

# Red Bull Bragantino

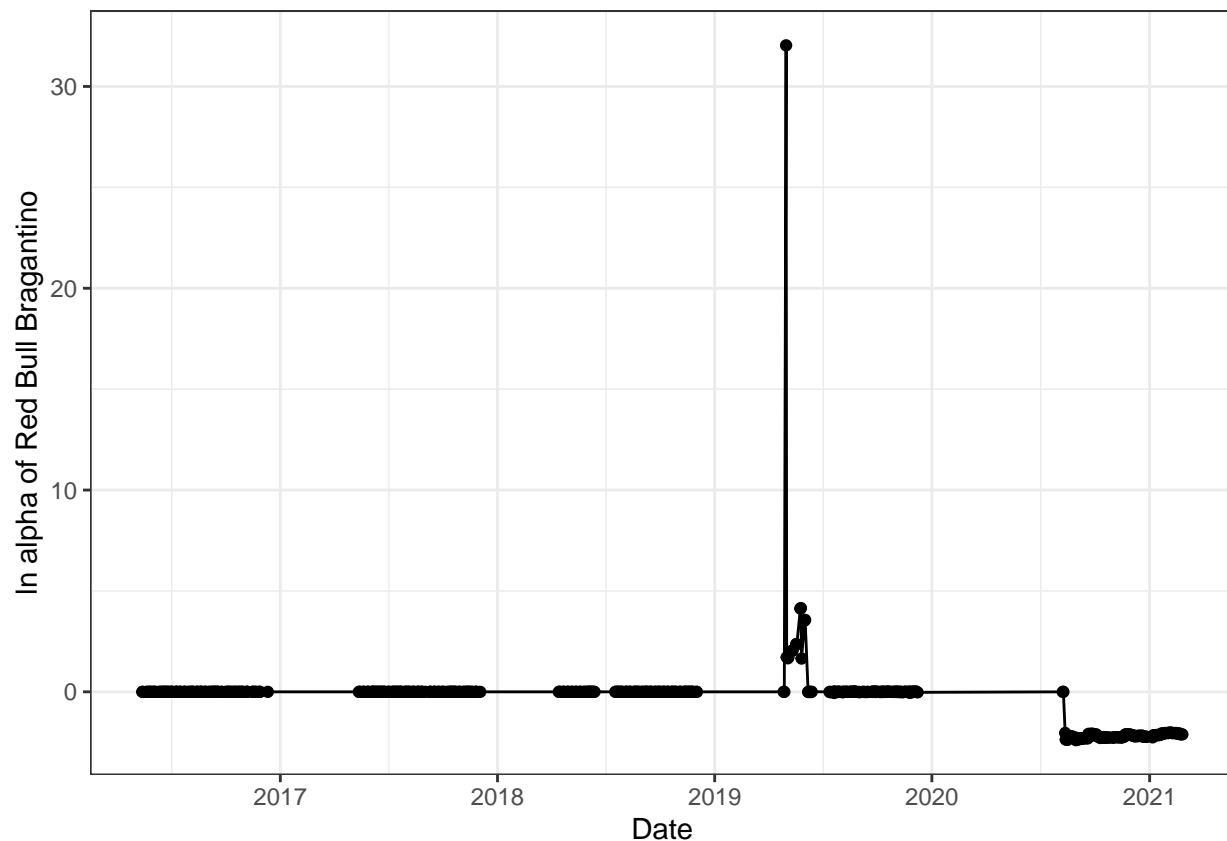
```
library(dplyr)
library(ggplot2)
library(CVXR)
library(future.apply)
library(knitr)
source("~/GitHub/soccer-live-predictions/soccer-live-predictions/pred/pred_mod_8.R")

date = "2019-05-01"
csi = 0.0065/3.5
```

```
load("~/GitHub/soccer-live-predictions/soccer-live-predictions/weight/data/mod_8_dc.RData")
load("~/GitHub/soccer-live-predictions/soccer-live-predictions/scrape/data/results2.RData")

y = NULL
x = as.Date(names(mod_8_dc), format = "%Y-%m-%d")
for(i in 1:length(mod_8_dc)) {
  y[i] = mod_8_dc[[i]]$alpha["Red Bull Bragantino"]
}

tibble(x = x, y = y) %>%
  ggplot(aes(x, y)) +
  geom_line() +
  theme_bw() +
  geom_point() +
  xlab("Date") +
  ylab("ln alpha of Red Bull Bragantino")
```



```
results %>%
  filter(Home_Team == "Red Bull Bragantino" | Away_Team == "Red Bull Bragantino") %>%
  arrange(Date) %>%
  select(Season, Match, Date, Home_Team, Score_Home, Score_Away, Away_Team) %>%
  head(20) %>%
  kable()
```

