**Novel Techniques For Ranking Of NFL Teams**

**Texas A&M University**

**Department of Industrial & Systems Engineering**

**Fall 2018**

**Prof. Petar Momĉiloviĉ**

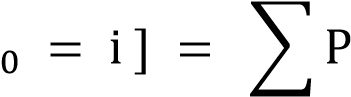
**Problem Statement:**

The National Football League (NFL) is a professional American football league consisting of 32 teams, divided equally between the National Football Conference (NFC) and the American Foot-ball Conference (AFC). Both conferences consist of four four-team divisions. Each team plays 16 regular-season games; thus, teams do not play all other teams during a single regular season. The focus of this assignment is the 2007 NFL regular season. All scores, for that season, can be downloaded from https://www.pro-footballreference.com. The goal of this assignment is to rank the 32 teams after the regular season using an algorithm based on Markov chains. Each team should be represented by a state in a Markov chain, and team ranks should be based on values that are proportional to the stationary probabilities of the constructed Markov chain. Your task is to convert the available game scores into a transition matrix and provide a justification for such a conversion. Note that game scores serve as proxies for relative team strengths. At least two different versions of the matrix should be proposed. Note that one of the teams won all its 16 games. For the constructed matrices, the corresponding stationary distributions should be computed, and team ranks should be produced.

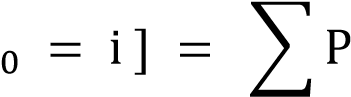
This report contains three approaches to rank NFL (National Football League) teams from “best” to ”worst” using the stationary distribution of a Markov-chain. There are total of 32 teams in the league and each team plays 16 games in a single regular season. The scores of 2007 NFL regular season are taken into consideration to rank the teams. If we take each team as a state, then the ranking of these teams can be done using DTMC.

Let {X0 = i}be the rank of team X initially and we want to find the probability of {Xn = j, n>1}.

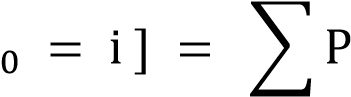
Consider the state-space E made up of all the teams and i, j, k as ranks such that {32 ≥ i, j, k ≥ 1}. Here, the ranking of any team at any time is i.i.d and we can find the ranking of teams considering the probabilities of each team at any time using transition matrix multiplication.

P[ Xn = j / X[ Xn  j , Xn−1 = k / X0 = i ]

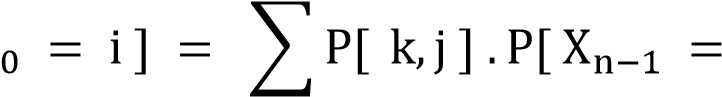
kE

P[ Xn = j / X[ Xn  j / Xn−1 = k , X0 = i ] . P[ Xn−1 = k / X0 = i ]

kE

P[ Xn = j / X[ Xn  j / Xn−1 = k ] . P[ Xn−1 = k / X0 = i ]

kE

P[ Xn = j / X k / X0 = i ]

kE

P[ Xn = j / X0 = i ] = P[ k , j ] . Pn−1[ i , k ]

P[ Xn = j / X0 = i ] = Pn[ i , j ]

In this report, team ranks are based on the values that are proportional to the stationary probabilities of a constructed algorithm. The basic idea of ranking these teams is the transition of points from losing team to winning team and/or winning team to losing team and/or giving points to team by team itself. The final ranking is done based on the number of points each team has. Since, there are two conferences and each team plays only 16 games, there is no team which plays games with every other team. So, the transition of points occurs only between the teams that play together.

|  |
| --- |
| **library**(openxlsx) **library**(markovchain)  ## Package: markovchain  ## Version: 0.6.9.12  ## Date: 2018-08-23  ## BugReport: <http://github.com/spedygiorgio/markovchain/issues>  **options**(max.print=9999)  scores <- **read.xlsx**("C:/Users/Dell/Desktop/609 A1/NFL\_Scores.xlsx") **dim**(scores)  ## [1] 256 4 |

The scores are imported from the Microsoft Excel file- “NFL\_Scores.xlsx” and stored into the “scores” data-frame. The “scores” contains 256 rows and 4 columns.

