Georgia Tech Institute of Technology
Bayesian Statistics - ISyE 6420

Course Project

Genci Ymeri

Materializing on Sports with the Help of the Bayesian Statistics.

0.1 Introduction:

Technological breakthroughs have historically helped the democratisation of monopolistic controlled power and resources. For example, the inventions of electricity, the printing press, computers, on-line shopping, etc. have provided economic opportunities for individual levels, making it possible to enrich millions of individuals like never even thought of before.

Nowadays, billions of people can not only watch their favorite sports from their couch, but with a tap of a button, can get any historical or current information they desire about any player from any sport.

Similar ideas of making use of data analytics to improve games have been explored and applied to be applied for more than a decade by now. A prime example of this is the non-fictional story written in the ”***Moneyball: The Art Of Winning An Unfair Game***” book by Michael Lewis which was later made a movie. (one of my favorite ones) ”***Moneyball”***.

As may be expected, in this story, a baseball coach hired a data analyst to make decisions for which player to play in which position. While the coach himself and his assistants were biased towards certain players in evaluating their skills, the hired data analysts let the data do the talking:

A picture from the ”Moneyball” movie, which is based on a real life story where using data to hire the right players for the right positions brought the desired and predicted success.

In this movie story, the data analyst used a simple linear regression *(more in details can be found in this link - https://towardsdatascience.com/moneyball-linear-regression-76034259af5e)* and his knowledge to serve a single team.

The the question is, how can this be applied to all teams/all people?

Fortunately, in the end of our ISyE 6420 course, we should be able to turn the challenge up a notch, and do just that with the help of the Bayesian statistics.

In this project, we will create the basis for a mobile app, which after analysing the team players of any team or any sport, can give us multitudes of probabilities of which fans are very much interested in, let's say: the probability of a certain team to be in the top 3, or what the probability is of being ranked the 4th team, and many more.

User Case: Lyon and its new key player Xherdan Shaqiri

Olympique Lyonnais, commonly referred to simply as Lyon, is a French professional football club based in Lyon in Auvergne-Rhône-Alpes. It plays in France's highest football division, Ligue 1 (French Premier League). Founded in 1899, the club won its first Ligue 1 championship in 2002, starting a national record-setting streak of seven successive titles. Lyon is one of the most supported clubs in France, and they have recently been struggling to maintain their position as 'the best'. Last year, they only finished as 7th in the league.

At the beginning of the 2020/21 season, fans believed that the club would finish in the Top 3 of the league, with only a probability of 0.2, or 4th place with the probability of 0.3, and outside of Top 4 with the probability of 0.5

These probabilities hold true only if the club's new key player, Xherdan Shaqiri, is not badly injured throughout the season. The probability of a severe injury occurring to this player is 0.3. If an injury happens, the probabilities for the above end of season positions are 0.1, 0.2, and 0.7 respectively.

If the team finishes in the Top 3, they will be qualified for next year's Ligue 1 Group Stage. If they finish 4th, they will have to go through Champions League Qualification Round, so the probability of them proceeding to the Champions League Group Stage is reduced to 0.7.

Lyon is also competing in the Europa League. As a former winner, they have a 0.6 probability of winning the Europa League again if Shaqiri is in good shape. Otherwise, the probability is reduced to 0.4. If they win the Europa League, they are guaranteed to have a place in next year's Champions League Group Stage with 0.99 probability, even if they don't get into Top 3, unless UEFA changes the rules for next year.

If the club is qualified for next year's Champions League Group Stage, they will have a 0.7 chance to sign one of their top transfer targets, Granit Xhaka from Premier League club Arsenal. If not, the chance to sign Xhaka is reduced to 0.4.

Here is a list of questions fans may ask:

- What is the probability that Shaqiri is injured?