Season	Match	Date	Home_Team	Score_Home	Score_Away	Away_Team
2020	4	2020-08-09	Santos	1	1	Red Bull Bragantino
2020	14	2020-08-12	Red Bull Bragantino	1	1	Botafogo
2020	27	2020-08-16	Bahia	2	1	Red Bull Bragantino
2020	34	2020-08-19	Red Bull Bragantino	2	1	Fluminense
2020	44	2020-08-23	Red Bull Bragantino	1	2	Coritiba
2020	59	2020-08-29	Fortaleza	3	0	Red Bull Bragantino
2020	68	2020-09-02	Athletico-PR	1	1	Red Bull Bragantino
2020	74	2020-09-06	Red Bull Bragantino	1	2	Palmeiras
2020	83	2020-09-09	São Paulo	1	1	Red Bull Bragantino
2020	95	2020-09-13	Atlético-MG	2	1	Red Bull Bragantino
2020	104	2020-09-19	Red Bull Bragantino	4	2	Ceará
2020	112	2020-09-27	Vasco da Gama	1	1	Red Bull Bragantino
2020	124	2020-10-03	Red Bull Bragantino	0	0	Corinthians
2020	134	2020-10-08	Red Bull Bragantino	0	2	Internacional
2020	150	2020-10-11	Atlético-GO	2	1	Red Bull Bragantino
2020	151	2020-10-15	Flamengo	1	1	Red Bull Bragantino
2020	164	2020-10-18	Red Bull Bragantino	2	0	Sport

Season	Match	Date	Home_Team	Score_Home	Score_Away	Away_Team
2020	174	2020-10-24	Red Bull Bragantino	2	0	Goiás
2020	186	2020-11-02	Grêmio	2	1	Red Bull Bragantino
2020	194	2020-11-08	Red Bull Bragantino	1	1	Santos

```
results %>%
  filter(Season == 2019) %>%
  arrange(Date) %>%
  select(Season, Match, Date, Home_Team, Score_Home, Score_Away, Away_Team) %>%
  head(20) %>%
  kable()
```

