Sport Betting Presentation

author: Group 6 date: autosize: true

```
install.packages('dplyr')
```

```
## Installing package into '/usr/local/lib/R/site-library'
## (as 'lib' is unspecified)
```

```
install.packages('dummies')
```

```
## Installing package into '/usr/local/lib/R/site-library'
## (as 'lib' is unspecified)
```

```
install.packages('lpSolveAPI')
```

```
## Installing package into '/usr/local/lib/R/site-library'
## (as 'lib' is unspecified)
```

a) Model relative strengths of the teams

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df <- read.csv('RegularSeasonData.csv', fileEncoding = 'latin1')
teams = sort(unique(df$home))
df_ot = df %>% filter(ot==1)

df<- df%>% mutate(h = ifelse(ot==1,pmin(h,v),h), v = ifelse(ot==1,pmin(h,v),v))

df1 = df %>% select(home, visitor, h) %>% rename(Attacker = home, Defender = visitor, Goals = h) %>% mu
df2 = df %>% select(home, visitor, v) %>% rename(Attacker = visitor, Defender = home, Goals = v) %>% mu
data = df1 %>% rbind(df2)
library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
data2 = dummy.data.frame(data,names = c("Attacker","Defender"), sep="_")
glm_model = glm(data=data2, formula = Goals ~ -1+., family = poisson())
```

```
glm_model
```

```
##
## Call:  glm(formula = Goals ~ -1 + ., family = poisson(), data = data2)
##
## Coefficients:
##    Attacker_Ässät      Attacker_HIFK       Attacker_HPK
```

```
##          0.80812          0.77720          0.68236
##    Attacker_Ilves   Attacker_Jukurit       Attacker_JYP
##          0.80918          0.60504          0.95279
##    Attacker_KalPa    Attacker_Kärpät    Attacker_KooKoo
##          0.65263          0.97122          0.67446
##    Attacker_Lukko  Attacker_Pelicans     Attacker_SaiPa
##          0.62337          0.80403          0.68889
##    Attacker_Sport   Attacker_Tappara       Attacker_TPS
##          0.74640          0.73109          0.93071
##    Defender_Ässät      Defender_HIFK       Defender_HPK
##          0.19937         -0.17447          0.06781
##    Defender_Ilves   Defender_Jukurit       Defender_JYP
##          0.25927          0.11891          0.02168
##    Defender_KalPa    Defender_Kärpät    Defender_KooKoo
##         -0.09860         -0.09561          0.28237
##    Defender_Lukko  Defender_Pelicans     Defender_SaiPa
##         -0.05286          0.17050          0.09067
##    Defender_Sport   Defender_Tappara       Defender_TPS
##          0.33720         -0.14036                 NA
##             Home
##          0.16295
##
## Degrees of Freedom: 900 Total (i.e. Null);  870 Residual
## Null Deviance:       2521
## Residual Deviance: 926   AIC: 3291
```

Since the Defender_TPS coefficient is currently NA, we re-regress the model without this parameter. The result is as followed:

```
glm_fixed = glm(data=data2, formula = Goals ~ -1+.-Defender_TPS, family=poisson())
glm_fixed$coefficients
```

```
##    Attacker_Ässät     Attacker_HIFK      Attacker_HPK    Attacker_Ilves
##        0.80811993        0.77719578        0.68235582        0.80917525
##  Attacker_Jukurit      Attacker_JYP    Attacker_KalPa   Attacker_Kärpät
##        0.60504042        0.95279308        0.65262828        0.97121635
##   Attacker_KooKoo    Attacker_Lukko Attacker_Pelicans    Attacker_SaiPa
##        0.67446285        0.62337371        0.80402755        0.68888973
##    Attacker_Sport  Attacker_Tappara      Attacker_TPS    Defender_Ässät
##        0.74639756        0.73109391        0.93070865        0.19936543
##     Defender_HIFK      Defender_HPK    Defender_Ilves  Defender_Jukurit
##       -0.17447225        0.06781017        0.25927475        0.11890728
##      Defender_JYP    Defender_KalPa   Defender_Kärpät   Defender_KooKoo
##        0.02167827       -0.09859855       -0.09560925        0.28236836
##    Defender_Lukko Defender_Pelicans    Defender_SaiPa    Defender_Sport
##       -0.05286361        0.17050400        0.09066661        0.33719941
##  Defender_Tappara              Home
##       -0.14036259        0.16294700
```

