prem

Alex Mansourati

07/06/2020

```
library(here)
library(rstan)
library(tidyverse)
library(ggplot2)
library(bayesplot)
library(parallel)
library(tibble)
library(purrr)
library(stringi)
library(ggimage)
```

Load data and split between played (past) and unplayed (future) games

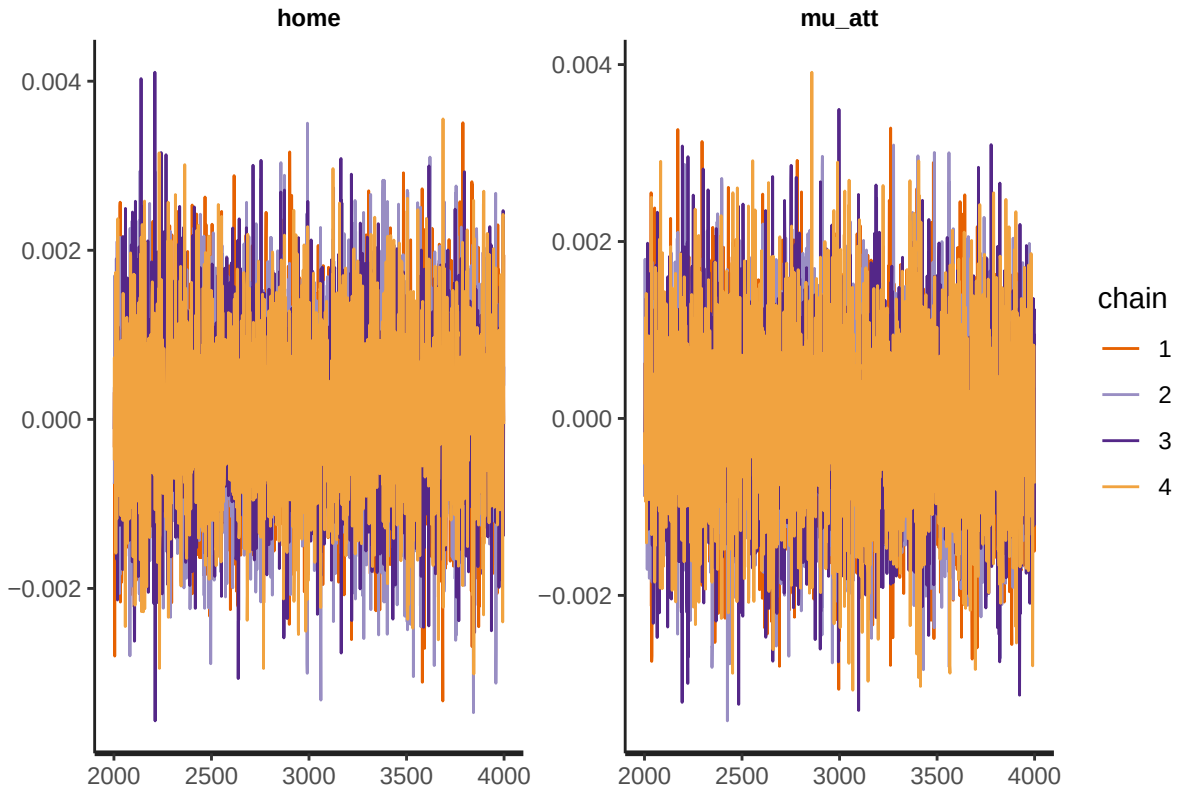
```
df = read.csv(here('premierleague1920.csv'))
df_past = df %>% filter(!is.na(xG))

df_past = df_past %>%
  mutate(h_g = as.integer(as.factor(Home)),
         a_g = as.integer(as.factor(Away))) %>%
  separate(Score, c('y_g1', 'y_g2'), sep='-') %>%
  mutate(g = seq.int(nrow(df_past))) %>%
  mutate(h_g = as.integer(h_g), a_g = as.integer(a_g), y_g1 = as.integer(y_g1), y_g2 = as.integer(y_g2))
```

```
df_future = df %>% filter(is.na(xG))

df_future = df_future %>%
  mutate(h_g = as.integer(as.factor(Home)),
         a_g = as.integer(as.factor(Away))) %>%
  mutate(g = seq.int(nrow(df_future))) %>%
  mutate(h_g = as.integer(h_g), a_g = as.integer(a_g))
```