Season	Match	Date	Home_Team	Score_Home	Score_Away	Away_Team
2019	2	2019-04-27	Atlético-MG	2	1	Avaí
2019	5	2019-04-27	São Paulo	2	0	Botafogo
2019	6	2019-04-27	Flamengo	3	1	Cruzeiro
2019	8	2019-04-27	Chapecoense	2	0	Internacional
2019	1	2019-04-28	Grêmio	1	2	Santos
2019	3	2019-04-28	Ceará	4	0	Csa
2019	4	2019-04-28	Palmeiras	4	0	Fortaleza
2019	7	2019-04-28	Fluminense	0	1	Goiás
2019	9	2019-04-28	Bahia	3	2	Corinthians
2019	10	2019-04-28	Athletico-PR	4	1	Vasco da Gama
2019	11	2019-05-01	Internacional	2	1	Flamengo
2019	12	2019-05-01	Cruzeiro	1	0	Ceará
2019	13	2019-05-01	Fortaleza	2	1	Athletico-PR
2019	14	2019-05-01	Corinthians	1	0	Chapecoense
2019	16	2019-05-01	Vasco da Gama	1	2	Atlético-MG
2019	18	2019-05-01	Avaí	1	1	Grêmio
2019	19	2019-05-01	Goiás	1	2	São Paulo
2019	20	2019-05-01	Csa	1	1	Palmeiras
2019	15	2019-05-02	Santos	2	1	Fluminense
2019	17	2019-05-02	Botafogo	3	2	Bahia

```
load("~/GitHub/soccer-live-predictions/soccer-live-predictions/2015-2020/data/input.RData")
load("~/GitHub/soccer-live-predictions/soccer-live-predictions/2015-2020/data/input_mod_7.RData")

lines = match_dates %>%
  filter(Date < date)

lines1 = lines$Lines1 %>%
  unlist()

lines2 = lines$Lines2 %>%
  unlist()

lines1s = lines$Lines1s %>%
  unlist()

lines2s = lines$Lines2s %>%
```

```

unlist()

M1_lambda = M1_lambda[lines1,]
M1_mu = M1_mu[lines1,]
M2_lambda = M2_lambda[lines2,]
M2_mu = M2_mu[lines2,]
delta1 = delta1[lines1]
delta2 = delta2[lines2]
H1r = H1r[lines1]
H2r = H2r[lines2]
A1r = A1r[lines1]
A2r = A2r[lines2]

delta1s = delta1s[lines1s]
delta2s = delta2s[lines2s]
int_reds_1 = int_reds_1[lines1s]
int_reds_2 = int_reds_2[lines2s]
H1s = H1s[lines1s]
H2s = H2s[lines2s]
A1s = A1s[lines1s]
A2s = A2s[lines2s]

ind = lines$Ind

U1 = U1[ind]
U2 = U2[ind]
r1 = r1[ind]
r2 = r2[ind]
c = c[ind]

dates_1 = dates_1[which(dates_1 < date)]
dates_2 = dates_2[which(dates_2 < date)]
dates_1s = dates_1s[which(dates_1s < date)]
dates_2s = dates_2s[which(dates_2s < date)]

dif_1 = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(dates_1, "%Y-%m-%d"), units = "days"))
dif_2 = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(dates_2, "%Y-%m-%d"), units = "days"))
dif_1s = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(dates_1s, "%Y-%m-%d"), units = "days"))
dif_2s = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(dates_2s, "%Y-%m-%d"), units = "days"))
dif_st = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(lines$Date, "%Y-%m-%d"), units = "days"))

w_1 = exp(- csi * dif_1)
w_2 = exp(- csi * dif_2)
w_1s = exp(- csi * dif_1s)
w_2s = exp(- csi * dif_2s)
w_st = exp(- csi * dif_st)

alpha = Variable(n)
beta = Variable(n)
gamma = Variable(1)
tau = Variable(1)
omega_xy = Variable(2)
omega = Variable(1)

