2022 STUDENT RESEARCH CASE STUDY CHALLENGE: FOOTBALL/SOCCER

THE ESTEEMED GECKOS

AARON VU AIMON MOSTOFI JAMES NGO LIAM LA NATHAN TRUONG

UNIVERSITY OF NEW SOUTH WALES

TABLE OF CONTENTS

OBJECTIVE OF ANALYSIS

The objective of this report is to create a competitive national football team for Rarita to build a brand and create positive economic opportunities. A team is defined as competitive by Commissioner Bayes if they:

1. Rank within the top 10 members of the FSA for the season within the next 5 years with,
2. High probability of achieving an FSA championship within the next 10 years.

An implementation plan on the creation of the team will be explored as well as the economic impacts and risks of the team on the Raritan economy.

TEAM SELECTION

<u>Criteria for Selection</u>

Football teams have different formations for positioning themselves on the pitch. The most common and effective formation was determined to be the 1-4-3-3 formation consisting of 1 goalkeeper, 4 defenders, 3 midfielders and 3 forwards (Aquino et al. 2017). The Raritan football team was selected based on this formation including a substitute for each role. The competitive football team comprises 15 of the top Raritan born players that have participated in a league in 2021. They were chosen based on specific attributes that are most important to their role.

These attributes were chosen through exploring the correlation between each statistic and the rank from the teams that participated in the 2020 and 2021 FSA tournament. Statistics most correlated with rank were used as variables to build a model and the statistic that most embodied the definition of the role was chosen as the dependent variable. The dependent variables chosen were:

- Goalkeepers - Percentage of saves per shot on target (Saves%)
- Defenders - Successful pressure percentage (Pressure%)
- Midfielders - Pass completion percentage (Completion%)
- Forwards - Number of goals (Goals)

A combination of different statistics between shooting, passing, defense and goalkeeping were chosen as independent variables based on both correlation and importance for each role. The independent variables chosen for each role are in *Appendix 1*.

Multi-linear regression, K-nearest neighbours, bagging, random forest and boosting were the models chosen to model the players. A combination of the FSA tournament and league data was utilised to test and train the models. For each role, the model with the lowest mean squared error and highest R squared was used to predict the Raritan players' performance (see *Appendix 2*). The players were then ranked and the top players were chosen for the team. The players chosen were under age 30 in 2021 and their salaries were also considered to ensure economic viability.

Probability of Competitiveness

The 2021 FSA tournament data and rankings were used to train a bagging model with 5000 trees to forecast the probabilities of the team becoming competitive within the next 10 years. A team average for each of the 4 variables, Saves%, Pressure%, Completion% and Goals was used to train the model. Two separate bagging models were trained, one whose output was the teams rank (Rank Model) and the other a probability of the team ranking top 10 in the next season (TT Model). To account for changes in player skill from age and experience, each player in the FSA tournament and the selected Raritan team was exposed to a scaled increase/decrease in their Saves%, Pressure%, Completion% or Goals based on their age each year (Mujika et al. 2009). For certain age ranges, a player has a chance to either increase or decrease their score by a value between 0-10%. A uniform distribution was used to determine whether their value increased or decreased and the percentage their value would change by. *Table 1* showcases the age ranges, the probabilities of their value changing and the percentage change.

Age	<= 24	24 - 27	27 - 32	32 - 35	35 <=
Chance to Increase/Decrease Value	90% chance to **increase** by 0% - 10%	75% chance to **increase** by 0% - 10%	100% chance to **stay the same**	75% chance to **decrease** by 0% - 10%	90% chance to **decrease** by 0% - 10%

Table 1 - Probability of Value Changes

Using the forecasts from the TT Model, 5000 simulations were averaged to determine the probability of the Raritan football team ranking at least top 10 in the next 5 years.

Year	2022	2023	2024	2025	2026
Average Probability of Top 10	0.8729	0.8827	0.8834	0.8832	0.8816
Probability Range	0.8384 - 0.9048	0.8384 - 0.9048	0.8384 - 0.9048	0.8210 - 0.9112	0.7474 - 0.9130

Table 2 – Annual Top Ten Probabilities

It is evident in *Table 2* that over the next 5 years, the selected Raritan team has an average 88% chance of ranking in the top 10 which is extremely high. Using the forecasts from the Rank Model, 5000 simulations were run to determine the probability of the Raritan team ranking 1st in the FSA tournament in the next 10 years.

Year	2022	2023	2024	2025	2026
Probability of Placing 1st	0.2048	0.4910	0.6206	0.7042	0.8008

Year	2027	2028	2029	2030	2031
Probability of Placing 1st	0.8928	0.9462	0.9768	0.9882	0.9884

Table 3 – Annual Championship Probabilities

From *Table 3*, it showcases that Rarita has a high chance of ranking 1st in the tournament, especially in 2028 - 2031. As the selected Raritan team is forecasted to have a high chance of ranking top 10 in the next 5 years and high chance of claiming a FSA championship in the next 10 years, the selected team can be considered competitive.

Funding the Team

Over the next 10 years, the players are signed to long-term 5-year contracts in which their salaries will increase based on their performance during those years as long-term contracts can improve player productivity (Carmichael, F 2011). As all players are chosen from Rarita, there is no need to borrow players from other countries, resulting in less expenses. The combined yearly salary to fund the players are represented in *Table 4*.

Year	Salary (∂)	Year	Salary (∂)
2022	92 340 000	**2027**	104 827 470
2023	92 340 000	**2028**	104 827 470
2024	92 340 000	**2029**	104 827 470
2025	92 340 000	**2030**	104 827 470
2026	92 340 000	**2031**	104 827 470

Table 4 - Combined Annual Salaries

As the total salary across the 10 years is far less than the ∂995,000,000 provided by the government, there is no need for non-government funding. Other expenses such as funding for staff will be expanded upon in the Economic Impact.

Direct Team Revenues

Rarita’s competitive team will directly boost revenues through increased matchday, broadcasting and commercial sources such as increased ticket sales and merchandising. This revenue will increase as the team becomes more competitive due to higher support from fans both in Rarita and internationally. *Table 5* highlights the forecast of the direct revenues over the next 10 years.

Year	Revenue (∂)	Year	Revenue (∂)
2022	84 340 456	**2027**	165 137 758
2023	100 238 462	**2028**	181 689 400
2024	116 267 196	**2029**	198 371 769
2025	132 426 656	**2030**	215 184 865
2026	148 716 844	**2031**	231 128 688

Table 5 - Annual Revenues

ECONOMIC IMPACT

For this analysis the chosen team is assumed to satisfy the “competitive” definition over the next 10 years. Forecasts of GDP, GNI and relevant league revenues/expenses have been provided, allowing an assessment of the team’s impact on the Raritan economy.

Additional revenue can be found above in *Table 5*, and additional expenses are shown in *Table 6*. Expenses are expected to increase in later years as the team becomes more competitive and more money is invested into staff and training facilities.

Year	Expenses (∂)
2022	71 577 826
2023	77 591 022
2024	83 652 798
2025	89 762 615
2026	95 920 473

Year	Expenses (∂)
2027	102 126 372
2028	108 380 311
2029	114 682 291
2030	121 032 312
2031	127 430 373

Table 6 - Annual Expenses

Rarita’s historical football league data was used to forecast revenues and expenses, producing two ETS models with and without the team. The difference between revenue minus expenses in these two models represents additional profit resulting from the national team, found in *Table 7* alongside total profits for each year. This data shows that in the later years of the forecast the additional profit accounts for up to 15% of total football profit, indicating a significant financial benefit as a result of continued international success.

Year	Total Profit (∂)	Additional Profit (∂)	% of Total
2022	427 465 532	12 763 170	2.99%
2023	453 516 194	22 647 440	4.99%
2024	479 770 791	32 614 397	7.00%
2025	506 229 323	42 664 041	8.43%
2026	532 891 789	52 796 370	9.91%

Year	Total Profit (∂)	Additional Profit (∂)	% of Total
2027	559 758 191	63 011 386	11.26%
2028	586 828 528	73 309 089	12.49%
2029	614 102 800	83 689 478	13.63%
2030	641 581 007	94 152 553	14.68%
2031	669 293 149	104 698 314	15.64%

Table 7 - Annual Profits

ARIMA models were produced using historical data for both GDP and GNI and forecasts for 2021-2031 were produced. Given the historical data doesn't account for the national team, the additional profit was added to GDP and GNI to capture the effect of the team on these economic indices. These forecasts are seen in *Figure 1 and 2* and highlight the positive impact of the proposed implementation plan on the economy.

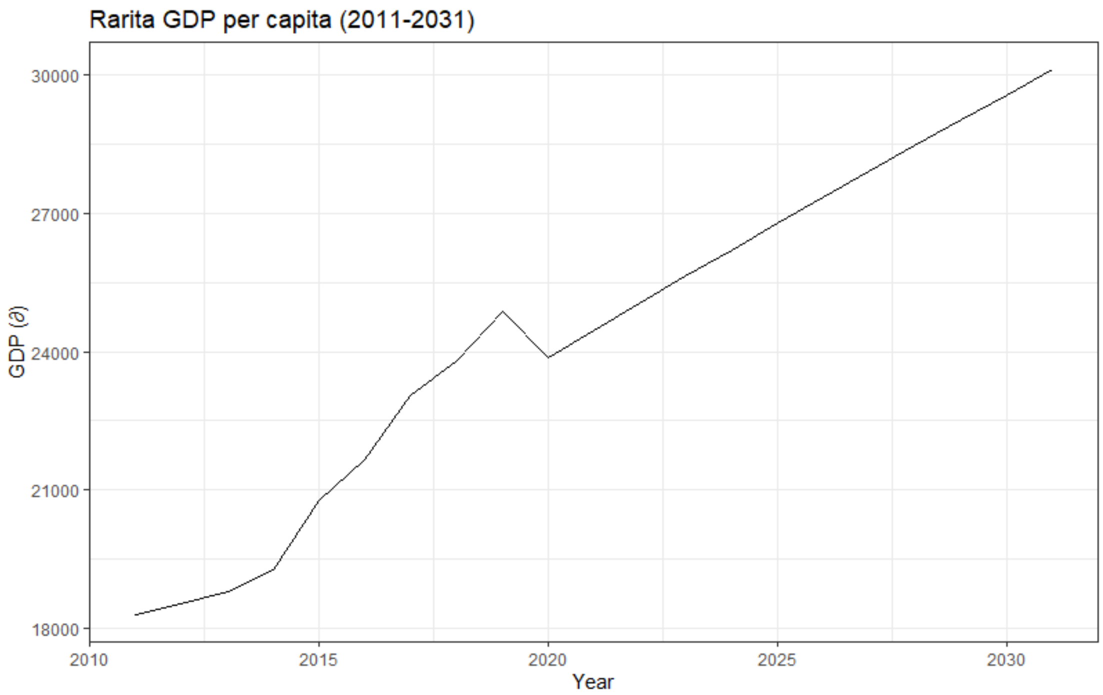


Figure 1 - GDP Forecast

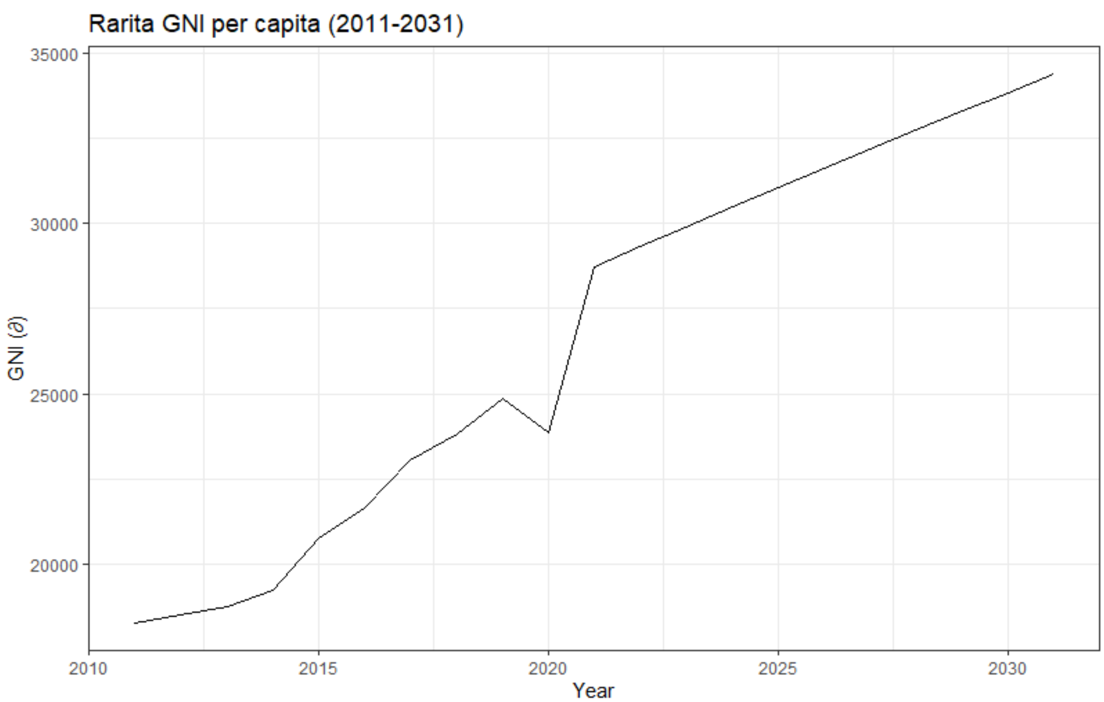


Figure 2 - GNI Forecast

The above additional profit calculations don’t account for the initial one-time sum of 995,000,000 or player salaries. To account for these factors, each year player salaries were subtracted from additional profit generated and the net result was taken from the remaining funding available. This remaining funding was then invested at the forecasted 1-year spot rate for the given year to determine remaining funding in the following year. *Table 8* showcases these results and highlights the increase in remaining funding seen in 2031, a positive indicator for subsequent years.