**head**(scores, 16)

## Winner/tie Loser/tie PtsW PtsL

## 1 Indianapolis Colts New Orleans Saints 41 10

## 2 Carolina Panthers St. Louis Rams 27 13

## 3 Minnesota Vikings Atlanta Falcons 24 3

## 4 Denver Broncos Buffalo Bills 15 14

## 5 Green Bay Packers Philadelphia Eagles 16 13

## 6 Washington Redskins Miami Dolphins 16 13

## 7 Pittsburgh Steelers Cleveland Browns 34 7

## 8 New England Patriots New York Jets 38 14

## 9 Tennessee Titans Jacksonville Jaguars 13 10

## 10 Houston Texans Kansas City Chiefs 20 3

## 11 Detroit Lions Oakland Raiders 36 21

## 12 San Diego Chargers Chicago Bears 14 3

## 13 Seattle Seahawks Tampa Bay Buccaneers 20 6

## 14 Dallas Cowboys New York Giants 45 35

## 15 Cincinnati Bengals Baltimore Ravens 27 20

## 16 San Francisco 49ers Arizona Cardinals 20 17

The first 16 entries of the “scores” can be seen above. As we can see all 16 winning teams and all 16 losing teams are different. We can combine 1-16 winning teams and 1-16 losing teams to make a vector containing all 32 teams. So, the vector – “Teams” contains all the 32 teams and the order of teams in this vector can be seen below. This vector containing all the team names is fed as states to create Markov chain later.

Team1<-scores**$**`Winner/tie`[1**:**16]

Team2<-scores**$**`Loser/tie`[1**:**16]

Teams <- **c**(Team1,Team2)

Teams

## [1] "Indianapolis Colts" "Carolina Panthers" "Minnesota Vikings"

## [4] "Denver Broncos" "Green Bay Packers" "Washington Redskins"

## [7] "Pittsburgh Steelers" "New England Patriots" "Tennessee Titans"

## [10] "Houston Texans" "Detroit Lions" "San Diego Chargers"

## [13] "Seattle Seahawks" "Dallas Cowboys" "Cincinnati Bengals"

## [16] "San Francisco 49ers" "New Orleans Saints" "St. Louis Rams"

## [19] "Atlanta Falcons" "Buffalo Bills" "Philadelphia Eagles"

## [22] "Miami Dolphins" "Cleveland Browns" "New York Jets"

## [25] "Jacksonville Jaguars" "Kansas City Chiefs" "Oakland Raiders"

## [28] "Chicago Bears" "Tampa Bay Buccaneers" "New York Giants"

## [31] "Baltimore Ravens" "Arizona Cardinals"

# **Approach (1)**

The concept behind this approach is that every team would give points proportional to its score to itself and proportional of the opponent team’s score to the opponent team. In this approach if team A beats team B, then there is a high probability of points transitioning from B to A and small probability of points transitioning from A to B. The final ranking would be based on the points each team has at steady state.

If we consider only two teams A and B, we can form the transition matrix M such that, 𝐴 𝐵

𝐴 𝐴𝐴 𝐴𝐵

𝑀 = [ ]

𝐵 𝐵𝐴 𝐵𝐵

Where,

AA = Team A score / Total score BB = Team B score / Total score AB = Team B score / Total score BA = Team A score / Total score

Since, each team is playing 16 games, the addition of all the elements of each row would be 16. So, to normalize this transition matrix, whole transition matrix is divided by 16.

|  |
| --- |
| #Matrix Generation  TransMatrix1 = **matrix**(0, nrow = 32, ncol=32) **for**(i **in** 1**:**32){ **for**(j **in** 1**:**256){  **if**(scores**$**`Winner/tie`[j] **==** Teams[i]){  TransMatrix1[i,i] = TransMatrix1[i,i] **+** (scores**$**PtsW[j]**/**(scores**$**PtsW[j] **+** scores**$**PtsL[j])) |

