3 Pythagorean expectation and the NBA

*Jiannan Xu**

17 July 2020

Pythagorean Expectation and English Soccer

In soccer, teams score goals, and we can calculate Pythagorean Expectations based on goals scored and goals conceded.

The structure of competition in soccer in most countries around the world is different from the sports we have looked at so far. Rather than leagues operating as independent entities, they are connected through a hierarchical system, sometimes called "the pyramid". In England, the English Premier League is at the top of the pyramid (it used to be called the First Division) and contains 20 teams.

Beneath the Premier League is The Football League Championship (it used to be called Division Two) and it contains 24 teams. The Premier League and the Championship are linked via the system of promotion and relegation. At the end of each season, the three worst performing teams (measured by points won in competition) are relegated to play Championship soccer in the following season, to be replaced by the three best performing teams in the Championship. Beneath the Championship are two more leagues - League One (formerly Third Division) and League Two (formerly Fourth Division). These leagues are also linked, hierarchically, through promotion and relegation. Thus it makes sense to think of these four divisions as part of a common system.

In any one season, there are 92 teams in the system. Even though teams compete in different divisions, we can define both win percentage and Pythagorean Expectation for each team, in order to see how well the data fits.

In each of the four divisions, every team plays every other team twice in a season, once at home and once away. There is no playoff, so the champion is the team at the end of the season with the largest number of points (3 points for a win, one for a draw (tie)). Unlike the leagues we have looked at so far, draws are not only possible but are quite common. We need to adjust our definition of win percentage for this case. We could create a statistic such as the percentage of maximum possible points, but instead, we do something simpler- we give a value of 1 for a win, 0 for a loss, and 1/2 for a draw.

We now follow the same procedure we have used to date.

```
# Load the packages

options(warn = -1)
library("readxl",quietly = TRUE)
```

