

Lucky vs. Unlucky Teams in Sports

Introduction

Assuming gambling odds give true probabilities, one can classify a team as having been lucky or unlucky so far. Do results of matches between lucky and unlucky teams fit the gambling odds?

Determining “Luck”

In this project, we will define the luck of, for example, team A to be $\text{Luck (A)} = (\text{actual number of wins}) - (\text{expected number of wins})$. For our purposes, the expected number of wins will be defined as the summation of all the team’s probabilities of winning.

As an example, suppose team A has played 5 games so far and had the following probabilities of winning and actual results:

Game	Probability of Winning	Result
1	.70	Win
2	.60	Loss
3	.65	Win
4	.45	Loss
5	.40	Loss

We would calculate the luck of team A as follows:

$$\begin{aligned} \text{luck(A)} &= (\text{actual number of wins}) - (\text{expected number of wins}) \\ &= 2 - (.70 + .60 + .65 + .45 + .40) = -0.8 \end{aligned}$$

Therefore, at this point in the season, Team A’s “luck” is -0.8. Therefore, Team A has been “unlucky” up to this point in the season.

*Note: “Luck” refers to games that have already been played, so when analyzing the first game of a season, every team has $\text{luck}=0$.

Calculating Probabilities

Many websites are available to convert moneylines to probabilities. For example, I used the website: http://www.covers.com/sportsbetting/money_lines.aspx.

However, these probabilities take into account the bookie's take in the bet, so the sum of the probabilities ends up being greater than 1. For example, in Week 9 of the last NFL season, there was Atlanta (-300) vs. Indianapolis (+250) converts to Atlanta (75.0%) vs. Indianapolis (28.6%) so we have $.750 + .286 = 1.036$. For our purposes, these probabilities need to sum to one as there is 100% that either one team or the other will win. [Ties are possible in the NFL. However they are not very likely. There have only been two in the past 15 years. Thus, the possibility of tie will be ignored for the purposes of this research.] Therefore, the new probability of team A beating team B will computed as follows:

$$P_{\text{New}}(\text{team A wins}) = P(\text{A wins})/[P(\text{A wins}) + P(\text{B wins})]$$

Therefore, in the example above, we would now have the following probabilities:

$$P(\text{Atlanta wins}) = .75/ (.75 + .286) = .724$$

$$P(\text{Indianapolis wins}) = .286/ (.75 + .286) = .276$$

The probabilities now sum to 1, as desired.

Using “Luck” for Analysis

Going back to the previous example, suppose team A is about to play team B. Take whichever of the two teams is relatively luckier—say $\text{luck}(B) = -0.2$, so it's B—and calculate the difference $\text{luck-difference} = \text{luck}(B) - \text{luck}(A) = 0.6$.

So now when A plays B, we take the relatively luckier team, B, look at the gambling odds of B winning, say 0.45, and then we record the three pieces of data

Gambling Odds	Luck Difference	Result (win or loss for Team B)
0.45	0.60	Loss

We want this data for every game in the season. Once you've got the data we can do a linear regression.

Results

```
> summary(research.lm)
Call:
lm(formula = result ~ probability + luck, data = research)

Residuals:
    Min       1Q   Median       3Q      Max
-0.8633 -0.4444  0.1699  0.3367  0.8059

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.04891    0.10150   0.482   0.6303
probability  1.06660    0.16957   6.290 1.52e-09 ***
luck        -0.04447    0.02634  -1.689  0.0926 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4519 on 237 degrees of freedom
Multiple R-squared:  0.1432,    Adjusted R-squared:  0.1359
```

F-statistic: 19.8 on 2 and 237 DF, p-value: 1.119e-08

Betting Strategy

Let's try the following betting strategy:

Bet one dollar on the more lucky team. If the probability (from betting odds) of that team winning is p then we will lose with probability $1-p$ or gain $(1/p) - 1$ with probability p , and the variance from this bet is $(1-p)/p$. If we bet c (dollars) then the variance is $c^2 (1-p)/p$. Here, we will try the strategy where we choose c to make the variance = 1, that is choose $c = \sqrt{p/(1-p)}$

We will use the data to see how much one would have gained or lost by making this bet on each of the N matches in a season. The point is that the theoretical variance (assuming betting odds correspond to true probabilities) = N so we can compare the observed overall gain-or-loss to the "standard deviation" = \sqrt{N} .

Results of the Betting Strategy

The betting strategy above yields a loss of approximately \$7.61. Therefore, the use of "luck" has not been able effective in helping us implement a successful betting strategy.

Appendix

```
#####
# Python Code to Create CSV file with the gambling odds, luck      #
# difference, and result for the relatively luckier team          #
#####

import csv, sys
filename = 'all_games_new'

try:
    csv_in = csv.reader(open(filename + '.csv', 'r'))
except:
    print 'Invalid log file name:', filename
    sys.exit(1)

research = {}

for row in csv_in:
    if row == []:
        continue
    if row[0] == 'Number':
        continue
    Number = int(row[0])
    Week = int(row[1])
    Home = row[2]
    Prob_H = float(row[3])
    Result_H = int(row[4])
```

```

Visitor = row[5]
Prob_V = float(row[6])
Result_V = int(row[7])
Luck_H = float(row[8])
Luck_V = float(row[9])
Luck_Diff = float(row[10])

if Number not in research:
    research[Number] = {}
research[Number] = Prob_H, Result_H, Luck_H, Prob_V, Result_V, Luck_V,
Luck_Diff

# Create a csv file:
file = open('final_research_data.csv', 'wb')
csv_out= csv.writer(file)
for number in sorted(research):
    if research[number][2] > research[number][5]:
        csv_out.writerow([research[number][0], research[number][6],
research[number][1]])
    else:
        csv_out.writerow([research[number][3], research[number][6],
research[number][4]])
file.close()

```

```

#####
# Python Code to Create CSV file with results from betting strategy #
#####

```

```

import csv, sys, math

filename = 'betting_strategy_53'

# Function to compute return on bet

def bet_return(moneyline, result, bet):
    if moneyline < 0:
        if result == 1:
            x = 100/((-1*moneyline)/bet)
            return x
        else:
            x = -1*bet
            return x
    else:
        if result == 1:
            x = moneyline/(100/bet)
            return x
        else:
            x = -1*bet
            return x

try:
    csv_in = csv.reader(open(filename + '.csv', 'r'))
except:

```

```

    print 'Invalid log file name:', filename
    sys.exit(1)

new_research = {}

for row in csv_in:
    if row[0] == []:
        continue
    if row[0] == 'Number':
        continue

    number = int(row[0])
    team = row[1]
    probability = float(row[2])
    luck = float(row[3])
    moneyline = int(row[4])
    result = int(row[5])
    bet = float(row[6])

    if number not in new_research:
        new_research[number] = {}
    new_research[number] = team, probability, luck, moneyline, result, bet,
bet_return(moneyline, result, bet)

file = open('betting_with_results.csv', 'wb')
csv_out = csv.writer(file)
csv_out.writerow(['Team', 'Probability', 'Luck', 'Moneyline', 'Result',
'Bet', 'Winnings'])
for number in sorted(new_research):
    csv_out.writerow([new_research[number][0], new_research[number][1],
new_research[number][2], new_research[number][3], new_research[number][4],
new_research[number][5], new_research[number][6]])
file.close()

```