Lecture 9: Statistics in hockey

Skidmore College

Goals

- Shot mapping
- Multiple logistic regression
- Shot efficiency
- Hosmer Lemeshow test
- Score effects

Set-up:

NHL shot data

```
library(RCurl); library(tidyverse)
githubURL <- "https://raw.githubusercontent.com/statsbylopez/StatsSports/master/Data/pbp_da
pbp_data <- readRDS(gzcon(url(githubURL)))
names(pbp_data)
```

```
##  [1] "season"         "game_id"        "game_date"      "session"
##  [5] "event_index"    "game_period"    "game_seconds"   "event_type"
##  [9] "home_team"      "away_team"      "home_skaters"   "away_skaters"
## [13] "home_score"     "away_score"     "event_detail"   "event_team"
## [17] "event_player_1" "event_player_2" "coords_x"       "coords_y"
## [21] "home_goalie"    "away_goalie"    "event_circle"   "event_distance"
## [25] "event_angle"    "shot_prob"
```

Goals as rare events

```
pbp_data %>% count(event_type)
```

```
## # A tibble: 3 x 2
##   event_type      n
##   <chr>       <int>
## 1 GOAL        14902
## 2 MISS        60996
## 3 SHOT       145482
```

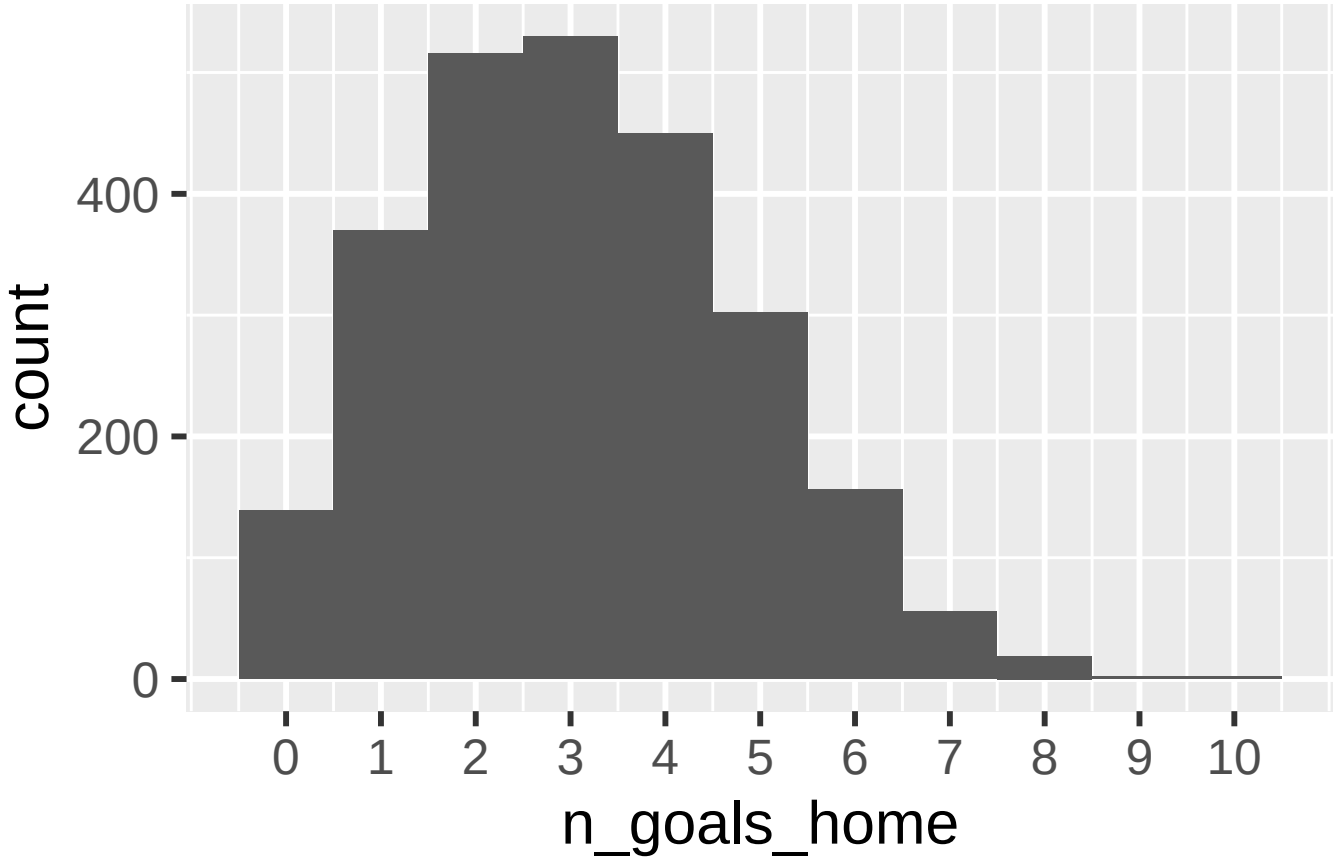
```
goal_counts <- pbp_data %>%
  group_by(game_id) %>%
  summarise(n_goals_home = sum(event_type == "GOAL" & event_team == home_team),
            n_goals_away = sum(event_type == "GOAL" & event_team == away_team))
  ungroup()

goal_counts %>%
  summarise(ave_home_goals = mean(n_goals_home),
            ave_away_goals = mean(n_goals_away))
```

```
## # A tibble: 1 x 2
##   ave_home_goals ave_away_goals
##            <dbl>          <dbl>
## 1           3.08           2.79
```