*ansonxjn@umich.edu; Master student at Department of Statistics, University of Michigan.

```
library("tidyverse",quietly = TRUE)
library("dplyr",quietly = TRUE)
library("ggplot2",quietly = TRUE)
```

```
# Load the data.
# Our data covers the 2017/18 season

Eng18 = read_excel('Engsoccer2017-18.xlsx')
names(Eng18)
```

```
## [1] "Div"      "Date"     "HomeTeam" "AwayTeam" "FTHG"     "FTAG"
## [7] "FTR"
```

```
# We can see what our dataframe looks like simply by using head()
# and tail function
head(Eng18)
```

```
## # A tibble: 6 x 7
##   Div   Date  HomeTeam       AwayTeam      FTHG  FTAG FTR
##   <chr> <chr> <chr>          <chr>        <dbl> <dbl> <chr>
## 1 EPL   43047 Arsenal        Leicester        4     3 H
## 2 EPL   43077 Brighton       Man City         0     2 A
## 3 EPL   43077 Chelsea        Burnley          2     3 A
## 4 EPL   43077 Crystal Palace Huddersfield     0     3 A
## 5 EPL   43077 Everton        Stoke            1     0 H
## 6 EPL   43077 Southampton    Swansea          0     0 D
```

```
tail(Eng18)
```

```
## # A tibble: 6 x 7
##   Div   Date  HomeTeam     AwayTeam      FTHG  FTAG FTR
##   <chr> <chr> <chr>        <chr>        <dbl> <dbl> <chr>
## 1 FL2   43225 Forest Green Grimsby          0     3 A
## 2 FL2   43225 Lincoln      Yeovil           1     1 D
## 3 FL2   43225 Mansfield    Crawley Town     1     1 D
## 4 FL2   43225 Notts County Luton            0     0 D
## 5 FL2   43225 Swindon      Accrington       3     0 H
## 6 FL2   43225 Wycombe      Stevenage        1     0 H
```

```
Eng18['hwinvalue']=ifelse(Eng18$FTR=='H',1,ifelse(Eng18$FTR=='D',0.5,0))
Eng18['awinvalue']=ifelse(Eng18$FTR=='A',1,ifelse(Eng18$FTR=='D',0.5,0))
Eng18['count'] = 1
```

```
# Once again we have to create separate dfs to calculate home team
# and away team performance.
# Here is the home team df, including only the variables we need.
Enghome <-  Eng18 %>% group_by(HomeTeam,Div)%>%
```

```
            dplyr::summarise(count = sum(count),
                             hwinvalue = sum(hwinvalue),
                             FTHG = sum(FTHG),
                             FTAG = sum(FTAG)
                            )%>%
                            ungroup()%>%
                            rename(team = HomeTeam,
                                   Ph = count,
                                   FTHGh = FTHG,
                                   FTAGh = FTAG)%>%
                            arrange(team)
head(Enghome)
```

```
## # A tibble: 6 x 6
##   team           Div      Ph hwinvalue FTHGh FTAGh
##   <chr>          <chr> <dbl>     <dbl> <dbl> <dbl>
## 1 Accrington     FL2      23      18.5    42    19
## 2 AFC Wimbledon FL1      23      11      25    30
## 3 Arsenal        EPL      19      16      54    20
## 4 Aston Villa    FLCH     23      17.5    42    19
## 5 Barnet         FL2      23      11      24    25
## 6 Barnsley       FLCH     23       9.5    25    32
```

```
tail(Enghome)
```

```
## # A tibble: 6 x 6
##   team       Div      Ph hwinvalue FTHGh FTAGh
##   <chr>      <chr> <dbl>     <dbl> <dbl> <dbl>
## 1 West Brom EPL      19       7.5    21    29
## 2 West Ham  EPL      19      10      24    26
## 3 Wigan     FL1      23      17      37    11
## 4 Wolves    FLCH     23      18.5    47    18
## 5 Wycombe   FL2      23      14.5    43    35
## 6 Yeovil    FL2      23      10.5    29    26
```

```
# Now we create the mirror image df for the away team results.
Engaway <-  Eng18 %>% group_by(AwayTeam)%>%
            dplyr::summarise(count = sum(count),
                             awinvalue = sum(awinvalue),
                             FTHG = sum(FTHG),
                             FTAG = sum(FTAG))%>%
                            ungroup()%>%
                            rename(team = AwayTeam,