|  |
| --- |
| index <- **which**(Teams **==** scores**$**`Loser/tie`[j])  TransMatrix1[i,index] = TransMatrix1[i,index] **+** (scores**$**PtsL[j]**/**(scores  **$**PtsW[j] **+** scores**$**PtsL[j]))  }  **if**(scores**$**`Loser/tie`[j] **==** Teams[i]){  TransMatrix1[i,i] = TransMatrix1[i,i] **+** (scores**$**PtsL[j]**/**(scores**$**PtsW[j] **+** scores**$**PtsL[j]))  index <- **which**(Teams **==** scores**$**`Winner/tie`[j])  TransMatrix1[i,index] = TransMatrix1[i,index] **+** (scores**$**PtsW[j]**/**(scores **$**PtsW[j] **+** scores**$**PtsL[j]))  }  }  }  #Normalization  TransMatrix1 = TransMatrix1**/**16    #Solving Transition Matrix  dtmc1 <- **new**("markovchain", states = Teams, transitionMatrix = TransMatrix1, name = "Ranking1") **t**(**steadyStates**(dtmc1))  ## [,1]  ## Indianapolis Colts 0.05105737  ## Carolina Panthers 0.02040019  ## Minnesota Vikings 0.03296356  ## Denver Broncos 0.02372360  ## Green Bay Packers 0.04513173  ## Washington Redskins 0.03803848  ## Pittsburgh Steelers 0.04667978  ## New England Patriots 0.06855865  ## Tennessee Titans 0.03332510  ## Houston Texans 0.03136654  ## Detroit Lions 0.02576864  ## San Diego Chargers 0.04711369  ## Seattle Seahawks 0.03243077  ## Dallas Cowboys 0.04267364  ## Cincinnati Bengals 0.02818755  ## San Francisco 49ers 0.01458686  ## New Orleans Saints 0.02573810  ## St. Louis Rams 0.01496042  ## Atlanta Falcons 0.01633589  ## Buffalo Bills 0.02135179  ## Philadelphia Eagles 0.03554460  ## Miami Dolphins 0.01838538  ## Cleveland Browns 0.03118659 |

|  |
| --- |
| ## New York Jets 0.02259028  ## Jacksonville Jaguars 0.04122230  ## Kansas City Chiefs 0.02209334  ## Oakland Raiders 0.02154983  ## Chicago Bears 0.03026679  ## Tampa Bay Buccaneers 0.03241109  ## New York Giants 0.03602619  ## Baltimore Ravens 0.02318024  ## Arizona Cardinals 0.02515103  #sorting the probabilities and Ranking  R1 <- **steadyStates**(dtmc1)  FinalRank1 <- **data.frame**(R1[1,], **rank**(**-**R1)) **names**(FinalRank1)<-**c**("Probs1","Ranks1") FinalRank1[**order**(FinalRank1**$**Ranks1),]  ## Probs1 Ranks1  ## New England Patriots 0.06855865 1  ## Indianapolis Colts 0.05105737 2  ## San Diego Chargers 0.04711369 3  ## Pittsburgh Steelers 0.04667978 4  ## Green Bay Packers 0.04513173 5  ## Dallas Cowboys 0.04267364 6  ## Jacksonville Jaguars 0.04122230 7  ## Washington Redskins 0.03803848 8  ## New York Giants 0.03602619 9  ## Philadelphia Eagles 0.03554460 10  ## Tennessee Titans 0.03332510 11  ## Minnesota Vikings 0.03296356 12  ## Seattle Seahawks 0.03243077 13  ## Tampa Bay Buccaneers 0.03241109 14  ## Houston Texans 0.03136654 15  ## Cleveland Browns 0.03118659 16  ## Chicago Bears 0.03026679 17  ## Cincinnati Bengals 0.02818755 18  ## Detroit Lions 0.02576864 19  ## New Orleans Saints 0.02573810 20  ## Arizona Cardinals 0.02515103 21  ## Denver Broncos 0.02372360 22  ## Baltimore Ravens 0.02318024 23  ## New York Jets 0.02259028 24  ## Kansas City Chiefs 0.02209334 25  ## Oakland Raiders 0.02154983 26  ## Buffalo Bills 0.02135179 27  ## Carolina Panthers 0.02040019 28  ## Miami Dolphins 0.01838538 29  ## Atlanta Falcons 0.01633589 30  ## St. Louis Rams 0.01496042 31  ## San Francisco 49ers 0.01458686 32 |

# **Approach (2)**

The concept behind this approach is that every losing team would give points to the winning team. We can understand this by the analogy of a bet. Consider there are equal number of bettors betting on each team at the starting of the season and if team A wins against team B then the certain number of bettors betting on team B, will start betting on team A. We can take this number of bettors as the difference of the points made by winning team and the points made by losing team.