```

```

theta = vstack(alpha, beta, gamma, tau, omega_xy[1], omega_xy[2], omega_xy[2], omega_xy[1], omega, omega)

eta = Variable(2)
rho = Variable(2)
kappa = Variable(1)
pi1 = eta[1] + rho[1] * r1
pi2 = eta[2] + rho[2] * r2 + c * kappa

a = Variable(2)

loglambda1 = log(delta1) + M1_lambda %*% theta
logmu1 = log(delta1) + M1_mu %*% theta
loglambda2 = log(delta2) + M2_lambda %*% theta
logmu2 = log(delta2) + M2_mu %*% theta

loglambda1s = log(int_reds_1) + a[1]
logmu1s = log(int_reds_1) + a[2]
loglambda2s = log(int_reds_2) + a[1]
logmu2s = log(int_reds_2) + a[2]

log_lik_goals = sum_entries((
  - exp(loglambda1) - exp(logmu1) +
  H1r*loglambda1 + A1r*logmu1) * w_1) +
  sum_entries((
    - exp(loglambda2) - exp(logmu2) +
    H2r*loglambda2 + A2r*logmu2) * w_2)

log_lik_reds = sum_entries((
  - exp(loglambda1s) - exp(logmu1s) +
  H1s*loglambda1s + A1s*logmu1s) * w_1s) +
  sum_entries((
    - exp(loglambda2s) - exp(logmu2s) +
    H2s*loglambda2s + A2s*logmu2s) * w_2s)

log_lik_st = sum_entries((U1 * log(pi1) + U2 * log(pi2) - pi1 - pi2) * w_st)

log_lik = log_lik_goals + log_lik_reds + log_lik_st

objective = Maximize(log_lik)
constraints = list(sum(alpha) - sum(beta) == 0)
problem = Problem(objective, constraints)
solution = solve(problem, solver = "MOSEK")

mod_8_cons = list(alpha = as.vector(c(solution$getValue(alpha))),
  beta = as.vector(solution$getValue(beta)),
  gamma = as.vector(solution$getValue(gamma)),
  tau = as.vector(solution$getValue(tau)),
  omega_xy = as.vector(solution$getValue(omega_xy)),
  omega = as.vector(solution$getValue(omega)),
  a = as.vector(solution$getValue(a)),
  eta = as.vector(solution$getValue(eta)),
  rho = as.vector(solution$getValue(rho)),
  kappa = as.vector(solution$getValue(kappa)),

```

```

loglik = solution$value)
names(mod_8_cons$alpha) = times$Time
names(mod_8_cons$beta) = times$Time
names(mod_8_cons$omega_xy) = c("ahead", "behind")
names(mod_8_cons$omega) = "red"
names(mod_8_cons$a) = c("lambda", "mu")

```

```
mod_8_cons
```

```

## $alpha
##      América-MG      Athletico-PR      Atlético-GO      Atlético-MG
##      -2.838196      -2.182798      -2.517537      -2.103145
##      Avaí          Bahia          Botafogo          Ceará
##      -2.678969      -2.352878      -2.447604      -2.661163
##      Chapecoense    Corinthians    Coritiba          Cruzeiro
##      -2.470348      -2.377367      -2.465443      -2.433229
##      Csa            Figueirense    Flamengo          Fluminense
##      -17.410185     -2.696496     -2.099090      -2.524967
##      Fortaleza      Goiás          Grêmio           Internacional
##      -17.437101     -2.449300     -2.270282      -2.320728
##      Joinville      Palmeiras      Paraná           Ponte Preta
##      -2.866756      -1.961524     -3.263105      -2.447408
## Red Bull Bragantino Santa Cruz      Santos           São Paulo
##      32.037125      -2.349861     -2.245080      -2.279207
##      Sport          Vasco da Gama    Vitória
##      -2.438185      -2.472952     -2.414405
##
## $beta
##      América-MG      Athletico-PR      Atlético-GO      Atlético-MG
##      -2.338389      -2.608690      -2.239281      -2.417258
##      Avaí          Bahia          Botafogo          Ceará
##      -2.329469      -2.457922      -2.489460      -2.650420
##      Chapecoense    Corinthians    Coritiba          Cruzeiro
##      -2.374987      -2.665832      -2.403680      -2.584921
##      Csa            Figueirense    Flamengo          Fluminense
##      -1.149360      -2.342272     -2.733107      -2.377072
##      Fortaleza      Goiás          Grêmio           Internacional
##      -1.660983      -2.616874     -2.741199      -2.736669
##      Joinville      Palmeiras      Paraná           Ponte Preta
##      -2.350373      -2.751432     -2.154857      -2.341986
## Red Bull Bragantino Santa Cruz      Santos           São Paulo
##      0.000000      -1.927020     -2.654380      -2.608088
##      Sport          Vasco da Gama    Vitória
##      -2.225726      -2.366431     -2.140048
##
## $gamma
## [1] 0.5165079
##
## $tau
## [1] 0.254763
##
## $omega_xy
##      ahead      behind

