Time-Decay Coefficient Model

We only need two packages for this: tidyverse for data manipulation and lme4 for mixed-effects model estimation.

```
library(tidyverse)
library(lme4)

team_results <- read_csv("team_results.csv")
```

Creating an exponential time decay function to apply recency weighting to observations in parameter estimation based on half_life variable — observations from half_life days away from the current date will receive 50% as much weight as present day observations.

```
time_decay_function <- function(game_date, half_life, current_date = Sys.Date()) {

  game_date <- as.Date(game_date)
  current_date <- as.Date(current_date)

  days_elapsed <- as.numeric(difftime(current_date, game_date, units = "days"))

  decay_value <- (1/2) ^ (days_elapsed / half_life)

  return(decay_value)
}
```

The next step is to apply the half_life function to create time weights for the model. We filter to only data since 2000 to match the new (reduced) scope of the project.

The chosen half-life initially is 2 years. (This can be tweaked easily in the below code)

```
model_half_life <- 365 * 2

model_data <- team_results %>%
  mutate(time_weight = time_decay_function(date, model_half_life)) %>%
  filter(year(date) >= 2000)
```

Now we fit the models using lmer:

```
off_model <- lme4::lmer(score ~ (1 | team) + (1 | opponent) + location,
                    data = model_data, weights = time_weight)

def_model <- lme4::lmer(opp_score ~ (1 | team) + (1 | opponent) + location,
                    data = model_data, weights = time_weight)
```

We extract the random effects:

```r
off_eff <- ranef(off_model) %>%
  as.data.frame() %>%
  filter(grpvar == "team") %>%
  select(team = grp, o_effect = condval, o_sd = condsd)

def_eff <- ranef(def_model) %>%
  as.data.frame() %>%
  filter(grpvar == "team") %>%
  select(team = grp, d_effect = condval, d_sd = condsd)

net_eff <- inner_join(off_eff, def_eff, by = "team") %>%
  mutate(net_effect = o_effect - d_effect)
```

Now we have our “team rankings”! (at least a first pass at them)

```r
net_eff_clean <- net_eff %>%
  select(team, o_effect, d_effect, net_effect)
```

Top 10 teams in attack, per the model. The `o_effect` term can be interpreted as “how many more goals than the average team do we expect this team to score against their opponent’s defense?”

```r
net_eff_clean %>%
  arrange(-o_effect) %>%
  head(10)
```

```
##           team o_effect  d_effect net_effect
## 1       France 1.970701 -1.736731   3.707432
## 2     Portugal 1.946859 -1.811944   3.758802
## 3        Spain 1.874129 -1.929302   3.803431
## 4       Brazil 1.856678 -1.943289   3.799967
## 5      Belgium 1.852972 -1.650775   3.503747
## 6      England 1.785587 -1.840038   3.625625
## 7    Argentina 1.750375 -2.014496   3.764871
## 8      Germany 1.729724 -1.307942   3.037666
## 9        Japan 1.712916 -1.214916   2.927832
## 10 Netherlands 1.686269 -1.507787   3.194055
```

Top 10 teams in defense, per the model. The `d_effect` term can be interpreted as “what’s the difference between how many goals this team would allow vs. their opponent and the average team?”

```r
net_eff_clean %>%
  arrange(d_effect) %>%
  head(10)
```

```
##         team o_effect  d_effect net_effect
## 1  Argentina 1.750375 -2.014496   3.764871
## 2     Brazil 1.856678 -1.943289   3.799967
## 3      Spain 1.874129 -1.929302   3.803431
## 4   Colombia 1.216939 -1.927538   3.144477
## 5    England 1.785587 -1.840038   3.625625
## 6    Uruguay 1.355811 -1.836224   3.192034
```

```
## 7   Portugal 1.946859 -1.811944   3.758802
## 8     France 1.970701 -1.736731   3.707432
## 9    Croatia 1.223542 -1.688496   2.912038
## 10     Italy 1.424980 -1.674131   3.099110
```