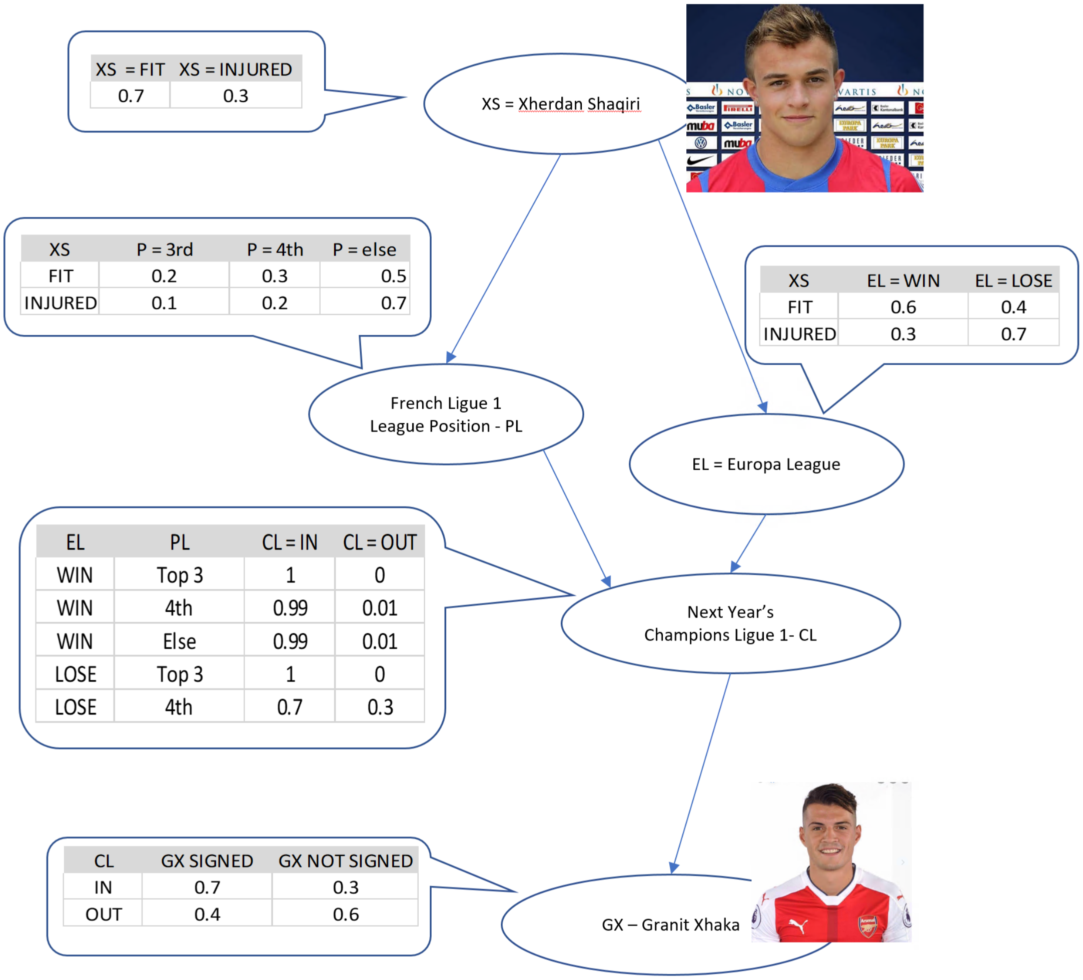
- What is the probability that the club was qualified for Ligue 1?
- What is the probability it finished in Top 3?
- What is the probability it finished 4th?
- What is the probability that the club won the Europa League?

0.2 User case implementation

We are going to make use of Bayesian network to help us implementing this user case. As a reminder (*ref: https://en.wikipedia.org/wiki/Bayesian_network*),

A Bayesian network (also known as a Bayes network, Bayes net, belief network, or decision network) is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG).

Bayesian networks are ideal for taking events that occur, and distinguishing a single contributing factor which is most likely the cause of the loss. One of several possible known causes was the contributing factor ."**Oriented**" which means that the arrows connecting the nodes of the graph are oriented (i.e. has directions) so the conditional dependencies have a ***one-way direction.*** "**Acylic**" means that it is ***impossible to loop through*** the network. It can only go up or down.



XS = FIT	XS = INJURED
0.7	0.3

XS	P = 3rd	P = 4th	P = else
FIT	0.2	0.3	0.5
INJURED	0.1	0.2	0.7

XS	EL = WIN	EL = LOSE
FIT	0.6	0.4
INJURED	0.3	0.7

EL	PL	CL = IN	CL = OUT
WIN	Top 3	1	0
WIN	4th	0.99	0.01
WIN	Else	0.99	0.01
LOSE	Top 3	1	0
LOSE	4th	0.7	0.3

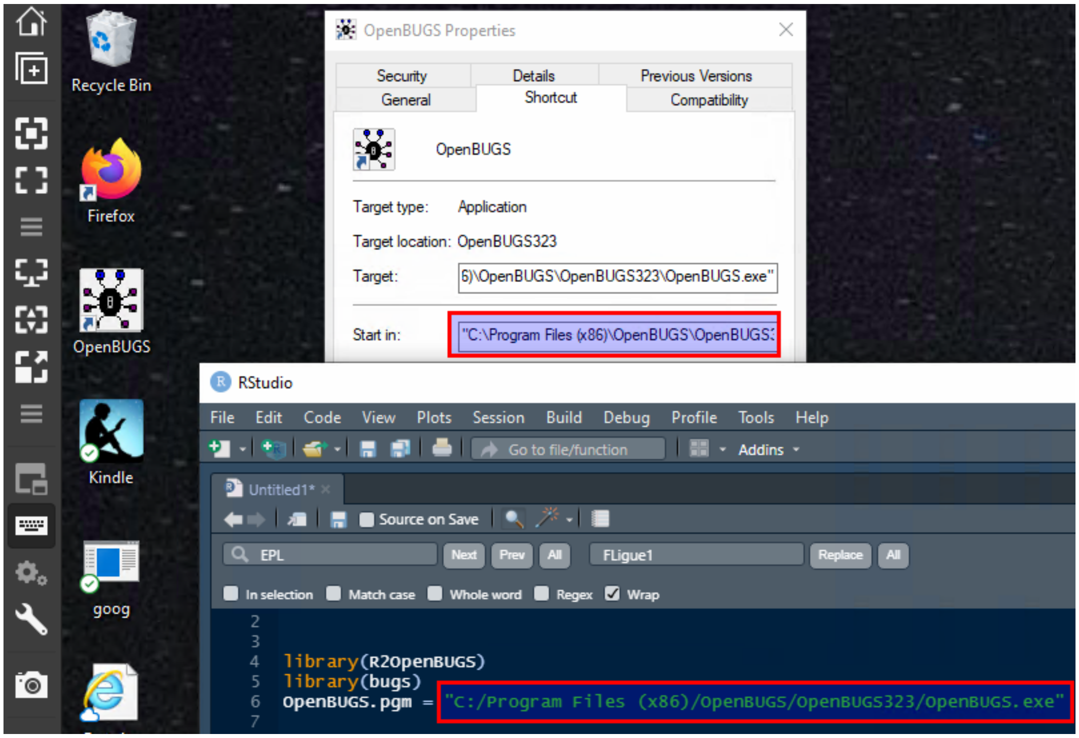
CL	GX SIGNED	GX NOT SIGNED
IN	0.7	0.3
OUT	0.4	0.6

Bayesian Network (DAG) for our User-Case.

