



SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Optimisation of Fantasy Premier League Squad using Linear Programming

April 22, 2023

Introduction

Project Inspiration

Data Analytics is starting to play a pivotal role in many sports teams and competitions as it empowers them to make more accurate decisions by analysing athletes and games better. It provides coaches of professional teams to study historical data on the team's performance, individual players' performance, create unique situations and other strategies for every match.

Additionally, data analytics of competitors' performance enables counter tactics to be devised to succeed against them. Data analytics in sports involves the use of statistics and mathematical principles. Sports analytics helps in improving the performance of a team or the performance of an individual. Data analytics in sports goes beyond traditional statistics to add accurate analysis to improve many aspects of the team and its players. Analytics is used to aid the team in optimising their performance for their games by developing tactics and tailoring solutions accordingly.

In this project, football was analysed as a sport for the purpose of building a fantasy football team and by using the concepts of Integer Programming, the objective was to optimise the best lineup for the fantasy football team, while considering several constraints, decision variables and assumptions.

Problem Statement

Fantasy Premier League (FPL) is a competition in Premier League football, in which fans create their own fantasy lineup of their ideal team according to the players' performances, talent, market value, individual accolades, and other criteria. The individual whose team accumulates the most points over the duration of the competition, depending on the real time performance of the actual players in each game week, will win attractive prizes from the competition organisers.

Participants have a budget of £100 million to select a squad of 15 players, with no more than three players from any one Premier League team. They must select 11 players to start, and 4 substitutes. **Only these 11 starters can contribute to the total points tally, and substitutes are only taken into account when there are players in the starting 11 who do not play in the actual Premier League for that week, and the substitute who takes their place contributes to the total points instead of them.**

Points are awarded based on various factors, including goals, assists, clean sheets, and bonus points, based on the real-time performance of the actual players in each game week.

Our project aims to find out what is the best possible fantasy lineup of 15 players that a busy fantasy premier league enthusiast can possibly create. This means **the fantasy lineup is only selected at the start of the season and left untouched for the whole season.**

Model

Objective Function

The goal for FPL participants is to maximise the total number of points achieved by their fantasy team across the whole season, which is a sum of points accumulated weekly for all weeks in the season. Therefore the objective function is

$$\max \sum_{j=1}^m \sum_{i=1}^n a_{i,j} x_{i,j}$$

where

j represents game week j from 1 to 38 ($m = 38$)

i represents players i from 1 to 741 ($n = 741$)

$a_{i,j}$ represents the points gained by a player i in game week j

Decision Variables

For each player, we can assign a variable $x_{i,j} \in \{0,1\}$ that captures if player i is part of the weekly squad that contributes to points, where

$$x_{i,j} \in \{0, 1\} ; \mathbf{1} \text{ if player } i \text{ is game week } j\text{'s game squad of 11, } \mathbf{0} \text{ otherwise}$$

However, these starting 11 players per game week must be part of the 15 players in the fantasy lineup, hence we introduce a new variable $x2_i \in \{0,1\}$ that captures if a player is in the fantasy lineup

$$x2_i \in \{0, 1\} ; \mathbf{1} \text{ if player } i \text{ is in fantasy lineup, } \mathbf{0} \text{ otherwise}$$

Within the fantasy lineup, there are starters and substitutes too, where starters will have priority to be included in the weekly 11 squad if they actually play that game week, where

$$starter_i \in \{0, 1\} ; \mathbf{1} \text{ if player } i \text{ is a starter in fantasy lineup, } \mathbf{0} \text{ otherwise}$$

If-else variables:

- $\delta_i \in \{0, 1\}$; if player i plays in any week j , he must be included in the fantasy lineup
- $\gamma_i \in \{0, 1\}$; if a starter in the fantasy lineup plays, he must always be included in the weekly 11 team
- $\theta_i \in \{0, 1\}$; if player is a starter, he must be part of the fantasy lineup

Constraints

Lineup constraints:

- 11 players in weekly game squad: $\sum_{i=1}^n x_{i,j} = 11 \forall j$
- 15 players in fantasy lineup: $\sum_{i=1}^n x2_i = 15$
- Player i selected weekly game squad for week j ($x_{i,j} = 1$) is a part of fantasy lineup

$$\circ \sum_{j=1}^m x_{i,j} \leq m(1 - \delta_i) \forall i$$

- $x_{2_i} = 1 - \delta_i \quad \forall i$
- If player i is selected in any weekly game squad, $\sum_{j=1}^m x_{i,j} > 0$, $\delta_i = 0$,
 $x_{2_i} = 1$, else $\sum_{j=1}^m x_{i,j} = 0$, $\delta_i = 1$, $x_{2_i} = 0$

Substitute constraints:

- Only 11 starters out of 15 players in fantasy lineup: $\sum_{i=1}^n starter_i = 15$, which indicates that remaining 4 players are substitutes
- Starter must be part of fantasy lineup
 - $1 - x_{2_i} \leq 1 - \theta_i \quad \forall i$
 - $starter_i \leq \theta_i \quad \forall i$
 - If player i is in fantasy lineup, $x_{2_i} = 1$, $\theta_i = 1$, $starter_i \leq 1$, meaning player i can be a starter. Otherwise, if player i not in fantasy lineup, $x_{2_i} = 0$, and $\theta_i = 0$, so $starter_i = 0$, hence player cannot be a starter
- Substitutes can only play if starters did not play that week
 - $played_{i,j} * starter_i \leq \gamma_{i,j} \quad \forall i, j$
 - $played_{i,j}$ is a data point that records if player i played in week j in the actual Premier League
 - $x_{i,j} \geq \gamma_{i,j} \quad \forall i$
 - If player i played in week j and is a starter, $played_{i,j} * starter_i = 1$, $\gamma_{i,j} = 1$, $x_{i,j} = 1$, meaning player i must be in the weekly game squad. This means substitutes are only considered if starters in the fantasy lineup did not play in week j in the actual Premier League

Position Constraints:

- 15 players in Fantasy Lineup
 - 2 out of the 15 players must be goalkeepers: $\sum_i^n gk_i x_{2_i} = 2$
 - 5 out of the 15 players must be defenders: $\sum_i^n def_i x_{2_i} = 5$
 - 5 out of the 15 players must be midfielders: $\sum_i^n mid_i x_{2_i} = 5$
 - 3 out of the 15 players must be forwards: $\sum_i^n fwd_i x_{2_i} = 3$

- 11 players in weekly game squad and starting squad
 - 1 out of the 11 players must be goalkeepers

$$\blacksquare \sum_i^n gk_i x_{i,j} = 1 \forall j$$

$$\blacksquare \sum_i^n gk_i starter_i = 1$$

- More than 3 players must be defenders

$$\blacksquare \sum_i^n def_i x_{i,j} \geq 3 \forall j$$

$$\blacksquare \sum_i^n def_i starter_i \geq 3$$

- More than 1 player must be a forward

$$\blacksquare \sum_i^n fwd_i x_{i,j} \geq 1 \forall j$$

$$\blacksquare \sum_i^n fwd_i starter_i \geq 1$$

Budget constraint:

- Total fantasy team budget (15 players) is £100 million: $\sum_i^n c_i x_i \leq 1000$

Club constraint:

- A maximum of 3 players from a single football club can be selected for the fantasy

$$\text{team: } \sum_{i=1}^n club_i x_i \leq 3 \forall \text{ all 20 clubs in the Premier League}$$

Assumptions

1. Best possible substitute is selected within constraints
2. The club a player plays for remains constant throughout the season, and is based off the club they are under at the end of the season
3. Player data on points scored, which week they played, position, and club is known and accurate

Results

Correctness of model

To test the correctness of the model, we compared the constraints with respect to the results obtained upon execution of the code. Firstly, in the final fantasy team, it is observed that 15 players form the fantasy team (refer to Appendix, Figure 1.0), following the position constraint of 15 players being selected from the pool of Premier League players.

