "Last10": winrate in the last 10 matches
- "Homescorewin": score won when at home
- "Homescorelost": score lost when at home
- "Awayscorewin": score win when playing away home
- "Awayscorelost": score lost when to play away home
- "Confrate": winrate when playing within the conference
- "Numberdayoff": number of days since the last match
- "Lastgame": is the last game played at home or away
- "totalmatch":number of matches played
- "confmatch":number of matches played within conference
- "Homematch": number of matches played at home
- "Awaymatch": number of matches played away
- "l1" ~ "l10": is the last 1~10 win or lose

Generalized Linear Regression

Since each

Permutation Test

The generalized linear model we used to predict the game result includes predictors that might be correlated to each other. For example, `homeTeamRate` and `homeTeamlast10` seem to be positively correlated in common sense. Therefore, to improve our regression fit, we want to thorougly examine the correlations among the variables and proceed to use ridge regression if the variables are confirmed to be correlated. The variables for testing are,

- homeTeamRate
- awayTeamRate
- homeTeamlast10
- awayTeamlast10
- histRate

A permutation test essentially checks if X and Y have the same distribution by doing the following procedure,

1. Observe a test statistic for the null hypothesis, H_0
2. For each replicated $b = 1, 2, ..., B$:
   - Generate a random permutation π
   - Generate a new test statistic from the random permutation
3. Get a Monte Carlo estimate of the p-value by calculating the probability of obtaining a new test statistic that is more extreme than the observed test statistic in the B replicates
4. Reject H_0 at a significance level α if the p-value is less than α.

In our case, we can inherit the idea of permutation test and adapt it to paired data to check the correlation in between. If there is truly no association between X and Y, the distribution of $(X_i, Y_{\pi(i)})$ will be the same as that of (X_i, Y_i), where $\pi(i)$ is the i-th element of a permutation π of $1, 2, ..., 13$. We implement this idea by randomly permuting Y and pairing it with a fixed X and get a p-value for testing $H_0 : \rho = 0$ versus $H_1 : \rho \neq 0$, where ρ is either the Pearson correlation coefficient or the Spearman correlation coefficient:

- The *Pearson Method* evaluates the **linear** relationship between two continuous variables, where a change in one variable is associated with a **proportional** change in the other variable.
- The *Spearman Method* is based on the ranked values for each variable rather than the raw data. It evaluates the **monotonic** relationship between two continuous or ordinal variables, where the variables tend to change together, but not necessarily at a constant rate. It is more general than the Pearson method.

To determine which correlation coefficient to use for which pair of comparison, we plot variables against each another and added some noise using `jitter()` to visualize the trend of relationship and decide which method to use.

According to the plots, *Home Rate vs. Home Last 10*, *Away Rate vs. Away Last 10*, *Home Last 10 vs. Historical Rate*, and *Away Last 10 vs. Historical Rate* appear to be linear. We use the Pearson method for these pairs. For the rest pairs, trends are not that obvious. Therefore, we use the more genearl Spearman method.

Ridge Regression

Results

Linear Regrssion

Permutation Test

By performing permutation tests, we obtained the following table, which shows all the pairs we tested, the method we used, the p-value for the permutation test, and the decisions based on the p-values.

Pairs	Method	p-value	Decision
Home Rate vs. Home Last 10	Pearson	9.999e-05	Correlated
Away Rate vs. Away Last 10	Pearson	9.999e-05	Correlated
Home Rate vs. Historical Rate	Spearman	9.999e-05	Correlated
Away Rate vs. Historical Rate	Spearman	9.999e-05	Correlated
Home Last 10 vs. Historical Rate	Pearson	9.999e-05	Correlated
Away Last 10 vs. Historical Rate	Pearson	9.999e-05	Correlated
Home Rate vs. Away Rate	Spearman	0.4131587	Not Correlated
Home Last 10 vs. Away Last 10	Spearman	9.999e-05	Correlated
Home Rate vs. Away Last 10	Spearman	0.02689731	Correlated
Away Rate vs. Home Last 10	Spearman	0.02329767	Correlated

We found that all pairs of variables have correlation except *Home Rate vs. Away Rate*. This observation suggests that there exists problem of collinearity among the predictors. It leads us to further examine the VIF (Variance Inflation Factor) of the predictors and fit a Ridge Regression model to remedy the problem of collinearity.