Let's write down some quick notations:

- **FLigue1**: French League 1 (French Premier league)
- **XS:** Xherdan Shaqiri
- **GX** : Granit Xhaka
- **Europa**: Europa League

Note*: *One nice trick I learned lately is the use of the "R2OpenBugs" library which makes possible to write the code in RStudio and automatically calls and runs the code in OpenBugs. First we need to install "R2OpenBugs" if not priviously installed, and then call it as a normal library.*



Target type: Application

Target location: OpenBUGS323

Target: 6)\OpenBUGS\OpenBUGS323\OpenBUGS.exe"

Start in: "C:\Program Files (x86)\OpenBUGS\OpenBUGS3

EPL FLigue1

□ In selection □ Match case □ Whole word □ Regex ☑ Wrap

```
library(R2OpenBUGS)
library(bugs)
OpenBUGS.pgm = "C:/Program Files (x86)/OpenBUGS/OpenBUGS323/OpenBUGS.exe"
```

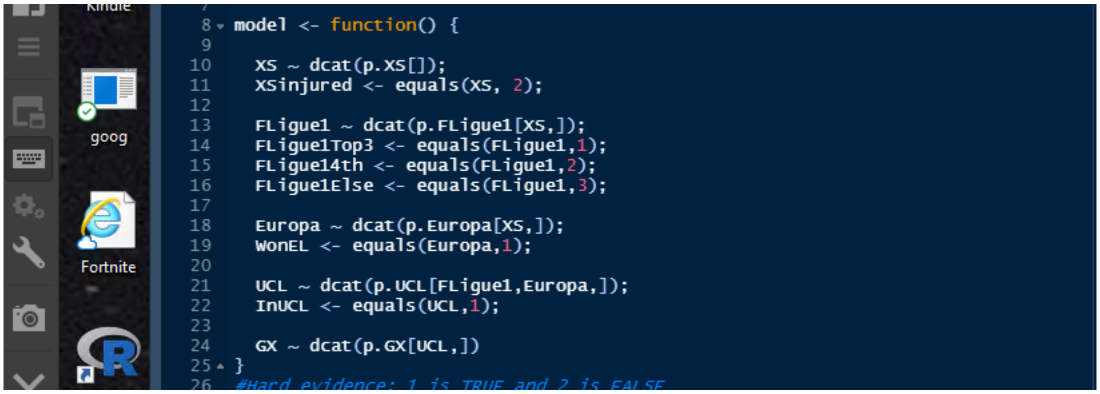
User's Case Bayesian Network - DAG.

Also, as a reminder in OpenBugs, counting starts from 1, so the mapping (Fit, Injured) goes as:

(Fit, Injured) == > (1, 2). Similarly, other variables are mapped.

How well Lyon will do in the French Ligue 1 (FLigue1) depends on the state of Xherdan Shaqiri, "XR" variable, which is included when the FLigue1 variable is defined. Other dependencies follow similar logic.

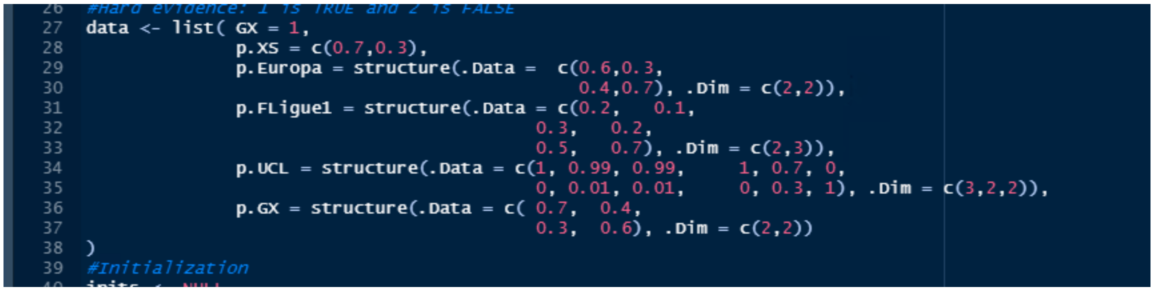
We will define our model as:



```
model <- function() {

  XS ~ dcat(p.XS[]);
  XSinjured <- equals(XS, 2);

  FLigue1 ~ dcat(p.FLigue1[XS,]);
  FLigue1Top3 <- equals(FLigue1,1);
  FLigue14th <- equals(FLigue1,2);
  FLigue1Else <- equals(FLigue1,3);

  Europa ~ dcat(p.Europa[XS,]);
  WonEL <- equals(Europa,1);

  UCL ~ dcat(p.UCL[FLigue1,Europa,]);
  InUCL <- equals(UCL,1);

  GX ~ dcat(p.GX[UCL,])
}
#Hard evidence: 1 is TRUE and 2 is FALSE
```

Model definition.