Next, observing the starting 11 column, it is noted that the model is able to segregate the starting 11 from the overall 15 players selected in the team as evident from the 1 (included in starting 11) versus 0 (not included in starting 11).

After which, the numerical constraint on the various positions of players within the fantasy 15 such as number of goalkeepers, defenders, midfielders and forwards that comprises are kept intact as evident from the “Position” column (refer to Appendix, Figure 1.0). Moreover, this constraint is also applied to the starting 11 of the fantasy team, as well as the 11 players in the weekly game squad (refer to Appendix, Figure 1.0)

Furthermore, to ensure substitutes’ points are only taken into account when the starter does not play in a particular game week, the lineup and substitute constraints were implemented in tandem to achieve optimal results. For example, from Appendix, Figure 1.1, Aaron Ramsdale does not play in game week 1,2, and 3, but plays in week 4 &5. Thus $x_{i,j}$ is 0 from week 1 to 3, and 1 from weeks 4 to 5. Therefore, he was substituted by 1 of the 4 substitutes, in game week 1 to 3, highlighting the correctness of the model. The same can be said about the other starting players.

Lastly, the final constraint that prevents more than 3 players from a single club (refer to Appendix, Figure 1.0, “Team” column) to be included in the fantasy 15 is operating accurately as based on the results of the model, none of the fantasy 15 players break the club constraint. Thus, the model is able to accurately provide an optimised points tally while working within the constraints of FPL.

Results of model

Unfortunately, results were not able to be obtained for the full 38 weeks due to a lack of computing power, hence a smaller subset of the problem with only 5 weeks was computed instead.

The results of the model are attached in the Appendix, with screenshots of the full code attached with the links to the GitHub repository and references to research links that provided us with guidance. The maximum points tally for this smaller problem is 451 (refer to Appendix, Figure 1.2), which is an average of 90.2 points per GW, which is comparable to top players on the leaderboard (refer to Appendix, Figure 1.3).

Appendix

References:

Dataset - <https://github.com/vaastav/Fantasy-Premier-League/tree/master/data/2021-22>

Code guidance -

<https://medium.com/geekculture/linearly-optimising-teams-for-pl-fantasy-league-11931aed18b7>

Leaderboard -

https://fantasy.premierleague.com/leagues/314/standings/c?phase=10&page_new_entries=1&page_standings=1

Constraints - <https://fantasy.premierleague.com/>

xi,1	xi,2	xi,3	xi,4	xi,5	Name	Team	Position	Starting 11	Fantasy lineup
0	0	0	1	1	Aaron Ramsdale	Arsenal	GK	1	1
1	1	1	1	1	Antonio Rüdiger	Chelsea	DEF	1	1
0	1	1	1	0	Aymeric Laporte	Man City	DEF	1	1
0	1	1	1	1	Conor Gallagher	Crystal Palace	MID	1	1
1	0	0	0	1	Emmanuel Dennis	Watford	FWD	0	1
1	1	1	0	0	Hugo Lloris	Spurs	GK	0	1
1	0	0	0	1	Ismaila Sarr	Watford	MID	0	1
1	1	1	1	1	Marcos Alonso	Chelsea	DEF	1	1
1	1	1	1	1	Mason Greenwood	Man Utd	MID	1	1
1	1	1	0	0	Michail Antonio	West Ham	FWD	0	1
1	1	1	1	1	Mohamed Salah	Liverpool	MID	1	1
0	0	0	1	1	Odsonne Edouard	Crystal Palace	FWD	1	1
1	1	1	1	1	Ruben Santos Gato Alves Dias	Man City	DEF	1	1
1	1	1	1	1	Saïd Benrahma	West Ham	MID	1	1
1	1	1	1	0	Trent Alexander-Arnold	Liverpool	DEF	1	1

Figure 1.0 - The fantasy 15 generated upon running the model

Column1	played week 1	played week 2	played week 3	played week 4	played week 5	Starting 11	Fantasy lineup
Aaron Ramsdale	0	0	0	1	1	1	1
Antonio Rüdiger	1	1	1	1	1	1	1
Aymeric Laporte	0	1	1	1	0	1	1
Conor Gallagher	0	1	1	1	1	1	1
Emmanuel Dennis	1	1	1	1	1	0	1
Hugo Lloris	1	1	1	1	1	0	1
Ismaila Sarr	1	1	1	1	1	0	1
Marcos Alonso	1	1	1	1	1	1	1
Mason Greenwood	1	1	1	1	1	1	1
Michail Antonio	1	1	1	1	0	0	1
Mohamed Salah	1	1	1	1	1	1	1
Odsonne Edouard	0	0	0	1	1	1	1
Ruben Santos Gato Alves Dias	1	1	1	1	1	1	1
Saïd Benrahma	1	1	1	1	1	1	1
Trent Alexander-Arnold	1	1	1	1	0	1	1

Figure 1.1 - Aaron Ramsdale does not play in game week 1,2, and 3, but plays in week 4 &5, thus x_{ij} is 0 from week 1 to 3, and 1 from weeks 4 to 5.

```
In [64]: objective_value(model)
```

```
Out[64]: 451.00000000000057
```

Figure 1.2 - Maximum points tally from week 1 to 5

Rank	Team & Manager	GW	TOT
1	Nothing2seeHere Fernando Ambursley	77	404
2	Harris St Germain Sean Harris	68	403
3	Forera ahmed zaki	72	398
4	algeria 23 viking Algérie	92	397
5	Rateta Ball FC Khalid Alawar	92	394
6	Zarra Utd Hassan Zareer	68	394
7	Soccer Savvy Steve Dryden	78	393
8	Schroeder Nicolas Walter	75	393
9	The Soft One Lynne McDonald	77	393

Figure 1.3 - Fantasy football leaderboard

Full working code

Github link to full repository: <https://github.com/yuliangod/Optimisation-1D>

```
In [7]: using JuMP, MosekTools

In [8]: import CSV
import DataFrames

In [11]: csv_df = CSV.read("data.csv", DataFrames.DataFrame)

735 Zack Steffen      7 GK      44 Man City    0 0 0 0 0 0 0 0 0 0
736 Zak Swanson      0 DEF     40 Arsenal    1 0 0 0 0 0 0 0 0 0
737 Álvaro Fernández 31 GK      45 Man Utd    0 0 0 0 0 0 0 0 0 0
738 Álvaro Fernández 31 GK      45 Brentford  0 0 1 0 0 0 0 0 0 0
739 Álvaro Fernández  0 DEF     40 Man Utd    0 0 0 0 0 0 0 0 0 0
740 Álvaro Fernández  0 DEF     40 Brentford  0 0 1 0 0 0 0 0 0 0
741 Çağlar Soyüncü   66 DEF     49 Leicester  0 0 0 0 0 0 0 0 0 0

In [12]: total_points = csv_df[:, "total_points"]
cost = csv_df[:, "now_cost"]

gk = csv_df[:, "GK"]
def = csv_df[:, "DEF"]
fwd = csv_df[:, "FWD"]
mid = csv_df[:, "MID"]

club_df = csv_df[:, 6:6+19]
club = Matrix(club_df)

#print(total_points, fwd, mid)
club_df
```



```
In [12]: total_points = csv_df[:, "total_points"]
cost = csv_df[:, "now_cost"]

gk = csv_df[:, "GK"]
def = csv_df[:, "DEF"]
fwd = csv_df[:, "FWD"]
mid = csv_df[:, "MID"]

club_df = csv_df[:, 6:6+19]
club = Matrix(club_df)

#print(total_points, fwd, mid)
club_df
```

Out[12]: 741x20 DataFrame 716 rows omitted

Row	Arsenal	Aston Villa	Brentford	Brighton	Burnley	Chelsea	Crystal Palace	Everton	Leeds	Leicester	Liverpool	Man City	Man Utd	Newcastle	Norwich	Southampton
	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

```
In [13]: points_df = csv_df[:, 7+19+4:7+19+4+37]
points = Matrix(points_df)
points_df
```