```

```
## -0.2369025  0.1470653
##
## $omega
##      red
## 0.2563516
##
## $a
##      lambda      mu
## -10.89324 -10.39585
##
## $eta
## [1] 2.265243 3.510628
##
## $rho
## [1] 0.8934947 0.2740488
##
## $kappa
## [1] 0.9272729
##
## $loglik
## [1] -2768.927
```

```
load("~/GitHub/soccer-live-predictions/soccer-live-predictions/2015-2020/data/input.RData")
load("~/GitHub/soccer-live-predictions/soccer-live-predictions/2015-2020/data/input_mod_7.RData")
```

```
lines = match_dates %>%
  filter(Date < date)
```

```
lines1 = lines$Lines1 %>%
  unlist()
```

```
lines2 = lines$Lines2 %>%
  unlist()
```

```
lines1s = lines$Lines1s %>%
  unlist()
```

```
lines2s = lines$Lines2s %>%
  unlist()
```

```
M1_lambda = M1_lambda[lines1,]
M1_mu = M1_mu[lines1,]
M2_lambda = M2_lambda[lines2,]
M2_mu = M2_mu[lines2,]
delta1 = delta1[lines1]
delta2 = delta2[lines2]
H1r = H1r[lines1]
H2r = H2r[lines2]
A1r = A1r[lines1]
A2r = A2r[lines2]
```

```
delta1s = delta1s[lines1s]
delta2s = delta2s[lines2s]
int_reds_1 = int_reds_1[lines1s]
```

```

int_reds_2 = int_reds_2[lines2s]
H1s = H1s[lines1s]
H2s = H2s[lines2s]
A1s = A1s[lines1s]
A2s = A2s[lines2s]

ind = lines$Ind

U1 = U1[ind]
U2 = U2[ind]
r1 = r1[ind]
r2 = r2[ind]
c = c[ind]

dates_1 = dates_1[which(dates_1 < date)]
dates_2 = dates_2[which(dates_2 < date)]
dates_1s = dates_1s[which(dates_1s < date)]
dates_2s = dates_2s[which(dates_2s < date)]

dif_1 = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(dates_1, "%Y-%m-%d"), units = "days"))
dif_2 = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(dates_2, "%Y-%m-%d"), units = "days"))
dif_1s = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(dates_1s, "%Y-%m-%d"), units = "days"))
dif_2s = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(dates_2s, "%Y-%m-%d"), units = "days"))
dif_st = as.integer(difftime(as.Date(date, "%Y-%m-%d"), as.Date(lines$Date, "%Y-%m-%d"), units = "days"))

w_1 = exp(- csi * dif_1)
w_2 = exp(- csi * dif_2)
w_1s = exp(- csi * dif_1s)
w_2s = exp(- csi * dif_2s)
w_st = exp(- csi * dif_st)

alpha = Variable(n)
beta = Variable(n)
gamma = Variable(1)
tau = Variable(1)
omega_xy = Variable(2)
omega = Variable(1)
theta = vstack(alpha, beta, gamma, tau, omega_xy[1], omega_xy[2], omega_xy[2], omega_xy[1], omega, omega)

eta = Variable(2)
rho = Variable(2)
kappa = Variable(1)
pi1 = eta[1] + rho[1] * r1
pi2 = eta[2] + rho[2] * r2 + c * kappa

a = Variable(2)

loglambda1 = log(delta1) + M1_lambda %>% theta
logmu1 = log(delta1) + M1_mu %>% theta
loglambda2 = log(delta2) + M2_lambda %>% theta
logmu2 = log(delta2) + M2_mu %>% theta

loglambda1s = log(int_reds_1) + a[1]