Top 10 best teams by "net effect," linear combo of offense and defense:

```
net_eff_clean %>%
  arrange(-net_effect) %>%
  head(10)
```

```
##             team o_effect  d_effect net_effect
## 1          Spain 1.874129 -1.929302   3.803431
## 2         Brazil 1.856678 -1.943289   3.799967
## 3      Argentina 1.750375 -2.014496   3.764871
## 4       Portugal 1.946859 -1.811944   3.758802
## 5         France 1.970701 -1.736731   3.707432
## 6        England 1.785587 -1.840038   3.625625
## 7        Belgium 1.852972 -1.650775   3.503747
## 8    Netherlands 1.686269 -1.507787   3.194055
## 9        Uruguay 1.355811 -1.836224   3.192034
## 10      Colombia 1.216939 -1.927538   3.144477
```

```
games_played <- team_results %>%
  group_by(team) %>%
  count() %>%
  rename(games_played = n)

net_eff_clean <- net_eff_clean %>%
  left_join(games_played, by = "team")
```

Match Prediction Visualization

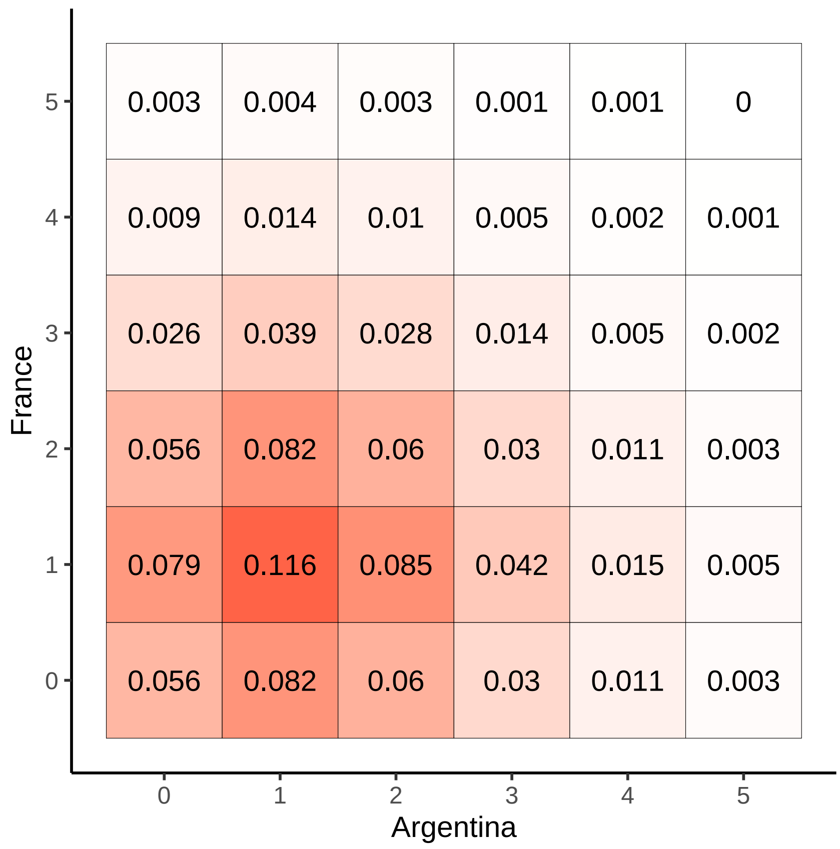
2022 World Cup final:

```
france <- net_eff_clean %>%
  filter(team == "France")

argentina <- net_eff_clean %>%
  filter(team == "Argentina")

avg_goals <- mean(team_results$score)

france_xgoals <- avg_goals + france$o_effect + argentina$d_effect
argentina_xgoals <- avg_goals + argentina$o_effect + france$d_effect
```

```
library(bivpois)

get_score_likelihood <- function(team1, team2, go_to = 5) {

  goals <- expand.grid(seq(0,go_to), seq(0,go_to))
```

