cbbdata

```
# persistent log-in
cbbdata::cbd_login()
```

```
API Key set!
```

```
library(cbbdata)
library(tidyverse)
library(tidymodels)
library(caret)
```

```
duke_data <- cbd_torvik_game_factors() %>%
  filter(team == 'Duke')
```

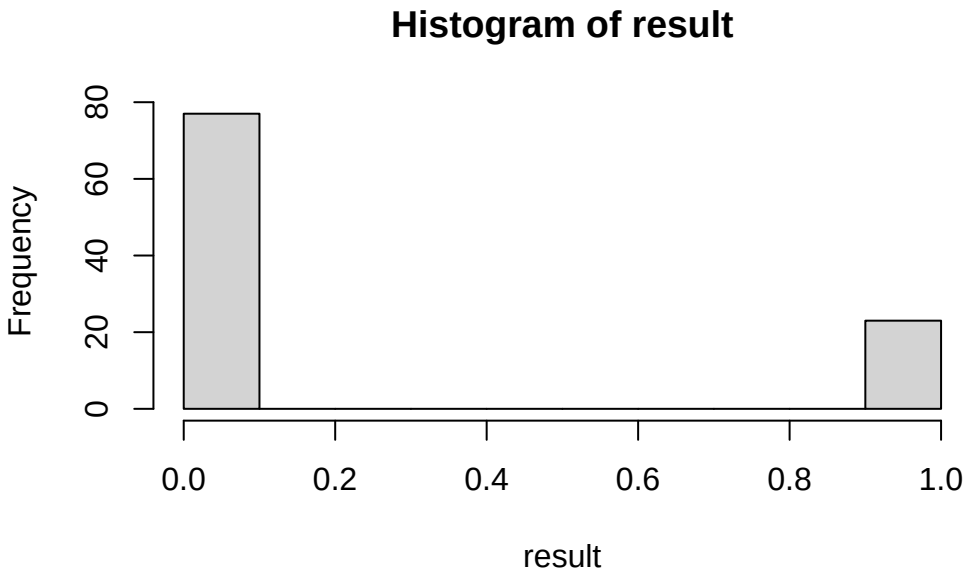
```
duke_unc_game <- cbd_torvik_season_prediction('Duke',2024) %>%
  filter(opp == 'North Carolina', game_location == 'A')
duke_unc_game
```

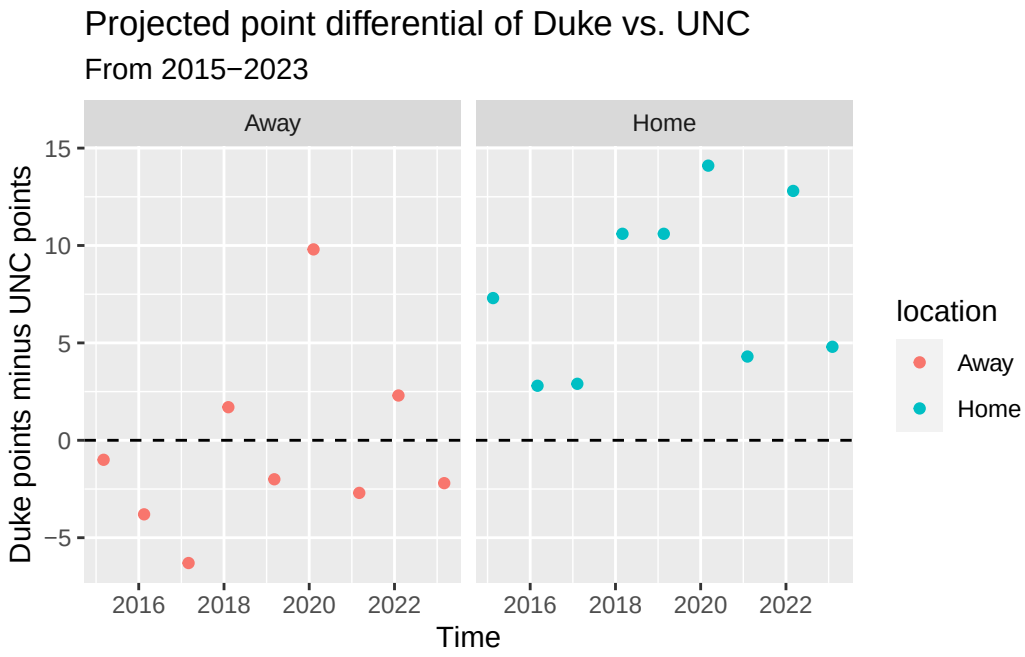
```
        date team            opp game_location    tempo      ppp  pts  win_per
1 2024-02-03 Duke North Carolina             A 71.05691 1.037392 73.7 22.89543
  did_win simulate_date year
1   FALSE    2024-02-03 2024
```

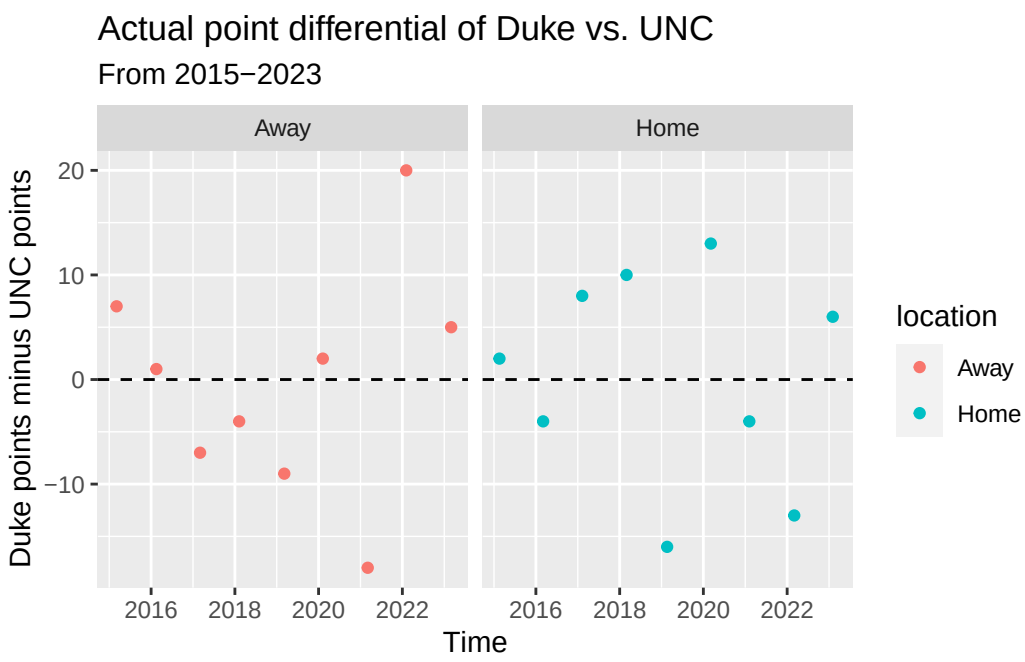
```
unc_duke_game <- cbd_torvik_season_prediction('North Carolina',2024) %>%
  filter(opp == 'Duke', game_location == 'H')
unc_duke_game
```

```
       date           team  opp game_location    tempo      ppp  pts  win_per
1 2024-02-03 North Carolina Duke             H 71.05691 1.152916 81.9 77.10457
  did_win simulate_date year
1    TRUE    2024-02-03 2024
```

```r
result <- rbinom(100, 1, 0.2346)
hist(result)