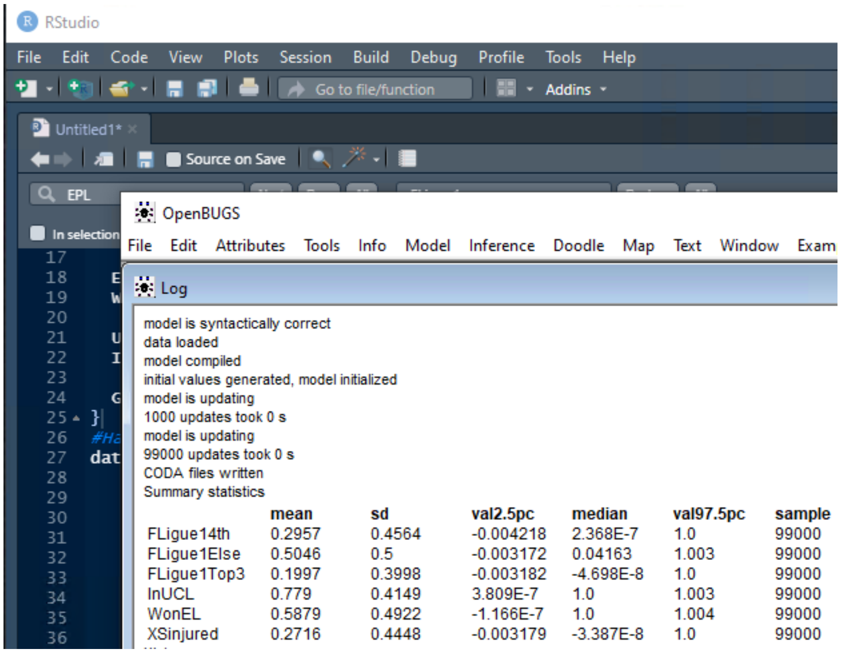
Then, we assign the data as given in the description above of our user case, into our variables as shown below:



```
#Hard evidence: 1 is TRUE and 2 is FALSE
data <- list( GX = 1,
              p.XS = c(0.7,0.3),
              p.Europa = structure(.Data =  c(0.6,0.3,
                                              0.4,0.7), .Dim = c(2,2)),
              p.FLigue1 = structure(.Data = c(0.2,   0.1,
                                            0.3,   0.2,
                                            0.5,   0.7), .Dim = c(2,3)),
              p.UCL = structure(.Data = c(1, 0.99, 0.99,      1, 0.7, 0,
                                          0, 0.01, 0.01,      0, 0.3, 1), .Dim = c(3,2,2)),
              p.GX = structure(.Data = c( 0.7,  0.4,
                                          0.3,  0.6), .Dim = c(2,2))
)
#Initialization
inits <- NULL
```

Assigning the given data.

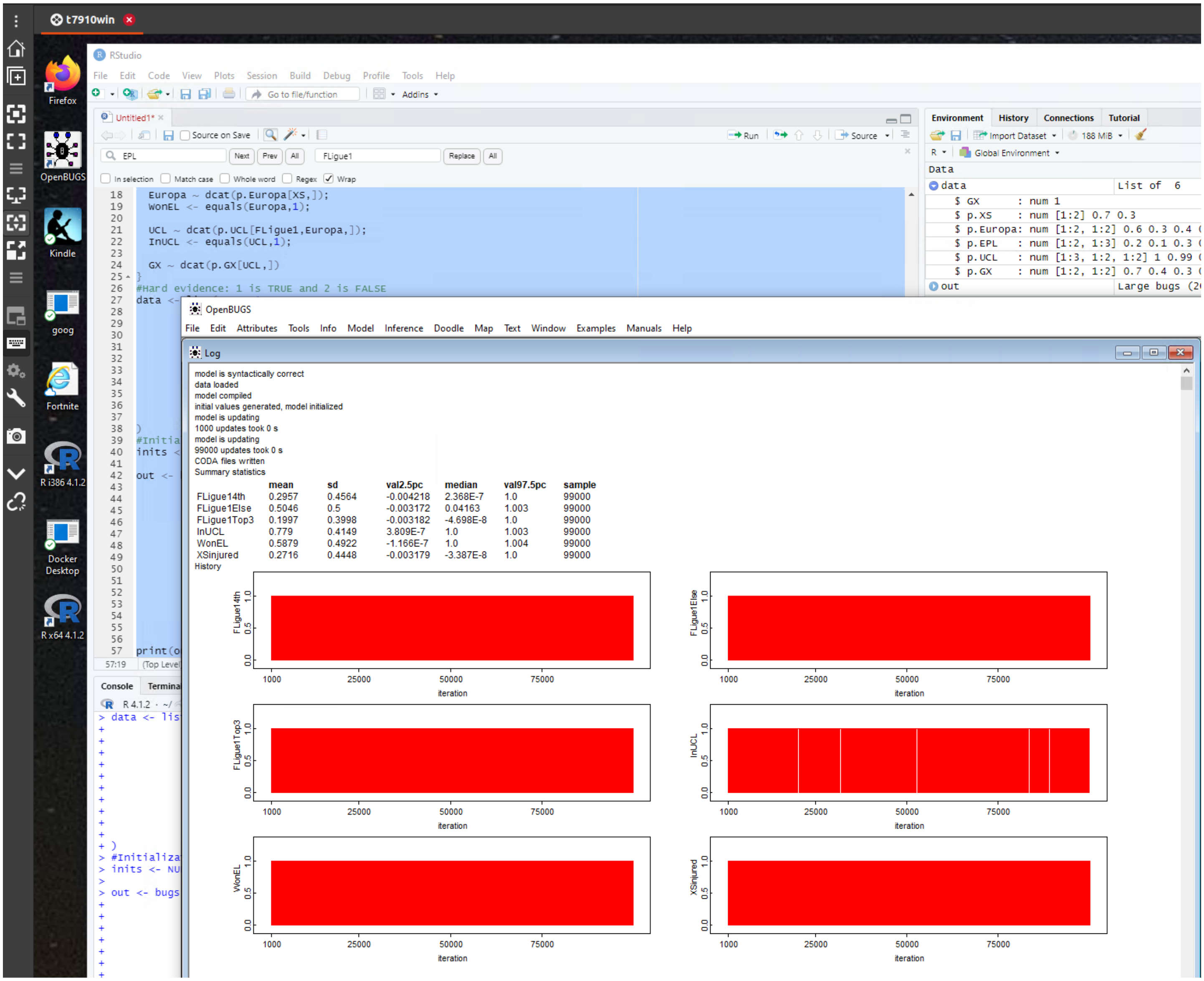
Then we select the entire code (***Appendix A)*** and click the run button, to have RStudio launch OpenBugs and we get these results as shown below.