Fit our model

```
stan_data = list(
  G = nrow(df_past),
  N_TEAMS = max(df_past$h_g),
  h_g = df_past$h_g,
  a_g = df_past$a_g,
  y_g1 = df_past$y_g1,
  y_g2 = df_past$y_g2,
  PRED_G = nrow(df_future),
  future_h_g = df_future$h_g,
  future_a_g = df_future$a_g
)

mod = stan(data = stan_data,
  file = here::here("models/model_w_constraint.stan"),
  iter = 4000,
  seed = 123,
  cores=4)
```

```
traceplot(mod, pars=c('home', 'mu_att'))
```




```
summary = df_past %>%
  group_by(h_g) %>%
  summarise(team = first(Home)) %>%
  mutate(attack_mean = summary(mod)$summary[6:25,1]) %>%
  mutate(defense_mean = summary(mod)$summary[26:45,1])
```

Total Points Today

```
home_points = df_past %>%
  group_by(h_g) %>%
  summarise(team = first(Home),
            home_points = sum(case_when(
    y_g1 > y_g2 ~ 3,
    y_g1 == y_g2 ~ 1,
    y_g1 < y_g2 ~ 0
  )),
  goals_against=sum(y_g2),
  goals_forced=sum(y_g1))

away_points = df_past %>%
  group_by(a_g) %>%
  summarise(team = first(Away),
            away_points = sum(case_when(
    y_g1 < y_g2 ~ 3,
    y_g1 == y_g2 ~ 1,
    y_g1 > y_g2 ~ 0
  )),
  goals_against=sum(y_g1),
  goals_forced=sum(y_g2))
```

```
summary$points = away_points$away_points + home_points$home_points
summary$goals_against = away_points$goals_against + home_points$goals_against
summary$goals_forced = away_points$goals_forced + home_points$goals_forced
summary$goal_differential = summary$goals_forced - summary$goals_against
```

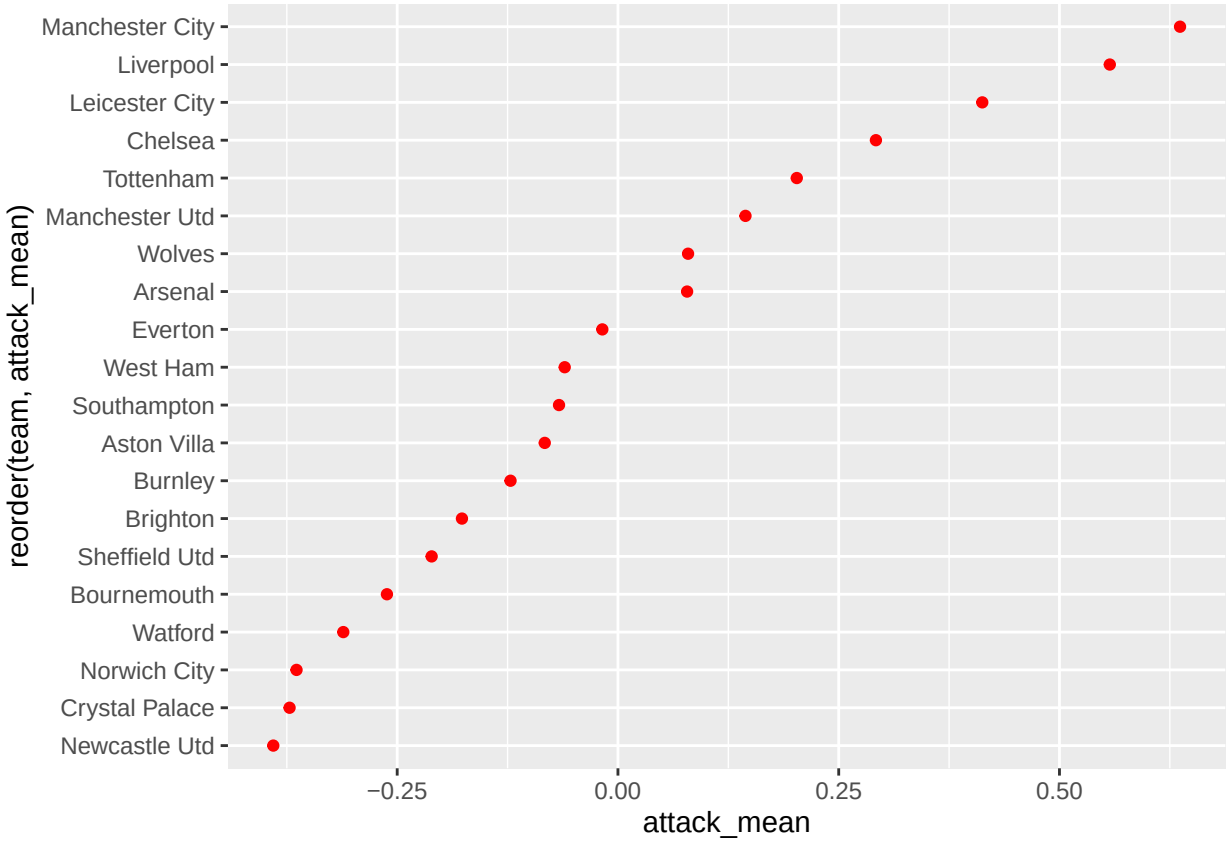
```
summary
```

```
## # A tibble: 20 x 8
##      h_g team  attack_mean defense_mean points goals_against goals_forced
##    <int> <fct>       <dbl>        <dbl>  <dbl>         <int>        <int>
##  1     1 Arse…      0.0783      -0.0108     40            36           40
##  2     2 Asto…     -0.0829       0.337      25            56           34
##  3     3 Bour…     -0.262        0.166      27            47           29
##  4     4 Brig…     -0.177        0.0423     29            40           32
##  5     5 Burn…     -0.122        0.0229     39            40           34
##  6     6 Chel…      0.292        0.0192     48            39           51
##  7     7 Crys…     -0.372       -0.135      39            32           26
##  8     8 Ever…     -0.0177       0.147      37            46           37
##  9     9 Leic…      0.413       -0.242      53            28           58
## 10    10 Live…      0.557       -0.397      82            21           66
## 11    11 Manc…      0.637       -0.129      57            31           68
## 12    12 Manc…      0.144       -0.203      45            30           44
## 13    13 Newc…     -0.390        0.0414     35            41           25
## 14    14 Norw…     -0.364        0.239      21            52           25
## 15    15 Shef…     -0.211       -0.297      43            25           30
```