```





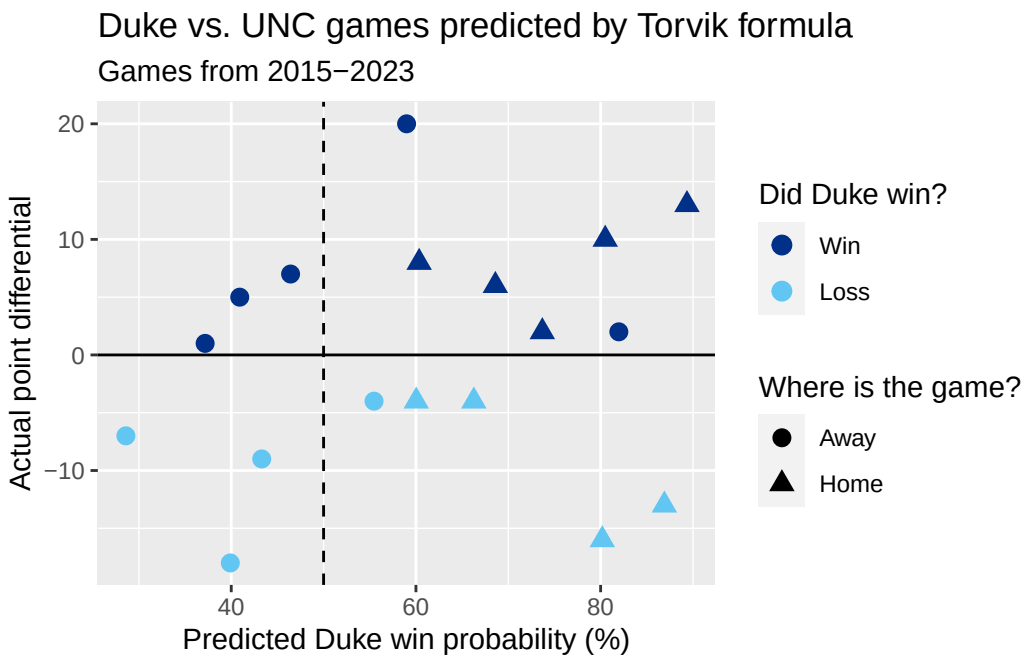



```
real_results <- real_results %>%
  mutate(bin_win = as.factor(if_else(diff > 0, 1,0)))
```

```r
full_predictions <- full_predictions %>%
  mutate(bin_win = as.factor(if_else(diff > 0, 1,0)))

filtered_predictions <- full_predictions %>%
  filter(team == "Duke")

confusionMatrix(data=filtered_predictions$bin_win, reference = real_results$bin_win, posit
```

```
Warning in confusionMatrix.default(data = filtered_predictions$bin_win, :
Levels are not in the same order for reference and data. Refactoring data to
match.
```

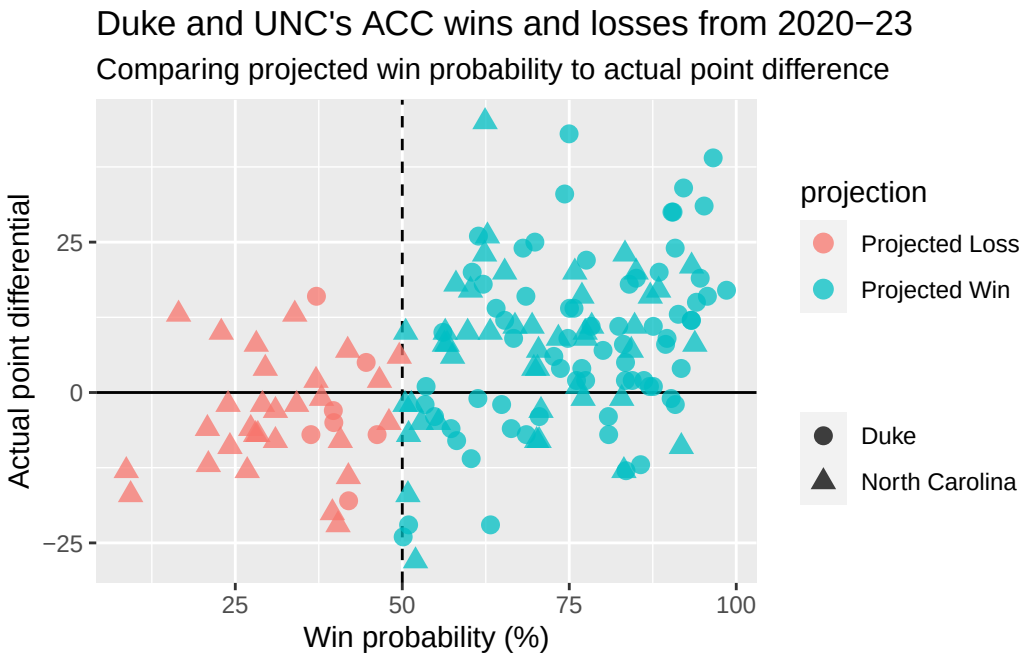
```
Confusion Matrix and Statistics

          Reference
Prediction 0 1
         0 3 3
         1 5 7

               Accuracy : 0.5556
                 95% CI : (0.3076, 0.7847)
    No Information Rate : 0.5556
    P-Value [Acc > NIR] : 0.5966

                  Kappa : 0.0769

 Mcnemar's Test P-Value : 0.7237

            Sensitivity : 0.7000
            Specificity : 0.3750
         Pos Pred Value : 0.5833
         Neg Pred Value : 0.5000
             Prevalence : 0.5556
         Detection Rate : 0.3889