Out[13]: 741x38 DataFrame 716 rows omitted

Row	total_points week 1	total_points week 2	total_points week 3	total_points week 4	total_points week 5	total_points week 6	total_points week 7	total_points week 8	total_points week 9	total_points week 10	total_points week 11	total_pc week 12
	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64
1	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	7.0	2.0	5.0	8.0	1.0	2.0	1.0	7.0	12.0	2.0	1.0	
3	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	6.0	9.0	3.0	6.0	2.0	3.0	10.0	7.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	2.0	2.0	8.0	2.0	2.0	2.0	2.0	-1.0	-1.0	7.0	1.0	
7	8.0	4.0	3.0	10.0	2.0	11.0	5.0	2.0	0.0	0.0	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	6.0	2.0	5.0	2.0	2.0	2.0	2.0	0.0	1.0	5.0	9.0	

```
In [14]: played_df = csv_df[:, 7+19+4+37+1:7+19+4+37*2+1]
played = Matrix(played_df)
played_df
```

Out[14]: 741x38 DataFrame 716 rows omitted

Row	played week 1	played week 2	played week 3	played week 4	played week 5	played week 6	played week 7	played week 8	played week 9	played week 10	played week 11	played week 12	played week 13	played week 14	played week 15	played week 16	played week 17	p v 1
	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64	Float64
1	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	F
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	
4	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	

```
In [15]: model = Model(Mosek.Optimizer)
```

```
Out[15]: A JuMP Model
Feasibility problem with:
Variables: 0
Model mode: AUTOMATIC
CachingOptimizer state: EMPTY_OPTIMIZER
Solver name: Mosek
```

```
In [16]: n = length(cost) # no. of players
m = 38 # no. of game weeks
```

```
Out[16]: 38
```

Initialise variables

```
In [17]: @variable(model, x[1:n, 1:m], Bin)
@variable(model, x2[1:n], Bin)
@variable(model, starter[1:n], Bin)
```

```
Out[17]: 741-element Vector{VariableRef}:
starter[1]
starter[2]
starter[3]
starter[4]
starter[5]
starter[6]
starter[7]
starter[8]
starter[9]
starter[10]
starter[11]
starter[12]
starter[13]
⋮
starter[730]
starter[731]
starter[732]
starter[733]
```

Decision variables

```
In [18]: @variable(model, delta[1:n], Bin)
@variable(model, gamma[1:n, 1:m], Bin)
@variable(model, theta[1:n], Bin)
```

```
Out[18]: 741-element Vector{VariableRef}:
theta[1]
theta[2]
theta[3]
theta[4]
theta[5]
theta[6]
theta[7]
theta[8]
theta[9]
theta[10]
theta[11]
theta[12]
theta[13]
⋮
theta[730]
theta[731]
theta[732]
theta[733]
```

No. of players per week constraint

```
In [19]: @constraint(model, [j=1:m, sum(x[i,j] for i in 1:n) == 11)
```

```
Out[19]: 38-element Vector{ConstraintRef{Model, MathOptInterface.ConstraintIndex{MathOptInterface.ScalarAffineFunction{Float64}, MathOptInterface.EqualTo{Float64}}, ScalarShape}}:
x[1,1] + x[2,1] + x[3,1] + x[4,1] + x[5,1] + x[6,1] + x[7,1] + x[8,1] + x[9,1] + x[10,1] + x[11,1] + x[12,1] + x[13,1] + x[14,1] + x[15,1] + x[16,1] + x[17,1] + x[18,1] + x[19,1] + x[20,1] + x[21,1] + x[22,1] + x[23,1] + x[24,1] + x[25,1] + x[26,1] + x[27,1] + x[28,1] + x[29,1] + x[30,1] + x[31,1] + x[32,1] + x[33,1] + x[34,1] + x[35,1] + x[36,1] + x[37,1] + x[38,1] + x[39,1] + x[40,1] + x[41,1] + x[42,1] + x[43,1] + x[44,1] + x[45,1] + x[46,1] + x[47,1] + x[48,1] + x[49,1] + x[50,1] + x[51,1] + x[52,1] + x[53,1] + x[54,1] + x[55,1] + x[56,1] + x[57,1] + x[58,1] + x[59,1] + x[60,1] + x[61,1] + x[62,1] + x[63,1] + x[64,1] + x[65,1] + x[66,1] + x[67,1] + x[68,1] + x[69,1] + x[70,1] + x[71,1] + x[72,1] + x[73,1] + x[74,1] + x[75,1] + x[76,1] + x[77,1] + x[78,1] + x[79,1] + x[80,1] + x[81,1] + x[82,1] + x[83,1] + x[84,1] + x[85,1] + x[86,1] + x[87,1] + x[88,1] + x[89,1] + x[90,1] + x[91,1] + x[92,1] + x[93,1] + x[94,1] + x[95,1] + x[96,1] + x[97,1] + x[98,1] + x[99,1] + x[100,1] + x[101,1] + x[102,1] + x[103,1] + x[104,1] + x[105,1] + x[106,1] + x[107,1] + x[108,1] + x[109,1] + x[110,1] + x[111,1] + x[112,1] + x[113,1] + x[114,1] + x[115,1] + x[116,1] + x[117,1] + x[118,1] + x[119,1] + x[120,1] + x[121,1] + x[122,1] + x[123,1] + x[124,1] + x[125,1] + x[126,1] + x[127,1] + x[128,1] + x[129,1] + x[130,1] + x[131,1] + x[132,1] + x[133,1] + x[134,1] + x[135,1] + x[136,1] + x[137,1] + x[138,1] + x[139,1] + x[140,1] + x[141,1] + x[142,1] + x[143,1] + x[144,1] + x[145,1] + x[146,1] + x[147,1] + x[148,1] + x[149,1] + x[150,1] + x[151,1] + x[152,1] + x[153,1] + x[154,1] + x[155,1] + x[156,1] + x[157,1] + x[158,1] + x[159,1] + x[160,1] + x[161,1] + x[162,1] + x[163,1] + x[164,1] + x[165,1] + x[166,1] + x[167,1] + x[168,1] + x[169,1]
```