If we consider only two teams A and B and if team A wins against team B, we can form the transition matrix M such that,

|  |  |
| --- | --- |
| 𝐴 | 𝐵 |
| 𝐴 𝐴𝐴 𝑀 = [  𝐵 𝐵𝐴 | 𝐴𝐵  ]  𝐵𝐵 |

Where,

AA = 0 BB = 0 AB = 0

BA = Team A score – Team B score

To make a transition matrix we must normalize each row. We can define ( i , j )th entry of TransMatrix2 as,

𝑤𝑖𝑗

# TransMatrix2( i ,j ) = ∑ 𝑤𝑖𝑗

Since, the team “New England Patriots” remains undefeated during the whole season, there is a zero row against the state- New England Patriots. In this case we must convert the zero row into a row with entries equal to 1/n where n is equal to the total number of teams which is 32 in our case. The explanation of this conversion can be that on any match day, there is an equal probability of this team to lose against any team.

## #Matrix Generation

TransMatrix2 = **matrix**(0, nrow = 32, ncol=32) **for**(i **in** 1**:**256){

indexW <- **which**(Teams **==** scores**$**`Winner/tie`[i]) indexL <- **which**(Teams **==** scores**$**`Loser/tie`[i])

|  |
| --- |
| TransMatrix2[indexL,indexW] = TransMatrix2[indexL,indexW] **+** scores**$**PtsW[i]  **-** scores**$**PtsL[i]  }    #Normalization sum = 0 **for**(j **in** 1**:**32){ **for**(k **in** 1**:**32){  sum = sum **+** TransMatrix2[j,k]  }  **if**(sum **!=** 0){  TransMatrix2[j,] = TransMatrix2[j,]**/**sum  } **else**{  rowNumber = j  } sum = 0 }  #Finding the zero row and replacing it. Zero row means undefeated team. **for**(j **in** 1**:**32){  TransMatrix2[rowNumber,j]= 1**/**32  }    #Solving Transition Matrix  dtmc2 <- **new**("markovchain", states = Teams, transitionMatrix = TransMatrix2, name = "Ranking2") **t**(**steadyStates**(dtmc2))  ## [,1]  ## Indianapolis Colts 0.055070372  ## Carolina Panthers 0.016259069  ## Minnesota Vikings 0.037739920  ## Denver Broncos 0.024173069  ## Green Bay Packers 0.050418698  ## Washington Redskins 0.054649405  ## Pittsburgh Steelers 0.038521106  ## New England Patriots 0.138622802  ## Tennessee Titans 0.044721618  ## Houston Texans 0.029301449  ## Detroit Lions 0.026448758  ## San Diego Chargers 0.049489209  ## Seattle Seahawks 0.028063226  ## Dallas Cowboys 0.059674428  ## Cincinnati Bengals 0.027116249  ## San Francisco 49ers 0.009854448  ## New Orleans Saints 0.030577128  ## St. Louis Rams 0.007356663 |

|  |
| --- |
| ## Atlanta Falcons 0.010077405  ## Buffalo Bills 0.010608141  ## Philadelphia Eagles 0.028991175  ## Miami Dolphins 0.004800550  ## Cleveland Browns 0.014462354  ## New York Jets 0.008007175  ## Jacksonville Jaguars 0.040599487  ## Kansas City Chiefs 0.017916438  ## Oakland Raiders 0.007779364  ## Chicago Bears 0.049489844  ## Tampa Bay Buccaneers 0.023703807  ## New York Giants 0.025448843  ## Baltimore Ravens 0.011402292  ## Arizona Cardinals 0.018655510  #Sorting and Ranking  R2 <- **steadyStates**(dtmc2)  FinalRank2 <- **data.frame**(R2[1,], **rank**(**-**R2)) **names**(FinalRank2)<-**c**("Probs2","Ranks2") FinalRank2[**order**(FinalRank2**$**Ranks2), ]  ## Probs2 Ranks2  ## New England Patriots 0.138622802 1  ## Dallas Cowboys 0.059674428 2  ## Indianapolis Colts 0.055070372 3  ## Washington Redskins 0.054649405 4  ## Green Bay Packers 0.050418698 5  ## Chicago Bears 0.049489844 6  ## San Diego Chargers 0.049489209 7  ## Tennessee Titans 0.044721618 8  ## Jacksonville Jaguars 0.040599487 9  ## Pittsburgh Steelers 0.038521106 10  ## Minnesota Vikings 0.037739920 11  ## New Orleans Saints 0.030577128 12  ## Houston Texans 0.029301449 13  ## Philadelphia Eagles 0.028991175 14  ## Seattle Seahawks 0.028063226 15  ## Cincinnati Bengals 0.027116249 16  ## Detroit Lions 0.026448758 17  ## New York Giants 0.025448843 18  ## Denver Broncos 0.024173069 19  ## Tampa Bay Buccaneers 0.023703807 20  ## Arizona Cardinals 0.018655510 21  ## Kansas City Chiefs 0.017916438 22  ## Carolina Panthers 0.016259069 23  ## Cleveland Browns 0.014462354 24  ## Baltimore Ravens 0.011402292 25  ## Buffalo Bills 0.010608141 26  ## Atlanta Falcons 0.010077405 27 |