Year	Remaining Funding (∂)
2022	919 888 128
2023	861 621 841
2024	812 680 079
2025	773 264 941
2026	743 588 341

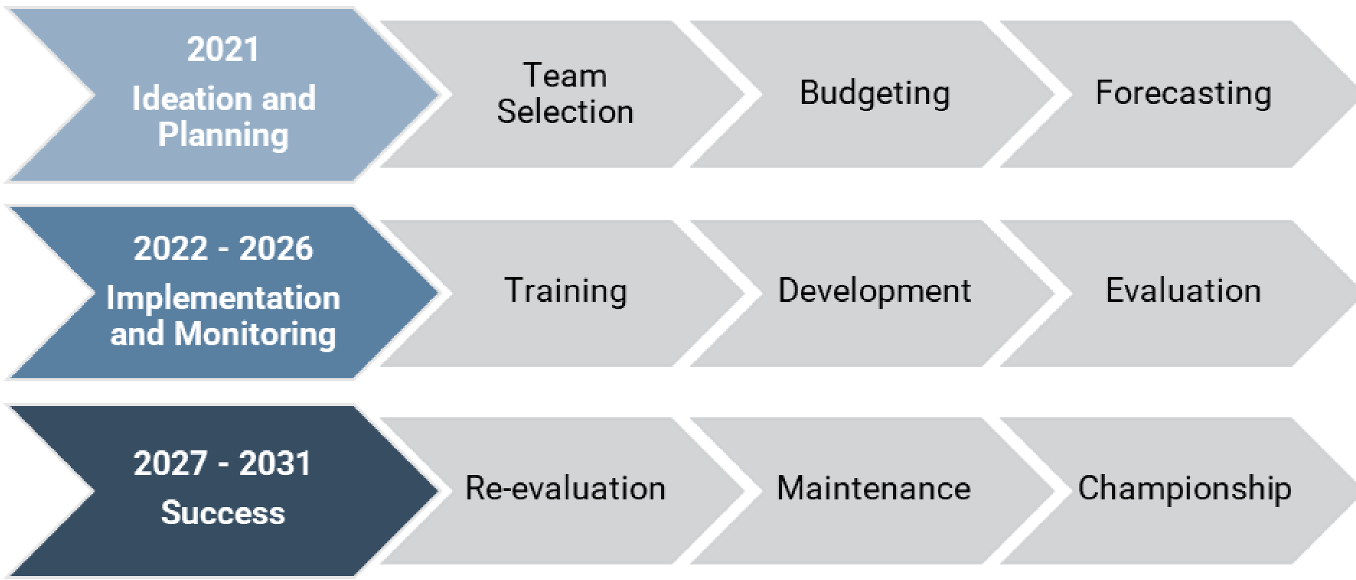
Year	Remaining Funding (∂)
2027	711 209 738
2028	688 831 694
2029	676 672 796
2030	674 954 169
2031	683 900 010

Table 8 - Yearly Remaining Funding

As stated by Rarita, the success of the international team increases global visibility, allowing for increased investment and tourism opportunities. Furthermore, the prize awarded to the FSA champion and the increase in league attendance resulting from team success are factors that were unable to be included in the additional profit calculation, however these factors are expected to have a positive impact on overall football profit. In regard to the impact on Rarita provinces, the increase in GDP shown in Figure 1 indicates the Raritan government would have additional funding that could be invested into the relatively poorer province of West Rarita.

IMPLEMENTATION PLAN

This section aims to implement a strategy that creates a competitive football team to reap the benefits of an internationally successful team. This report's key objectives necessitate a medium-term strategy, meaning success will need to be achieved in the short and long term. The implementation timeline is broken down into three phases, as seen below.



Phase 1 involves selecting the team per skill and budgeting considerations. As aforementioned in the Team Selection, the forecasted probability of placing in the top ten starts high. However, the chance of winning the championship begins low and increases rapidly over several years. This is achieved through a combination of young and prime-aged players, allowing for short-term success and a long-term championship. The selected players are below in *Table 9*.

Name	Year of Birth	Position	2022-2026 Salary (∂)	2027-2031 Salary (∂)
C. Tukamushaba	1995	Defender	970 000	1 082 580
E. Mudzingwa	1994	Forward	16 240 000	18 124 848
F. Ajio	1991	Forward	1 990 000	2 309 802
F. Akumu	2000	Goalkeeper	5 600 000	6 624 944
F. Chin	1997	Midfield	1 340 000	1 555 344
F. Ithungu	1992	Goalkeeper	1 530 000	1 741 726

Name	Year of Birth	Position	2022-2026 Salary (∂)	2027-2031 Salary (∂)
F. Yunusa	1994	Defender	18 150 000	20 256 527
H. Makumbi	1993	Forward	7 430 000	8 458 188
H. Zare	1991	Defender	15 430 000	17 909 671
N. Terzi	1998	Defender	5 000 000	5 803 523
O. Wanjala	1996	Midfield	1 750 000	1 992 171
Q. bin Ismail	1996	Defender	9 20 000	1 047 313
S. Barman	1995	Midfield	8 680 000	9 687 419
X. Leroy	1994	Midfield	5 630 000	6 283 429
X. Thomas	1991	Forward	1 680 000	1 949 984

Table 9 - Selected Players

The players will be offered five-year contracts with a chance to be renewed at the end of 2026. Upon retention, the player will receive a higher inflation-adjusted salary and a bonus based on performance. Without additional data, an estimated team selection in five years is unlikely to be realistic and forecasts were completed with the original 15 players.

As mentioned in the Economic Impact, the team has budgeted to maintain positive net profits throughout the ten-year plan. The revenues forecasted are driven by various sources such as matchday ticket sales. Broadcasting revenues are forecasted to increase as the international reputation of the league benefits from a competitive team. Commercial revenues will also increase from higher merchandise sales and new sponsorships. GDP is expected to increase as the team becomes competitive. Annual forecasts may be found in the previous section.

The implementation and monitoring of the short-term plan will be the focus of Phase 2. Players will undergo rigorous training and development programs. These programs will develop the team's skills, strategy and culture. The minimum requirement is placing top ten in the championship. Progress will further be monitored annually through three metrics. Forecasted tournament rank, net profits and GDP will be compared with actual data. If progress is underwhelming, revaluations of the team and strategy will be considered.

After the second phase, there will be a five-year re-evaluation to adapt the plan to changing player and financial performances. This involves renewing contracts and potentially selecting new players. The final phase covers the last five years where a championship is the target. Phase 3 will continue to implement development programs and focus on maintaining current success and pushing for the championship.

ASSUMPTIONS

Key Assumptions

1. **The best players now will be the best players in the next ten years**
   This allows us to select the players that the model deems best given the available data.

2. **All teams in the FSA tournament remain the same including Rarita, no players get replaced**
   We assume that all the teams remain the same following the next ten years due to the ambiguity of many factors that may occur over the next ten years (e.g. new players coming in and being selected, current players may become injured, current players may retire).

3. **Each player in Rarita has a 5-year contract which increases with performance**
   The rationale for this assumption is that salary increases should be tied in with performance and that players are trying to sign longer term contracts.

4. **Negative statistics were treated as correct and the statistics were scaled between 0 and 1**
   By scaling the statistics to between 0 and 1, it allows the players to be ranked against each other on a relative scale with more meaningful interpretations.

Other Assumptions

- Population, GDP, GNI and spot rate forecasts follow an ARIMA model.
- Revenue, expense and inflation forecasts follow an ETS model.
- Team synergy doesn't affect their playing ability.
- Each player is offered a 5-year contract.
- A player's age affects their playing ability.
- A competitive national football team will increase the GDP.

RISK AND RISK MITIGATION CONSIDERATIONS

Risk Analysis

Risk	Quantifiable or Qualitative	Likelihood of Risk	Mitigation Techniques
Players selected sustain major injury/are unable to continue playing in the next 10 years	Quantifiable	Low (4-5%) (Leena et al. 2012)	● Ensure regular and safe warmup and exercise routines for players ● Conduct regular preseason physicals ● Educate players on risks and dangers of sporting activities ● Strict enforcing of proper equipment and gear to be worn while playing (shin guards, boots) ● Have additional players ready to substitute
Players behave in a manner that negatively affects public image and reputation	Qualitative	Low	● Develop a player code of conduct ● Educate players thoroughly on the code of conduct ● Instill a strong culture of good behaviour in players
Citizens of Rarita oppose large monetary allocations to the football team and become hostile to the Raritan government and football related events	Qualitative	Low	● Host regular charity football matches to give back to key groups in Rarita ● Donate sporting equipment to local schools and junior clubs ● Develop strong football programs for diversified groups in Rarita ● Volunteering in local community events

Sensitivity Analysis on Total Player Salary Expense per Year (∂)

		Contract Duration (Years)			
		3	4	5	6
Salary Increase	1%	1 006 604 253	998 303 671	981 411 605	978 910 847
	1.5%	1 009 276 560	1 001 144 494	983 624 478	980 719 898
	2%	1 011 956 220	1 003 985 954	985 837 350	982 528 949
	2.5%	1 014 643 232	1 006 828 051	988 050 222	984 338 000
	3%	1 017 337 596	1 009 670 785	990 263 094	986 147 051

Table 10 - Salary Sensitivity Analysis

A sensitivity analysis was conducted on the total player salary expenses, where the two assumptions;

1. Salary increases percentage and
2. Contract duration

were varied in order to observe changes in the total player salary expense. In *Table 10*, our assumption used is highlighted in green, with a 5-year contract on a 2% salary increase based on performance.

Recommended ranges for these assumptions are highlighted in blue, with 3-6 year contract durations on 1.5%-3% salary increases, based on the two desirable outcomes of less total salary expenses (an acceptable range of ∂980 000 000 to ∂1 010 000 000), reasonable salary increases (greater than 1% as a 1% salary increase is unrealistic) and the assumption of players opting for longer term contracts.

DATA AND DATA LIMITATIONS

The nature of data implies the existence of missing and outlier values. To manipulate the provided data into workable formats, some assumptions and limitations on analysis were required.

- **Tournament Data Missing 2020 Values**
  The absence of 2020 data for tournament passing and goalkeeping meant the probability models were trained solely through 2021. This resulted in a slightly weaker analysis as the data size was reduced.
- **Tournament, League, Revenue and Expense Data NA Values**
  This investigation dealt with NA values by omitting them from the analysis. This resulted in data loss however was mitigated by selecting relevant columns first. This enabled the investigations to omit fewer values.
- **Tournament and League Data Negative Values**
  The data contained negative values which had meaningless real-life interpretations. As these values represented a significant portion of the total dataset, they were adjusted. The adjustment involved scaling the values between a range of 0 to 1 to provide a parameter based on relative performance.
- **Limited Historic Data**
  The limited years of data for economic and player data limited the accuracy of models. This was specifically prevalent for social media where only a single year was provided.

The scope of this investigation only required the data sources provided. These being “economic”, “football” and “player”.