No. of players in fantasy lineup constraint

```
In [20]: @constraint(model, sum(x2[i] for i in 1:n) == 15)
```

```
Out[20]: x2_1 + x2_2 + x2_3 + x2_4 + x2_5 + x2_6 + x2_7 + x2_8 + x2_9 + x2_10 + x2_11 + x2_12 + x2_13 + x2_14 + x2_15 + x2_16 + x2_17 + x2_18 + x2_19 + x2_20
+ x2_21 + x2_22 + x2_23 + x2_24 + x2_25 + x2_26 + x2_27 + x2_28 + x2_29 + x2_30 + x2_31 + x2_32 + x2_33 + x2_34 + x2_35 + x2_36 + x2_37 + x2_38 + x2_39
+ x2_40 + x2_41 + x2_42 + x2_43 + x2_44 + x2_45 + x2_46 + x2_47 + x2_48 + x2_49 + x2_50 + x2_51 + x2_52 + x2_53 + x2_54 + x2_55 + x2_56 + x2_57 + x2_58
+ x2_59 + x2_60 + x2_61 + x2_62 + x2_63 + x2_64 + x2_65 + x2_66 + x2_67 + x2_68 + x2_69 + x2_70 + x2_71 + x2_72 + x2_73 + x2_74 + x2_75 + x2_76 + x2_77
+ x2_78 + x2_79 + x2_80 + x2_81 + x2_82 + x2_83 + x2_84 + x2_85 + x2_86 + x2_87 + x2_88 + x2_89 + x2_90 + x2_91 + x2_92 + x2_93 + x2_94 + x2_95 + x2_96
+ x2_97 + x2_98 + x2_99 + x2_100 + x2_101 + x2_102 + x2_103 + x2_104 + x2_105 + x2_106 + x2_107 + x2_108 + x2_109 + x2_110 + x2_111 + x2_112 + x2_113
+ x2_114 + x2_115 + x2_116 + x2_117 + x2_118 + x2_119 + x2_120 + x2_121 + x2_122 + x2_123 + x2_124 + x2_125 + x2_126 + x2_127 + x2_128 + x2_129 + x2_130
+ x2_131 + x2_132 + x2_133 + x2_134 + x2_135 + x2_136 + x2_137 + x2_138 + x2_139 + x2_140 + x2_141 + x2_142 + x2_143 + x2_144 + x2_145 + x2_146 + x2_147
+ x2_148 + x2_149 + x2_150 + x2_151 + x2_152 + x2_153 + x2_154 + x2_155 + x2_156 + x2_157 + x2_158 + x2_159 + x2_160 + x2_161 + x2_162 + x2_163 + x2_164
+ x2_165 + x2_166 + x2_167 + x2_168 + x2_169 + x2_170 + x2_171 + x2_172 + x2_173 + x2_174 + x2_175 + x2_176 + x2_177 + x2_178 + x2_179 + x2_180 + x2_181
+ x2_182 + x2_183 + x2_184 + x2_185 + x2_186 + x2_187 + x2_188 + x2_189 + x2_190 + x2_191 + x2_192 + x2_193 + x2_194 + x2_195 + x2_196 + x2_197 + x2_198
+ x2_199 + x2_200 + x2_201 + x2_202 + x2_203 + x2_204 + x2_205 + x2_206 + x2_207 + x2_208 + x2_209 + x2_210 + x2_211 + x2_212 + x2_213 + x2_214 + x2_215
+ x2_216 + x2_217 + x2_218 + x2_219 + x2_220 + x2_221 + x2_222 + x2_223 + x2_224 + x2_225 + x2_226 + x2_227 + x2_228 + x2_229 + x2_230 + x2_231 + x2_232
+ x2_233 + x2_234 + x2_235 + x2_236 + x2_237 + x2_238 + x2_239 + x2_240 + x2_241 + x2_242 + x2_243 + x2_244 + x2_245 + x2_246 + x2_247 + x2_248 + x2_249
+ x2_250 + x2_251 + x2_252 + x2_253 + x2_254 + x2_255 + x2_256 + x2_257 + x2_258 + x2_259 + x2_260 + x2_261 + x2_262 + x2_263 + x2_264 + x2_265 + x2_266
```

```
In [21]: # if player i plays in any week j, he is included in fantasy line up with x2[i] = 1
@constraint(model, [i=1:n], sum(x[i,j] for j in 1:m) <= m*(1-delta[i]))
@constraint(model, [i=1:n], x2[i] == 1-delta[i])
```

```
Out[21]: 741-element Vector{ConstraintRef{Model, MathOptInterface.ConstraintIndex{MathOptInterface.ScalarAffineFunction{Float64}}, MathOptInterface.EqualTo{Float64}}, ScalarShape}:
x2[1] + delta[1] == 1.0
x2[2] + delta[2] == 1.0
x2[3] + delta[3] == 1.0
x2[4] + delta[4] == 1.0
x2[5] + delta[5] == 1.0
x2[6] + delta[6] == 1.0
x2[7] + delta[7] == 1.0
x2[8] + delta[8] == 1.0
x2[9] + delta[9] == 1.0
x2[10] + delta[10] == 1.0
x2[11] + delta[11] == 1.0
x2[12] + delta[12] == 1.0
x2[13] + delta[13] == 1.0
⋮
x2[730] + delta[730] == 1.0
x2[731] + delta[731] == 1.0
x2[732] + delta[732] == 1.0
```

11 starters and 4 subs constraint

```
In [22]: @constraint(model, sum(starter[i] for i in 1:n) == 11)
#@constraint(model, sum(x2[i]*(1-starter[i]) for i in 1:n) == 4)
```

```
Out[22]: starter_1 + starter_2 + starter_3 + starter_4 + starter_5 + starter_6 + starter_7 + starter_8 + starter_9 + starter_10 + starter_11 + starter_12 + starter_13
+ starter_14 + starter_15 + starter_16 + starter_17 + starter_18 + starter_19 + starter_20 + starter_21 + starter_22 + starter_23 + starter_24 + starter_25
+ starter_26 + starter_27 + starter_28 + starter_29 + starter_30 + starter_31 + starter_32 + starter_33 + starter_34 + starter_35 + starter_36 + starter_37
+ starter_38 + starter_39 + starter_40 + starter_41 + starter_42 + starter_43 + starter_44 + starter_45 + starter_46 + starter_47 + starter_48 + starter_49
+ starter_50 + starter_51 + starter_52 + starter_53 + starter_54 + starter_55 + starter_56 + starter_57 + starter_58 + starter_59 + starter_60 + starter_61
+ starter_62 + starter_63 + starter_64 + starter_65 + starter_66 + starter_67 + starter_68 + starter_69 + starter_70 + starter_71 + starter_72 + starter_73
+ starter_74 + starter_75 + starter_76 + starter_77 + starter_78 + starter_79 + starter_80 + starter_81 + starter_82 + starter_83 + starter_84 + starter_85
+ starter_86 + starter_87 + starter_88 + starter_89 + starter_90 + starter_91 + starter_92 + starter_93 + starter_94 + starter_95 + starter_96 + starter_97
+ starter_98 + starter_99 + starter_100 + starter_101 + starter_102 + starter_103 + starter_104 + starter_105 + starter_106 + starter_107 + starter_108
+ starter_109 + starter_110 + starter_111 + starter_112 + starter_113 + starter_114 + starter_115 + starter_116 + starter_117 + starter_118 + starter_119
+ starter_120 + starter_121 + starter_122 + starter_123 + starter_124 + starter_125 + starter_126 + starter_127 + starter_128 + starter_129 + starter_130
+ starter_131 + starter_132 + starter_133 + starter_134 + starter_135 + starter_136 + starter_137 + starter_138 + starter_139 + starter_140 + starter_141
+ starter_142 + starter_143 + starter_144 + starter_145 + starter_146 + starter_147 + starter_148 + starter_149 + starter_150 + starter_151 + starter_152
+ starter_153 + starter_154 + starter_155 + starter_156 + starter_157 + starter_158 + starter_159 + starter_160 + starter_161 + starter_162 + starter_163
+ starter_164 + starter_165 + starter_166 + starter_167 + starter_168 + starter_169 + starter_170 + starter_171 + starter_172 + starter_173 + starter_174
```

```
In [23]: # starter must be part of 15 fantasy team
@constraint(model, [i=1:n], 1-x2[i] <= 1-theta[i])
@constraint(model, [i=1:n], starter[i] <= theta[i])
```

```
Out[23]: 741-element Vector{ConstraintRef{Model, MathOptInterface.ConstraintIndex{MathOptInterface.ScalarAffineFunction{Float64}}, MathOptInterface.LessThan{Float64}}, ScalarShape}:
starter[1] - theta[1] <= 0.0
starter[2] - theta[2] <= 0.0
starter[3] - theta[3] <= 0.0
starter[4] - theta[4] <= 0.0
starter[5] - theta[5] <= 0.0
starter[6] - theta[6] <= 0.0
starter[7] - theta[7] <= 0.0
starter[8] - theta[8] <= 0.0
starter[9] - theta[9] <= 0.0
starter[10] - theta[10] <= 0.0
starter[11] - theta[11] <= 0.0
starter[12] - theta[12] <= 0.0
starter[13] - theta[13] <= 0.0
⋮
starter[730] - theta[730] <= 0.0
starter[731] - theta[731] <= 0.0
```