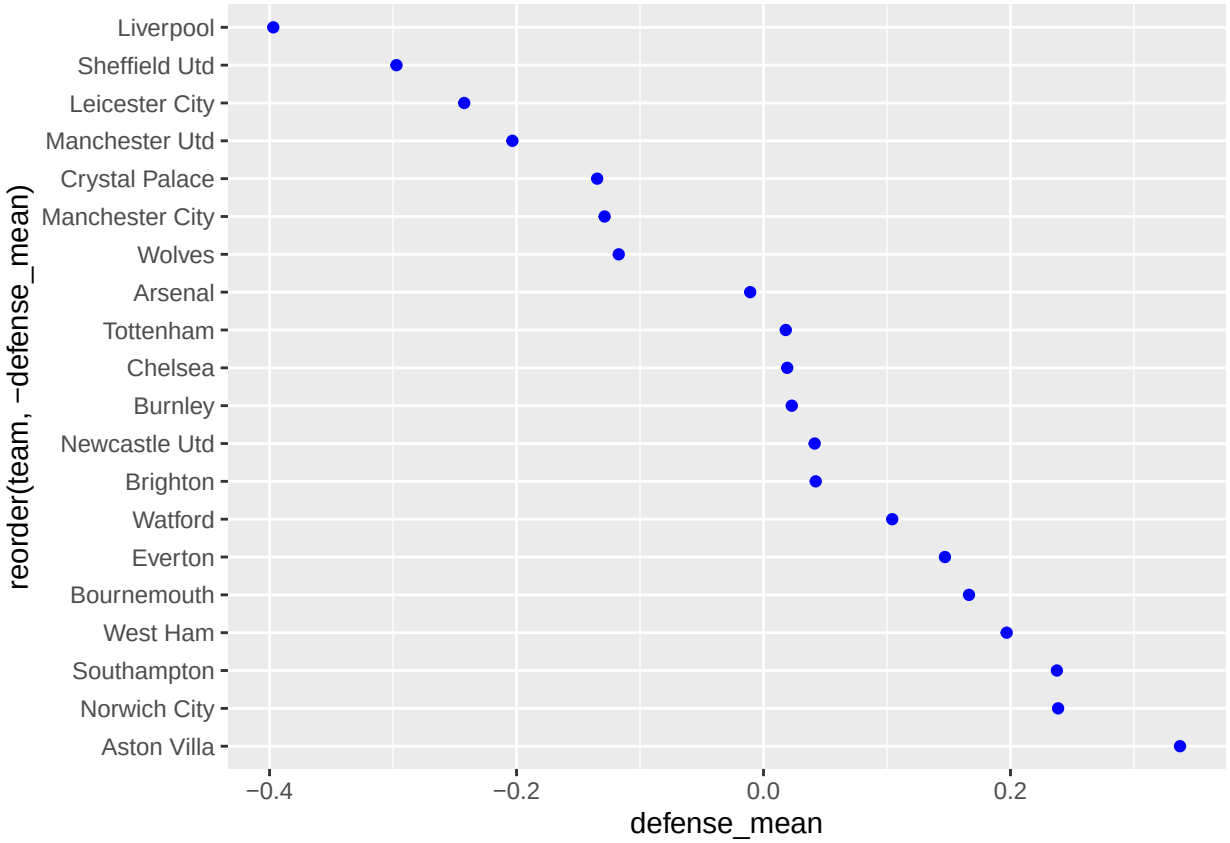
```
## 16    16 Sout…     -0.0667       0.238      34           52           35
## 17    17 Tott…      0.203        0.0180     41           40           47
## 18    18 Watf…     -0.311        0.104      27           44           27
## 19    19 West…     -0.0603       0.197      27           50           35
## 20    20 Wolv…      0.0794      -0.117      43           34           41
## # … with 1 more variable: goal_differential <int>
```