REFERENCES

Aquino, R, Vieira, L, Carling, C, Martins, G, Alves, I & Puggina, E 2017, ‘Effects of competitive standard, team formation and playing position on match running performance of Brazil professional soccer players’, *International Journal of Performance Analysis in Sport*, accessed 14 March 2022 from Taylor & Francis Online 2021

Carmichael, F, Rossi, G & Simmons, R 2011, ‘Contract duration and player performance in Serie A’, accessed 14 March 2022, <https://www.sheffield.ac.uk/polopoly_fs/1.679571!/file/F3_1_Simmons.pdf>

Leena, R, Kettunen, J, Kujala, U and Heinonen, A 2012, ‘Sport injuries as the main cause of sport career termination among Finnish top-level athletes’, *European Journal of Sport Science*, accessed 14 March 2022 from ResearchGate database

Mujika, I, Vaeyens, R, Matthys, S, Santisteban, J, Goiriena, J & Philippaerts, R 2009, ‘The relative age effect in professional football club setting’, *Journal of Sports Sciences*, accessed 14 March 2022 from Taylor & Francis Online 2021

APPENDIX

Appendix 1 - Independent Predictor Variables and their Correlation with Rank

Independent Predictor Variables

Goalkeepers: Age, Playing Time MP, Performance GA90, Performance SoTA, W, L, Performance CS%, Penalty Kicks PKsv, Penalty Kicks Save%.

Defenders: Age, Tackles Tkl, Vs Dribbles Past, Pressures Def 3rd, Pressures Press, Blocks Blocks, Tkl+Int, Total Cmp%, 1/3, Prog.

Midfielders: Age, 90s, Total Cmp, Medium Att, 1/3, Total TotDist, Tackles Mid 3rd, Prog, Total PrgDist, Blocks Pass.

Forwards: Age, 90s, Standard Sh, Standard FK, Total Cmp, KP, Total Att, xA, Expected xG, Gls.

Position	Predictors	Correlation With Rank
Goalkeepers	Age	0.13
	Playing Time MP	-0.69
	Performance GA90	0.66
	Performance SoTA	0.65
	W	-0.81
	L	0.58
	Performance CS%	-0.56
	Penalty Kicks PKsv	0.37
	Penalty Kicks Save%	0.28
Defenders	Age	-0.10
	Tackles Tkl	0.21
	Vs Dribbles Past	0.16
	Pressures Def 3rd	0.36

	Pressures Press	0.35
	Blocks Blocks	0.15
	Tkl+Int	0.17
	Total Cmp%	-0.34
	1/3	-0.29
	Prog	-0.18
Midfielders	Age	-0.01
	90s	-0.44
	Total Cmp	-0.31
	Medium Att	-0.31
	1/3	-0.30
	Total TotDist	-0.29
	Tackles Mid 3rd	-0.27
	Prog	-0.26
	Total PrgDist	-0.20
	Blocks Pass	-0.13
Forwards	Age	0.21
	90s	-0.30
	Standard Sh	-0.16
	Standard FK	-0.22
	Total Cmp	-0.17
	KP	-0.16
	Total Att	-0.15
	xA	-0.14
	Expected xG	-0.13
	Gls	-0.10

Appendix 2 - MSE and R Square Results from the Models Used to Choose the Best Players

Position	Model	MSE	R Square
Goalkeepers	Multi-linear Regression	0.007275711	0.5507444
	K-Nearest Neighbours	0.004998123	0.7290712
	Bagging	0.007080357	0.5908597
	Random Forest	0.007319475	0.5770460
	Boosting	0.009113703	0.4429678
Defenders	Multi-linear Regression	0.005752477	0.11062766
	K-Nearest Neighbours	0.006058334	0.07471167
	Bagging	0.006335879	0.07147108
	Random Forest	0.005986248	0.09114180
	Boosting	0.006331765	0.11040315
Midfielders	Multi-linear Regression	0.00005803288	0.9914495
	K-Nearest Neighbours	0.00013630598	0.9795390
	Bagging	0.00008030216	0.9879938
	Random Forest	0.00008036564	0.9879725
	Boosting	0.00013228586	0.9805513
Forwards	Multi-linear Regression	0.0008249397	0.4811610
	K-Nearest Neighbours	0.0014446711	0.2622476
	Bagging	0.0009745583	0.4216689
	Random Forest	0.0009426277	0.4284739
	Boosting	0.0013975958	0.1560272

Appendix 3 - Economic and Population Forecasts

Year	GDP per Capita (∂)
2022	25 044.49
2023	25 628.06
2024	26 206.75
2025	26 780.61
2026	27 349.72

Year	GDP per Capita (∂)
2027	27 914.12
2028	28 473.88
2029	29 029.06
2030	29 579.70
2031	30 125.87

Year	GNI per Capita (∂)
2022	29 315.77
2023	29 896.51
2024	30 472.40
2025	31 043.49
2026	31 609.84

Year	GNI per Capita (∂)
2027	32 171.51
2028	32 728.56
2029	33 281.05
2030	33 829.03
2031	34 372.56

Year	Population
2022	12 676 549
2023	12 730 087
2024	12 783 626
2025	12 837 164
2026	12 890 703

Year	Population
2027	12 944 241
2028	12 997 780
2029	13 051 318
2030	13 104 856
2031	13 158 395

Appendix 4 - Economic Rate Forecasts

Year	Spot Rate
2022	0.0049
2023	0.0134
2024	0.0134
2025	0.0134
2026	0.0134

Year	Spot Rate
2027	0.0134
2028	0.0134
2029	0.0134
2030	0.0134
2031	0.0134

Year	Inflation Rate
2022	0.224
2023	0.0223
2024	0.0222
2025	0.0221
2026	0.0220

Year	Inflation Rate
2027	0.0219
2028	0.0218
2029	0.0217
2030	0.0217
2031	0.0216

Appendix 5 - R Code

```
# AUTHORS: Aaron Vu, Aimon Mostofi, James Ngo, Liam La, Nathan Truong
#
# Table of Contents
#
#   1. Libraries
#   2. Directories
#   3. Data
#   4. Functions
#   5. Goalkeeper Selection
#   6. Defenders Selection
#   7. Midfielder Selection
#   8. Forwards Selection
#   9. Probability
#   10. Economic Forecasts
#   11. Salary
#
#
#
#
#
#


### Libraries ### ----------------------------------------------------

library(readxl)
library(dplyr)
library(tidyr)
library(reshape2)
library(ggplot2)
library(randomForest)
library(gbm)
library(future)
library(caret)
library(fpp3)
library(future.apply)
library(forecast)
library(class)
library(visdat)
```

END ### ---

Directories ### --

ecoDirectory <- paste(dirname(rstudioapi::getSourceEditorContext()$path),'/2022-student-research-case-study-economic-data.xlsx',sep="")

footballDirectory <- paste(dirname(rstudioapi::getSourceEditorContext()$path),'/2022-student-research-case-study-football-soccer-data.xlsx',sep="")

playerdataDirectory <- paste(dirname(rstudioapi::getSourceEditorContext()$path),'/2022-student-research-case-study-player-data.xlsx',sep="")

END ### ---

Data ### --

Naming convention: directory + sheetname

Economic Data

ecoRarSpot <- read_excel(ecoDirectory, sheet = 'Rarita Spot Rates', range = 'B12:P72')

ecoRarInflation <- read_excel(ecoDirectory, sheet = 'Rarita Inflation Rates', range = 'B11:C41')

ecoRarGDP <- read_excel(ecoDirectory, sheet = 'Rarita Economic', range = 'B12:F22')

ecoRarGNI <- read_excel(ecoDirectory, sheet = 'Rarita Economic', range = 'H12:L22')

ecoRarPopulation <- read_excel(ecoDirectory, sheet = 'Rarita Economic', range = 'B26:F36')

ecoRarPopDensity <- read_excel(ecoDirectory, sheet = 'Rarita Economic', range = 'H26:L36')

ecoRarHealthSpend <- read_excel(ecoDirectory, sheet = 'Rarita Economic', range = 'B40:F40')

ecoRarHouseholdSave <- read_excel(ecoDirectory, sheet = 'Rarita Economic', range = 'H40:L40')

ecoRarCurrency <- read_excel(ecoDirectory, sheet = 'Rarita Economic', range = 'B54:D59')