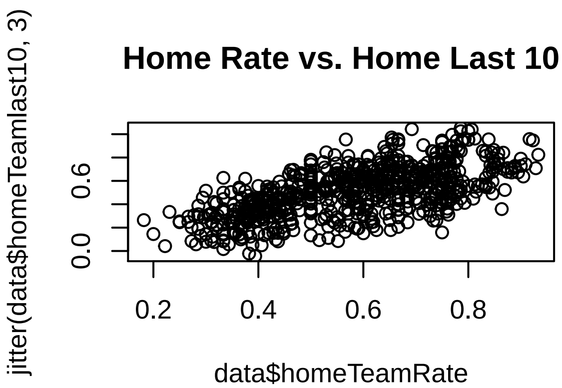
Ridge Regression

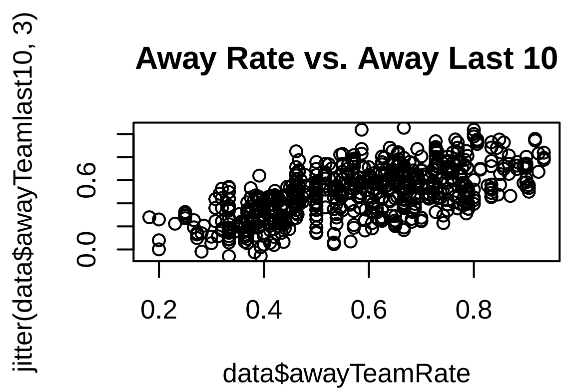
Discussion

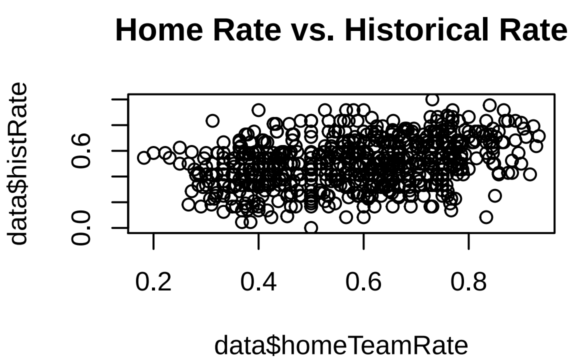
The test for correlation is not very accurate because the “Last 10” data (i.e. `homeTeamlast10` and `awayTeamlast10`) are highly categorical (discrete) because based on their calculation criteria: the number of times the team wins, which is an integer, divided by 10. When the “Last 10” data is paired with the other continuous variables, such as `homeTeamRate` and `histRate`, the correlation between a continuous variable and a somewhat discrete variable cannot be simply determined by correlation coefficients.