Some summary plots

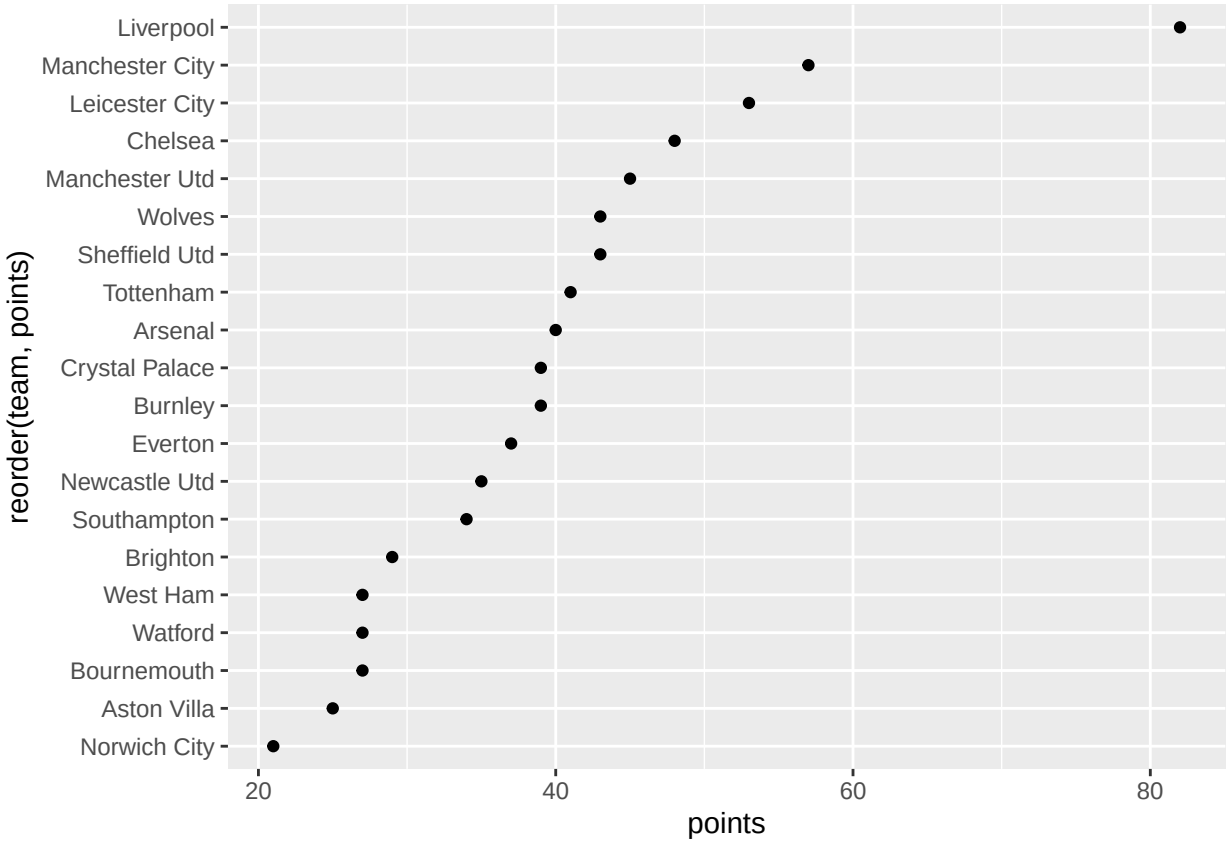
```
summary %>%
  ggplot() + geom_point(aes(x=reorder(team, attack_mean), y=attack_mean), color='red') + coord_flip()
```