```



```

logmu1s = log(int_reds_1) + a[2]
loglambda2s = log(int_reds_2) + a[1]
logmu2s = log(int_reds_2) + a[2]

log_lik_goals = sum_entries((
  - exp(loglambda1) - exp(logmu1) +
  H1r*loglambda1 + A1r*logmu1) * w_1) +
sum_entries((
  - exp(loglambda2) - exp(logmu2) +
  + H2r*loglambda2 + A2r*logmu2) * w_2)

log_lik_reds = sum_entries((
  - exp(loglambda1s) - exp(logmu1s) +
  H1s*loglambda1s + A1s*logmu1s) * w_1s) +
sum_entries((
  - exp(loglambda2s) - exp(logmu2s) +
  + H2s*loglambda2s + A2s*logmu2s) * w_2s)

log_lik_st = sum_entries((U1 * log(pi1) + U2 * log(pi2) - pi1 - pi2) * w_st)

log_lik = log_lik_goals + log_lik_reds + log_lik_st

objective = Maximize(log_lik)
problem = Problem(objective)
solution = solve(problem, solver = "MOSEK")

mod_8_no_cons = list(alpha = as.vector(c(solution$getValue(alpha))),
  beta = as.vector(solution$getValue(beta)),
  gamma = as.vector(solution$getValue(gamma)),
  tau = as.vector(solution$getValue(tau)),
  omega_xy = as.vector(solution$getValue(omega_xy)),
  omega = as.vector(solution$getValue(omega)),
  a = as.vector(solution$getValue(a)),
  eta = as.vector(solution$getValue(eta)),
  rho = as.vector(solution$getValue(rho)),
  kappa = as.vector(solution$getValue(kappa)),
  loglik = solution$value)
names(mod_8_no_cons$alpha) = times$Time
names(mod_8_no_cons$beta) = times$Time
names(mod_8_no_cons$omega_xy) = c("ahead", "behind")
names(mod_8_no_cons$omega) = "red"
names(mod_8_no_cons$a) = c("lambda", "mu")

mod_8_no_cons

```

```

## $alpha
##      América-MG      Athletico-PR      Atlético-GO      Atlético-MG
##      -2.838196      -2.182798      -2.517537      -2.103145
##      Avaí          Bahia          Botafogo          Ceará
##      -2.678969      -2.352878      -2.447604      -2.661163
##      Chapecoense    Corinthians    Coritiba          Cruzeiro
##      -2.470348      -2.377367      -2.465443      -2.433229
##      Csa           Figueirense    Flamengo          Fluminense