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled1*
Source on Save
EPL
In selection
OpenBUGS
File Edit Attributes Tools Info Model Inference Doodle Map Text Window Exam
Log

```
model is syntactically correct
data loaded
model compiled
initial values generated, model initialized
model is updating
1000 updates took 0 s
model is updating
99000 updates took 0 s
CODA files written
Summary statistics
             mean     sd       val2.5pc   median     val97.5pc  sample
FLigue14th   0.2957   0.4564   -0.004218  2.368E-7   1.0        99000
FLigue1Else  0.5046   0.5      -0.003172  0.04163    1.003      99000
FLigue1Top3  0.1997   0.3998   -0.003182  -4.698E-8  1.0        99000
InUCL        0.779    0.4149   3.809E-7   1.0        1.003      99000
WonEL        0.5879   0.4922   -1.166E-7  1.0        1.004      99000
XSinjured    0.2716   0.4448   -0.003179  -3.387E-8  1.0        99000
```

Results.



```
  Europa ~ dcat(p.Europa[XS,]);
  WonEL <- equals(Europa,1);

  UCL ~ dcat(p.UCL[FLigue1,Europa,]);
  InUCL <- equals(UCL,1);

  GX ~ dcat(p.GX[UCL,])
}
#Hard evidence: 1 is TRUE and 2 is FALSE
data <-
```

```
)
#Initia
inits <

out <-
```

```
print(o
```

```
Data
data                                  List of  6
   $ GX      : num 1
   $ p.XS    : num [1:2] 0.7 0.3
   $ p.Europa: num [1:2, 1:2] 0.6 0.3 0.4 (
   $ p.EPL   : num [1:2, 1:3] 0.2 0.1 0.3 (
   $ p.UCL   : num [1:3, 1:2, 1:2] 1 0.99 (
   $ p.GX    : num [1:2, 1:2] 0.7 0.4 0.3 (
out                                   Large bugs (2
```

```
> data <- lis
+
+
+
+
+
+
+
+
+
+
+ )
> #Initializa
> inits <- NU
>
> out <- bugs
+
+
+
+
+
+
+
```

```
model is syntactically correct
data loaded
model compiled
initial values generated, model initialized
model is updating
1000 updates took 0 s
model is updating
99000 updates took 0 s
CODA files written
Summary statistics
             mean     sd       val2.5pc   median     val97.5pc  sample
FLigue14th   0.2957   0.4564   -0.004218  2.368E-7   1.0        99000
FLigue1Else  0.5046   0.5      -0.003172  0.04163    1.003      99000
FLigue1Top3  0.1997   0.3998   -0.003182  -4.698E-8  1.0        99000
InUCL        0.779    0.4149   3.809E-7   1.0        1.003      99000
WonEL        0.5879   0.4922   -1.166E-7  1.0        1.004      99000
XSinjured    0.2716   0.4448   -0.003179  -3.387E-8  1.0        99000
History
```

A full picture of the our implementation results.

Given the event that Granit Xhaka signed with Lyon, the probability that Lyon for next year's Champions League is 77.8 %. The probability that they may win the Europa League also increases to 58.7%.

The probability of Lyon to get to Top 3 or 4th place in the Premier League is also marginally higher; at 20% and 29.6%. Noticeably, the probability that Xherdan Shaqiri is injured is lowered to 27% (*a bit down from the 30% from the initial data*).

But this is just the tip of the iceberg what we can do with with help of Bayes analysis. Here is another feature we can add to our app, how about using Bayes Analysis on the feedback data given by ex-players' or experts' to help their favorite team, and yet e***liminate as much as possible their personal bias***?

Reducing Bias Using Bayesian Statistics on Survey Data.

Bayesian analysis on survey data been used very successfully for a long period of time ***eliminating bias much better*** than the fractionists.

In our example, we will show how to use/adapt Bayes Analysis on Survey Data for our application.