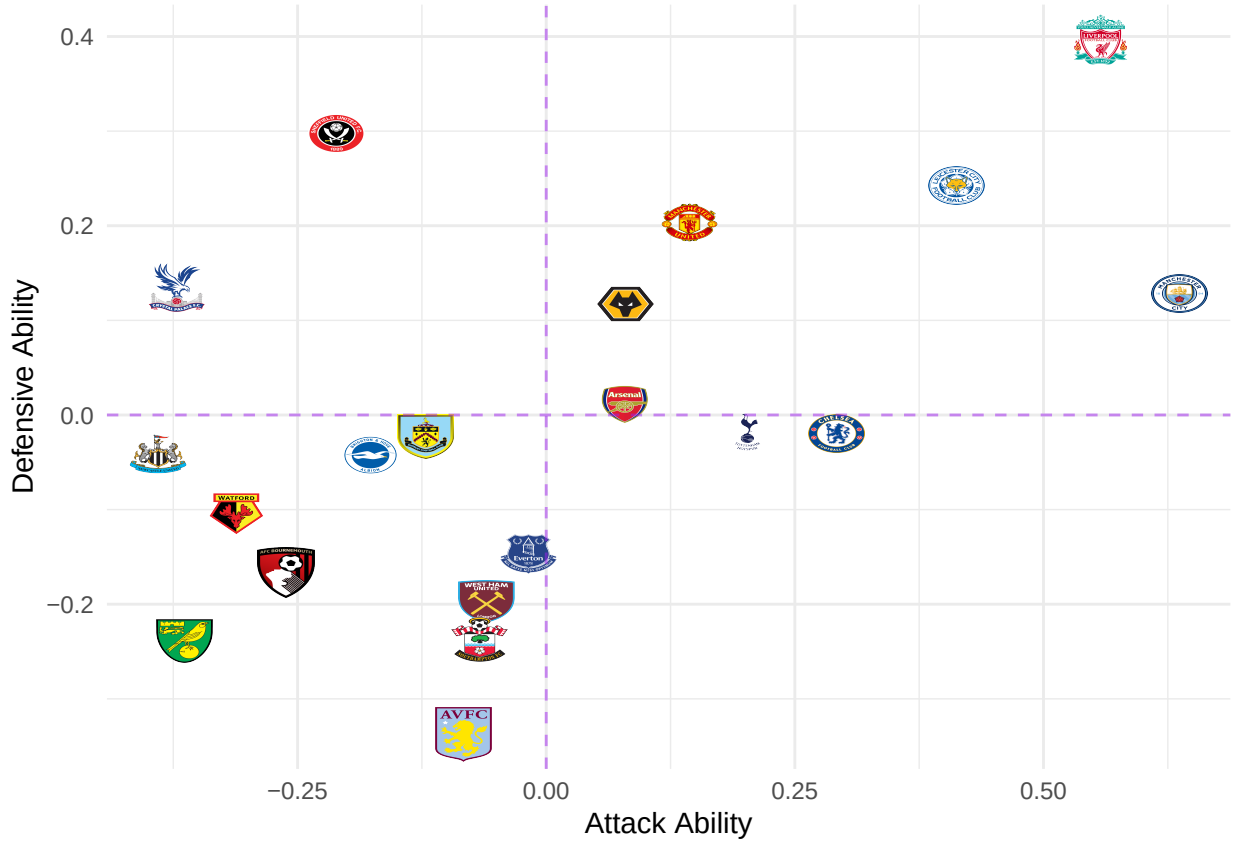

```
summary %>%
  ggplot() + geom_point(aes(x=reorder(team, -defense_mean), y=defense_mean), color='blue') + coord_flip
```




```
summary %>%
  ggplot() + geom_point(aes(x=reorder(team, points), y=points)) + coord_flip()
```




```
logos = read.csv(here('team_logos.csv'))

summary %>%
  left_join(logos, by=c("team"="Team")) %>%
  ggplot(aes(x=attack_mean, y=-defense_mean)) +
  geom_image(aes(image = Logo), size = 0.05) +
  xlab("Attack Ability") +
  ylab("Defensive Ability") +
  hline_at(0, linetype='dashed', col='purple', alpha=0.5) +
  vline_at(0, linetype='dashed', col='purple', alpha=0.5) +
  theme_minimal()
```