ecoGDP <- read_excel(ecoDirectory, sheet = 'Other Countries GDP', range = 'B12:G32')

```
# Football Data

footballRevenueRaw <- read_excel(footballDirectory, sheet = 'Revenue', range = 'B13:V34')

footballTotalRevenue <- footballRevenueRaw[,c(1,2,6,10,14,18)]
colnames(footballTotalRevenue) <- c('Nation', '2020', '2019', '2018', '2017', '2016')

footballMatchday <- footballRevenueRaw[,c(1,3,7,11,15,19)]
colnames(footballMatchday) <- c('Nation', '2020', '2019', '2018', '2017', '2016')

footballBroadcast <- footballRevenueRaw[,c(1,4,8,12,16,20)]
colnames(footballBroadcast) <- c('Nation', '2020', '2019', '2018', '2017', '2016')

footballCommericial <- footballRevenueRaw[,c(1,5,9,13,17,21)]
colnames(footballCommericial) <- c('Nation', '2020', '2019', '2018', '2017', '2016')


footballExpenseRaw <- read_excel(footballDirectory, sheet = 'Expense', range = 'B13:Q34')

footballTotalExpense <- footballExpenseRaw[,c(1,2,5,8,11,14)]
colnames(footballTotalExpense) <- c('Nation', '2020', '2019', '2018', '2017', '2016')
footballTotalExpense <- footballTotalExpense %>% filter(Nation != 'Eastern Sleboube')

footballStaffCosts <- footballExpenseRaw[,c(1,3,6,9,12,15)]
colnames(footballStaffCosts) <- c('Nation', '2020', '2019', '2018', '2017', '2016')
footballStaffCosts <- footballStaffCosts %>% filter(Nation != 'Eastern Sleboube')

footballOtherExpenses <- footballExpenseRaw[,c(1,4,7,10,13,16)]
colnames(footballOtherExpenses) <- c('Nation', '2020', '2019', '2018', '2017', '2016')
footballOtherExpenses <- footballOtherExpenses %>% filter(Nation != 'Eastern Sleboube')


footballAttend <- read_excel(footballDirectory, sheet = 'Attendance', range = 'B13:C34')

footballSocial <- read_excel(footballDirectory, sheet = 'Social Media', range = 'B13:G34')
footballSocial <- footballSocial %>% mutate(Tiktok=as.double(Tiktok)) %>% replace(is.na(.),0)

# Player Data

playLeagueShoot <- read_excel(playerdataDirectory, sheet = 'League Shooting', range = 'B12:AA5566')
```

```
playLeaguePass <- read_excel(playerdataDirectory, sheet = 'League Passing', range = 'B12:AF5566')

playLeagueDefend<- read_excel(playerdataDirectory, sheet = 'League Defense', range = 'B12:AG5566')

playLeagueGoalkeep <-read_excel(playerdataDirectory, sheet = 'League Goalkeeping', range =
'B12:AB425')

playTournResult2020 <-read_excel(playerdataDirectory, sheet = 'Tournament Results', range = 'B11:C27')

playTournResult2021 <-read_excel(playerdataDirectory, sheet = 'Tournament Results', range = 'E11:F35')

playTournShoot<-read_excel(playerdataDirectory, sheet = 'Tournament Shooting', range = 'B12:Z2027',
guess_max = 2000)

playTournPass <-read_excel(playerdataDirectory, sheet = 'Tournament Passing', range = 'B12:AE500')

playTournDefend <-read_excel(playerdataDirectory, sheet = 'Tournament Defense', range = 'B12:AF500')

playTournGoalkeep <-read_excel(playerdataDirectory, sheet = 'Tournament Goalkeeping', range =
'B12:AA141')

play2020Salary <-read_excel(playerdataDirectory, sheet = '2020 Salaries', range = 'B12:G2744')

play2021Salary <-read_excel(playerdataDirectory, sheet = '2021 Salaries', range = 'B12:G2834')


# Adjusted Datasets

#Tournament Rankings
#2020 Tournament Results
ranks2020 <- playTournResult2020 %>%
  mutate(Year = 2020) %>%
  rename('Rank' = `2020 Tournament Place`)

#2021 Tournament Results
ranks2021 <- playTournResult2021 %>%
  mutate(Year = 2021) %>%
  rename('Rank' = `2021 Tournament Place`)

# Combined tournament rankings
ranks <- rbind(ranks2020, ranks2021)
```

```
#Salary
#2020 Goalkeeper salary
salary2020 <- play2020Salary %>%
  mutate(Year = 2020) %>%
  filter(Position == 'GK')

#2021 Goalkeeper salary
salary2021 <- play2021Salary %>%
  mutate(Year = 2021) %>%
  filter(Position == 'GK')

#Combined Goalkeeper salary
salaryCombined <- rbind(salary2020, salary2021)
### END ### ---------------------------------------------------------------


### Functions ### -------------------------------------------------------

#Function that standardises the values to between 0 and 1
standardise <- function (column) {
  result <- (column - min(column))/(max(column) - min(column))
  return(result)
}

RSquare <- function(y_actual,y_predict){
  cor(y_actual,y_predict)^2
}

playerStat <- function(Born,Stat,Year) {
  ifelse((Year - Born) <= 24,Stat*(1 + ifelse(runif(1) <= 0.9, runif(1)*0.1, -runif(1)*0.1)),
         ifelse((Year - Born) <= 27,Stat*(1 + ifelse(runif(1) <= 0.75, runif(1)*0.1, -runif(1)*0.1)),
                ifelse((Year - Born) <= 32, Stat,
                       ifelse((Year - Born) <= 35, Stat*(1 - ifelse(runif(1) <= 0.75, runif(1)*0.1, -runif(1)*0.1)),
                              Stat*(1 - ifelse(runif(1) <= 0.9, runif(1)*0.1, -runif(1)*0.1))))))
}

### END ### ---------------------------------------------------------------


### Goalkeeper Selection ### ---------------------------------------------
```

```
#CORRELATION ANALYSIS TO PICK THE VARIABLES
#Attach the ranks
goalkeeperCorr <- na.omit(merge(playTournGoalkeep, ranks, by.x = c('Year', 'Nation'), by.y =
c('Year','Country'), all.x = TRUE)) %>%
  select(-c(Nation, Born, Player, Pos, League, Year)) %>%
  mutate_all(standardise)


# Correlation
corr <- round(cor(goalkeeperCorr),2)

# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)
}

upper_tri <- get_upper_tri(corr)
melted_cormat <- melt(upper_tri, na.rm = TRUE)
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()


#DATA SELECTION

#Transform playLeagueGoalkeep to combine with playTournGoalKeep
leagueGoalkeep <- playLeagueGoalkeep %>%
  select(-League) %>%
  rename('League' = `Squad`)

#Combine to create training dataset
goalkeeperTotal <- na.omit(rbind(leagueGoalkeep, playTournGoalkeep)) %>%
  select(-c(Born, Player, Pos, League, `Playing Time Starts`,
          `Playing Time Min`, `Playing Time 90s`, `Performance GA`,
```

```
          `Performance PKatt`, `D`, `Performance CS`,
          `Performance PKatt`, `Penalty Kicks PKA`, `Penalty Kicks PKm`, `Performance Saves`)) %>%
  mutate_if(is.numeric,standardise) %>%
  rename_with(~gsub(" ", "_", .)) %>%
  rename('Performance_Saves' = `Performance_Save%`) %>%
  rename('Performance_CS' = `Performance_CS%`) %>%
  rename('Penalty_Kick_Save' = `Penalty_Kicks_Save%`)


#Training and test dataset.
goalkeeperTraining <- goalkeeperTotal %>%
  filter(Nation != 'Rarita') %>%
  select(-c(Nation, Year))

set.seed(0)
GKtrain <- sample(nrow(goalkeeperTraining), round(nrow(goalkeeperTraining)*0.75))

goalkeeperTest <- goalkeeperTraining[-GKtrain,]


#Rarita dataset
goalkeeperRarita <- goalkeeperTotal %>%
  filter(Nation == 'Rarita' & Year == 1) %>%
  select(-c(Nation,Year))

goalkeeperRaritaName <- na.omit(rbind(leagueGoalkeep, playTournGoalkeep)) %>%
  filter(Nation == 'Rarita' & Year == 2021)


#MODELLING
#Linear Regression
linearGK <- glm(Performance_Saves ~ ., data = goalkeeperTraining, subset = GKtrain, family = 'gaussian')
summary(linearGK)
yhat_lr <- predict(linearGK, newdata = goalkeeperTraining[-GKtrain,])
gk_LR_MSE <- mean((yhat_lr - goalkeeperTraining$Performance_Saves[-GKtrain])^2)
gk_LR_R <- RSquare(goalkeeperTraining$Performance_Saves[-GKtrain],yhat_lr)


#KNN
i=1
k.optm=1
for (i in seq(from=1, to=20, by=1)){
```

```
  knn.mod <-  knnreg(goalkeeperTraining[GKtrain,-4], goalkeeperTraining$Performance_Saves[GKtrain],
k=i)
  yhat <- predict(knn.mod,goalkeeperTraining[-GKtrain,-4])
  k.optm[i] <- mean((yhat - goalkeeperTraining$Performance_Saves[-GKtrain])^2)
}
plot(k.optm)


knnGK <- knnreg(goalkeeperTraining[GKtrain,-4], goalkeeperTraining$Performance_Saves[GKtrain], k=2)
yhat_knn<- predict(knnGK,goalkeeperTraining[-GKtrain,-4])
gk_KNN_MSE <- mean((yhat_knn - goalkeeperTraining$Performance_Saves[-GKtrain])^2)
gk_KNN_R <- RSquare(goalkeeperTraining$Performance_Saves[-GKtrain],yhat_knn)

#Bagging
set.seed(1)
bagGK <- randomForest(Performance_Saves ~ ., data = goalkeeperTraining, subset = GKtrain, mtry = 9,
ntree = 5000, importance = TRUE)
yhat_bg <- predict(bagGK, newdata = goalkeeperTraining[-GKtrain,], n.trees = 5000)
gk_bag_MSE <- mean((yhat_bg - goalkeeperTraining$Performance_Saves[-GKtrain])^2)
gk_bag_R <- RSquare(goalkeeperTraining$Performance_Saves[-GKtrain],yhat_bg)

#Random Forest
set.seed(1)
tuneRF(goalkeeperTraining[GKtrain,-4], goalkeeperTraining$Performance_Saves[GKtrain], stepFactor =
0.9, ntree=5000)
rfGK <- randomForest(Performance_Saves ~ ., data = goalkeeperTraining, subset = GKtrain, mtry = 8,
ntree = 5000, importance = TRUE)
yhat_rf <- predict(rfGK, newdata = goalkeeperTraining[-GKtrain,], n.trees = 5000)
gk_RF_MSE <- mean((yhat_rf - goalkeeperTraining$Performance_Saves[-GKtrain])^2)
gk_RF_R <- RSquare(goalkeeperTraining$Performance_Saves[-GKtrain],yhat_rf)
# varImpPlot(rfgk)

#Boosting
boostGK <- gbm(Performance_Saves ~ ., data = goalkeeperTraining[GKtrain,], distribution = "gaussian",
n.trees = 1000, interaction.depth = 5, cv.folds = 10)
best.iter <- gbm.perf(boostGK, method="cv")
yhat_boost <- predict(boostGK, newdata = goalkeeperTraining[-GKtrain,],n.trees = 1000)
gk_boost_MSE <- mean((yhat_boost - goalkeeperTraining$Performance_Saves[-GKtrain])^2)
gk_boost_R <- RSquare(goalkeeperTraining$Performance_Saves[-GKtrain],yhat_boost)

#Model MSE
Model <- c("Linear Regression","KNN","Bagging","Random Forest","Boosting")
```

```
MSE <- c(gk_LR_MSE,gk_KNN_MSE,gk_bag_MSE,gk_RF_MSE,gk_boost_MSE)
`R^2` <- c(gk_LR_R,gk_KNN_R,gk_bag_R,gk_RF_R,gk_boost_R)
gkMSE <- data.frame(Model,MSE,`R^2`)



#New prediction

gkPred <- predict(knnGK, newdata = goalkeeperRarita[,-5])

goalkeeperBest <- cbind(goalkeeperRaritaName,gkPred) %>%
  arrange(desc(gkPred))




### END ### ----------------------------------------------------------------


### Defenders Selection ### -------------------------------------------------------

# Data for correlation analysis
defendCorrDef <- na.omit(merge(playTournDefend, ranks, by.x = c('Year', 'Nation'), by.y =
c('Year','Country'), all.x = TRUE)) %>%
  filter(Pos %in% c('DF','DFFW','DFMF','FWDF','MFDF')) %>%
  select(-c(Born, Pos, League, Year))

defendCorrPass <- na.omit(playTournPass) %>%
  filter(Pos %in% c('DF','DFFW','DFMF','FWDF','MFDF')) %>%
  select(-c(Age, `90s`, Born, Pos, League, Year))

defendCorr <- na.omit(merge(defendCorrDef, defendCorrPass, by.x = c('Player','Nation'),