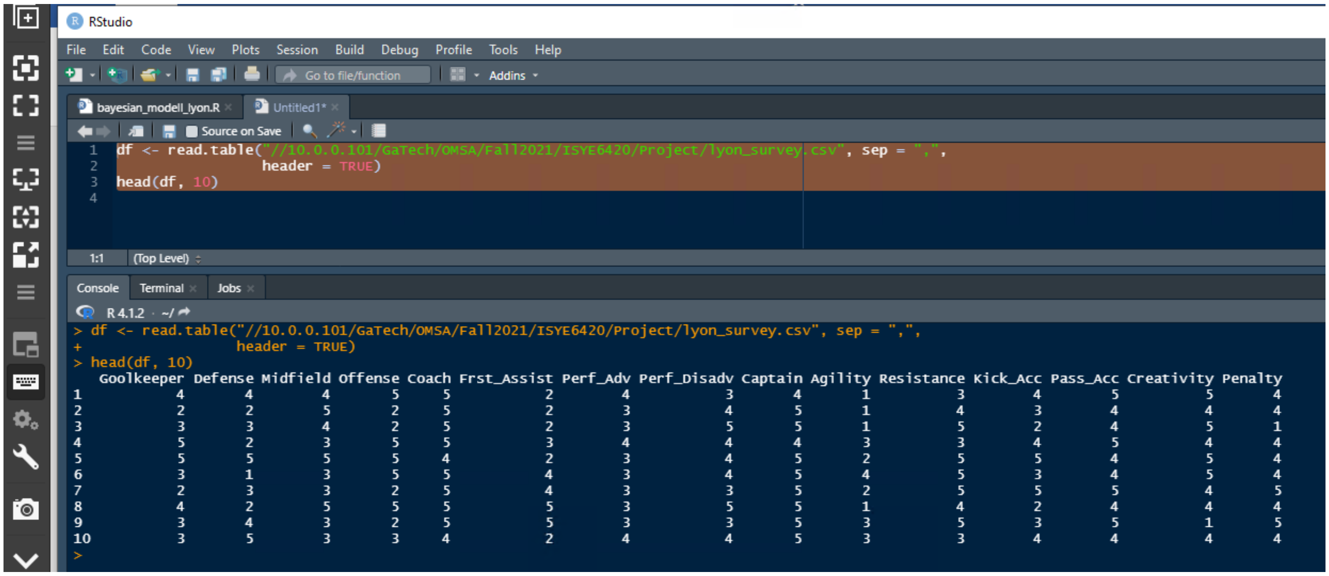
Our ”Experts' feedback” on the online questionnaire contains m = 15 questions, with responses on a five-point scale ranging from 1 (poor performance) to 5 (Messi alike), where the specific interpretation of responses are question dependent.

The team areas to be rated per their performance as follows:

1. Goalkeeper
2. Defense
3. Midfield
4. Offense
5. Coach
6. Coach 1st assistant
7. Team performance when in disadvantage
8. Team performance when in advantage
9. Captain
10. Team agility
11. Team resistance in overtime
12. Kicking accuracy
13. Passing accuracy
14. Team's game creativity
15. Penalty accuracy

For our example we will assume that the data () was collected from top experts who were ex-players, coaches, or experts in the football (soccer) field. (*in our case, for implementation purposes, the data is collected from people/friends of mine who have followed soccer since their childhood, n = 50*) .

Let's take a peak at the data.



```
df <- read.table("//10.0.0.101/GaTech/OMSA/Fall2021/ISYE6420/Project/lyon_survey.csv", sep = ",",
                 header = TRUE)
head(df, 10)
```

```
> df <- read.table("//10.0.0.101/GaTech/OMSA/Fall2021/ISYE6420/Project/lyon_survey.csv", sep = ",",
+                  header = TRUE)
> head(df, 10)
   Goolkeeper Defense Midfield Offense Coach Frst_Assist Perf_Adv Perf_Disadv Captain Agility Resistance Kick_Acc Pass_Acc Creativity Penalty
1           4       4        4       5     5           2        4           3       4       1          3        4        5          5       4
2           2       2        5       2     5           2        3           4       5       1          4        3        4          4       4
3           3       3        4       2     5           2        3           5       5       1          5        2        4          5       1
4           5       2        3       5     5           3        4           4       4       3          3        4        5          4       4
5           5       5        5       5     4           2        3           4       5       2          5        5        4          5       4
6           3       1        3       5     5           4        3           4       5       4          5        3        4          5       4
7           2       3        3       2     5           4        3           3       5       2          5        5        5          4       5
8           4       2        5       5     5           5        3           5       5       1          4        2        4          4       4
9           3       4        3       2     5           5        3           3       5       3          5        3        5          1       5
10          3       5        3       3     4           2        4           4       5       3          3        4        4          4       4
>
```

Survey data about Lyon performance.

We are going to use MCMC simulation, using WinBUGS with a burn-in period of 1000 iterations. After the simulation (***code in Apendix B***), the data will look similar like this table below:

	Sample Mean	Posterior Mean
mu1	3.52	3.5552
mu2	3.70	3.922
mu3	3.94	4.137
mu4	3.42	3.5226
mu5	4.48	4.6592
mu6	3.58	3.8664
mu7	3.48	3.6192
mu8	4.00	4.4
mu9	4.52	5.1528
mu10	2.44	2.5376
mu11	3.98	4.6566
mu12	3.48	3.7932
mu13	4.32	4.8384
mu14	4.36	4.578
mu15	4.26	4.7712
mu	3.83	4.13

Survey posterior data.