Only play subs if starters did not play that week constraint

```
In [24]: # if starter played, must always include within the lineup
#@constraint(model, [i=1:n, j=1:m], x[i,j] >= played[i,j]*starter[i])
```

```
In [25]: # if starter played, must always include within the lineup
@constraint(model, [i=1:n, j=1:m], played[i,j]*starter[i] <= gamma[i,j])
#@constraint(model, [i=1:n, j=1:m], x[i,j] >= gamma[i,j])
```

```
Out[25]: 741x38 Matrix{ConstraintRef{Model, MathOptInterface.ConstraintIndex{MathOptInterface.ScalarAffineFunction{Float64}, MathOptInterface.GreaterThan{Float64}}, ScalarShape}}:
x[1,1] - gamma[1,1] >= 0.0 ... x[1,38] - gamma[1,38] >= 0.0
x[2,1] - gamma[2,1] >= 0.0 ... x[2,38] - gamma[2,38] >= 0.0
x[3,1] - gamma[3,1] >= 0.0 ... x[3,38] - gamma[3,38] >= 0.0
x[4,1] - gamma[4,1] >= 0.0 ... x[4,38] - gamma[4,38] >= 0.0
x[5,1] - gamma[5,1] >= 0.0 ... x[5,38] - gamma[5,38] >= 0.0
x[6,1] - gamma[6,1] >= 0.0 ... x[6,38] - gamma[6,38] >= 0.0
x[7,1] - gamma[7,1] >= 0.0 ... x[7,38] - gamma[7,38] >= 0.0
x[8,1] - gamma[8,1] >= 0.0 ... x[8,38] - gamma[8,38] >= 0.0
x[9,1] - gamma[9,1] >= 0.0 ... x[9,38] - gamma[9,38] >= 0.0
x[10,1] - gamma[10,1] >= 0.0 ... x[10,38] - gamma[10,38] >= 0.0
x[11,1] - gamma[11,1] >= 0.0 ... x[11,38] - gamma[11,38] >= 0.0
x[12,1] - gamma[12,1] >= 0.0 ... x[12,38] - gamma[12,38] >= 0.0
x[13,1] - gamma[13,1] >= 0.0 ... x[13,38] - gamma[13,38] >= 0.0
⋮
x[730,1] - gamma[730,1] >= 0.0 ... x[730,38] - gamma[730,38] >= 0.0
x[731,1] - gamma[731,1] >= 0.0 ... x[731,38] - gamma[731,38] >= 0.0
x[732,1] - gamma[732,1] >= 0.0 ... x[732,38] - gamma[732,38] >= 0.0
⋮
```

```
In [26]: # if less than 11 starters played that week, num of subs can be more than 0
#@constraint(model, [j=1:m], 11 - sum(x[i,j]*starter[i] for i in 1:n) <= 11*(1-gamma[j]))
#@constraint(model, [j=1:m], sum(x[i,j]*(1-starter[i]) for i in 1:n) <= 4*(1-gamma[j]))
```