                            by.y = c('Player','Nation'), all.x = TRUE)) %>%
  select(-c(Player, Nation)) %>%
  mutate_all(standardise)


corr <- round(cor(defendCorr),2)

# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
```

```
  return(cormat)
}

upper_tri <- get_upper_tri(corr)
melted_cormat <- melt(upper_tri, na.rm = TRUE)
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()


#DATA SELECTION

#Transform playLeagueDefend to combine with playTournDefend
leagueDefend <- playLeagueDefend %>%
  select(-League) %>%
  rename('League' = `Squad`) %>%
  filter(Pos %in% c('DF','DFFW','DFMF','FWDF','MFDF'))

#Combine to create defence dataset
defenceTotal <- rbind(leagueDefend, playTournDefend) %>%
  select(c(Player, Year, Nation, League, Age, `Tackles Tkl`, `Vs Dribbles Past`, `Pressures %`, `Pressures
Def 3rd`,
         `Pressures Press`, `Blocks Blocks`, `Tkl+Int`))


#Transform playLeaguePass to combine with playTournPass
leaguePass <- playLeaguePass %>%
  select(-League) %>%
  rename('League' = `Squad`) %>%
  filter(Pos %in% c('DF','DFFW','DFMF','FWDF','MFDF'))

#Combine to create pass dataset
passTotal <- rbind(leaguePass, playTournPass) %>%
  select(c(Player, Nation, League, Age, `Total Cmp%`, `1/3`, `Prog`))
```

```
#Combine to create Training dataset
defendTotal <- na.omit(merge(defenceTotal, passTotal, by.x = c('Player', 'Nation', 'Age', 'League'), by.y =
c('Player', 'Nation', 'Age', 'League'))) %>%
  select(-c(Player, League)) %>%
  mutate_if(is.numeric,standardise) %>%
  rename_with(~gsub(" ", "_", .)) %>%
  rename('Pressures' = `Pressures_%`) %>%
  rename('TklInt' = `Tkl+Int`) %>%
  rename('Total_Cmp' = `Total_Cmp%`) %>%
  rename('ThirdComplete' = `1/3`)


#Training and test dataset.
defendTraining <- defendTotal %>%
  filter(Nation != 'Rarita') %>%
  select(-c(Nation, Year))

set.seed(0)
DFtrain <- sample(nrow(defendTraining), round(nrow(defendTraining)*0.75))

defendTest <- defendTraining[-DFtrain,]


#Rarita dataset
defendRarita <- defendTotal %>%
  filter(Nation == 'Rarita' & Year == 1) %>%
  select(-c(Nation, Year))

defendRaritaName <- na.omit(merge(defenceTotal, passTotal, by.x = c('Player', 'Nation', 'Age', 'League'),
by.y = c('Player', 'Nation', 'Age', 'League'))) %>%
  filter(Nation == 'Rarita' & Year == 2021)


#MODELLING
#Linear Regression
linearDF <- glm(Pressures ~ ., data = defendTraining, subset = DFtrain, family = 'gaussian')
summary(linearDF)
yhat_lr <- predict(linearDF, newdata = defendTraining[-DFtrain,])
df_LR_MSE <- mean((yhat_lr - defendTraining$Pressures[-DFtrain])^2)
df_LR_R <- RSquare(defendTraining$Pressures[-DFtrain],yhat_lr)

#KNN
```

```
i=1
k.optm=1
for (i in seq(from=1, to=30, by=1)){
  knn.mod <-  knnreg(defendTraining[DFtrain,-4], defendTraining$Pressures[DFtrain], k=i)
  yhat <- predict(knn.mod,defendTraining[-DFtrain,-4])
  k.optm[i] <- mean((yhat - defendTraining$Pressures[-DFtrain])^2)
}
plot(k.optm)


knnDF <- knnreg(defendTraining[DFtrain,-4], defendTraining$Pressures[DFtrain], k=25)
yhat_knn<- predict(knnDF,defendTraining[-DFtrain,-4])
df_KNN_MSE <- mean((yhat_knn - defendTraining$Pressures[-DFtrain])^2)
df_KNN_R <- RSquare(defendTraining$Pressures[-DFtrain],yhat_knn)

#Bagging
set.seed(1)
bagDF <- randomForest(Pressures ~ ., data = defendTraining, subset = DFtrain, mtry = 9, ntree = 5000,
importance = TRUE)
yhat_bg <- predict(bagDF, newdata = defendTraining[-DFtrain,], n.trees = 5000)
df_bag_MSE <- mean((yhat_bg - defendTraining$Pressures[-DFtrain])^2)
df_bag_R <- RSquare(defendTraining$Pressures[-DFtrain],yhat_bg)


#Random Forest
set.seed(1)
tuneRF(defendTraining[DFtrain,-4], defendTraining$Pressures[DFtrain], stepFactor = 0.9, ntree=5000)
rfDF <- randomForest(Pressures ~ ., data = defendTraining, subset = DFtrain, mtry = 2, ntree = 5000,
importance = TRUE)
yhat_rf <- predict(rfDF, newdata = defendTraining[-DFtrain,], n.trees = 5000)
df_RF_MSE <- mean((yhat_rf - defendTraining$Pressures[-DFtrain])^2)
df_RF_R <- RSquare(defendTraining$Pressures[-DFtrain],yhat_rf)
# varImpPlot(rfDF)

#Boosting
boostDF <- gbm(Pressures ~ ., data = defendTraining[DFtrain,], distribution = "gaussian", n.trees = 1000,
interaction.depth = 5, cv.folds = 10)
best.iter <- gbm.perf(boostDF, method="cv")
yhat_boost <- predict(boostDF, newdata = defendTraining[-DFtrain,],n.trees = 1000)
df_boost_MSE <- mean((yhat_boost - defendTraining$Pressures[-DFtrain])^2)
df_boost_R <- RSquare(defendTraining$Pressures[-DFtrain],yhat_boost)
```

```
#Model MSE
Model <- c("Linear Regression","KNN","Bagging","Random Forest","Boosting")
MSE <- c(df_LR_MSE,df_KNN_MSE,df_bag_MSE,df_RF_MSE,df_boost_MSE)
`R^2` <- c(df_LR_R,df_KNN_R,df_bag_R,df_RF_R,df_boost_R)
dfMSE <- data.frame(Model,MSE,`R^2`)


#New prediction
dfPred <- predict(linearDF, newdata = defendRarita[,-4])

defendBest <- cbind(defendRaritaName,dfPred) %>%
  arrange(desc(dfPred))


# REGRESSION ON TWO VARIABLES
defendTotalMulti <- defendTotal %>%
  mutate('Pressures+Total_Cmp' = 0.5 * Pressures + 0.5 * Total_Cmp) %>%
  select(-c(Pressures, Total_Cmp))

#Training and test dataset.
defendTraining <- defendTotalMulti %>%
  filter(Nation != 'Rarita') %>%
  select(-c(Nation, Year))

set.seed(0)
DFtrain <- sample(nrow(defendTraining), round(nrow(defendTraining)*0.75))

defendTest <- defendTraining[-DFtrain,]


#Rarita dataset
defendRarita <- defendTotalMulti %>%
  filter(Nation == 'Rarita' & Year == 1) %>%
  select(-c(Nation, Year))

defendRaritaName <- na.omit(merge(defenceTotal, passTotal, by.x = c('Player', 'Nation', 'Age', 'League'),
by.y = c('Player', 'Nation', 'Age', 'League'))) %>%
  filter(Nation == 'Rarita' & Year == 2021)


#MODELLING
#Linear Regression
```

```
linearDF <- glm(`Pressures+Total_Cmp` ~ ., data = defendTraining, subset = DFtrain, family = 'gaussian')
summary(linearDF)
yhat_lr <- predict(linearDF, newdata = defendTraining[-DFtrain,])
df_LR_MSE <- mean((yhat_lr - defendTraining$`Pressures+Total_Cmp`[-DFtrain])^2)
df_LR_R <- RSquare(defendTraining$`Pressures+Total_Cmp`[-DFtrain],yhat_lr)

#KNN
i=1
k.optm=1
for (i in seq(from=1, to=30, by=1)){
  knn.mod <-  knnreg(defendTraining[DFtrain,-10], defendTraining$`Pressures+Total_Cmp`[DFtrain], k=i)
  yhat <- predict(knn.mod,defendTraining[-DFtrain,-10])
  k.optm[i] <- mean((yhat - defendTraining$`Pressures+Total_Cmp`[-DFtrain])^2)
}
plot(k.optm)


knnDF <- knnreg(defendTraining[DFtrain,-10], defendTraining$`Pressures+Total_Cmp`[DFtrain], k=12)
yhat_knn<- predict(knnDF,defendTraining[-DFtrain,-10])
df_KNN_MSE <- mean((yhat_knn - defendTraining$`Pressures+Total_Cmp`[-DFtrain])^2)
df_KNN_R <- RSquare(defendTraining$`Pressures+Total_Cmp`[-DFtrain],yhat_knn)


#Bagging
set.seed(1)
bagDF <- randomForest(`Pressures+Total_Cmp` ~ ., data = defendTraining, subset = DFtrain, mtry = 9,
ntree = 5000, importance = TRUE)
yhat_bg <- predict(bagDF, newdata = defendTraining[-DFtrain,], n.trees = 5000)
df_bag_MSE <- mean((yhat_bg - defendTraining$`Pressures+Total_Cmp`[-DFtrain])^2)
df_bag_R <- RSquare(defendTraining$`Pressures+Total_Cmp`[-DFtrain],yhat_bg)


#Random Forest
set.seed(1)
tuneRF(defendTraining[DFtrain,-10], defendTraining$`Pressures+Total_Cmp`[DFtrain], stepFactor = 0.9,
ntree=5000)
rfDF <- randomForest(`Pressures+Total_Cmp` ~ ., data = defendTraining, subset = DFtrain, mtry = 2, ntree
= 5000, importance = TRUE)
yhat_rf <- predict(rfDF, newdata = defendTraining[-DFtrain,], n.trees = 5000)
df_RF_MSE <- mean((yhat_rf - defendTraining$`Pressures+Total_Cmp`[-DFtrain])^2)
df_RF_R <- RSquare(defendTraining$`Pressures+Total_Cmp`[-DFtrain],yhat_rf)
# varImpPlot(rfDF)
```

```
#Boosting
boostDF <- gbm(`Pressures+Total_Cmp` ~ ., data = defendTraining[DFtrain,], distribution = "gaussian",
n.trees = 1000, interaction.depth = 5, cv.folds = 10)
best.iter <- gbm.perf(boostDF, method="cv")
yhat_boost <- predict(boostDF, newdata = defendTraining[-DFtrain,],n.trees = 1000)
df_boost_MSE <- mean((yhat_boost - defendTraining$`Pressures+Total_Cmp`[-DFtrain])^2)
df_boost_R <- RSquare(defendTraining$`Pressures+Total_Cmp`[-DFtrain],yhat_boost)

#Model MSE
Model <- c("Linear Regression","KNN","Bagging","Random Forest","Boosting")
MSE <- c(df_LR_MSE,df_KNN_MSE,df_bag_MSE,df_RF_MSE,df_boost_MSE)
`R^2` <- c(df_LR_R,df_KNN_R,df_bag_R,df_RF_R,df_boost_R)
dfMSE <- data.frame(Model,MSE,`R^2`)

#New prediction
dfPred <- predict(linearDF, newdata = defendRarita[,-10])

plot(yhat_knn,defendTraining$Pressures[-DFtrain])
abline(0,1)

defendBest <- cbind(defendRaritaName,dfPred) %>%
  arrange(desc(dfPred))


View(cbind(yhat_knn,defendTraining$Pressures[-DFtrain]))

### END ### ----------------------------------------------------------------


### Midfielders Selection ### ------------------------------------------------------
# combine shooting passing and defending tournament
# use tournament 2021 data only for variable selection
playTournDefend21 <- playTournDefend %>% filter(Year == "2021")
# use shooting 90s
playTournDefend21 <- playTournDefend21 %>% select(-c('90s','League'))

# data set with shooting passing and defending tournament 2021
TSPD <- merge(TSP,playTournDefend21, by = c("Player","Nation", "Year", "Pos","Age","Born"),all = TRUE)

#only the MF
```

```
TSPDMF <- TSPD %>% filter(Pos == "MF" | Pos == "FWMF" | Pos == "MFDF" | Pos == "MFFW" | Pos ==
"DFMF")


# league data for training
playLeagueDefendfix <- playLeagueDefend %>% select(-c("League","90s"))
LSPD <- merge(LSP,playLeagueDefendfix, by = c("Player", "Squad","Year","Pos","Age","Born","Nation"),all =
TRUE)
#only MF
LSPDMF <- LSPD %>% filter(Pos == "MF" | Pos == "FWMF" | Pos == "MFDF" | Pos == "MFFW" | Pos ==
"DFMF") %>%
  select(-Squad)

# correleation for variable selection
MFcorr <- merge(TSPDMF, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'), all.x = TRUE) %>%
  mutate(Rank = ifelse(is.na(Rank) == TRUE,25,Rank)) %>%
  select(-c(Nation, Born, Player,League, Pos, Year, 'Standard SoT%','Total Cmp%','Short Cmp%','Medium
Cmp%','Long Cmp%'))

MFcorr2 <- round(cor(MFcorr, use = "complete.obs"),2)
MFupper_tri <- get_upper_tri(MFcorr2)
MFmelted_cormat <- melt(MFupper_tri, na.rm = TRUE)
ggplot(data = MFmelted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                      midpoint = 0, limit = c(-1,1), space = "Lab",