## San Francisco 49ers 0.009854448 28

## New York Jets 0.008007175 29

## Oakland Raiders 0.007779364 30

## St. Louis Rams 0.007356663 31

## Miami Dolphins 0.004800550 32

## **Approach (3)**

The concept behind this approach is that each team would give points to its opponent according to the score of the opponent. We can understand this approach by the analogy of a bet. Consider there are equal number of people betting on each team at the starting of the season. If team A wins against team B, certain number of bettors previously betting on B would start betting on A and certain number of people previously betting on A would start betting on B depending upon the scores. The number of bettors transitioning from A to B will be less compared to the number of bettors transitioning from B to A.

If we consider only two teams A and B, we can form the transition matrix M such that,

|  |  |
| --- | --- |
| 𝐴 | 𝐵 |
| 𝐴 𝐴𝐴 𝑀 = [  𝐵 𝐵𝐴 | 𝐴𝐵  ]  𝐵𝐵 |

Where,

1. = 0
2. = 0

AB = Team B score

BA = Team A score

To make a transition matrix we must normalize each row. We can define ( i , j )th entry of TransMatrix2 as,

𝑤𝑖𝑗

# TransMatrix2( i ,j ) = ∑ 𝑤𝑖𝑗

## #Matrix Generation

|  |
| --- |
| TransMatrix3 = **matrix**(0, nrow = 32, ncol=32) **for**(i **in** 1**:**256){ indexW <- **which**(Teams **==** scores**$**`Winner/tie`[i]) indexL <- **which**(Teams **==** scores**$**`Loser/tie`[i])  TransMatrix3[indexL,indexW] = TransMatrix3[indexL,indexW] **+** scores**$**PtsW[i] TransMatrix3[indexW,indexL] = TransMatrix3[indexW,indexL] **+** scores**$**PtsL[i] }  #Normalization sum = 0 **for**(j **in** 1**:**32){ **for**(k **in** 1**:**32){ |

|  |
| --- |
| sum = sum **+** TransMatrix3[j,k]  }  **if**(sum **!=** 0){  TransMatrix3[j,] = TransMatrix3[j,]**/**sum  } sum = 0  }  #Solving Transition Matrix  dtmc3 <- **new**("markovchain", states = Teams, transitionMatrix = TransMatrix3, name = "Ranking3") **t**(**steadyStates**(dtmc3))  ## [,1]  ## Indianapolis Colts 0.04113256  ## Carolina Panthers 0.02182697  ## Minnesota Vikings 0.03356159  ## Denver Broncos 0.02991381  ## Green Bay Packers 0.03922541  ## Washington Redskins 0.03458882  ## Pittsburgh Steelers 0.03082364  ## New England Patriots 0.05582468  ## Tennessee Titans 0.02810602  ## Houston Texans 0.03535301  ## Detroit Lions 0.03398943  ## San Diego Chargers 0.03762890  ## Seattle Seahawks 0.02807106  ## Dallas Cowboys 0.04506343  ## Cincinnati Bengals 0.03084870  ## San Francisco 49ers 0.01714713  ## New Orleans Saints 0.03090227  ## St. Louis Rams 0.02128674  ## Atlanta Falcons 0.02171786  ## Buffalo Bills 0.02436825  ## Philadelphia Eagles 0.03496819  ## Miami Dolphins 0.02710426  ## Cleveland Browns 0.03272741  ## New York Jets 0.02533123  ## Jacksonville Jaguars 0.03680030  ## Kansas City Chiefs 0.02253871  ## Oakland Raiders 0.02591833  ## Chicago Bears 0.03323930  ## Tampa Bay Buccaneers 0.02562127  ## New York Giants 0.03819844  ## Baltimore Ravens 0.02698499  ## Arizona Cardinals 0.02918728 |