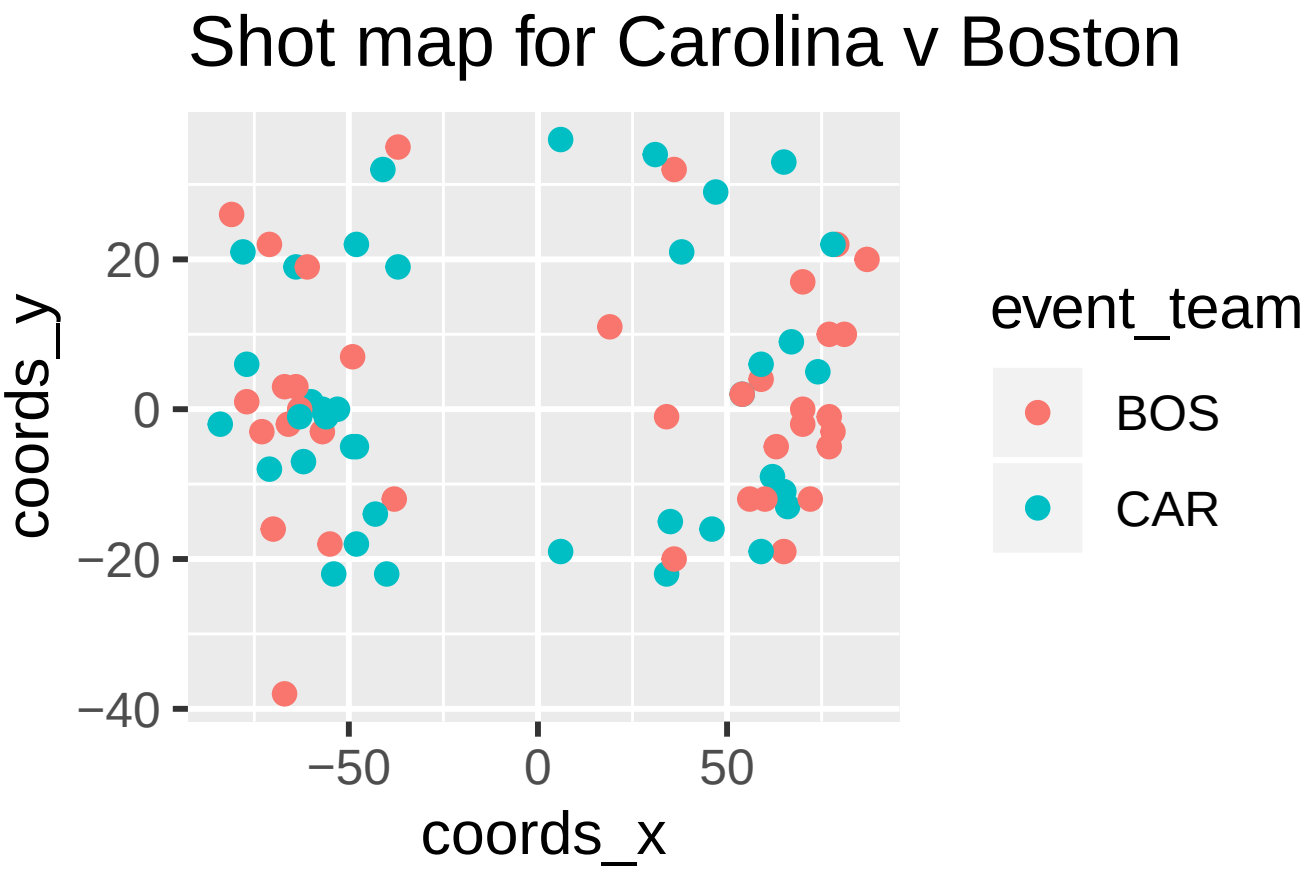
Goals as rare events

```
ggplot(goal_counts, aes(x = n_goals_home)) +
  geom_histogram(binwidth = 1) +
  scale_x_continuous(breaks = 0:10)
```