Position constraints

15 overall fantasy squad constraint

```
In [27]: @constraint(model, sum(gk[i] * x2[i] for i in 1:n) == 2)
```

```
Out[27]: x2_4 + x2_19 + x2_27 + x2_34 + x2_37 + x2_43 + x2_49 + x2_51 + x2_62 + x2_67 + x2_72 + x2_79 + x2_94 + x2_100 + x2_112 + x2_134 + x2_157 + x2_164 + x2_167
+ x2_168 + x2_169 + x2_172 + x2_193 + x2_196 + x2_198 + x2_199 + x2_207 + x2_226 + x2_230 + x2_236 + x2_237 + x2_254 + x2_256 + x2_267 + x2_275 + x2_283
+ x2_307 + x2_322 + x2_327 + x2_336 + x2_344 + x2_346 + x2_347 + x2_357 + x2_371 + x2_375 + x2_387 + x2_388 + x2_390 + x2_396 + x2_405 + x2_411 + x2_431
+ x2_439 + x2_459 + x2_462 + x2_465 + x2_467 + x2_477 + x2_485 + x2_498 + x2_536 + x2_567 + x2_578 + x2_594 + x2_601 + x2_604 + x2_607 + x2_617 + x2_626
+ x2_632 + x2_640 + x2_675 + x2_679 + x2_688 + x2_703 + x2_705 + x2_706 + x2_712 + x2_718 + x2_724 + x2_735 + x2_737 + x2_738 = 2.0
```

```
In [28]: @constraint(model, sum(def[i] * x2[i] for i in 1:n) == 5)
```

```
Out[28]: x2_2 + x2_6 + x2_13 + x2_14 + x2_21 + x2_25 + x2_28 + x2_41 + x2_44 + x2_45 + x2_50 + x2_56 + x2_61 + x2_66 + x2_68 + x2_69 + x2_73 + x2_75 + x2_76 + x2_77
+ x2_78 + x2_80 + x2_81 + x2_82 + x2_83 + x2_84 + x2_85 + x2_86 + x2_88 + x2_90 + x2_92 + x2_97 + x2_98 + x2_101 + x2_110 + x2_115 + x2_116 + x2_119
+ x2_122 + x2_127 + x2_128 + x2_131 + x2_133 + x2_135 + x2_137 + x2_138 + x2_139 + x2_143 + x2_144 + x2_145 + x2_146 + x2_150 + x2_151 + x2_153 + x2_156
+ x2_163 + x2_170 + x2_177 + x2_179 + x2_181 + x2_185 + x2_190 + x2_192 + x2_202 + x2_203 + x2_210 + x2_211 + x2_213 + x2_214 + x2_215 + x2_217
+ x2_219 + x2_222 + x2_224 + x2_227 + x2_231 + x2_234 + x2_239 + x2_240 + x2_242 + x2_246 + x2_252 + x2_255 + x2_262 + x2_263 + x2_266 + x2_268 + x2_269
+ x2_273 + x2_281 + x2_289 + x2_292 + x2_297 + x2_298 + x2_299 + x2_300 + x2_305 + x2_308 + x2_309 + x2_314 + x2_316 + x2_317 + x2_318 + x2_319 + x2_320
+ x2_323 + x2_330 + x2_334 + x2_337 + x2_339 + x2_340 + x2_345 + x2_348 + x2_350 + x2_352 + x2_353 + x2_359 + x2_365 + x2_369 + x2_370 + x2_373 + x2_376
+ x2_379 + x2_384 + x2_398 + x2_400 + x2_402 + x2_403 + x2_406 + x2_407 + x2_408 + x2_409 + x2_410 + x2_412 + x2_414 + x2_415 + x2_416 + x2_419 + x2_424
+ x2_427 + x2_433 + x2_440 + x2_442 + x2_443 + x2_444 + x2_445 + x2_446 + x2_448 + x2_450 + x2_453 + x2_454 + x2_456 + x2_457 + x2_461 + x2_468 + x2_472
+ x2_479 + x2_481 + x2_482 + x2_483 + x2_484 + x2_488 + x2_489 + x2_491 + x2_492 + x2_495 + x2_497 + x2_502 + x2_511 + x2_522 + x2_524 + x2_525 + x2_526
+ x2_531 + x2_532 + x2_534 + x2_537 + x2_539 + x2_542 + x2_543 + x2_544 + x2_547 + x2_554 + x2_555 + x2_557 + x2_561 + x2_564 + x2_569 + x2_573 + x2_575
+ x2_579 + x2_583 + x2_587 + x2_588 + x2_590 + x2_591 + x2_592 + x2_595 + x2_596 + x2_598 + x2_600 + x2_602 + x2_606 + x2_611 + x2_612 + x2_618 + x2_619
+ x2_621 + x2_623 + x2_625 + x2_630 + x2_634 + x2_642 + x2_643 + x2_645 + x2_647 + x2_649 + x2_661 + x2_665 + x2_668 + x2_672 + x2_673 + x2_681 + x2_682
+ x2_683 + x2_684 + x2_693 + x2_694 + x2_695 + x2_701 + x2_702 + x2_704 + x2_707 + x2_708 + x2_710 + x2_711 + x2_713 + x2_719 + x2_721 + x2_723 + x2_725
+ x2_728 + x2_729 + x2_730 + x2_734 + x2_736 + x2_739 + x2_740 + x2_741 = 5.0
```

```
In [29]: @constraint(model, sum(mid[i] * x2[i] for i in 1:n) == 5)
```

```
Out[29]: x2_3 + x2_5 + x2_7 + x2_10 + x2_12 + x2_15 + x2_17 + x2_18 + x2_20 + x2_22 + x2_23 + x2_24 + x2_26 + x2_29 + x2_31 + x2_32 + x2_33 + x2_35 + x2_38 + x2_42
+ x2_46 + x2_47 + x2_48 + x2_52 + x2_53 + x2_54 + x2_57 + x2_58 + x2_60 + x2_65 + x2_70 + x2_71 + x2_74 + x2_87 + x2_89 + x2_91 + x2_93 + x2_95 + x2_96
+ x2_99 + x2_102 + x2_103 + x2_104 + x2_105 + x2_106 + x2_107 + x2_108 + x2_113 + x2_117 + x2_118 + x2_120 + x2_123 + x2_126 + x2_129 + x2_130 + x2_132
+ x2_136 + x2_140 + x2_141 + x2_142 + x2_148 + x2_149 + x2_152 + x2_154 + x2_158 + x2_159 + x2_160 + x2_166 + x2_171 + x2_173 + x2_174 + x2_175 + x2_176
+ x2_178 + x2_180 + x2_183 + x2_186 + x2_187 + x2_189 + x2_191 + x2_195 + x2_200 + x2_205 + x2_206 + x2_209 + x2_212 + x2_218 + x2_220 + x2_223 + x2_225
```

```
In [29]: @constraint(model, sum(mid[i] * x2[i] for i in 1:n) == 5)
Out[29]: x2_3 + x2_5 + x2_7 + x2_10 + x2_12 + x2_15 + x2_17 + x2_18 + x2_20 + x2_22 + x2_23 + x2_24 + x2_26 + x2_29 + x2_31 + x2_32 + x2_33 + x2_35 + x2_38 + x2_42
+ x2_46 + x2_47 + x2_48 + x2_52 + x2_53 + x2_54 + x2_57 + x2_58 + x2_60 + x2_65 + x2_70 + x2_71 + x2_74 + x2_87 + x2_89 + x2_91 + x2_93 + x2_95 + x2_96
+ x2_99 + x2_102 + x2_103 + x2_104 + x2_105 + x2_106 + x2_107 + x2_108 + x2_113 + x2_117 + x2_118 + x2_120 + x2_123 + x2_126 + x2_129 + x2_130 + x2_132
+ x2_136 + x2_140 + x2_141 + x2_142 + x2_148 + x2_149 + x2_152 + x2_154 + x2_158 + x2_159 + x2_160 + x2_166 + x2_171 + x2_173 + x2_174 + x2_175 + x2_176
+ x2_178 + x2_180 + x2_183 + x2_186 + x2_187 + x2_189 + x2_191 + x2_195 + x2_200 + x2_205 + x2_206 + x2_209 + x2_212 + x2_218 + x2_220 + x2_223 + x2_225
+ x2_232 + x2_233 + x2_235 + x2_238 + x2_243 + x2_244 + x2_245 + x2_247 + x2_248 + x2_250 + x2_251 + x2_257 + x2_258 + x2_259 + x2_260 + x2_261 + x2_265
+ x2_270 + x2_271 + x2_272 + x2_274 + x2_276 + x2_277 + x2_278 + x2_280 + x2_284 + x2_285 + x2_287 + x2_288 + x2_290 + x2_291 + x2_293 + x2_294 + x2_295
+ x2_296 + x2_301 + x2_302 + x2_303 + x2_304 + x2_306 + x2_310 + x2_311 + x2_312 + x2_315 + x2_321 + x2_325 + x2_328 + x2_329 + x2_331 + x2_332 + x2_333
+ x2_338 + x2_342 + x2_343 + x2_349 + x2_351 + x2_355 + x2_358 + x2_360 + x2_362 + x2_363 + x2_364 + x2_366 + x2_372 + x2_378 + x2_380 + x2_382 + x2_383
+ x2_385 + x2_386 + x2_389 + x2_394 + x2_395 + x2_397 + x2_399 + x2_401 + x2_404 + x2_413 + x2_417 + x2_418 + x2_420 + x2_421 + x2_422 + x2_425 + x2_429
+ x2_430 + x2_432 + x2_434 + x2_435 + x2_436 + x2_437 + x2_438 + x2_441 + x2_447 + x2_449 + x2_451 + x2_452 + x2_455 + x2_458 + x2_460 + x2_464 + x2_466
+ x2_469 + x2_470 + x2_471 + x2_473 + x2_475 + x2_476 + x2_478 + x2_480 + x2_486 + x2_487 + x2_490 + x2_493 + x2_494 + x2_500 + x2_504 + x2_505 + x2_507
+ x2_508 + x2_509 + x2_510 + x2_513 + x2_514 + x2_515 + x2_516 + x2_517 + x2_518 + x2_519 + x2_520 + x2_521 + x2_527 + x2_528 + x2_530 + x2_535 + x2_538
+ x2_540 + x2_541 + x2_546 + x2_548 + x2_551 + x2_552 + x2_553 + x2_556 + x2_558 + x2_559 + x2_560 + x2_562 + x2_563 + x2_570 + x2_571 + x2_572 + x2_574
+ x2_576 + x2_577 + x2_580 + x2_582 + x2_584 + x2_585 + x2_586 + x2_593 + x2_599 + x2_608 + x2_609 + x2_614 + x2_615 + x2_616 + x2_620 + x2_622 + x2_624
+ x2_627 + x2_628 + x2_633 + x2_635 + x2_636 + x2_637 + x2_638 + x2_639 + x2_641 + x2_644 + x2_646 + x2_648 + x2_652 + x2_653 + x2_654 + x2_655 + x2_656
+ x2_658 + x2_659 + x2_660 + x2_662 + x2_664 + x2_666 + x2_667 + x2_670 + x2_671 + x2_676 + x2_678 + x2_685 + x2_686 + x2_687 + x2_690 + x2_691
+ x2_692 + x2_697 + x2_698 + x2_699 + x2_700 + x2_709 + x2_715 + x2_716 + x2_717 + x2_720 + x2_722 + x2_727 + x2_731 + x2_732 + x2_733 = 5.0
```

```
In [30]: @constraint(model, sum(fwd[i] * x2[i] for i in 1:n) == 3)
Out[30]: x2_1 + x2_8 + x2_9 + x2_11 + x2_16 + x2_30 + x2_36 + x2_39 + x2_40 + x2_55 + x2_59 + x2_63 + x2_64 + x2_109 + x2_111 + x2_114 + x2_121 + x2_124 + x2_125
+ x2_147 + x2_155 + x2_161 + x2_162 + x2_165 + x2_182 + x2_184 + x2_188 + x2_194 + x2_197 + x2_201 + x2_208 + x2_216 + x2_221 + x2_228 + x2_229 + x2_241
+ x2_249 + x2_253 + x2_264 + x2_279 + x2_282 + x2_286 + x2_313 + x2_324 + x2_326 + x2_335 + x2_341 + x2_354 + x2_356 + x2_361 + x2_367 + x2_368 + x2_374
+ x2_377 + x2_381 + x2_391 + x2_392 + x2_393 + x2_423 + x2_426 + x2_428 + x2_463 + x2_474 + x2_496 + x2_499 + x2_501 + x2_503 + x2_506 + x2_512 + x2_523
+ x2_529 + x2_533 + x2_545 + x2_549 + x2_550 + x2_565 + x2_566 + x2_568 + x2_581 + x2_589 + x2_597 + x2_603 + x2_605 + x2_610 + x2_613 + x2_629 + x2_631
+ x2_650 + x2_651 + x2_657 + x2_663 + x2_669 + x2_674 + x2_677 + x2_680 + x2_696 + x2_714 + x2_726 = 3.0
```