```
string = ""
for (i in (1:15)){
  string = cat(string,as.character(teams[i]),'&')
  string = cat(string,glm_fixed$coefficients[i],'&',glm_fixed$coefficients[i+15],'& \\')
}
```

```
##  Ässät & 0.8081199 & 0.1993654 & \ HIFK & 0.7771958 & -0.1744723 & \ HPK & 0.6823558 & 0.06781017 & \
```

After getting the coefficients vector, we add back TPS_Defender = 0 to this vector.

```
coefs = glm_fixed$coefficients
coefs = c(coefs[1:29],0,coefs[30])
coefs = matrix(coefs)
```

b) Predict number of goals scored by teams

Test case:

```
home_team = "HIFK"
away_team = "SaiPa"
```

Function to predict goals scored by each team:

```
predict_goals = function(home_team, away_team){
  home_id = which(teams==home_team)
  away_id = which(teams==away_team)
  new_match = matrix(rep(0,62),ncol=31)
  new_match[1,home_id] = 1
  new_match[1,away_id+15] = 1
  new_match[1,31] = 1
  new_match[2,away_id] = 1
  new_match[2,home_id+15] = 1
  predicted_goals = exp(new_match%*%coefs)
  return(predicted_goals)
}
```

```
predicted_goals = predict_goals(home_team,away_team)
sprintf("Predict number of goals scored by Home team: %f", predicted_goals[1])
```

```
## [1] "Predict number of goals scored by Home team: 2.803334"
```

```
sprintf("Predict number of goals scored by Visiting team: %f", predicted_goals[2])
```

```
## [1] "Predict number of goals scored by Visiting team: 1.672664"
```

c)Estimate winning probabilities of individual games

```
home_team = "HIFK"
away_team = "Ilves"
```

Use Monte Carlo simulation to attain 1000000 simulations of goals scored by each team, then compare them element-wise. The playoff variable in the function tells if the result accepts draw (playoff=True - no draws)

```
match_result = function(home_team, away_team, n_simu = 1000000, playoff=T){
  p_goals = predict_goals(home_team,away_team)
  hgoals = rpois(n_simu, p_goals[1])
  vgoals = rpois(n_simu, p_goals[2])
  hwin = sum(hgoals>vgoals)/n_simu
  vwin = sum(hgoals<vgoals)/n_simu
  if (playoff){
    wplayoff = hwin+vwin
    hwin = hwin/wplayoff
```

```r
    vwin = vwin/wplayoff
    return(c(hwin,vwin,0))
  }
  draw = sum(hgoals==vgoals)/n_simu
  return(c(hwin,vwin,draw))
}
```

```r
p_result = match_result(home_team, away_team, playoff = F)
sprintf("Home team has %d%% chance to win", round(p_result[1]*100))
```

```
## [1] "Home team has 66% chance to win"
```

```r
sprintf("Away team has %d%% chance to win", round(p_result[2]*100))
```

```
## [1] "Away team has 19% chance to win"
```

```r
sprintf("Chance that the match draws is %d%%", round(p_result[3]*100))
```

```
## [1] "Chance that the match draws is 15%"
```

d)Likelihood of different outcomes for the entire playoff bracket

Generate winner for one set of playoff matches (best of 7). The teams take turns to be the host, and the result of the games are simulated accordingly. If one game is drew, then it is simulated again until there is a winner.

```r
best_of_seven = function(high_team,low_team){
  pgoals = cbind(predict_goals(high_team,low_team),rev(predict_goals(low_team, high_team)))
  high_wins = 0
  low_wins = 0
  i = 1
  while (high_wins<4 & low_wins<4) {
    g1 = 0
    g2 = 0
    while (g1 == g2){
      g1 = rpois(1,pgoals[1,i])
      g2 = rpois(1,pgoals[2,i])
    }
    if (g1>g2){
      high_wins = high_wins + 1
    } else {
      low_wins = low_wins + 1
    }
    i = i + 1
    if (i == 3) {i=1}
  }
  if (high_wins >=4) {
    return(high_team)
  } else {
    return(low_team)
  }
}
```