                      name="Pearson\nCorrelation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                  size = 12, hjust = 1))+
  coord_fixed()

# data to train MF and use Total Cmp as the dependent variable
MFtotal <- rbind(LSPDMF,TSPDMF) %>% filter(Nation != "Rarita") %>%
  select(-c(Born, Player, Pos, League, Year,Nation)) %>%
  select(c(Age, '90s', 'Total Cmp', 'Medium Att', '1/3', 'Total TotDist', 'Tackles Mid 3rd', 'Prog', 'Total PrgDist',
'Blocks Pass')) %>%
  mutate_all(standardise)

# rarita MF

MFRarita <- rbind(LSPDMF,TSPDMF) %>% filter(Nation == "Rarita") %>%
```

```
  select(-c(Born, Player, Pos, League, Year,Nation)) %>%
  select(c(Age, '90s', 'Total Cmp', 'Medium Att', '1/3', 'Total TotDist', 'Tackles Mid 3rd', 'Prog', 'Total PrgDist',
'Blocks Pass')) %>%
  mutate_all(standardise)

MFName <- rbind(LSPDMF,TSPDMF) %>% filter(Nation == "Rarita") %>% select(Player)

#splitting the data
set.seed(0)
MFtrain <- sample(nrow(MFtotal), round(nrow(MFtotal)*0.75))

MFTest <- MFtotal[-MFtrain,]

#GLM
MFlinearFit <- glm(MFtotal$`Total Cmp` ~ ., data = MFtotal, subset = MFtrain, family = 'gaussian')
summary(MFlinearFit)
MFyhat_lr <- predict(MFlinearFit, newdata = MFtotal[-MFtrain,])
MF_LR_MSE <- mean((MFyhat_lr - MFtotal$'Total Cmp'[-MFtrain])^2)
MF_LR_R <- RSquare(MFtotal$'Total Cmp'[-MFtrain],MFyhat_lr)

#KNN
i=1
MF.k.optm=1
for (i in seq(from=1, to=20, by=1)){
  MF.knn.mod <-  knnreg(MFtotal[MFtrain,], MFtotal$'Total Cmp'[MFtrain], k=i)
  MF.yhat <- predict(MF.knn.mod,MFtotal[-MFtrain,])
  MF.k.optm[i] <- mean((MF.yhat - MFtotal$'Total Cmp'[-MFtrain])^2)
}
plot(MF.k.optm)


MFknn <- knnreg(MFtotal[MFtrain,], MFtotal$'Total Cmp'[MFtrain], k=2)
MFyhat_knn<- predict(MFknn,MFtotal[-MFtrain,])
MF_KNN_MSE <- mean((MFyhat_knn - MFtotal$'Total Cmp'[-MFtrain])^2)
MF_KNN_R <- RSquare(MFtotal$'Total Cmp'[-MFtrain],MFyhat_knn)

#Bagging
set.seed(1)
colnames(MFtotal) <-
c("Age","min","TotalCmp","MediumAtt","third","TotalTotDist","TacklesMid3rd","Prog","TotalPrgDist","Blocks
Pass")
```

```
MFbag <- randomForest(MFtotal$TotalCmp ~ ., data = MFtotal, subset = MFtrain, mtry = 9, ntree = 5000,
importance = TRUE)
MFyhat_bg <- predict(MFbag, newdata = MFtotal[-MFtrain,], n.trees = 5000)
MF_bag_MSE <- mean((MFyhat_bg - MFtotal$TotalCmp[-MFtrain])^2)
MF_bag_R <- RSquare(MFtotal$TotalCmp[-MFtrain],MFyhat_bg)

#Random Forest
set.seed(1)
tuneRF(MFtotal[MFtrain,-4], MFtotal[MFtrain], stepFactor = 0.9, ntree=5000)
MFGK <- randomForest(TotalCmp ~ ., data = MFtotal, subset = MFtrain, mtry = 8, ntree = 5000,
importance = TRUE)
MFyhat_rf <- predict(MFGK, newdata = MFtotal[-MFtrain,], n.trees = 5000)
MF_RF_MSE <- mean((MFyhat_rf - MFtotal$TotalCmp[-MFtrain])^2)
MF_RF_R <- RSquare(MFtotal$TotalCmp[-MFtrain],MFyhat_rf)
# varImpPlot(rfgk)

#Boosting
MFboost <- gbm(TotalCmp ~ ., data = MFtotal[MFtrain,], distribution = "gaussian", n.trees = 1000,
interaction.depth = 5, cv.folds = 10)
MFbest.iter <- gbm.perf(MFboost, method="cv")
MFyhat_boost <- predict(MFboost, newdata = MFtotal[-MFtrain,],n.trees = 1000)
MF_boost_MSE <- mean((MFyhat_boost - MFtotal$TotalCmp[-MFtrain])^2)
MF_boost_R <- RSquare(MFtotal$TotalCmp[-MFtrain],MFyhat_boost)

#Model MSE
Model <- c("Linear Regression","KNN","Bagging","Random Forest","Boosting")
MSE <- c(MF_LR_MSE,MF_KNN_MSE,MF_bag_MSE,MF_RF_MSE,MF_boost_MSE)
`R^2` <- c(MF_LR_R,MF_KNN_R,MF_bag_R,MF_RF_R,MF_boost_R)
MFMSE <- data.frame(Model,MSE,`R^2`)

#New prediction

MFPred <- predict(MFlinearFit, newdata = MFRarita)
MFBest <- cbind(MFName,MFPred) %>%
  arrange(desc(MFPred))

MFRarita2 <- rbind(LSPDMF,TSPDMF) %>% filter(Nation == "Rarita")
MFData <- merge(MFBest,MFRarita2, by = "Player")

### END ### ----------------------------------------------------------
```

```
### Forwards Selection ### ---------------------------------------------------
# combine league shooting and passing for training
# taking only league and 90s from shoot data
playLeaguePassfix <- playLeaguePass %>% select(-c("League","90s"))
LSP <- merge(playLeagueShoot,playLeaguePassfix, by = c("Player",
"Squad","Year","Pos","Age","Born","Nation"),all = TRUE)
#only FW
LSPFW <- LSP %>% filter(Pos == "FW" | Pos == "FWMF" | Pos == "FWDF" | Pos == "MFFW" | Pos ==
"DFFW") %>%
  select(-Squad)

# combine tournament shooting and passing
# use tournament 2021 data only for variable selection
playTournShoot21 <- playTournShoot %>% filter(Year == "2021")
PlayTourpassfix <- playTourPass %>% select(-c('90s','League')) # 90s, league are different for shot and
pass

TSP <- merge(playTournShoot21,PlayTourpassfix, by = c("Player","Nation", "Year", "Pos","Age","Born"),all =
TRUE)
# passing no 2020 data

# get only fw
TSPFW <- TSP %>% filter(Pos == "FW" | Pos == "FWMF" | Pos == "FWDF" | Pos == "MFFW" | Pos ==
"DFFW")

# correlation and merge the ranks
FWcorr <- merge(TSPFW, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'), all.x = TRUE) %>%
  mutate(Rank = ifelse(is.na(Rank) == TRUE,25,Rank)) %>%
  select(-c(Nation, Born, Player, Pos, League, Year, 'Standard SoT%','Total Cmp%','Short Cmp%','Medium
Cmp%','Long Cmp%'))

# inspected and visualising na
colSums(is.na(FWcorr))
vis_miss(FWcorr)

# Correlation for FW to choose variables
FWcorr2 <- round(cor(FWcorr, use = "complete.obs"),2)
FWupper_tri <- get_upper_tri(FWcorr2)
FWmelted_cormat <- melt(FWupper_tri, na.rm = TRUE)
ggplot(data = FWmelted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
```

```
                    midpoint = 0, limit = c(-1,1), space = "Lab",
                    name="Pearson\nCorrelation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()

# From choose Age, 90s,  Standard sh, standard Fk, Total cmp, KP, Total Att, xA, Expected xG (Gls)

# not choosen as high correlation with chosen

# use this to standardise
mutate_all(standardise)

# data to train FW and use gls as the dependent variable and need to remove rarita nation
FWtotal <- rbind(LSPFW,TSPFW) %>% filter(Nation != "Rarita") %>%
  select(-c(Born, Player, Pos, League, Year,Nation)) %>%
  select(c(Age, '90s',  'Standard Sh', 'Standard FK', 'Total Cmp', 'KP', 'Total Att', 'xA', 'Expected xG', 'Gls'))
%>%
  mutate_all(standardise)

# rarita FW

FWRarita <- rbind(LSPFW,TSPFW) %>% filter(Nation == "Rarita") %>%
  select(-c(Born, Player, Pos, League, Year,Nation)) %>%
  select(c(Age, '90s',  'Standard Sh', 'Standard FK', 'Total Cmp', 'KP', 'Total Att', 'xA', 'Expected xG', 'Gls'))
%>%
  mutate_all(standardise)

FWName <- rbind(LSPFW,TSPFW) %>% filter(Nation == "Rarita") %>% select(Player)

#splitting the data
set.seed(0)
FWtrain <- sample(nrow(FWtotal), round(nrow(FWtotal)*0.75))

FWTest <- FWtotal[-FWtrain,]

#GLM
FWlinearFit <- glm(Gls ~ ., data = FWtotal, subset = FWtrain, family = 'gaussian')
summary(FWlinearFit)
FWyhat_lr <- predict(FWlinearFit, newdata = FWtotal[-FWtrain,])
FW_LR_MSE <- mean((FWyhat_lr - FWtotal$Gls[-FWtrain])^2)
```

```
FW_LR_R <- RSquare(FWtotal$Gls[-FWtrain],FWyhat_lr)

#KNN
i=1
FW.k.optm=1
for (i in seq(from=1, to=20, by=1)){
  FW.knn.mod <-  knnreg(FWtotal[FWtrain,-10], FWtotal$Gls[FWtrain], k=i)
  FW.yhat <- predict(FW.knn.mod,FWtotal[-FWtrain,-10])
  FW.k.optm[i] <- mean((FW.yhat - FWtotal$Gls[-FWtrain])^2)
}
plot(FW.k.optm)


FWknn <- knnreg(FWtotal[FWtrain,-10], FWtotal$Gls[FWtrain], k=2)
FWyhat_knn<- predict(FWknn,FWtotal[-FWtrain,-10])
FW_KNN_MSE <- mean((FWyhat_knn - FWtotal$Gls[-FWtrain])^2)
FW_KNN_R <- RSquare(FWtotal$Gls[-FWtrain],FWyhat_knn)

#Bagging
set.seed(1)
colnames(FWtotal) <-
c("Age","min","StandardSh","StandardFK","TotalCmp","KP","TotalAtt","xA","ExpectedxG","Gls" )
FWbag <- randomForest(Gls ~ ., data = FWtotal, subset = FWtrain, mtry = 9, ntree = 5000, importance =
TRUE)
FWyhat_bg <- predict(FWbag, newdata = FWtotal[-FWtrain,], n.trees = 5000)
FW_bag_MSE <- mean((FWyhat_bg - FWtotal$Gls[-FWtrain])^2)
FW_bag_R <- RSquare(FWtotal$Gls[-FWtrain],FWyhat_bg)

#Random Forest
set.seed(1)
tuneRF(FWtotal[FWtrain,-4], FWtotal[FWtrain], stepFactor = 0.9, ntree=5000)
FWGK <- randomForest(Gls ~ ., data = FWtotal, subset = FWtrain, mtry = 8, ntree = 5000, importance =
TRUE)
FWyhat_rf <- predict(FWGK, newdata = FWtotal[-FWtrain,], n.trees = 5000)
FW_RF_MSE <- mean((FWyhat_rf - FWtotal$Gls[-FWtrain])^2)
FW_RF_R <- RSquare(FWtotal$Gls[-FWtrain],FWyhat_rf)
# varImpPlot(rfgk)

#Boosting
FWboost <- gbm(Gls ~ ., data = FWtotal[FWtrain,], distribution = "gaussian", n.trees = 1000,
interaction.depth = 5, cv.folds = 10)
FWbest.iter <- gbm.perf(FWboost, method="cv")
```

```
FWyhat_boost <- predict(FWboost, newdata = FWtotal[-FWtrain,],n.trees = 1000)
FW_boost_MSE <- mean((FWyhat_boost - FWtotal$Gls[-FWtrain])^2)
FW_boost_R <- RSquare(FWtotal$Gls[-FWtrain],FWyhat_boost)

#Model MSE
Model <- c("Linear Regression","KNN","Bagging","Random Forest","Boosting")
MSE <- c(FW_LR_MSE,FW_KNN_MSE,FW_bag_MSE,FW_RF_MSE,FW_boost_MSE)
`R^2` <- c(FW_LR_R,FW_KNN_R,FW_bag_R,FW_RF_R,FW_boost_R)
FWMSE <- data.frame(Model,MSE,`R^2`)

#New prediction

FWPred <- predict(FWknn, newdata = FWRarita[,-10])
FWBest <- cbind(FWName,FWPred) %>%
  arrange(desc(FWPred))

FWRarita2 <- rbind(LSPFW,TSPFW) %>% filter(Nation == "Rarita")

FWData <- merge(FWBest,FWRarita2, by = "Player")

### END ### ----------------------------------------------------------


### Probability ### ---------------------------------------------------
#Players
gkRar <- c('F. Akumu', 'F. Ithungu')
dfRar <- c('N. Terzi?', 'H. Zare', 'F. Yunusa', 'Q. bin Ismail', 'C. Tukamushaba')
mfRar <- c('O. Wanjala', 'X. Leroy', 'F. Chin', 'S. Barman')
fwRar <- c('H. Makumbi', 'X. Thomas', 'E. Mudzingwa', 'F. Ajio')


#Data Selection
gkRarita <- playLeagueGoalkeep %>%
  filter(Player %in% gkRar & Year == 2021) %>%
  select(-Squad)

dfRarita <- playLeagueDefend %>%
  filter(Player %in% dfRar & Year == 2021) %>%
  select(-Squad)

mfRarita <- playLeaguePass %>%
  filter(Player %in% mfRar & Year == 2021) %>%
```

```
  select(-Squad)

fwRarita <- playLeagueShoot %>%
  filter(Player %in% fwRar & Year == 2021) %>%
  select(-Squad)

#Top 10 Data
gkTeamDataRaritaTT <- rbind(playTournGoalkeep, gkRarita) %>%
  select(c('Player','Born','Nation','Performance Save%','Year')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('PerfSavePer' = standardise(`Performance Save%`)) %>%