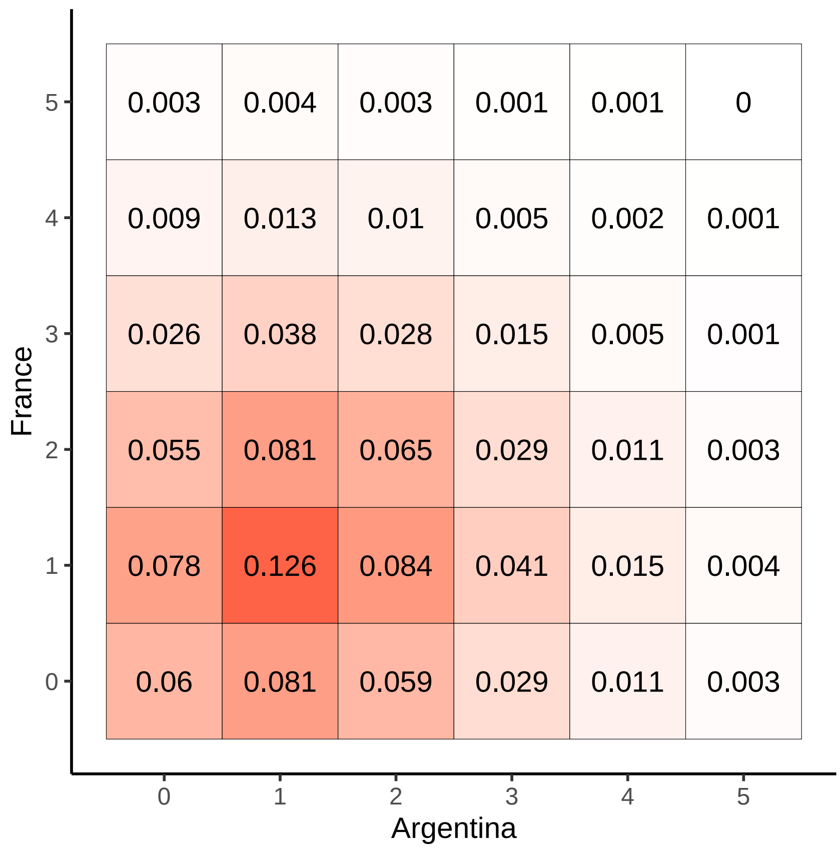
```
  lik <- map2_vec(.x = goals$Var1, .y = goals$Var2, .f = ~ bivpois::dbp(
    x1 = .x, x2 = .y, lambda = c(team1, team2, 0), logged = FALSE
  ))

  goals <- cbind(goals, lik)
  colnames(goals) <- c("ARG", "FRA", "lik")
  return(goals)

}

likelihoods <- get_score_likelihood(argentina_xgoals, france_xgoals)
```

```
ggplot(likelihoods, aes(x = ARG, y = FRA)) +
  geom_tile(color = "black", aes(fill = lik), show.legend = FALSE) +
  scale_fill_gradient(low = "white", high = "tomato") +
  geom_text(aes(label = round(lik, 3))) +
  theme_classic() +
  scale_x_continuous(breaks = seq(0,5)) +
  scale_y_continuous(breaks= seq(0,5)) +
  coord_equal() +
  labs(x = "Argentina", y = "France")
```