### #Sorting and Ranking

R3 <- **steadyStates**(dtmc3)

FinalRank3 <- **data.frame**(R3[1,], **rank**(**-**R3)) **names**(FinalRank3)<-**c**("Probs3","Ranks3") FinalRank3[**order**(FinalRank3**$**Ranks3),]

## Probs3 Ranks3

## New England Patriots 0.05582468 1

## Dallas Cowboys 0.04506343 2

## Indianapolis Colts 0.04113256 3

## Green Bay Packers 0.03922541 4

## New York Giants 0.03819844 5

## San Diego Chargers 0.03762890 6

## Jacksonville Jaguars 0.03680030 7

## Houston Texans 0.03535301 8

## Philadelphia Eagles 0.03496819 9

## Washington Redskins 0.03458882 10

## Detroit Lions 0.03398943 11

## Minnesota Vikings 0.03356159 12

## Chicago Bears 0.03323930 13

## Cleveland Browns 0.03272741 14

## New Orleans Saints 0.03090227 15

## Cincinnati Bengals 0.03084870 16

## Pittsburgh Steelers 0.03082364 17

## Denver Broncos 0.02991381 18

## Arizona Cardinals 0.02918728 19

## Tennessee Titans 0.02810602 20

## Seattle Seahawks 0.02807106 21

## Miami Dolphins 0.02710426 22

## Baltimore Ravens 0.02698499 23

## Oakland Raiders 0.02591833 24

## Tampa Bay Buccaneers 0.02562127 25

## New York Jets 0.02533123 26

## Buffalo Bills 0.02436825 27

## Kansas City Chiefs 0.02253871 28

## Carolina Panthers 0.02182697 29

## Atlanta Falcons 0.02171786 30

## St. Louis Rams 0.02128674 31

## San Francisco 49ers 0.01714713 32

# **Results**

We, can compare ranks achieved by all the three approaches and comparison can done below. Any approach can be considered to rank the teams.

compareRanks<- **cbind.data.frame**(Teams,FinalRank1[,2],FinalRank2[,2],FinalRank

3[,2])

**names**(compareRanks)<-**c**("Teams","Approach1","Approach2","Approach3") compareRanks

## Teams Approach1 Approach2 Approach3

## 1 Indianapolis Colts 2 3 3

## 2 Carolina Panthers 28 23 29

## 3 Minnesota Vikings 12 11 12

## 4 Denver Broncos 22 19 18

## 5 Green Bay Packers 5 5 4

## 6 Washington Redskins 8 4 10

## 7 Pittsburgh Steelers 4 10 17

## 8 New England Patriots 1 1 1

## 9 Tennessee Titans 11 8 20

## 10 Houston Texans 15 13 8

## 11 Detroit Lions 19 17 11

## 12 San Diego Chargers 3 7 6

## 13 Seattle Seahawks 13 15 21

## 14 Dallas Cowboys 6 2 2

## 15 Cincinnati Bengals 18 16 16

## 16 San Francisco 49ers 32 28 32

## 17 New Orleans Saints 20 12 15

## 18 St. Louis Rams 31 31 31

## 19 Atlanta Falcons 30 27 30

## 20 Buffalo Bills 27 26 27

## 21 Philadelphia Eagles 10 14 9

## 22 Miami Dolphins 29 32 22

## 23 Cleveland Browns 16 24 14

## 24 New York Jets 24 29 26

## 25 Jacksonville Jaguars 7 9 7

## 26 Kansas City Chiefs 25 22 28

## 27 Oakland Raiders 26 30 24

## 28 Chicago Bears 17 6 13

## 29 Tampa Bay Buccaneers 14 20 25

## 30 New York Giants 9 18 5

## 31 Baltimore Ravens 23 25 23

## 32 Arizona Cardinals 21 21 19