```r
print(best_of_seven(home_team,away_team))
```

```
## [1] "HIFK"
```

Function the_champion generate from a list of 8 teams a champion:

```
the_champion = function(teams){
  q1_winner = best_of_seven(teams[1],teams[2])
  q2_winner = best_of_seven(teams[3],teams[4])
  q3_winner = best_of_seven(teams[5],teams[6])
  q4_winner = best_of_seven(teams[7],teams[8])
  s1_winner = best_of_seven(q1_winner,q4_winner)
  s2_winner = best_of_seven(q2_winner,q3_winner)
  champion = best_of_seven(s1_winner,s2_winner)
  return(champion)
}
```

Simulating 1000000 times to attain chances of each team to be the winner:

```
final_teams = teams[c(8,1,15,12,14,7,6,2)]
scores = rep(0,8)
n_simu = 1000000
for (i in (1:n_simu)){
  new_champion = the_champion(final_teams)
  champ_index = which(final_teams==new_champion)
  temp = scores[champ_index] + 1
  scores[champ_index] = temp
}
chances = scores/n_simu

for (i in (1:8)){
  print(paste("Team ",final_teams[i]," has ",chances[i]*100,"% chance of winning the title."))
}
```

```
## [1] "Team  Kärpät  has  44.641 % chance of winning the title."
## [1] "Team  Ässät  has  0.3296 % chance of winning the title."
## [1] "Team  TPS  has  20.354 % chance of winning the title."
## [1] "Team  SaiPa  has  0.4776 % chance of winning the title."
## [1] "Team  Tappara  has  9.6297 % chance of winning the title."
## [1] "Team  KalPa  has  2.6903 % chance of winning the title."
## [1] "Team  JYP  has  10.4182 % chance of winning the title."
## [1] "Team  HIFK  has  11.4596 % chance of winning the title."
```

```
string = ""
for (i in (1:8)){
  string = cat(string, as.character(final_teams[i]), '&', chances[i]*100, '//')
}
```

```
##  Kärpät & 44.641 // Ässät & 0.3296 // TPS & 20.354 // SaiPa & 0.4776 // Tappara & 9.6297 // KalPa & 2
```

```
x = data.frame(teams = final_teams,chances = chances)
write.csv(x,'chances.csv')
```

e)Solve the allocation of a 1000 euros budget

Using the chances calculated above and the betting odds, getting optimal allocation of budget using lpSolveAPI

```r
betting_odds = read.csv('BettingOdds.csv')
bet_odds = c()
for (team in final_teams){
  odd = betting_odds[which(betting_odds$Team==team),2]
  bet_odds = c(bet_odds,odd)
}
budget = 1000
max_porp = 0.5
nteam = length(final_teams)
A = matrix(0,nrow=nteam,ncol=nteam)
for (i in (1:nteam)){
  A[i,i] = 1
}
A = rbind(rep(1,nteam),A)
b = c(budget,rep(max_porp*budget,nteam))
f = chances*bet_odds
```

```
## Warning in chances * bet_odds: longer object length is not a multiple of
## shorter object length
```

```r
library(lpSolveAPI)
lp = make.lp(nrow(A),ncol(A))
for (c in (1:ncol(A))){
  set.column(lp, c, A[,c])
}
set.constr.type(lp,rep("<=",nteam+1))
set.rhs(lp,b)
set.objfn(lp,f)
lp.control(lp,sense='max')
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
```

```
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```r
solve(lp)
```

```
## [1] 0
```

```r
OptimalSolution <- get.variables(lp)
maxValue = get.objective(lp)
```

```r
cat("The maximum expected value of the bet is ",maxValue,"\n")
```

```
## The maximum expected value of the bet is  4562.311
```

```
for (i in (1:nteam)){
  print(paste("Bet", OptimalSolution[i],"euros on team",final_teams[i]))
}
```

```
## [1] "Bet 500 euros on team Kärpät"
## [1] "Bet 0 euros on team Ässät"
## [1] "Bet 0 euros on team TPS"
## [1] "Bet 0 euros on team SaiPa"
## [1] "Bet 0 euros on team Tappara"
## [1] "Bet 0 euros on team KalPa"
## [1] "Bet 0 euros on team JYP"
## [1] "Bet 500 euros on team HIFK"
```

f) Another allocation

We feel that 50% is still a way too big number, so we limit the maximum proportion on each team to 25%.