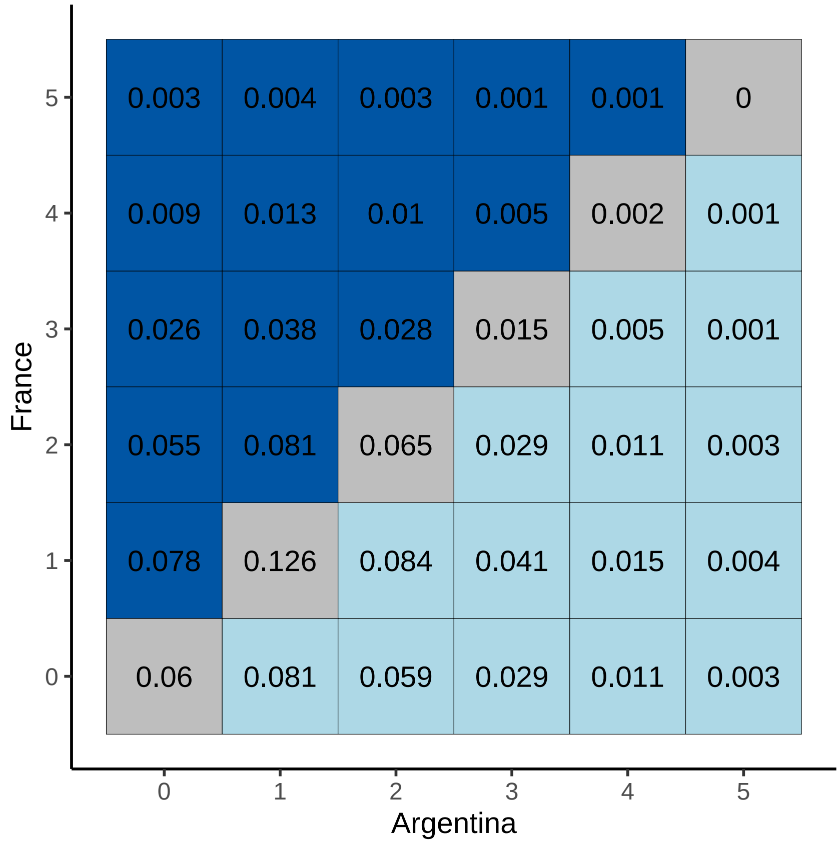

```
likelihoods <- likelihoods %>%
  mutate(result = case_when(
```

```
    ARG > FRA ~ "W",
    ARG == FRA ~ "D",
    ARG < FRA ~ "L",
    TRUE ~ NA
  )) %>%
  mutate(lik = ifelse(result == "D", lik * 1.1, lik))

# rescaling to be valid PMF

scale_factor <- 1 / sum(likelihoods$lik)

likelihoods$lik <- likelihoods$lik * scale_factor

ggplot(likelihoods, aes(x = ARG, y = FRA)) +
  geom_tile(color = "black", aes(fill = lik), show.legend = FALSE) +
  scale_fill_gradient(low = "white", high = "tomato") +
  geom_text(aes(label = round(lik, 3))) +
  theme_classic() +
  scale_x_continuous(breaks = seq(0,5)) +
  scale_y_continuous(breaks= seq(0,5)) +
  coord_equal() +
  labs(x = "Argentina", y = "France")
```




```
ggplot(likelihoods, aes(x = ARG, y = FRA)) +
  geom_tile(color = "black", aes(fill = result), show.legend = FALSE) +
```

```r
  scale_fill_manual(values = c("W" = "lightblue", "D" = "grey", "L" = "#0055A4")) +
  geom_text(aes(label = round(lik, 3))) +
  theme_classic() +
  scale_x_continuous(breaks = seq(0,5)) +
  scale_y_continuous(breaks= seq(0,5)) +
  coord_equal() +
  labs(x = "Argentina", y = "France")
```




```r
match_probs <- likelihoods %>%
  group_by(result) %>%
  summarise(lik_sum = sum(lik)) %>%
  mutate(team = "Argentina") %>%
  mutate(result = factor(result, levels = c("W","D","L"))) %>%
  mutate(team_result = paste(team, result)) %>%
  arrange(result)

library(gt)
library(gtExtras)

gt(match_probs %>% select(team_result, lik_sum)) %>%
  cols_label(team_result = "Result",
             lik_sum = "Odds") %>%
  fmt_percent(lik_sum, decimals = 1) %>%
  gt_theme_538()
```

Result	Odds
Argentina W	37.8%
Argentina D	26.8%
Argentina L	35.3%

Team Ratings

```
net_eff %>%
  arrange(-net_effect) %>%
  select(team, o_effect, d_effect, net_effect) %>%
  head(10) %>%
  gt() %>%
  cols_label(o_effect = "Offense",
             d_effect = "Defense",
             net_effect = "Total",
             team = "Team") %>%
  fmt_number(o_effect:net_effect, decimals = 2, force_sign = TRUE)
```

Team	Offense	Defense	Total
Spain	+1.87	−1.93	+3.80
Brazil	+1.86	−1.94	+3.80
Argentina	+1.75	−2.01	+3.76
Portugal	+1.95	−1.81	+3.76
France	+1.97	−1.74	+3.71
England	+1.79	−1.84	+3.63
Belgium	+1.85	−1.65	+3.50
Netherlands	+1.69	−1.51	+3.19
Uruguay	+1.36	−1.84	+3.19
Colombia	+1.22	−1.93	+3.14