                                   Pa = count,
                                   FTHGa = FTHG,
                                   FTAGa = FTAG)
```

```r
head(Engaway)
```

```
## # A tibble: 6 x 5
##   team              Pa awinvalue FTHGa FTAGa
##   <chr>          <dbl>     <dbl> <dbl> <dbl>
## 1 Accrington        23      13.5    27    34
## 2 AFC Wimbledon     23       9      28    22
## 3 Arsenal           19       6      31    20
## 4 Aston Villa       23      12      23    30
## 5 Barnet            23       6      40    22
## 6 Barnsley          23       6.5    40    23
```

```r
tail(Engaway)
```

```
## # A tibble: 6 x 5
##   team          Pa awinvalue FTHGa FTAGa
##   <chr>      <dbl>     <dbl> <dbl> <dbl>
## 1 West Brom     19       5      27    10
## 2 West Ham      19       6      42    24
## 3 Wigan         23      17.5    18    52
## 4 Wolves        23      16      21    35
## 5 Wycombe       23      15.5    25    36
## 6 Yeovil        23       7.5    49    30
```

```r
# Now we merge the two dfs to obtain a full record for each team across the season.

Eng18 <- merge(x=Enghome,y=Engaway,by=c('team'))
head(Eng18)
```

```
##            team  Div Ph hwinvalue FTHGh FTAGh Pa awinvalue FTHGa FTAGa
## 1    Accrington  FL2 23      18.5    42    19 23      13.5    27    34
## 2 AFC Wimbledon  FL1 23      11.0    25    30 23       9.0    28    22
## 3       Arsenal  EPL 19      16.0    54    20 19       6.0    31    20
## 4   Aston Villa FLCH 23      17.5    42    19 23      12.0    23    30
## 5        Barnet  FL2 23      11.0    24    25 23       6.0    40    22
## 6      Barnsley FLCH 23       9.5    25    32 23       6.5    40    23
```

```r
tail(Eng18)
```

```
##         team  Div Ph hwinvalue FTHGh FTAGh Pa awinvalue FTHGa FTAGa
## 87 West Brom  EPL 19       7.5    21    29 19       5.0    27    10
## 88  West Ham  EPL 19      10.0    24    26 19       6.0    42    24
## 89     Wigan  FL1 23      17.0    37    11 23      17.5    18    52
## 90    Wolves FLCH 23      18.5    47    18 23      16.0    21    35
## 91   Wycombe  FL2 23      14.5    43    35 23      15.5    25    36
## 92    Yeovil  FL2 23      10.5    29    26 23       7.5    49    30
```

```
# We now aggregate the home and away data for wins, games played and runs

Eng18['W'] = Eng18['hwinvalue'] + Eng18['awinvalue']
Eng18['G'] = Eng18['Ph'] + Eng18['Pa']
Eng18['GF'] = Eng18['FTHGh'] + Eng18['FTAGa']
Eng18['GA'] = Eng18['FTAGh'] + Eng18['FTHGa']

head(Eng18)
```

```
##            team  Div Ph hwinvalue FTHGh FTAGh Pa awinvalue FTHGa FTAGa
## 1    Accrington  FL2 23      18.5    42    19 23      13.5    27    34
## 2 AFC Wimbledon  FL1 23      11.0    25    30 23       9.0    28    22
## 3       Arsenal  EPL 19      16.0    54    20 19       6.0    31    20
## 4   Aston Villa FLCH 23      17.5    42    19 23      12.0    23    30
## 5        Barnet  FL2 23      11.0    24    25 23       6.0    40    22
## 6      Barnsley FLCH 23       9.5    25    32 23       6.5    40    23
##      W  G GF GA
## 1 32.0 46 76 46
## 2 20.0 46 47 58
## 3 22.0 38 74 51
## 4 29.5 46 72 42
## 5 17.0 46 46 65
## 6 16.0 46 48 72
```

```
tail(Eng18)
```

```
##         team  Div Ph hwinvalue FTHGh FTAGh Pa awinvalue FTHGa FTAGa    W
## 87 West Brom  EPL 19       7.5    21    29 19       5.0    27    10 12.5
## 88  West Ham  EPL 19      10.0    24    26 19       6.0    42    24 16.0
## 89     Wigan  FL1 23      17.0    37    11 23      17.5    18    52 34.5
## 90    Wolves FLCH 23      18.5    47    18 23      16.0    21    35 34.5
## 91   Wycombe  FL2 23      14.5    43    35 23      15.5    25    36 30.0
## 92    Yeovil  FL2 23      10.5    29    26 23       7.5    49    30 18.0
##     G GF GA
## 87 38 31 56
## 88 38 48 68
## 89 46 89 29
## 90 46 82 39
## 91 46 79 60
## 92 46 59 75
```