Shot maps

```
boston_game <- pbp_data %>% filter(game_id == 2017020636)
ggplot(boston_game, aes(x = coords_x, y = coords_y, colour = event_team)) +
  geom_point() +
  labs(title = "Shot map for Carolina v Boston")
```

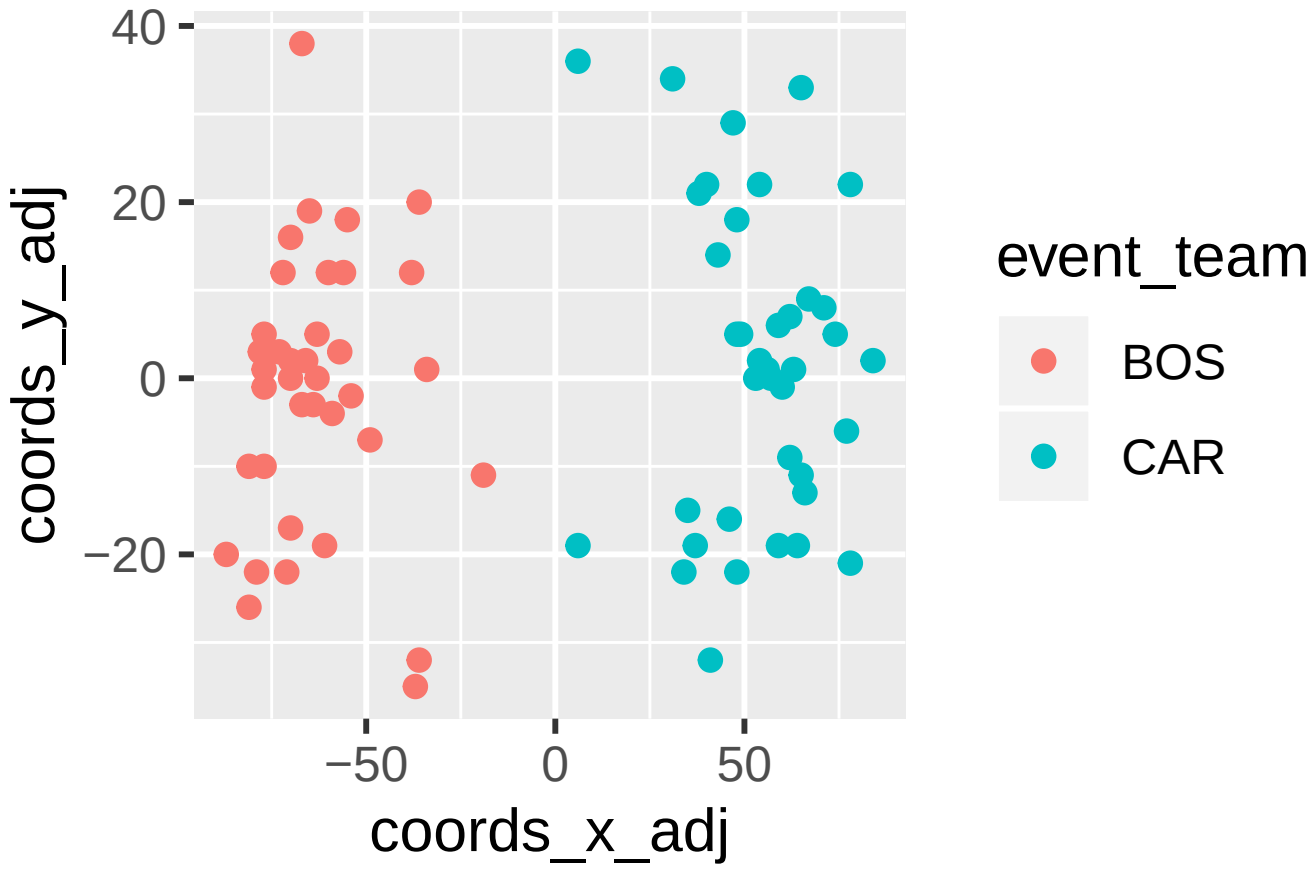


Adjusted coordinates

```
pbp_data <- pbp_data %>%
  mutate(coords_x_adj = ifelse(event_team == home_team,
                               -1*abs(coords_x), abs(coords_x)),
         coords_y_adj = ifelse(event_team == home_team & coords_x < 0,