We observe that nearly all of the posterior means of the μ_i exceed the corresponding sample means of the individual questions. This suggests that the ratings of the Lyon team can be viewed more favourably (i.e. higher scores) once the personality traits of the surveyors have been removed.

The highest posterior mean was recorded for the 9th question, which asked about ***“the Captain” performance.*** This definitely makes sense.
The smallest mean was recorded for the 10th ”**Team Agility**”. This may probably be less understood, or is a redundant piece of information.

It is also beneficial to look at the posterior mean of the variance-covariance matrix, which describes the relationships amongst the m = 15 survey questions. The largest correlation value(0.61), occurred between survey questions 13 and 15.

However, the elimination of questions on the basis of redundancy should not be done solely on the basis of high correlations. In addition to high correlations, we should also have similar posterior means, e.g.:

”Passing accuracy”, and ”Penalty accuracy”, with posterior means (4.83 & 4.77) seem to be highly correlated, and we should feel safe to drop one of them (”Penalty accuracy”). Furthermore, from the common sense point, ”Accurate passing” is almost the same as ”Accurate penalty”.

We can apply same Bayesian code-template to different surveys to help us with our application.

Goodness-of-Fit

In Bayesian statistics, the assessment of goodness/of fit in complex models is problematic. A possible explanation for this is that aposteriori testing of model adequacy is not a Bayesian construct, and may be seen as violating the Bayesian paradigm. From the point of view of a subjective Bayesian purist, any uncertainty concerning a model ought to be expressed via prior opinion. In a practical point of view, it is typically difficult/impossible to determine the space of possible sampling models and parameters, and to assign prior opinion to the space.

Conclusion

Bayesian statistics, combined with modern technological breakthroughs, can help us spread the ability to more accurate investing, being pro-active, understanding, or even predicting what is most likely going to happen in specified fields on an individual level - in our project case: Soccer Analytic.

Overall, such knowledge results in a never-ending cycle of improvement for all.

References

- https://totalfootballanalysis.com/data-analysis/major-league-soccer-2019-analyzing-trends-in-the-mls-data-analysis-statistics
- https://www.whoscored.com/Statistics
- http://www.datatales.eu/moneyball-statistics-data-science-sports/
- BAYESIAN METHODS AND APPLICATIONS
  http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=637A9749A82B8E56829CD30DC320832B?doi=10.1.1.668.4031&rep=rep1&type=pdf

Appendix A

Implementation of a user case

```
library(R2OpenBUGS)
library(bugs)
OpenBUGS.pgm = ”C:/Program Files (x86)/OpenBUGS/OpenBUGS323/OpenBUGS.exe”
```

```
model <- function() {

  XS ~ dcat(p.XS[]);
  XSinjured <- equals(XS, 2);

  FLigue1 ~ dcat(p.FLigue1[XS,]);
  FLigue1Top3 <- equals(FLigue1,1);
  FLigue14th <- equals(FLigue1,2);
  FLigue1Else <- equals(FLigue1,3);

  Europa ~ dcat(p.Europa[XS,]);
  WonEL <- equals(Europa,1);

  UCL ~ dcat(p.UCL[FLigue1,Europa,]);
  InUCL <- equals(UCL,1);

  GX ~ dcat(p.GX[UCL,])
}
#Hard evidence: 1 is TRUE and 2 is FALSE
data <- list( GX = 1,
  p.XS = c(0.7,0.3),
  p.Europa = structure(.Data =  c(0.6,0.3,
                                  0.4,0.7), .Dim = c(2,2)),
  p.FLigue1 = structure(.Data = c(0.2,    0.1,
                                  0.3,    0.2,
                                  0.5,    0.7), .Dim = c(2,3)),
  p.UCL = structure(.Data = c(1, 0.99, 0.99,     1, 0.7, 0,
                              0, 0.01, 0.01,     0, 0.3, 1), .Dim = c(3,2,2)),
  p.GX = structure(.Data = c( 0.7,  0.4,
                              0.3,  0.6), .Dim = c(2,2))
)
#Initialization
inits <- NULL

out <- bugs(data = data,
     inits = inits,
     parameters.to.save = c(”XSinjured”, ”FLigue1Top3”, ”FLigue14th”,
         ”FLigue1Else”, ”WonEL”, ”InUCL”),
     model.file = model,
     digits = 5,
     n.chains = 1,
     n.burnin = 1000,
     n.iter = 100000,
     OpenBUGS.pgm=OpenBUGS.pgm,
     WINE = WINE,
     WINEPATH = WINEPATH,
     useWINE=F,
     debug = T,
     working.directory=getwd(),
     DIC = F)
```

```
print(out$summary)
```