11 game week constraint and starter constraint

```
In [31]: @constraint(model, [j=1:m], sum(gk[i]*x[i,j] for i in 1:n) == 1)
@constraint(model, sum(gk[i]*starter[i] for i in 1:n) == 1)
Out[31]: starter_4 + starter_19 + starter_27 + starter_34 + starter_37 + starter_43 + starter_49 + starter_51 + starter_62 + starter_67 + starter_72 + starter_79
+ starter_94 + starter_100 + starter_112 + starter_134 + starter_157 + starter_164 + starter_167 + starter_168 + starter_169 + starter_172 + starter_193
+ starter_196 + starter_198 + starter_199 + starter_207 + starter_226 + starter_230 + starter_236 + starter_237 + starter_254 + starter_256 + starter_267
```

```
In [32]: @constraint(model, [j=1:m], sum(def[i]*x[i,j] for i in 1:n) >= 3)
@constraint(model, sum(def[i]*starter[i] for i in 1:n) >= 3)
Out[32]: starter_2 + starter_6 + starter_13 + starter_14 + starter_21 + starter_25 + starter_28 + starter_41 + starter_44 + starter_45 + starter_50 + starter_56
+ starter_61 + starter_66 + starter_68 + starter_69 + starter_73 + starter_75 + starter_76 + starter_77 + starter_78 + starter_80 + starter_81 + starter_82
+ starter_83 + starter_84 + starter_85 + starter_86 + starter_88 + starter_90 + starter_92 + starter_97 + starter_98 + starter_101 + starter_110 + starter_115
+ starter_116 + starter_119 + starter_122 + starter_127 + starter_128 + starter_131 + starter_133 + starter_135 + starter_137 + starter_138 + starter_139
+ starter_143 + starter_144 + starter_145 + starter_146 + starter_150 + starter_151 + starter_153 + starter_156 + starter_170 + starter_177
+ starter_179 + starter_181 + starter_185 + starter_190 + starter_192 + starter_202 + starter_203 + starter_204 + starter_210 + starter_211 + starter_213
+ starter_214 + starter_215 + starter_217 + starter_219 + starter_222 + starter_224 + starter_227 + starter_231 + starter_234 + starter_239 + starter_240
+ starter_242 + starter_246 + starter_252 + starter_255 + starter_262 + starter_263 + starter_266 + starter_268 + starter_273 + starter_281
+ starter_289 + starter_292 + starter_297 + starter_298 + starter_299 + starter_300 + starter_305 + starter_308 + starter_309 + starter_314 + starter_316
+ starter_317 + starter_318 + starter_319 + starter_320 + starter_323 + starter_330 + starter_334 + starter_337 + starter_339 + starter_340 + starter_345
+ starter_348 + starter_350 + starter_352 + starter_353 + starter_359 + starter_365 + starter_369 + starter_370 + starter_373 + starter_376 + starter_379
+ starter_384 + starter_398 + starter_400 + starter_402 + starter_403 + starter_406 + starter_407 + starter_408 + starter_409 + starter_410 + starter_412
+ starter_414 + starter_415 + starter_416 + starter_419 + starter_424 + starter_427 + starter_433 + starter_440 + starter_442 + starter_443 + starter_444
+ starter_445 + starter_446 + starter_448 + starter_450 + starter_453 + starter_454 + starter_456 + starter_457 + starter_461 + starter_468 + starter_472
+ starter_479 + starter_481 + starter_482 + starter_483 + starter_484 + starter_488 + starter_489 + starter_491 + starter_492 + starter_495 + starter_497
+ starter_502 + starter_511 + starter_522 + starter_524 + starter_525 + starter_526 + starter_531 + starter_532 + starter_534 + starter_537 + starter_539
+ starter_542 + starter_543 + starter_544 + starter_547 + starter_554 + starter_555 + starter_557 + starter_561 + starter_564 + starter_569 + starter_573
+ starter_575 + starter_579 + starter_583 + starter_587 + starter_588 + starter_590 + starter_591 + starter_592 + starter_595 + starter_596 + starter_598
+ starter_600 + starter_602 + starter_606 + starter_611 + starter_612 + starter_618 + starter_619 + starter_621 + starter_623 + starter_625 + starter_630
+ starter_634 + starter_642 + starter_643 + starter_645 + starter_647 + starter_649 + starter_661 + starter_665 + starter_668 + starter_672 + starter_673
+ starter_681 + starter_682 + starter_683 + starter_684 + starter_693 + starter_694 + starter_695 + starter_701 + starter_702 + starter_704 + starter_707
+ starter_708 + starter_710 + starter_711 + starter_713 + starter_719 + starter_721 + starter_723 + starter_725 + starter_728 + starter_729 + starter_730
+ starter_734 + starter_736 + starter_739 + starter_740 + starter_741 >= 3.0
```

```
In [33]: @constraint(model, [j=1:m], sum(fwd[i]*x[i,j] for i in 1:n) >= 1)
@constraint(model, sum(fwd[i]*starter[i] for i in 1:n) >= 1)
Out[33]: starter_1 + starter_8 + starter_9 + starter_11 + starter_16 + starter_30 + starter_36 + starter_39 + starter_40 + starter_55 + starter_59 + starter_63
+ starter_64 + starter_109 + starter_111 + starter_114 + starter_121 + starter_124 + starter_125 + starter_147 + starter_155 + starter_161 + starter_162
+ starter_165 + starter_182 + starter_184 + starter_188 + starter_194 + starter_197 + starter_201 + starter_208 + starter_216 + starter_221 + starter_228
+ starter_229 + starter_241 + starter_249 + starter_253 + starter_264 + starter_279 + starter_282 + starter_286 + starter_313 + starter_324 + starter_326
+ starter_335 + starter_341 + starter_354 + starter_356 + starter_361 + starter_367 + starter_368 + starter_374 + starter_377 + starter_381 + starter_391
+ starter_392 + starter_393 + starter_423 + starter_426 + starter_428 + starter_463 + starter_474 + starter_496 + starter_499 + starter_501 + starter_503
+ starter_506 + starter_512 + starter_523 + starter_529 + starter_533 + starter_545 + starter_549 + starter_550 + starter_565 + starter_566 + starter_568
+ starter_581 + starter_589 + starter_597 + starter_603 + starter_605 + starter_610 + starter_613 + starter_629 + starter_631 + starter_650 + starter_651
+ starter_657 + starter_663 + starter_669 + starter_674 + starter_677 + starter_680 + starter_696 + starter_714 + starter_726 >= 1.0
```