                               coords_y, -1*coords_y),
         coords_y_adj = ifelse(event_team == away_team & coords_x > 0,
                               coords_y, -1*coords_y))
```

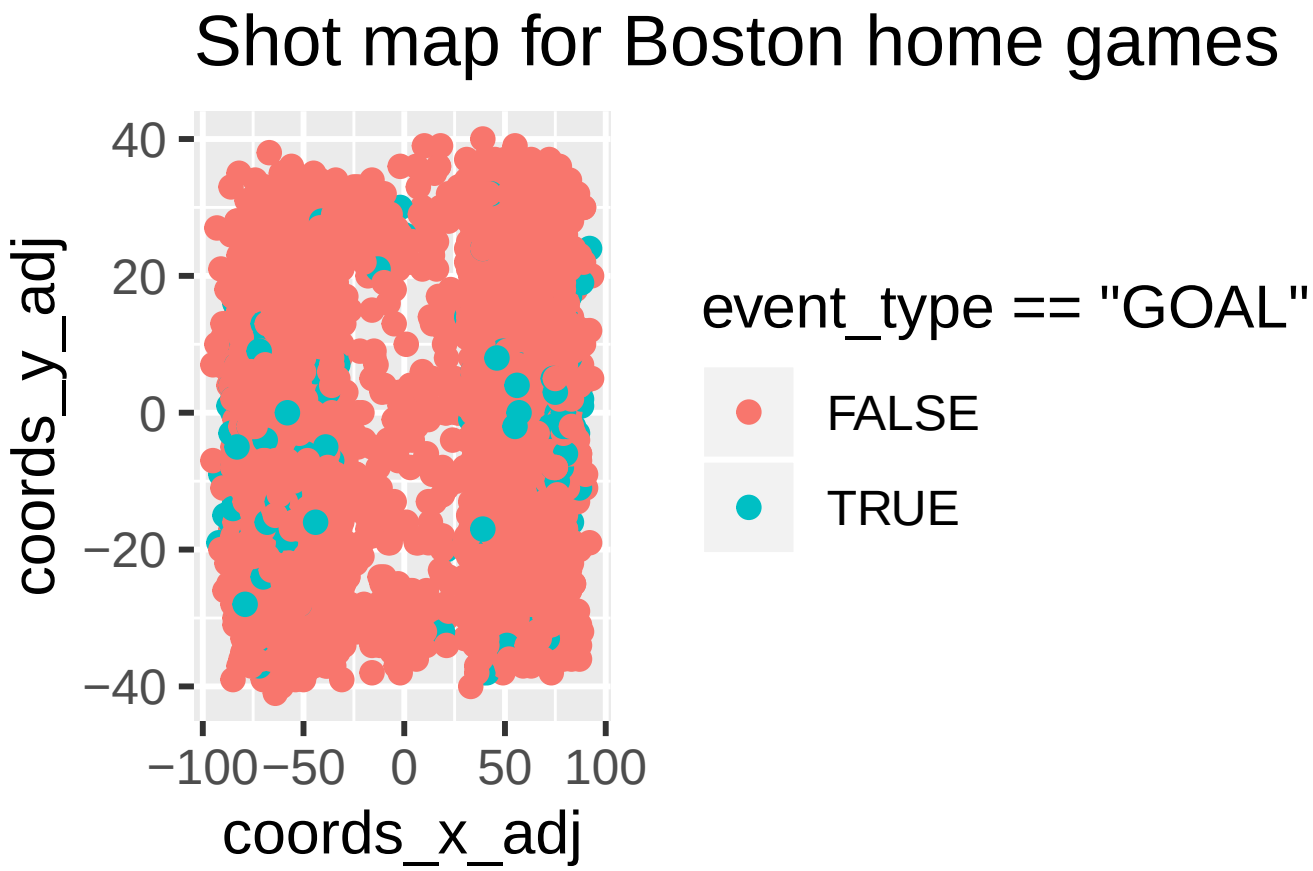
Adjusted coordinates

```
sample_game <- pbp_data %>% filter(game_id == 2017020636)
ggplot(sample_game, aes(x = coords_x_adj, y = coords_y_adj,
                        colour = event_team)) +
  geom_point()
```