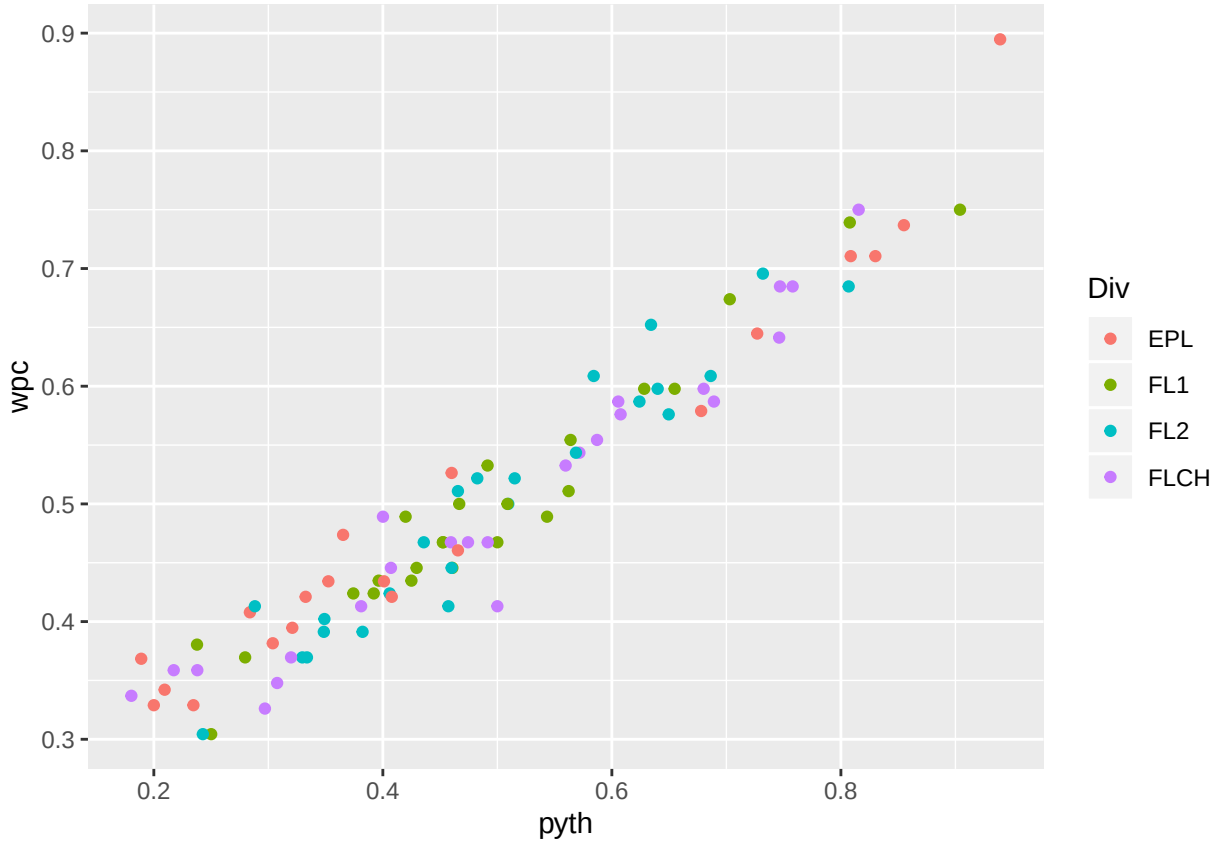
```
# The last step in organizing the data is to create variables for win percentage (wpc)

Eng18['wpc'] = Eng18['W']/Eng18['G']
Eng18['pyth'] = Eng18['GF']**2/(Eng18['GF']**2 + Eng18['GA']**2)
head(Eng18)
```

```
##            team  Div Ph hwinvalue FTHGh FTAGh Pa awinvalue FTHGa FTAGa
## 1    Accrington  FL2 23      18.5    42    19 23      13.5    27    34
## 2 AFC Wimbledon  FL1 23      11.0    25    30 23       9.0    28    22
## 3       Arsenal  EPL 19      16.0    54    20 19       6.0    31    20
## 4   Aston Villa FLCH 23      17.5    42    19 23      12.0    23    30
## 5        Barnet  FL2 23      11.0    24    25 23       6.0    40    22
## 6      Barnsley FLCH 23       9.5    25    32 23       6.5    40    23
##      W  G GF GA       wpc      pyth
## 1 32.0 46 76 46 0.6956522 0.7318804
## 2 20.0 46 47 58 0.4347826 0.3963754
## 3 22.0 38 74 51 0.5789474 0.6779745
## 4 29.5 46 72 42 0.6413043 0.7461140
## 5 17.0 46 46 65 0.3695652 0.3337013
## 6 16.0 46 48 72 0.3478261 0.3076923
```

```
tail(Eng18)
```

```
##         team  Div Ph hwinvalue FTHGh FTAGh Pa awinvalue FTHGa FTAGa    W
## 87 West Brom  EPL 19       7.5    21    29 19       5.0    27    10 12.5
## 88  West Ham  EPL 19      10.0    24    26 19       6.0    42    24 16.0
## 89     Wigan  FL1 23      17.0    37    11 23      17.5    18    52 34.5
## 90    Wolves FLCH 23      18.5    47    18 23      16.0    21    35 34.5
## 91   Wycombe  FL2 23      14.5    43    35 23      15.5    25    36 30.0
## 92    Yeovil  FL2 23      10.5    29    26 23       7.5    49    30 18.0
##     G GF GA       wpc      pyth
## 87 38 31 56 0.3289474 0.2345619
## 88 38 48 68 0.4210526 0.3325635
## 89 46 89 29 0.7500000 0.9040173
## 90 46 82 39 0.7500000 0.8155246
## 91 46 79 60 0.6521739 0.6341835
## 92 46 59 75 0.3913043 0.3822754
```

```
# Having prepared the data, we are now ready to examine it. First,
# we generate and xy plot use the Seaborn package.
# This illustrates nicely the close correlation between win percentage
# and the Pythagorean Expectation.
ggplot(data = Eng18,aes(x = pyth,y = wpc,color = Div)) + geom_point()
```



Self test

run ggplot again, but this time write y= W instead of y= wpc. What do you find? Does it make a difference?

```
# Finally we generate a regression.

pyth_lm = lm(formula = 'wpc ~ pyth', data = Eng18)
pyth_lm %>% summary()
```

```
##
## Call:
## lm(formula = "wpc ~ pyth", data = Eng18)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.092318 -0.021430 -0.005752  0.023913  0.103900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.180273   0.009587   18.80   <2e-16 ***
## pyth        0.650177   0.018283   35.56   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03191 on 90 degrees of freedom
## Multiple R-squared:  0.9336, Adjusted R-squared:  0.9328
## F-statistic:  1265 on 1 and 90 DF,  p-value: < 2.2e-16
```

Self test

Run the regression above but instead write 'wpc ~ W' instead of 'wpc ~ pyth' in the line starting pyth_lm. What difference does this make?

Conclusion

Notwithstanding the different organizational structures of soccer, we have found the Pythagorean Expectation model fits the data well.

We have now looked at league results from four different sports and found that the Pythagorean model fits the data well in three of the four.

But we now want to consider a different question: does the Pythagorean model work as a forecasting model? We address this question in the next notebook.