Cost constraint

```
In [34]: @constraint(model, sum(cost[i] * x2[i] for i in 1:n) <= 1000)
```

```
Out[34]: 53x2_1 + 54x2_2 + 46x2_3 + 51x2_4 + 44x2_5 + 51x2_6 + 53x2_7 + 45x2_8 + 57x2_9 + 45x2_10 + 48x2_11 + 53x2_12 + 43x2_13 + 43x2_14 + 56x2_15 + 45x2_16 + 59x2_17 + 45x2_18 + 43x2_19 + 52x2_20 + 40x2_21 + 48x2_22 + 45x2_23 + 58x2_24 + 40x2_25 + 44x2_26 + 43x2_27 + 49x2_28 + 58x2_29 + 81x2_30 + 51x2_31 + 45x2_32 + 55x2_33 + 61x2_34 + 43x2_35 + 67x2_36 + 47x2_37 + 50x2_38 + 45x2_39 + 54x2_40 + 46x2_41 + 50x2_42 + 39x2_43 + 39x2_44 + 73x2_45 + 50x2_46 + 52x2_47 + 49x2_48 + 39x2_49 + 49x2_50 + 40x2_51 + 48x2_52 + 47x2_53 + 45x2_54 + 76x2_55 + 61x2_56 + 56x2_57 + 45x2_58 + 53x2_59 + 44x2_60 + 44x2_61 + 39x2_62 + 52x2_63 + 48x2_64 + 52x2_65 + 47x2_66 + 40x2_67 + 44x2_68 + 60x2_69 + 45x2_70 + 57x2_71 + 45x2_72 + 39x2_73 + 62x2_74 + 40x2_75 + 40x2_76 + 45x2_77 + 45x2_78 + 42x2_79 + 44x2_80 + 47x2_81 + 39x2_82 + 46x2_83 + 40x2_84 + 46x2_85 + 45x2_86 + 45x2_87 + 56x2_88 + 45x2_89 + 52x2_90 + 55x2_91 + 40x2_92 + 70x2_93 + 45x2_94 + 57x2_95 + 43x2_96 + 45x2_97 + 39x2_98 + 48x2_99 + 40x2_100 + 39x2_101 + 45x2_102 + 50x2_103 + 116x2_104 + 57x2_105 + 52x2_106 + 67x2_107 + 53x2_108 + 71x2_109 + 42x2_110 + 45x2_111 + 43x2_112 + 45x2_113 + 50x2_114 + 40x2_115 + 39x2_116 + 45x2_117 + 45x2_118 + 44x2_119 + 45x2_120 + 65x2_121 + 45x2_122 + 44x2_123 + 63x2_124 + 59x2_125 + 56x2_126 + 43x2_127 + 40x2_128 + 50x2_129 + 79x2_130 + 44x2_131 + 48x2_132 + 44x2_133 + 40x2_134 + 39x2_135 + 45x2_136 + 44x2_137 + 40x2_138 + 47x2_139 + 45x2_140 + 60x2_141 + 49x2_142 + 40x2_143 + 42x2_144 + 49x2_145 + 48x2_146 + 122x2_147 + 45x2_148 + 50x2_149 + 42x2_150 + 56x2_151 + 44x2_152 + 44x2_153 + 47x2_154 + 45x2_155 + 40x2_156 + 40x2_157 + 55x2_158 + 45x2_159 + 60x2_160 + 45x2_161 + 77x2_162 + 42x2_163 + 40x2_164 + 60x2_165 + 45x2_166 + 43x2_167 + 44x2_168 + 50x2_169 + 42x2_170 + 45x2_171 + 47x2_172 + 50x2_173 + 63x2_174 + 54x2_175 + 49x2_176 + 45x2_177 + 45x2_178 + 44x2_179 + 83x2_180 + 40x2_181 + 48x2_182 + 45x2_183 + 77x2_184 + 44x2_185 + 57x2_186 + 43x2_187 + 52x2_188 + 55x2_189 + 40x2_190 + 59x2_191 + 40x2_192 + 62x2_193 + 84x2_194 + 45x2_195 + 61x2_196 + 57x2_197 + 39x2_198 + 40x2_199 + 45x2_200 + 45x2_201 + 46x2_202 + 45x2_203
```

No. of players from same club constraint

```
In [35]: @constraint(model, [c=1:20], sum(club[i,c] * x2[i] for i in 1:n) <= 3)
```

```
Out[35]: 20-element Vector{ConstraintRef{Model, MathOptInterface.ConstraintIndex{MathOptInterface.ScalarAffineFunction{Float64}}, MathOptInterface.LessThan{Float64}}, ScalarShape}:
 x2[4] + x2[20] + x2[22] + x2[25] + x2[30] + x2[62] + x2[85] + x2[94] + x2[107] + x2[117] + x2[150] + x2[197] + x2[205] + x2[229] + x2[242] + x2[243] + x2[245] + x2[268] + x2[388] + x2[402] + x2[406] + x2[435] + x2[458] + x2[469] + x2[490] + x2[495] + x2[505] + x2[506] + x2[509] + x2[538] + x2[542] + x2[551] + x2[561] + x2[581] + x2[593] + x2[600] + x2[618] + x2[626] + x2[628] + x2[642] + x2[661] + x2[676] + x2[719] + x2[722] + x2[734] + x2[736] <= 3.0
 x2[5] + x2[66] + x2[89] + x2[95] + x2[97] + x2[110] + x2[111] + x2[113] + x2[142] + x2[162] + x2[187] + x2[206] + x2[207] + x2[217] + x2[226] + x2[240] + x2[291] + x2[293] + x2[327] + x2[343] + x2[365] + x2[384] + x2[392] + x2[408] + x2[416] + x2[420] + x2[433] + x2[449] + x2[470] + x2[484] + x2[496] + x2[515] + x2[550] + x2[577] + x2[607] + x2[678] + x2[690] + x2[702] + x2[705] + x2[714] <= 3.0
 x2[106] + x2[116] + x2[126] + x2[129] + x2[168] + x2[185] + x2[199] + x2[215] + x2[227] + x2[235] + x2[249] + x2[282] + x2[315] + x2[347] + x2[379] + x2[410] + x2[446] + x2[447] + x2[448] + x2[463] + x2[478] + x2[479] + x2[485] + x2[494] + x2[518] + x2[529] + x2[562] + x2[567] + x2[572] + x2[583] + x2[598] + x2[633] + x2[646] + x2[648] + x2[666] + x2[709] + x2[731] + x2[738] + x2[740] <= 3.0
 x2[1] + x2[12] + x2[14] + x2[31] + x2[33] + x2[39] + x2[92] + x2[165] + x2[171] + x2[192] + x2[209] + x2[216] + x2[228] + x2[263] + x2[296] + x2[322] + x2[331] + x2[376] + x2[381] + x2[405] + x2[418] + x2[424] + x2[453] + x2[455] + x2[477] + x2[502] + x2[513] + x2[533] + x2[544] + x2[563] + x2[604] + x2[640] + x2[653] + x2[655] + x2[665] + x2[667] + x2[675] + x2[733] <= 3.0
```

```
In [36]: @objective(model, Max, sum(points[i,j] * x[i,j] for i in 1:n, j in 1:m))
```

```
Out[36]: [Math Processing Error]
```

```
In [37]: print(model)
```

```
[Math Processing Error]
```

```
In [ ]: optimize!(model)
```

```
Problem
  Name           :
  Objective sense : maximize
  Type           : LO (linear optimization problem)
  Constraints     : 118742
  Affine conic cons. : 0
  Disjunctive cons. : 0
  Cones          : 0
  Scalar variables : 59280
  Matrix variables : 0
  Integer variables : 59280

Optimizer started.
Mixed integer optimizer started.
Threads used: 4
Presolve started.
Presolve terminated. Time = 0.91, probing time = 0.11
Presolved problem: 29640 variables, 11351 constraints, 98466 non-zeros
Presolved problem: 0 general integer, 29640 binary, 0 continuous
```

```
In [*]: starting_11 = value.(x2)
starting_11
```

```
In [*]: objective_value(model)
```

```
In [*]: csv_df[!, "Starting 11"] = value.(starter)
csv_df[!, "Fantasy lineup"] = value.(x2)
```

```
In [*]: csv_df
```

```
In [*]: CSV.write("results.csv", csv_df)
```

```
In [*]: using DataFrames
```

```
In [*]: xij_df = DataFrame(value.(x), :auto)
```

```
In [*]: xij_df[!, "Name"] = csv_df[1:n, "Column1"]
xij_df[!, "Team"] = csv_df[1:n, "team"]
xij_df[!, "Position"] = csv_df[1:n, "element_type"]
xij_df
```

```
In [*]: starting_11 = value.(x2)
starting_11
```

```
In [*]: objective_value(model)
```

```
In [*]: csv_df[!, "Starting 11"] = value.(starter)
csv_df[!, "Fantasy lineup"] = value.(x2)
```

```
In [*]: csv_df
```

```
In [*]: CSV.write("results.csv", csv_df)
```

```
In [*]: using DataFrames
```

```
In [*]: xij_df = DataFrame(value.(x), :auto)
```

```
In [*]: xij_df[!, "Name"] = csv_df[1:n, "Column1"]
xij_df[!, "Team"] = csv_df[1:n, "team"]
xij_df[!, "Position"] = csv_df[1:n, "element_type"]
xij_df
```

```
In [*]: xij_df[!, "Starting 11"] = value.(starter)
xij_df[!, "Fantasy lineup"] = value.(x2)
xij_df
```

```
In [*]: CSV.write("Xij.csv", xij_df)
```