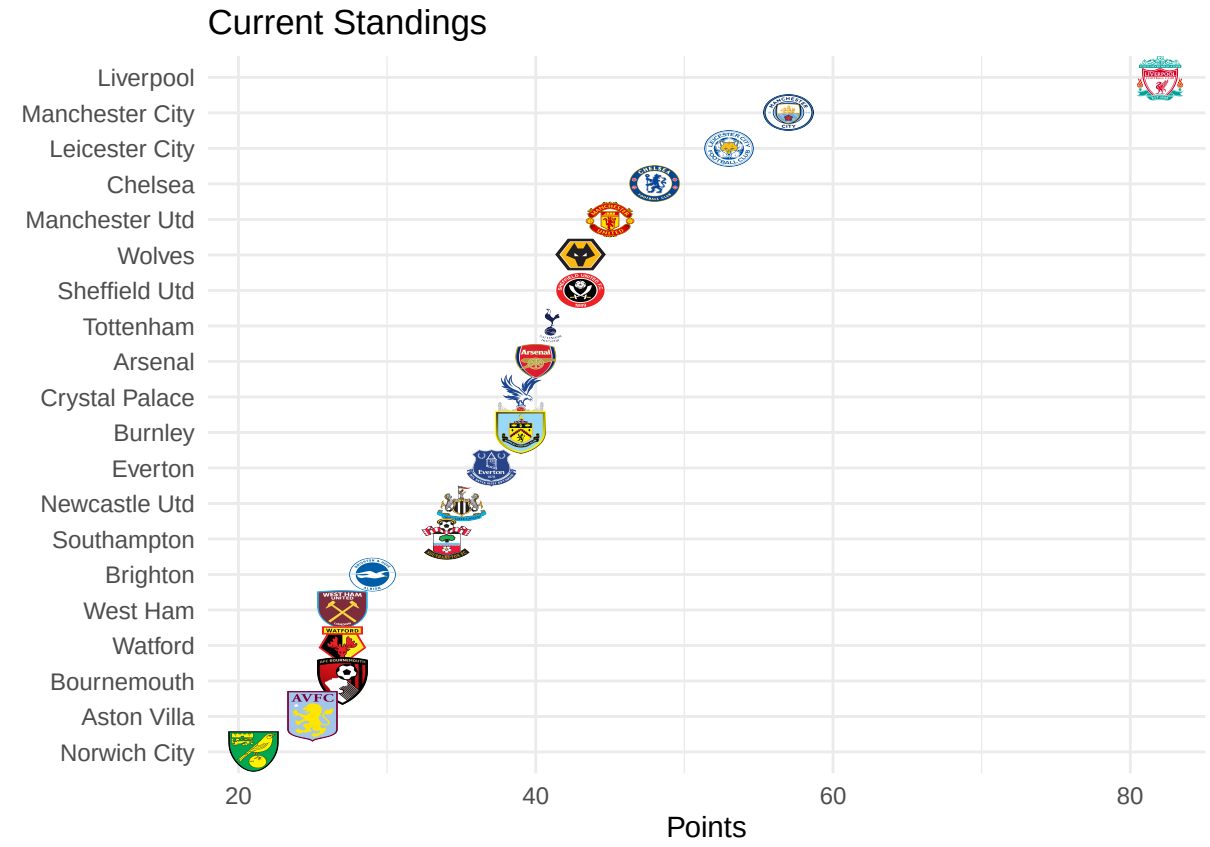
Simulate the rest of season

```
y_g1_pred = rstan::extract(mod)[['y_g1_pred']]
y_g2_pred = rstan::extract(mod)[['y_g2_pred']]
```

```
point_sims = matrix(0, nrow=20, ncol=8000)
goal_diff_sims = matrix(0, nrow=20, ncol=8000)

score_diffs = y_g1_pred - y_g2_pred
for (i in 1:8000){
  for (j in 1:92){
    a = score_diffs[i,j]
    if (a > 0){
     point_sims[df_future[j,'h_g'],i] =  point_sims[df_future[j,'h_g'],i] + 3
    }
    if (a < 0){
     point_sims[df_future[j,'a_g'],i] =  point_sims[df_future[j,'a_g'],i] + 3
    }
    if (a == 0){
     point_sims[df_future[j,'h_g'],i] =  point_sims[df_future[j,'h_g'],i] + 1
     point_sims[df_future[j,'a_g'],i] =  point_sims[df_future[j,'a_g'],i] + 1
    }
    goal_diff_sims[df_future[j,'h_g'],i] = goal_diff_sims[df_future[j,'h_g'],i] + a
```

```
    goal_diff_sims[df_future[j,'a_g'],i] = goal_diff_sims[df_future[j,'a_g'],i] + a
  }
  point_sims[1:20,i] = point_sims[1:20,i] + summary$points
  goal_diff_sims[1:20, i] = goal_diff_sims[1:20, i] + summary$goal_differential
}
```

```
summary %>%
  left_join(logos, by=c("team"="Team")) %>%
  ggplot(aes(x=reorder(team, points), y=points)) +
  geom_image(aes(image = Logo), size = 0.05) +
  ggtitle("Current Standings") +
  xlab("") +
  ylab("Points") +
  coord_flip() +
  theme_minimal()
```