```

##	-17.410185	-2.696496	-2.099090	-2.524967
##	Fortaleza	Goiás	Grêmio	Internacional
##	-17.437101	-2.449300	-2.270282	-2.320728
##	Joinville	Palmeiras	Paraná	Ponte Preta
##	-2.866756	-1.961524	-3.263105	-2.447408
##	Red Bull Bragantino	Santa Cruz	Santos	São Paulo
##	0.000000	-2.349861	-2.245080	-2.279207
##	Sport	Vasco da Gama	Vitória	
##	-2.438185	-2.472952	-2.414405	
##				
##	\$beta			
##	América-MG	Athletico-PR	Atlético-GO	Atlético-MG
##	-2.338389	-2.608690	-2.239281	-2.417258
##	Avaí	Bahia	Botafogo	Ceará
##	-2.329469	-2.457922	-2.489460	-2.650420
##	Chapecoense	Corinthians	Coritiba	Cruzeiro
##	-2.374987	-2.665832	-2.403680	-2.584921
##	Csa	Figueirense	Flamengo	Fluminense
##	-1.149360	-2.342272	-2.733107	-2.377072
##	Fortaleza	Goiás	Grêmio	Internacional
##	-1.660983	-2.616874	-2.741199	-2.736669
##	Joinville	Palmeiras	Paraná	Ponte Preta
##	-2.350373	-2.751432	-2.154857	-2.341986
##	Red Bull Bragantino	Santa Cruz	Santos	São Paulo
##	0.000000	-1.927020	-2.654380	-2.608088
##	Sport	Vasco da Gama	Vitória	
##	-2.225726	-2.366431	-2.140048	
##				
##	\$gamma			
##	[1] 0.5165079			
##				
##	\$tau			
##	[1] 0.254763			
##				
##	\$omega_xy			
##	ahead behind			
##	-0.2369025 0.1470653			
##				
##	\$omega			
##	red			
##	0.2563516			
##				
##	\$a			
##	lambda mu			
##	-10.89324 -10.39585			
##				
##	\$eta			
##	[1] 2.265243 3.510628			
##				
##	\$rho			
##	[1] 0.8934947 0.2740488			
##				
##	\$kappa			
##	[1] 0.9272729			

```
##
## $loglik
## [1] -2768.927
```

```
pred_mod_8(mod_8_cons, home_team = "Santos", away_team = "Fluminense", n = 10^6)
```

```
## $Result
##      Santos      Draw Fluminense
## 0.648044 0.243234 0.108722
##
## $Score
##
##      1-0      2-0      1-1      2-1      0-0      3-0      3-1      0-1
## 0.183577 0.138382 0.115608 0.109187 0.086956 0.068206 0.052887 0.050085
##      2-2      1-2      4-0      3-2      4-1      0-2      4-2      5-0
## 0.035680 0.031146 0.025061 0.021270 0.019123 0.012136 0.007537 0.007397
##      2-3      5-1      3-3      1-3      5-2      4-3      0-3      6-0
## 0.006412 0.005679 0.004649 0.004640 0.002170 0.001984 0.001946 0.001816
##      6-1      2-4      3-4      1-4      5-3      6-2      7-0      4-4
## 0.001368 0.000725 0.000594 0.000549 0.000531 0.000517 0.000430 0.000329
##      7-1      0-4      6-3      5-4      7-2      8-0      3-5      2-5
## 0.000303 0.000235 0.000146 0.000124 0.000104 0.000072 0.000063 0.000061
##      8-1      1-5      4-5      7-3      6-4      0-5      8-2      5-5
## 0.000061 0.000055 0.000037 0.000036 0.000028 0.000021 0.000018 0.000011
##      9-1      2-6      1-6      9-0      10-0      4-6      9-2      11-0
## 0.000008 0.000007 0.000006 0.000006 0.000004 0.000003 0.000003 0.000002
##      6-5      8-3      10-1      3-6      6-6      7-4      8-4
## 0.000002 0.000002 0.000001 0.000001 0.000001 0.000001 0.000001
```

```
pred_mod_8(mod_8_no_cons, home_team = "Santos", away_team = "Fluminense", n = 10^6)
```

```
## $Result
##      Santos      Draw Fluminense
## 0.647751 0.243881 0.108368
##
## $Score
##
##      1-0      2-0      1-1      2-1      0-0      3-0      3-1      0-1
## 0.183859 0.137753 0.116197 0.108897 0.087320 0.068503 0.053170 0.049768
##      2-2      1-2      4-0      3-2      4-1      0-2      4-2      5-0
## 0.035392 0.031268 0.024877 0.020992 0.019287 0.011944 0.007710 0.007415
##      2-3      5-1      1-3      3-3      5-2      0-3      4-3      6-0
## 0.006227 0.005545 0.004784 0.004610 0.002145 0.002002 0.001966 0.001815
##      6-1      2-4      5-3      3-4      1-4      6-2      7-0      4-4
## 0.001361 0.000685 0.000603 0.000599 0.000594 0.000523 0.000398 0.000343
##      7-1      0-4      6-3      7-2      5-4      8-0      2-5      8-1
## 0.000316 0.000254 0.000139 0.000107 0.000098 0.000072 0.000067 0.000063
##      3-5      1-5      6-4      4-5      7-3      9-0      0-5      8-2
## 0.000051 0.000046 0.000036 0.000027 0.000027 0.000024 0.000022 0.000022
##      5-5      9-1      1-6      2-6      3-6      7-4      8-3      4-6
## 0.000019 0.000011 0.000008 0.000008 0.000006 0.000006 0.000004 0.000003
##      6-5      0-6      0-7      7-5      6-7      9-2      9-3
## 0.000003 0.000002 0.000002 0.000002 0.000001 0.000001 0.000001
```