Appendix B

OpenBugs Model for Lyon survey

```
model
{

alpha[1]<- -5
alpha[6]<-10
alpha[2]<- 1.5
alpha[3]<- 2.5
alpha[4]<-3.5
alpha[5]<-4.5
for(i in 1:n){
        for(j in 1:m)
        {
                lo[i,j]<-((alpha[x[i,j]]-3)/b[i])+3 -a[i]
                up[i,j]<-((alpha[x[i,j]+1]-3)/b[i])+3 -a[i]
        }
}
# Prior for mu
for (i in 1:m) {
mu[i] ~ dunif(0,6)
}

for(i in 1:n) {
        y[i,1:m] ~ dmnorm(mu[] , G[,])I(lo[i,],up[i,])
}

# Prior for variance-covariance

G[1:m,1:m] ~ dwish(R[,],m)

varcov[1:m,1:m] <- inverse(G[,])
for(j in 1:m)
{
        cor[j,j] <- varcov[j,j]
}
for(i in 1:m-1)
{
        for(j in i+1:m)
        {

                cor[i,j]<- varcov[i,j]/(sqrt(varcov[i,i]*varcov[j,j]))
                cor[j,i]<-cor[i,j]
        }
}

# DP Priors
for ( i in 1:n) {
        a[i]<-aa[i,1]
        b[i]<-(aa[i,2])
}
for ( j in 1:n) {
        for ( kk in 1:2) {
                aa[j,kk]<- theta1[latent[j],kk]
        }
}

for (i in 1:n) {
        latent[i]~dcat(pi[1:L1])
}

pi[1]<-r[1]

for ( j in 2:(L1-1)) {
        log(pi[j])<-log(r[j])+sum(R1[j,1:j-1])

        for ( l in 1:j-1) {
                R1[j,l]<-log(1-r[l])
        }
}

pi[L1]<-1-sum(pi[1:(L1-1)])

for ( j in 1:L1) {
        r[j]~dbeta(1,mm)
}

for ( i in 1:L1) {
        theta1[i,1:2]~dmnorm(zero[1:2],Sab[1:2,1:2])I(LB[],)
}

zero[1] <- 0; zero[2] <- 1
Sab[1:2,1:2] ~ dwish(Omega[1:2,1:2], 2)
varcovab[1:2,1:2] <- inverse(Sab[,])
corab<- varcovab[1,2]/sqrt(varcovab[1,1]*varcovab[2,2])
mm~dunif(0.4,10)
for( i in 1:n) {
        for (j in (i+1):n-1) {
                equalsmatrix[i,j]<-equals(aa[i,1],aa[j,1])*equals(aa[i,2],aa[j,2])
                equalsmatrix[j,i]<-equalsmatrix[i,j]
                }
        }
}
```

Data simulation

```
data=matrix(scan(”//10.0.0.101/GaTech/OMSA/Fall2021/ISYE6420/Project/lyon_survey.csv”
s=vector(length=50)
mu=vector(length=15)
s=mean(data.frame(t(data)))
for (i in 1:20){
  sigma = matrix(c(4,2,2,4),2,2)
  n=50 # no. of subjects
  m=15 # no.of questions
  L1=30 # truncation in stick breaking method
  mumu=matrix(ncol=15,nrow=n)
  theta1=matrix(ncol=2,nrow=L1)
  R1=matrix(ncol=L1-2,nrow=L1)
  aa=matrix(ncol=2,nrow=n)
  v=c(1:30)
  a=vector(length=n)
  latent=vector(length=n)
  b=a
  zero=vector(length=2)
  r=vector(length=L1)
  mm=runif(1,0.4,10)
  for (i in 1:m){
    mu[i]=runif(1,2,5)
  }
  G = matrix(ncol=15,nrow=15)
  for (i in 1:m){
    G[i,i]=4
  }
  for (i in 1:(m-1)){
    for (j in (i+1):m){
      G[i,j]=2
      G[j,i]=2
    }}
  Sab = matrix(c(.01,0,0,.01),2,2)
  zero[1] <- 0; zero[2] <- 1
  for ( i in 1:L1) {
    theta1[i,1:2]<-mvrnorm(1,zero[1:2],Sab[1:2,1:2])
  }

  for ( j in 1:L1) {
    r[j]<- rbeta(1,1,mm)
  }
  pi[1]<-r[1]
  R1[2,1]<-log(1-r[1])
  for ( j in 2:(L1-1)) {
    for ( l in 1:j-1) {
      R1[j,l]<-log(1-r[l])
    }
  }
  for ( j in 2:(L1-1)) {
    pi[j]<-exp(log(r[j])+sum(R1[j,1:j-1]))
  }
  pi[L1]<-1-sum(pi[1:(L1-1)])
  for (i in 1:n) {
    latent[i]<-sample(v,1,replace=TRUE, pi)
  }
  for ( j in 1:n) {
    for ( kk in 1:2) {
      aa[j,kk]<- theta1[latent[j],kk]
    }}
  for ( i in 1:n) {
    a[i]<-aa[i,1]
    b[i]<-(aa[i,2])
  }
  z=matrix(ncol=m, nrow=n)
  y=matrix(ncol=m, nrow=n)
  z= mvrnorm(n=n,mu[], G)
  for (i in 1:n){
    for (j in 1:m){
      y[i,j] = b[i]*(z[i,j]+a[i]-3)+3
    }}
  x=matrix(ncol=m, nrow=n)
  alpha=c(-10,1.5,2.5,3.5,4.5,15)


  # converting latent matrix to observed data
  for (i in 1:n){
    for (j in 1:m){
      x[i,j] = ifelse(y[i,j]<(alpha[2]),1,x[i,j])
      x[i,j] = ifelse(y[i,j]>(alpha[2]) & y[i,j]<(alpha[3]),2,x[i,j])
      x[i,j] = ifelse(y[i,j]>(alpha[3]) & y[i,j]<(alpha[4]),3,x[i,j])
      x[i,j] = ifelse(y[i,j]>(alpha[4]) & y[i,j]<(alpha[5]),4,x[i,j])
      x[i,j] = ifelse(y[i,j]>(alpha[5]),5,x[i,j])
    }}
  s=rbind(s,mean(data.frame(t(x))))
}
```