```
max_porp = 0.25
A = matrix(0,nrow=nteam,ncol=nteam)
for (i in (1:nteam)){
  A[i,i] = 1
}
A = rbind(rep(1,nteam),A)
b = c(budget,rep(max_porp*budget,nteam))
f = chances*bet_odds
```

```
## Warning in chances * bet_odds: longer object length is not a multiple of
## shorter object length
```

```
library(lpSolveAPI)
lp = make.lp(nrow(A),ncol(A))
for (c in (1:ncol(A))){
  set.column(lp, c, A[,c])
}
set.constr.type(lp,rep("<=",nteam+1))
set.rhs(lp,b)
set.objfn(lp,f)
lp.control(lp,sense='max')
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
```

```
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```r
solve(lp)
```

```
## [1] 0
```

```r
OptimalSolution <- get.variables(lp)
maxValue = get.objective(lp)
```

```r
cat("The maximum expected value of the bet is ",maxValue,"\n")
```

```
## The maximum expected value of the bet is  2682.378
```

```r
for (i in (1:nteam)){
  print(paste("Bet", OptimalSolution[i],"euros on team",final_teams[i]))
}
```

```
## [1] "Bet 250 euros on team Kärpät"
## [1] "Bet 0 euros on team Ässät"
## [1] "Bet 250 euros on team TPS"
## [1] "Bet 0 euros on team SaiPa"
## [1] "Bet 250 euros on team Tappara"
## [1] "Bet 0 euros on team KalPa"
## [1] "Bet 0 euros on team JYP"
## [1] "Bet 250 euros on team HIFK"
```

g)

```r
library(dplyr)
df.new <- read.csv('RegularSeasonData.csv', fileEncoding = 'latin1')
teams = sort(unique(df.new$home))

df.new<- df.new%>% mutate(h = ifelse(ot==1,pmin(h,v),h), v = ifelse(ot==1,pmin(h,v),v))

df.new1 = df.new %>% select(home, visitor, h) %>% rename(Attacker = home, Defender = visitor, Goals = h
df.new2 = df.new %>% select(home, visitor, v) %>% rename(Attacker = visitor, Defender = home, Goals = v
data.new = df.new1 %>% rbind(df.new2)
library(dummies)
data.new = dummy.data.frame(data.new,names = c("Attacker","Defender","Home"), sep="_")
data.new = data.new[-47]
new_glm_model = glm(data=data.new, formula = Goals ~ -1+.-Defender_TPS, family = poisson())
new_glm_model
```

```
##
## Call:  glm(formula = Goals ~ -1 + . - Defender_TPS, family = poisson(),
##     data = data.new)
##
## Coefficients:
##    Attacker_Ässät      Attacker_HIFK       Attacker_HPK
##           0.79888            0.57215            0.87684
##    Attacker_Ilves   Attacker_Jukurit       Attacker_JYP
##           0.79355            0.60848            1.00988
##    Attacker_KalPa    Attacker_Kärpät    Attacker_KooKoo
##           0.46183            0.90007            0.73711
##    Attacker_Lukko  Attacker_Pelicans     Attacker_SaiPa
##           0.58850            0.76786            0.80146
```

```
##    Attacker_Sport   Attacker_Tappara       Attacker_TPS
##           0.77483            0.71126            0.99302
##    Defender_Ässät      Defender_HIFK       Defender_HPK
##           0.19937           -0.17447            0.06781
##    Defender_Ilves   Defender_Jukurit       Defender_JYP
##           0.25927            0.11891            0.02168
##    Defender_KalPa    Defender_Kärpät    Defender_KooKoo
##          -0.09860           -0.09561            0.28237
##    Defender_Lukko  Defender_Pelicans     Defender_SaiPa
##          -0.05286            0.17050            0.09067
##    Defender_Sport   Defender_Tappara         Home_Ässät
##           0.33720           -0.14036            0.17997
##         Home_HIFK           Home_HPK         Home_Ilves
##           0.51427           -0.23293            0.19167
##      Home_Jukurit           Home_JYP         Home_KalPa
##           0.15657            0.05466            0.49141
##       Home_Kärpät        Home_KooKoo         Home_Lukko
##           0.29080            0.04380            0.22653
##     Home_Pelicans         Home_SaiPa         Home_Sport
##           0.22884           -0.05635            0.10970
##      Home_Tappara           Home_TPS
##           0.19933            0.04445
##
## Degrees of Freedom: 900 Total (i.e. Null);  856 Residual
## Null Deviance:       2521
## Residual Deviance: 907.9     AIC: 3300
```