Shot maps (team-season)

```
boston_home <- pbp_data %>% filter(home_team == "BOS")  %>%
  mutate(is_boston_shot = event_team == "BOS")
ggplot(boston_home, aes(x = coords_x_adj, y = coords_y_adj, colo
  geom_point() +
  labs(title = "Shot map for Boston home games")
```



Expected goal model

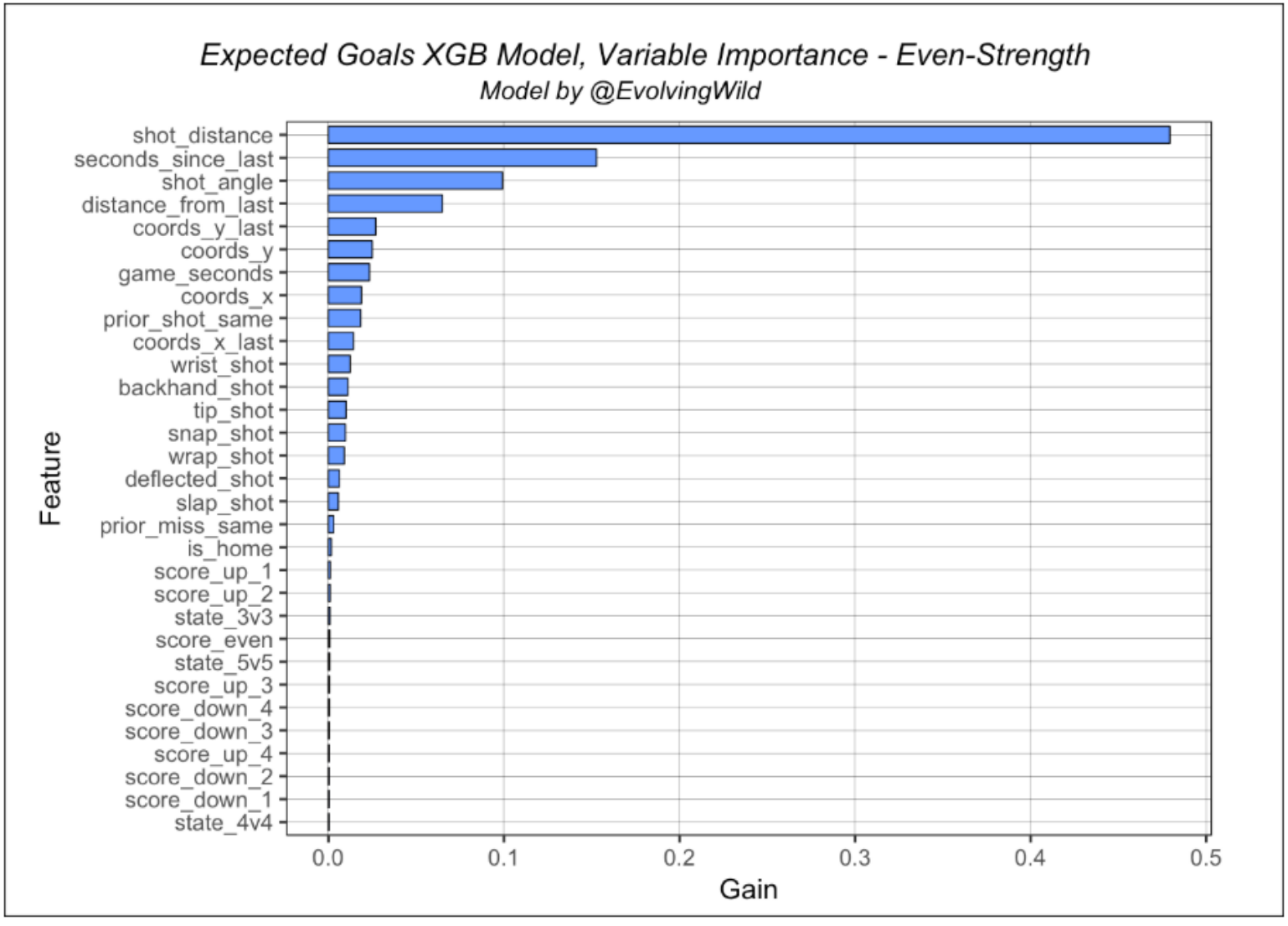
```
library(broom)
fit_1 <- glm(event_type == "GOAL" ~ event_distance +
               event_angle + event_detail ,
             family = "binomial", data = pbp_data)
tidy(fit_1)
```

```
## # A tibble: 9 x 5
##   term                    estimate std.error statistic   p.value
##   <chr>                      <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)              -1.16    0.0329      -35.4  6.14e-275
## 2 event_distance           -0.0416  0.000646    -64.3  0.
## 3 event_angle              -0.0144  0.000446    -32.3  4.38e-229
## 4 event_detailDeflected    -0.154   0.0556       -2.77 5.68e-  3
## 5 event_detailSlap          0.345   0.0423        8.15 3.49e- 16
## 6 event_detailSnap          0.397   0.0370       10.7  7.93e- 27
## 7 event_detailTip-In       -0.175   0.0400       -4.38 1.21e-  5
## 8 event_detailWrap-around  -0.447   0.101        -4.41 1.03e-  5
## 9 event_detailWrist         0.234   0.0312        7.51 5.86e- 14
```