How often does Liverpool finish first?

```
for (i in 1:8000){
  point_sims[1:20, i]
}
```

```
j = 0
for (i in 1:8000){
  if (max(point_sims[1:20,i]) == point_sims[10, i]){
    j = j+1
  }
}
j
```

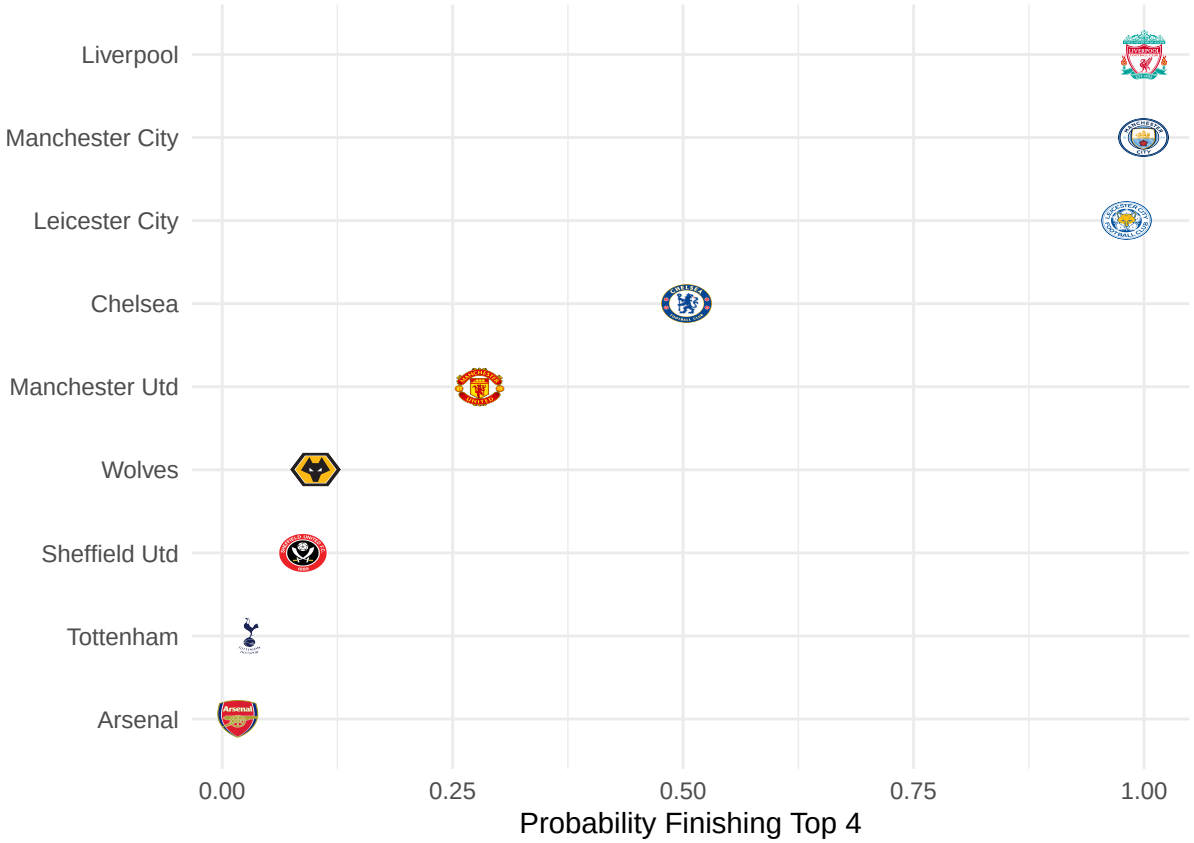
```
## [1] 8000
```

What is probability of each team finishing in top 4?

```
top_4_finishers = matrix(0, nrow=20, ncol=1)
for (i in 1:8000){
  summary$total_points = point_sims[1:20, i]
  summary$goal_differential = goal_diff_sims[1:20, i]
  a = list(array(summary %>%
              arrange(-total_points, -goal_differential) %>%
              select(h_g) %>%
              head(4) %>%
              pull()))
  for (i in a[[1]]){
    top_4_finishers[i] = top_4_finishers[i] + 1
  }
}
```

```
summary$prob_top_4 = top_4_finishers / 8000.
```

```
summary %>%
  filter(prob_top_4 > 0.001) %>%
  left_join(logos, by=c("team"="Team")) %>%
  ggplot(aes(x=reorder(team, prob_top_4), y=prob_top_4)) +
  geom_image(aes(image = Logo), size = 0.05) +
  xlab("") +
  ylab("Probability Finishing Top 4") +
  coord_flip() +
  theme_minimal()
```




```
summary %>% filter(prob_top_4 > 0.001)
```

```
## # A tibble: 9 x 10
##     h_g team  attack_mean defense_mean points goals_against goals_forced
##   <int> <fct>       <dbl>        <dbl>  <dbl>         <int>        <int>
## 1     1 Arse…      0.0783      -0.0108     40            36           40
## 2     6 Chel…      0.292        0.0192     48            39           