```
coefs2 = new_glm_model$coefficients
coefs2 = c(coefs2[1:29],0,coefs2[30:44])
coefs2 = matrix(coefs2)
```

```
predict_goals = function(home_team, away_team){
  home_id = which(teams==home_team)
  away_id = which(teams==away_team)
  new_match = matrix(rep(0,90),ncol=45)
  new_match[1,home_id] = 1
  new_match[1,away_id+15] = 1
  new_match[1,home_id+30] = 1
  new_match[2,away_id] = 1
  new_match[2,home_id+15] = 1
  predicted_goals = exp(new_match%*%coefs2)
  return(predicted_goals)
}
```

```
final_teams = teams[c(8,1,15,12,14,7,6,2)]
scores = rep(0,8)
n_simu = 1000000
for (i in (1:n_simu)){
  new_champion = the_champion(final_teams)
  champ_index = which(final_teams==new_champion)
  temp = scores[champ_index] + 1
  scores[champ_index] = temp
}
chances = scores/n_simu
for (i in (1:8)){
```

```
  print(paste("Team ",final_teams[i]," has ",chances[i]*100,"% chance of winning the title."))
}
```

```
## [1] "Team  Kärpät  has  45.8598 % chance of winning the title."
## [1] "Team  Ässät  has  0.3493 % chance of winning the title."
## [1] "Team  TPS  has  20.3848 % chance of winning the title."
## [1] "Team  SaiPa  has  0.5478 % chance of winning the title."
## [1] "Team  Tappara  has  10.1996 % chance of winning the title."
## [1] "Team  KalPa  has  1.983 % chance of winning the title."
## [1] "Team  JYP  has  11.271 % chance of winning the title."
## [1] "Team  HIFK  has  9.4047 % chance of winning the title."
```

```
string = ""
for (i in (1:8)){
  string = cat(string, as.character(final_teams[i]), '&', chances[i]*100, '//')
}
```

```
##  Kärpät & 45.8598 // Ässät & 0.3493 // TPS & 20.3848 // SaiPa & 0.5478 // Tappara & 10.1996 // KalPa
```

```
x = data.frame(teams = final_teams,chances = chances)
write.csv(x = x,file = 'chances2.csv')
```

```
max_porp = 0.5
A = matrix(0,nrow=nteam,ncol=nteam)
for (i in (1:nteam)){
  A[i,i] = 1
}
A = rbind(rep(1,nteam),A)
b = c(budget,rep(max_porp*budget,nteam))
f = chances*bet_odds
```

```
## Warning in chances * bet_odds: longer object length is not a multiple of
## shorter object length
```

```
library(lpSolveAPI)
lp = make.lp(nrow(A),ncol(A))
for (c in (1:ncol(A))){
  set.column(lp, c, A[,c])
}
set.constr.type(lp,rep("<=",nteam+1))
set.rhs(lp,b)
set.objfn(lp,f)
lp.control(lp,sense='max')
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
```

```
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```r
solve(lp)
```

```
## [1] 0
```

```r
OptimalSolution <- get.variables(lp)
maxValue = get.objective(lp)
```

```r
cat("The maximum expected value of the bet is ",maxValue,"\n")
```

```
## The maximum expected value of the bet is  4029.223
```

```r
for (i in (1:nteam)){
  print(paste("Bet", OptimalSolution[i],"euros on team",final_teams[i]))
}
```

```
## [1] "Bet 500 euros on team Kärpät"
## [1] "Bet 0 euros on team Ässät"
## [1] "Bet 0 euros on team TPS"
## [1] "Bet 0 euros on team SaiPa"
## [1] "Bet 0 euros on team Tappara"
## [1] "Bet 0 euros on team KalPa"
## [1] "Bet 0 euros on team JYP"
## [1] "Bet 500 euros on team HIFK"
```