A better model

https://rpubs.com/evolvingwild/395136/

Even-Strength



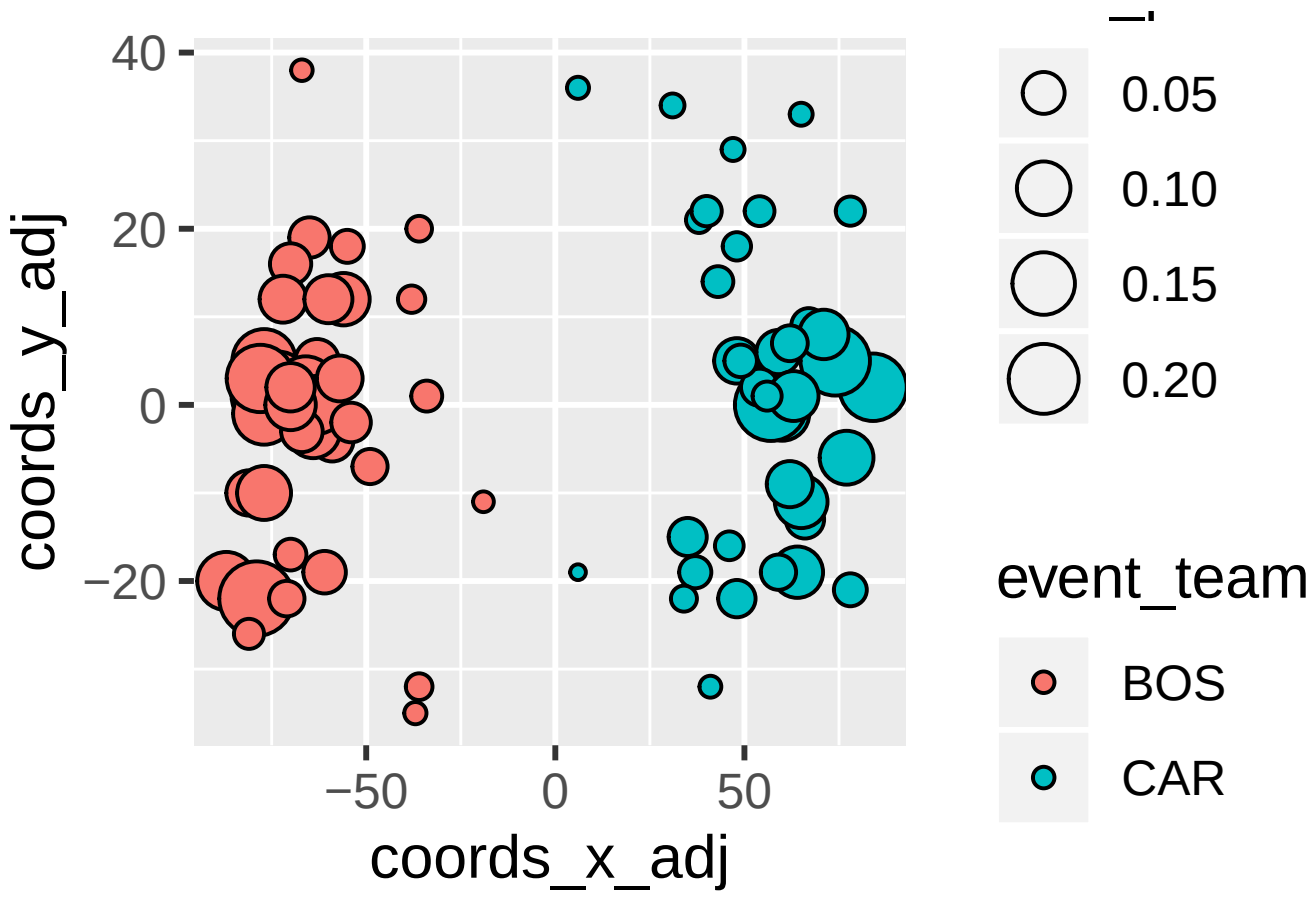
Sample shot

```
pbp_data %>% slice(5)
```

```
##     season    game_id  game_date session event_index game_period
## 1 20172018 2017020001 2017-10-04       R          36           1
##   game_seconds event_type home_team away_team home_skaters away_skaters
## 1          106       SHOT       WPG       TOR            5            5
##   home_score away_score event_detail event_team event_player_1
## 1          0          0        Wrist        TOR      ERIC.FEHR
##   event_player_2 coords_x coords_y home_goalie       away_goalie
## 1           <NA>       79       -2 STEVE.MASON FREDERIK.ANDERSEN
##   event_circle event_distance event_angle shot_prob coords_x_adj
## 1            9           10.2        11.3 0.1566066           79
##   coords_y_adj
## 1           -2
```

A better shot map

```
boston_game <- pbp_data %>% filter(game_id == 2017020636)
ggplot(boston_game, aes(x = coords_x_adj, y = coords_y_adj, size
  geom_point(pch = 21, colour = "black")
```



Game-level

```
boston_game %>%
  filter(event_team == "BOS") %>%
  summarise(bos_xg = sum(shot_prob),
            bos_g = sum(event_type == "GOAL"))
```

```