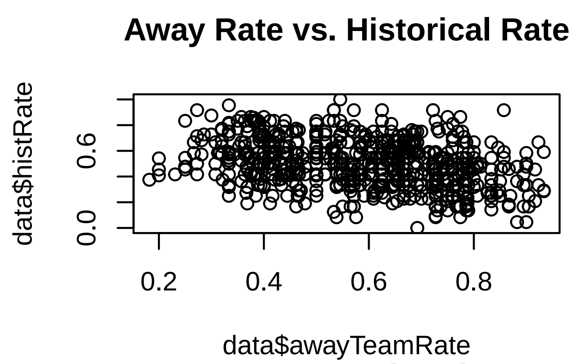
Appendix

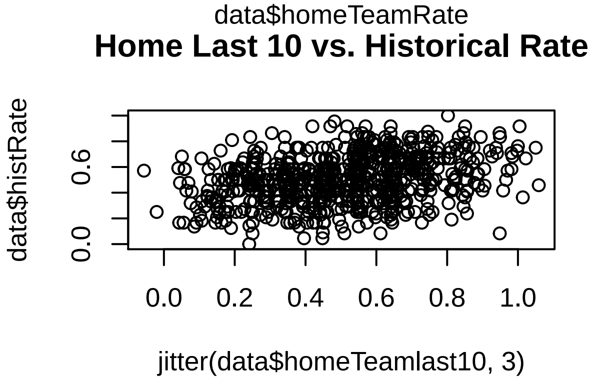
Scatterplots for Correlation

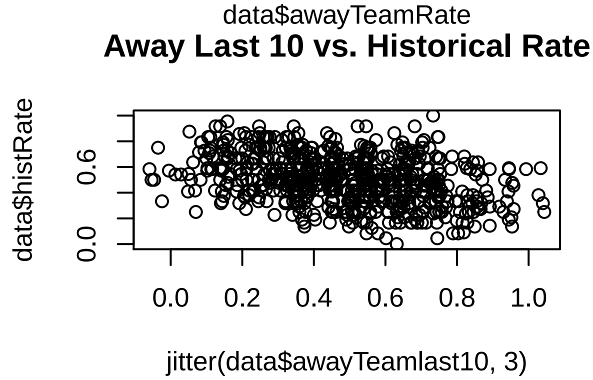


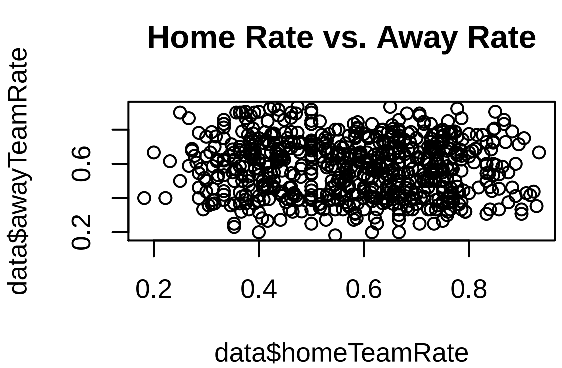


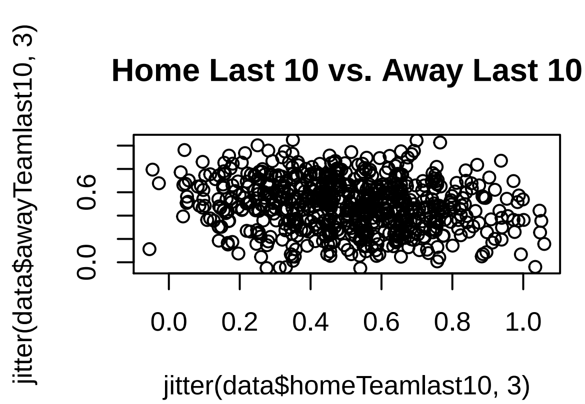


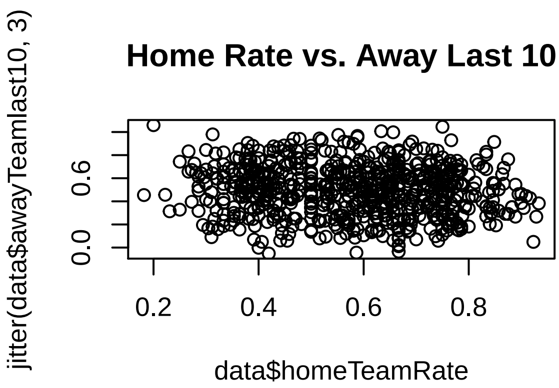


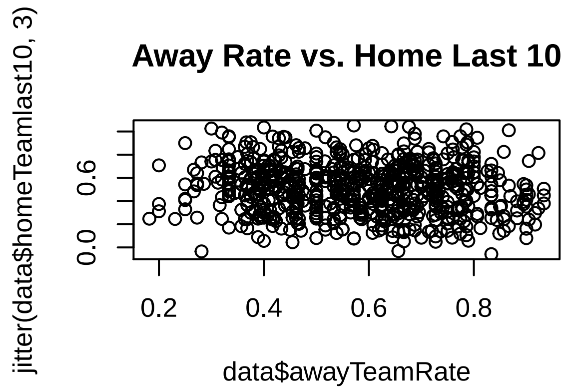












Permutation Test with either correlation coefficient

```
perm_test = function(X, Y, B = 10000, method = "pearson") {
  nu = seq_along(X)
  reps = numeric(B)

  if (method == "pearson") { # Pearson Method - default
    rho0 = abs(cor(X, Y))
    for ( i in 1:B ) {
      perm = sample(nu, size = length(X), replace = FALSE)
      X1 = X[perm]
      reps[i] = abs(cor(X1, Y))
    }
    pval = mean(c(rho0, reps) >= rho0)
  } else if (method == "spearman") { # Spearman Method
    rho0 = cor(X, Y, method = "spearman")
    t0 = abs(rho0*sqrt((length(X) - 2)/(1 - rho0^2)))
    for ( i in 1:B ) {
      perm = sample(nu, size = length(X), replace = FALSE)
      X1 = X[perm]
      rho = cor(X1, Y, method = "spearman")
      reps[i] = abs(rho*sqrt((length(X) - 2)/(1 - rho^2)))
    }
    pval = mean(c(t0, reps) >= t0)
  }

  return(pval)
}

# running the tests
perm_test(data$homeTeamRate, data$homeTeamlast10, 10000, "pearson")
perm_test(data$awayTeamRate, data$awayTeamlast10, 10000, "pearson")
perm_test(data$homeTeamRate, data$histRate, 10000, "spearman")
perm_test(data$awayTeamRate, data$histRate, 10000, "spearman")
perm_test(data$homeTeamlast10, data$histRate, 10000, "spearman")
perm_test(data$awayTeamlast10, data$histRate, 10000, "spearman")
perm_test(data$homeTeamRate, data$awayTeamRate, 10000, "spearman")
perm_test(data$homeTeamlast10, data$awayTeamlast10, 10000, "spearman")
perm_test(data$homeTeamRate, data$awayTeamlast10, 10000, "spearman")
perm_test(data$awayTeamRate, data$homeTeamlast10, 10000, "spearman")
```