51
## 3     9 Leic…      0.413       -0.242      53            28           58
## 4    10 Live…      0.557       -0.397      82            21           66
## 5    11 Manc…      0.637       -0.129      57            31           68
## 6    12 Manc…      0.144       -0.203      45            30           44
## 7    15 Shef…     -0.211       -0.297      43            25           30
## 8    17 Tott…      0.203        0.0180     41            40           47
## 9    20 Wolv…      0.0794      -0.117      43            34           41
## # … with 3 more variables: goal_differential <dbl>, total_points <dbl>,
## #   prob_top_4[,1] <dbl>
```

Relegation

```
bottom_3_finishers = matrix(0, nrow=20, ncol=1)

for (i in 1:8000){
```

```
  summary$total_points = point_sims[1:20, i]
  summary$goal_differential = goal_diff_sims[1:20, i]
  a = list(array(summary %>%
              arrange(-total_points, -goal_differential) %>%
              select(h_g) %>%
              tail(3) %>%
              pull()))
  for (i in a[[1]]){
    bottom_3_finishers[i] = bottom_3_finishers[i] + 1
  }
}
```

```
summary$prob_bottom_3 = bottom_3_finishers / 8000.
```

```
summary %>%
  filter(prob_bottom_3 > 0.001) %>%
  left_join(logos, by=c("team"="Team")) %>%
  ggplot(aes(x=reorder(team, prob_bottom_3), y=prob_bottom_3)) +
  geom_image(aes(image = Logo), size = 0.05) +
  xlab("") +
  ylab("Probability Finishing Bottom 3") +
  coord_flip() +
  theme_minimal()
```