##     bos_xg bos_g
## 1 2.648966     7
```

```
boston_game %>%
  filter(event_team == "CAR") %>%
  summarise(bos_xg = sum(shot_prob),
            bos_g = sum(event_type == "GOAL"))
```

```
##     bos_xg bos_g
## 1 1.864243     1
```

Does the model work?

```
tab_check <- pbp_data %>%
  mutate(shot_prob_cat = cut(shot_prob, 10)) %>%
  group_by(shot_prob_cat) %>%
  summarise(ave_exp_goals = sum(shot_prob),
            ave_act_goals = sum(event_type == "GOAL"),
            n_shots = n())

tab_check
```

```
## # A tibble: 10 x 4
##    shot_prob_cat     ave_exp_goals ave_act_goals n_shots
##    <fct>                     <dbl>         <int>   <int>
##  1 (0.000708,0.0994]         6486.          6935  178259
##  2 (0.0994,0.197]            4145.          4223   30569
##  3 (0.197,0.295]             1978.          1860    8297
##  4 (0.295,0.392]              793.           727    2408
##  5 (0.392,0.49]               255.           245     586
##  6 (0.49,0.588]               189.           187     355
##  7 (0.588,0.685]              166.           161     262
##  8 (0.685,0.783]              143.           141     196
##  9 (0.783,0.881]              137.           142     164
## 10 (0.881,0.979]              261.           281     284
```