   Detection Prevalence : 0.6667
      Balanced Accuracy : 0.5375

       'Positive' Class : 1
```




```
both_teams <- full_join(duke_final,unc_final)
```

```
Joining with `by = join_by(date, team.x, opp.x, game_location, tempo.x, ppp,
pts, win_per, did_win, simulate_date, year.x, type, team.y, conf, opp.y,
opp_conf, loc, result, pts_scored, pts_allowed, adj_o, adj_d, off_ppp, off_efg,
off_to, off_or, off_ftr, def_ppp, def_efg, def_to, def_or, def_ftr, game_score,
season, tempo.y, game_id, coach, opp_coach, year.y, diff, projection)`
```

```
both_teams %>%
  ggplot(aes(x = win_per, y = diff, color = projection, shape = team.x)) +
  #facet_wrap(~year.y) +
  geom_hline(yintercept =  0, linetype = 1) +
  geom_vline(xintercept = 50, linetype = 2) +
  geom_point(size = 3, alpha = 0.75) +
  #geom_rect(xmin = 45, xmax = 55, ymin = -1000, ymax = 1000, alpha = 0, color = "White",l
  labs(title = "Duke and UNC's ACC wins and losses from 2020-23",
       subtitle = "Comparing projected win probability to actual point difference",
       shape = "",
       x = "Win probability (%)",
       y = "Actual point differential")
```




```
conf_mat_data_combo <- both_teams %>%
  filter(projection != "Too close to call") %>%
  mutate(projection_bin = fct_relevel(as.factor(if_else(projection == "Projected Win", 1,
         result_bin = fct_relevel(as.factor(if_else(result == "W", 1, 0))),"1")
conf_matrix_combo <- confusionMatrix(data=conf_mat_data_combo$projection_bin, reference =
```

```
Warning in confusionMatrix.default(data = conf_mat_data_combo$projection_bin, :
Levels are not in the same order for reference and data. Refactoring data to
match.
```

```
conf_matrix_combo
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 25 11
         1 33 85

               Accuracy : 0.7143
```

```
               95% CI : (0.636, 0.7841)
    No Information Rate : 0.6234
    P-Value [Acc > NIR] : 0.011316

                  Kappa : 0.3421

 Mcnemar's Test P-Value : 0.001546

            Sensitivity : 0.8854
            Specificity : 0.4310
         Pos Pred Value : 0.7203
         Neg Pred Value : 0.6944
             Prevalence : 0.6234
         Detection Rate : 0.5519
   Detection Prevalence : 0.7662
      Balanced Accuracy : 0.6582

       'Positive' Class : 1
```