  filter(Nation != 'Rarita')

gkTeamDataTT <- merge(gkTeamDataRaritaTT, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'),
all.x = TRUE) %>%
  mutate(Rank = ifelse(is.na(Rank) == TRUE,25,Rank)) %>%
  mutate(Rank = ifelse(Rank>10,0,1)) %>%
  group_by(Nation)%>%
  summarise_at(c('Rank', 'PerfSavePer'),mean)

dfTeamDataRaritaTT <- rbind(playTournDefend, dfRarita) %>%
  filter(Pos %in% c('DF','DFFW','DFMF','FWDF','MFDF')) %>%
  select(c('Player','Born','Nation','Pressures %','Year','Pos')) %>%
  drop_na()%>%
  filter(Year == 2021)%>%
  mutate('PressurePer' = standardise(`Pressures %`)) %>%
  filter(Nation != 'Rarita')

dfTeamDataTT <- merge(dfTeamDataRaritaTT, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'),
all.x = TRUE) %>%
  mutate(Rank = ifelse(Rank>10,0,1)) %>%
  group_by(Nation)%>%
  summarise_at(c('Rank', 'PressurePer'),mean)

mfTeamDataRaritaTT <- rbind(playTournPass, mfRarita) %>%
  filter(Pos %in% c('MF','FWMF','MFDF','MFFW','DFMF')) %>%
  select(c('Player','Year','Born','Nation','Total Cmp')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('TotalCmp' = standardise(`Total Cmp`)) %>%
  filter(Nation != 'Rarita')
```

```
mfTeamDataTT <- merge(mfTeamDataRaritaTT, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'),
all.x = TRUE) %>%
  mutate(Rank = ifelse(Rank>10,0,1)) %>%
  group_by(Nation)%>%
  summarise_at(c('Rank', 'TotalCmp'),mean)

fwTeamDataRaritaTT <- rbind(playTournShoot, fwRarita) %>%
  filter(Pos %in% c('FW','FWMF','FWDF','MFFW','DFFW')) %>%
  select(c('Player','Born','Nation','Gls','Year')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('Gls' = standardise(`Gls`)) %>%
  filter(Nation != 'Rarita')

fwTeamDataTT <- merge(fwTeamDataRaritaTT, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'),
all.x = TRUE) %>%
  mutate(Rank = ifelse(is.na(Rank) == TRUE,25,Rank)) %>%
  mutate(Rank = ifelse(Rank>10,0,1)) %>%
  group_by(Nation)%>%
  summarise_at(c('Rank', 'Gls'),mean)

modelDataTT <-
merge(gkTeamDataTT,merge(merge(dfTeamDataTT,mfTeamDataTT),fwTeamDataTT))[,-1]


#Models
lrprobTT <- glm(as.factor(Rank)~.,data=modelDataTT,family="binomial")

bagprobTT <- randomForest(factor(Rank)~.,data=modelDataTT,mtry=4,ntree=5000,importance=TRUE)

rfprobTT <-randomForest(factor(Rank)~.,data=modelDataTT,mtry=3,ntree=5000,importance=TRUE)



#Actual Rank Data

gkTeamDataRarita <- rbind(playTournGoalkeep, gkRarita) %>%
  select(c('Player','Born','Nation','Performance Save%','Year')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('PerfSavePer' = standardise(`Performance Save%`)) %>%
```

```
  filter(Nation != 'Rarita')

gkTeamData <- merge(gkTeamDataRarita, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'), all.x =
TRUE) %>%
  mutate(Rank = ifelse(is.na(Rank) == TRUE,25,Rank)) %>%
  group_by(Nation)%>%
  summarise_at(c('Rank', 'PerfSavePer'),mean)

dfTeamDataRarita <- rbind(playTournDefend, dfRarita) %>%
  filter(Pos %in% c('DF','DFFW','DFMF','FWDF','MFDF')) %>%
  select(c('Player','Born','Nation','Pressures %','Year','Pos')) %>%
  drop_na()%>%
  filter(Year == 2021)%>%
  mutate('PressurePer' = standardise(`Pressures %`)) %>%
  filter(Nation != 'Rarita')

dfTeamData <- merge(dfTeamDataRarita, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'), all.x =
TRUE) %>%
  group_by(Nation) %>%
  summarise_at(c('Rank', 'PressurePer'),mean)

mfTeamDataRarita <- rbind(playTournPass, mfRarita) %>%
  filter(Pos %in% c('MF','FWMF','MFDF','MFFW','DFMF')) %>%
  select(c('Player','Year','Born','Nation','Total Cmp')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('TotalCmp' = standardise(`Total Cmp`)) %>%
  filter(Nation != 'Rarita')

mfTeamData <- merge(mfTeamDataRarita, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'), all.x =
TRUE) %>%
  group_by(Nation)%>%
  summarise_at(c('Rank','TotalCmp'),mean)

fwTeamDataRarita <- rbind(playTournShoot, fwRarita) %>%
  filter(Pos %in% c('FW','FWMF','FWDF','MFFW','DFFW')) %>%
  select(c('Player','Born','Nation','Gls','Year')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('Gls' = standardise(`Gls`)) %>%
  filter(Nation != 'Rarita')
```

```
fwTeamData <- merge(fwTeamDataRarita, ranks, by.x = c('Year', 'Nation'), by.y = c('Year','Country'), all.x =
TRUE) %>%
  group_by(Nation)%>%
  summarise_at(c('Rank', 'Gls'),mean)

modelData <- merge(gkTeamData,merge(merge(dfTeamData,mfTeamData),fwTeamData))[,-1]

#Linear Regression
lrprob <- glm(Rank~.,data=modelData,family="gaussian")

bagprob <- randomForest((Rank)~.,data=modelData,mtry=4,ntree=5000,importance=TRUE)

rfprob <-randomForest((Rank)~.,data=modelData,mtry=3,ntree=5000,importance=TRUE)




simulationTT <- function() {
  #GOALKEEPERS
  gkTeamData2 <- rbind(playTournGoalkeep, gkRarita) %>%
    select(c('Player','Born','Nation','Performance Save%','Year')) %>%
    drop_na() %>%
    filter(Year == 2021)%>%
    mutate('Stat' = standardise(`Performance Save%`)) %>%
    mutate("2022" = playerStat(Born,Stat,2022))

  for (i in 2023:2031) {
    gkTeamData2 <- gkTeamData2 %>%
      mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
  }

  gkTeamData2 <- gkTeamData2 %>%
    group_by(Nation)%>%
    summarise_at(c('2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031'),mean) %>%
    melt(id = c('Nation'), value.name = 'PerfSavePer')

  #DEFENDERS
  dfTeamData2 <- rbind(playTournDefend, dfRarita) %>%
    filter(Pos %in% c('DF','DFFW','DFMF','FWDF','MFDF')) %>%
    select(c('Player','Born','Nation','Pressures %','Year','Pos')) %>%
    drop_na()%>%
```

```
  filter(Year == 2021)%>%
  mutate('Stat' = standardise(`Pressures %`)) %>%
  mutate("2022" = playerStat(Born,Stat,2022))

for (i in 2023:2031) {
  dfTeamData2 <- dfTeamData2 %>%
    mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
}

dfTeamData2 <- dfTeamData2 %>%
  group_by(Nation)%>%
  summarise_at(c('2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031'),mean) %>%
  melt(id = c('Nation'), value.name = 'PressurePer')

#MIDFIELDERS
mfTeamData2 <- rbind(playTournPass, mfRarita) %>%
  filter(Pos %in% c('MF','FWMF','MFDF','MFFW','DFMF')) %>%
  select(c('Player','Year','Born','Nation','Total Cmp')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('Stat' = standardise(`Total Cmp`)) %>%
  mutate("2022" = playerStat(Born,Stat,2022))

for (i in 2023:2031) {
  mfTeamData2 <- mfTeamData2 %>%
    mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
}

mfTeamData2 <- mfTeamData2 %>%
  group_by(Nation)%>%
  summarise_at(c('2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031'),mean) %>%
  melt(id = c('Nation'), value.name = 'TotalCmp')

#FORWARDS
fwTeamData2 <- rbind(playTournShoot, fwRarita) %>%
  filter(Pos %in% c('FW','FWMF','FWDF','MFFW','DFFW')) %>%
  select(c('Player','Born','Nation','Gls','Year')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('Stat' = standardise(`Gls`)) %>%
  mutate("2022" = playerStat(Born,Stat,2022))
```

```
  for (i in 2023:2031) {
    fwTeamData2 <- fwTeamData2 %>%
      mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
  }

  fwTeamData2 <- fwTeamData2 %>%
    group_by(Nation)%>%
    summarise_at(c('2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031'),mean) %>%
    melt(id = c('Nation'), value.name = 'Gls')


  predictData <- merge(merge(gkTeamData2 ,dfTeamData2, by.x = c('Nation', 'variable')),
                      merge(mfTeamData2 ,fwTeamData2, by.x = c('Nation', 'variable')), by.x = c('Nation',
'variable'))


  predictRarita <- predictData %>%
    filter(Nation == 'Rarita')  %>%
    select(-c('Nation',variable))

  resultsRarita <- predict(bagprobTT, newdata = predictRarita, type = 'prob')[,2]

  return(resultsRarita)

}
set.seed(1)
probabilityTT <- c(2022:2031)

for (i in 1:5000) {
  probabilityTT <- rbind(probabilityTT,simulationTT())
}
probabilityTT <- as.data.frame(probabilityTT)[-1,]
summary(probabilityTT)
library(resample)
colVars(as.matrix(probabilityTT[sapply(probabilityTT, is.numeric)]))*1.96*sqrt(5000)



simulationRank <- function() {
  #GOALKEEPERS
  gkTeamData2 <- rbind(playTournGoalkeep, gkRarita) %>%
    select(c('Player','Born','Nation','Performance Save%','Year')) %>%
```

```
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('Stat' = standardise(`Performance Save%`)) %>%
  mutate("2022" = playerStat(Born,Stat,2022))

 for (i in 2023:2031) {
  gkTeamData2 <- gkTeamData2 %>%
    mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
 }

 gkTeamData2 <- gkTeamData2 %>%
  group_by(Nation)%>%
  summarise_at(c('2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031'),mean) %>%
  melt(id = c('Nation'), value.name = 'PerfSavePer')

 #DEFENDERS
 dfTeamData2 <- rbind(playTournDefend, dfRarita) %>%
  filter(Pos %in% c('DF','DFFW','DFMF','FWDF','MFDF')) %>%
  select(c('Player','Born','Nation','Pressures %','Year','Pos')) %>%
  drop_na()%>%
  filter(Year == 2021)%>%
  mutate('Stat' = standardise(`Pressures %`)) %>%
  mutate("2022" = playerStat(Born,Stat,2022))

 for (i in 2023:2031) {
  dfTeamData2 <- dfTeamData2 %>%
    mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
 }

 dfTeamData2 <- dfTeamData2 %>%
  group_by(Nation)%>%
  summarise_at(c('2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031'),mean) %>%
  melt(id = c('Nation'), value.name = 'PressurePer')

 #MIDFIELDERS
 mfTeamData2 <- rbind(playTournPass, mfRarita) %>%
  filter(Pos %in% c('MF','FWMF','MFDF','MFFW','DFMF')) %>%
  select(c('Player','Year','Born','Nation','Total Cmp')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('Stat' = standardise(`Total Cmp`)) %>%
  mutate("2022" = playerStat(Born,Stat,2022))
```

```
for (i in 2023:2031) {
  mfTeamData2 <- mfTeamData2 %>%
    mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
}

mfTeamData2 <- mfTeamData2 %>%
  group_by(Nation)%>%
  summarise_at(c('2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031'),mean) %>%
  melt(id = c('Nation'), value.name = 'TotalCmp')

#FORWARDS
fwTeamData2 <- rbind(playTournShoot, fwRarita) %>%
  filter(Pos %in% c('FW','FWMF','FWDF','MFFW','DFFW')) %>%
  select(c('Player','Born','Nation','Gls','Year')) %>%
  drop_na() %>%
  filter(Year == 2021)%>%
  mutate('Stat' = standardise(`Gls`)) %>%
  mutate("2022" = playerStat(Born,Stat,2022))

for (i in 2023:2031) {
  fwTeamData2 <- fwTeamData2 %>%
    mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
}

fwTeamData2 <- fwTeamData2 %>%
  group_by(Nation)%>%
  summarise_at(c('2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031'),mean) %>%
  melt(id = c('Nation'), value.name = 'Gls')


predictData <- merge(merge(gkTeamData2 ,dfTeamData2, by.x = c('Nation', 'variable')),
           merge(mfTeamData2 ,fwTeamData2, by.x = c('Nation', 'variable')), by.x = c('Nation',
'variable'))

predictRarita <- predictData %>%
  filter(Nation == 'Rarita')  %>%
  select(-c('Nation',variable))

resultsRarita <- predict(rfprobTT, newdata = predictRarita, type = 'prob')

predictRank <- predictData %>%
```

```
    select(-c(Nation,variable))

  predictRankname <- predictData %>%
    select(c(Nation,variable))

  resultsRank <- predict(rfprob, newdata = predictRank)

  totalRanks <- cbind(predictRankname,resultsRank) %>%
    group_by(variable) %>%
    mutate(rank = dense_rank(resultsRank))


  successiteration<-totalRanks %>%
    mutate(variable = as.numeric(variable) + 2021) %>%
    as_tsibble(index = variable, key = Nation) %>%
    filter(Nation=="Rarita")%>%
    mutate(`rank`=ifelse(`rank`==1,1,0))
  successiteration<-successiteration[,4]
  return(successiteration)
}
set.seed(1)
simulationRanks <- as_tibble(2022:2031)
for (i in 1:5000) {simulationRanks<-cbind(simulationRanks,simulationRank())
}

colnames(simulationRanks)<-c("rank",1:5000)

prob1st<-simulationRanks%>%
  mutate(Prob=rowMeans(select(.,-`rank`)))%>%
  select(Prob)
prob1st



#AUTOPLOT
predictRank <- predictData %>%
  select(-c(Nation,variable))

predictRankname <- predictData %>%
  select(c(Nation,variable))

resultsRank <- predict(rfprob, newdata = predictRank)
```

```
totalRanks <- cbind(predictRankname,resultsRank) %>%
  group_by(variable) %>%
  mutate(rank = dense_rank(resultsRank)) %>%
  select(-resultsRank)


totalRanks %>%
  mutate(variable = as.numeric(variable) + 2021) %>%
  as_tsibble(index = variable, key = Nation) %>%
  autoplot(rank)



### END ### ----------------------------------------------------------


### Economy ### -----------------------------------------------------

#Population Forecast
tsPop <- ts(ecoRarPopulation$Rarita, start=2011)
mPop <- auto.arima(tsPop)
mPop %>% forecast(h=11) %>% autoplot()
fPop <- mPop %>% forecast(h=11)

#GDP Rarita Forecast
rarGDPTotal <- cbind(Year = ecoRarGDP$Year,as.data.frame(ecoRarGDP*ecoRarPopulation/10^9)[-1])
tsGDP <- ts(rarGDPTotal$Rarita,start=2011)
mGDP <- auto.arima(tsGDP)
mGDP %>% forecast(h=11) %>% autoplot()
fGDP <- mGDP %>% forecast(h=11)

#GDP per capita for Rarita
GDPc <- as.numeric(fGDP$mean*10^9/fPop$mean)
rbind(ecoRarGDP%>%select(Year, Rarita),cbind(Year = c(2021:2030),Rarita = GDPc)) %>%
  as_tsibble(index = Year) %>%
  autoplot(Rarita)


#GNI Rarita Forecast
rarGNITotal <- cbind(Year = ecoRarGNI$Year,as.data.frame(ecoRarGNI*ecoRarPopulation/10^9)[-1])
tsGNI <- ts(rarGNITotal$Rarita, start = 2011)
```

```
mGNI <- auto.arima(tsGNI)
mGNI %>% forecast(h = 11) %>% autoplot()
fGNI <- mGNI %>% forecast(h = 11)

#GNI per capita for Rarita
GNIc <- as.numeric(fGNI$mean*10^9/fPop$mean)
rbind(ecoRarGDP%>%select(Year,Rarita),cbind(Year = c(2021:2030),Rarita = GNIc)) %>%
  as_tsibble(index = Year) %>%
  autoplot(Rarita)

tsInf <- ts(ecoRarInflation$`Annual Inflation Rate`, start=1991)
mInf <- auto.arima(tsInf)
mInf %>% forecast(h=10) %>% autoplot()
fInf <- mInf %>% forecast(h=10)


fInfets <- ets(tsInf,model = "AAN") %>% forecast(h=10)
fInfets %>% forecast(h=10) %>% autoplot()

# Forecast Matchday
# Get Rarita match day revenue

R <- footballRevenueRaw %>% filter(Nation == "Rarita")
RM <- ts(as.numeric(c(R[1,3],R[1,7],R[1,11],R[1,15],R[1,19])), start = 2016)
RB <- ts(as.numeric(c(R[1,4],R[1,8],R[1,12],R[1,16],R[1,20])), start = 2016)
RC <- ts(as.numeric(c(R[1,5],R[1,9],R[1,13],R[1,17],R[1,21])), start = 2016)

TR <- ts(as.numeric(c(R[1,18],R[1,14],R[1,10],R[1,6],R[1,2])), start = 2016) # assume constant
TR2 <- ts(as.numeric(c(R[1,18],R[1,14],R[1,10],R[1,6])), start = 2016) # assume constant

NNTR <- ets(TR,model = "AAN") %>% forecast(h=11) # model for total revenue assuming no nation team

YNTR <- ets(TR,model = "AAN", alpha = 0.1) %>% forecast(h=11) # assuming national team

# add in competitive team

# get expenses

E <- footballExpenseRaw %>% filter(Nation == "Rarita")

ES <- ts(as.numeric(c(E[1,15],E[1,12],E[1,9],E[1,6],E[1,3])), start = 2016)
TE <- ts(as.numeric(c(E[1,14],E[1,11],E[1,8],E[1,5],E[1,2])), start = 2016)
```

```
NNTE <- ets(TE,model = "AAN") %>% forecast(h=11) #expenses w/o national team

YNTE <- ets(TE,model = "AAN", alpha = 0.75, beta = 0.9) %>% forecast(h=11) #expenses w/ national team

# assumption

# Additional revenues, expenses and net profit
addRev <- YNTR$mean - NNTR$mean # Additional revenue per capita
addRev <- addRev * fPop$mean

addExp <- YNTE$mean - NNTE$mean # Additional expenses per capita
addExp <- addExp * fPop$mean

addProf <- addRev[-1] - addExp[-1]
addNP <- addProf - colSums(playerSalary5[,-1])

totalProf <- (YNTR$mean[-1] - YNTE$mean[-1]) * fPop$mean[-1]

# Adding additional profit to GDP
tGDP <- fGDP$mean*10^9 + addRev - addExp
tGDPc <- as.numeric(tGDP/fPop$mean)

rbind(ecoRarGDP%>%select(Year,Rarita),cbind(Year = c(2021:2031),Rarita = tGDPc)) %>%
  as_tsibble(index = Year) %>%
  autoplot(Rarita) +
  xlab("Year") +
  ylab("GDP (Doubloons)") +
  ggtitle("Rarita GDP per capita (2011-2031)") +
  theme_bw()

# Adding additional profit to GNI
tGNI <- fGNI$mean*10^9 + addRev - addExp
tGNIc <- as.numeric(tGNI/fPop$mean)

rbind(ecoRarGDP%>%select(Year,Rarita),cbind(Year = c(2021:2031),Rarita = tGNIc)) %>%
  as_tsibble(index = Year) %>%
  autoplot(Rarita) +
  xlab("Year") +
  ylab("GNI (Doubloons)") +
  ggtitle("Rarita GNI per capita (2011-2031)") +
  theme_bw()
```

```
tsSpot <- ts(t(ecoRarSpot[2,])[-1],start=2008)
fSpot <- auto.arima(tsSpot) %>% forecast(h=10)

funding <- c(995000000,0,0,0,0,0,0,0,0,0,0)

for (x in 1:10) {
  funding[x + 1] = (funding[x] + addNP[x]) * (1 + fSpot$mean[x])
}



### END ### -----------------------------------------------------


### Salary ### ---------------------------------------------
playerNames <- c('F. Akumu', 'F. Ithungu','N. Terzi?', 'H. Zare', 'F. Yunusa', 'Q. bin Ismail', 'C.
Tukamushaba','O. Wanjala', 'X. Leroy', 'F. Chin', 'S. Barman','H. Makumbi', 'X. Thomas', 'E. Mudzingwa', 'F.
Ajio')

playTournCompiled<-
rbind(playLeagueDefend[,c('Player','Born','Year')],playLeaguePass[,c('Player','Born','Year')],playLeagueGoal
keep[,c('Player','Born','Year')],playLeagueShoot[,c('Player','Born','Year')])

playerBorn<-playTournCompiled%>%
  filter(`Player`%in% playerNames & Year == 2021)%>%
  group_by(Player)%>%
  summarise(Born=mean(Born))

salaries2021<-play2021Salary%>%
  mutate(Year=2021)

playerSalaries<-salaries2021%>%
  filter(`Player Name`%in% playerNames)%>%
  mutate(Player=`Player Name`)%>%
  select(`Player`,`Annualized Salary`)

playerSalaryData<-merge(playerBorn,playerSalaries,by = "Player")

playerSalaryData<-playerSalaryData%>%
  mutate("2022"=`Annualized Salary`)%>%
  select(-`Annualized Salary`)
```

```
playerSalaryBonus<-playerSalaryData

set.seed(4)

for (i in 2023:2031) {
  playerSalaryBonus <- playerSalaryBonus %>%
    mutate_(.dots = setNames(list(paste0("playerStat(Born,`", i - 1,"`,", i,")")), i))
}

for (i in 2022:2030){
  playerSalaryBonus<-playerSalaryBonus%>%
    mutate_(.dots=setNames(list(paste0("ifelse(`",i+1,"`/","`",i,"`>1,1,0)")),i))
}

playerSalaryBonus<-playerSalaryBonus%>%
  mutate(PB3a=rowSums(select(.,`2022`,`2023`,`2023`)),
       PB3b=rowSums(select(.,`2024`,`2025`,`2026`)),
       PB3c=rowSums(select(.,`2027`,`2028`,`2029`)),
       PB4a=rowSums(select(.,`2022`,`2023`,`2023`,`2025`)),
       PB4b=rowSums(select(.,`2026`,`2027`,`2028`,`2029`)),
       PB5=rowSums(select(.,`2022`,`2023`,`2024`,`2025`,`2026`)),
       PB6=rowSums(select(.,`2022`,`2023`,`2024`,`2025`,`2026`,`2027`)),
  )%>%
  select(-c(3:10))

InfCumulated3a <-(1+fInfets$mean[1])*(1+fInfets$mean[2])*(1+fInfets$mean[3])
InfCumulated3b <-(1+fInfets$mean[4])*(1+fInfets$mean[5])*(1+fInfets$mean[6])
InfCumulated3c <-(1+fInfets$mean[7])*(1+fInfets$mean[8])*(1+fInfets$mean[9])
InfCumulated4a <-(1+fInfets$mean[1])*(1+fInfets$mean[2])*(1+fInfets$mean[3])*(1+fInfets$mean[4])
InfCumulated4b <-(1+fInfets$mean[5])*(1+fInfets$mean[6])*(1+fInfets$mean[7])*(1+fInfets$mean[8])
InfCumulated5 <-
(1+fInfets$mean[1])*(1+fInfets$mean[2])*(1+fInfets$mean[3])*(1+fInfets$mean[4])*(1+fInfets$mean[5])
InfCumulated6 <-
(1+fInfets$mean[1])*(1+fInfets$mean[2])*(1+fInfets$mean[3])*(1+fInfets$mean[4])*(1+fInfets$mean[5])*
(1+fInfets$mean[6])

playerSalary5<-playerSalaryData%>%
  mutate("PB5"=playerSalaryBonus$PB5,"2023"=`2022`,"2024"=`2022`,"2025"=`2022`,"2026"=`2022`)%>%
  mutate("2027"=`2022`*(1+PB5*0.02)*InfCumulated5)%>%
  mutate("2028"=`2027`,"2029"=`2027`,"2030"=`2027`,"2031"=`2027`)%>%
  select(-Born,-PB5)
```

```
sum(playerSalary5[2:11])

# 3 year contract
playerSalary3<-playerSalaryData%>%

mutate("PB3a"=playerSalaryBonus$PB3a,"PB3b"=playerSalaryBonus$PB3b,"PB3c"=playerSalaryBonus$P
B3c,"2023"=`2022`,"2024"=`2022`) %>%
  mutate("2025"=`2022`*(1+PB3a*0.02)*InfCumulated3a)%>%
  mutate("2026"=`2025`,"2027"=`2025`) %>%
  mutate("2028"=`2027`*(1+PB3b*0.02)*InfCumulated3b) %>%
  mutate("2029"=`2028`,"2030"=`2028`)%>%
  mutate("2031"=`2030`*(1+PB3c*0.02)*InfCumulated3c) %>%
  select(-Born,-PB3a,-PB3b,-PB3c)
sum(playerSalary3[2:11])
# 4 year contract

playerSalary4<-playerSalaryData%>%

mutate("PB4a"=playerSalaryBonus$PB4a,"PB4b"=playerSalaryBonus$PB4b,"2023"=`2022`,"2024"=`2022`,
"2025"=`2022`) %>%
  mutate("2026"=`2022`*(1+PB4a*0.02)*InfCumulated4a)%>%
  mutate("2027"=`2026`,"2028"=`2026`,"2029"=`2026`) %>%
  mutate("2030"=`2029`*(1+PB4b*0.02)*InfCumulated4b) %>%
  mutate("2031"=`2030`) %>%
  select(-Born,-PB4a,-PB4b)
sum(playerSalary4[2:11])

# 6 year contract
playerSalary6<-playerSalaryData%>%

mutate("PB6"=playerSalaryBonus$PB6,"2023"=`2022`,"2024"=`2022`,"2025"=`2022`,"2026"=`2022`,"2027"
=`2022`)%>%
  mutate("2028"=`2022`*(1+PB6*0.02)*InfCumulated6)%>%
  mutate("2029"=`2028`,"2030"=`2028`,"2031"=`2028`)%>%
  select(-Born,-PB6)
sum(playerSalary6[2:11])

### END ### ----------------------------------------------------
```