Hosmer Lemeshow Goodness of Fit

Hosmer Lemeshow Goodness of Fit

```
tab_check <- tab_check %>%
  mutate(diff_sq = (ave_exp_goals - ave_act_goals)^2/
           ((ave_exp_goals)*(1-ave_exp_goals/n_shots)))

tab_check
```

```
## # A tibble: 10 x 5
##    shot_prob_cat     ave_exp_goals ave_act_goals n_shots diff_sq
##    <fct>                     <dbl>         <int>   <int>   <dbl>
##  1 (0.000708,0.0994]         6486.          6935  178259 32.2
##  2 (0.0994,0.197]            4145.          4223   30569  1.69
##  3 (0.197,0.295]             1978.          1860    8297  9.21
##  4 (0.295,0.392]              793.           727    2408  8.29
##  5 (0.392,0.49]               255.           245     586  0.657
##  6 (0.49,0.588]               189.           187     355  0.0529
##  7 (0.588,0.685]              166.           161     262  0.491
##  8 (0.685,0.783]              143.           141     196  0.160
##  9 (0.783,0.881]              137.           142     164  1.02
## 10 (0.881,0.979]              261.           281     284 18.7
```

Hosmer Lemeshow

```
hm_test <- tab_check %>%
  summarise(test_stat = sum(diff_sq))
hm_test
```

```
## # A tibble: 1 x 1
##   test_stat
##       <dbl>
## 1      72.5
```

```
1-pchisq(hm_test$test_stat, df = 8, lower.tail = TRUE)
```

```
## [1] 1.579181e-12
```

Player metrics

```
season_2018 <- pbp_data %>%
  filter(season == 20172018) %>%
  group_by(event_player_1, season) %>%
  summarise(n_goals_18 = sum(event_type == "GOAL"),
            n_xGs_18 = sum(shot_prob),
            n_shots_18 = n()) %>%
  filter(n_shots_18 >= 100) %>%
  select(-season)

season_2019 <- pbp_data %>%
  filter(season == 20182019) %>%
  group_by(event_player_1, season) %>%
  summarise(n_goals_19 = sum(event_type == "GOAL"),
            n_xGs_19 = sum(shot_prob),
            n_shots_19 = n()) %>%
  filter(n_shots_19 >= 100) %>%
  select(-season)
```

Player metrics

```
season_combine <- season_2018 %>% inner_join(season_2019)
head(season_combine)
```

```
## # A tibble: 6 x 7
## # Groups:   event_player_1 [6]
##   event_player_1 n_goals_18 n_xGs_18 n_shots_18 n_goals_19 n_xGs_19
##   <chr>               <int>    <dbl>      <int>      <int>    <dbl>
## 1 AARON.EKBLAD           16    12.3         283         13     9.94
## 2 ADAM.HENRIQUE          24    23.2         212         18    17.2
## 3 ADAM.LARSSON            4     4.01        130          3     3.87
## 4 ADAM.PELECH             3     4.03        150          5     4.58
## 5 ADRIAN.KEMPE           16    11.8         161         12    11.3
## 6 ALEC.MARTINEZ           9     4.96        152          4     4.48
## # ... with 1 more variable: n_shots_19 <int>
```

```
library(corrplot)
```

Player metrics

```
cor_players <- cor(season_combine[,2:7])
corrplot(cor_players, method = "number")
```

Score effects

```
pbp_data <- pbp_data %>%
  mutate(score_diff = ifelse(event_team == home_team,
                             home_score - away_score,
                             away_score - home_score),
         score_diff_cat = case_when(score_diff <= -1 ~ "Down",
                                    score_diff == 0 ~ "Tied",
                                    score_diff >= 1 ~ "Up"),
         is_goal = event_type == "GOAL")

pbp_data %>%
  group_by(score_diff_cat) %>%
  summarise(ave_goal = mean(is_goal),
            ave_distance = mean(event_distance, na.rm = TRUE),
            ave_Xg = mean(shot_prob))
```

```
## # A tibble: 3 x 4
##   score_diff_cat ave_goal ave_distance ave_Xg
##   <chr>             <dbl>        <dbl>  <dbl>
## 1 Down             0.0611         36.4 0.0615
## 2 Tied             0.0639         36.2 0.0623
## 3 Up               0.0791         37.0 0.0752
```